Systems and Control: A Computational Approach to Modeling, Analysis, and Design

Second Edition

Stanislaw H. Żak
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907

August 26, 2011

To
J.M.J.

Contents

Preface

Objectives

This book presents an introduction to the modeling, analysis, and control of linear and nonlinear dynamical systems. Its objective is to familiarize the reader with the basics of dynamical system theory while, at the same time, equipping him with the tools necessary for control system design. The emphasis is on design in order to show how dynamical system theory fits into practical applications. In particular there is a chapter on the design of proportional-integral-derivative (PID) controllers as well as a chapter devoted to the model-based predictive control (MPC) design. In addition there is a chapter presenting a powerful control design tool—the linear matrix inequalities (LMIs). The above approaches to the controller synthesis are the "bread and butter" of control design in industry. Even in chapters that are concerned with analysis, design examples and problems are included wherever possible.

The scope of the book is broad in order to show the multidisciplinary role of dynamics and control. In particular, neural networks, fuzzy systems, and genetic algorithms are presented and an introduction to chaotic systems is provided. Classical methods and techniques of post-modern control engineering, like fuzzy logic, neural networks, and genetic algorithms are presented in a unified fashion. This book aims to demonstrate how these tools supplement the classical ones.

Coverage

Whenever one wishes to analyze a real world system, the first order of business is the construction of a mathematical model of the system. In this book, we discuss mathematical modeling of systems from mechanical and electrical engineering, as well as from physics and biology. We also construct models of dynamical systems using linguistic descriptions that employ IF-THEN rules.

Two main types of dynamical systems are common in applications: those for which the time variable is discrete and those for which the time variable is continuous. When the time variable is discrete, the dynamics of such systems are usually modeled using difference equations. In the case when the time is continuous, ordinary differential equations are frequently chosen for modeling purposes. Both types of models are considered in the book.

Nonlinear systems are emphasized to acknowledge the critical role that nonlinear phenomena play in science and technology. The models presented are the ones that can be used to design controllers. They are constructed from the control engineering point of view. This book is an invitation to the control engineering of complex, nonlinear, dynamical systems using multidisciplinary tools.

Lyapunov's stability theory is used in the analysis of dynamical system models. The objective of the stability analysis is to determine the system behavior without solving the differential, or difference, equations modeling the system. In fact, the Lyapunov theory is used as a unifying medium for the different types of dynamical systems analyzed. In particular, optimal, sliding mode, fuzzy, and chaotic controllers, as well as unknown input observers are all constructed with the aid of the Lyapunov method.

Changes in the Second Edition

Six new chapters were added to this edition. Three chapters are concerned with controller design methods. These chapters make this text more attractive for the reader interested in applications. A chapter on unknown input observers was added to account for recent interest in applying those types of observers in fault detection. A separate chapter collects models of the different dynamical systems that are being referred to throughout the text. Finally, a chapter on linear time-varying (LTV) system was added to make this book also suitable for a first graduate course on dynamical systems. On the other hand, a number of sections in different chapters containing esoteric material at the advanced graduate level were deleted.

Intended Audience

This text can be used in a variety of ways. It is suitable for an introductory graduate-level course on systems and control. The text can also be used in an introductory course on nonlinear systems. Finally, it can be used in an introductory course to modern automatic control. No special background is necessary to follow this text beyond basic differential equations and elements of linear algebra. A one semester undergraduate junior level course on feedback systems is all that is needed to enjoy this text. The book is also suitable for self-study. To accomplish this goal, the required mathematical techniques and terminology are reviewed in the Appendix. Depending on the background, the reader may refer to the material in the Appendix as needed.

Resources

In addition to worked examples, the book also contains exercises at the end of each chapter. A number of solved problems, as well as the exercises, require the use of a computer. Although there are now many quality software packages available that could be used with this book, MATLAB[1] was chosen because it appears to be the most widely used math-tools program in engineering circles, and it is also popular among science students. The solutions manual, with complete solutions to all end-of-chapter exercises, is available from the publisher to the instructors who adopt this book in their courses.

There is a companion web-site for this book, where the current errata are available as well as details of solutions of examples solved in the text along with MATLAB programs used to obtain the solutions. The URL for the web-site is

`http://cobweb.ecn.purdue.edu/~zak/systems/Index.html`

[1]MATLAB is a registered trademark of the MathWorks, Inc., 24 Prime Park Way, Natick, MA 01760-1500. Phone: 508-647-7000, fax: 508-647-7001, e-mail: info@mathworks.com, http://www.mathworks.com

Chapter 1

Vocabulary and Description of Dynamical Systems

> It's in the preface [of *In Retrospect*]. I know the lines. I know what they say:
> "We shall not cease from exploration
> And the end of all our exploring
> Will be to arrive where we started
> And know the place for the first time."
>
> Now, I haven't ceased from exploring, but I'm a little farther along than I was fifteen or thirty years ago, and I think I see a little more clearly—not as clearly as I will a few years hence or before I die—but I see events more clearly today than I saw five, ten, thirty years ago.
> Robert McNamara in [202, pages 344, 345]

1.1 What Is a System?

A *system* is a collection of interacting components. An electric motor, an airplane, as well as a biological unit such as the human arm are examples of systems. A system is characterized by two properties. They are:

1. the interrelations between the components that are contained within the system,

2. the system boundaries that separate the components within the system from the components outside.

The system boundaries can be real or imagined. They are elastic in the sense that we may choose, at any stage of the system analysis, to consider only a part of the original system as a system on its own. We call it a *subsystem* of the original system. On the other hand, we may decide to expand the boundaries of the original system to include new components. In Figure 1.1, we represent system's boundaries using a box.

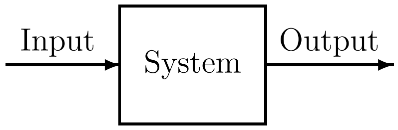


Figure 1.1: Representation of a system.

The interactions between the system components may be governed, for example, by physical, biological, or economical laws. In dealing with systems, we are interested in the effects of external quantities upon the behavior of the system quantities. We refer to the external quantities acting on the system as the *inputs* to the system. The *output* of the system is its response to the input. The outputs are internal quantities available for measurement.

A system is a *continuous-time* system if it accepts continuous-time signals as its inputs and produces continuous-time signals as its outputs. We use the lowercase italic $u(t)$ to denote the input of the continuous-time single-input system, where the time t is assumed to range from $-\infty$ to ∞. For the multi-input continuous-time system, we use the boldface italic $\boldsymbol{u}(t)$ to denote its input, where $\boldsymbol{u}(t)$ is an $m \times 1$ column vector, that is,

$$\boldsymbol{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix}.$$

Similarly, the output of the single-output continuous-time system is denoted as $y(t)$, while the output of the multi-output continuous-time system is denoted as

$$\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_p(t) \end{bmatrix},$$

that is, the output $\boldsymbol{y}(t)$ is a column vector composed of p components.

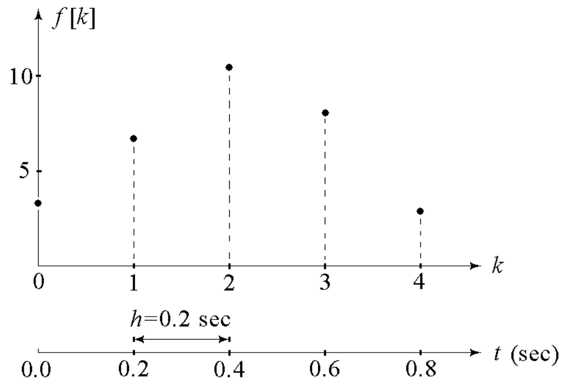


Figure 1.2: An example of a discrete-time signal with the sampling period $h = 0.2$ sec.

A system is a *discrete-time* system if it accepts discrete-time signals as its inputs and produces discrete-time signals as its outputs. An example of a discrete-time signal is shown in Figure 1.2.

A discrete-time signal can be viewed as a sequence of points obtained from measurements at successive times. Thus a discrete-time sinal has a natural temporal ordering. We consider only the case where all discrete-time signals have the same sampling period denoted h.

We use the lowercase italic $u[k]$ to denote the input of the discrete-time single-input system, where k denotes discrete time instant and is assumed to range from $-\infty$ to ∞. Thus,

$$u[k] := u(kh),$$

where the symbol ":=" denotes arithmetic assignment. Thus the statement "$u[k] := u(kh)$" should be interpreted to mean "$u[k]$ becomes $u(kh)$." For the multi-input discrete-time system, we use the boldface italic $\boldsymbol{u}[k]$ to denote its input, where $\boldsymbol{u}[k] := \boldsymbol{u}(kh)$ is an $m \times 1$ vector, that is,

$$\boldsymbol{u}[k] = \begin{bmatrix} u_1[k] \\ u_2[k] \\ \vdots \\ u_m[k] \end{bmatrix}.$$

Similarly, the output of the single-output discrete-time system will be denoted as $y[k]$, where $y[k] := y(kh)$. The output of the multi-output discrete-time system will be denoted as

$$\boldsymbol{y}[k] = \begin{bmatrix} y_1[k] \\ y_2[k] \\ \vdots \\ y_p[k] \end{bmatrix},$$

that is, the output $\boldsymbol{y}[k]$ is a column vector composed of p components.

1.1.1 Memoryless, Causal, and Lumped Systems

A system is a *memoryless system* if its output at time t_0 depends only on the input applied at t_0 and is independent of the input applied before t_0, that is, current output of a memoryless system depends only on its current input; it is independent of past inputs. An example of memoryless system is a circuit consisting of only resistors.

In a *causal* or *non-anticipatory* system its current output may depend on current and past inputs but not on future inputs. In a *non-causal* system, its current output depends on future input. Such a system can predict or anticipate what inputs will be applied in the future. Physical systems have no such capability, that is, physical systems are causal.

To proceed, we consider, following Kailath [168, p. 63], a simple system consisting of a capacitor and a voltage source as shown in Figure 1.3. The input u is a current through the

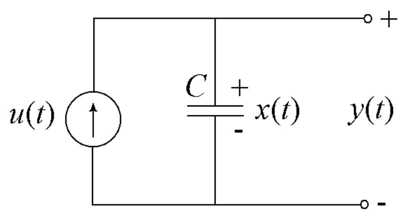


Figure 1.3: A circuit where the the past input from $-\infty$ up to t affects the current output at time t.

capacitor and the output y is a voltage x across the capacitor. The relation between voltage and current for a capacitor has the form,

$$y(t) = \frac{1}{C}\int_{-\infty}^{t} u(\tau)d\tau.$$

It follows from the above that the system output is affected by its past input from $-\infty$ to time t. Tracking the past inputs may be impractical if not impossible. To alleviate the problem, the concept of state is introduced that we discuss next.

The condition or the state of the system are described by the *state variables* denoted x_i. The state variables provide the information that together with the knowledge of the system inputs enable us to determine the future state of the system. In other words, "A *dynamical system* consists of a set of possible states, together with a rule that determines the present state in terms of past states" [7, page 1].

Definition 1.1 [55, p. 6] *The state $\boldsymbol{x}(t_0)$ of a system at time t_0 is the information at t_0 that, together with the input $\boldsymbol{u}(t)$, for $t \geq t_0$, determines uniquely the output $\boldsymbol{y}(t)$ for all $t \geq t_0$.*

◇

Returning to the example above, we select the voltage across the capacitor as the system state, which is also its output. Then, we write

$$\begin{aligned} y(t) &= x(t) \\ &= \frac{1}{C}\int_{-\infty}^{t} u(\tau)d\tau \\ &= \frac{1}{C}\int_{-\infty}^{t_0} u(\tau)d\tau + \frac{1}{C}\int_{t_0}^{t} u(\tau)d\tau \\ &= x(t_0) + \frac{1}{C}\int_{t_0}^{t} u(\tau)d\tau. \end{aligned}$$

Thus, if we know the initial state $x(t_0)$ and the current input $u(t)$, then the past input is irrelevant in the process of calculating the output $y(t)$ for $t \geq t_0$ because the initial state "summarizes" the past input. Kailath [168, p. 63] compares this with a situation when one is faced with a new expenditure; then it is the current state of one's bank balance and not the many ways it could have reached its present value that is relevant to the issue at hand.

In summary, the voltage across the capacitor is

$$x(t) = x(t_0) + \frac{1}{C}\int_{t_0}^{t} u(\tau)d\tau.$$

Differentiating the above, we obtain equivalent representation in the form of a differential equation,

$$\dot{x}(t) = \frac{1}{C}u(t), \quad x(t_0) = x_0.$$

A system is a *lumped* system if the number of state variables is finite, that is, the state vector $\boldsymbol{x}$ is composed of finite number of components. A simple example of lumped system is the circuit shown in Figure 1.3. More examples of lumped systems are given in Section 2.4.

A system is *distributed* if its state consists of infinitely many components. We now give an example of a distributed system.

Example 1.1 [55, p. 7] Consider the unit-time delay system model of the form,

$$y(t) = u(t-1).$$

In this example, the output is the input delayed by one time unit. To determine $y(t)$ for $t > t_0$, we need the information about $u(t)$ on the time interval $[t_0 - 1,\ t_0]$. Thus the initial state of this time-delay system is the set of points, $\{u(t) : t \in [t_0 - 1,\ t_0]\}$. However, there are infinitely many points $u(t)$ on this interval. Thus, the above time-delay system is a distributed system because its state is infinitely dimensional.

In practice it is often not possible or too expensive to measure or determine the values of all of the state variables. Instead, only their subset or combination can be measured. The system quantities whose behavior can be measured or observed constitute the system's outputs.

1.1.2 Formulation of the Control Problem

In engineering applications, when dealing with dynamical systems, we are interested in specifying the system inputs that force the system states or outputs to behave with time in some pre-specified manner. That is, we are interested in *controlling* the system states or outputs. This is accomplished by means of a *controller* whose task is to produce the required system's inputs that in turn result in the desired system's outputs. An interconnection of the system and a controller is called a *control system.* In Figure 1.4 and 1.5, we show two different types of control systems that we discuss in the next section. Constructing a controller is a part of the *control problem.* The essential elements of the control problem, as described by Owens [260, page 181], are:

1. a dynamical system to be controlled,
2. a specified objective for the system,
3. a set of admissible controllers, and
4. a means of measuring the performance of any given control strategy to evaluate its effectiveness.

We now examine these elements one by one. The first step in the controller design procedure is the construction of a *truth model* of the dynamics of the process to be controlled. The truth model is a *simulation model* that includes all the relevant characteristics of the process. The truth model is too complicated for use in the controller design. Thus, we need to develop a simplified model that can be used to design a controller. Such a simplified model is labeled by Friedland [108] as the *design model*. The design model should capture the essential feature of the process. During the modeling process, properties of the system, physical constraints, and technical requirements should be taken into account. We discuss in more detail the process of constructing mathematical models of physical systems in Chapter 2.

The objective of a control system is to complete some specified task. This can be expressed as a combination of constraints on the output or state variables and limits on the time available to complete the control objective. For example, the objective of a controller might be to force the output, of a single output system, to settle within a certain percentage of the given value after pre-specified time.

Control input signals are obtained from physical devices capable of providing only a limited amount of energy. We term the class of controllers that can be considered for the given control design problem as the set of admissible controllers.

We can evaluate the performance of any given control law by visual inspection of the transient characteristics of the controlled system after the design is complete. But this method of assessment of the effectiveness of the given control strategy is inaccurate. We therefore try to quantify the means of evaluating performance in the form of a performance index or cost functional at the beginning of the design process. We construct a numerical index of performance whose value reflects the quality of any admissible controller in accomplishing the system objective. This performance index is assumed to have the property that its numerical value decreases as the quality of the controller increases. This is the reason why we sometimes refer to the performance index as the penalty functional. The "best" controller then is the one that produces the smallest value of the performance index. We call an admissible controller that simultaneously ensures the completion of the system objective and the minimization of the performance index an optimal controller for the system.

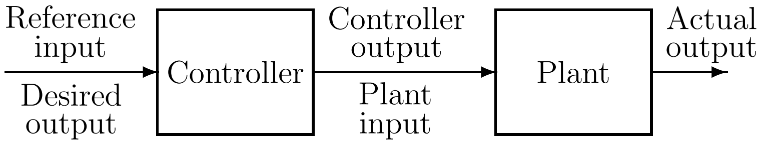


Figure 1.4: Open-loop control system.

1.1.3 Open-Loop Versus Closed-Loop

We distinguish between two types of control systems. They are:

- open-loop control systems, and
- closed-loop control systems.

An open-loop control system usually contains:

1. A process to be controlled, labeled *plant*,
2. The controlling variable of the plant, called the *plant input*, or just *input* for short,
3. The controlled variable of the plant, called the *plant output*, or just *output* for short,
4. A *reference input*, which dictates the desired value of the output,
5. A *controller* that acts upon the reference input in order to form the system input which would force the behavior of the output in accordance with the reference signal.

Note that the plant and controller themselves can be considered as systems on their own. A schematic representation of an open-loop system is depicted in Figure 1.4. In an open-loop control system the output has no influence on the input or reference signal. The controller operates without taking into account the output. Thus, the plant input is formed with no influence of the output. A household appliance such as an iron is a simple example of an open-loop control system. In this example, we consider the iron itself as the plant or process to be controlled. Here we set the temperature control mechanism for the desired temperature. This setting, for example very hot, is the reference input. We can view the potentiometer, inside the iron, as a controller, or a control actuator. It regulates the temperature of the iron by allowing the necessary amount of resistance to produce the desired temperature. The plant's output is the temperature of the heated iron.

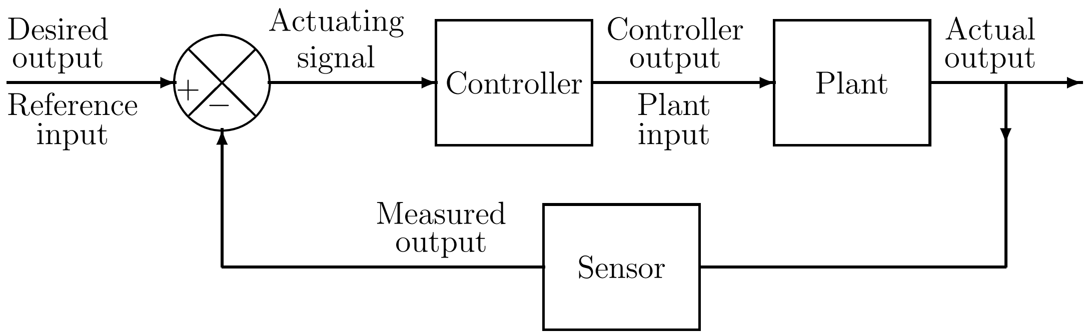


Figure 1.5: Closed-loop control system.

Not every control system is an open-loop system. Another type of a control system is a closed-loop system. We can convert an open-loop system into a closed-loop system by adding, to an open-loop system, the following components:

6. The *feedback loop* where the output signal is measured with a sensor and then the measured signal is fed back to the *summing junction*,

7. The summing junction, where the measured output signal is subtracted from the reference, command, input signal in order to generate an *error signal*, also labeled as an *actuating signal*.

A schematic of a closed-loop control system, or feedback system, is shown in Figure 1.5. In a closed-loop system the error signal causes an appropriate action of the controller, which in turn "instructs" the plant to behave in a certain way in order to approach the desired output, as specified by the reference input signal. Thus, in the closed-loop system, output information is fed back to the controller, and the controller then appropriately modifies the plant output behavior. A controller, also called a *compensator*, can be placed either in the *forward loop*, or in the *feedback loop*. Thus, a central component of the closed-loop system is feedback. Norbert Wiener defined feedback in 1954 as follows: "Feedback is a method of controlling a system by reinserting into it the results of its past performance." [175, p. 439].

If the goal of the controller is to maintain the output at a constant value, then we have a *regulator* control system. A "centrifugal speed regulator", or the "flyball governor," commonly known as Watt's governor, is an example of a regulator in a closed-loop system. A schematic of Watt's governor is shown in Figure 1.6.

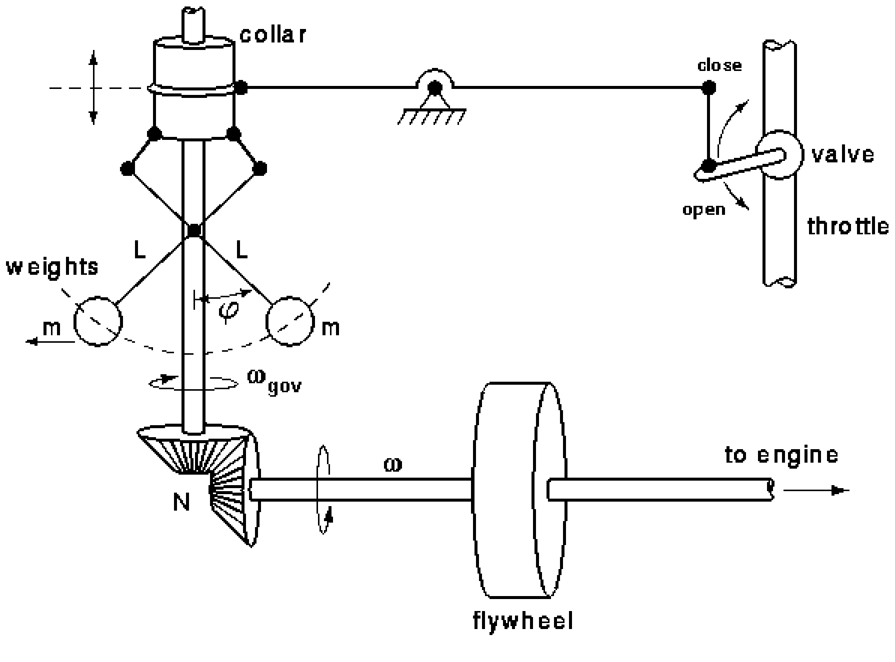


Figure 1.6: A simplified schematic of the flyball governor.

The task of the centrifugal speed regulator is to control setting of a throttle valve automatically to maintain the desired engine speed. The nominal speed is set by adjusting the valve in the throttle. As the engine speeds up the weights are thrown outwards, the throttle partially closes and the engine slows down. As the engine slows down below its nominal speed, the throttle is opened up and the engine gains its speed. The operation of an engine equipped with a centrifugal speed regulator can be represented using the block diagram of Figure 1.5. The reference input to the closed-loop system is the desired engine speed. The desired engine speed is obtained by appropriately setting the valve. The system output is the actual engine speed. The flyball governor is our controller, while the engine represents the plant. The control signal generated by the controller is the setting of the throttle valve.

We mention at this juncture that Watt's application of the centrifugal speed governor in 1788, to regulate the speed of the steam engine, marks the starting point for the development of automatic control, and in particular feedback control, as a science [226]. Before 1788, steam engines were controlled manually using the throttle valve. The flyball governor spread widely and quickly. It has been estimated that by 1868 there were about 75,000 flyball governors working in England alone [226]. The flyball governor is credited for the hatching of the industrial revolution. We devise a mathematical model of this revolutionary invention in Subsection 2.4.1, and in Section 7.7 we perform a quantitative analysis of its dynamical behavior.

If the goal of the controller is to force the output to follow a desired trajectory, then we have a *servomechanism* or *tracking* control system. As we mentioned before, to design a controller, we need first to analyze the plant quantitatively. The analysis requires a mathematical or linguistic description of the interrelations between the system quantities themselves as well as the interrelations between system quantities and system inputs.

1.2 Linearity

It follows from Definition 1.1 that a system accepts the input, $\boldsymbol{u}(t)$, and the initial state, $\boldsymbol{x}(t_0)$, to produce its output, $\boldsymbol{y}(t)$. We view a system as an operator that acts on two inputs, $\boldsymbol{u}(t)$ and $\boldsymbol{x}(t_0)$ to produce an output, $\boldsymbol{y}(t)$. We use the symbol L to represent a system. Then its operation can be described as

$$L(\boldsymbol{u}(t), \boldsymbol{x}(t_0)) = \boldsymbol{y}(t).$$

A system L is said to possess an *additivity property* if for any t_0 and any $\boldsymbol{x}_i(t_0)$ and $\boldsymbol{u}_i(t)$,

$t \geq t_0$, if

$$L(\boldsymbol{u}_i(t), \boldsymbol{x}_i(t_0)) = \boldsymbol{y}_i(t), \quad i = 1, 2,$$

then

$$L(\boldsymbol{u}_1(t) + \boldsymbol{u}_2(t), \boldsymbol{x}_1(t_0) + \boldsymbol{x}_2(t_0)) = \boldsymbol{y}_1(t) + \boldsymbol{y}_2(t).$$

We illustrate the above property in Figure 1.7.

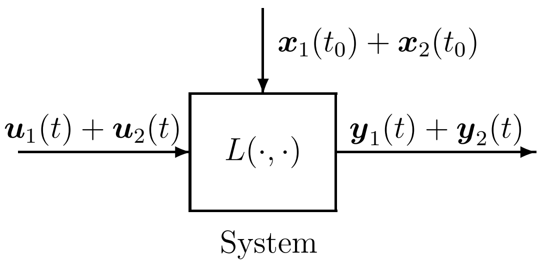


Figure 1.7: Illustration of the additivity property.

A system L is said to possess a *homogeneity property* if for any t_0 and any $\boldsymbol{x}(t_0)$ and $\boldsymbol{u}(t)$, $t \geq t_0$, if

$$L(\boldsymbol{u}(t), \boldsymbol{x}(t_0)) = \boldsymbol{y}(t),$$

then

$$L(c\boldsymbol{u}(t), c\boldsymbol{x}(t_0)) = c\boldsymbol{y}(t),$$

where c is real or complex constant. We illustrate the above property in Figure 1.8.

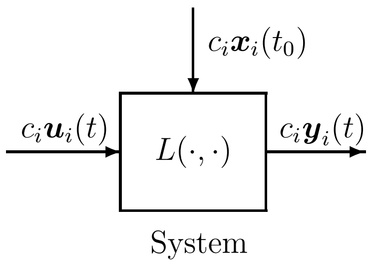


Figure 1.8: Illustration of the homogeneity property.

A system L is said to possess a *superposition property* if for any t_0 and any $\boldsymbol{x}_i(t_0)$ and $\boldsymbol{u}_i(t)$, $t \geq t_0$, if

$$L(\boldsymbol{u}_i(t), \boldsymbol{x}_i(t_0)) = \boldsymbol{y}_i(t), \quad i = 1, 2,$$

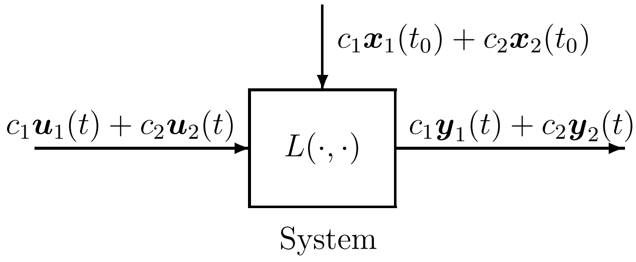


Figure 1.9: Illustration of the superposition property.

then

$$L(c_1\boldsymbol{u}_1(t) + c_2\boldsymbol{u}_2(t), c_1\boldsymbol{x}_1(t_0) + c_2\boldsymbol{x}_2(t_0)) = c_1\boldsymbol{y}_1(t) + c_2\boldsymbol{y}_2(t),$$

where c_1 and c_2 are real or complex constants. We illustrate the above property in Figure 1.9.

Note that the superposition property is a combination of the additivity and homogeneity properties.

Definition 1.2 *A system is said to be linear if it possesses the superposition property. A system is said to be nonlinear if it does not have the superposition property.*

◇

For an informative discussion on some subtleties in the definition of linearity, we recommend Kailath [168, Section 1.1].

If $\boldsymbol{u}(t) = \mathbf{0}$ for $t \geq 0$, then the system's output will be produced exclusively by the initial condition $\boldsymbol{x}(t_0)$. We call such an output the *zero-input response* and denote it as $\boldsymbol{y}_{zi}$.

If, on the other hand, the initial state $\boldsymbol{x}(t_0) = \mathbf{0}$, then the output will be produced exclusively by the input. This output is called the *zero-state response* and is denoted $\boldsymbol{y}_{zs}$.

If the system is linear, then the additivity property implies that the total response, $\boldsymbol{y}(t)$ is the sum of the zero-input and zero-state responses, that is,

$$\boldsymbol{y}(t) = \boldsymbol{y}_{zi}(t) + \boldsymbol{y}_{zs}(t), \quad t \geq t_0.$$

For linear systems the superposition property holds for zero-state and zero-input responses separately as well. In our discussion here, for the sake of simplicity, we do not display the second argument in the operator L when we set zero initial state or zero input.

Now, if we set the initial state to zero, $\boldsymbol{x}(t_0) = \mathbf{0}$ and if

$$L(\boldsymbol{u}_j(t)) = \boldsymbol{y}_{zs,j}(t), \quad j = 1, 2,$$

then

$$L(c_1\boldsymbol{u}_1(t) + c_2\boldsymbol{u}_2(t)) = c_1\boldsymbol{y}_{zs,1}(t) + c_2\boldsymbol{y}_{zs,2}(t).$$

Similarly, if we set $\boldsymbol{u}(t) = \boldsymbol{0}$ for $t \geq t_0$ and if

$$L(\boldsymbol{x}_j(t_0)) = \boldsymbol{y}_{zi,j}(t), \quad j = 1, 2,$$

then

$$L(c_1\boldsymbol{x}_1(t_0) + c_2\boldsymbol{x}_2(t_0)) = c_1\boldsymbol{y}_{zi,1}(t) + c_2\boldsymbol{y}_{zi,2}(t).$$

1.3 Input-Output Description of Linear Systems

In this Section, we develop a mathematical description of the zero-state response of linear systems. We assume that the initial state is zero and so the system's output is exclusively due to the system input.

Definition 1.3 *We say that the system is relaxed at t_0 if its initial condition at t_0 is zero, that is, $\boldsymbol{x}(t_0) = \boldsymbol{0}$.*

◇

To proceed, we need certain facts about the impulse function, $\delta(t)$.

1.3.1 The Impulse Function and Its Properties

The impulse function, denoted $\delta(t)$, also called the Dirac function, is a signal of infinite amplitude, zero duration, and unity area. We can construct an impulse function as the limit of pulse functions

$$p_{\varepsilon_i}(t) = \frac{1}{\varepsilon_i}\left(1(t) - 1(t - \varepsilon_i)\right)$$

as $\varepsilon_i \to 0$, where $1(t)$ is the unit step function defined as

$$1(t) = \begin{cases} 1 & \text{if} \quad t \geq 0 \\ 0 & \text{if} \quad t < 0. \end{cases}$$

The pulse p_{ε_i} is illustrated in Figure 1.10. Note that the pulse functions have the following features as $\varepsilon_i \to 0$:

1. the amplitude approaches infinity,

2. the duration of the pulses approaches zero,

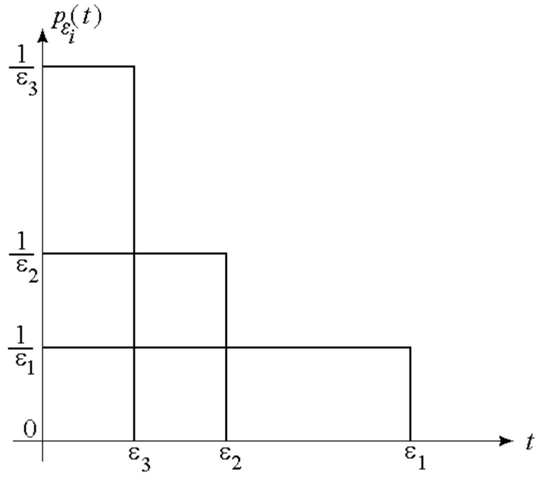


Figure 1.10: Generating the unit impulse function as the limit of the pulse functions.

3. the area under each pulse is constant; in our example the area equals unity.

The unit impulse function is defined as

$$\int_{-\infty}^{\infty} \delta(t)dt = 1 \quad \text{and} \quad \delta(t) = 0 \;\; \text{for} \;\; t \neq 0.$$

The above definition states that the area under the impulse function is constant. The area represents the strength of the impulse function. The impulse function of strength K is denoted $K\delta(t)$. Its graphical representation is depicted in Figure 1.11. The strength of the impulse is shown next to the arrow's head. The shifted impulse function of strength K is also shown in Figure 1.11. The unit impulse function can be thought of as a derivative of the unit step function, that is,

$$\delta(t) = \frac{d1(t)}{dt}.$$

The function shown in Figure 1.12 approaches the unit step function as $\varepsilon \to 0$. The function shown in Figure 1.13, which is the derivative of the function from Figure 1.12, approaches the unit impulse function as $\varepsilon \to 0$.

The impulse function has the *sifting property*,

$$\boxed{\int_{-\infty}^{\infty} f(t)\delta(t-a)dt = f(a)} \tag{1.1}$$

where $f(t)$ is a continuous function of time. It follows from the above that the impulse function sifts out everything except the value of f at $t = a$—hence the name of the property.

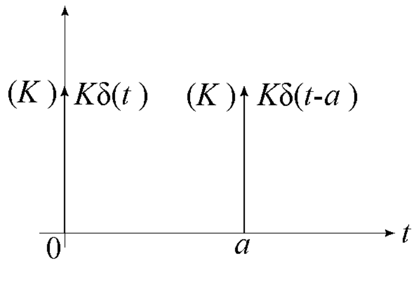


Figure 1.11: Graphical representation of the impulse $K\delta(t)$ and $K\delta(t-a)$.

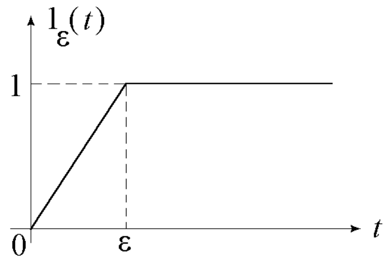


Figure 1.12: The function 1_ϵ approaches the unit step as $\varepsilon \to 0$.

To verify the validity of the sifting property, we note that $\delta(t-a)$ is zero everywhere except at $t = a$. Hence, we can write,

$$\int_{-\infty}^{\infty} f(t)\delta(t-a)dt = \int_{a-\varepsilon}^{a+\varepsilon} f(t)\delta(t-a)dt.$$

By assumption, f is continuous at a. Therefore, it must take the value of $f(a)$ as $t \to a$. Thus,

$$\int_{a-\varepsilon}^{a+\varepsilon} f(t)\delta(t-a)dt = f(a)\int_{a-\varepsilon}^{a+\varepsilon} \delta(t-a)dt = f(a),$$

which was to be demonstrated.

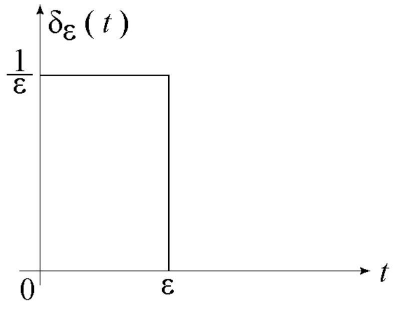


Figure 1.13: The derivative of $1_\varepsilon(t)$ shown in Figure 1.12. As $\varepsilon \to 0$, $\delta_\varepsilon(t)$ approches $\delta(t)$.

1.3.2 Zero-State Response Description of Continuous-Time, Relaxed, Causal Systems

Suppose that the system under consideration is a single-input single-output (SISO) linear system. Let $\delta_\Delta(t-t_1)$ be the pulse depicted in Figure 1.14. The pulse has width Δ and height $1/\Delta$.

We can use the pulse $\delta_\Delta(t-t_1)$ to approximate an input signal $u(t)$ as illustrated in Figure 1.15. That is, we can approximate the control signal $u(t)$ as

$$u(t) \approx \sum_i u(t_i)\delta_\Delta(t-t_i)\Delta. \tag{1.2}$$

Let $y(t) = g_\Delta(t, t_i)$ be the system output at time t produced by the pulse $u(t) = \delta_\Delta(t-t_i)$ applied at time t_i—see Figure 1.16. Then by the homogeneity property, the input $u(t) = \delta_\Delta(t-t_i)u(t_i)\Delta$ would produce the output, $y(t) = g_\Delta(t, t_i)u(t_i)\Delta$. Applying the additivity property, we conclude that the input given by (1.2) will produce the output

$$y(t) \approx \sum_i g_\Delta(t, t_i)u(t_i)\Delta. \tag{1.3}$$

Now if $\Delta \to 0$, then

$$\delta_\Delta(t-t_i) \to \delta(t-t_i)$$

and the corresponding output of the system is denoted $g(t, t_i)$. Next, as $\Delta \to 0$ the approximation in (1.3) becomes an equality, the summation an integration, the discrete time t_i becomes a continuum that we denote using the symbol τ, while Δ will be represented as $d\tau$. Taking the above into account, (1.3) takes the form,

$$y(t) = \int_{-\infty}^{\infty} g(t, \tau)u(\tau)d\tau. \tag{1.4}$$

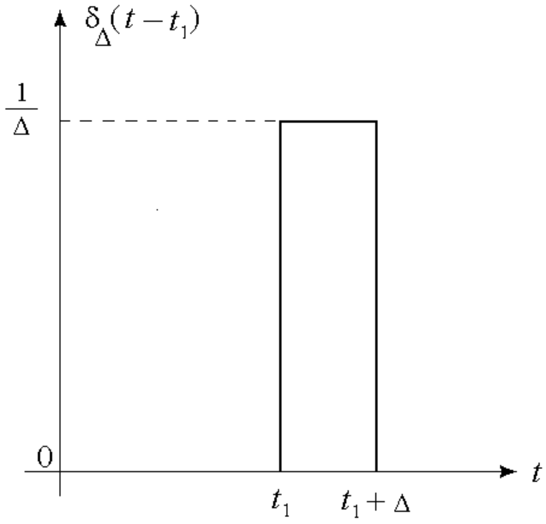


Figure 1.14: The pulse $\delta_\Delta(t - t_1)$ at $t = t_1$.

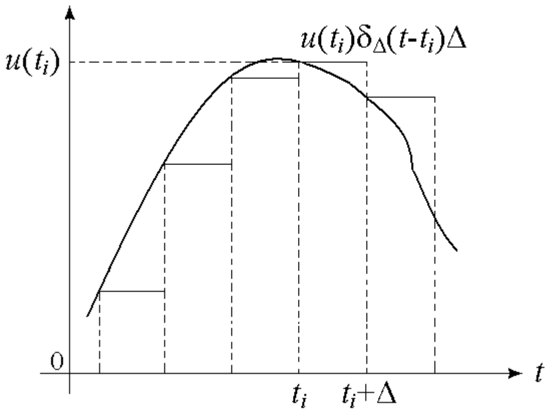


Figure 1.15: Approximating the control signal $u(t)$.

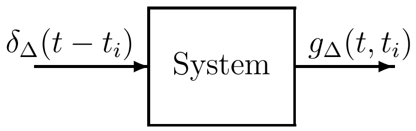


Figure 1.16: The system output $g_\Delta(t,t_i)$ is produced by the input $\delta_\Delta(t-t_i)$.

We note that $g(t,\tau)$ is a function of two variables. The first variable denotes the time t at which the output is observed, while the second variable, τ, denotes the time at which the impulse input is applied.

We call $g(t,\tau)$ the impulse response because it is the system's output when excited with the impulse input.

For the causal system, the output cannot appear before the input is applied. Therefore, for the causal systems,

$$g(t,\tau) = 0 \quad \text{for} \quad t < \tau$$

and we can replace the upper integration limit in (1.4) with t to obtain

$$y(t) = \int_{-\infty}^{t} g(t,\tau)u(\tau)d\tau. \tag{1.5}$$

For the relaxed system at t_0, its initial state, $\boldsymbol{x}(t_0) = \mathbf{0}$. Hence, the system output is produced exclusively by the input $u(t)$ for $t \geq t_0$. Thus we can replace the lower integration limit in (1.5) with t_0, and we obtain for the linear, relaxed at t_0, causal system the following expression for its response in terms of its impulse response;

$$y(t) = \int_{t_0}^{t} g(t,\tau)u(\tau)d\tau. \tag{1.6}$$

For the multi-input multi-output (MIMO) linear, relaxed at t_0, casual system with p output components and m inputs, the above expression generalizes to

$$\boldsymbol{y}(t) = \int_{t_0}^{t} \boldsymbol{G}(t,\tau)\boldsymbol{u}(\tau)d\tau, \tag{1.7}$$

where

$$\boldsymbol{G}(t,\tau) = \begin{bmatrix} g_{11}(t,\tau) & g_{12}(t,\tau) & \cdots & g_{1m}(t,\tau) \\ g_{21}(t,\tau) & g_{22}(t,\tau) & \cdots & g_{2m}(t,\tau) \\ \vdots & \vdots & \ddots & \vdots \\ g_{p1}(t,\tau) & g_{p2}(t,\tau) & \cdots & g_{pm}(t,\tau) \end{bmatrix}.$$

In the above, $g_{ij}(t,\tau)$ is the impulse response at time t at the i-th output due to an impulse applied at time τ at the j-th input, with other inputs set to zero.

We refer to the matrix $\boldsymbol{G}$ as the *impulse response matrix*.

1.3.3 Input-Output Description of Continuous Time-Invariant Systems

Suppose that we have a system with the initial state $\boldsymbol{x}(t_0)$, the input $\boldsymbol{u}(t)$ applied from t_0 that produced the output $y(t)$ for $t \geq t_0$. We compactly describe the above mental experiment as

$$L(\boldsymbol{u}(t), \boldsymbol{x}(t_0)) = \boldsymbol{y}(t), \quad \text{for} \quad t \geq t_0.$$

The above system is said to be time-invariant if for any T, if the initial state is shifted to time $t_0 + T$, the same input is applied from $t_0 + T$ rather than at t_0, then the output is the same except that now it starts to appear at $t_0 + T$, that is,

$$L(\boldsymbol{u}(t-T), \boldsymbol{x}(t_0+T)) = \boldsymbol{y}(t-T), \quad \text{for} \quad t \geq t_0 + T.$$

Therefore, the impulse response of a linear, relaxed at t_0, causal, time invariant system can be described as

$$g(t,\tau) = g(t+T, \tau+T) = g(t-\tau, 0), \tag{1.8}$$

which we compactly denote as $g(t-\tau)$. Note that the impulse response of a linear time-invariant (LTI) system, $g(t-\tau)$, is a function of one variable, where $g(t) = g(t-0)$ is the system output at time t to an impulse applied at $\tau = 0$. The LTI system is causal if $g(t) = 0$ for $t < 0$.

Taking into account (1.8), we represent the expression for the output (1.6) of a relaxed at t_0, causal, LTI system as

$$y(t) = \int_0^t g(t-\tau)u(\tau)d\tau, \tag{1.9}$$

where, without loss of generality, we replaced t_0 with 0. The integration in (1.9) is known as the *convolution* integral. The above integral relation is also written in a shorthand notation as

$$y(t) = g(t) * u(t),$$

where the asterisk signifies the integral relation between $g(t)$ and $u(t)$. The integral relation $g(t) * u(t)$ is read as "$g(t)$ is convolved with $u(t)$."

Note that the convolution is a commutative operation, that is,

$$y(t) = \int_0^t g(t-\tau)u(\tau)d\tau = \int_0^t g(\tau)u(t-\tau)d\tau, \tag{1.10}$$

which can be represented as

$$y(t) = g(t) * u(t) = u(t) * g(t).$$

A very useful tool in the analysis and design of LTI dynamic systems is the Laplace transform that we review in Section A.15. Applying the Laplace transform to (1.9) gives

$$\begin{aligned} Y(s) &= \mathcal{L}(y(t)) \\ &= \int_{0^-}^{\infty} \left(\int_{\tau=0}^{t} g(t-\tau)u(\tau)d\tau \right) e^{-st}dt. \end{aligned}$$

For causal systems, $g(t) = 0$ for $t < 0^-$; equivalently, $g(t-\tau) = 0$ for $\tau > t^-$. This means that we can replace the upper integration limit t in the above with ∞, and performing some simple manipulations, we obtain

$$\begin{aligned} Y(s) &= \int_{0^-}^{\infty} \left(\int_{\tau=0^-}^{\infty} g(t-\tau)u(\tau)d\tau \right) e^{-st}dt \\ &= \int_{0^-}^{\infty} \left(\int_{\tau=0^-}^{\infty} g(t-\tau)u(\tau)d\tau \right) e^{-s(t-\tau)}e^{-s\tau}dt \\ &= \int_{0^-}^{\infty} \left(\int_{\tau=0^-}^{\infty} g(t-\tau)e^{-s(t-\tau)}dt \right) u(\tau)e^{-s\tau}d\tau \\ &= \int_{0^-}^{\infty} g(t-\tau)e^{-s(t-\tau)}dt \int_{\tau=0^-}^{\infty} u(\tau)e^{-s\tau}d\tau \\ &= G(s)U(s), \end{aligned} \tag{1.11}$$

where $G(s)$ is called the transfer function of the system and $U(s)$ is the Laplace transform of the input signal.

The Laplace transform of the convolution of two functions of time is equal to the product of their Laplace transforms, that is,

$$\mathcal{L}(g(t) * u(t)) = \mathcal{L}(g(t))\mathcal{L}(u(t)) = G(s)U(s).$$

Thus, to the product in the s-domain there corresponds the convolution in the time domain.

If a linear time-invariant system is lumped, its transfer function, $G(s)$, is a rational function of s, that is,

$$G(s) = \frac{N(s)}{D(s)},$$

where $N(s)$ and $D(s)$ are polynomials in s.

For the MIMO LTI dynamical system, (1.11) takes the form,

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \\ \vdots \\ Y_p(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1m}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{p1}(s) & G_{p2}(s) & \cdots & G_{pm}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \\ \vdots \\ U_m(s) \end{bmatrix}.$$

We represent the above in a vector-matrix notation as

$$\boldsymbol{Y}(s) = \boldsymbol{G}(s)\boldsymbol{U}(s),$$

where $\boldsymbol{G}(s)$ is called the *transfer function matrix*.

Example 1.2 If we apply the impulse input in the unit-time delay system of Example 1.1, then the output is $\delta(t-1)$. Hence the impulse response of this system is

$$g(t) = \delta(t-1). \tag{1.12}$$

The transfer function of the unit-time delay system of Example 1.1 is

$$G(s) = e^{-s}, \tag{1.13}$$

which is not a rational function because this system is distributed.

Example 1.3 The integrator can be described as

$$y(t) = k_I \int_{-\infty}^{t} u(\tau)d\tau, \tag{1.14}$$

where k_I is the gain. We can equivalently describe the integrator as

$$\frac{dy(t)}{dt} = k_I u(t). \tag{1.15}$$

The impulse response of the integrator is the unit step function also known as the Heaviside step function.

The transfer function of the integrator described by (1.14) or by (1.15) is

$$G(s) = \frac{k_I}{s}.$$

Example 1.4 The ideal differentiator is described as

$$y(t) = k_D \frac{du(t)}{dt},$$

where k_D is the differentiator's gain. Its impulse response is

$$\begin{aligned} g(t) &= k_D \frac{d\delta(t)}{dt} \\ &= \begin{cases} 0 & \text{for} \quad t \neq 0 \\ +\infty & \text{for} \quad t = 0^- \\ -\infty & \text{for} \quad t = 0^+, \end{cases} \end{aligned} \tag{1.16}$$

The transfer function of the ideal differentiator is

$$G(s) = k_D s.$$

1.3.4 State-Space Description of Continuous Time-Invariant Systems

Every linear time-invariant (LTI) lumped causal system can be described by a set of equations of the form,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{1.17}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t), \tag{1.18}$$

where

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

is the state vector. The matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are constant real matrices. A block diagram representation of the state-space description given by (1.17) and (1.18) is depicted in Figure 1.17.

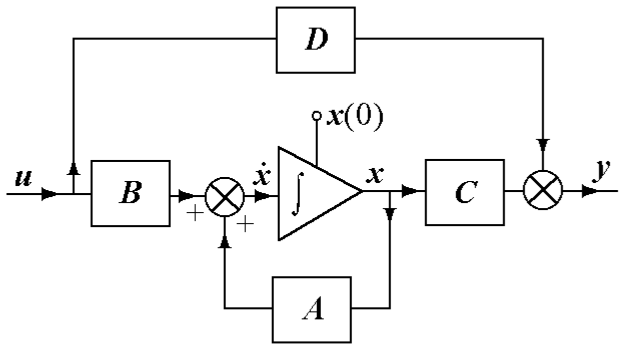


Figure 1.17: Block diagram representation of the state-space description given by (1.17) and (1.18).

Taking the Laplace transform of (1.17) and (1.18) gives

$$\begin{aligned} s\boldsymbol{X}(s) - \boldsymbol{x}(0^-) &= \boldsymbol{A}\boldsymbol{X}(s) + \boldsymbol{B}\boldsymbol{U}(s) \qquad (1.19) \\ \boldsymbol{Y}(s) &= \boldsymbol{C}\boldsymbol{X}(s) + \boldsymbol{D}\boldsymbol{U}(s). \qquad (1.20) \end{aligned}$$

We compute $\boldsymbol{X}(s)$ from (1.19),

$$\boldsymbol{X}(s) = (s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{x}(0^-) + (s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{U}(s)$$

Substituting the above into (1.20) yields

$$\boldsymbol{Y}(s) = \boldsymbol{C}\,(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{x}(0^-) + \boldsymbol{C}\,(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{U}(s) + \boldsymbol{D}\boldsymbol{U}(s).$$

If the initial condition is zero, then

$$\boldsymbol{Y}(s) = \left(\boldsymbol{C}\,(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}\right)\boldsymbol{U}(s),$$

where

$$\boldsymbol{G}(s) = \boldsymbol{C}\,(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D} \qquad (1.21)$$

is the transfer function matrix as a function of the parameters of a state-space description of a given causal, lumped LTI system.

1.3.5 Input-Output Description of Discrete Time-Invariant Systems

Recall that a discrete-time system accepts discrete-time signals and produces discrete-time signals as its output.

As in with a continuous-time system, a discrete-time system is linear if it possesses the superposition property.

The response of a discrete-time linear system is the sum of the zero-state and zero-input responses. Both the zero-state and zero-input responses possess the superposition property.

A discrete-time system is causal if its current output depends on the current and past inputs.

The discrete system state, $\boldsymbol{x}[k_0]$, is the information at time k_0 that together with $\boldsymbol{u}[k]$ for $k \geq k_0$ uniquely determines the system output $\boldsymbol{y}[k]$ for $k \geq k_0$.

If the number of the components in the state vector $\boldsymbol{x}$ is finite, then this discrete system is lumped; otherwise it is distributed. If there is a time delay in a continuous-time system, then this system is distributed. On the other hand, if in a discrete-time system the time delay is an integer multiple of the sampling period h, then such a system is a lumped system.

To proceed, we define the impulse sequence denoted $\delta[k-m]$ and defined as

$$\delta[k-m] = \begin{cases} 1 & \text{if} \quad k = m \\ 0 & \text{if} \quad k \neq m, \end{cases}$$

where both k and m are integers that denote sampling instances.

Suppose now that we have an input sequence denoted $u[k]$. We express this input sequence as

$$u[k] = \sum_{m=-\infty}^{\infty} u[k]\delta[k-m]. \tag{1.22}$$

Let $g[k,m]$ denote the discrete-time system output at time k when excited with the impulse sequence, $\delta[k-m]$, at the time instant m. Because by assumption the discrete system under consideration is linear, hence by the homogeneity property the input $\delta[k-m]u[m]$ will yield the output, $g[k,m]u[m]$. Combining the above with the additivity property, we conclude that the input, $\sum_{m=-\infty}^{\infty} \delta[k-m]u[m]$ will yield the output, $\sum_{m=-\infty}^{\infty} g[k,m]u[m]$. We illustrate the above in Figure 1.18.

In a causal discrete-time linear system, no output can appear before an input is applied. Hence,

$$g[k,m] = 0 \quad \text{for} \quad k < m.$$

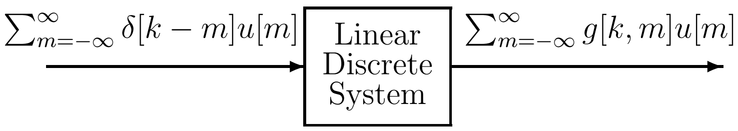


Figure 1.18: Output of linear discrete-time system in response to the input sequence (1.22).

If, in addition, the system is relaxed at k_0, then the output will take the form,

$$y[k] = \sum_{m=k_0}^{k} g[k,m]u[m].$$

Note that for time-invariant linear causal system we can always set $k_0 = 0$ to obtain

$$y[k] = \sum_{m=0}^{k} g[k-m]u[m]. \tag{1.23}$$

The above is referred to as a *discrete convolution*. It is easy to check that the discrete convolution possesses the commutativity property, that is,

$$y[k] = \sum_{m=0}^{k} g[k-m]u[m] = \sum_{m=0}^{k} g[m]u[k-m]. \tag{1.24}$$

Applying the z-transform to (1.23) gives

$$\begin{aligned} Y(z) &= \mathcal{Z}(y[k]) \\ &= \sum_{k=0}^{\infty} \left(\sum_{m=0}^{k} g[k-m]u[m] \right) z^{-k}. \end{aligned}$$

Taking into account the causality assumption, we can replace the upper integration limit k with ∞. Then performing simple manipulations, we obtain

$$\begin{aligned} Y(z) &= \sum_{k=0}^{\infty} \left(\sum_{m=0}^{\infty} g[k-m]u[m] \right) z^{-(k-m)} z^{-m} \\ &= \sum_{m=0}^{\infty} \left(\sum_{m=0}^{\infty} g[k-m]z^{-(k-m)} \right) u[m]z^{-m} \\ &= \left(\sum_{l=0}^{\infty} g[l]z^{-l} \right) \left(\sum_{m=0}^{\infty} u[m]z^{-m} \right) \\ &= G(z)U(z), \end{aligned}$$

where we interchanged the order of summations, introduced the new variable $l = k - m$ and took into account the fact that $g[l] = 0$ for $l < 0$.

The z-transform of the impulse response, denoted $G(z)$, is called the *discrete transfer function.* Note that transfer functions describe only the zero-state responses.

Example 1.5 [55, p. 33] Consider the unit-sampling-time delay system model of the form,

$$y[k] = u[k-1].$$

The impulse response of the above system is

$$g[k] = \delta[k-1].$$

The discrete transfer function is obtained by taking the z-transform of the impulse response. We obtain

$$\begin{aligned} G(z) &= \mathcal{Z}(\delta[k-1]) \\ &= \frac{1}{z}. \end{aligned}$$

The above transfer function is a rational function of z. Thus the above system is a lumped system. Note that in the continuous-time case, systems involving time delays are distributed system. As the above example shows, this is not necessarily the case in discrete-time systems.

1.3.6 State-Space Description of Discrete Time-Invariant Systems

Every discrete linear time-invariant (LTI) lumped causal system can be described by a set of equations of the form,

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k], \quad \boldsymbol{x}[0] = \boldsymbol{x}_0 \tag{1.25}$$

$$\boldsymbol{y}[k] = \boldsymbol{C}\boldsymbol{x}[k] + \boldsymbol{D}\boldsymbol{u}[k], \tag{1.26}$$

where

$$\boldsymbol{x}[k] = \begin{bmatrix} x_1[k] \\ \vdots \\ x_n[k] \end{bmatrix}$$

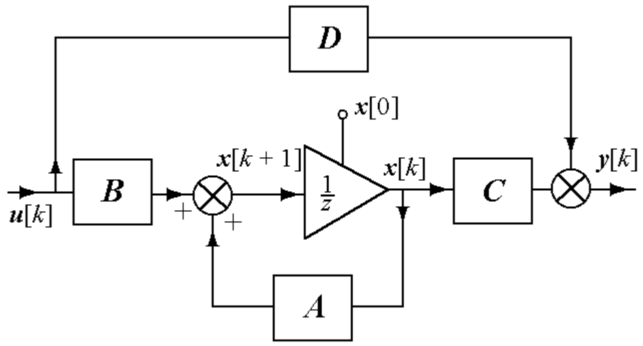


Figure 1.19: Block diagram representation of the state-space description given by (1.25) and (1.26).

is the state vector. The matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are constant real matrices. A block diagram representation of the state-space description given by (1.25) and (1.26) is depicted in Figure 1.19.

Let $\boldsymbol{X}(z)$ denote the z-transform of $\boldsymbol{x}[k]$, that is,

$$\boldsymbol{X}(z) = \mathcal{Z}(\boldsymbol{x}[k]).$$

Then, the z-transform of $\boldsymbol{x}[k+1]$ is

$$\begin{aligned}
\mathcal{Z}(\boldsymbol{x}[k+1]) &= \sum_{k=0}^{\infty} \boldsymbol{x}[k+1] z^{-k} \\
&= z \sum_{k=0}^{\infty} \boldsymbol{x}[k+1] z^{-(k+1)} \\
&= z \left(\sum_{j=1}^{\infty} \boldsymbol{x}[j] z^{-j} + \boldsymbol{x}[0] - \boldsymbol{x}[0] \right) \\
&= z \left(\boldsymbol{X}(z) - \boldsymbol{x}[0] \right). \qquad (1.27)
\end{aligned}$$

Applying the z-transform to (1.25) and (1.26) and taking into account (1.27) gives

$$z\boldsymbol{X}(z) - z\boldsymbol{x}[0] = \boldsymbol{A}\boldsymbol{X}(z) + \boldsymbol{B}\boldsymbol{U}(z), \quad \boldsymbol{x}[0] = \boldsymbol{x}_0 \qquad (1.28)$$

$$\boldsymbol{Y}(z) = \boldsymbol{C}\boldsymbol{X}(z) + \boldsymbol{D}\boldsymbol{U}(z). \qquad (1.29)$$

We calculate $\boldsymbol{X}(z)$ from (1.28) to obtain

$$\boldsymbol{X}(z) = (z\boldsymbol{I}_n - \boldsymbol{A})^{-1} z\boldsymbol{x}[0] + (z\boldsymbol{I}_n - \boldsymbol{A})^{-1} \boldsymbol{B}\boldsymbol{U}(z).$$

Substituting the above into (1.29) yields

$$\boldsymbol{Y}(z) = \boldsymbol{C}\,(z\boldsymbol{I}_n - \boldsymbol{A})^{-1} z\boldsymbol{x}[0] + \boldsymbol{C}\,(z\boldsymbol{I}_n - \boldsymbol{A})^{-1} \boldsymbol{B}\boldsymbol{U}(z) + \boldsymbol{D}\boldsymbol{U}(z).$$

If set the initial condition to zero, that is, $\boldsymbol{x}[0] = \boldsymbol{0}$, then we have

$$\boldsymbol{Y}(z) = \left(\boldsymbol{C}\,(z\boldsymbol{I}_n - \boldsymbol{A})^{-1} \boldsymbol{B} + \boldsymbol{D}\right) \boldsymbol{U}(z).$$

Thus the transfer function of the discrete time-invariant system described by (1.25) and (1.26) is

$$\boldsymbol{G}(z) = \boldsymbol{C}\,(z\boldsymbol{I}_n - \boldsymbol{A})^{-1} \boldsymbol{B} + \boldsymbol{D}. \tag{1.30}$$

Note that the above is the discrete equivalent of (1.21).

We add that to a model represented in a state-space format, there corresponds a unique transfer function matrix. On the hand, there are infinitely many state-space representations that have the same transfer function matrix.

A classification of linear systems discussed by us in this chapter is shown in Table 1.1.

Table 1.1: Table summarizing linear system classification discussed in Chapter 1; adapted from Chen [55, p. 37].

System	**Internal Description**	**External Description**
Distributed, linear		$\boldsymbol{y}(t) = \int_{t_0}^{t} \boldsymbol{G}(t,\tau)\boldsymbol{u}(\tau)d\tau$
Lumped, linear	$\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u}$ $\boldsymbol{y} = \boldsymbol{C}(t)\boldsymbol{x} + \boldsymbol{D}(t)\boldsymbol{u}$	$\boldsymbol{y}(t) = \int_{t_0}^{t} \boldsymbol{G}(t,\tau)\boldsymbol{u}(\tau)d\tau$
Distributed, linear, time-invariant		$\boldsymbol{y}(t) = \int_{t_0}^{t} \boldsymbol{G}(t-\tau)\boldsymbol{u}(\tau)d\tau$ $\boldsymbol{Y}(s) = \boldsymbol{G}(s)\boldsymbol{U}(s)$, $\boldsymbol{G}(s)$ irrational
Lumped, linear, time-invariant	$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$ $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}$	$\boldsymbol{y}(t) = \int_{t_0}^{t} \boldsymbol{G}(t-\tau)\boldsymbol{u}(\tau)d\tau$ $\boldsymbol{Y}(s) = \boldsymbol{G}(s)\boldsymbol{U}(s)$, $\boldsymbol{G}(s)$ rational

1.4 From Transfer Function to State-Space Description

If the Laplace variable s is replaced in (1.21) with the z-transform variable z then we obtain (1.30), and vice versa. Thus a method of converting a transfer function into a state-space representation for continuous linear time-invariant lumped systems is applicable also for discrete linear time-invariant lumped systems.

We say that a transfer function $\boldsymbol{G}(s)$ is realizable if there exists a quadruple of constant matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ such that $\boldsymbol{G}(s) = \boldsymbol{C}\left(s\boldsymbol{I}_n - \boldsymbol{A}\right)^{-1}\boldsymbol{B} + \boldsymbol{D}$. We call such a quadruple $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ a realization of $\boldsymbol{G}(s)$.

Definition 1.4 *The dimension of a realization is the size of the matrix $\boldsymbol{A}$, that is, if $\boldsymbol{A}$ is an n-by-n matrix then we say that the dimension of the corresponding realization is n.*

◇

We begin our discussion with a single-input single-output (SISO) system modeled by a transfer function,

$$\begin{aligned}\frac{Y(s)}{U(s)} &= G(s)\\ &= \frac{N(s)}{D(s)}\\ &= \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}\end{aligned} \tag{1.31}$$

that is a proper rational function, which means that

$$\deg_s N(s) \le \deg_s D(s).$$

Note that in (1.31) the highest coefficient of the denominator polynomial is unity. If this was not the case, we would divide the numerator and the denominator by the highest coefficient of the denominator thus forcing $a_n = 1$.

If $G(s)$ is proper but not strictly, that is, $\deg_s N(s) = \deg_s D(s)$, equivalently, $m = n$, then we divide the numerator $N(s)$ by the denominator $D(s)$ to obtain

$$G(s) = G(s)_{\mathrm{sp}} + G(\infty), \tag{1.32}$$

where $G(s)_{\mathrm{sp}}$ denotes the strictly proper part of $G(s)$ and $G(\infty) = b_m$.

Our goal is to find a realization of $G(s)$, that is, a quadruple, $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)$ such that

$$G(s) = \boldsymbol{c}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} + d. \tag{1.33}$$

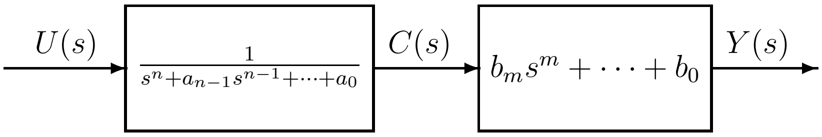


Figure 1.20: Decomposition of $G(s)$ to construct its state-space realization.

Note that $\boldsymbol{c}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{b}$ is a strictly proper rational function and d is a scalar. Therefore, in our construction of a realization of $G(s)$, we first extract the strictly proper part as in (1.32). We then find a triple $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ such that

$$G(s)_{\text{sp}} = \boldsymbol{c}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{b}. \tag{1.34}$$

A realization of $G(s)$ will have the form $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, G(\infty))$, that is, $d = G(\infty)$.

To proceed, we assume for simplicity that $G(s) = G(s)_{\text{sp}}$. We split our procedure of finding a triple $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ such that (1.34) holds into two steps. In the first step, we introduce an intermediate Laplace variable, $C(s)$, such that

$$\frac{Y(s)}{U(s)} = \frac{C(s)}{U(s)}\frac{Y(s)}{C(s)}.$$

Thus

$$\begin{aligned}
\frac{Y(s)}{U(s)} &= \frac{1}{D(s)}N(s) \\
&= \frac{1}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}\left(b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0\right) \\
&= \frac{C(s)}{U(s)}\frac{Y(s)}{C(s)}.
\end{aligned} \tag{1.35}$$

We illustrate the operation given by (1.35) in Figure 1.20. We first concern ourselves with the transfer function

$$\frac{C(s)}{U(s)} = \frac{1}{s^n + a_{n-1}s^{n-1} + \cdots + a_0}.$$

Performing cross-multiplication gives

$$\left(s^n + a_{n-1}s^{n-1} + \cdots + a_0\right)C(s) = U(s).$$

Taking the inverse Laplace transform and assuming zero initial conditions yields

$$c^{(n)} + a_{n-1}c^{(n-1)} + \cdots + a_1\dot{c} + a_0 c = u.$$

We define the state variables:

$$\left.\begin{array}{rcl} x_1 & = & c \\ x_2 & = & \dot{c} = \dot{x}_1 \\ & \vdots & \\ x_{n-1} & = & c^{(n-2)} = \dot{x}_{n-2} \\ x_n & = & c^{(n-1)} = \dot{x}_{n-1} \end{array}\right\} \tag{1.36}$$

Note that

$$\begin{aligned} \dot{x}_n &= c^{(n)} \\ &= -a_0 c - a_1 \dot{c} - \cdots - a_{n-1} c^{(n-1)} + u \\ &= -a_0 x_1 - a_1 x_2 - \cdots - a_{n-1} x_n + u. \end{aligned} \tag{1.37}$$

We represent (1.36) and (1.37) in the matrix-vector format as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u. \tag{1.38}$$

The above set of n first-order ordinary differential equations can be simulated using a circuit shown in Figure 1.21.

We next concern ourselves with the transfer function

$$\frac{Y(s)}{C(s)} = b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0,$$

which we represent as

$$Y(s) = b_m s^m C(s) + b_{m-1} s^{m-1} C(s) + \cdots + b_1 s C(s) + b_0 C(s).$$

Taking the inverse Laplace transform, while assuming zero initial conditions, we obtain

$$y = b_m c^{(m)} + b_{m-1} c^{(m-1)} + \cdots + b_1 \dot{c} + b_0 c \tag{1.39}$$

Note that

$$\begin{aligned} c &= x_1 \\ \dot{c} &= \dot{x}_1 = x_2 \\ &\vdots \\ c^{(m-1)} &= \dot{x}_{m-1} = x_m \\ c^{(m)} &= \dot{x}_m = x_{m+1}. \end{aligned}$$

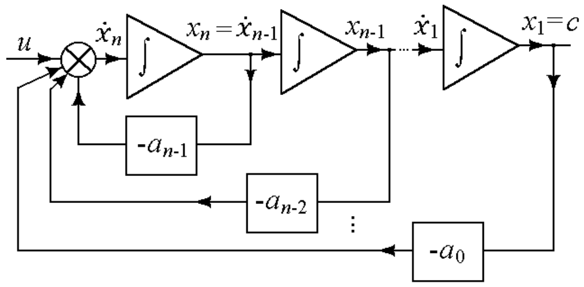


Figure 1.21: Implementation of (1.38).

Hence, we can represent y in terms of the state variables,

$$y = b_m x_{m+1} + b_{m-1} x_m + \cdots + b_1 x_2 + b_0 x_1,$$

or, equivalently, as

$$y = \begin{bmatrix} b_0 & b_1 & \cdots & b_{m-1} & b_m & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}. \tag{1.40}$$

Combining (1.38) and (1.40), we obtain a realization of a strictly proper transfer function (1.31). This state-space realization has the form,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u \tag{1.41}$$

$$y = \begin{bmatrix} b_0 & \cdots & b_m & \cdots & 0 \end{bmatrix} \boldsymbol{x}. \tag{1.42}$$

We depict the above realization in Figure 1.22.

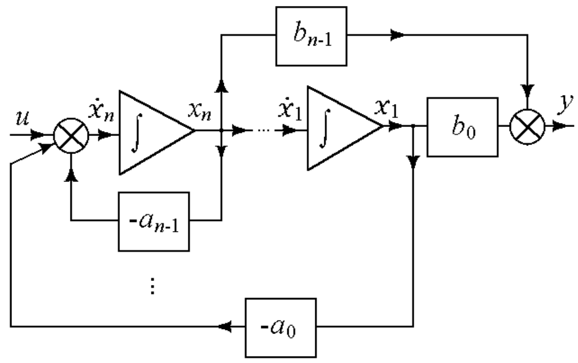


Figure 1.22: Implementation of (1.22).

Example 1.6 To find a realization of the transfer function

$$G(s) = \frac{4s^3 - 2s^2 + 3s + 1}{s^3 + 3s^2 - 5s + 7} \tag{1.43}$$

we represent it first as

$$\begin{aligned} G(s) &= G(s)_{\text{sp}} + G(\infty) \\ &= \frac{-14s^2 + 23s - 27}{s^3 + 3s^2 - 5s + 7} + 4. \end{aligned}$$

Then, using the results of the above discussion, we obtain the following realization of the transfer function (1.43),

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -7 & 5 & -3 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} -27 & 23 & -14 \end{bmatrix} \boldsymbol{x} + 4u. \end{aligned}$$

The above realization is only one of the infinitely many possible realizations of a given rational function $G(s)$. We can easily obtain another realization of $G(s)$. First note that $G(s)$ can be viewed as an 1×1 matrix; therefore, its transpose equals itself, that is,

$$G(s)^\top = G(s).$$

Applying the transposition operation to both sides of (1.33) and using the property of the transpose of the product of matrices, we obtain

$$\begin{aligned} G(s) &= G(s)^\top \\ &= \left(\boldsymbol{c}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{b}+d\right)^\top \\ &= \boldsymbol{b}^\top\left((s\boldsymbol{I}-\boldsymbol{A})^{-1}\right)^\top\boldsymbol{c}^\top+d^\top \\ &= \boldsymbol{b}^\top\left(s\boldsymbol{I}-\boldsymbol{A}^\top\right)^{-1}\boldsymbol{c}^\top+d \\ &= \tilde{\boldsymbol{c}}\left(s\boldsymbol{I}-\tilde{\boldsymbol{A}}\right)^{-1}\tilde{\boldsymbol{b}}+d. \end{aligned}$$

Thus, we have another realization of $G(s)$ of the form

$$\dot{\tilde{\boldsymbol{x}}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & -a_{n-2} \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix}\tilde{\boldsymbol{x}} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix} u \tag{1.44}$$

$$y = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}\tilde{\boldsymbol{x}} + G(\infty). \tag{1.45}$$

We generalize the above discussion for multi-input multi-output transfer functions in the form of the following theorem.

Theorem 1.1 *A transfer function $\boldsymbol{G}(s)$ is realizable if and only if $\boldsymbol{G}(s)$ is a proper rational matrix.*

Proof ($\Longleftarrow$) We first prove that for a given a rational proper transfer function $\boldsymbol{G}(s) \in \mathbb{R}^{p\times m}(s)$ there is a realization. The proof is constructive. We begin by decomposing $\boldsymbol{G}(s)$ as

$$\boldsymbol{G}(s) = \boldsymbol{G}(s)_{\text{sp}} + \boldsymbol{G}(\infty),$$

where $\boldsymbol{G}(s)_{\text{sp}}$ denotes the strictly proper part of $\boldsymbol{G}(s)$. Let

$$d(s) = s^r + \alpha_{r-1}s^{r-1} + \cdots + \alpha_1 s + \alpha_0$$

be the least common denominator of all rational function components of $\boldsymbol{G}(s)_{\text{sp}}$. Note that the highest coefficient of $d(s)$ is unity. We can always make $\alpha_r = 1$ by dividing, if necessary, the corresponding numerator polynomials and $d(s)$ by α_r. We then express $\boldsymbol{G}(s)_{\text{sp}}$ as

$$\begin{aligned} \boldsymbol{G}(s)_{\text{sp}} &= \frac{\boldsymbol{N}(s)}{d(s)} \\ &= \frac{\boldsymbol{N}_{r-1}s^{r-1} + \cdots + \boldsymbol{N}_1 s + \boldsymbol{N}_0}{d(s)}, \end{aligned}$$

where each $\boldsymbol{N}_i$ is an $p \times m$ constant matrix.

We next show that the following state-space representation yields a realization of $\boldsymbol{G}(s)$:

$$\left.\begin{aligned}
\dot{\boldsymbol{x}} &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_m & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{I}_m \\ -\alpha_0\boldsymbol{I}_m & -\alpha_1\boldsymbol{I}_m & -\alpha_2\boldsymbol{I}_m & \cdots & -\alpha_{r-2}\boldsymbol{I}_m & -\alpha_{r-1}\boldsymbol{I}_m \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{O} \\ \vdots \\ \boldsymbol{O} \\ \boldsymbol{I}_m \end{bmatrix} \boldsymbol{u} \\
\boldsymbol{y} &= \begin{bmatrix} \boldsymbol{N}_0 & \boldsymbol{N}_1 & \boldsymbol{N}_2 & \cdots & \boldsymbol{N}_{r-2} & \boldsymbol{N}_{r-1} \end{bmatrix} \boldsymbol{x} + \boldsymbol{G}(\infty)\boldsymbol{u},
\end{aligned}\right\} \tag{1.46}$$

where $\boldsymbol{D} = \boldsymbol{G}(\infty)$. Note that the matrix $\boldsymbol{A} \in \mathbb{R}^{rm\times rm}$ in the above realization is a block matrix; its blocks are $m \times m$ matrices.

To show that the above state-space representation is indeed a realization of $\boldsymbol{G}(s)$, we define

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{Z}_0 \\ \boldsymbol{Z}_1 \\ \vdots \\ \boldsymbol{Z}_{r-1} \end{bmatrix} = (s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}, \tag{1.47}$$

where $\boldsymbol{Z}_i$ is an $m \times m$ sub-matrix of the $rm \times m$ block matrix $\boldsymbol{Z}$. Then the transfer function of (1.46) is

$$\boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{G}(\infty) = \boldsymbol{N}_0\boldsymbol{Z}_0 + \boldsymbol{N}_1\boldsymbol{Z}_1 + \cdots + \boldsymbol{N}_{r-1}\boldsymbol{Z}_{r-1} + \boldsymbol{G}(\infty). \tag{1.48}$$

We pre-multiply both sides of (1.47) by $(s\boldsymbol{I} - \boldsymbol{A})$ and represent the result as

$$s\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{Z} + \boldsymbol{B}. \tag{1.49}$$

Applying the above operation to the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ given in (1.46), we obtain

$$s\boldsymbol{Z}_0 = \boldsymbol{Z}_1, \quad s\boldsymbol{Z}_1 = \boldsymbol{Z}_2, \quad s\boldsymbol{Z}_{r-2} = \boldsymbol{Z}_{r-1}, \tag{1.50}$$

and

$$s\boldsymbol{Z}_{r-1} = -\alpha_0\boldsymbol{Z}_0 - \alpha_1\boldsymbol{Z}_1 - \cdots - \alpha_{r-1}\boldsymbol{Z}_{r-1} + \boldsymbol{I}_m. \tag{1.51}$$

Note that $\boldsymbol{Z}_2 = s\boldsymbol{Z}_1 = s^2\boldsymbol{Z}_0$ and in general,

$$\boldsymbol{Z}_i = s^i\boldsymbol{Z}_0, \quad i = 1, \ldots, r-1 \tag{1.52}$$

Substituting (1.52) into (1.51) and performing simple manipulations gives

$$\left(s^r + \alpha_{r-1}s^{r-1} + \cdots + \alpha_1 s + \alpha_0\right)\boldsymbol{Z}_0 = \boldsymbol{I}_m,$$

that is,

$$\boldsymbol{Z}_0 = \frac{1}{d(s)}\boldsymbol{I}_m. \tag{1.53}$$

Using (1.50) or (1.52) and the above yields

$$\boldsymbol{Z}_1 = \frac{s}{d(s)}\boldsymbol{I}_m \quad \ldots \quad \boldsymbol{Z}_{r-1} = \frac{s^{r-1}}{d(s)}\boldsymbol{I}_m. \tag{1.54}$$

Substituting (1.53) and (1.54) into (1.48), we obtain

$$\begin{aligned} \boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{G}(\infty) &= \frac{1}{d(s)}\left(\boldsymbol{N}_0+\boldsymbol{N}_1 s+\cdots+\boldsymbol{N}_{r-1}s^{r-1}\right)+\boldsymbol{G}(\infty) \\ &= \boldsymbol{G}(s), \end{aligned} \tag{1.55}$$

which shows that (1.46) is a realization of $\boldsymbol{G}(s)$. The proof of this part of the theorem is complete.

($\Longrightarrow$) If the transfer function $\boldsymbol{G}(s)$ is realizable then there exists a quadruple $(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D})$ such that

$$\boldsymbol{G}(s) = \boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}.$$

Note that $\boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}$ is a strictly proper rational matrix and because $\boldsymbol{D}$ is a constant matrix, $\boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}$ is a proper rational matrix, which completes the proof of this part of the theorem.

□

Example 1.7 We find a state-space realization of the transfer function matrix,

$$\boldsymbol{G}(s) = \begin{bmatrix} \frac{1}{s(s+2)} & \frac{2s-1}{s(s+2)} & \frac{s-1}{s(s+2)} \\ -\frac{1}{s+2} & \frac{1-s}{s^2+3s+2} & \frac{1}{s^2+3s+2} \end{bmatrix}. \tag{1.56}$$

The above transfer function is strictly proper; therefore

$$\boldsymbol{D} = \boldsymbol{G}(\infty) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

We proceed to represent $\boldsymbol{G}(s)$ as

$$\begin{aligned} \boldsymbol{G}(s) &= \frac{1}{s^3+3s^2+2s}\begin{bmatrix} s+1 & 2s^2+s-1 & s^2-1 \\ -s^2-s & -s^2+s & s \end{bmatrix} \\ &= \frac{1}{s^3+3s^2+2s}\left(\begin{bmatrix} 0 & 2 & 1 \\ -1 & -1 & 0 \end{bmatrix}s^2+\begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}s+\begin{bmatrix} 1 & -1 & -1 \\ 0 & 0 & 0 \end{bmatrix}\right) \\ &= \frac{1}{s^3+3s^2+2s}\left(\boldsymbol{N}_2 s^2+\boldsymbol{N}_1 s+\boldsymbol{N}_0\right). \end{aligned}$$

Using (1.46), we obtain

$$
\begin{aligned}
\boldsymbol{A} &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_3 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_3 \\ \boldsymbol{O} & -2\boldsymbol{I}_3 & -3\boldsymbol{I}_3 \end{bmatrix} \\
&= \begin{bmatrix}
0 & 0 & 0 & \vdots & 1 & 0 & 0 & \vdots & 0 & 0 & 0 \\
0 & 0 & 0 & \vdots & 0 & 1 & 0 & \vdots & 0 & 0 & 0 \\
0 & 0 & 0 & \vdots & 0 & 0 & 1 & \vdots & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & \vdots & 1 & 0 & 0 \\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & \vdots & 0 & 1 & 0 \\
0 & 0 & 0 & \vdots & 0 & 0 & 0 & \vdots & 0 & 0 & 1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \vdots & -2 & 0 & 0 & \vdots & -3 & 0 & 0 \\
0 & 0 & 0 & \vdots & 0 & -2 & 0 & \vdots & 0 & -3 & 0 \\
0 & 0 & 0 & \vdots & 0 & 0 & -2 & \vdots & 0 & 0 & -3
\end{bmatrix}.
\end{aligned}
\tag{1.57}
$$

The input matrix $\boldsymbol{B}$ is

$$
\boldsymbol{B} = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{O} \\ \boldsymbol{I}_3 \end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\cdots & \cdots & \cdots \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\cdots & \cdots & \cdots \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, \tag{1.58}
$$

and the matrix $\boldsymbol{C}$ has the form

$$
\boldsymbol{C} = \begin{bmatrix} \boldsymbol{N}_0 & \boldsymbol{N}_1 & \boldsymbol{N}_2 \end{bmatrix} = \begin{bmatrix}
1 & -1 & -1 & \vdots & 1 & 1 & 0 & \vdots & 0 & 2 & 1 \\
0 & 0 & 0 & \vdots & -1 & 1 & 1 & \vdots & -1 & -1 & 0
\end{bmatrix}. \tag{1.59}
$$

The dimension of this realization is 9.

We can obtain another realization of $\boldsymbol{G}(s)$ from (1.46). We begin by constructing a realization for $\boldsymbol{G}(s)^\top$ using (1.46) to obtain

$$\boldsymbol{G}(s)^\top = \tilde{\boldsymbol{C}}(s\boldsymbol{I} - \tilde{\boldsymbol{A}})^{-1}\tilde{\boldsymbol{B}} + \tilde{\boldsymbol{D}}. \tag{1.60}$$

We next transpose both sides of (1.60) to obtain

$$\begin{aligned}
\left(\boldsymbol{G}(s)^\top\right)^\top &= \boldsymbol{G}(s) \\
&= \left(\tilde{\boldsymbol{C}}(s\boldsymbol{I} - \tilde{\boldsymbol{A}})^{-1}\tilde{\boldsymbol{B}} + \tilde{\boldsymbol{D}}\right)^\top \\
&= \tilde{\boldsymbol{B}}^\top \left(s\boldsymbol{I} - \tilde{\boldsymbol{A}}^\top\right)^{-1} \tilde{\boldsymbol{C}}^\top + \tilde{\boldsymbol{D}}^\top.
\end{aligned}$$

The resulting realization of $\boldsymbol{G}(s)$ has the form

$$\left.\begin{aligned}
\dot{\tilde{\boldsymbol{x}}} &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & -\alpha_0 \boldsymbol{I}_p \\ \boldsymbol{I}_p & \boldsymbol{O} & \boldsymbol{I}_m & \cdots & \boldsymbol{O} & -\alpha_1 \boldsymbol{I}_p \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & -\alpha_{r-2} \boldsymbol{I}_p \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{I}_p & -\alpha_{r-1} \boldsymbol{I}_p \end{bmatrix} \tilde{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{N}_0 \\ \boldsymbol{N}_1 \\ \vdots \\ \boldsymbol{N}_{r-2} \\ \boldsymbol{N}_{r-1} \end{bmatrix} \boldsymbol{u} \\
\boldsymbol{y} &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{I}_p \end{bmatrix} \tilde{\boldsymbol{x}} + \boldsymbol{G}(\infty)\boldsymbol{u}.
\end{aligned}\right\} \tag{1.61}$$

Note that although the realization (1.61) is based on the realization (1.46), their dimensions will be different if $p \neq m$.

Example 1.8 We use the above method to construct a state-space realization of the transfer function matrix given by (1.56). We apply (1.46) to $\boldsymbol{G}(s)^\top$, where

$$\begin{aligned}
\boldsymbol{G}(s)^\top &= \begin{bmatrix} \frac{1}{s(s+2)} & \frac{2s-1}{s(s+2)} & \frac{s-1}{s(s+2)} \\ -\frac{1}{s+2} & \frac{1-s}{s^2+3s+2} & \frac{1}{s^2+3s+2} \end{bmatrix}^\top \\
&= \frac{1}{s^3+3s^2+2s} \begin{bmatrix} s+1 & -s^2-s \\ 2s^2+s-1 & -s^2+s \\ s^2-1 & s \end{bmatrix} \\
&= \frac{1}{s^3+3s^2+2s} \left(\begin{bmatrix} 0 & -1 \\ 2 & -1 \\ 1 & 0 \end{bmatrix} s^2 + \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} s + \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ -1 & 0 \end{bmatrix} \right) \\
&= \frac{1}{s^3+3s^2+2s} \left(\boldsymbol{N}_2^\top s^2 + \boldsymbol{N}_1^\top s + \boldsymbol{N}_0^\top \right).
\end{aligned}$$

Applying to the above (1.61), we obtain another state-space realization of the transfer function $\boldsymbol{G}(s)$ given by (1.56), where

$$\hat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{I}_2 & \boldsymbol{O} & -2\boldsymbol{I}_2 \\ \boldsymbol{O} & \boldsymbol{I}_2 & -3\boldsymbol{I}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 0 & \vdots & 0 & 0 & \vdots & -2 & 0 \\ 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & -2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 1 & 0 & \vdots & -3 & 0 \\ 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & -3 \end{bmatrix}. \tag{1.62}$$

The input matrix is

$$\hat{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{N}_0 \\ \boldsymbol{N}_1 \\ \boldsymbol{N}_2 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 \\ 0 & 0 & 0 \\ \cdots & \cdots & \cdots \\ 1 & 1 & 0 \\ -1 & 1 & 1 \\ \cdots & \cdots & \cdots \\ 0 & 2 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \tag{1.63}$$

and the output matrix has the form

$$\hat{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 \end{bmatrix}. \tag{1.64}$$

The obtained realization is of lower dimension than the previous one. This does not mean that the alternative method always yields lower dimension realizations than the first method.

We present yet another method of constructing a state-space realization of a given transfer function matrix. This method is being used by the MATLAB's function `tf2ss`. We begin by denoting the i-th column of a proper transfer function matrix $\boldsymbol{G}(s)$ as $\boldsymbol{g}_i(s)$, and observe that the input-output relation of an MIMO system can be expressed as

$$\begin{aligned}\boldsymbol{Y}(s) &= \boldsymbol{G}(s)\boldsymbol{U}(s)\\ &= \boldsymbol{g}_1(s)U_1(s)+\boldsymbol{g}_2(s)U_2(s)+\cdots+\boldsymbol{g}_m(s)U_m(s),\end{aligned}$$

where $U_i(s)$ is the i-th component of the input vector $\boldsymbol{U}(s)$. We construct a state-space realization of each column $\boldsymbol{g}_i(s)$ and then combine these realizations to obtain the state-space realization of $\boldsymbol{G}(s)$ as follows. If the quadruple $(\boldsymbol{A}_i, \boldsymbol{b}_i, \boldsymbol{C}_i, \boldsymbol{d}_i)$ is the realization of $\boldsymbol{g}_i(s)$, then a realization $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ of $\boldsymbol{G}(s)$ has the form,

$$\left.\begin{aligned}\boldsymbol{A} &= \begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{O} & \cdots & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{A}_2 & \cdots & \boldsymbol{O}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{A}_m\end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{b}_2 & \cdots & \boldsymbol{0}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{b}_m\end{bmatrix},\\ \boldsymbol{C} &= \begin{bmatrix}\boldsymbol{C}_1 & \boldsymbol{C}_2 & \cdots & \boldsymbol{C}_m\end{bmatrix}, \quad \text{and} \quad \boldsymbol{D} = \begin{bmatrix}\boldsymbol{d}_1 & \boldsymbol{d}_2 & \cdots & \boldsymbol{d}_m\end{bmatrix}.\end{aligned}\right\} \tag{1.65}$$

We can also use the row-by-row approach to construct a state-space-realization of a given transfer function matrix—see Exercise 1.14.

Example 1.9 We use the column-by-column method to construct a realization of the transfer function matrix,

$$\boldsymbol{G}(s) = \begin{bmatrix}\frac{1}{s+1} & \frac{2}{s+1}\\ \frac{-1}{s^2+3s+2} & \frac{1}{s+2}\end{bmatrix} = \begin{bmatrix}\boldsymbol{g}_1(s) & \boldsymbol{g}_2(s)\end{bmatrix}. \tag{1.66}$$

This transfer function matrix is strictly proper, therefore

$$\boldsymbol{D} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}.$$

We apply (1.46) to

$$\begin{aligned}\boldsymbol{g}_1(s) &= \begin{bmatrix}\frac{1}{s+1}\\ \frac{-1}{s^2+3s+2}\end{bmatrix}\\ &= \frac{1}{s^2+3s+2}\begin{bmatrix}s+2\\ -1\end{bmatrix}\\ &= \frac{1}{s^2+3s+2}\left(\begin{bmatrix}1\\ 0\end{bmatrix}s+\begin{bmatrix}2\\ -1\end{bmatrix}\right)\\ &= \frac{1}{s^2+3s+2}\left(\boldsymbol{N}_1 s+\boldsymbol{N}_0\right).\end{aligned}$$

We obtain

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{C}_1 = \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}, \quad \boldsymbol{d}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

We next concern ourselves with the second column of $\boldsymbol{G}(s)$,

$$\begin{aligned} \boldsymbol{g}_2(s) &= \begin{bmatrix} \frac{2}{s+1} \\ \frac{1}{s+2} \end{bmatrix} \\ &= \frac{1}{s^2+3s+2} \begin{bmatrix} 2(s+2) \\ s+1 \end{bmatrix} \\ &= \frac{1}{s^2+3s+2} \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix} s + \begin{bmatrix} 4 \\ 1 \end{bmatrix} \right) \\ &= \frac{1}{s^2+3s+2} \left(\boldsymbol{N}_1 s + \boldsymbol{N}_0 \right). \end{aligned}$$

We apply (1.46) to obtain

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad \boldsymbol{b}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{C}_2 = \begin{bmatrix} 4 & 2 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{d}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Combining the above realizations gives an overall realization of $\boldsymbol{G}(s)$ of the form,

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & \vdots & 0 & 0 \\ -2 & -3 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 1 \\ 0 & 0 & \vdots & -2 & -3 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} 0 & \vdots & 0 \\ 1 & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 0 \\ 0 & \vdots & 1 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & \vdots & 4 & 2 \\ -1 & 0 & \vdots & 1 & 1 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{D} = \begin{bmatrix} \boldsymbol{d}_1 & \boldsymbol{d}_2 \end{bmatrix} = \begin{bmatrix} 0 & \vdots & 0 \\ 0 & \vdots & 0 \end{bmatrix}.$$

Example 1.10 We combine the method that generates the state-space realization given by (1.61) with the column-by-column method and apply it to the transfer function (1.66) from the previous example. We begin by transposing $\boldsymbol{G}(s)$ to obtain

$$\boldsymbol{G}(s)^\top = \begin{bmatrix} \frac{1}{s+1} & \frac{-1}{s^2+3s+2} \\ \frac{2}{s+1} & \frac{1}{s+2} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{g}}_1(s) & \tilde{\boldsymbol{g}}_2(s) \end{bmatrix}.$$

The column $\tilde{\boldsymbol{g}}_1(s)$ can be represented as

$$\tilde{\boldsymbol{g}}_1(s) = \frac{1}{s+1}\begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Hence, we have one-dimensional realization of $\tilde{\boldsymbol{g}}_1(s)$,

$$\tilde{\boldsymbol{A}}_1 = \begin{bmatrix} -1 \end{bmatrix}, \quad \tilde{\boldsymbol{b}}_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad \tilde{\boldsymbol{C}}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \text{nad} \quad \tilde{\boldsymbol{d}}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1.67}$$

We then find a realization of $\tilde{\boldsymbol{g}}_2(s)$, which we first represent as

$$\begin{aligned} \tilde{\boldsymbol{g}}_2(s) &= \begin{bmatrix} \frac{-1}{s^2+3s+2} \\ \frac{1}{s+2} \end{bmatrix} \\ &= \frac{1}{s^2+3s+2}\begin{bmatrix} -1 \\ s+1 \end{bmatrix} \\ &= \frac{1}{s^2+3s+2}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} s + \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) \\ &= \frac{1}{s^2+3s+2}\left(\tilde{\boldsymbol{N}}_1 s + \tilde{\boldsymbol{N}}_0\right). \end{aligned}$$

We obtain a second-order realization of $\tilde{\boldsymbol{g}}_2(s)$, where

$$\tilde{\boldsymbol{A}}_2 = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad \tilde{\boldsymbol{b}}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \tilde{\boldsymbol{C}}_2 = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \text{and} \quad \tilde{\boldsymbol{d}}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1.68}$$

Combining (1.67) and (1.68) and applying the transposition operation we obtain a third-order realization of (1.66),

$$\begin{aligned} \hat{\boldsymbol{A}} &= \begin{bmatrix} \tilde{\boldsymbol{A}}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{A}}_2 \end{bmatrix}^\top = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 1 & -3 \end{bmatrix}, \quad \hat{\boldsymbol{B}} = \begin{bmatrix} \tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{C}}_2 \end{bmatrix}^\top = \begin{bmatrix} 1 & 2 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}, \\ \hat{\boldsymbol{C}} &= \begin{bmatrix} \tilde{\boldsymbol{b}}_1 & 0 \\ \boldsymbol{0} & \tilde{\boldsymbol{b}}_2 \end{bmatrix}^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \hat{\boldsymbol{D}} = \begin{bmatrix} \tilde{\boldsymbol{D}}_1 & \tilde{\boldsymbol{D}}_2 \end{bmatrix}^\top = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

In the above examples, we obtained realizations of different dimensions for the same transfer function. Recall that the dimension of a realization is the size n of the matrix $\boldsymbol{A}$. Natural question arises of how to find a minimal realization for a given transfer function matrix. To answer this question we need notions of controllability and observability. It turns out that a realization is minimal if and only if it is both reachable and observable. We present methods for constructing minimal realizations in Section 5.3.

1.5 System Zeros

Recall that the zeros of a transfer function of a single-input single-output (SISO) system are defined as the zeros of the numerator polynomial of this transfer function. Our objective is to extend this definition to multi-input multi-output (MIMO) systems. To proceed, suppose that we have a SISO dynamical system modeled as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u \\ y &= \boldsymbol{c}\boldsymbol{x}. \end{aligned}$$

Then, we have

$$\begin{aligned} G(s) &= \frac{N(s)}{D(s)} \\ &= \boldsymbol{c}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} \\ &= \frac{\boldsymbol{c}\,\mathrm{adj}(s\boldsymbol{I}_n - \boldsymbol{A})\boldsymbol{b}}{\det(s\boldsymbol{I}_n - \boldsymbol{A})} \\ &= \frac{\det \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \boldsymbol{c} & 0 \end{bmatrix}}{\det(s\boldsymbol{I}_n - \boldsymbol{A})}. \end{aligned} \tag{1.69}$$

To verify the above, we note that

$$\begin{bmatrix} I_n & \mathbf{0} \\ -\boldsymbol{c}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1} & 1 \end{bmatrix} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \boldsymbol{c} & 0 \end{bmatrix} = \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \mathbf{0} & \boldsymbol{c}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} \end{bmatrix}.$$

Then observe that

$$\begin{aligned} \det\left(\begin{bmatrix} I_n & \mathbf{0} \\ -\boldsymbol{c}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1} & 1 \end{bmatrix} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \boldsymbol{c} & 0 \end{bmatrix} \right) &= 1 \times \det \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \boldsymbol{c} & 0 \end{bmatrix} \\ &= \det \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \mathbf{0} & \boldsymbol{c}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} \end{bmatrix}. \end{aligned}$$

Hence

$$\begin{aligned} \det \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \boldsymbol{c} & 0 \end{bmatrix} &= \det \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \mathbf{0} & \boldsymbol{c}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} \end{bmatrix} \\ &= \det(s\boldsymbol{I}_n - \boldsymbol{A}) \times \boldsymbol{c}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} \\ &= \det(s\boldsymbol{I}_n - \boldsymbol{A})\frac{\boldsymbol{c}\,\mathrm{adj}(s\boldsymbol{I}_n - \boldsymbol{A})\boldsymbol{b}}{\det(s\boldsymbol{I}_n - \boldsymbol{A})} \\ &= \boldsymbol{c}\,\mathrm{adj}(s\boldsymbol{I}_n - \boldsymbol{A})\boldsymbol{b}, \end{aligned}$$

which gives us (1.69).

We define the system zeros of a SISO dynamical system in terms of its state-space description as those roots of the polynomial equation

$$\det \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \boldsymbol{c} & 0 \end{bmatrix} = 0.$$

If

$$\deg D(s) = \deg\left(\det[s\boldsymbol{I}_n - \boldsymbol{A}]\right) = n,$$

then the transfer function zeros and the system zeros are the same. When, on the other hand,

$$\deg D(s) < \deg\left(\det[s\boldsymbol{I}_n - \boldsymbol{A}]\right) = n$$

then the transfer function zeros are a subset of the system zeros. We add that the transfer function zeros are also referred to as the transmission zeros.

We now analyze a MIMO dynamical system model

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \quad \boldsymbol{x}(0^-) = \boldsymbol{x}_0 \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}. \end{aligned}$$

Taking the Laplace transforms of the above equations gives

$$\begin{aligned} s\boldsymbol{X}(s) - \boldsymbol{x}(0^-) &= \boldsymbol{A}\boldsymbol{X}(s) + \boldsymbol{B}\boldsymbol{U}(s) \\ \boldsymbol{Y}(s) &= \boldsymbol{C}\boldsymbol{X}(s). \end{aligned}$$

We represent the above system of equations in the form

$$\begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}(s) \\ \boldsymbol{Y}(s) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}(0^-) \\ \boldsymbol{Y}(s) \end{bmatrix}.$$

We refer to the matrix

$$\boldsymbol{P}(s) = \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix}$$

as the Rosenbrock's system matrix. Note that $\boldsymbol{P}(s)$ is a polynomial matrix.

We wish to generalize the definition of system zeros to the MIMO case. For this we need the following definition.

Definition 1.5 *The* ***normal rank****, denoted normal-rank, of a polynomial matrix is the maximally possible rank of this matrix for at least one s. When we say that a polynomial matrix is of rank r, we refer to its normal rank.*

◇

Example 1.11 The rank of the polynomial matrix

$$\boldsymbol{M}_1(s) = \begin{bmatrix} s+1 & s^2-1 \\ s & s(s-1) \end{bmatrix}$$

is 1 because $\det \boldsymbol{M}_1(s) = 0$ for all s.

On the other hand the rank of the matrix

$$\boldsymbol{M}_2(s) = \begin{bmatrix} s+1 & s+1 \\ s & 1 \end{bmatrix}$$

is 2 because for any $s \neq \pm 1$ the rank of $\boldsymbol{M}_2$ is 2.

We now formally define system zeros of an MIMO LTI system.

Definition 1.6 *A complex number z_0 is a system zero, or an invariant zero, of the system $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ if*

$$\textit{rank} \begin{bmatrix} z_0\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} < \textit{normal-rank} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}.$$

◇

Example 1.12 The system matrix of the system defined by the quadruple

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -3 & -2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \text{and} \quad d = 0,$$

is

$$\boldsymbol{P}(s) = \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{b} \\ \boldsymbol{c} & d \end{bmatrix} = \begin{bmatrix} s & -1 & 0 \\ 3 & s+2 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

Note that $\det \boldsymbol{P}(s) = 1$, which means that the above system has no system zeros. (Strictly speaking this system has no finite system zeros.)

1.6 Summary and Notes

A landmark paper on mathematical description of linear dynamical systems from control point of view is by Kalman [169]. The first chapter of Sontag's book [310] is a comprehensive introduction to the subject of mathematical control theory.

Mayr writes on page 109 in [235], "It is still widely believed that the steam-engine governor is the oldest feedback device, and that James Watt had not only invented but also patented it. While both errors are easily refuted, we are still not able to reconstruct the history of this invention in all desired completeness." Watt did not patent the governor. He did not invent it either. On page 112 of his book [235], Mayr adds the following: "But the application of the centrifugal pendulum in a system of speed regulation of steam engines was a new breakthrough for which the firm of Boulton & Watt, if not James Watt himself, clearly deserves the credit."

1.7 Exercises

Exercise 1.1 Draw a block diagram illustrating the operation of the Watt's governor shown in Figure 1.6.

Exercise 1.2 An example of a closed-loop feedback system is a toilet flushing device connected to a water tank. A toilet system is shown in Figure 1.23. The control objective is to maintain the level of water in the tank at a constant level. Draw a block diagram of the system.

Exercise 1.3 Consider a system in which its input u and output y are related as,

$$\begin{aligned} y(t) &= L_{\alpha,\beta}(u(t)) \\ &= \begin{cases} u(t) & \text{for} \quad t \leq \alpha \\ \beta & \text{for} \quad t > \alpha, \end{cases} \end{aligned} \tag{1.70}$$

where α and β are design parameters. For what values of the design parameters the above system is linear?

Exercise 1.4 (based on Chen [55, p. 33])
Compute the transfer function of a discrete-time system whose impulse response is

$$g[k] = \begin{cases} 0 & \text{for} \quad k \leq 0 \\ \frac{1}{k} & \text{for} \quad k \geq 1. \end{cases}$$

Is this system lumped or distributed?

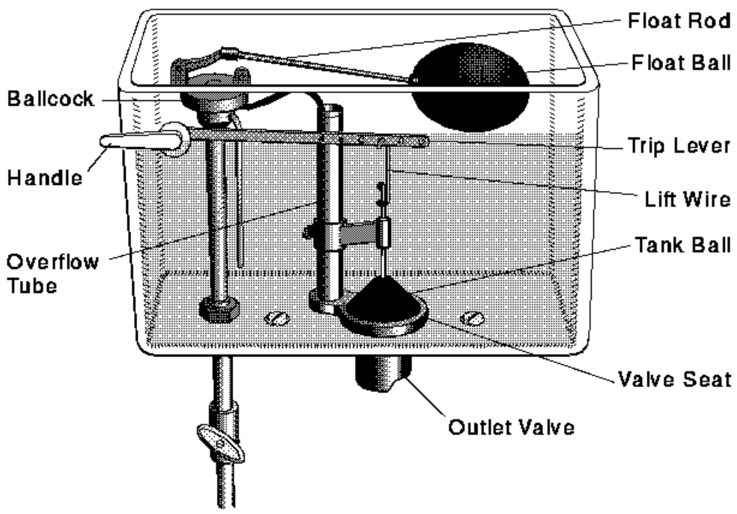


Figure 1.23: Toilet-flushing system for Exercise 1.2.

Exercise 1.5 Compute the impulse response of a dynamical system whose transfer function is given by

$$\boldsymbol{G}(s) = \begin{bmatrix} \frac{s+1}{(s+2)^2} & \frac{1}{(2s+1)(s+2)} & \frac{3s-1}{s} \\ \frac{3}{s+2} & \frac{4s-10}{2s+1} & \frac{2s+1}{s^2+1} \end{bmatrix}.$$

Exercise 1.6 Can you construct a state-space model of a linear system whose output is zero in response to any input?

Exercise 1.7 For the circuit shown in Figure 1.24 draw a block diagram with the current $i(t)$ as the system's input and the voltage $v(t)$ as the system's output. Then, find the transfer function, $V(s)/I(s)$.

Exercise 1.8 In the circuit shown in Figure 1.25, the input is the source current and the output is the voltage across the source terminals.

- Draw a block diagram of the circuit;
- Find the circuit transfer function;
- Construct a state-space realization;

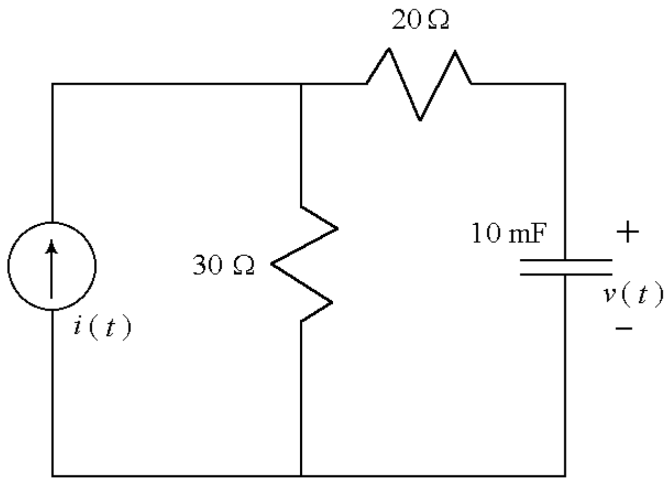


Figure 1.24: Circuit for Exercise 1.7.

- Find the impulse response and determine if the circuit is bounded-input bounded output (BIBO) stable. If not, then find a bounded input, I, that produces an unbounded output, V.

Exercise 1.9 In the circuit shown in Figure 1.26, the input is the source voltage, v_i and the output is the voltage across the capacitor, v_c.

- Draw a block diagram of the circuit;
- Find the circuit transfer function;
- Construct a state-space realization;
- Find the impulse response and determine if the circuit is BIBO stable. If not, then find a bounded input that produces an unbounded output.

Exercise 1.10 Find the transfer function $V_{out}(s)/V_{in}(s)$ of the circuit shown in Figure 1.27 that contains an ideal operational amplifier.

Exercise 1.11 The purpose of this Exercise is to show some benefits that the state-space description can offer in the analysis of control systems. Begin by verifying that the transfer

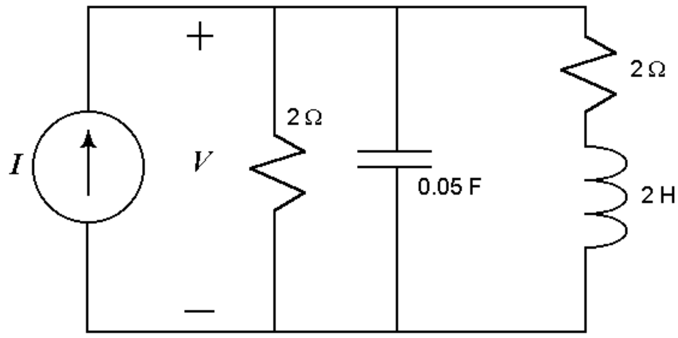


Figure 1.25: Circuit for Exercise 1.8.

function $Y(s)/R(s)$ for both systems shown in Figure 1.28 and 1.29 is the same. Obtain the step response for this transfer function.

Next obtain state-space realizations of the systems shown in Figure 1.28 and 1.29. That is, construct state-space realizations of $G_c(s) = \frac{s-1}{s+2}$ and $G_p(s) = \frac{3}{s-1}$ and then construct the overall state-space model of the cascade systems. The state-space models should of of second-order. Verify that the transfer functions of the models are the same as in the part above. Implement the obtained state-space models in SIMULINK. Obtain plots of the states and the outputs in response to a unit step input. Perform your simulations for different initial conditions that you can impose on the integrators. Compare the external, that is, input-output system behavior against the internal behavior. Can you give an explanation for an unexpected behavior of the state variables even when the external behavior seems to be acceptable.

Exercise 1.12 Suppose that $\boldsymbol{G}(s) \in \mathbb{R}^{p\times m}$ is a proper rational matrix such that

$$\boldsymbol{G}(s) = \boldsymbol{G}(s)_{\mathrm{sp}} + \boldsymbol{G}(\infty),$$

where

$$\boldsymbol{G}(s)_{\mathrm{sp}} = \frac{\boldsymbol{N}_{r-1}s^{r-1} + \cdots + \boldsymbol{N}_1 s + \boldsymbol{N}_0}{d(s)}$$

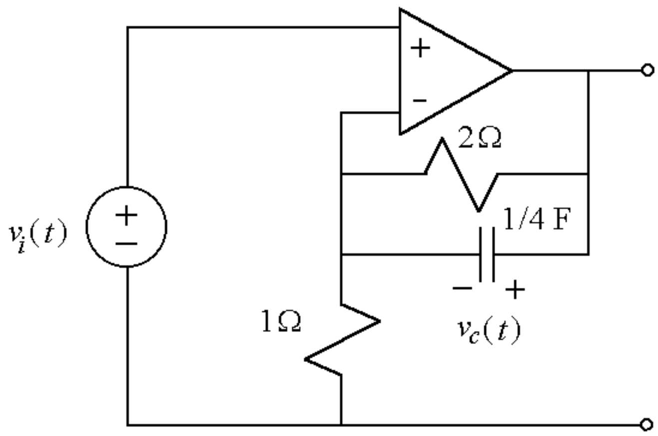


Figure 1.26: Circuit for Exercise 1.9.

and $d(s) = s^r + \alpha_{r-1}s^{r-1} + \cdots + \alpha_1 s + \alpha_0$. Show that the following state-space representation is a realization of $\boldsymbol{G}(s)$:

$$\left.\begin{aligned} \dot{\hat{\boldsymbol{x}}} &= \begin{bmatrix} -\alpha_{r-1}\boldsymbol{I}_m & -\alpha_{r-2}\boldsymbol{I}_m & -\alpha_{r-3}\boldsymbol{I}_m & \cdots & -\alpha_1\boldsymbol{I}_m & -\alpha_0\boldsymbol{I}_m \\ \boldsymbol{I}_m & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{I}_m & \boldsymbol{O} \end{bmatrix} \hat{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{O} \\ \vdots \\ \boldsymbol{O} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{u} \\ \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{N}_{r-1} & \boldsymbol{N}_{r-2} & \boldsymbol{N}_{r-3} & \cdots & \boldsymbol{N}_1 & \boldsymbol{N}_0 \end{bmatrix} \hat{\boldsymbol{x}} + \boldsymbol{G}(\infty)\boldsymbol{u}, \end{aligned}\right\} \tag{1.71}$$

Exercise 1.13 For a proper rational matrix

$$\boldsymbol{G}(s) = \begin{bmatrix} \frac{s+1}{(s+2)^2} & \frac{1}{(2s+1)(s+2)} & \frac{3s-1}{s} \\ \frac{3}{s+2} & \frac{4s-10}{2s+1} & \frac{2s+1}{s^2+1} \end{bmatrix}, \tag{1.72}$$

(i) construct its two realizations;

(ii) use MATLAB's `tf2ss` command to find a realization of $\boldsymbol{G}(s)$.

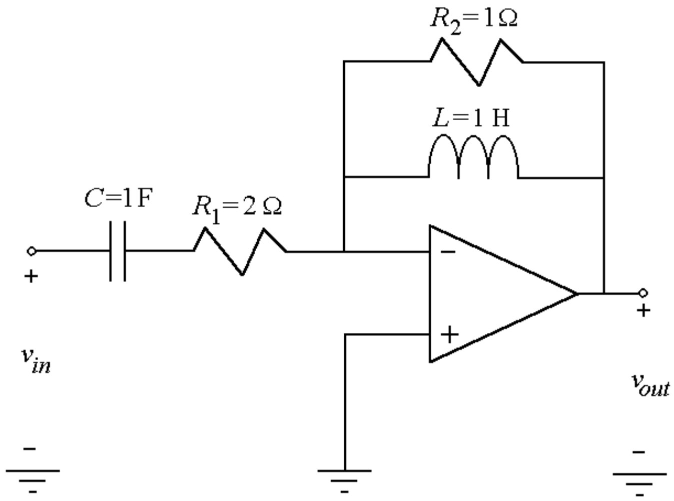


Figure 1.27: Circuit for Exercise 1.10.

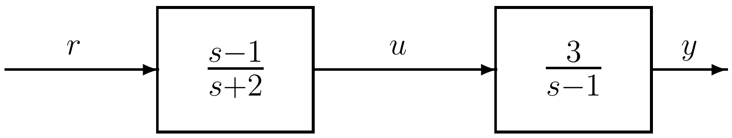


Figure 1.28: Cascade compensation for Exercise 1.11: compensator preceding the plant.

Exercise 1.14 Let $\boldsymbol{g}_i(s)^\top$ denote an i-th row of a given proper transfer function matrix $\boldsymbol{G}(s)$. Thus, we have

$$Y_i(s) = \boldsymbol{g}_i(s)^\top \boldsymbol{U}(s), \quad i = 1, 2 \ldots, p$$

where $Y_i(s)$ is the i-th component of the output vector $\boldsymbol{Y}(s)$. Suppose that a realization of $\boldsymbol{g}_i(s)^\top$ has the form $(\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{c}_i, \boldsymbol{d}_i)$, where $i = 1, 2, \ldots, p$. What are the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ of a realization of $\boldsymbol{G}(s)$ in terms of $(\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{c}_i, \boldsymbol{d}_i)$? Test this realization algorithm on the transfer function given by (1.56).

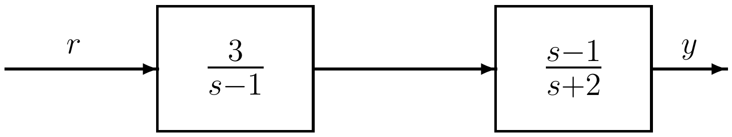


Figure 1.29: Compensator following the plant.

Chapter 2

Dynamical System Modeling

> To catch a mouse alive, try this trick: When you see the beast running around a room, open a black umbrella and hold it on the floor at an angle that allows the mouse to run in (it helps if someone else can chase the mouse). Your prey, drawn to dark places, will probably head straight for the umbrella. Then quickly close it, go outside, and set your prisoner free. [281, page 135]

In this chapter, we discuss the Lagrange equations of motion that are useful in mathematical modeling of dynamic systems. We first review relevant concepts that are needed to derive the Lagrange equations. We then construct mathematical models of selected dynamical systems.

2.1 Mathematical Modeling

Mathematical models of systems that we wish to control are key elements in their analysis and controller design. We devote this section 2.1 to mathematical modeling. Constructing models using linguistic description of the process to be controlled will be discussed in Subsection 15.4.1.

The goal of *mathematical modeling* is to provide a mathematical description of the interrelations between the system quantities as well as the relations between these quantities and external inputs. A mathematical model can be viewed as "a mathematical representation of the significant, relevant aspects of an existing or proposed physical system. (Significance and relevance being in relation to an application where the model is to be used)." [208, page xi]. Another definition of mathematical model that we found in [97, page 1] is: "A mathematical model is a system of equations that purports to represent some phenomenon in a way that gives insight into the origins and consequences of its behavior."

At this point, we wish to point out that it is important to distinguish between a physical system and its model. A physical system may be modeled using different models. In particular, it may be advantageous to model the physical system using different models for its different operating regions. For example, it is common to model an electric circuit using different models for different ranges of frequencies in which the circuit operates. In what follows, we refer to the physical system's model as the system.

In general, the more accurate the mathematical model is, the more complex are the modeling equations. This means that during the mathematical modeling process we may be forced to find a middle ground between accuracy of the model and its complexity. Our goal is to generate the most accurate possible description of the given system while still being able to carry out the model analysis and computations in a reasonable amount of time. At first, we tend to make oversimplifying assumptions in order to obtain a very simple model. However, we should follow the principle of maximum simplicity, which states that "everything should be as simple as possible—but no simpler." After extracting some information about the system behavior based on this first, possibly crude model, it may be desirable to construct a more accurate model. We try to create a mathematical description of a given physical process that incorporates realistic assumptions and physical constraints. We then extract information from this model and compare it with the experimental data. We perform this comparative study to determine the value of the obtained model. If the model appears to be inadequate for our needs, then it must be adjusted. The process of mathematical modeling is summarized in the diagram in Figure 2.1. In the process of mathematical modeling, one can use different types of equations and their combinations like algebraic, ordinary and partial differential equations, difference, integral, or functional equations. In this book, only difference and ordinary differential equations are used to model dynamical systems.

For the controller design purposes, we first construct a truth model of the given plant and then its design model. The truth model is the simulation model that should include all the relevant characteristics of the physical system to be controlled. The control designs are being simulated using the truth model. A design model is developed by eliminating "all unpleasant nonlinear effects while capturing all the essential features of the process, is much more amenable to use for the design of a control system" [108, p. 26].

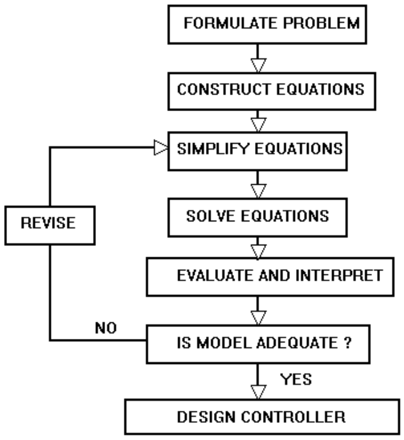


Figure 2.1: Diagram of mathematical modeling process and its use in the controller design process.

2.2 Review of Work and Energy Concepts

The purpose of this section is to review and reacquaint ourselves with some basic notions from physics like work, energy, etc. After a brief review of work and energy concepts, we derive the Lagrange equations of motion and use them to devise mathematical models of dynamic systems.

Suppose we are given a particle of a constant mass m subjected to a force $\boldsymbol{F}$. Then, by Newton's second law we have

$$\boldsymbol{F} = m\frac{d\boldsymbol{v}}{dt} = m\boldsymbol{a}. \tag{2.1}$$

Here the force $\boldsymbol{F}$ and the velocity $\boldsymbol{v}$ are vectors. Therefore, they can be represented as

$$\boldsymbol{F} = \begin{bmatrix} F_{x_1} & F_{x_2} & F_{x_3} \end{bmatrix}^\top \quad \text{and} \quad \boldsymbol{v} = \begin{bmatrix} v_{x_1} & v_{x_2} & v_{x_3} \end{bmatrix}^\top = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \dot{x}_3 \end{bmatrix}^\top.$$

Using the above notation, we represent equation (2.1) as

$$\begin{bmatrix} F_{x_1} \\ F_{x_2} \\ F_{x_3} \end{bmatrix} = m \begin{bmatrix} \frac{dv_{x_1}}{dt} \\ \frac{dv_{x_2}}{dt} \\ \frac{dv_{x_3}}{dt} \end{bmatrix}. \tag{2.2}$$

Suppose now that a force $\boldsymbol{F}$ is acting on a particle located at a point A and the particle moves to a point B. The work δW done by $\boldsymbol{F}$ along infinitesimally small distance $\delta \boldsymbol{s}$ is

$$\begin{aligned}\delta W &= \boldsymbol{F}^\top \delta \boldsymbol{s} \\ &= \begin{bmatrix} F_{x_1} & F_{x_2} & F_{x_3} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \end{bmatrix} \\ &= F_{x_1}\delta x_1 + F_{x_2}\delta x_2 + F_{x_3}\delta x_3.\end{aligned}$$

where $\delta \boldsymbol{s} = \begin{bmatrix} \delta x_1 & \delta x_2 & \delta x_3 \end{bmatrix}^\top$. The work W_{AB} done on the path from A to B is obtained by integrating the above equation,

$$W_{AB} = \int_A^B \left(F_{x_1} dx_1 + F_{x_2} dx_2 + F_{x_3} dx_3\right).$$

We now would like to establish a relation between work and kinetic energy. For this observe that

$$\ddot{x}\, dx = \frac{d\dot{x}}{dt} dx = \dot{x} d\dot{x}.$$

Using the above relation and Newton's second law, we express W_{AB} in a different way as

$$\begin{aligned}W_{AB} &= \int_A^B \left(m\ddot{x}_1 dx_1 + m\ddot{x}_2 dx_2 + m\ddot{x}_3 dx_3\right) \\ &= \int_A^B m\left(\dot{x}_1 d\dot{x}_1 + \dot{x}_2 d\dot{x}_2 + \dot{x}_3 d\dot{x}_3\right) \\ &= m\left(\frac{\dot{x}_1^2}{2} + \frac{\dot{x}_2^2}{2} + \frac{\dot{x}_3^2}{2}\right)\Bigg|_A^B \\ &= m\left(\frac{\|\boldsymbol{v}_B\|^2}{2} - \frac{\|\boldsymbol{v}_A\|^2}{2}\right) \\ &= \frac{m\|\boldsymbol{v}_B\|^2}{2} - \frac{m\|\boldsymbol{v}_A\|^2}{2}. \end{aligned} \tag{2.3}$$

where $\boldsymbol{v}_A = \begin{bmatrix} \dot{x}_{1A} & \dot{x}_{2A} & \dot{x}_{3A} \end{bmatrix}^\top$, and $\boldsymbol{v}_B$ is defined in an analogous manner, and $\|\boldsymbol{v}\|^2 = \boldsymbol{v}^\top \boldsymbol{v}$. Observing that $\frac{m\|\boldsymbol{v}_B\|^2}{2}$ is the *kinetic energy* of the particle at the point B and $\frac{m\|\boldsymbol{v}_A\|^2}{2}$ is its kinetic energy at the point A, we obtain

$$W_{AB} = K_B - K_A.$$

The above relation can be used to define the kinetic energy of a particle as the work required to change its velocity from some value $\boldsymbol{v}_A$ to a final value $\boldsymbol{v}_B$. The relation, $W_{AB} = K_B -$

$K_A = \Delta K$ is also known as the *work-energy theorem for a particle*. A force is conservative if the work done by the force on a particle that moves through any round trip is zero. In other words, a force is conservative if the work done by it on a particle that moves between two points depends only on these points and not on the path followed.

We next review the notion of *potential energy* or what is the same, the *energy of configuration*. Recall that if the kinetic energy K of a particle changes by ΔK, then the potential energy U must change by an equal but opposite amount so that the sum of the two changes is zero, that is,

$$\Delta K + \Delta U = 0.$$

This is equivalent to saying that any change in the kinetic energy of the particle is compensated for by an equal but opposite change in the potential energy U of the particle so that their sum remains constant, that is,

$$K + U = \text{constant}.$$

See Exercise 2.1 for a rigorous proof of the above relation. The potential energy of a particle represents a form of stored energy that can be recovered and converted into the kinetic energy. If we now use the work-energy theorem and the equation $\Delta K + \Delta U = 0$, then we obtain

$$W = \Delta K = -\Delta U.$$

The work done by a conservative force depends only on the starting and the end points of motion and not on the path followed between them. Therefore, for motion in one dimension,

$$\Delta U = -W = -\int_{x_0}^{x} F(s)ds. \tag{2.4}$$

We thus can write

$$F(x) = -\frac{dU(x)}{dx}. \tag{2.5}$$

A generalization of equation (2.4) to motion in three dimensions yields

$$\Delta U = -\int_{x_{10}}^{x_1} F_{x_1} ds - \int_{x_{20}}^{x_2} F_{x_2} ds - \int_{x_{30}}^{x_3} F_{x_3} ds,$$

and a generalization of equation (2.5) to motion in three dimensions is

$$\boldsymbol{F}(\boldsymbol{x}) = -\begin{bmatrix} \frac{\partial U}{\partial x_1} & \frac{\partial U}{\partial x_2} & \frac{\partial U}{\partial x_3} \end{bmatrix}^\top = -\nabla U(\boldsymbol{x}), \tag{2.6}$$

where $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top$.

For a formal proof of the property of a conservative vector field that the work done by it on a particle that moves between two points depends only on these points and not on the path followed, see Exercise 2.2.

We next represent Newton's equation in an equivalent format that establishes a connection with the Lagrange equations of motion to be discussed in the following section. To proceed, note that

$$K = \frac{m\left\|\dot{\boldsymbol{x}}\right\|^2}{2} = \frac{m\dot{\boldsymbol{x}}^\top\dot{\boldsymbol{x}}}{2}. \tag{2.7}$$

Hence,

$$\frac{\partial K}{\partial \dot{x}_i} = m\dot{x}_i, \quad i = 1, 2, 3. \tag{2.8}$$

Combining (2.6) and (2.8), we represent Newton's equations as

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{x}_i}\right) + (\nabla U)_i = 0, \quad i = 1, 2, 3. \tag{2.9}$$

Let

$$L = K - U. \tag{2.10}$$

The function L defined above is called the *Lagrangian function* or just the Lagrangian. Note that

$$\frac{\partial L}{\partial \dot{x}_i} = \frac{\partial K}{\partial \dot{x}_i}, \tag{2.11}$$

and

$$\frac{\partial L}{\partial x_i} = -\frac{\partial U}{\partial x_i}, \tag{2.12}$$

Substituting (2.11) and (2.12) into (2.9), we obtain

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_i}\right) - \frac{\partial L}{\partial x_i} = 0, \quad i = 1, 2, 3. \tag{2.13}$$

Equations (2.13) are called the Lagrange equations of motion, in Cartesian coordinates, for a single particle. They are just an equivalent representation of Newton's equations given by (2.2).

2.3 The Lagrange Equations of Motion in Generalized Coordinates

A single particle with no constraints is free to take up any position in the three dimensional space. This is the reason we need three coordinates to describe its position. In the case of a system of p particles there are $3p$ degrees of freedom. In general, if there are constraints imposed on the particles, then the number of the degrees of freedom is

$$n = 3p - \text{number of constraints}.$$

For example, if the system consists of only one particle in space and if the particle is restricted to move along a line, then $n = 3 - 2 = 1$. We can use many different coordinate systems to describe a location of a particle in space. We use the letter q as a symbol for a coordinate regardless of its nature. A generalized coordinate may represent, for example, angular displacement or electric charge. This indicates that the results we are going to obtain are of general nature and are not be restricted to mechanical systems. We refer to q as a generalized coordinate. Thus, if

$$\left.\begin{array}{rcl} x &=& x(q_1, q_2, q_3) \\ y &=& y(q_1, q_2, q_3) \\ z &=& z(q_1, q_2, q_3) \end{array}\right\}$$

then

$$\dot{x} = \frac{\partial x}{\partial q_1}\dot{q}_1 + \frac{\partial x}{\partial q_2}\dot{q}_2 + \frac{\partial x}{\partial q_3}\dot{q}_3,$$

and so on. We assume that the functions x, y, z are twice continuously differentiable, that is, $x, y, z \in \mathcal{C}^2$, which implies that their corresponding mixed second partial derivatives are equal.

Suppose now that a particle is confined to move on a surface. Therefore, to describe the location of the particle, we need only two generalized coordinates. The position of the particle on a surface in a three-dimensional space is then described as

$$\begin{array}{rcl} x &=& x(q_1, q_2), \\ y &=& y(q_1, q_2), \\ z &=& z(q_1, q_2), \end{array}$$

and hence

$$\left.\begin{array}{rcl} \delta x &=& \frac{\partial x}{\partial q_1}\delta q_1 + \frac{\partial x}{\partial q_2}\delta q_2 \\ \delta y &=& \frac{\partial y}{\partial q_1}\delta q_1 + \frac{\partial y}{\partial q_2}\delta q_2 \\ \delta z &=& \frac{\partial z}{\partial q_1}\delta q_1 + \frac{\partial z}{\partial q_2}\delta q_2. \end{array}\right\} \tag{2.14}$$

We know that the work δW done by the force $\boldsymbol{F}$ is $\delta W = F_x\delta x + F_y\delta y + F_z\delta z$. On the other hand, this work is equal to the change in the kinetic energy. Therefore,

$$\boxed{m\left(\ddot{x}\delta x + \ddot{y}\delta y + \ddot{z}\delta z\right) = F_x\delta x + F_y\delta y + F_z\delta z} \tag{2.15}$$

The above equation is known as *D'Alembert's* equation.

If we have a system of particles acted upon by forces $\boldsymbol{F}_1, \ldots, \boldsymbol{F}_p$ and if each of the particles undergoes displacements $\delta\boldsymbol{s}_1, \ldots, \delta\boldsymbol{s}_p$, then the total work done by the forces is

$$\delta W = \sum_{i=1}^{p} \left(F_{x_i}\delta x_i + F_{y_i}\delta y_i + F_{z_i}\delta z_i\right). \tag{2.16}$$

Substituting (2.14) into D'Alembert's equation and rearranging, we obtain

$$\begin{aligned}
\delta W &= m\left(\ddot{x}\frac{\partial x}{\partial q_1} + \ddot{y}\frac{\partial y}{\partial q_1} + \ddot{z}\frac{\partial z}{\partial q_1}\right)\delta q_1 + m\left(\ddot{x}\frac{\partial x}{\partial q_2} + \ddot{y}\frac{\partial y}{\partial q_2} + \ddot{z}\frac{\partial z}{\partial q_2}\right)\delta q_2 \\
&= \left(F_x\frac{\partial x}{\partial q_1} + F_y\frac{\partial y}{\partial q_1} + F_z\frac{\partial z}{\partial q_1}\right)\delta q_1 + \left(F_x\frac{\partial x}{\partial q_2} + F_y\frac{\partial y}{\partial q_2} + F_z\frac{\partial z}{\partial q_2}\right)\delta q_2.
\end{aligned} \tag{2.17}$$

Let

$$F_{q_1} = F_x\frac{\partial x}{\partial q_1} + F_y\frac{\partial y}{\partial q_1} + F_z\frac{\partial z}{\partial q_1},$$

and

$$F_{q_2} = F_x\frac{\partial x}{\partial q_2} + F_y\frac{\partial y}{\partial q_2} + F_z\frac{\partial z}{\partial q_2}.$$

Then, (2.17) can be represented as,

$$\delta W = F_{q_1}\delta q_1 + F_{q_2}\delta q_2.$$

To proceed further, note that

$$\frac{d}{dt}\left(\dot{x}\frac{\partial x}{\partial q_1}\right) = \ddot{x}\frac{\partial x}{\partial q_1} + \dot{x}\frac{d}{dt}\left(\frac{\partial x}{\partial q_1}\right).$$

Hence

$$\boxed{\ddot{x}\frac{\partial x}{\partial q_1} = \frac{d}{dt}\left(\dot{x}\frac{\partial x}{\partial q_1}\right) - \dot{x}\frac{d}{dt}\left(\frac{\partial x}{\partial q_1}\right)} \tag{2.18}$$

The time derivative of $x = x(q_1, q_2)$ is

$$\dot{x} = \frac{\partial x}{\partial q_1}\dot{q}_1 + \frac{\partial x}{\partial q_2}\dot{q}_2. \tag{2.19}$$

Differentiating the above partially with respect to $\dot{q}_1$ yields

$$\boxed{\frac{\partial \dot{x}}{\partial \dot{q}_1} = \frac{\partial x}{\partial q_1}} \tag{2.20}$$

Finally, we show that

$$\boxed{\frac{d}{dt}\left(\frac{\partial x}{\partial q_1}\right) = \frac{\partial \dot{x}}{\partial q_1}} \tag{2.21}$$

Indeed, because $x = x(q_1, q_2)$ and the partial derivative $\partial x/\partial q_1$ is in general a function of q_1 and q_2, we have

$$\frac{d}{dt}\left(\frac{\partial x}{\partial q_1}\right) = \frac{\partial}{\partial q_1}\left(\frac{\partial x}{\partial q_1}\right)\dot{q}_1 + \frac{\partial}{\partial q_2}\left(\frac{\partial x}{\partial q_1}\right)\dot{q}_2. \tag{2.22}$$

Taking the partial derivative of (2.19) with respect to q_1, we get

$$\begin{aligned}\frac{\partial \dot{x}}{\partial q_1} &= \frac{\partial}{\partial q_1}\left(\frac{\partial x}{\partial q_1}\right)\dot{q}_1 + \frac{\partial}{\partial q_1}\left(\frac{\partial x}{\partial q_2}\right)\dot{q}_2 \\ &= \frac{\partial}{\partial q_1}\left(\frac{\partial x}{\partial q_1}\right)\dot{q}_1 + \frac{\partial}{\partial q_2}\left(\frac{\partial x}{\partial q_1}\right)\dot{q}_2, \end{aligned} \tag{2.23}$$

because by assumption $x, y, z \in \mathcal{C}^2$ and hence

$$\frac{\partial}{\partial q_1}\left(\frac{\partial x}{\partial q_2}\right) = \frac{\partial}{\partial q_2}\left(\frac{\partial x}{\partial q_1}\right).$$

With the above in mind, comparing the right hand sides of (2.22) and (2.23) yields (2.21). We now use (2.18), (2.20), and (2.21) to obtain the Lagrange equation of motion for a single particle. We first substitute (2.20) and (2.21) into (2.18) to get

$$\ddot{x}\frac{\partial x}{\partial q_1} = \frac{d}{dt}\left(\dot{x}\frac{\partial \dot{x}}{\partial \dot{q}_1}\right) - \dot{x}\frac{\partial \dot{x}}{\partial q_1}. \tag{2.24}$$

Note that

$$\frac{\partial}{\partial \dot{q}_1}\left(\frac{\dot{x}^2}{2}\right) = \dot{x}\frac{\partial \dot{x}}{\partial \dot{q}_1}, \tag{2.25}$$

and

$$\frac{\partial}{\partial q_1}\left(\frac{\dot{x}^2}{2}\right) = \dot{x}\frac{\partial \dot{x}}{\partial q_1}. \tag{2.26}$$

Substituting (2.25) and (2.26) into (2.24) gives

$$\ddot{x}\frac{\partial x}{\partial q_1} = \frac{d}{dt}\left(\frac{\partial\left(\frac{\dot{x}^2}{2}\right)}{\partial \dot{q}_1}\right) - \frac{\partial\left(\frac{\dot{x}^2}{2}\right)}{\partial q_1}.$$

We can obtain similar expressions for y and z. Taking this into account, the expression (2.17) when $\delta q_2 = 0$ becomes,

$$\begin{aligned}\delta W &= \left(\frac{d}{dt}\left(\frac{\partial}{\partial \dot{q}_1}\frac{m\left(\dot{x}^2+\dot{y}^2+\dot{z}^2\right)}{2}\right) - \frac{\partial}{\partial q_1}\frac{m\left(\dot{x}^2+\dot{y}^2+\dot{z}^2\right)}{2}\right)\delta q_1 \\ &= \left(F_x\frac{\partial x}{\partial q_1} + F_y\frac{\partial y}{\partial q_1} + F_z\frac{\partial z}{\partial q_1}\right)\delta q_1 \\ &= F_{q_1}\delta q_1. \end{aligned} \tag{2.27}$$

Let $K = \frac{1}{2}m\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right)$ denote the kinetic energy of the particle. Then, we represent (2.27) as

$$\boxed{\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_1}\right) - \frac{\partial K}{\partial q_1} = F_{q_1}} \tag{2.28}$$

This equation is called the *Lagrange equation of motion* for the q_1 coordinate. Using the same arguments as above, we can derive the Lagrange equation of motion for the q_2 coordinate. In general there are as many the Lagrange equations of motion as there are degrees of freedom of the particle.

Recall that the expression $\delta W = F_x\delta x + F_y\delta y + F_z\delta z$ is a work done by a force $\boldsymbol{F} = \left[\begin{array}{ccc} F_x & F_y & F_z \end{array}\right]^\top$ for a general displacement $\delta\boldsymbol{s} = \left[\begin{array}{ccc} \delta x & \delta y & \delta z \end{array}\right]^\top$. In terms of generalized coordinates and when, say, $\delta q_2 = 0$, we have

$$\begin{aligned} \delta W &= \left(F_x\frac{\partial x}{\partial q_1} + F_y\frac{\partial y}{\partial q_1} + F_z\frac{\partial z}{\partial q_1}\right)\delta q_1 \\ &= F_{q_1}\delta q_1 \\ &= \delta W_{q_1}. \end{aligned}$$

A generalized force F_{q_r} is of such nature that the product $F_{q_r}\delta q_r$ is the work done by driving forces when q_r alone is changed by δq_r. We mention that a generalized force does not have to be a force in the usual sense. For example, if q_r is an angle, then F_{q_r} must be a torque in order that $F_{q_r}\delta q_r$ be work.

We now derive a more general version of the Lagrange equations of motion for a single particle. In our derivation we need the notion of *nonconservative* forces that we discuss next. First recall that we interpret the kinetic energy of a particle as its ability to do work by virtue of its motion. This motion is the result of conservative and nonconservative forces acting upon the particle. A force is nonconservative if the work done by the force on a particle that moves through any round trip is not zero. Thus a force is nonconservative if the work done by that force on a particle that moves between two points depends on the path taken between these points.

Let us now suppose that in addition to the conservative forces, a single nonconservative force due to friction acts on the particle. Denote by $\sum W_c$ the sum of the work done by the conservative forces, and by W_f the work done by friction. We then have

$$W_f + \sum W_c = \Delta K. \tag{2.29}$$

Thus, if a nonconservative force, like friction, acts on a particle, then the total mechanical energy is not constant, but changes by the amount of work done by a nonconservative force according to the relation (2.29). Let us denote by E the final mechanical energy of the particle and by E_0 its initial mechanical energy. Then, we can write

$$E - E_0 = \Delta E = W_f.$$

The work done by friction, the nonconservative force, is always negative. Hence, the final mechanical energy E is less than the initial mechanical energy. The "lost" mechanical energy is transformed into *internal energy* U_{int}. In other words, the loss of mechanical energy equals the gain in internal energy and we write

$$\Delta E + U_{int} = 0.$$

In general, we have

$$\Delta K + \sum \Delta U + U_{int} + (\text{change in other forms of energy}) = 0.$$

This means that the total energy, kinetic plus potential plus internal plus all other forms of energy, does not change. This is the *principle of the conservation of energy.* To proceed further, recall that any generalized force F_{q_i} acting on our particle can be expressed as

$$F_{q_i} = F_x \frac{\partial x}{\partial q_i} + F_y \frac{\partial y}{\partial q_i} + F_z \frac{\partial z}{\partial q_i}.$$

Assuming that the forces are conservative, we get

$$F_{q_i} = -\left(\frac{\partial U}{\partial x}\frac{\partial x}{\partial q_i} + \frac{\partial U}{\partial y}\frac{\partial y}{\partial q_i} + \frac{\partial U}{\partial z}\frac{\partial z}{\partial q_i} \right).$$

We thus have

$$F_{q_i} = -\frac{\partial U}{\partial q_i}.$$

Therefore, the Lagrange equations of motion for a single particle for conservative forces can be represented as

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) - \frac{\partial K}{\partial q_i} = -\frac{\partial U}{\partial q_i}.$$

We can rearrange the above equations as

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) - \frac{\partial}{\partial q_i}(K - U) = 0.$$

We now introduce the *Lagrangian function* L, defined as

$$L = K - U.$$

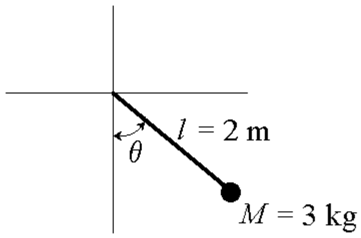


Figure 2.2: A simple pendulum.

Note that because U is not a function of $\dot{q}_i$, we have

$$\frac{\partial K}{\partial \dot{q}_i} = \frac{\partial L}{\partial \dot{q}_i}.$$

Taking the above into account, we can represent the Lagrange equations of motion for conservative forces as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0.$$

If some of the forces acting on the particle are nonconservative, the Lagrange equations of motion for our particle will take the form

$$\boxed{\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tilde{F}_{q_i}} \tag{2.30}$$

where $\tilde{F}_{q_i}$ are nonconservative forces acting on the particle.

Example 2.1 Consider the simple pendulum shown in Figure 2.2. The simple pendulum is an idealized body consisting of a point mass M, suspended by a weightless inextensible cord of length l. The simple pendulum is an example of a one degree of freedom system with a generalized coordinate being the angular displacement θ. The pendulum kinetic energy is

$$K = \frac{1}{2}Ml^2\dot{\theta}^2.$$

We assume that the potential energy of the pendulum is zero when the pendulum is at rest. Hence, its potential energy is

$$U = Mgl(1 - \cos\theta),$$

where $g = 10\,\text{m/sec}^2$ is the acceleration due to gravity. The Lagrangian function is

$$\begin{aligned} L &= K - U \\ &= \frac{1}{2}Ml^2\dot{\theta}^2 - Mgl(1-\cos\theta). \end{aligned}$$

We compute

$$\frac{\partial L}{\partial \theta} = -Mgl\sin\theta$$

and

$$\frac{\partial L}{\partial \dot{\theta}} = Ml^2\dot{\theta}.$$

Hence

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = Ml^2\ddot{\theta}.$$

Combining the above terms together we obtain the Lagrange equation describing the pendulum motion,

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} &= Ml^2\ddot{\theta} + Mgl\sin\theta \\ &= 0. \end{aligned}$$

Solving for $\ddot{\theta}$, we obtain

$$\ddot{\theta} = -\frac{g}{l}\sin\theta. \tag{2.31}$$

In our example the above equation evaluates to

$$\ddot{\theta} = -5\sin\theta. \tag{2.32}$$

Example 2.2 In Figure 2.3, the θ-r robot manipulator is shown. This example was adapted from Snyder [306, Chapter 10]. A lumped mass representation of the θ-r robot manipulator is shown in Figure 2.4. The mass $m_1 = 10\,\text{kg}$ represents the mass of the outer cylinder positioned at its center of mass. The constant distance $r_1 = 1\text{m}$ designates the fixed distance between the center of the mass of the outer cylinder and the center of rotation. The mass of the load is represented by $m_2 = 3\text{kg}$ and is assumed to be located at the end of a piston of a telescoping arm that is a variable radial distance r from the hub or center of rotation. The angle of rotation of the manipulator arm is denoted by θ. The inputs to the system are assumed to be a torque T_θ applied at the hub in the direction of θ, and a translational force F_r applied in the direction r. Finally, $g = 10\,\text{m/s}^2$ is the gravitational constant. We

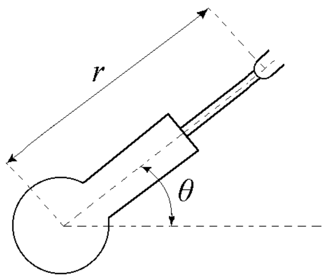


Figure 2.3: The θ-r robot manipulator.

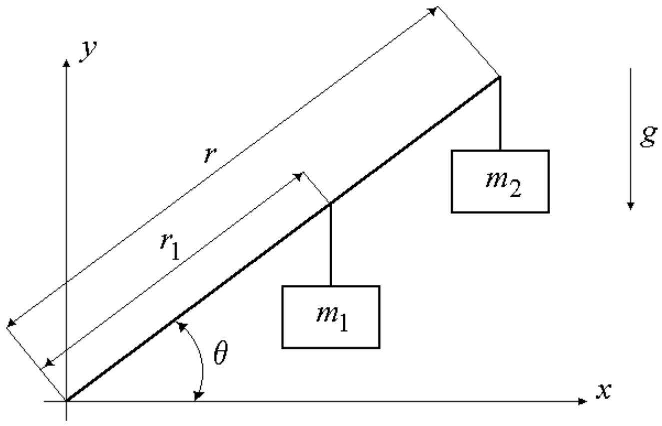


Figure 2.4: A lumped mass representation of the θ-r robot manipulator.

construct the Lagrange equations of motion for the system, and then represent the obtained model in state space format. We first find the total kinetic energy of the system. The position of the mass m_1 is

$$\left.\begin{aligned} x_1 &= r_1\cos\theta \\ y_1 &= r_1\sin\theta. \end{aligned}\right\}$$

Note that r_1 is a constant. Differentiating x_1 and y_1 with respect to time yields

$$\left.\begin{aligned} \dot{x}_1 &= -r_1\dot{\theta}\sin\theta \\ \dot{y}_1 &= r_1\dot{\theta}\cos\theta. \end{aligned}\right\}$$

The magnitude squared of the velocity vector of the mass m_1 is

$$\begin{aligned} v_1^2 &= \dot{x}_1^2 + \dot{y}_1^2 \\ &= r_1^2\dot{\theta}^2\sin^2\theta + r_1^2\dot{\theta}^2\cos^2\theta \\ &= r_1^2\dot{\theta}^2. \end{aligned}$$

Therefore, the kinetic energy of the mass m_1 is

$$K_1 = \frac{1}{2}m_1 v_1^2 = \frac{1}{2}m_1 r_1^2\dot{\theta}^2.$$

We now derive an expression for the kinetic energy of the second mass. The position of the mass m_2 is

$$\left.\begin{aligned} x_2 &= r\cos\theta \\ y_2 &= r\sin\theta. \end{aligned}\right\}$$

Note that the distance r is not a constant. Differentiating x_2 and y_2 with respect to time yields

$$\left.\begin{aligned} \dot{x}_2 &= \dot{r}\cos\theta - r\dot{\theta}\sin\theta \\ \dot{y}_2 &= \dot{r}\sin\theta + r\dot{\theta}\cos\theta. \end{aligned}\right\}$$

The magnitude squared of the velocity vector of the mass m_2 is

$$\begin{aligned} v_2^2 &= (\dot{x}_2)^2 + (\dot{y}_2)^2 \\ &= \left(\dot{r}\cos\theta - r\dot{\theta}\sin\theta\right)^2 + \left(\dot{r}\sin\theta + r\dot{\theta}\cos\theta\right)^2 \\ &= \dot{r}^2 + r^2\dot{\theta}^2. \end{aligned}$$

Hence, the kinetic energy of the mass m_2 is

$$K_2 = \frac{1}{2}m_2 v_2^2 = \frac{1}{2}m_2\left(\dot{r}^2 + r^2\dot{\theta}^2\right).$$

The total kinetic energy of the system is

$$K = K_1 + K_2 = \frac{1}{2}m_1 r_1^2 \dot{\theta}^2 + \frac{1}{2}m_2\left(\dot{r}^2 + r^2\dot{\theta}^2\right).$$

We now find the potential energy of the system. The potential energy of the mass m_1 is

$$U_1 = m_1 g r_1 \sin\theta.$$

The potential energy of the mass m_2 is

$$U_2 = m_2 g r \sin\theta.$$

The total potential energy of the system is

$$U = U_1 + U_2 = m_1 g r_1 \sin\theta + m_2 g r \sin\theta.$$

The Lagrangian function for the system is

$$L = K - U = \frac{1}{2}m_1 r_1^2\dot{\theta}^2 + \frac{1}{2}m_2\dot{r}^2 + \frac{1}{2}m_2 r^2\dot{\theta}^2 - m_1 g r_1 \sin\theta - m_2 g r \sin\theta.$$

The manipulator has two degrees of freedom. Hence, we have two Lagrange's equations of motion for this system,

$$\left.\begin{array}{rcl} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} & = & T_\theta \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{r}}\right) - \frac{\partial L}{\partial r} & = & F_r. \end{array}\right\}$$

The first Lagrange equation yields

$$m_1 r_1^2 \ddot{\theta} + m_2 r^2 \ddot{\theta} + 2m_2 r\dot{r}\dot{\theta} + g\cos\theta(m_1 r_1 + m_2 r) = T_\theta.$$

The second Lagrange equation gives

$$m_2\ddot{r} - m_2 r\dot{\theta}^2 + m_2 g\sin\theta = F_r.$$

We next represent the above modeling equations in state space format. (One can use, for example, MATLAB's Symbolic Math Toolbox to perform needed calculations.) Let the state variables be defined as:

$$z_1 = \theta, \quad z_2 = \dot{\theta}, \quad z_3 = r, \quad z_4 = \dot{r}.$$

Then, we can write

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{-2m_2 z_2 z_3 z_4 - g\cos z_1(m_1 r_1 + m_2 z_3) + u_1}{m_1 r_1^2 + m_2 z_3^2} \\ z_4 \\ z_2^2 z_3 - g\sin z_1 + u_2/m_2 \end{bmatrix}. \tag{2.33}$$

In the following section, we develop models of a few more dynamical systems using basic laws of physics.

2.4 Model Catalog

2.4.1 Centrifugal Governor

In this subsection, we model the centrifugal governor. In our modeling, we follow Pontryagin [276, pages 213–220]. For an alternative approach to modeling the governor see Maxwell [231] and Fuller [109]. Some, like for example MacFarlane [226, page 250], claim that Maxwell founded the theory of automatic control systems with his 1868 paper "On governors". However, this landmark paper is written in a terse and cryptic style. Fuller [109] does a wonderful job elucidating, correcting, and expanding Maxwell's paper. Mayr [235, Chapter X] gives a fascinating historical account of the application of the centrifugal governor to the speed regulation of the steam engine and tracks the origins of the flyball governor.

A schematic of the governor is depicted in Figure 1.6, while forces acting on the governor are shown in Figure 2.5. If the weights rotate with angular velocity ω_{gov}, then the arms form

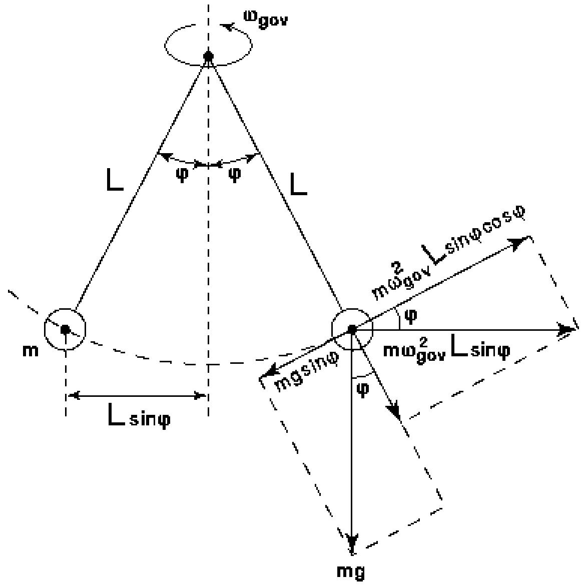


Figure 2.5: Forces acting on the weights in the governor.

an angle φ with the vertical position. Let v_{weight} denote the linear velocity of the weights, and r be the radius of the circle along which the weights rotate. Then, $v_{weight} = \omega_{gov} r$. Note that $r = L\sin\varphi$. The resulting centrifugal force acting on each weight is

$$\begin{aligned} m\frac{v_{weight}^2}{r} &= m\omega_{gov}^2 r \\ &= m\omega_{gov}^2 L\sin\varphi. \end{aligned}$$

Simultaneously, a gravitational force mg acts on each weight. In addition, a frictional force in the hinge joints acts on each weight. We assume that the frictional force is proportional to $\dot{\varphi}$ with the friction coefficient b. Summing forces along the axis perpendicular to an arm yields an equation of motion of the weight

$$\begin{aligned} m\ddot{\varphi}L &= m\omega_{gov}^2 L\sin\varphi\cos\varphi - mg\sin\varphi - b\dot{\varphi} \\ &= \frac{1}{2}m\omega_{gov}^2 L\sin 2\varphi - mg\sin\varphi - b\dot{\varphi}. \end{aligned} \tag{2.34}$$

The governor is connected to the engine via a gear with the gear ratio N such that

$$\omega_{gov} = N\omega, \tag{2.35}$$

where ω is the angular velocity of the flywheel. Substituting (2.35) into (2.34) gives

$$m\ddot{\varphi}L = \frac{1}{2}mN^2\omega^2 L\sin 2\varphi - mg\sin\varphi - b\dot{\varphi}.$$

Let I be the moment of inertia of the flywheel, τ_1 the torque delivered by the engine, and τ_l the load torque. Then, the equation modeling the engine is

$$I\dot{\omega} = \tau_1 - \tau_l. \tag{2.36}$$

The collar of the governor is connected with the valve so that at the nominal value of $\varphi = \varphi_0$ the engine delivers the nominal torque τ_d that results in the desired angular velocity $\omega = \omega_d$ of the flywheel. We model the interaction of the collar and the valve using the equation

$$\tau_1 = \tau_d + \kappa(\cos\varphi - \cos\varphi_0), \tag{2.37}$$

where $\kappa > 0$ is a proportionality constant that depends on the governor. Substituting (2.37) into (2.36) yields

$$\begin{aligned} I\dot{\omega} &= \tau_1 - \tau_l \\ &= \tau_d + \kappa(\cos\varphi - \cos\varphi_0) - \tau_l \\ &= \kappa\cos\varphi - \tau, \end{aligned}$$

where $\tau = \tau_l - \tau_d + \kappa \cos \varphi_0$. We thus have two equations that model the governor,

$$\begin{aligned} m\ddot{\varphi}L &= \frac{1}{2} m N^2 \omega^2 L \sin 2\varphi - mg \sin\varphi - b\dot{\varphi}, \\ I\dot{\omega} &= \kappa \cos\varphi - \tau. \end{aligned}$$

We use the state variables

$$x_1 = \varphi, \quad x_2 = \dot{\varphi}, \quad \text{and} \quad x_3 = \omega$$

to represent the above model in state space format,

$$\left.\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \tfrac{1}{2} N^2 x_3^2 \sin 2x_1 - \tfrac{g}{L} \sin x_1 - \tfrac{b}{mL} x_2 \\ \dot{x}_3 &= \tfrac{\kappa}{I} \cos x_1 - \tfrac{\tau}{I}. \end{aligned}\right\} \tag{2.38}$$

2.4.2 Ground Vehicle

In this subsection, we model a ground vehicle in a turning maneuver, such as, a step lane change maneuver. During such a maneuver, the vehicle undergoes translational as well as rotational motion. When modeling a moving vehicle, it is convenient to use reference coordinate frame attached to and moving with the vehicle. This is because with respect to the vehicle coordinate frame moments of inertias of the vehicle are constant. We use the Society of Automotive Engineers (SAE) standard coordinate system as shown in Figure 2.6. Our modeling follows that of Wong [354] and Will and Żak [350]. We use lower case letters to label axes of the vehicle coordinate system. Following the SAE convention, we define the coordinates as follows:

- the x-coordinate is directed forward and on the longitudinal plane of symmetry of the vehicle,
- the y-coordinate has a lateral direction out the right side of the vehicle,
- the z-coordinate is directed downward with respect to the vehicle,
- the yaw velocity about the z-axis is denoted $\dot{\theta}$.

In developing our simple model, we neglect the effects of rollindexroll—seeground vehicle and pitchindexpitch—seeground vehicle. The origin of the vehicle coordinate system is located at the vehicle center of gravity (CG). We also use a global, or fixed coordinate system. Its axes are labeled using upper case letters as shown in Figure 2.7. To form the equations of motion

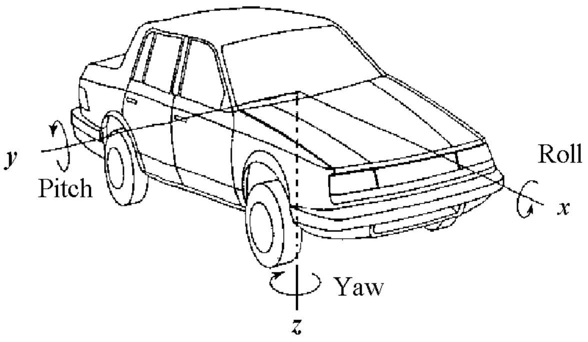


Figure 2.6: The Society of Automotive Engineers (SAE) coordinate system attached to the vehicle.

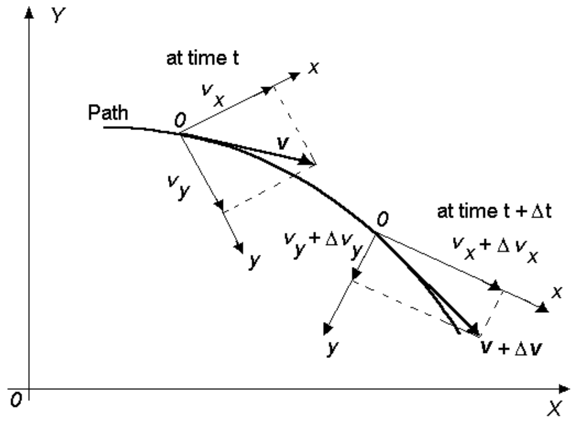


Figure 2.7: Analyzing vehicle motion using coordinate frame attached to the vehicle; snapshots of the vehicle velocity at times t and $t + \Delta t$.

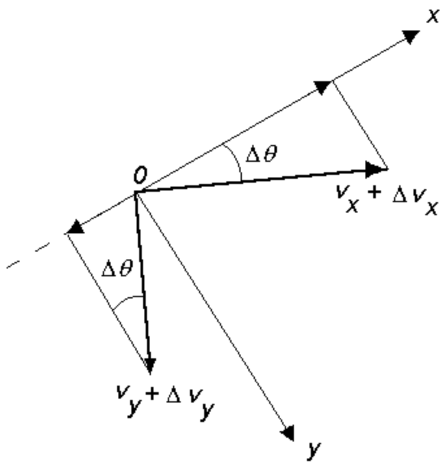


Figure 2.8: Computing the change in the longitudinal and lateral components of the vehicle velocity.

of the vehicle, we need an expression for the vehicle CG acceleration. We first compute the change in the longitudinal component of the velocity. Using Figure 2.7 and Figure 2.8, and assuming small angles, we obtain

$$\begin{aligned}(v_x + \Delta v_x)\cos\Delta\theta - (v_y + \Delta v_y)\sin\Delta\theta - v_x &\approx \Delta v_x - v_y \Delta\theta \\ &= \Delta\dot{x} - \dot{y}\Delta\theta.\end{aligned}$$

We form the Newton's quotient using the above. Then, taking the limit, we obtain the total longitudinal acceleration component a_x,

$$\begin{aligned}a_x &= \lim_{\Delta t\to 0}\frac{\Delta v_x - v_y\Delta\theta}{\Delta t} \\ &= \dot{v}_x - v_y\dot{\theta} \\ &= \ddot{x} - \dot{y}\omega_z, \end{aligned} \tag{2.39}$$

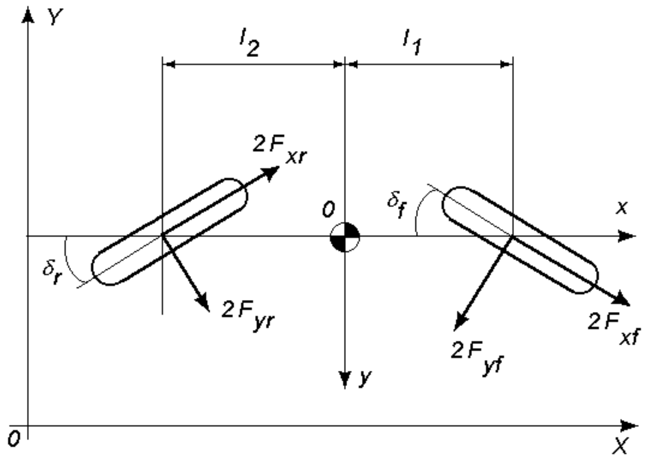


Figure 2.9: A simplified, bicycle model of a vehicle.

where $\omega_z = \dot{\theta}$. Using similar argument to the one above, we obtain the lateral acceleration component a_y to be

$$\begin{aligned} a_y &= \dot{v}_y + v_x\dot{\theta} \\ &= \ddot{y} + \dot{x}\omega_z. \end{aligned} \tag{2.40}$$

We assume that the vehicle is symmetric with respect to the x-z plane. We use the so called bicycle model that is shown in Figure 2.9, where δ_f denotes the front wheels' steering angle and δ_r is the rear wheels' steering angle. We denote by $2F_{xf}$ lumped longitudinal force at the front axle and by $2F_{xr}$ lumped longitudinal force at the rear axle. The lumped lateral force at the front axle is denoted $2F_{yf}$, while lumped lateral force at the rear axle is denoted $2F_{yr}$. Summing the forces along the x axis and assuming small angles, we obtain

$$\begin{aligned} m\left(\ddot{x} - \dot{y}\omega_z\right) &= 2F_{xf}\cos\delta_f - 2F_{yf}\sin\delta_f + 2F_{xr}\cos\delta_r + 2F_{yr}\sin\delta_r \\ &\approx 2F_{xf} + 2F_{xr}. \end{aligned} \tag{2.41}$$

Next, summing the forces along the y axis and assuming small angles yields

$$\begin{aligned} m\left(\ddot{y} + \dot{x}\omega_z\right) &= 2F_{xf}\sin\delta_f + 2F_{yf}\cos\delta_f + 2F_{yr}\cos\delta_r - 2F_{xr}\sin\delta_r \\ &\approx 2F_{yf} + 2F_{yr}. \end{aligned} \tag{2.42}$$

The third equation of motion is obtained by summing the torques with respect to the z-axis. Let I_z denote the moment of inertia of the vehicle about the z axis. Then, referring again to Figure 2.9 and assuming small angles, we obtain:

$$\begin{aligned} I_z\dot{\omega}_z &= 2l_1F_{yf}\cos\delta_f + 2l_1F_{xf}\sin\delta_f - 2l_2F_{yr}\cos\delta_r + 2l_2F_{xr}\sin\delta_r \\ &\approx 2l_1F_{yf} - 2l_2F_{yr}. \end{aligned} \tag{2.43}$$

In our further analysis, we assume $v_x = \dot{x} =$ constant, that is, the driver neither is accelerating nor braking the vehicle. This means that we need to use only the last two equations of motion in our model. To proceed, we need to define tire slip angles and cornering stiffness coefficients. The tire slip angle is the difference between the tire's actual travel direction and the tire's heading direction, which is the steer angle. In other words, the tire slip angle is the angle between the desired direction of motion of the tire specified by the steer angle and the actual direction of travel of the center of the tire contact. We denote by $C_{\alpha f}$ the cornering stiffness coefficient of the front tires and by $C_{\alpha r}$ the cornering stiffness coefficient of the rear tires. The coefficients $C_{\alpha f}$ and $C_{\alpha r}$ are measured experimentally. A typical relation between a tire slip angle and a cornering force is shown in Figure 2.10. It is a nonlinear relation, where the cornering coefficient is

$$C_{\alpha w} = \frac{dF_{yw}}{d\alpha_w},$$

where $w = f$ or $w = r$. The lateral forces F_{yf} and F_{yr} that appear in (2.41)–(2.43) are functions of the corresponding tire slip angles α_f and α_r and cornering stiffness coefficients $C_{\alpha f}$ and $C_{\alpha r}$. In our simple linearized model, we assume that $C_{\alpha f} = C_{\alpha r} =$ constant, and model dependencies between the lateral forces and tire slip angles as

$$F_{yf} = C_{\alpha f}\alpha_f, \tag{2.44}$$

and

$$F_{yr} = C_{\alpha r}\alpha_r. \tag{2.45}$$

We formulate expressions for the tire slip angles using Figure 2.11. Assuming small angles, we form an expression for the front tire slip angle

$$\begin{aligned} \alpha_f &= \delta_f - \tan^{-1}\left(\frac{l_1\omega_z + v_y}{v_x}\right) \\ &\approx \delta_f - \frac{l_1\omega_z + v_y}{v_x} \\ &= \delta_f - \frac{l_1\omega_z + \dot{y}}{\dot{x}}. \end{aligned} \tag{2.46}$$

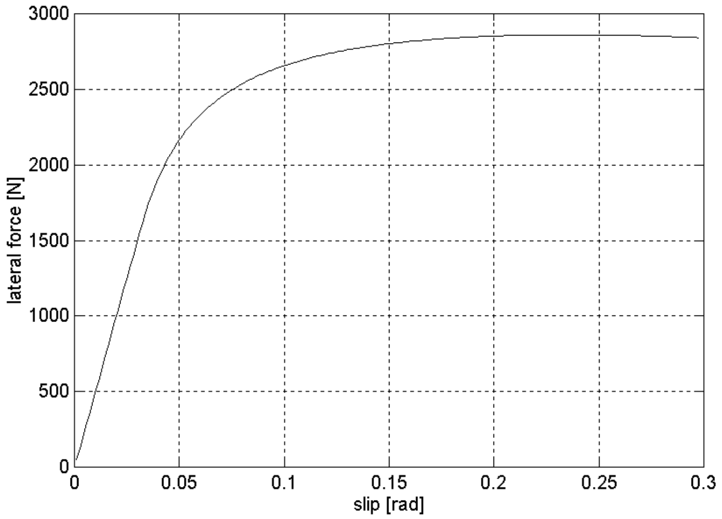


Figure 2.10: Typical cornering characteristic.

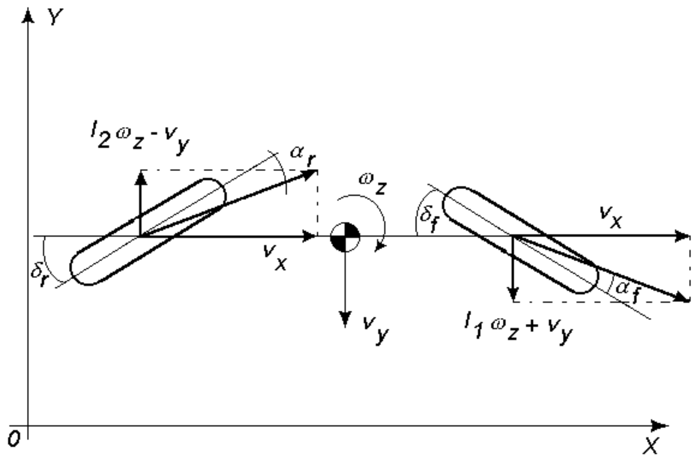


Figure 2.11: Simplified vehicle model for computing tire slip angles.

The rear tire slip angle is

$$\begin{aligned}
\alpha_r &= \delta_r - \tan^{-1}\left(-\frac{l_2\omega_z - v_y}{v_x}\right) \\
&= \delta_r + \tan^{-1}\left(\frac{l_2\omega_z - v_y}{v_x}\right) \\
&\approx \delta_r + \frac{l_2\omega_z - v_y}{v_x} \\
&= \delta_r + \frac{l_2\omega_z - \dot{y}}{\dot{x}}. \qquad (2.47)
\end{aligned}$$

Hence

$$F_{yf} = C_{\alpha f}\left(\delta_f - \frac{l_1\omega_z + \dot{y}}{\dot{x}}\right), \qquad (2.48)$$

and

$$F_{yr} = C_{\alpha r}\left(\delta_r + \frac{l_2\omega_z - \dot{y}}{\dot{x}}\right). \qquad (2.49)$$

Substituting the above two equations into (2.42), and assuming small angles yields

$$m\ddot{y} + m\dot{x}\omega_z = 2C_{\alpha f}\delta_f - 2C_{\alpha f}\frac{l_1\omega_z + \dot{y}}{v_x} + 2C_{\alpha r}\delta_r + 2C_{\alpha r}\frac{l_2\omega_z - \dot{y}}{v_x}.$$

Rearranging terms and performing some manipulations, we get

$$\ddot{y} = -\left(\frac{2C_{\alpha f} + 2C_{\alpha r}}{mv_x}\right)\dot{y} + \left(-v_x - \frac{2C_{\alpha f}l_1 - 2C_{\alpha r}l_2}{mv_x}\right)\omega_z + \frac{2C_{\alpha f}}{m}\delta_f + \frac{2C_{\alpha r}}{m}\delta_r. \tag{2.50}$$

Substituting the expressions for F_{yf} and F_{yr} into (2.43) and assuming small angles, we obtain

$$I_z\dot{\omega}_z = 2l_1C_{\alpha f}\left(\delta_f - \frac{l_1\omega_z + \dot{y}}{v_x}\right) - 2l_2C_{\alpha r}\left(\delta_r + \frac{l_2\omega_z - \dot{y}}{v_x}\right).$$

Rearranging and performing manipulations yields

$$\dot{\omega}_z = -\left(\frac{2l_1C_{\alpha f} - 2l_2C_{\alpha r}}{I_zv_x}\right)\dot{y} - \left(\frac{2l_1^2C_{\alpha f} + 2l_2^2C_{\alpha r}}{I_zv_x}\right)\omega_z + \frac{2l_1C_{\alpha f}}{I_z}\delta_f - \frac{2l_2C_{\alpha r}}{I_z}\delta_r. \tag{2.51}$$

Later, we will analyze a vehicle performance in the fixed coordinates. In particular, we will be interested in an emergency lane change maneuver. This requires that we transform the vehicle lateral variable into the fixed coordinates. Assuming small angles and referring to Figure 2.12, we obtain

$$\begin{aligned}\dot{Y} &= -\dot{x}\sin\psi - \dot{y}\cos\psi \\ &\approx -v_x\psi - \dot{y}.\end{aligned} \tag{2.52}$$

Note that

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi.$$

For the step lane change, the yaw rate of the vehicle is equal to the yaw rate error, that is,

$$\omega_z = \dot{\psi}.$$

We further assume that

$$\theta = \psi. \tag{2.53}$$

We can now represent the vehicle modeling equations in a state space format. Let

$$x_1 = \dot{y}, \quad x_2 = \theta, \quad x_3 = \omega_z, \quad x_4 = Y.$$

Then, using (2.50), (2.51), (2.53) and (2.52), we obtain

$$\begin{aligned}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} &= \begin{bmatrix} -\frac{2C_{\alpha f}+2C_{\alpha r}}{mv_x} & 0 & -v_x - \frac{2C_{\alpha f}l_1 - 2C_{\alpha r}l_2}{mv_x} & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{2l_1C_{\alpha f}-2l_2C_{\alpha r}}{I_zv_x} & 0 & -\frac{2l_1^2C_{\alpha f}+2l_2^2C_{\alpha r}}{I_zv_x} & 0 \\ -1 & -v_x & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} \\ &+ \begin{bmatrix}\frac{2C_{\alpha f}}{m} & \frac{2C_{\alpha r}}{m} \\ 0 & 0 \\ \frac{2l_1C_{\alpha f}}{I_z} & -\frac{2l_2C_{\alpha r}}{I_z} \\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_f\\ \delta_r\end{bmatrix}.\end{aligned} \tag{2.54}$$

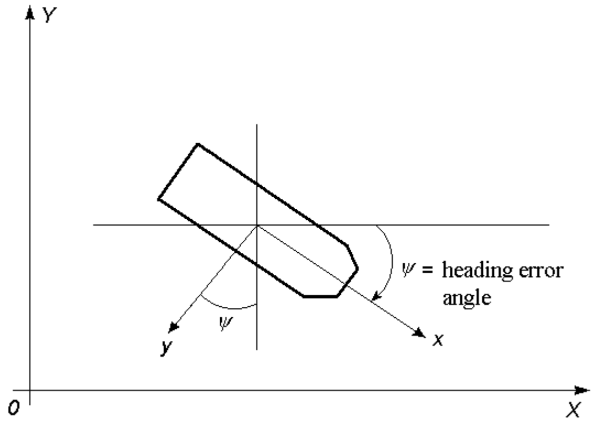


Figure 2.12: Computing the velocity of a vehicle with respect to the fixed coordinates.

A possible vehicle parameters are given in Table 2.1.

2.4.3 Permanent Magnet Stepper Motor

In this subsection, we develop a model of a permanent magnet (PM) stepper motor. Stepper motors can be viewed as digital positioning devices. Specifically, stepper motors are electromechanical motion devices that convert input information in digital form into output that has mechanical form. The stepper motor can be driven by digital pulses, for example of the form: move clockwise or move counterclockwise. We now describe an idea of an operation

Table 2.1: Typical vehicle data.

$C_{\alpha f} = C_{\alpha r}$	$30,000.00\,\mathrm{N/rad}$
l_1	$1.20\,\mathrm{m}$
l_2	$1.22\,\mathrm{m}$
m	$1280\,\mathrm{kg}$
I_z	$2500\,\mathrm{kg{\cdot}m^2}$
v_x	$18.3\,\mathrm{m/sec}$

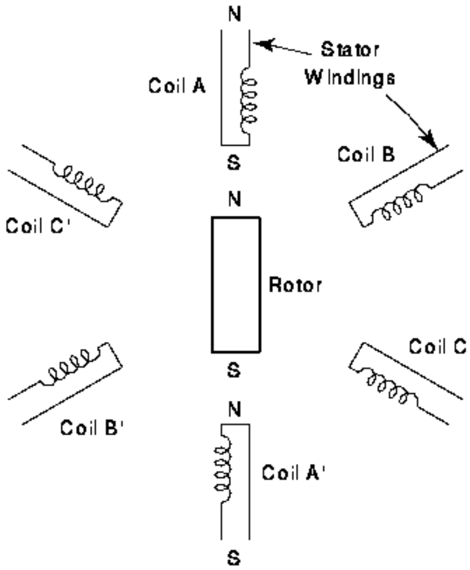


Figure 2.13: A simplified schematic of an operation of a stepper motor.

of a PM stepper motor and then derive its modeling equations.

A simplified organization of a stepper motor is shown in Figure 2.13. The armature, or rotor, is a magnet. At any instant of the operation, exactly one pair of windings of the stator is energized. A pair of windings represents a phase. Each pair of the stator windings is wound in such a way that their magnetic fields are collinear, for example, windings A and A'. The rotor then rotates until its magnetic field is aligned with the fields of the energized pair of the stator windings. Figure 2.13 shows the situation where the pair of windings A and A' is energized. If the rotor is at rest, as shown in Figure 2.13, and the pair A, A' is de-energized, while the pair B, B' is energized, the rotor rotates to the new position so that its magnetic field is aligned with the magnetic field of the pair B, B'. In this example of the three-phase motor, a single step results in the rotor motion of $360°/6 = 60°$. We next give a more detailed description of the electrical organization of the stepper motor and construct its model.

The basic PM stepper motor consists of a slotted stator and a rotor. There can be two sets of teeth on the rotor. These two sets of teeth are out of alignment with each other by

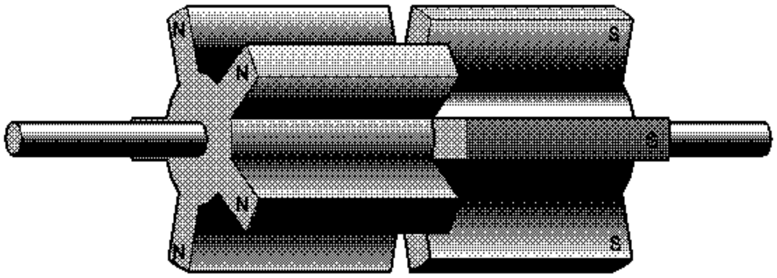


Figure 2.14: A possible rotor structure of a PM stepper motor.

a tooth width, as shown in Figure 2.14. Specifically, the rotor teeth of the left end-cap are displaced one half pitch from the teeth on the right end-cap. A magnet, or an electromagnet, that is mounted on the rotor magnetizes the iron end-caps. The left end-cap is magnetized as a north pole, while the right end-cap is magnetized as a south pole. The slotted stator is equipped with two or more individual coils. Figure 2.15 shows a four-pole stator and a five-pole rotor structure.

We now discuss the way in which the PM stepper motor can be used to perform precise positioning. Assume that the starting point is as represented in Figure 2.16. Here, coil A-A' is energized so that the upper stator pole is given an S-pole polarity and the corresponding lower stator is given an N-pole polarity. The rotor remains in this position as long as stator coil voltage, and hence stator polarities, remain unchanged. This is because the net torque acting on the rotor is zero. If we now de-energize coil A-A' and energize coil B-B' so that the left pole becomes an S-pole and the right pole assumes an N-polarity, then this results in the new steady-state position shown in Figure 2.17. We can proceed by alternatively energizing the two stator coils. The PM stepper motor can thus be controlled by digital pulses applied

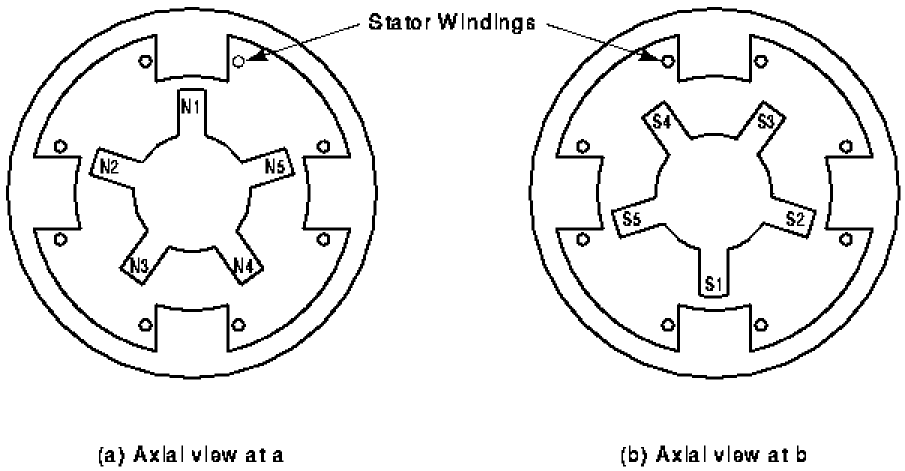


Figure 2.15: Construction features of a four-pole stator and a five-pole rotor PM stepper motor.

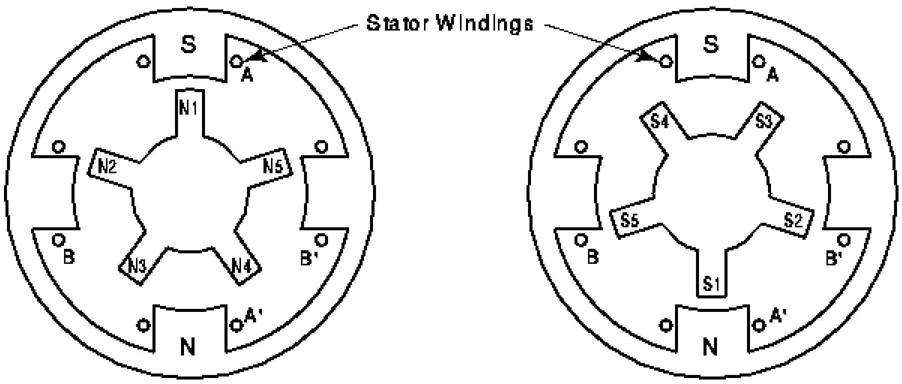


Figure 2.16: Coil A-A' energized for an S-N orientation, while coil B-B' de-energized.

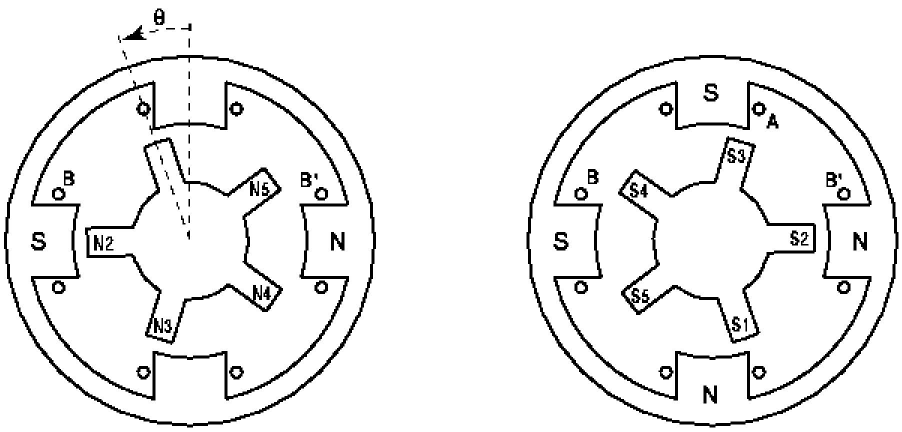


Figure 2.17: Coil A-A' de-energized, while coil B-B' energized for an S-N orientation.

to its two inputs: move clockwise or move counterclockwise.

Our next objective is to devise a mathematical model of the PM stepper motor. In our discussion we use the following notation. We denote by R_t the number of rotor teeth. The angular displacement between rotor teeth is called the tooth pitch and denoted T_p. The angular displacement of the rotor as we change the input voltage from one pair of windings to the other is called the step length and denoted S_l. The number of phases, that is, the number of pairs of stator windings, is denoted N_p. In our example, we have $N_p = 2$, $R_t = 5$, and $T_p = 2\pi/R_t = 72°$. In this example, it takes four input voltage switchings, and hence steps, to advance the rotor one tooth pitch. Indeed, because

$$T_p = 2N_pS_l = \frac{2\pi}{R_t}, \tag{2.55}$$

hence

$$S_l = \frac{\pi}{R_tN_p} = 18°. \tag{2.56}$$

We denote the rotor angular displacement by θ.

Following Krause and Wasynczuk [192, pages 350–351], we trace the main path of flux linking the A-A' windings for the rotor position shown in Figure 2.18. Flux leaves the left end-cap through the rotor tooth at the top that is aligned with the stator tooth having the A part of the A-A' winding. The flux then travels up through the stator tooth in the stator iron. Next, the flux splits and travels around the circumference of the stator, as shown in Figure 2.18(a). The flux returns to the south pole of the rotor through the stator tooth positioned at the bottom of the A' part of the A-A' winding, as shown in Figure 2.18(b). We denote the peak flux linkage produced by the rotor permanent magnet, in the A-A' stator winding by $n\phi_m$, where n is the number of turns of the winding. For $\theta = 0°$, corresponding to the rotor position in Figure 2.18, the flux linkage in the A-A' winding of the stator is maximal and is approximately equal to $n\phi_m$. For $\theta = 90°/R_t$, corresponding to the rotor position in Figure 2.17, the flux linkage in the A-A' winding of the stator is minimal, approximately equal to zero. We model the flux linkage, ψ_a, produced by the permanent magnet, in the A-A' stator winding as

$$\psi_a = n\phi_m \cos(R_t\theta). \tag{2.57}$$

The (main) flux, linking the B-B' winding for the rotor position shown in Figure 2.17, would enter the stator tooth on which the B part of the B-B' winding is located from the rotor tooth on the left. The flux then would travel around the circumference of the stator and return to the rotor through the stator pole on which the B' part of the B-B' winding is wound. For $\theta = 90°/R_t$, corresponding to the rotor position in Figure 2.17, the flux linkage

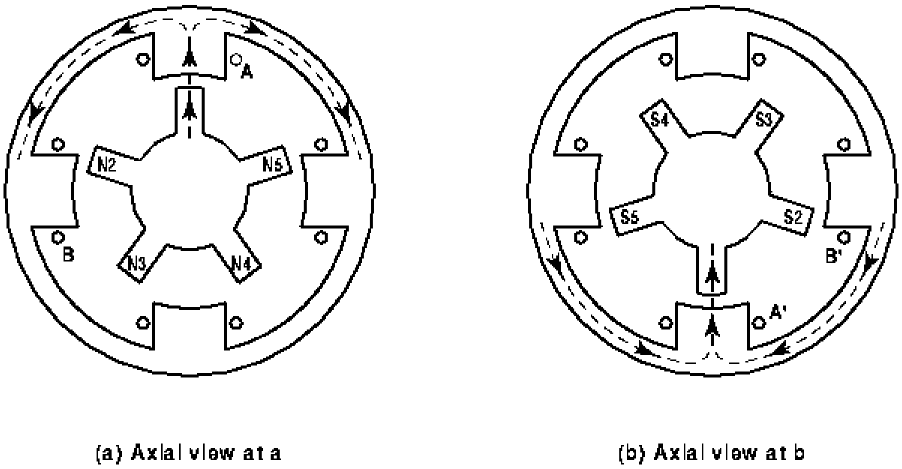


Figure 2.18: The main path of flux linking the A-A' winding.

Table 2.2: Symbols used in modeling equations of the PM stepper motor.

T_p = tooth pitch
S_l = step length
R_t = number of teeth on rotor
N_p = number of phases, that is, stator winding pairs
θ = angular displacement of the rotor
ω = angular velocity of the rotor
$v_a(t)$ = voltage applied to phase a, that is, A-A' stator winding
$v_b(t)$ = voltage applied to phase b, that is, B-B' stator winding
$i_a(t)$ = current in phase a
$i_b(t)$ = current in phase b
R = stator winding resistance per phase
L = stator winding inductance per phase
B = viscous friction coefficient
I = rotor inertia
K_b = back emf constant

in the B-B' winding of the stator is maximal and is approximately equal to $n\phi_m$. For $\theta = 0°$, corresponding to the rotor position in Figure 2.18, the flux linkage in the B-B' winding is minimal, approximately equal to zero. We model the flux linkage in the B-B' phase of the stator by

$$\psi_b = n\phi_m \sin(R_t\theta). \tag{2.58}$$

In the following development, we use symbols that are described in Table 2.2. Applying Kirchhoff's voltage law to the circuit of phase a yields

$$v_a(t) = Ri_a(t) + L\frac{di_a(t)}{dt} + e_a(t),$$

where e_a is the back emf induced in phase a, that is,

$$\begin{aligned} e_a &= \frac{d\psi_a}{dt} \\ &= \frac{d\psi_a}{d\theta}\frac{d\theta}{dt} \\ &= -nR_t\phi_m \sin(R_t\theta)\frac{d\theta}{dt} \\ &= -K_b\,\omega \sin(R_t\theta), \end{aligned}$$

where $K_b = nR_t\phi_m$. Hence, the voltage equation for phase a can be represented as

$$v_a = Ri_a + L\frac{di_a}{dt} - K_b\,\omega\sin(R_t\theta).$$

Applying now Kirchhoff's voltage law to the circuit of phase b yields

$$v_b(t) = Ri_b(t) + L\frac{di_b(t)}{dt} + e_b(t),$$

where e_b is the back emf induced in phase b, that is,

$$\begin{aligned} e_b &= \frac{d\psi_b}{dt} \\ &= \frac{d\psi_b}{d\theta}\frac{d\theta}{dt} \\ &= nR_t\phi_m\cos(R_t\theta)\frac{d\theta}{dt} \\ &= K_b\,\omega\cos(R_t\theta), \end{aligned}$$

where as before $K_b = nR_t\phi_m$. Hence, the voltage equation for phase b can be represented as

$$v_b = Ri_b + L\frac{di_b}{dt} + K_b\,\omega\cos(R_t\theta).$$

Let T_a and T_b denote torques produced by the currents i_a and i_b, respectively. Then, applying Newton's second law to the rotor yields

$$I\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} = T_a + T_b.$$

The PM stepper motor is an electromechanical device in which electric power is transformed into mechanical power. The mechanical power, P_a, due to torque T_a is

$$P_a = T_a\frac{d\theta}{dt}. \tag{2.59}$$

Assuming that there are no losses, this mechanical power is equal to the power developed in phase a,

$$P_a = i_a\frac{d\psi_a}{dt} = i_a\frac{d\psi_a}{d\theta}\frac{d\theta}{dt}. \tag{2.60}$$

Equating the right hand sides of (2.59) and (2.60) and performing simple manipulations, we obtain

$$T_a = i_a\frac{d\psi_a}{d\theta} = -i_aK_b\sin(R_t\theta).$$

Applying similar arguments to phase b, we get

$$T_b = i_b\frac{d\psi_b}{d\theta} = i_bK_b\cos(R_t\theta).$$

Taking the above equations into account, we represent the equation of motion of the rotor as

$$I\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} = T_a + T_b = -i_a K_b \sin(R_t\theta) + i_b K_b \cos(R_t\theta).$$

Combining the above equations, we obtain a model of a two-phase, that is, a two input PM stepper motor. We have

$$\left.\begin{array}{rcl}
\frac{d\theta}{dt} & = & \omega \\
\frac{d\omega}{dt} & = & -\frac{K_b}{I} i_a \sin(R_t\theta) + \frac{K_b}{I} i_b \cos(R_t\theta) - \frac{B}{I}\omega \\
\frac{di_a}{dt} & = & -\frac{R}{L} i_a + \frac{K_b}{L}\omega \sin(R_t\theta) + \frac{1}{L} v_a \\
\frac{di_b}{dt} & = & -\frac{R}{L} i_b - \frac{K_b}{L}\omega \cos(R_t\theta) + \frac{1}{L} v_b.
\end{array}\right\}$$

We now define the state variables: $x_1 = \theta$, $x_2 = \omega$, $x_3 = i_a$, and $x_4 = i_b$. We let $K_1 = B/I$, $K_2 = K_b/I$, $K_3 = R_t$, $K_4 = K_b/L$, $K_5 = R/L$, $u_1 = \frac{1}{L}v_a$, and $u_2 = \frac{1}{L}v_b$. Using the above notation, we represent the PM stepper motor model in state space format:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ -K_1 x_2 - K_2 x_3 \sin(K_3 x_1) + K_2 x_4 \cos(K_3 x_1) \\ K_4 x_2 \sin(K_3 x_1) - K_5 x_3 \\ -K_4 x_2 \cos(K_3 x_1) - K_5 x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.61}$$

2.4.4 Stick Balancer

In this subsection, we develop a nonlinear model of the stick balancer, also known as the cart with an inverted pendulum. In our derivation of the model, we use Newton's laws, in a similar fashion as Kwakernaak and Sivan [199], and Ogata [255] did, where linearized models of the stick balancer were developed. An alternative derivation, using the method of D'Alembert, can be found in Cannon [50, Section 22.4].

A free body diagram of a cart on which a stick, that is, an inverted pendulum, is mounted is shown in Figure 2.19. Let $H = H(t)$ and $V = V(t)$ be horizontal and vertical reaction forces, respectively. The x and y are the coordinates of the fixed, nonrotating coordinate frame $x - y$. The angular displacement of the stick from the vertical position is $\theta = \theta(t)$. The mass of the cart is M, while the mass of the stick is m. The length of the stick is $2l$, and its center of gravity is at its geometric center. The control force applied to the cart is labeled as u. We assume that the wheels of the cart do not slip. We model the frictional force of the cart wheels on the track by

$$f_c = \mu_c \text{sign}(\dot{x}), \tag{2.62}$$

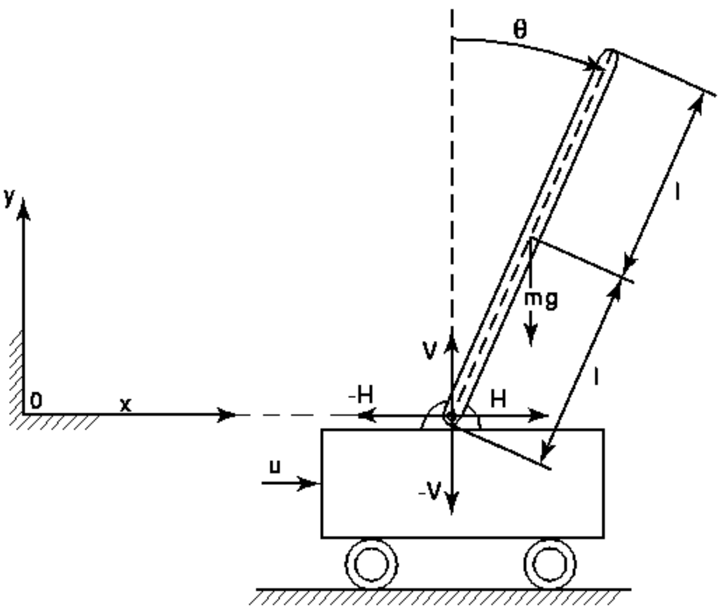


Figure 2.19: Free body diagram of a cart with an inverted pendulum.

where μ_c is the cart friction coefficient. Friedland [108, page 201] refers to this model of friction as the classical Coulomb friction model. Let (x_G, y_G) be the coordinates of the center of gravity of the stick. Then,

$$\begin{cases} x_G &= x + l\sin(\theta) \\ y_G &= l\cos(\theta). \end{cases} \tag{2.63}$$

We are now ready to write down equations modeling the system. The equation that describes the rotational motion of the stick about its center of gravity is obtained by applying the rotational version of Newton's second law. Summing the moments about the center of gravity of the stick, we obtain

$$I\frac{d^2\theta}{dt^2} = Vl\sin(\theta) - Hl\cos(\theta), \tag{2.64}$$

where

$$I = \int_{-l}^{l} r^2 dm = ml^2/3, \tag{2.65}$$

is the moment of inertia of the stick with respect to its center of gravity.

We next write the equation that describes the horizontal motion of the center of gravity of the stick. Applying Newton's second law along the x axis yields

$$m\frac{d^2}{dt^2}\left(x + l\sin(\theta)\right) = H. \tag{2.66}$$

Performing the required differentiation, we get

$$m\left(\ddot{x} + l\left(-\dot{\theta}^2\sin(\theta) + \ddot{\theta}\cos(\theta)\right)\right) = H. \tag{2.67}$$

The equation that describes the vertical motion of the center of gravity of the stick is obtained by applying Newton's second law along the y axis. We have

$$m\frac{d^2}{dt^2}\left(l\cos(\theta)\right) = V - mg. \tag{2.68}$$

Performing the needed differentiation yields

$$ml\left(-\dot{\theta}^2\cos(\theta) - \ddot{\theta}\sin(\theta)\right) = V - mg. \tag{2.69}$$

Finally, we apply Newton's second law to the cart to get

$$M\frac{d^2x}{dt^2} = u - H - f_c. \tag{2.70}$$

Substituting (2.70) into (2.67) gives

$$m\ddot{x} + ml\ddot{\theta}\cos(\theta) - ml\dot{\theta}^2\sin(\theta) + f_c = u - M\ddot{x}. \tag{2.71}$$

Substituting now equations (2.69) and (2.70) into (2.64) yields

$$I\ddot{\theta} = \left(mg - ml\dot{\theta}^2\cos(\theta) - ml\ddot{\theta}\sin(\theta)\right) l\sin(\theta) + f_c l\cos(\theta) - (u - M\ddot{x})\, l\cos(\theta). \tag{2.72}$$

We next substitute $u - M\ddot{x}$ from (2.71) into (2.72) and perform manipulations to get

$$I\ddot{\theta} = mgl\sin(\theta) - ml^2\ddot{\theta} - m\ddot{x}l\cos(\theta). \tag{2.73}$$

Let

$$a = \frac{1}{m+M}.$$

Then, we can represent (2.71) as

$$\ddot{x} = -mal\ddot{\theta}\cos(\theta) + mal\dot{\theta}^2\sin(\theta) - af_c + au. \tag{2.74}$$

We substitute (2.74) into (2.73) to obtain

$$\ddot{\theta} = \frac{mgl\sin(\theta) - m^2l^2a\dot{\theta}^2\sin(2\theta)/2 + mal\cos(\theta)f_c - mal\cos(\theta)u}{I - m^2l^2a\cos^2(\theta) + ml^2}. \tag{2.75}$$

Let $x_1 = \theta$ and $x_2 = \dot{\theta}$. Using the expression for I, given by (2.65), we represent (2.75) in state space format as follows

$$\left.\begin{array}{rcl} \dot{x}_1 &=& x_2 \\ \dot{x}_2 &=& \frac{g\sin(x_1) - mlax_2^2\sin(2x_1)/2 + a\cos(x_1)f_c}{4l/3 - mla\cos^2(x_1)} - \frac{mal\cos(x_1)u}{4ml^2/3 - m^2l^2a\cos^2(x_1)}. \end{array}\right\} \tag{2.76}$$

We next substitute $\ddot{\theta}$, obtained from (2.73), into (2.74) to get

$$\ddot{x} = \frac{-mag\sin(2x_1)/2 + ax_2^2\sin(x_1)4ml/3 + (u - f_c)4a/3}{4/3 - ma\cos^2(x_1)}. \tag{2.77}$$

Let $x_3 = x$ and $x_4 = \dot{x}$. Then, combining (2.76) and (2.77) we obtain a state space model of the inverted pendulum on a cart of the form,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g\sin(x_1) - mlax_2^2\sin(2x_1)/2}{4l/3 - mla\cos^2(x_1)} \\ x_4 \\ \frac{-mag\sin(2x_1)/2 + alx_2^2\sin(x_1)4m/3}{4/3 - ma\cos^2(x_1)} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-a\cos(x_1)}{4l/3 - mla\cos^2(x_1)} \\ 0 \\ \frac{4a/3}{4/3 - ma\cos^2(x_1)} \end{bmatrix} (u - f_c). \tag{2.78}$$

While designing a controller for the stick balancer system, we will use an equivalent representation of (2.78),

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g\sin(x_1)}{4l/3 - mla\cos^2(x_1)} \\ x_4 \\ \frac{-mag\sin(2x_1)/2}{4/3 - ma\cos^2(x_1)} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-a\cos(x_1)}{4l/3 - mla\cos^2(x_1)} \\ 0 \\ \frac{4a/3}{4/3 - ma\cos^2(x_1)} \end{bmatrix} (u - f_c + mlx_2^2\sin(x_1)). \tag{2.79}$$

2.4.5 One-Link Robot Manipulator

The example comes from [368]. It is a simple model of a one-link robot manipulator shown in Figure 2.20. The motion of the robot arm is controlled by a DC motor via a gear. The

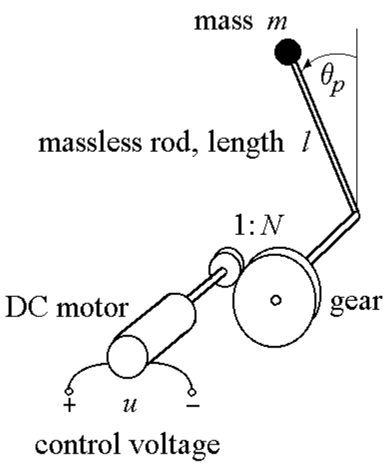


Figure 2.20: One-link manipulator controlled by a DC motor via a gear.

DC motor is armature-controlled and its schematic is shown in Figure 2.21. We assume that the motor moment of inertia is negligible compared with that of the robot arm. We model the arm as a point mass m attached to the end of a massless rod of length l. Hence the arm inertia is $I_a = ml^2$. We assume that the gear train has no backlash, and all connecting shafts are rigid. As can be seen from Figure 2.20, counterclockwise rotation of the arm is defined as positive and clockwise rotation of the arm as negative, while counter-clockwise rotation of the motor shaft is defined to be negative, and clockwise rotation of the shaft as positive.

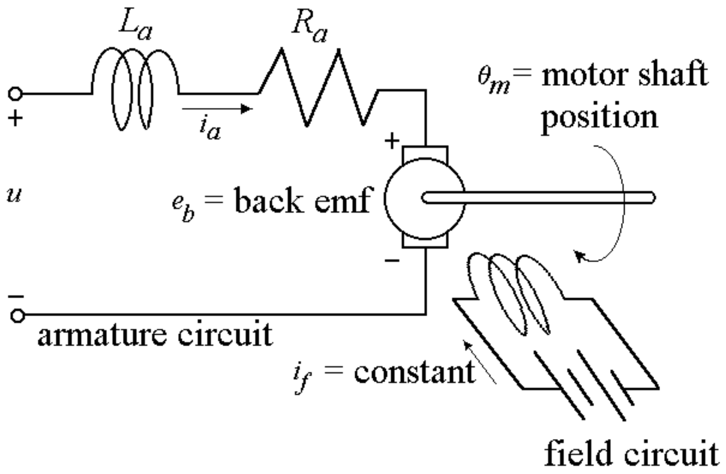


Figure 2.21: Schematic of an armature-controlled DC motor of Example 9.13.

The torque delivered by the motor is

$$T_m = K_m i_a,$$

where K_m is the motor-torque constant, and i_a is the armature current. Let N denote the gear ratio. Then we have

$$\frac{\theta_p}{\theta_m} = \frac{\text{radius of motor gear}}{\text{radius of arm gear}} = \frac{\text{number of teeth of motor gear}}{\text{number of teeth of arm gear}} = \frac{1}{N}.$$

This is because the gears are in contact and thus

$$\theta_p \times \text{radius of arm gear} = \theta_m \times \text{radius of motor gear},$$

and the radii of the gears are proportional to their number of teeth. The work done by the gears must be equal. Let T_p denote the torque applied to the robot arm. Then,

$$T_p\theta_p = T_m\theta_m.$$

Thus, the torque applied to the pendulum is

$$T_p = NT_m = NK_m i_a.$$

We use Newton's second law to write the equation modeling the arm dynamics,

$$I_a \frac{d^2\theta_p}{dt^2} = mgl\sin\theta_p + T_p. \tag{2.80}$$

Substituting into (2.80) the expressions for I_a and T_p and then rearranging yields

$$ml^2 \frac{d^2\theta_p}{dt^2} = mgl\sin\theta_p + NK_m i_a, \tag{2.81}$$

where $g = 9.8$ m/s^2 is the gravitational constant. We next apply Kirchhoff's voltage law to the armature circuit to obtain

$$L_a \frac{di_a}{dt} + Ri_a + e_b = L_a \frac{di_a}{dt} + Ri_a + K_b N \frac{d\theta_p}{dt} = u,$$

where K_b is the back emf constant. We can now construct a third-order state-space model of the one-link robot. For this we choose the following state variables:

$$x_1 = \theta_p, \quad x_2 = \frac{d\theta_p}{dt} = \omega_p, \quad x_3 = i_a.$$

Then, the model in state-space format is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g}{l}\sin x_1 + \frac{NK_m}{ml^2} x_3 \\ -\frac{K_b N}{L_a} x_2 - \frac{R_a}{L_a} x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} u, \tag{2.82}$$

Reasonable parameters of the robot are: $l = 1\,\mathrm{m}$, $m = 1\,\mathrm{kg}$, $N = 10$, $K_m = 0.1\,\mathrm{Nm/A}$, $K_b = 0.1\,\mathrm{Vs/rad}$, $R_a = 1\,\Omega$, $L_a = 100\,\mathrm{mH}$. With the above parameters the robot model takes the form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ 9.8\,\sin x_1 + x_3 \\ -10\,x_2 - 10\,x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} u. \tag{2.83}$$

We assume that the output, y, is

$$y = x_1. \tag{2.84}$$

2.4.6 Double Pendulum

We construct a model of the double pendulum shown in Figure 2.22. This double pendulum can be viewed as a model of a double-link robot manipulator. We assume that the input torques, T_1 and T_2 are applied at each joint, $\omega_i = \dot{\theta}_i$, $i = 1, 2$, are the angular velocities of the pendulum arms, $\Delta\theta = \theta_2 - \theta_1$, and m_i and l_i are the masses and arm lengths, respectively.

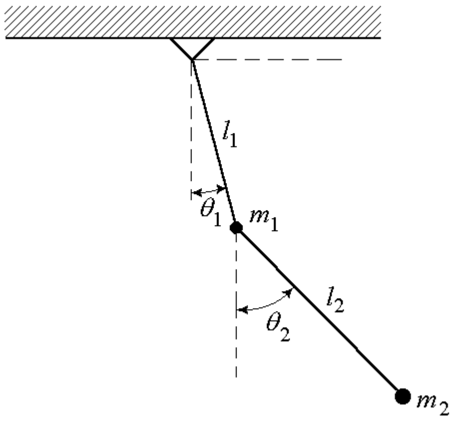


Figure 2.22: Double pendulum.

Table 2.3: Parameter numerical values of double pendulum system shown in Figure 2.22.

$m_1 = m_2$	1 kg
$l_1 = l_2$	0.5 m
g	9.8 m/sec^2

The parameter values are given in Table 2.3. We write the Lagrange equations of motion for the system and represent them in state-space format, where the state variables are defined as, $x_1 = \theta_1$, $x_2 = \dot{\theta}_1$, $x_3 = \theta_2$, and $x_4 = \dot{\theta}_2$.

Lagrange Equations of Motion of the Double Pendulum System

Recall that the Lagrangian L is defined as $L = K - U$, where K is the kinetic energy of the double pendulum system shown in Figure 2.22 and U is the potential energy of the system. The kinetic energy is

$$\begin{aligned} K &= K_{m_1} + K_{m_2} \\ &= \frac{1}{2} m_1 \|\boldsymbol{v}_{m_1}\|^2 + \frac{1}{2} m_2 \|\boldsymbol{v}_{m_2}\|^2, \end{aligned}$$

where $\boldsymbol{v}_{m_i}$, $i = 1, 2$, are the velocities of the point masses. The velocity vectors can be computed from the position of the point masses. The position of the first mass is described by

$$x_1 = l_1 \sin(\theta_1), \quad y_1 = l_1 \cos(\theta_1). \tag{2.85}$$

The position of m_2 is determined by

$$x_2 = l_1 \sin(\theta_1) + l_2 \sin(\theta_2), \quad y_2 = l_1 \cos(\theta_1) + l_2 \cos(\theta_2). \tag{2.86}$$

We next find time derivatives of the expressions in (2.85) and (2.86). The time derivatives of x_1 and y_1 are,

$$\dot{x}_1 = l_1 \dot{\theta}_1 \cos(\theta_1), \quad \dot{y}_1 = -l_1 \dot{\theta}_1 \sin(\theta_1).$$

The time derivatives of x_2 and y_2 are,

$$\dot{x}_2 = l_1 \dot{\theta}_1 \cos(\theta_1) + l_2 \dot{\theta}_2 \cos(\theta_2), \quad \dot{y}_2 = -l_1 \dot{\theta}_1 \sin(\theta_1) - l_2 \dot{\theta}_2 \sin(\theta_2).$$

Using the above expressions, we compute

$$\|\boldsymbol{v}_{m_1}\|^2 = \dot{x_1}^2 + \dot{y_1}^2 = \left(l_1 \dot{\theta}_1\right)^2$$

and

$$\begin{aligned}
\|\boldsymbol{v}_{m_2}\|^2 &= \dot{x_2}^2 + \dot{y_2}^2 \\
&= \left(l_1 \dot{\theta}_1\right)^2 \cos^2(\theta_1) + 2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1) \cos(\theta_1) + \left(l_2 \dot{\theta}_2\right)^2 \cos^2(\theta_2) \\
&\quad + \left(l_1 \dot{\theta}_1\right)^2 \sin^2(\theta_1) + 2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1) \sin(\theta_1) + \left(l_2 \dot{\theta}_2\right)^2 \sin^2(\theta_2) \\
&= \left(l_1 \dot{\theta}_1\right)^2 + \left(l_2 \dot{\theta}_2\right)^2 + 2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_2 - \theta_1),
\end{aligned}$$

where we used the well known trigonometric identity,

$$\cos(\theta_2 - \theta_1) = \cos(\theta_2) \cos(\theta_1) + \sin(\theta_2) \sin(\theta_1).$$

In view of the above, the kinetic energy of the system is

$$\begin{aligned}
K &= K_{m_1} + K_{m_2} \\
&= \frac{1}{2} m_1 \|\boldsymbol{v}_{m_1}\|^2 + \frac{1}{2} m_2 \|\boldsymbol{v}_{m_2}\|^2 \\
&= \frac{1}{2} m_1 \left(l_1 \dot{\theta}_1\right)^2 + \frac{1}{2} m_2 \left(\left(l_1 \dot{\theta}_1\right)^2 + \left(l_2 \dot{\theta}_2\right)^2 + 2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_2 - \theta_1)\right).
\end{aligned}$$

To compute the potential energy, we observe the positive direction of the y-coordinate and the direction of the gravity force are the same. Therefore, the potential energy is

$$\begin{aligned}
U &= -m_1 g y_1 - m_2 g y_2 \\
&= -m_1 g l_1 \cos(\theta_1) - m_2 g \left(l_1 \cos(\theta_1) + l_2 \cos(\theta)\right) \\
&= -(m_1 + m_2) g l_1 \cos(\theta_1) - m_2 g l_2 \cos(\theta_2).
\end{aligned}$$

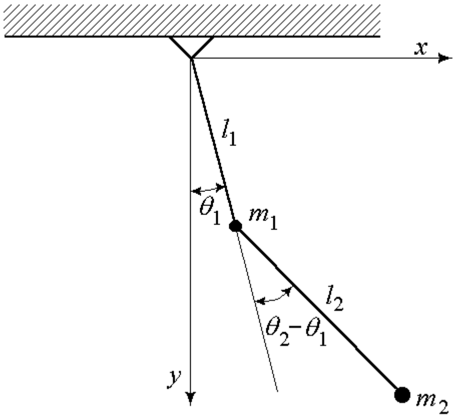


Figure 2.23: Angles in the double pendulum system of Figure 2.22.

Hence, the Lagrangian function of the double pendulum system is

$$\begin{aligned} L &= K - U \\ &= \frac{1}{2}m_1\left(l_1\dot{\theta}_1\right)^2 + \frac{1}{2}m_2\left(\left(l_1\dot{\theta}_1\right)^2 + \left(l_2\dot{\theta}_2\right)^2 + 2l_1l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_2-\theta_1)\right) \\ &\quad +(m_1+m_2)gl_1\cos(\theta_1) + m_2gl_2\cos(\theta_2). \end{aligned}$$

We next derive the Lagrange equations of motion of the system. For this, we need to determine the generalized forces that constitute the right hand sides of the Lagrange equations of motion. Referring to Figure 2.23, we write,

$$\begin{aligned} \delta W &= T_1\delta\theta_1 + T_2\delta(\theta_2-\theta_1) \\ &= (T_1-T_2)\delta\theta_1 + T_2\delta\theta_2, \end{aligned} \tag{2.87}$$

where T_1 and T_2 are the torques applied at each joint. Hence the generalized forces are

$$F_{\theta_1} = T_1 - T_2 \quad \text{and} \quad F_{\theta_2} = T_2. \tag{2.88}$$

To proceed, we need to find partial derivatives of the Lagrangian function. We first evaluate $\partial L/\partial\dot{\theta}_1$ to get

$$\frac{\partial L}{\partial\dot{\theta}_1} = m_1l_1^2\dot{\theta}_1 + m_2l_1^2\dot{\theta}_1 + m_2l_1l_2\dot{\theta}_2\cos(\theta_2-\theta_1). \tag{2.89}$$

Next, we find

$$\frac{d}{dt}\frac{\partial L}{\partial\dot{\theta}_1} = m_1l_1^2\ddot{\theta}_1 + m_2l_1^2\ddot{\theta}_1 + m_2l_1l_2\ddot{\theta}_2\cos(\theta_2-\theta_1) - m_2l_1l_2\dot{\theta}_2(\dot{\theta}_2-\dot{\theta}_1)\sin(\theta_2-\theta_1). \tag{2.90}$$

The partial derivative $\partial L/\partial\theta_1$ is

$$\frac{\partial L}{\partial \theta_1} = m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_2 - \theta_1) - (m_1 + m_2) g l_1 \sin(\theta_1). \tag{2.91}$$

Combining (2.88), (2.90), and (2.91), we obtain the first Lagrange equation of motion,

$$(m_1 + m_2) l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta} \cos(\theta_2 - \theta_1) - m_2 l_1 l_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_1) + (m_1 + m_2) g l_1 \sin(\theta_1) = T_1 - T_2. \tag{2.92}$$

We next compute partial derivatives of L that we use to write the second Lagrange equation of motion. We first find $\partial L/\partial\dot{\theta}_2$,

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \dot{\theta}_2 + m_2 l_2 l_2 \dot{\theta}_1 \cos(\theta_2 - \theta_1). \tag{2.93}$$

The time derivative of the above expression is

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_2 - \theta_1) - m_2 l_1 l_2 (\dot{\theta}_2 - \dot{\theta}_1)\dot{\theta}_1 \sin(\theta_2 - \theta_1). \tag{2.94}$$

The partial derivative $\partial L/\partial\theta_2$ is

$$\frac{\partial L}{\partial \theta_2} = -m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_2 - \theta_1) - m_2 g l_2 \sin(\theta_2). \tag{2.95}$$

Combining (2.88), (2.94), and (2.95), we obtain the second Lagrange equation of motion,

$$m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_2 - \theta_1) + m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) + m_2 g l_2 \sin(\theta_2) = T_2. \tag{2.96}$$

State-Space Model of the Double Pendulum System

Taking into account the definition of the state variables, we obtain

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \ddot{\theta}_1 \\ x_4 \\ \ddot{\theta}_2 \end{bmatrix}. \tag{2.97}$$

Thus, to obtain a state-space model of the double pendulum system, we need to find the acceleration of each link. For this, we represent Lagrange equations (2.92) and (2.96) in matrix form,

$$\begin{bmatrix} (m_1 + m_2) l_1^2 & m_2 l_1 l_2 \cos(\theta_2 - \theta_1) \\ m_2 l_1 l_2 \cos(\theta_2 - \theta_1) & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$
$$= \begin{bmatrix} T_1 - T_2 + m_2 l_1 l_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_1) - (m_1 + m_2) g l_1 \sin(\theta_1) \\ T_2 - m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) - m_2 g l_2 \sin(\theta_2) \end{bmatrix}. \tag{2.98}$$

Note that the square matrix, call it $\boldsymbol{M}$, on the left-hand side of the above equation is invertible because its determinant is

$$\det \boldsymbol{M} = (m_1 + m_2)m_2 l_1^2 l_2^2 - m_2^2 l_1^2 l_2^2 \cos^2(\theta_2 - \theta_1) > 0.$$

Solving (2.98) for $\ddot{\theta}_1$ and $\ddot{\theta}_2$, we obtain

$$\ddot{\theta}_1 = \frac{\begin{array}{l} m_2 l_2^2 \left(T_1 - T_2 + m_2 l_1 l_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_1) - (m_1 + m_2) g l_1 \sin(\theta_1)\right) \\ -m_2 l_1 l_2 \cos(\theta_2 - \theta_1) \left(T_2 - m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) - m_2 g l_2 \sin(\theta_2)\right) \end{array}}{(m_1 + m_2)m_2 l_1^2 l_2^2 - m_2^2 l_1^2 l_2^2 \cos^2(\theta_2 - \theta_1)} \tag{2.99}$$

and

$$\ddot{\theta}_2 = \frac{\begin{array}{l} -m_2 l_1 l_2 \cos(\theta_2 - \theta_1) \left(T_1 - T_2 + m_2 l_1 l_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_1) - (m_1 + m_2) g l_1 \sin(\theta_1)\right) \\ +(m_1 + m_2) l_1^2 \left(T_2 - m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) - m_2 g l_2 \sin(\theta_2)\right) \end{array}}{(m_1 + m_2)m_2 l_1^2 l_2^2 - m_2^2 l_1^2 l_2^2 \cos^2(\theta_2 - \theta_1)}. \tag{2.100}$$

The input vector is defined as

$$\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\top = \begin{bmatrix} T_1 & T_2 \end{bmatrix}^\top.$$

Taking into account the above definition of the input, the definition of the state variables given by (2.97) and relations (2.99) and (2.100), we obtain the following state-space model of the double pendulum system

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ &= \begin{bmatrix} x_2 \\ \dfrac{\begin{array}{l} m_2 l_2^2 \left(u_1 - u_2 + m_2 l_1 l_2 x_4^2 \sin(x_3 - x_1) - (m_1 + m_2) g l_1 \sin(x_1)\right) \\ -m_2 l_1 l_2 \cos(x_3 - x_1) \left(u_2 - m_2 l_1 l_2 x_2^2 \sin(x_3 - x_1) - m_2 g l_2 \sin(x_3)\right) \end{array}}{(m_1+m_2) m_2 l_1^2 l_2^2 - m_2^2 l_1^2 l_2^2 \cos^2(x_3 - x_1)} \\ x_4 \\ \dfrac{\begin{array}{l} -m_2 l_1 l_2 \cos(x_3 - x_1) \left(u_1 - u_2 + m_2 l_1 l_2 x_4^2 \sin(x_3 - x_1) - (m_1 + m_2) g l_1 \sin(x_1)\right) \\ +(m_1 + m_2) l_1^2 \left(u_2 - m_2 l_1 l_2 x_2^2 \sin(x_3 - x_1) - m_2 g l_2 \sin(x_3)\right) \end{array}}{(m_1+m_2) m_2 l_1^2 l_2^2 - m_2^2 l_1^2 l_2^2 \cos^2(x_3 - x_1)} \end{bmatrix}. \end{aligned}$$

2.4.7 Population Dynamics

We first model dynamics of a single population. Then, we use the obtained model to investigate two interacting populations.

Let $x = \phi(t)$ be the population of the given species at a time t. The simplest model of the population dynamics of the given species is the one resulting from the hypothesis that the net rate of change of x is proportional to the current value of x. Let r be the rate of growth of the population. Then, the simple model of the population dynamics is

$$\dot{x} = rx. \tag{2.101}$$

Solving (2.101) subject to the initial condition, $x(0) = x_0$, yields

$$x = e^{rt} x_0. \tag{2.102}$$

In the above linear model, the growth rate, r, is constant. It does not account for the limitation of space, food supply, etc. Therefore, we modify the unrestricted growth rate r to take into account the fact that the environment of the population can only support a certain number, say K, of the species. The constant K is called the *carrying capacity* of the environment or the *saturation level*. Specifically, if $x > K$, then because there is not enough food or space to support x, more species will die than are born, which results in a negative growth rate. On the other hand, if $x < K$, then there is extra food and space and the population growth is positive. One function that satisfies the above requirements has the form

$$r\left(1 - \frac{x}{K}\right).$$

Using the above model of the growth rate, we obtain the following model of the population growth,

$$\dot{x} = r\left(1 - \frac{x}{K}\right)x. \tag{2.103}$$

Let $c = r/K$. Then, we represent (2.103) as

$$\boxed{\dot{x} = (r - cx)\,x} \tag{2.104}$$

Equation (2.104) is known as the *Verhulst equation* or the *logistic equation*. According to Boyce and DiPrima [40, page 60], Verhulst, a Belgian mathematician, introduced equation (2.104) in 1838 to model human population growth. For reasons somewhat unclear, Verhulst referred to it as logistic growth, hence (2.104) is often called the logistic equation.

We use the logistic equation to analyze interacting species. We first model two interacting species that do not prey upon each other, but compete for the same resources, for example for a common food supply. Let x_1 and x_2 be the populations, at time t, of the two competing species. We assume that the population of each of the species in the absence of the other is governed by a logistic equation. Thus, a model of two non-interacting species has the form

$$\left.\begin{aligned} \dot{x}_1 &= (r_1 - c_1 x_1)\, x_1 \\ \dot{x}_2 &= (r_2 - c_2 x_2)\, x_2, \end{aligned}\right\} \tag{2.105}$$

where r_1 and r_2 are the unrestricted growth rates of the two populations, while r_1/c_1 and r_2/c_2 are their saturation levels. When both species are present, each will affect the growth rate of the other. Since x_1 and x_2 use the same resources, the growth rate of x_1 will be lowered by an amount proportional to the size of x_2. We modify the growth equation for species x_1 as

$$\dot{x}_1 = (r_1 - c_1 x_1 - a_1 x_2)\, x_1, \tag{2.106}$$

where a_1 is a measure by which species x_2 lower the growth rate of x_1. Similarly, the growth rate of x_2 is reduced by $a_2 x_1$, where a_2 is a measure of the degree by which species x_1 lower the growth rate of x_2. Therefore, the growth equation for species x_2 becomes

$$\dot{x}_2 = (r_2 - c_2 x_2 - a_2 x_1)\, x_2. \tag{2.107}$$

In summary, equations (2.106) and (2.107) model two species that interact by competing for a common resource like food or other supply.

We now analyze the situation in which one species, the predator, preys upon the other species, the prey, and there is no competition between the predator and prey for a common resource. Let x_1 and x_2 denote the populations of the prey and predator, respectively. We make the following modeling assumptions:

1. When there is no predator, that is, $x_2 = 0$, the prey enjoys an exponential growth governed by
$$\dot{x}_1 = r_1 x_1, \quad r_1 > 0. \tag{2.108}$$

2. When there is no prey, that is, $x_1 = 0$, the predator perishes. We model this scenario using the equation
$$\dot{x}_2 = -r_2 x_2, \quad r_2 > 0. \tag{2.109}$$

3. The larger the population of the predator, the more prey is eaten. The larger the population of the prey, the easier hunting is for the predator. We model the growth rate decrease of the prey by including in (2.108) the term $-d x_1 x_2$, $d > 0$. The growth rate increase of the predator is modeled by adding the term $e x_1 x_2$ in (2.109), where $e > 0$.

The resulting model of the predator-prey relationship is

$$\boxed{\begin{array}{rcl} \dot{x}_1 & = & (r_1 - dx_2)\, x_1 \\ \dot{x}_2 & = & (-r_2 + ex_1)\, x_2 \end{array}} \tag{2.110}$$

Equations (2.110) are known as the *Lotka-Volterra equations*. They were published by Lotka in 1925 and by Volterra in 1926.

2.4.8 The Hypothalamic-Pituitary-Adrenal Axis Model

The hypothalamic-pituitary-adrenal (HPA) axis is the body's primary stress management system. The role of the HPA axis is to maintain bodily functions in the presence of physical and mental stressors. The (HPA) axis is a part of the endocrine system. The endocrine system as well as its subsystem, the HPA axis, uses hormones to communicate between the regions of the body. The regulation of hormones maintains homeostasis—the process by which bodily functions are maintained at a constant level. This leads to a definition of stress as a state of disharmony in which the homeostasis of the organism is threatened.

Another approach to define stress was proposed by McEwen [237] in 2002. To define stress McEwen introduced a notion of allostasis—the process by which the body functions change in response to surrounding stimuli. The term allostasis is the opposite to the notion of homeostasis. An example of allostais is the fight-or-flight response in which the sympathetic nervous system as well as the HPA axis are involved. Dysregulation of the HPA axis is associated with a number of neuroimmune disorders such as chronic fatigue syndrome (CFS), depression, Gulf War illness (GWI), or posttraumatic stress disorder (PTSD), among other stress related diseases [29]. At present, it is not clear what causes dysregulation of the HPA axis.

The HPA model presented here was developed by Gupta et al. [122] and modified by Ben-Zvi et al. [29]. A simplified schematic diagram of the HPA is shown in Figure 2.24.

The HPA axis is responsible for a rapid response to stress stimuli. An activation of the hypothalamus by a stressor causes the release of the corticotropin releasing hormone (CRH). The hypothalamus is the control center of most of the body's hormonal systems. Upon reaching the pituitary gland, the CRH hormone induces the release of the adrenocorticotropic hormone (ACTH) by the pituitary into the circulation that reaches the adrenal glands that are located on top of the kidneys. The ACTH stimulates the secretion of cortisol by the adrenal glands. The release of cortisol initiates metabolic effects to fight the harmful effects of stress through negative feedback to the hypothalamus and pituitary—see Figure 2.24.

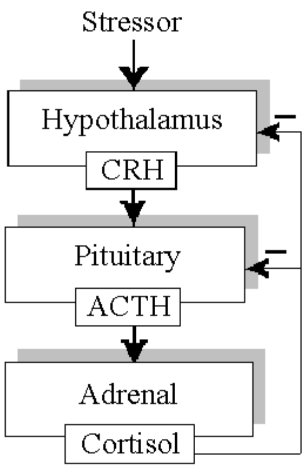


Figure 2.24: A simplified schematic diagram of the HPA axis.

Table 2.4: Description of variables in the HPA model.

Variable	Description
x_1	CRH concentration
x_2	ACTH concentration
x_3	Free GR concentration
x_4	Cortisol concentration
d	Unknown input modeling stress action
u	Control variable modeling treatment action

Once the state of stress subsides, the concentration of ACTH and cortisol decreases. In their model, Gupta et al. [122] also include the dynamics of the glucocorticoid receptor (GR) to model more accurately the HPA axis dynamics.

We use the variables described in Table 2.4 in the HPA axis modeling equations.

We present the HPA model following the development of Gupta et al. [122]. In this model, stress, modeled by the signal d, stimulates the hypothalamus to secrete the CRH denoted x_1. The equation modeling the hypothalamus response to stress is

$$\dot{x}_1 = \left(1 - \frac{x_4}{k_{i1}}\right)(k_c + d) - k_{cd}x_1, \tag{2.111}$$

where the terem $-k_{cd}x_1$ models a constant degradation rate of the CRH in the blood. The term $\left(1-\frac{x_4}{k_{i1}}\right)(k_c+d)$ models a circadian production and stress being reduced by the inhibition term $\left(1-\frac{x_4}{k_{i1}}\right)$. For small values of x_4/k_{i1}, we may write

$$\left(1-\frac{x_4}{k_{i1}}\right)(k_c+d) \approx \frac{k_c+d}{1+\frac{x_4}{k_{i1}}}.$$

Using the above approximation, we obtain the following equation modeling the hypothalamus dynamics,

$$\dot{x}_1 = \frac{k_c+d}{1+\frac{x_4}{k_{i1}}} - k_{cd}x_1. \tag{2.112}$$

The adrenocorticotropic hormone (ACTH) produced by the pituitary gland is modeled as

$$\dot{x}_2 = \frac{x_1}{1+\frac{x_1 x_4}{k_{i2}}} - k_{ad}x_2. \tag{2.113}$$

The dynamics of the glucocorticoid receptor (GR) is modeled by Gupta et al. [122] as

$$\dot{x}_3 = \frac{(x_1x_4)^2}{k+(x_3x_4)^2} - k_{rd}x_3. \tag{2.114}$$

Finally, the production of the cortisol by the adrenal glands is modeled as

$$\dot{x}_4 = x_2 - x_4.$$

Ben-Zvi et al. [29] propose to augment the above equation with the control term u modeling the control action,

$$\dot{x}_4 = x_2 - x_4 + u. \tag{2.115}$$

Combining equations (2.112), (2.113), (2.114), and (2.115), we obtain a fourth-order model of the HPA axis,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{1+\frac{x_4}{k_{i1}}} - k_{cd}x_1 \\ \frac{x_1}{1+\frac{x_1x_4}{k_{i2}}} - k_{ad}x_2 \\ \frac{(x_1x_4)^2}{k+(x_3x_4)^2} - k_{rd}x_3 \\ x_2 - x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} \frac{1}{1+\frac{x_4}{k_{i1}}} \\ 0 \\ 0 \\ 0 \end{bmatrix} d. \tag{2.116}$$

The normalized parameter values listed in Ben-Zvi et al. [29] are given in Table 2.5.

Table 2.5: Parameter values in the HPA model

Parameter	Description	Value
k_{i1}	Inhibition constant for CRH synthesis	0.100
k_{cd}	CRH degradation constant	1.000
k_{i2}	Inhibition constant for ACTH synthesis	0.100
k_{ad}	ACTH degradation constant	10.000
k_c	circadian production	1
k_{cr}	GR synthesis constant	0.050
k_{rd}	GR degradation constant	0.900
k	Inhibition constant for GR synthesis	0.001

2.5 Notes

For more examples of dynamical system models from mechanical and electrical engineering, the reader is referred to Truxal [329], Banks [21], Raven [280], Mohler [244], and D'Azzo and Houpis [73]. Wertz [349] has many excellent real-life models from space science. A well-written paper on vehicle modeling and control for broad audience is by Ackermann [2]. In addition to references [354] and [350], vehicle and tire dynamics modeling are investigated in detail by Nalecz [248], Nalecz and Bindemann [250, 249], and Smith and Starkey [302, 303]. For more on stepper motors, the reader is referred to Acarnley [1], Kenjo [176], Kenjo and Nagamori [177], Del Toro [78, Chapter 9], Krause and Wasynczuk [192, Chapter 8], and Zribi and Chiasson [376]. For further analysis of the equations modeling competing species and the predator-prey equations, we recommend Boyce and DiPrima [40, Sections 9.4 and 9.5]. Our derivation of the logistic equation follows that of Sandefur [293]. The logistic equation as well as equations modeling interacting species can be used to mathematical modeling of tumor dynamics and interactions between tumor and immune system. A survey of models for tumor-immune system dynamics can be found in the well-edited book by Adam and Bellomo [4].

A number of mathematical models of the hypothalamic-pituitary-adrenal (HPA) axis were proposed in the last six decades. Building on the previous models, Bingzhen, Zhenye, and Lainsong [34] proposed a third-order dynamical model and tested it on clinical data. This model was later improved by Liu et al. [219]. Kyrylov, Severyanova and Vieira [201] modified the model of Liu et al. [219] to include additional properties of the HPA axis. Conrad et al. [64] proposed and analyzed a second-order non-linear mathematical model of the HPA axis. Lenbury and Pornsawad [209] used delay-differential equations in their HPA axis model to account for the delays associated with the action of the glands in response to the stimulating hormones. Gupta et al. [122], on the other hand, proposed a fourth-order non-linear state-space model of the HPA axis in which the effect of stress on the HPA axis dynamical behavior is modeled. Ben-Zvi, Vernon, and Broderick [29] modified the model of Gupta et al. [122] by adding the control input representing the treatment. The HPA model proposed by Ben-Zvi et al. [29] can be viewed as a dynamical system with unknown input. Such systems are a subject of Chapter 12.

2.6 Exercises

Exercise 2.1 Prove the law of conservation of energy as stated in the following theorem:

Theorem 2.1 *Given conservative vector field $\boldsymbol{F}$, that is, $\boldsymbol{F} = -\nabla U$, where U is the potential energy. Assume that a particle of mass m, satisfying Newton's law, moves on a differentiable curve $\boldsymbol{C}$. Then, the sum of the potential energy and kinetic energy is constant.*

Hint Denote by $\boldsymbol{C}(t)$ a differentiable curve along which a particle of mass m moves. Hence

$$\boldsymbol{F}(\boldsymbol{C}(t)) = m\ddot{\boldsymbol{C}}(t).$$

Then, show that

$$U(\boldsymbol{C}(t)) + \frac{1}{2}m\left\|\dot{\boldsymbol{C}}(t)\right\|^2 = \text{constant}.$$

It is enough to differentiate $U(\boldsymbol{C}(t)) + \frac{1}{2}m\left\|\dot{\boldsymbol{C}}(t)\right\|^2$ with respect to time and show that the time derivative is zero, hence the sum of the energies is constant.

Exercise 2.2 Show that in a conservative vector field the work done by the field on a particle that moves between two points depends only on these points and not on the path followed.
Hint Note the work done against the conservative force field $\boldsymbol{F}$ along a curve $\boldsymbol{C}$ between points $\boldsymbol{C}(t_1)$ and $\boldsymbol{C}(t_2)$ is

$$\int_{t_1}^{t_2} \boldsymbol{F}(\boldsymbol{C}(t))^\top \dot{\boldsymbol{C}}(t)dt.$$

By assumption $\boldsymbol{F}$ is a conservative vector field. Hence,

$$\boldsymbol{F} = -\nabla U. \tag{2.117}$$

Combining the above yields

$$\int_{t_1}^{t_2} \boldsymbol{F}(\boldsymbol{C}(t))^\top \dot{\boldsymbol{C}}(t)dt = -\int_{t_1}^{t_2} \nabla U(\boldsymbol{C}(t))^\top \dot{\boldsymbol{C}}(t)dt.$$

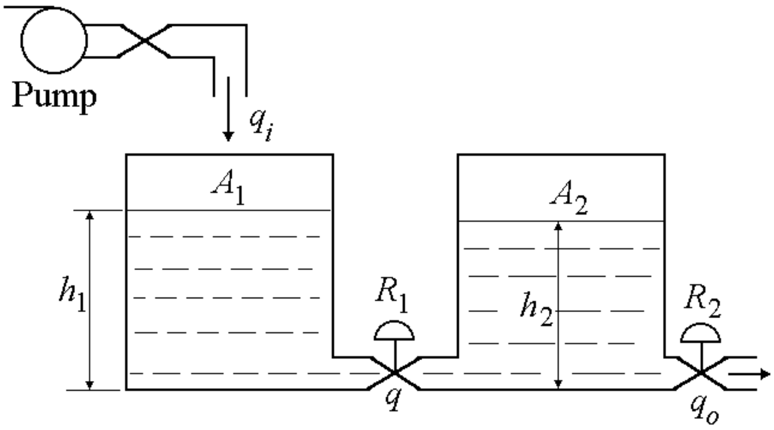


Figure 2.25: A two-tank pumped storage system for Exercise 2.3.

Use the chain rule to represent the expression inside the integral on the right hand side as

$$\nabla U(\boldsymbol{C}(t))^\top \dot{\boldsymbol{C}}(t) = \frac{dU(\boldsymbol{C}(t))}{dt}.$$

Exercise 2.3 The goal of this exercise is to develop a model of a two-tank pumped storage system. Figure 2.25 illustrates such a system. A pumped storage system can be used to supply water to a turbine producing electric power. The two storage tanks are connected together through a valve that may be modeled as a linear resistance R_1. The resistance of the turbine supply pump is modeled by a linear resistance R_2. Suppose that we wish to design a controller that would control the inflow rate q_i in order to maintain the output flow rate q_o at some desired value q_d. The first stage of our project is to develop a design model of the pumped storage system. We assume that linearized flow rates $q(t)$ and $q_o(t)$ are proportional to the corresponding water heights, that is,

$$\begin{aligned} q(t) &= \frac{h_1(t) - h_2(t)}{R_1}, \\ q_o(t) &= \frac{h_2(t)}{R_2}. \end{aligned}$$

Let A_1 and A_2 represent the surface areas in the two tanks. Then, the principle of conser-

Table 2.6: Numerical data for the two-tank pumped storage system model for Exercise 2.3.

Tank 1 diameter	2 m
Tank 2 diameter	3 m
R_1	$10\,\text{sec/m}^2$
R_2	$20\,\text{sec/m}^2$

vation of mass for an incompressible fluid yields

$$\begin{aligned} q_i(t) - q(t) &= A_1 \dot{h}_1(t), \\ q(t) - q_o(t) &= A_2 \dot{h}_2(t). \end{aligned}$$

Numerical data of a two-tank pumped storage system is given in Table 2.6. Find the transfer function $\mathcal{L}(q_o(t))/\mathcal{L}(q_i(t))$.

Exercise 2.4 The objective of this exercise is to derive a model of an automotive suspension. The role of an automotive suspension is to support the vehicle body on the axles. A simple linear model of the suspension assembly associated with a single wheel, called quarter-car model, is shown in Figure 2.26. The vehicle body is represented by the "sprung" mass, m_s, while the tire and axle by the "unsprung" mass, m_u. Model the tire as a spring with spring coefficient k_t, and assume its damping to be negligible. The suspension consists of the spring, shock absorber, and the variable force element. The role of the variable force element is to generate a force that compensates for the uneven road which is the source of the system disturbances. Assume that the variable force element F_a can instantaneously provide any desired force. The above described model of an automotive suspension is referred to as a linear active suspension. Write down equations modeling the dynamics of the two degree of freedom system depicted in Figure 2.26 by applying Newton's law to each mass of the system, and taking into account the positive direction of the z coordinates. Define the state variables to be:

$$\begin{aligned} x_1 &= z_s - z_u = \text{suspension stroke}, \\ x_2 &= \dot{z}_s = \text{sprung mass velocity}, \\ x_3 &= z_u - z_r = \text{tire deflection}, \\ x_4 &= \dot{z}_u = \text{unsprung mass velocity}. \end{aligned}$$

Let $u = F_a$ be the system input, and let $d(t) = \dot{z}_r(t)$ be the disturbance input due to the road roughness. Represent the obtained linear model of active suspension in state-space

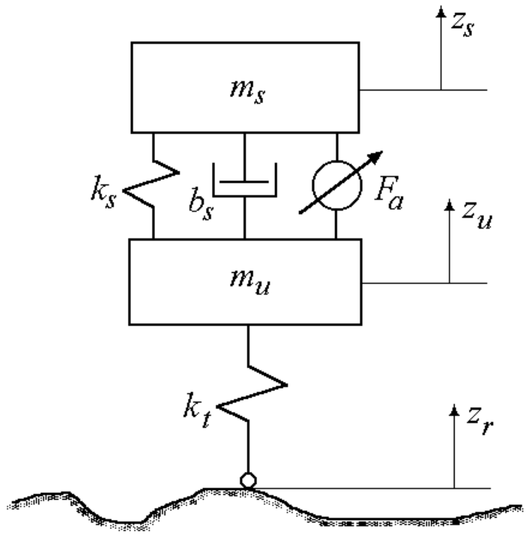


Figure 2.26: Linear quarter-car model for Exercise 2.4.

format. Use data of a commercial vehicle given in Table 2.7. The data comes from Sunwoo and Cheok [314]. Substitute the vehicle data into the modeling equations. A design goal of an automotive engineer may be to design a control law to isolate the sprung mass from road disturbances. In other words, he may wish to design a control law such that the vertical acceleration of the sprung mass $\ddot{z}_s(t) = \dot{x}_2$ is as close to 0 as possible in spite of road disturbances. Assume that we can measure the vertical acceleration of the sprung mass. Write the output equation.

Exercise 2.5 The objective of this exercise is to develop a model of a single axis machine

Table 2.7: Parameters of the vehicle suspension of Exercise 2.4.

m_s	355 kg
m_u	46 kg
k_s	14384 N/m
b_s	1860 N/m/sec
k_t	300000 N/m

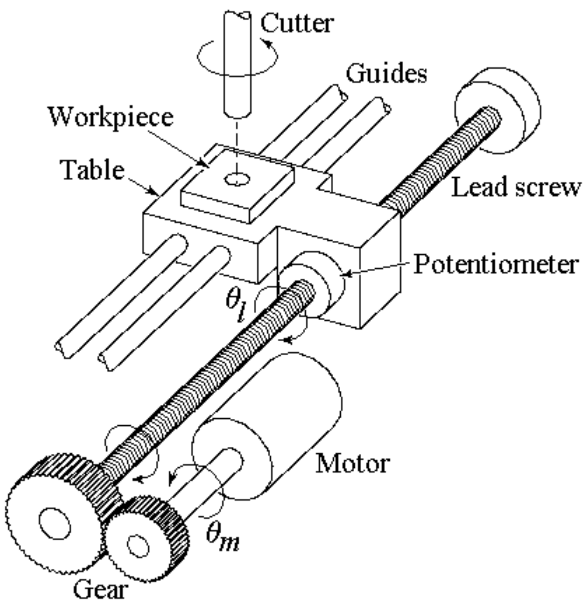


Figure 2.27: A schematic of a machine tool drive system for Exercise 2.5.

tool drive in which a motor drives a lead screw through a gear. Figure 2.27 illustrates such a system. This model is adapted from Driels [87, pages 613–619]. The system operates as follows. The lead screw moves the table with a workpiece on it under the cutter to perform the desired operation. The table is positioned by a control system. We wish to place the table in a desirable position as rapidly and accurately as possible. This means, that while modeling the machine tool drive system we should, in particular, take into account the flexibility of the lead screw. The length of the lead screw implies that its rotation at the gear box will not be the same as its rotation at the table. Thus, the angular twist along the length of the lead screw cannot be neglected. Model the lead screw flexibility using a torsional spring as illustrated in Figure 2.28. The system input is the motor torque τ_m, and the system output is the rotation of the end of the lead screw labeled by θ_l. Assume that the damping torque, due to viscous friction, acting on the motor is proportional to its angular velocity $\dot{\theta}_m = \omega_m$ with the coefficient of viscous friction denoted c_m. Denote the motor moment of inertia by I_m, and by τ_1 the load torque on the motor gear due to the rest of the gear train. Write the equation modeling the dynamics of the motor shaft.

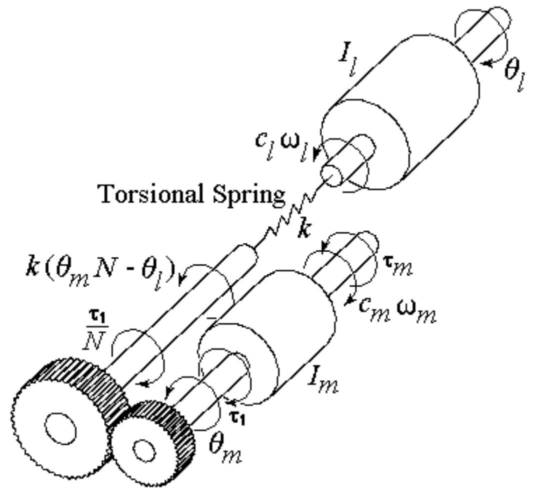


Figure 2.28: A graphical model of the machine tool drive system for Exercise 2.5.

Table 2.8: Data for the machine tool drive system model for Exercise 2.5.

I_m	0.0001 Nm·s^2/rad
I_l	0.001 Nm·s^2/rad
$c_m = c_l$	0.01 Nm·s/rad
k	1 Nm/rad
N	1/5

Table 2.9: Vehicle data for Exercise 2.6.

$l_1 = l_2$	1 m
v_x	10 m/sec
m	1,000 kg
I_z	1,000 kg·m^2
$C_f = C_r$	10,000 N/rad

Let τ_2 be the torque transmitted to the load shaft, and let $N = N_1/N_2$ be the gear ratio, where N_1 denotes the number of teeth of the first gear, and respectively N_2 is the number of teeth of the second gear. The work done by the first gear is equal to that of the second. Model the load shaft flexibility with a torsional spring. The torsional spring coefficient is k. Assume that the moment of inertia of the lead screw is negligible compared to that of the load. Denote the moment of inertia of the load by I_l. The torque due to viscous friction of the load is $c_l\omega_l$, where c_l is the coefficient of the viscous friction of the load. The angular displacement of the end of the lead screw is θ_l. Write the equation modeling the dynamics of the load. Define the state variables to be

$$x_1 = \theta_m, \quad x_2 = \dot{\theta}, \quad x_3 = \theta_l, \quad x_4 = \dot{\theta}_l.$$

Represent the modeling equations in state space format. Typical numerical data for a machine tool drive system are shown in Table 2.8.

Exercise 2.6 Consider a bicycle model of a front wheel steering vehicle shown in Figure 2.29. Construct a two-dimensional state-space model of the vehicle, where $x_1 = \dot{y}$ and $x_2 = \dot{\theta}$. The input $u = \delta_f$ and the output $y = x_2$. In your derivation assume small angles. Vehicle data are given in Table 2.9.

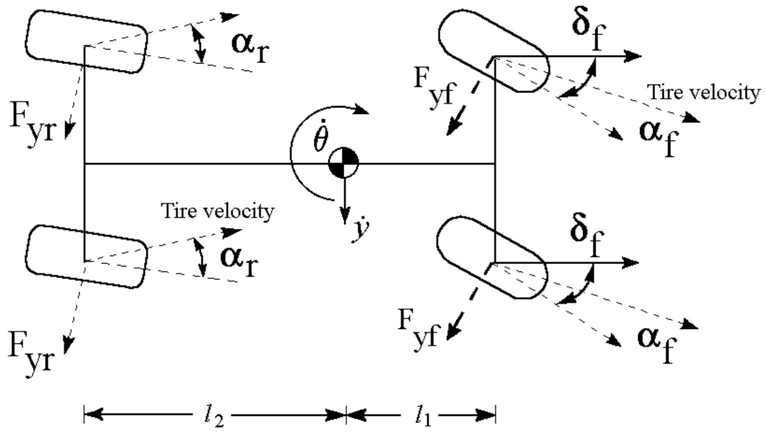


Figure 2.29: Bicycle model of a vehicle for Exercise 2.6.

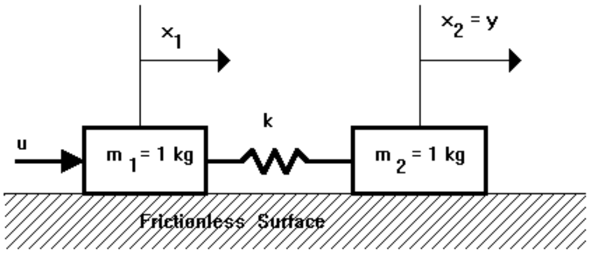


Figure 2.30: The two-mass-spring system for Exercise 2.7.

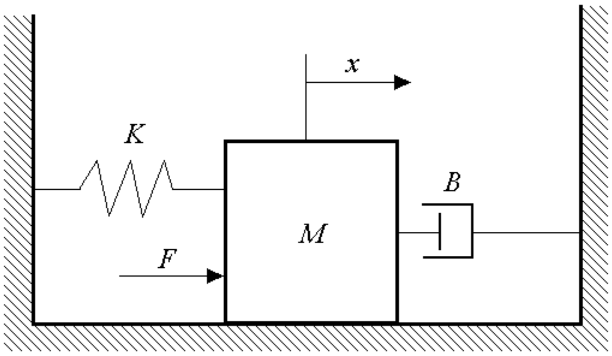


Figure 2.31: Mechanical system for Exercise 2.8.

Exercise 2.7 Consider the two-mass-spring system depicted in Figure 2.30.

(i) Form the Lagrangian of the system.

(ii) Write the Lagrange equations of motion for the system.

(iii) Represent the obtained Lagrange equations of motion in state-space format.

Exercise 2.8 Consider the mechanical system shown in Figure 2.31.

(i) Form the Lagrangian for the system.

(ii) Write the Lagrange equation of motion for the system.

(iii) Represent the system model in state space format.

Exercise 2.9 Consider a dynamical system that consists of a cart with an inverted pendulum (point mass on a massless shaft) attached to it as depicted in Figure 2.32.

(i) Write the Lagrange equations of motion for the system,

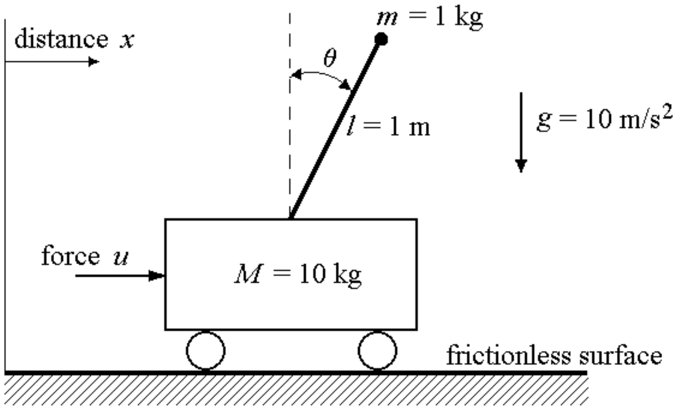


Figure 2.32: An inverted pendulum moving on a cart for Exercise 2.9.

(ii) Represent the obtained model in state-space format using the state variables

$$x_1 = x, \quad x_2 = \dot{x}, \quad x_3 = \theta, \quad x_4 = \dot{\theta}.$$

Exercise 2.10 Consider a schematic of a field-controlled DC motor shown in Figure 2.33. Let $x_1 = \theta$, $x_2 = \dot{\theta}$, $x_3 = i_f$, and $y = x_1$. Treat the field inductance L_f as a parasitic element and denote it μ.

(i) Form the actual plant model P_μ in state space format;

(ii) Find the reduced-order design model P_0.

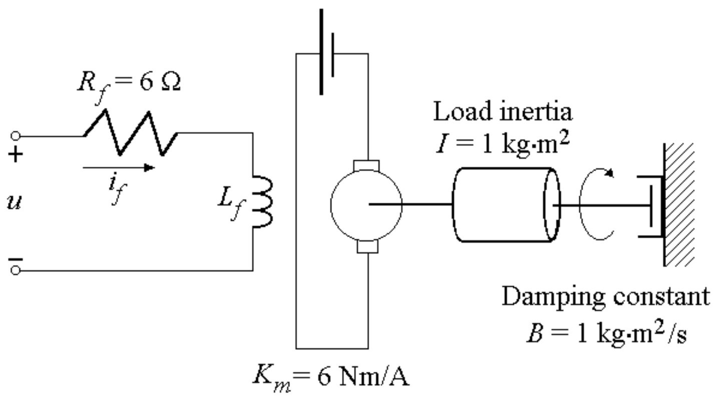


Figure 2.33: A diagram of a field-controlled DC motor for Exercise 2.10.

Chapter 3

Analysis of Modeling Equations

> Folk wisdom says that if you leave a cold alone, it lasts seven days; if you treat it, it will go away in a week. Nonetheless, people have sworn by such "cures" as this: hang your hat on the bedpost, drink gin until you see two hats, then pull the covers up until the siege is over. [281, page 52]

Most of dynamical systems analyzed analyzed by us in this book are modeled by ordinary differential equations. A main use of a mathematical model is to predict the system transient behavior. Unlike linear systems, where closed-form solutions can be written in terms of the system's eigenvalues and eigenvectors, finding analytical, exact, solutions to nonlinear differential equations can be very difficult or impossible. However, we can approximate the solutions of differential equations with difference equations whose solutions can be obtained easily using a computer. In this chapter, after discussing methods for solving state equations modeling linear systems, we present a number of methods for solving a more general class of differential equations. The first class of methods allows us to graphically determine solutions to second-order differential equations. Then, we discuss numerical techniques for solving differential equations.

Linear systems can be modeled using convolutions and in the case of lumped systems, we can use state-space equations. Analytical computation of convolutions can be quite involved. An efficient method of obtaining a solution to a model of a lumped system is to convert the transfer function into a state-space model and then solve the state equation. We begin with a time-invariant case and then proceed with the time-varying case.

3.1 State Equation Solution of Uncontrolled System

Given a dynamical system's time-invariant model of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) \tag{3.1}$$

subject to an initial condition

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{3.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. The solution to (3.1) subject to (3.2) is

$$\boxed{\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}_0} \tag{3.3}$$

The above is a generalization of the situation when $\boldsymbol{A}$ is a scalar. The exponential matrix $e^{\boldsymbol{A}t}$ is defined as

$$e^{\boldsymbol{A}t} = \boldsymbol{I}_n + t\boldsymbol{A} + \frac{t^2}{2!}\boldsymbol{A}^2 + \frac{t^3}{3!}\boldsymbol{A}^3 + \cdots$$

To verify that (3.3) solves (3.1) subject to (3.2), we use the fact that

$$\frac{d}{dt}e^{\boldsymbol{A}t} = \boldsymbol{A}e^{\boldsymbol{A}t} = e^{\boldsymbol{A}t}\boldsymbol{A}.$$

Suppose that the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ is replaced with a more general condition

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0,$$

where t_0 is a given initial time instant. Then, the solution to $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t)$ subject to $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ is

$$\boxed{\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}_0}$$

The matrix $e^{\boldsymbol{A}(t-t_0)}$ is often written as

$$e^{\boldsymbol{A}(t-t_0)} = \boldsymbol{\Phi}(t, t_0)$$

and is called the *state transition matrix*, because it relates the state at any instant of time t_0 to the state at any other time t.

3.2 Computing State Transition Matrix of LTI Systems

For an overview of methods to calculate the state transition matrix for a linear, time-invariant (LTI) system, we refer the reader to the classic paper on the subject by Moler and Van Loan [245], where the numerical issues related to the computation of the matrix exponential are discussed. In what follows, we present a few simple methods to compute the matrix exponential.

3.2.1 The Inverse Laplace Transform Method

In the case when the matrix $\boldsymbol{A}$ is of low dimensions, we can use the formula that results from applying the Laplace transform to the time-invariant system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0$. The Laplace transform of the above matrix differential equation is

$$s\boldsymbol{X}(s) - \boldsymbol{x}_0 = \boldsymbol{A}\boldsymbol{X}(s).$$

Hence

$$\boldsymbol{X}(s) = (s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{x}_0,$$

and the inverse transform yields

$$\boxed{e^{\boldsymbol{A}t} = \mathcal{L}^{-1}\left\{(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\right\}} \tag{3.4}$$

where $\mathcal{L}^{-1}$ denotes the inverse Laplace transform operator.

Example 3.1 We use the inverse Laplace method to compute $e^{\boldsymbol{A}t}$ for

$$\boldsymbol{A} = \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix}.$$

We have

$$\begin{aligned} e^{\boldsymbol{A}t} &= \mathcal{L}^{-1}\left([s\boldsymbol{I}_2 - \boldsymbol{A}]^{-1}\right) \\ &= \mathcal{L}^{-1}\left(\begin{bmatrix} s & -3 \\ -2 & s-1 \end{bmatrix}^{-1}\right) \\ &= \mathcal{L}^{-1}\left(\frac{1}{(s-3)(s+2)}\begin{bmatrix} s-1 & 3 \\ 2 & s \end{bmatrix}\right) \\ &= \begin{bmatrix} 0.4e^{3t} + 0.6e^{-2t} & 0.6\left(e^{3t} - e^{-2t}\right) \\ 0.4\left(e^{3t} - e^{-2t}\right) & 0.6e^{3t} + 0.4e^{-2t} \end{bmatrix}. \end{aligned} \tag{3.5}$$

3.2.2 Diagonalization Method

A convenient way to calculate $e^{\boldsymbol{A}t}$ is when the matrix $\boldsymbol{A}$ has distinct eigenvalues. In such a case the n-by-n matrix $\boldsymbol{A}$ has n linearly independent eigenvectors as per the following theorem.

Theorem 3.1 *Suppose the characteristic equation,*

$$\begin{aligned}\Delta(\lambda) &= \det[\lambda \boldsymbol{I}_n - \boldsymbol{A}] \\ &= \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 \\ &= 0,\end{aligned}$$

has n distinct roots, $\lambda_1, \lambda_2, \ldots, \lambda_n$. Then, there exist n linearly independent vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ such that

$$\boldsymbol{A}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i, \quad i = 1, 2, \ldots, n.$$

Proof See Exercise 3.2.

□

Let

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix}.$$

The matrix $\boldsymbol{T}$ is invertible because the eigenvectors $\boldsymbol{v}_i$, $i = 1, 2, \ldots, n$ are linearly independent. Then, because $\boldsymbol{T}^{-1}\boldsymbol{T} = \boldsymbol{I}_n$, we have

$$\begin{aligned}\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} &= \boldsymbol{T}^{-1}\boldsymbol{A}\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix} \\ &= \boldsymbol{T}^{-1}\begin{bmatrix} \boldsymbol{A}\boldsymbol{v}_1 & \boldsymbol{A}\boldsymbol{v}_2 & \cdots & \boldsymbol{A}\boldsymbol{v}_n \end{bmatrix} \\ &= \boldsymbol{T}^{-1}\begin{bmatrix} \lambda_1\boldsymbol{v}_1 & \lambda_2\boldsymbol{v}_2 & \cdots & \lambda_n\boldsymbol{v}_n \end{bmatrix} \\ &= \boldsymbol{T}^{-1}\boldsymbol{T}\begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_n \end{bmatrix} \\ &= \boldsymbol{\Lambda}. \end{aligned} \tag{3.6}$$

We use the above result to compute the transition matrix $e^{\boldsymbol{A}t}$. Recall that

$$e^{\boldsymbol{A}t} = \boldsymbol{I}_n + \boldsymbol{A}t + \frac{\boldsymbol{A}^2t^2}{2!} + \frac{\boldsymbol{A}^3t^3}{3!} + \cdots$$

Note that

$$\boldsymbol{\Lambda}^k = \begin{bmatrix} \lambda_1^k & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2^k & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_n^k \end{bmatrix}, \quad k \geq 1$$

Hence

$$e^{\boldsymbol{\Lambda} t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & e^{\lambda_n t} \end{bmatrix}.$$

From (3.6), we obtain

$$\boldsymbol{A} = \boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}.$$

Therefore,

$$\begin{aligned} \boldsymbol{A}^k &= \left(\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}\right)^k \\ &= \left(\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}\right)\left(\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}\right) \times \cdots \times \left(\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}\right) \\ &= \boldsymbol{T}\boldsymbol{\Lambda}^k\boldsymbol{T}^{-1}, \end{aligned}$$

and hence

$$\begin{aligned} e^{\boldsymbol{A}t} &= e^{\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}} \\ &= \boldsymbol{I}_n + \boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}t + \frac{\left(\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}\right)^2 t^2}{2!} + \frac{\left(\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^{-1}\right)^3 t^3}{3!} + \cdots \\ &= \boldsymbol{T}\left(\boldsymbol{I}_n + \boldsymbol{\Lambda}t + \frac{\boldsymbol{\Lambda}^2 t^2}{2!} + \frac{\boldsymbol{\Lambda}^3 t^3}{3!}\right)\boldsymbol{T}^{-1} \\ &= \boldsymbol{T}e^{\boldsymbol{\Lambda}t}\boldsymbol{T}^{-1}. \end{aligned} \tag{3.7}$$

Example 3.2 We use the diagonalization method to compute $e^{\boldsymbol{A}t}$ for

$$\boldsymbol{A} = \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix}.$$

Note that this is the same matrix $\boldsymbol{A}$ as in Example 3.1. The eigenvalues of $\boldsymbol{A}$ are distinct so the diagonalization method is applicable here. We first compute an eigenvector, $\boldsymbol{v}_1$, corresponding to the eigenvalue $\lambda_1 = 3$ by solving the eigenvalue-eigenvector equation,

$$\left(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_2\right) \boldsymbol{v}_1 = \boldsymbol{0}.$$

That is,

$$(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_2)\, \boldsymbol{v}_1 = \begin{bmatrix} -3 & 3 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

We take

$$\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We next compute an eigenvector corresponding to the eigenvalue $\lambda_2 = -2$ by solving the equation

$$(\boldsymbol{A} - \lambda_2 \boldsymbol{I}_2)\, \boldsymbol{v}_2 = \begin{bmatrix} 2 & 3 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

We take

$$\boldsymbol{v}_2 = \begin{bmatrix} -3 \\ 2 \end{bmatrix}.$$

We use the above eigenvectors to form the transformation matrix

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ 1 & 2 \end{bmatrix}.$$

Its inverse is

$$\boldsymbol{T}^{-1} = \frac{1}{5} \begin{bmatrix} 2 & 3 \\ -1 & 1 \end{bmatrix}.$$

We are now ready to compute $e^{\boldsymbol{A}t}$ using (3.7),

$$\begin{aligned} e^{\boldsymbol{A}t} &= \boldsymbol{T} e^{\Lambda t} \boldsymbol{T}^{-1} \\ &= \frac{1}{5} \begin{bmatrix} 1 & -3 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} e^{3t} & 0 \\ 0 & e^{-2t} \end{bmatrix} \begin{bmatrix} 2 & 3 \\ -1 & 1 \end{bmatrix} \\ &= \frac{1}{5} \begin{bmatrix} 2e^{3t} + 3e^{-2t} & 3\left(e^{3t} - e^{-2t}\right) \\ 2\left(e^{3t} - e^{-2t}\right) & 3e^{3t} + 2e^{-2t} \end{bmatrix}. \end{aligned}$$

The result is the same as when we used the inverse Laplace transform method.

The above example illustrates that the transformation $\boldsymbol{T}$ producing the diagonal form of $\boldsymbol{A}$ is a powerful computational tool. The existence of a diagonalizing $\boldsymbol{T}$ is not always guaranteed. The inability to transform $\boldsymbol{A}$ to diagonal form is connected with the existence of repeated eigenvalues that we discuss next.

3.3 Computing Functions of a Square Matrix

In this section, we discuss methods of computing functions of a square matrix that is not necessarily diagonalizable. The methods discussed allow us to easily compute the matrix exponential and hence the state transition matrix of LTI systems. A major tool used by the methods discussed is the so called Jordan canonical form that we discuss next.

3.3.1 The Jordan Canonical Form

If a square n-by-n matrix $\boldsymbol{A}$ has repeated eigenvalues, then it may not be diagonalizable. Although some matrices possessing repeated eigenvalues can be transformed into diagonal form, in many cases it is only possible to obtain the so-called "Jordan canonical form" or just the "Jordan from," for short. To illustrate the point, we consider the case when the characteristic equation, $\Delta(\lambda) = (\lambda - \lambda_1)^n$, that is, the matrix $\boldsymbol{A}$ has just one eigenvalue with multiplicity n. For simplicity, we analyze the case when $n = 4$ and suppose that

$$\operatorname{rank}\left[\ \boldsymbol{A} - \lambda_1 \boldsymbol{I}_4\ \right] = 3.$$

That is, the nullity of the matrix $\left[\ \boldsymbol{A} - \lambda_1 \boldsymbol{I}_4\ \right]$ is one, which means that there is only one eigenvector corresponding to the eigenvalue λ_1. Thus such a matrix $\boldsymbol{A}$ cannot be diagonalized. We need three more linearly independent vectors to represent $\boldsymbol{A}$ in a new basis, where it has a particularly simple form that is as close as possible to being diagonal.

To proceed, we need a notion of a generalized eigenvector that we define next.

Definition 3.1 *A vector $\boldsymbol{v}$ is called a generalized eigenvector of grade k if*

$$\left(\boldsymbol{A} - \lambda \boldsymbol{I}_n\right)^k \boldsymbol{v} = \boldsymbol{0}$$

and

$$\left(\boldsymbol{A} - \lambda \boldsymbol{I}_n\right)^{k-1} \boldsymbol{v} \neq \boldsymbol{0}.$$

$\diamond$

For $k = 1$, we have

$$\left(\boldsymbol{A} - \lambda \boldsymbol{I}_n\right) \boldsymbol{v} = \boldsymbol{0},$$

which means that $\boldsymbol{v}$ is an ordinary eigenvector corresponding to the eigenvalue λ.

Suppose now that $\boldsymbol{A} \in \mathbb{R}^{4\times 4}$ and $\boldsymbol{v}$ is a generalized eigenvector of grade $n = 4$ corresponding to the repeated eigenvalue λ_1 whose multiplicity is also 4. Then by Definition 3.1,

$$\left(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4\right)^4 \boldsymbol{v} = \boldsymbol{0} \quad \text{but} \quad \left(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4\right)^3 \boldsymbol{v} \neq \boldsymbol{0}.$$

Let

$$\boldsymbol{v}_1 = (\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)^3 \boldsymbol{v}.$$

Then,

$$\begin{aligned}(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)\, \boldsymbol{v}_1 &= (\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)^4 \boldsymbol{v} \\ &= \boldsymbol{0}.\end{aligned}$$

Hence

$$\boldsymbol{A}\boldsymbol{v}_1 = \lambda_1 \boldsymbol{v}_1. \tag{3.8}$$

We also have to have

$$(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)^2 \boldsymbol{v} \neq \boldsymbol{0},$$

otherwise we would have $(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)^3 \boldsymbol{v} = \boldsymbol{0}$ as well, which would mean that $\boldsymbol{v}$ is not a generalized eigenvector of grade 4.

Let

$$\boldsymbol{v}_2 = (\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)^2 \boldsymbol{v}.$$

Note that

$$\begin{aligned}(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)\, \boldsymbol{v}_2 &= (\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)^3 \boldsymbol{v} \\ &= \boldsymbol{v}_1.\end{aligned}$$

Therefore,

$$\boldsymbol{A}\boldsymbol{v}_2 = \boldsymbol{v}_1 + \lambda_1 \boldsymbol{v}_2. \tag{3.9}$$

We proceed in a similar way by letting

$$\boldsymbol{v}_3 = (\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)\, \boldsymbol{v}.$$

We observer that

$$\begin{aligned}(\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)\, \boldsymbol{v}_3 &= (\boldsymbol{A} - \lambda_1 \boldsymbol{I}_4)^2 \boldsymbol{v} \\ &= \boldsymbol{v}_2.\end{aligned}$$

Hence,

$$\boldsymbol{A}\boldsymbol{v}_3 = \boldsymbol{v}_2 + \lambda_1 \boldsymbol{v}_3. \tag{3.10}$$

Finally, we let

$$\boldsymbol{v}_4 = \boldsymbol{v}$$

and we note that

$$\begin{aligned}(\boldsymbol{A}-\lambda_1\boldsymbol{I}_4)\,\boldsymbol{v}_4 &= (\boldsymbol{A}-\lambda_1\boldsymbol{I}_4)\,\boldsymbol{v}\\ &= \boldsymbol{v}_3.\end{aligned}$$

Therefore,

$$\boldsymbol{A}\boldsymbol{v}_4 = \boldsymbol{v}_3 + \lambda_1\boldsymbol{v}_4. \tag{3.11}$$

Combining (3.8)–(3.11), we obtain

$$\begin{aligned}\boldsymbol{A}\begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4\end{bmatrix} &= \begin{bmatrix}\lambda_1\boldsymbol{v}_1 & \boldsymbol{v}_1+\lambda_1\boldsymbol{v}_2 & \boldsymbol{v}_2+\lambda_1\boldsymbol{v}_3 & \boldsymbol{v}_3+\lambda_1\boldsymbol{v}_4\end{bmatrix}\\ &= \begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4\end{bmatrix}\begin{bmatrix}\lambda_1 & 1 & 0 & 0\\ 0 & \lambda_1 & 1 & 0\\ 0 & 0 & \lambda_1 & 1\\ 0 & 0 & 0 & \lambda_1\end{bmatrix}.\end{aligned} \tag{3.12}$$

Let

$$\boldsymbol{T} = \begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4\end{bmatrix}.$$

The matrix $\boldsymbol{T}$ is invertible by construction, that is, the set of vectors $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3,\boldsymbol{v}_4\}$ forms a basis for $\mathbb{R}^4$. Therefore we can represent (3.12) as

$$\begin{aligned}\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} &= \begin{bmatrix}\lambda_1 & 1 & 0 & 0\\ 0 & \lambda_1 & 1 & 0\\ 0 & 0 & \lambda_1 & 1\\ 0 & 0 & 0 & \lambda_1\end{bmatrix}\\ &= \boldsymbol{J}.\end{aligned} \tag{3.13}$$

The representation of $\boldsymbol{A}$ in the basis $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3,\boldsymbol{v}_4\}$ has the form of a Jordan block of order 4.

Example 3.3 Consider the matrix

$$\boldsymbol{A} = \begin{bmatrix}5 & 2 & 4\\ 0 & 5 & 3\\ 0 & 0 & 5\end{bmatrix}.$$

The characteristic polynomial is $\Delta(\lambda) = (\lambda-5)^3$ and

$$\operatorname{rank}\begin{bmatrix}\boldsymbol{A}-\lambda\boldsymbol{I}_3\end{bmatrix}\Big|_{\lambda=5} = 2,$$

that is, the nullity of the characteristic matrix, $\left[\; \boldsymbol{A} - \lambda \boldsymbol{I}_3 \; \right]\Big|_{\lambda=5}$, is 1. We proceed to find $\boldsymbol{v}_1$ such that

$$\begin{aligned} [\boldsymbol{A} - \lambda\boldsymbol{I}_3]|_{\lambda=5}\, \boldsymbol{v}_1 &= \begin{bmatrix} 0 & 2 & 4 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \\ v_{31} \end{bmatrix} \\ &= \mathbf{0}. \end{aligned}$$

We take

$$\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Then we proceed to find $\boldsymbol{v}_2$ such that

$$[\boldsymbol{A} - \lambda\boldsymbol{I}_3]|_{\lambda=5}\, \boldsymbol{v}_2 = \boldsymbol{v}_1,$$

that is, we solve for $\boldsymbol{v}_2$ satisfying

$$\begin{bmatrix} 0 & 2 & 4 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{12} \\ v_{22} \\ v_{32} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

We can take

$$\boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 1/2 \\ 0 \end{bmatrix}.$$

Finally we find $\boldsymbol{v}_3$ such that

$$[\boldsymbol{A} - \lambda\boldsymbol{I}_3]|_{\lambda=5}\, \boldsymbol{v}_3 = \boldsymbol{v}_2,$$

that is, we solve for $\boldsymbol{v}_3$ satisfying

$$\begin{bmatrix} 0 & 2 & 4 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{13} \\ v_{23} \\ v_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \\ 0 \end{bmatrix}.$$

We take

$$\boldsymbol{v}_3 = \begin{bmatrix} 0 \\ -2/6 \\ 1/6 \end{bmatrix}.$$

The transformation matrix $\boldsymbol{T}$ has the form

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & -2/6 \\ 0 & 0 & 1/6 \end{bmatrix},$$

and

$$\boldsymbol{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 4 \\ 0 & 0 & 6 \end{bmatrix}.$$

The matrix $\boldsymbol{A}$ in the new basis has the form

$$\begin{aligned} \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 4 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & 2 & 4 \\ 0 & 5 & 3 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & -2/6 \\ 0 & 0 & 1/6 \end{bmatrix} \\ &= \begin{bmatrix} 5 & 1 & 0 \\ 0 & 5 & 1 \\ 0 & 0 & 5 \end{bmatrix}, \end{aligned}$$

which is a third order Jordan block.

If the rank of the characteristic matrix evaluated at λ_1 is $n-2$, then the equation

$$[\boldsymbol{A} - \lambda_1 \boldsymbol{I}_n]\, \boldsymbol{v} = \boldsymbol{0},$$

has two linearly independent solutions. Therefore, $\boldsymbol{A}$ has two linearly independent eigenvectors. Let $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ be linearly independent eigenvectors associated with the eigenvalue λ_i. In order to transform $\boldsymbol{A}$ into its Jordan form, we need to find $n-2$ generalized eigenvectors. In this case we have two chains of generalized eigenvectors, $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k\}$ and $\{\boldsymbol{v}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{v}_l\}$, where $k + l = n$. In the basis

$$\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k, \boldsymbol{v}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{v}_l\}$$

the matrix $\boldsymbol{A}$ will have a block diagonal form

$$\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} = \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{J}_2 \end{bmatrix},$$

where $\boldsymbol{J}_1$ is a Jordan block of order k and $\boldsymbol{J}_2$ is a Jordan block of order l.

Example 3.4 Let

$$\boldsymbol{A} = \begin{bmatrix} 3 & 1 & -1 \\ 0 & 6 & -3 \\ 0 & 3 & 0 \end{bmatrix}.$$

The characteristic polynomial

$$\Delta(\lambda) = (\lambda - 3)^3$$

and

$$\begin{aligned} \operatorname{rank} [\boldsymbol{A} - \lambda \boldsymbol{I}_3]|_{\lambda=3} &= \begin{bmatrix} 0 & 1 & -1 \\ 0 & 3 & -3 \\ 0 & 3 & -3 \end{bmatrix} \\ &= 1 \end{aligned}$$

As two linearly independent solutions to

$$[\boldsymbol{A} - \lambda \boldsymbol{I}_3]|_{\lambda=3}\, \boldsymbol{v} = \boldsymbol{0},$$

we can take

$$\boldsymbol{u}_1 = \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{v}_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

Now we have to find one generalized eigenvector that together with $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ would form a basis for $\mathbb{R}^3$ and enable us to construct a transformation that brings $\boldsymbol{A}$ into its Jordan form. We try first $\boldsymbol{u}_1$ to construct $\boldsymbol{u}_2$ such that

$$[\boldsymbol{A} - \lambda \boldsymbol{I}_3]|_{\lambda=3}\, \boldsymbol{u}_2 = \boldsymbol{u}_1,$$

that is, we solve for $\boldsymbol{u}_2$ satisfying

$$\begin{bmatrix} 0 & 1 & -1 \\ 0 & 3 & -3 \\ 0 & 3 & -3 \end{bmatrix} \begin{bmatrix} u_{12} \\ u_{22} \\ u_{32} \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix}.$$

We can take

$$\boldsymbol{u}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

We form the transformation matrix

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \boldsymbol{v}_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 3 & 1 & 1 \\ 3 & 0 & 1 \end{bmatrix}.$$

Its inverse is

$$\boldsymbol{T}^{-1} = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & -1 \\ -3 & 3 & -2 \end{bmatrix}.$$

It is then easy to check that

$$\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} = \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{J}_2 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix},$$

where

$$\boldsymbol{J}_1 = \begin{bmatrix} 3 & 1 \\ 0 & 3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{J}_2 = 3.$$

In general, the Jordan canonical form is a block diagonal matrix, where each diagonal block is a Jordan block corresponding to a possibly different eigenvalue, that is,

$$\begin{aligned} \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} &= \begin{bmatrix} \begin{matrix} \lambda_1 & 1 & 0 & \cdots & 0 \\ 0 & \lambda_1 & 1 & \cdots & 0 \\ 0 & 0 & \lambda_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_1 \end{matrix} & & \\ & \ddots & \\ & & \begin{matrix} \lambda_r & 1 & 0 & \cdots & 0 \\ 0 & \lambda_r & 1 & \cdots & 0 \\ 0 & 0 & \lambda_r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_r \end{matrix} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{J}_2 & & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{J}_q \end{bmatrix}, \end{aligned} \tag{3.14}$$

where each $r_i \times r_i$ Jordan block has the form $\boldsymbol{J}_i = \lambda_i$ if $r_i = 1$, or, for $r_i > 1$ the corresponding Jordan block has a non-diagonal structure. Note that

$$n = r_1 + r_2 + \cdots + r_q$$

and

$$q \geq r,$$

which means that there may be more than one Jordan blocks associated with one eigenvalue as was the case in Example 3.4.

Example 3.5 To illustrate the general case depicted in (3.14), we consider, as in Chen [55, p. 60], a 5-by-5 matrix $\boldsymbol{A}$ with repeated eigenvalue λ_1 of multiplicity 4 and a simple eigenvalue λ_2, that is, the characteristic polynomial of $\boldsymbol{A}$ is

$$\Delta(\lambda) = (\lambda - \lambda_1)^4(\lambda - \lambda_2).$$

Depending on the nullity of the matrix $[\boldsymbol{A} - \lambda_1 \boldsymbol{I}_5]$, the Jordan form of the matrix $\boldsymbol{A}$ make take one of the following forms:

- If the nullity of $[\boldsymbol{A} - \lambda_1 \boldsymbol{I}_5]$ is 1, then there is only one eigenvector, say $\boldsymbol{u}_1$ associated with λ_1 and we need to compute three generalized eigenvectors to form a basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4, \boldsymbol{v}_1\}$, where $\boldsymbol{v}_1$ is an eigenvector associated with λ_2. The matrix $\boldsymbol{A}$ in this basis will have the form:

$$\hat{\boldsymbol{A}}_1 = \boldsymbol{T}_1^{-1}\boldsymbol{A}\boldsymbol{T}_1 = \begin{bmatrix} \lambda_1 & 1 & 0 & 0 & \vdots & 0 \\ 0 & \lambda_1 & 1 & 0 & \vdots & 0 \\ 0 & 0 & \lambda_1 & 1 & \vdots & 0 \\ 0 & 0 & 0 & \lambda_1 & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \vdots & \lambda_2 \end{bmatrix}.$$

  Note that we can re-order the basis vectors as $\{\boldsymbol{v}_1, \boldsymbol{u}_1, \boldsymbol{u}_1, \boldsymbol{u}_3, \boldsymbol{u}_4\}$. In this basis the matrix $\boldsymbol{A}$ representation has the form

$$\hat{\boldsymbol{A}}_2 = \boldsymbol{T}_2^{-1}\boldsymbol{A}\boldsymbol{T}_2 = \begin{bmatrix} \lambda_2 & \vdots & 0 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \vdots & \lambda_1 & 1 & 0 & 0 \\ 0 & \vdots & 0 & \lambda_1 & 1 & 0 \\ 0 & \vdots & 0 & 0 & \lambda_1 & 1 \\ 0 & \vdots & 0 & 0 & 0 & \lambda_1 \end{bmatrix}.$$

- If the nullity of $[\boldsymbol{A} - \lambda_1 \boldsymbol{I}_5]$ is 2, then there are two eigenvectors associated with λ_1, that is, there are two Jordan blocks associated with λ_1. The matrix $\boldsymbol{A}$ may take one of the following forms:

$$\hat{\boldsymbol{A}}_3 = \boldsymbol{T}_3^{-1}\boldsymbol{A}\boldsymbol{T}_3 = \begin{bmatrix} \lambda_1 & 1 & 0 & \vdots & 0 & & 0 \\ 0 & \lambda_1 & 1 & \vdots & 0 & & 0 \\ 0 & 0 & \lambda_1 & \vdots & 0 & & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & 0 & \vdots & \lambda_1 & \vdots & 0 \\ & & & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & & 0 & \vdots & \lambda_2 \end{bmatrix}$$

or

$$\hat{\boldsymbol{A}}_4 = \boldsymbol{T}_4^{-1}\boldsymbol{A}\boldsymbol{T}_4 = \begin{bmatrix} \lambda_1 & 1 & \vdots & 0 & 0 & & 0 \\ 0 & \lambda_1 & \vdots & 0 & 0 & & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \vdots & \lambda_1 & 1 & \vdots & 0 \\ 0 & 0 & \vdots & 0 & \lambda_1 & \vdots & 0 \\ & & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & & 0 & 0 & \vdots & \lambda_2 \end{bmatrix}.$$

- If the nullity of $[\boldsymbol{A} - \lambda_1 \boldsymbol{I}_5]$ is 3, then there are three eigenvectors associated with λ_1, that is, there are three Jordan blocks associated with λ_1. The matrix $\boldsymbol{A}$ may take the following form:

$$\hat{\boldsymbol{A}}_5 = \boldsymbol{T}_5^{-1}\boldsymbol{A}\boldsymbol{T}_5 = \begin{bmatrix} \lambda_1 & 1 & \vdots & 0 & & 0 & & 0 \\ 0 & \lambda_1 & \vdots & 0 & & 0 & & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & & & \\ 0 & 0 & \vdots & \lambda_1 & \vdots & 0 & & 0 \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & & 0 & \vdots & \lambda_1 & \vdots & 0 \\ & & & & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & & 0 & & 0 & \vdots & \lambda_2 \end{bmatrix}.$$

- Finally, if the nullity of $[\boldsymbol{A} - \lambda_1 \boldsymbol{I}_5]$ is 4, then there are four eigenvectors associated with λ_1, that is, there are four Jordan blocks associated with λ_1. The Jordan form of the matrix matrix $\boldsymbol{A}$ is a diagonal matrix,

$$\hat{\boldsymbol{A}}_6 = \boldsymbol{T}_6^{-1}\boldsymbol{A}\boldsymbol{T}_6 = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & 0 & \lambda_1 & 0 \\ 0 & 0 & 0 & 0 & \lambda_2 \end{bmatrix}.$$

To summarize our discussion, we need the following definitions.

Definition 3.2 *For a given eigenvalue λ_i of an n-by-n matrix $\boldsymbol{A}$, its algebraic multiplicity is the multiplicity of it in the characteristics polynomial $\Delta(\lambda)$, that is, the algebraic multiplicity of λ_i is the sum of orders of all Jordan blocks associated with λ_i.*

◇

Definition 3.3 *For a given eigenvalue λ_i of an n-by-n matrix $\boldsymbol{A}$, its geometric multiplicity is the nullity of the associated characteristic matrix $\left[\ \boldsymbol{A} - \lambda_i \boldsymbol{I}_n\ \right]$, that is, the geometric multiplicity of λ_i equals the number of Jordan blocks associated with it.*

◇

Then we have the following theorem:

Theorem 3.2 *A given n-by-n matrix $\boldsymbol{A}$ is diagonalizable if and only if geometric multiplicities of its eigenvalues are the same as their algebraic multiplicities, that is, if and only if $\boldsymbol{A}$ has n linearly independent eigenvectors.*

We close our discussion on the Jordan canonical form with an observation by Owens [260, p. 34] that in almost all situations of practical relevance, the matrix $\boldsymbol{A}$ has distinct eigenvalues. So in a sense the cases described in this section can be considered "pathological" from the applications point of view. In the literature, see for example [55, p. 61], if $\boldsymbol{A}$ cannot be diagonalized then it is called to be defective. Calculation of the Jordan canonical is numerically ill conditioned. One can try MATLAB's function `jordan`. However, the matrix $\boldsymbol{A}$ must be known exactly, which in MATLAB's language means that its elements must be integers or ratios of small integers. Finally, to diagonalize a square matrix using MATLAB, one can call the function `eig`. If, however, $\boldsymbol{A}$ is not diagonalizable, then the function `eig` will give incorrect solution.

3.3.2 Computing Polynomial Functions of a Square Matrix

If $\boldsymbol{A}$ is an n-by-n square matrix then its k-th power is

$$\boldsymbol{A}^k = \underbrace{\boldsymbol{A}\boldsymbol{A} \times \cdots \times \boldsymbol{A}}_{k \text{ times}},$$

where

$$\boldsymbol{A}^0 = \boldsymbol{I}_n.$$

Suppose now that we have a polynomial

$$f(\lambda) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \cdots + \alpha_1\lambda + \alpha_0.$$

Then we define $f(\boldsymbol{A})$ as

$$f(\boldsymbol{A}) = \boldsymbol{A}^n + \alpha_{n-1}\boldsymbol{A}^{n-1} + \cdots + \alpha_1\boldsymbol{A} + \alpha_0\boldsymbol{I}_n.$$

For example, if

$$f(\lambda) = \lambda^2 + 2\lambda - 3 = (\lambda + 3)(\lambda - 1),$$

then

$$f(\boldsymbol{A}) = \boldsymbol{A}^2 + 2\boldsymbol{A} - 3\boldsymbol{I}_n = (\boldsymbol{A} + 3\boldsymbol{I}_n)(\boldsymbol{A} - \boldsymbol{I}_n).$$

Now if the matrix $\boldsymbol{A}$ is block diagonal, for example, such as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_3 \end{bmatrix},$$

where $\boldsymbol{A}_i$ are square matrices and $\boldsymbol{O}$ denotes a matrix of zeros of appropriate dimensions, then we have

$$\boldsymbol{A}^k = \begin{bmatrix} \boldsymbol{A}_1^k & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2^k & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_3^k \end{bmatrix}$$

and

$$f(\boldsymbol{A}) = \begin{bmatrix} f(\boldsymbol{A}_1) & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & f(\boldsymbol{A}_2) & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & f(\boldsymbol{A}_3) \end{bmatrix}.$$

Consider a similar matrix $\hat{\boldsymbol{A}}$ of the given matrix $\boldsymbol{A}$, that is,

$$\hat{\boldsymbol{A}} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T},$$

where $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ is an invertible matrix. The above can be re-written as

$$\boldsymbol{A} = \boldsymbol{T}\hat{\boldsymbol{A}}\boldsymbol{T}^{-1},$$

and we have

$$\begin{aligned}
\boldsymbol{A}^k &= \left(\boldsymbol{T}\hat{\boldsymbol{A}}\boldsymbol{T}^{-1}\right)\left(\boldsymbol{T}\hat{\boldsymbol{A}}\boldsymbol{T}^{-1}\right)\times\cdots\times\left(\boldsymbol{T}\hat{\boldsymbol{A}}\boldsymbol{T}^{-1}\right)\\
&= \boldsymbol{T}\hat{\boldsymbol{A}}^k\boldsymbol{T}^{-1},
\end{aligned}$$

or

$$\begin{aligned}
\hat{\boldsymbol{A}}^k &= \left(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\right)\left(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\right)\times\cdots\times\left(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\right)\\
&= \boldsymbol{T}^{-1}\boldsymbol{A}^k\boldsymbol{T}.
\end{aligned}$$

Thus, we have

$$f\left(\boldsymbol{A}\right) = \boldsymbol{T}f(\hat{\boldsymbol{A}})\boldsymbol{T}^{-1} \tag{3.15}$$

or

$$f(\hat{\boldsymbol{A}}) = \boldsymbol{T}^{-1}f\left(\boldsymbol{A}\right)\boldsymbol{T}. \tag{3.16}$$

Now let λ_i be the i-th eigenvalue of $\boldsymbol{A}$ and suppose that the algebraic multiplicity of λ_i is n_i. Then the characteristic polynomial of $\boldsymbol{A}$ can be written as,

$$\Delta(\lambda) = \det(\lambda\boldsymbol{I}_n - \boldsymbol{A}) = \prod_{i=1}^{r}\left(\lambda-\lambda_i\right)^{n_i}. \tag{3.17}$$

We now discuss a very useful property of Jordan blocks that we utilize in the proof of the very important theorem of linear algebra, the theorem of Cayley and Hamilton. Suppose that we have a fourth order Jordan block,

$$\boldsymbol{J} = \begin{bmatrix} \lambda_i & 1 & 0 & 0\\ 0 & \lambda_i & 1 & 0\\ 0 & 0 & \lambda_i & 1\\ 0 & 0 & 0 & \lambda_i \end{bmatrix}.$$

Then we have

$$(\boldsymbol{J} - \lambda_i\boldsymbol{I}) = \begin{bmatrix} 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Let us now compute successive powers of $(\boldsymbol{J} - \lambda \boldsymbol{I})$. We obtain

$$(\boldsymbol{J} - \lambda_i \boldsymbol{I})^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (\boldsymbol{J} - \lambda_i \boldsymbol{I})^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$(\boldsymbol{J} - \lambda_i \boldsymbol{I})^k = \boldsymbol{O} \quad \text{for} \quad k \geq 4.$$

A matrix with above property is called a *nilpotent* matrix.

In general, if we have a Jordan block of order r_i then

$$(\boldsymbol{J} - \lambda_i \boldsymbol{I})^k = \boldsymbol{O} \quad \text{for} \quad k \geq r_i. \tag{3.18}$$

We have all we need to state and prove the theorem of Cayley and Hamilton.

Theorem 3.3 *(Cayley-Hamilton theorem) Let*

$$\Delta(\lambda) = \det[\lambda \boldsymbol{I}_n - \boldsymbol{A}] = \prod_{i=1}^{r} (\lambda - \lambda_i)^{n_i},$$

be the characteristic polynomial of a matrix $\boldsymbol{A}$. A square matrix satisfies its characteristic polynomial equation,

$$\Delta(\boldsymbol{A}) = \prod_{i=1}^{r} (\boldsymbol{A} - \lambda_i \boldsymbol{I}_n)^{n_i} = \boldsymbol{O}.$$

Proof Suppose that $\hat{\boldsymbol{A}}$ is the Jordan canonical form of $\boldsymbol{A}$. Thus there exists a non-singular matrix $\boldsymbol{T}$ such that

$$\hat{\boldsymbol{A}} = \boldsymbol{T}^{-1} \boldsymbol{A} \boldsymbol{T} = \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{J}_2 & & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{J}_q \end{bmatrix}.$$

Let us now consider the matrix polynomial $\Delta(\boldsymbol{A})$. We use (3.15) to obtain

$$\begin{aligned}
\Delta(\boldsymbol{A}) &= \boldsymbol{T}\Delta(\hat{\boldsymbol{A}})\boldsymbol{T}^{-1} \\
&= \boldsymbol{T}\prod_{i=1}^{r}\left(\hat{\boldsymbol{A}} - \lambda_i \boldsymbol{I}_n\right)^{n_i}\boldsymbol{T}^{-1} \\
&= \boldsymbol{T}\prod_{i=1}^{r}\begin{bmatrix} \boldsymbol{J}_1 - \lambda_i\boldsymbol{I}_{r_1} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{J}_2 - \lambda_i\boldsymbol{I}_{r_2} & & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{J}_q - \lambda_i\boldsymbol{I}_{r_q} \end{bmatrix}^{n_i}\boldsymbol{T}^{-1} \\
&= \boldsymbol{T}\prod_{i=1}^{r}\begin{bmatrix} (\boldsymbol{J}_1 - \lambda_i\boldsymbol{I}_{r_1})^{n_i} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & (\boldsymbol{J}_2 - \lambda_i\boldsymbol{I}_{r_2})^{n_i} & & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & (\boldsymbol{J}_q - \lambda_i\boldsymbol{I}_{r_q})^{n_i} \end{bmatrix}\boldsymbol{T}^{-1}.
\end{aligned}$$

The order of any Jordan block associated with λ_i cannot exceed the corresponding algebraic multiplicity. Taking into account the nilpotent property given by (3.18), we obtain

$$\begin{aligned}
\Delta(\boldsymbol{A}) &= \boldsymbol{T}\begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & (\boldsymbol{J}_2 - \lambda_1\boldsymbol{I}_{r_2})^{n_1} & & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & (\boldsymbol{J}_q - \lambda_1\boldsymbol{I}_{r_q})^{n_1} \end{bmatrix} \times \\
& \qquad \cdots \times \begin{bmatrix} (\boldsymbol{J}_1 - \lambda_r\boldsymbol{I}_{r_1})^{n_r} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & (\boldsymbol{J}_2 - \lambda_r\boldsymbol{I}_{r_2})^{n_r} & & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} \end{bmatrix}\boldsymbol{T}^{-1} \\
&= \boldsymbol{T}\begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} \end{bmatrix}\boldsymbol{T}^{-1} \\
&= \boldsymbol{O},
\end{aligned} \tag{3.19}$$

which completes the proof of the theorem.

□

It follows from the Cayley-Hamilton theorem that

$$\Delta(\boldsymbol{A}) = \boldsymbol{A}^n + \alpha_{n-1}\boldsymbol{A}^{n-1} + \cdots + \alpha_1\boldsymbol{A} + \alpha_0\boldsymbol{I}_n = \boldsymbol{O},$$

that is,

$$\boldsymbol{A}^n = -\alpha_{n-1}\boldsymbol{A}^{n-1} - \cdots - \alpha_1\boldsymbol{A} - \alpha_0\boldsymbol{I}_n. \tag{3.20}$$

Thus $\boldsymbol{A}^n$ can be expressed as a linear combination of the matrices

$$\{\boldsymbol{I}_n, \boldsymbol{A}, \ldots, \boldsymbol{A}^{n-1}\}.$$

If we multiply both sides of (3.20) by $\boldsymbol{A}$, then we obtain

$$\boldsymbol{A}^{n+1} = -\alpha_{n-1}\boldsymbol{A}^n - \cdots - \alpha_1\boldsymbol{A}^2 - \alpha_0\boldsymbol{A}. \tag{3.21}$$

Thus, $\boldsymbol{A}^{n+1}$ can be expressed as a linear combination of

$$\{\boldsymbol{A}, \boldsymbol{A}^2, \ldots, \boldsymbol{A}^n\}.$$

Using (3.20), we conclude that $\boldsymbol{A}^{n+1}$ can be expressed as a linear combination of $\{\boldsymbol{I}_n, \boldsymbol{A}, \ldots, \boldsymbol{A}^{n-1}\}$. Using the same argument, we conclude that any $\boldsymbol{A}^k$, $k \geq 0$, can be expressed as a linear combination of $\{\boldsymbol{I}_n, \boldsymbol{A}, \ldots, \boldsymbol{A}^{n-1}\}$. Therefore, any matrix polynomial $f(\boldsymbol{A})$ can also be expressed as

$$f(\boldsymbol{A}) = \beta_0\boldsymbol{I}_n + \beta_1\boldsymbol{A} + \cdots + \beta_{n-1}\boldsymbol{A}^{n-1}$$

for some β_i, $i = 0, 1, \ldots, n-1$. The above property is used to efficiently compute matrix polynomials of $\boldsymbol{A}$, which can then be used to compute the matrix exponential $e^{\boldsymbol{A}t}$ as well as the state transition matrix for discrete-time systems. In what follows, we present such a method and illustrate it with examples. We follow the presentation of Chen [55, pp. 63–67].

Suppose we are given a polynomial $f(\lambda)$ and we are interested in calculating $f(\boldsymbol{A})$. We begin with dividing $f(\lambda)$ by $\Delta(\lambda)$ to obtain

$$f(\lambda) = q(\lambda)\Delta(\lambda) + h(\lambda), \tag{3.22}$$

where $q(\lambda)$ is the quotient, $h(\lambda)$ is the remainder, and

$$\deg_\lambda h(\lambda) < n.$$

We evaluate $f(\boldsymbol{A})$ using (3.22) and the Cayley-Hamilton theorem to obtain

$$\begin{aligned} f(\boldsymbol{A}) &= q(\boldsymbol{A})\Delta(\boldsymbol{A}) + h(\boldsymbol{A}) \\ &= q(\boldsymbol{A})\boldsymbol{O} + h(\boldsymbol{A}) \\ &= h(\boldsymbol{A}). \end{aligned}$$

Thus instead of evaluating the matrix polynomial $f(\boldsymbol{A})$ that may contain very high powers of $\boldsymbol{A}$, we can find $f(\boldsymbol{A})$ by computing $h(\boldsymbol{A})$, where the highest power of $\boldsymbol{A}$ is less than n.

In addition, rather than performing the division (3.22), we can find $h(\lambda)$ by assuming that it has the form

$$h(\lambda) = \beta_0 + \beta_1\lambda + \cdots + \beta_{n-1}\lambda^{n-1}$$

and then solve n equations for β_i, $i = 0, 1, \ldots, n-1$. Consider first the case when $\boldsymbol{A}$ has distinct eigenvalues. To obtain equations involving the coefficients β_i, we note that

$$f(\lambda_i) = q(\lambda_i)\Delta(\lambda_i) + h(\lambda_i) = h(\lambda_i), \quad i = 1, 2, \ldots, n.$$

Example 3.6 For a given

$$f(\lambda) = \lambda^7,$$

we compute

$$f(\boldsymbol{A}) = \boldsymbol{A}^7,$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix}. \tag{3.23}$$

The characteristic polynomial of $\boldsymbol{A}$ is

$$\Delta(\lambda) = \lambda^2 - \lambda - 2 = (\lambda+1)(\lambda-2).$$

Let

$$h(\lambda) = \beta_0 + \beta_1\lambda.$$

Then we have two equations:

$$f(\lambda_i) = h(\lambda_i), \quad i = 1, 2,$$

that is,

$$f(-1) = (-1)^7 = -1 = \beta_0 - \beta_1$$

and

$$f(2) = 2^7 = 128 = \beta_0 + 2\beta_1.$$

We represent the above linear equations for β_0 and β_1 in the matrix format as

$$\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} -1 \\ 128 \end{bmatrix}.$$

Solving, we obtain $\beta_0 = 42$ and $\beta_1 = 43$. Hence

$$f(\boldsymbol{A}) = \boldsymbol{A}^7 = 42\boldsymbol{I}_2 + 43\boldsymbol{A} = \begin{bmatrix} 42 & 86 \\ 43 & 85 \end{bmatrix}.$$

Example 3.7 We can use the above method to evaluate $e^{\boldsymbol{A}t}$, where $\boldsymbol{A}$ is given by (3.23). In other words, let $f(\lambda) = e^{\lambda t}$; we wish to find

$$f(\boldsymbol{A}) = e^{\boldsymbol{A}t}.$$

Let $h(\lambda) = \beta_0 + \beta_1\lambda$. Then

$$f(-1) = h(-1)$$

yields

$$e^{-t} = \beta_0 - \beta_1,$$

and

$$f(2) = h(2)$$

yields

$$e^{2t} = \beta_0 + 2\beta_1.$$

We thus have two equations for β_0 and β_1,

$$\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} e^{-t} \\ e^{2t} \end{bmatrix}.$$

Solving the above gives

$$\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} e^{-t} \\ e^{2t} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2e^{-t} + e^{2t} \\ -e^{-t} + e^{2t} \end{bmatrix}.$$

Hence

$$e^{\boldsymbol{A}t} = \beta_0 \boldsymbol{I}_2 + \beta_1 \boldsymbol{A} = \frac{1}{3}\begin{bmatrix} 2e^{-t} + e^{2t} & -2e^{-t} + 2e^{2t} \\ -e^{-t} + e^{2t} & e^{-t} + 2e^{2t} \end{bmatrix}.$$

When the characteristic polynomial has repeated zeros, we use the derivatives of $f(\lambda)$ and $h(\lambda)$ to obtain additional equations to determine the coefficients of $h(\lambda)$ as per the following theorem.

Theorem 3.4 *Chen [55, p. 64] We are given $f(\lambda)$ and an $n \times n$ matrix $\boldsymbol{A}$ with characteristic polynomial*

$$\Delta(\lambda) = \prod_{i=1}^{m} (\lambda - \lambda_i)^{n_i},$$

where $n = \sum_{i=1}^{m} n_i$. Let

$$h(\lambda) = \beta_0 + \beta_1 \lambda + \cdots + \beta_{n-1} \lambda^{n-1}.$$

It is a polynomial of degree $n-1$ with n unknown coefficients. These n unknowns are to be solved from the following set of n equations:

$$f^{(l)}(\lambda_i) = h^{(l)}(\lambda_i) \quad \text{for} \quad l = 0, 1, \ldots, n_i - 1 \quad \text{and} \quad i = 1, 2, \ldots, m$$

where

$$f^{(l)}(\lambda_i) = \frac{d^l f(\lambda)}{d\lambda^l} \Big|_{\lambda = \lambda_i}$$

and $h^{(l)}(\lambda_i)$ is similarly defined. Then we have

$$f(\boldsymbol{A}) = h(\boldsymbol{A})$$

and $h(\lambda)$ is said to be equal $f(\lambda)$ on the spectrum of $\boldsymbol{A}$.

Example 3.8 We compute $e^{\boldsymbol{A}t}$ for

$$\boldsymbol{A} = \begin{bmatrix} 3 & 0 & 1 \\ -1 & 1 & 2 \\ -2 & 0 & 0 \end{bmatrix}.$$

The characteristic polynomial is

$$\Delta(\lambda) = (\lambda - 1)^2 (\lambda - 2).$$

Let $f(\lambda) = e^{\lambda t}$ and let $h(\lambda) = \beta_0 + \beta_1 \lambda + \beta_2 \lambda^2$. Then using Theorem 3.4, we obtain

$$\begin{aligned} f(1) &= h(1), \quad e^t = \beta_0 + \beta_1 + \beta_2, \\ f'(1) &= h'(1), \quad te^t = \beta_1 + 2\beta_2, \\ f(2) &= h(2), \quad e^{2t} = \beta_0 + 2\beta_1 + 4\beta_2. \end{aligned}$$

Thus we have a set of equations

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} e^t \\ te^t \\ e^{2t} \end{bmatrix}.$$

Solving the above gives

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 0 & -2 & 1 \\ 2 & 3 & -2 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} e^t \\ te^t \\ e^{2t} \end{bmatrix} = \begin{bmatrix} -2te^t + e^{2t} \\ 2e^t + 3te^t - 2e^{2t} \\ -e^t - te^t + e^{2t} \end{bmatrix}.$$

Hence

$$\begin{aligned} e^{\boldsymbol{A}t} &= h(\boldsymbol{A}) \\ &= \beta_0 \boldsymbol{I}_3 + \beta_1 \boldsymbol{A} + \beta_2 \boldsymbol{A}^2 \\ &= \left(-2te^t + e^{2t}\right) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \left(2e^t + 3te^t - 2e^{2t}\right) \begin{bmatrix} 3 & 0 & 1 \\ -1 & 1 & 2 \\ -2 & 0 & 0 \end{bmatrix} \\ &\quad + \left(-e^t - te^t + e^{2t}\right) \begin{bmatrix} 7 & 0 & 3 \\ -8 & 1 & 1 \\ -6 & 0 & -2 \end{bmatrix} \\ &= \begin{bmatrix} -e^t + 2e^{2t} & 0 & -e^t + e^{2t} \\ 6e^t + 5te^t - 6e^{2t} & e^t & 3e^t + 5te^t - 3e^{2t} \\ 2e^t - 2e^{2t} & 0 & 2e^t - e^{2t} \end{bmatrix}. \end{aligned}$$

Example 3.9 Suppose we are given the Jordan block of order n_i,

$$\boldsymbol{J} = \begin{bmatrix} \lambda_i & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_i & 1 & \cdots & 0 & 0 \\ 0 & 0 & \lambda_i & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_i & 1 \\ 0 & 0 & 0 & \cdots & 0 & \lambda_i \end{bmatrix}.$$

The characteristic polynomial of $\boldsymbol{J}$ is $\Delta(\lambda) = (\lambda - \lambda_i)^{n_i}$. For computational simplicity we select the $(n_i - 1)$-st order polynomial $h(\lambda)$ of the form

$$h(\lambda) = \beta_0 + \beta_1(\lambda - \lambda_i) + \beta_2(\lambda - \lambda_i)^2 + \cdots + \beta_{n_i-1}(\lambda - \lambda_i)^{n_i-1}.$$

The condition $f(\lambda) = h(\lambda)$ on the spectrum of $\boldsymbol{J}$ yields

$$\beta_0 = f(\lambda_i), \quad \beta_1 = f'(\lambda_i), \quad \beta_2 = \frac{f''(\lambda_i)}{2!}, \ldots, \beta_{n_i-1} = \frac{f^{(n_i-1)}(\lambda_i)}{(n_i-1)!}.$$

Hence

$$\begin{aligned} f(\boldsymbol{J}) &= f(\lambda_i)\boldsymbol{I}_{n_i} + \frac{f'(\lambda_i)}{1!}(\boldsymbol{J}-\lambda_i\boldsymbol{I}_{n_i}) + \frac{f''(\lambda_i)}{2!}(\boldsymbol{J}-\lambda_i\boldsymbol{I}_{n_i})^2 + \cdots \\ &\quad \cdots + \frac{f^{(n_i-1)}(\lambda_i)}{(n_i-1)!}(\boldsymbol{J}-\lambda_i\boldsymbol{I}_{n_i})^{n_i-1}. \end{aligned}$$

Using the nilpotent property of the matrix $(\boldsymbol{J}-\lambda_i\boldsymbol{I}_{n_i})$, we obtain

$$f(\boldsymbol{J}) = \begin{bmatrix} f(\lambda_i) & f'(\lambda_i)/1! & f''(\lambda_i)/2! & \cdots & f^{(n_i-1)}(\lambda_i)/(n_i-1)! \\ 0 & f(\lambda_i) & f'(\lambda_i)/1! & \cdots & f^{(n_i-2)}(\lambda_i)/(n_i-2)! \\ 0 & 0 & f(\lambda_i) & \cdots & f^{(n_i-3)}(\lambda_i)/(n_i-3)! \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ddots & f'(\lambda_i)/1! \\ 0 & 0 & 0 & \cdots & f(\lambda_i) \end{bmatrix}.$$

In particular, if $f(\lambda) = e^{\lambda t}$, then we have

$$e^{\boldsymbol{J}t} = \begin{bmatrix} e^{\lambda_i t} & te^{\lambda_i t} & t^2e^{\lambda_i t}/2! & \cdots & t^{n_i-1}e^{\lambda_i t}/(n_i-1)! \\ 0 & e^{\lambda_i t} & te^{\lambda_i t} & \cdots & t^{n_i-2}e^{\lambda_i t}/(n_i-2)! \\ 0 & 0 & e^{\lambda_i t} & \cdots & t^{n_i-3}e^{\lambda_i t}/(n_i-3)! \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ddots & te^{\lambda_i t} \\ 0 & 0 & 0 & \cdots & e^{\lambda_i t} \end{bmatrix}.$$

3.4 State Equation Solution of Controlled System

Consider a model of a controlled dynamical system,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{3.24}$$

subject to a given initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, where $\boldsymbol{B}$ is an $n \times m$ matrix, and $m < n$ as is usually in practice. Premultiplying both sides of (3.24) by the integrating factor, $e^{-\boldsymbol{A}t}$, yields

$$e^{-\boldsymbol{A}t}\dot{\boldsymbol{x}}(t) = e^{-\boldsymbol{A}t}\boldsymbol{A}\boldsymbol{x}(t) + e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{u}(t).$$

We then rearrange the above equation as

$$e^{-\boldsymbol{A}t}\dot{\boldsymbol{x}}(t) - e^{-\boldsymbol{A}t}\boldsymbol{A}\boldsymbol{x}(t) = e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{u}(t). \tag{3.25}$$

Noting that

$$\frac{d}{dt}\left(e^{-\boldsymbol{A}t}\boldsymbol{x}(t)\right) = -\boldsymbol{A}e^{-\boldsymbol{A}t}\boldsymbol{x}(t) + e^{-\boldsymbol{A}t}\dot{\boldsymbol{x}}(t),$$

we can represent (3.25) in the form

$$\frac{d}{dt}\left(e^{-\boldsymbol{A}t}\boldsymbol{x}(t)\right) = e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{u}(t).$$

Integrating both sides of the resulting equation gives

$$e^{-\boldsymbol{A}t}\boldsymbol{x}(t) - \boldsymbol{x}(0) = \int_0^t e^{-\boldsymbol{A}\tau}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau.$$

Hence

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}(0) + \int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau.$$

If the initial condition is $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, then integration from t_0 to t yields

$$\boxed{\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau} \tag{3.26}$$

As in the case of the uncontrolled system, we can also use the Laplace transform technique to obtain the solution to (3.24) subject to the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$. Indeed, taking the Laplace transform of (3.24) and performing some manipulations yields

$$\boldsymbol{X}(s) = (sI_n - \boldsymbol{A})^{-1}\boldsymbol{x}(0) + (sI_n - \boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{U}(s).$$

Hence

$$\boldsymbol{x}(t) = \mathcal{L}^{-1}\left\{(sI_n - \boldsymbol{A})^{-1}\boldsymbol{x}(0)\right\} + \mathcal{L}^{-1}\left\{(sI_n - \boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{U}(s)\right\}.$$

If $\boldsymbol{x}_0 = \boldsymbol{x}(t_0)$, then

$$\boldsymbol{x}(t) = \mathcal{L}^{-1}\left\{e^{-t_0 s}(sI_n - \boldsymbol{A})^{-1}\boldsymbol{x}(t_0)\right\} + \mathcal{L}^{-1}\left\{e^{-t_0 s}(sI_n - \boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{U}(s)\right\}.$$

3.5 From Continuous to Discrete Models

Our objective here is to present a method for constructing linear discrete-time models from given linear continuous-time models.

We use a sample-and-hold device that transforms a continuous signal, $f(t)$, into the staircase signal,

$$f(kh), \quad kh \le t < (k+1)h,$$

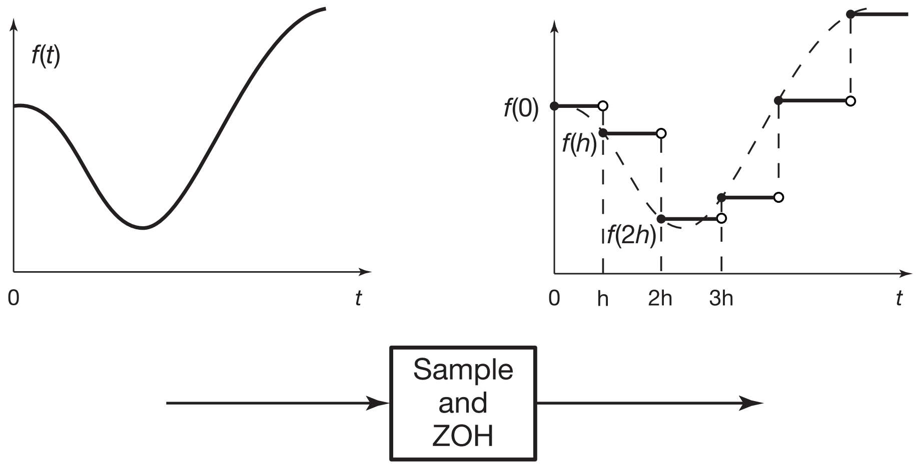


Figure 3.1: Sample and zero-order hold (ZOH) element operating on a continuous function.

where h is the sampling period. The sample and zero-order hold (ZOH) operation is illustrated in Figure 3.1.

Suppose that we are given a continuous-time model,

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}_0 = \boldsymbol{x}(t_0) && (3.27) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t). && (3.28) \end{aligned}$$

The solution to the state equation is

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau. \tag{3.29}$$

We assume that the input to the system is generated by a sample and zero-order device and has the form,

$$\boldsymbol{u}(t) = \boldsymbol{u}(k), \quad kh \le t < (k+1)h.$$

Let $t_0 = kh$ and $t = (k+1)h$ and let us use shorthand notation,

$$\boldsymbol{x}[k] = \boldsymbol{x}(kh)$$

Then taking into account that $\boldsymbol{u}(t)$ is constant on the interval $[kh, (k+1)h)$ we represent (3.29) as

$$\begin{aligned} \boldsymbol{x}[k+1] &= e^{\boldsymbol{A}h}\boldsymbol{x}[k] + \int_{kh}^{(k+1)h} e^{\boldsymbol{A}(kh+h-\tau)}\boldsymbol{B}\boldsymbol{u}[k]d\tau \\ &= e^{\boldsymbol{A}h}\boldsymbol{x}[k] + \int_{kh}^{(k+1)h} e^{\boldsymbol{A}(kh+h-\tau)}\boldsymbol{B}d\tau\, \boldsymbol{u}[k]. && (3.30) \end{aligned}$$

Consider now the second term on the right-hand side of the above equation. Let

$$\eta = kh + h - \tau.$$

Then we can represent (3.30) as

$$\begin{aligned}\boldsymbol{x}[k+1] &= e^{\boldsymbol{A}h}\boldsymbol{x}[k] + \int_{kh}^{(k+1)h} e^{\boldsymbol{A}(kh+h-\tau)}\boldsymbol{B}d\tau\, \boldsymbol{u}[k] \\ &= e^{\boldsymbol{A}h}\boldsymbol{x}[k] + \int_0^h e^{\boldsymbol{A}\eta}\boldsymbol{B}d\eta\, \boldsymbol{u}[k] \\ &= \boldsymbol{A}_d\boldsymbol{x}[k] + \boldsymbol{B}_d\boldsymbol{u}[k],\end{aligned} \tag{3.31}$$

where

$$\boldsymbol{A}_d = e^{\boldsymbol{A}h} \quad \text{and} \quad \boldsymbol{B}_d = \int_0^h e^{\boldsymbol{A}\eta}\boldsymbol{B}d\eta.$$

The discrete output equation has the form

$$\boldsymbol{y}[k] = \boldsymbol{C}\boldsymbol{x}[k].$$

We point out that the above discrete model, employing the zero-order hold, is exact in the sense that the state, input, and output of this model are exactly the same at the sampling times as the state, input, and output of the underlying continuous-time model.

Example 3.10 We construct an exact linear discrete state-space model for the continuous system model,

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\boldsymbol{x}(k) + \begin{bmatrix} 2 \\ 1 \end{bmatrix}u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix}\boldsymbol{x}(t).\end{aligned}$$

The sampling period is $h = 1$ sec.

We first evaluate $\boldsymbol{A}_d = e^{\boldsymbol{A}h}$. Note that $\boldsymbol{A}^2 = \boldsymbol{O}$. Hence

$$\begin{aligned}\boldsymbol{A}_d &= \boldsymbol{I}_2 + \boldsymbol{A}h \\ &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.\end{aligned}$$

Next, we evaluate

$$\boldsymbol{B}_d = \int_0^h e^{\boldsymbol{A}\tau}\boldsymbol{B}d\tau.$$

We obtain

$$\begin{aligned}
\boldsymbol{B}_d &= \int_0^h e^{\boldsymbol{A}\tau}\boldsymbol{B}d\tau \\
&= \int_0^h \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} d\tau \\
&= \int_0^1 \begin{bmatrix} 2+\tau \\ 1 \end{bmatrix} d\tau \\
&= \begin{bmatrix} 2\tau + \frac{1}{2}\tau^2 \\ \tau \end{bmatrix} \Bigg|_0^1 \\
&= \begin{bmatrix} 2.5 \\ 1 \end{bmatrix}.
\end{aligned}$$

The discrete model then has the form

$$\begin{aligned}
\boldsymbol{x}[k+1] &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}[k] + \begin{bmatrix} 2.5 \\ 1 \end{bmatrix} \\
y[k] &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}[k].
\end{aligned}$$

3.6 State Plane Analysis

The *state plane*, which is a state space of two-dimensional systems, is also called the *phase plane*. Analysis in the state plane is applicable to linear and nonlinear systems modeled by second-order ordinary differential equations. The state-plane methods are graphical procedures for solving such equations. Using state-plane methods one can graphically determine the transient response of a second-order dynamical system. We now introduce relevant terms. Consider a class of second-order systems whose dynamics are modeled by the differential equation,

$$\ddot{x} = f(x, \dot{x}). \tag{3.32}$$

Define the state variables,

$$x_1 = x \quad \text{and} \quad x_2 = \dot{x}.$$

Then, equation (3.32) can be represented as

$$\left.\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f(x_1, x_2)
\end{aligned}\right\} \tag{3.33}$$

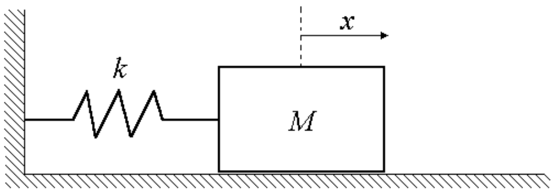


Figure 3.2: A spring-mass system of Example 3.11.

The plane with coordinates x_1 and x_2 is labeled as the state plane or phase plane. A solution of (3.32) can be illustrated by plotting x versus t, or by plotting x_2 versus x_1 using t as a parameter. To each value of the state $\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^T$ there corresponds a point in the (x_1, x_2)-plane called the *representative point* (RP). As t varies, the RP describes a curve in the state plane called a *trajectory*. A family of trajectories is called a *phase portrait*.

3.6.1 Examples of Phase Portraits

Example 3.11 Consider the spring-mass system shown in Figure 3.2, where x is the displacement of the mass M from equilibrium, and k is the linear spring constant. The spring is assumed to be linear, that is, it obeys Hooke's law. Applying Newton's law, we obtain the equation of motion of the mass,

$$M\ddot{x} + kx = 0.$$

Let $\omega^2 = k/M$. Then, the above equation can be represented in standard form,

$$\ddot{x} + \omega^2 x = 0. \tag{3.34}$$

Observe that

$$\ddot{x} = \dot{x}\frac{\ddot{x}}{\dot{x}} = \dot{x}\frac{\frac{d\dot{x}}{dt}}{\frac{dx}{dt}} = \dot{x}\frac{d\dot{x}}{dx}.$$

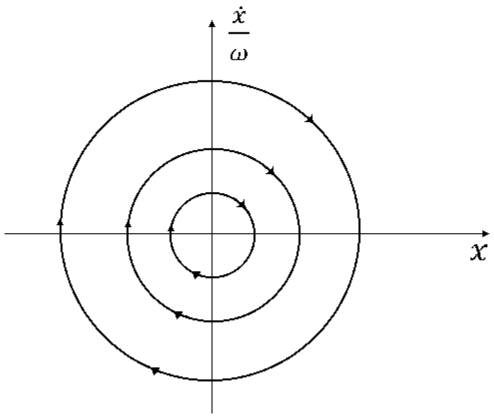


Figure 3.3: A phase portrait for the spring-mass system of Example 3.11.

Using the above, we represent equation (3.34) as

$$\dot{x}\frac{d\dot{x}}{dx} + \omega^2 x = 0.$$

Separating the variables yields

$$\frac{\dot{x}}{\omega^2}d\dot{x} + xdx = 0.$$

Integrating the above gives

$$\left(\frac{\dot{x}}{\omega}\right)^2 + x^2 = A^2,$$

where A is the integration constant determined by the initial conditions. We see that a phase portrait of equation (3.34) consists of a family of circles centered at the origin of the $(x, \frac{\dot{x}}{\omega})$-plane as illustrated in Figure 3.3. Different circular trajectories, in this plane, correspond to different values of the constant A representing initial conditions. Note that

$$x = \omega \int \frac{\dot{x}}{\omega} dt,$$

where $\omega > 0$. Hence, for positive values of $\dot{x}/\omega$ the displacement x increases. This means that for the values $\dot{x}/\omega > 0$ the arrows on the trajectories indicate x moving to the right. For negative values of $\dot{x}/\omega$, that is, for the values of $\dot{x}/\omega$ below the horizontal axis, the arrows on the trajectories indicate x decreasing by going to the left. Therefore, as time increases the RP moves in a clockwise direction along the trajectory.

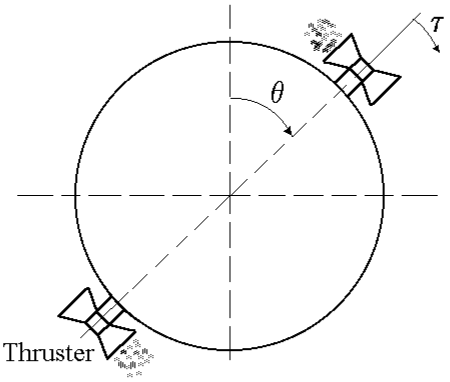


Figure 3.4: A model of a rigid satellite of Example 3.12.

Example 3.12 In this example, we consider a model of a satellite shown in Figure 3.4. We assume that the satellite is rigid, and in a frictionless environment. It can rotate about an axis perpendicular to the page as a result of torque τ applied to the satellite by firing the thrusters. The system input is the applied torque, and the system output is the attitude angle θ. The satellite's moment of inertia is I. Applying the rotational version of Newton's second law, we obtain the satellite's equation of motion,

$$I\frac{d^2\theta}{dt^2} = \tau,$$

which can be represented in the following equivalent form:

$$\frac{d^2\theta}{dt^2} = \frac{1}{I}\tau.$$

Let

$$\frac{1}{I}\tau = u,$$

then the equation describing the satellite's motion becomes

$$\frac{d^2\theta}{dt^2} = u. \tag{3.35}$$

We assume that when the thrusters fire, the thrust is constant, that is, $u = \pm U$. For simplicity, let $U = 1$. Using the state variables $x_1 = \theta$ and $x_2 = \dot{\theta}$, we obtain the satellite's state-space model,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{3.36}$$

where $u = \pm 1$. We first consider the case when $u = 1$. With this control, the system equations are:

$$\left.\begin{array}{rcl} \dot{x}_1 &=& x_2 \\ \dot{x}_2 &=& 1. \end{array}\right\}$$

Solving the second equation yields

$$x_2 = t + c_1,$$

where c_1 is an integration constant. We substitute $x_2 = t + c_1$ into $\dot{x}_1 = x_2$ to get

$$\dot{x}_1 = t + c_1.$$

Solving the above equation, we obtain

$$\begin{array}{rcl} x_1 &=& \displaystyle\int (t + c_1) dt \\ &=& \dfrac{1}{2}(t + c_1)^2 + c_2 \\ &=& \dfrac{1}{2}x_2^2 + c_2, \end{array}$$

where c_2 is an integration constant. In the state plane the above equation represents a family of parabolas open toward the positive x_1 axis. The representative point moves upward along these parabolas because $\dot{x}_2 = 1$, that is, $\dot{x}_2 > 0$. Typical system trajectories for this case are shown in Figure 3.5.

Similarly, when $u = -1$, we have

$$\dot{x}_2 = -1.$$

Solving the above differential equation, yields

$$x_2 = -t + \tilde{c}_1,$$

where $\tilde{c}_1$ is an integration constant. Substituting $x_2 = -t + \tilde{c}_1$ into $\dot{x}_1 = x_2$, and then solving the resulting equation yields

$$x_1 = -\frac{1}{2}x_2^2 + \tilde{c}_2,$$

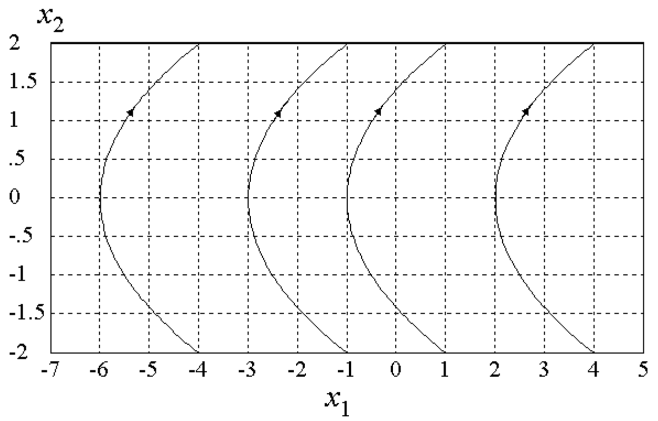


Figure 3.5: Trajectories of the system (3.36) for $u = 1$.

where $\tilde{c}_2$ is an integration constant. In the state plane, this equation represents a family of parabolas open toward the negative x_1 axis. The representative point moves downward along these parabolas because $\dot{x}_2 = -1$, that is, $\dot{x} < 0$. Typical system trajectories for this case are shown in Figure 3.6.

In the above examples, systems' trajectories were constructed analytically. Analytical methods are useful for systems modeled by differential equations that can be easily solved. If the system of differential equations cannot be easily solved analytically, we can use graphical or numerical methods. We now describe a graphical method for solving second-order, or a system of two first-order, differential equations.

3.6.2 The Method of Isoclines

An *isocline* is a curve along which the trajectories' slope is constant. In other words, an isocline is a locus, a set of points, of the trajectories' constant slope. To obtain equations that describe isoclines, we consider a system of two first-order differential equations of the

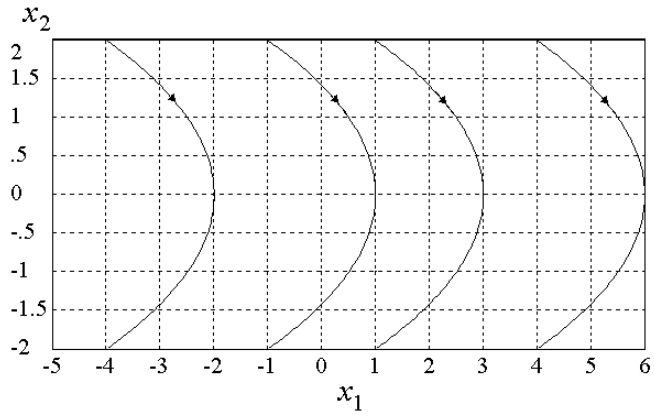


Figure 3.6: Trajectories of the system (3.36) for $u = -1$.

form

$$\left.\begin{aligned} \dot{x}_1 &= \tfrac{dx_1}{dt} = f_1(x_1, x_2) \\ \dot{x}_2 &= \tfrac{dx_2}{dt} = f_2(x_1, x_2). \end{aligned}\right\} \tag{3.37}$$

Dividing the second of the above equations by the first one, we obtain

$$\frac{dx_2}{dx_1} = \frac{f_2(x_1, x_2)}{f_1(x_1, x_2)}. \tag{3.38}$$

Thus, we eliminated the independent variable t from the set of the first-order differential equations given by (3.37). In equation (3.38), we consider x_1 as the independent variable, and x_2 as dependent. Let

$$m = m(x_1, x_2) = \frac{dx_2}{dx_1}.$$

Note that m is just the slope of the tangent to the trajectory passing through the point $\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$. The locus of the trajectory constant slope

$$\frac{dx_2}{dx_1} = m(x_1, x_2) = \text{constant},$$

can be obtained from the equation

$$f_2(x_1, x_2) = m f_1(x_1, x_2).$$

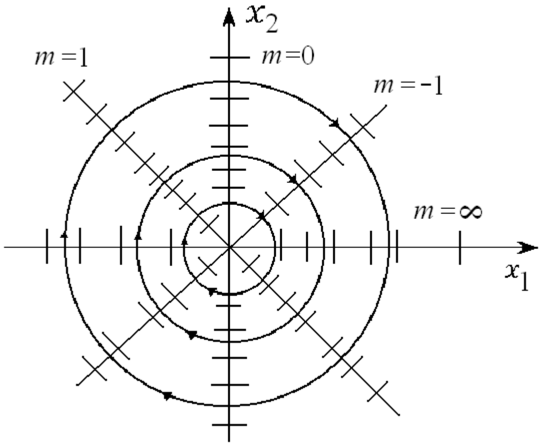


Figure 3.7: Isoclines and a typical trajectory for the spring-mass system of Example 3.13.

The curve that satisfies the above equation is an isocline corresponding to the trajectories' slope m because a trajectory crossing the isocline will have its slope equal to m. The idea of the method of isoclines is to construct several isoclines in the state plane. This then allows one to construct a field of local tangents m. Then, the trajectory passing through any given point in the state plane is obtained by drawing a continuous curve following the directions of the field.

Example 3.13 Consider the spring-mass system from Example 3.11. Defining the state variables $x_1 = x$ and $x_2 = \dot{x}$, we obtain the state equations

$$\left.\begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & -\omega^2 x_1. \end{array}\right\}$$

Let, for simplicity, $\omega = 1$. Then,

$$\frac{dx_2}{dx_1} = m = -\frac{x_1}{x_2}.$$

From the above we obtain the equation of isoclines

$$x_2 = -\frac{1}{m} x_1.$$

This equation describes a family of straight lines through the origin. In Figure 3.7, we show some of the isoclines and a typical trajectory for the above spring-mass system. Note that in this example the trajectory slope is m, while the isocline slope is $-1/m$. This means that trajectories of the spring-mass system are perpendicular to the isoclines, that is, the trajectories are circles centered at the origin of the state plane.

Example 3.14 Given a second-order differential equation

$$\ddot{x} = -4\eta\dot{x} - 4x, \tag{3.39}$$

where η is a parameter. We will find values of η for which there are isoclines along which the trajectory slope and the isocline slope are equal. Such isoclines, if they exist, are called the *asymptotes*.

Let $x_1 = x$ and $x_2 = \dot{x}$. Then, the second-order differential equation given by (3.39) can be represented as a system of first-order differential equations,

$$\left.\begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & -4\eta x_2 - 4x_1 \end{array}\right\}$$

The isocline equation is

$$\frac{dx_2}{dx_1} = \frac{-4\eta x_2 - 4x_1}{x_2} = m,$$

or equivalently

$$x_2 = -\left(\frac{4}{m + 4\eta}\right) x_1. \tag{3.40}$$

A trajectory slope is equal to an isocline slope if and only if

$$-\frac{4}{m + 4\eta} = m,$$

that is,

$$m^2 + 4\eta m + 4 = 0. \tag{3.41}$$

The discriminant of (3.41) is

$$\Delta = 4\sqrt{\eta^2 - 1}.$$

Hence, the condition for the existence of asymptotes is

$$|\eta| \geq 1. \tag{3.42}$$

Example 3.15 Consider the system of differential equations,

$$\left.\begin{aligned} \dot{x}_1 &= x_1 + x_2 + 1 \\ \dot{x}_2 &= -x_2 + 2. \end{aligned}\right\}$$

(i) Find the equation of the isocline corresponding to the trajectory slope $m = 1$;

(ii) Does the given system have asymptotes? If yes, then write the equations of the asymptotes.

The trajectory slope is

$$\frac{dx_2}{dx_1} = m = \frac{-x_2 + 2}{x_1 + x_2 + 1}.$$

Hence, the equation of the isocline corresponding to the trajectory slope $m = 1$ is obtained by solving

$$\frac{-x_2 + 2}{x_1 + x_2 + 1} = 1.$$

We obtain

$$x_2 = -\frac{1}{2}x_1 + \frac{1}{2}.$$

The trajectory slope is

$$m = \frac{-x_2 + 2}{x_1 + x_2 + 1}. \tag{3.43}$$

From (3.43), we obtain an equation of isoclines of the form

$$x_2 = -\frac{m}{m+1}x_1 - \frac{m-2}{m+1}.$$

An asymptote is the isocline for which its slope and the trajectory slope are equal. Therefore, we can obtain the equations of asymptotes by solving

$$m = -\frac{m}{m+1}.$$

We obtain

$$m^2 + 2m = m(m+2) = 0.$$

Hence, the equations of asymptotes are

$$x_2 = 2 \quad \text{and} \quad x_2 = -2x_1 - 4.$$

We can use a computer to draw phase portraits interactively. Below is an example of MATLAB's script that was used to sketch a phase portrait of the van der Pol equation shown in Figure 3.8.

```
t0=0;                           % initial time
tf=20;                          % final time
tspan=tf-t0;
x0=[-4 -4]';                    % first initial condition
button=1;
p=4*[-1 0;1 0];                 % preparation to draw internal axes
clf;plot(p(:,1),p(:,2))
hold on
plot(p(:,2),p(:,1))
axis(4*[-1 1 -1 1])
    while(button==1)

% Select a new initial condition and press the left button
% Press the right button if you wish to stop

      [t,x]=ode45(@my_xdot,tspan,x0); % invoke your own m-file
      plot(x(:,1),x(:,2))
      [x1,x2,button]=ginput(1);
      x0=[x1 x2]';
    end
```

The van der Pol equation has the form

$$\ddot{x} - \mu(1 - x^2)\dot{x} + x = 0, \tag{3.44}$$

where μ is a parameter. We can equivalently represent the van der Pol equation as

$$\left.\begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & \mu(1 - x_1^2)x_2 - x_1 \end{array}\right\} \tag{3.45}$$

An inspection of the phase portrait of the van der Pol equation reveals the existence of a closed trajectory in the state plane. This solution is an example of a limit cycle.

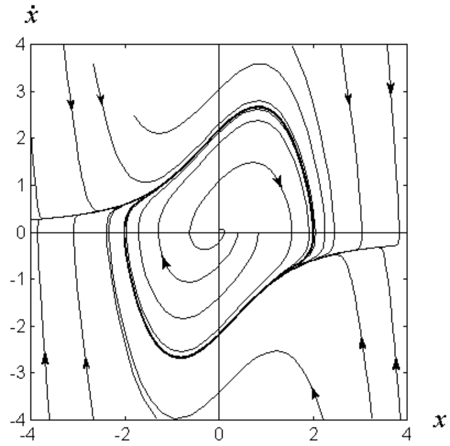


Figure 3.8: A phase-plane portrait of the van der Pol equation for $\mu = 1$.

Definition 3.4 *A limit cycle is a closed trajectory in the state plane, such that no other closed trajectory can be found arbitrarily close to it.*

◇

Note that the differential equation modeling the spring-mass system does not have a limit cycle because the solutions are not isolated. Limit cycles can only appear in nonlinear systems; they do not exist in linear systems.

Theorem 3.5 (Bendixson) *If f_1 and f_2 in (3.38) have continuous partial derivatives in a domain $\mathcal{A}$ bounded by a closed path $\boldsymbol{C}$ and if $\partial f_1/\partial x_1 + \partial f_2/\partial x_2$ does not vanish and is of constant sign in that domain, then (3.38) can have no limit cycle in that domain.*

Proof We prove the theorem by contradiction. Equation (3.38) is satisfied on any of its trajectory. Therefore, if (3.38) has a limit cycle then clearly

$$f_1 dx_2 - f_2 dx_1 = 0. \tag{3.46}$$

We now apply Green's theorem, stated on page 1030, to the vector field

$$\begin{bmatrix} -f_2 & f_1 \end{bmatrix}^T$$

to obtain

$$\int_{\boldsymbol{C}} f_1\, dx_2 - f_2 dx_1 = \iint_{\mathcal{A}} \left(\frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_2} \right) dx_1 dx_2. \tag{3.47}$$

By (3.46), the right-hand side of (3.47) must vanish, which contradicts the hypothesis. Thus, if $\partial f_1/\partial x_1 + \partial f_2/\partial x_2$ does not vanish and is of constant sign in a domain $\mathcal{A}$ bounded by a closed path $\boldsymbol{C}$, then no limit cycle can exist in $\mathcal{A}$. The proof of the theorem is complete.

□

Example 3.16 We apply Bendixson's theorem to the van der Pol equation represented by (3.45). We have

$$f_1 = x_2, \quad \text{and} \quad f_2 = \mu(1 - x_1^2)x_2 - x_1.$$

We calculate $\partial f_1/\partial x_1 + \partial f_2/\partial x_2$ to get

$$\partial f_1/\partial x_1 + \partial f_2/\partial x_2 = \mu(1 - x_1^2).$$

It follows from the Bendixson theorem that there are no limit cycles in the region $|x_1| < 1$ because $\partial f_1/\partial x_1 + \partial f_2/\partial x_2 > 0$ in that region.

Bendixson's theorem gives us a necessary condition for a closed curve to be a limit cycle. It is useful for the purpose of establishing the nonexistence of a limit cycle. In general, it is very difficult to establish the existence of a limit cycle. Sufficient conditions for the existence of a limit cycle are discussed, for example, by Hochstadt [138, Chapter 7]. A method for predicting limit cycles, called the describing function method, is presented in Section 4.3.

3.7 Numerical Techniques

In this section, we discuss numerical techniques for solving differential equations. We begin our presentation of numerical techniques that allow us to determine approximate solutions to the given system of differential equations with a method based on a Taylor series expansion. Much of this section presentation is based on the Class Notes of the late Professor V. B. Haas [124].

3.7.1 The Method of Taylor Series

The basis for many numerical methods for solving differential equations is the Taylor formula. Suppose we are given a state-space model of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{3.48}$$

The method, we discuss, depends on the fact that if $\boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$ is differentiable with respect to its $n+m+1$ variables $x_1, \ldots, x_n, u_1, \ldots, u_m, t$ in some $(n+m+1)$-dimensional region $\mathcal{D}$, and if $\boldsymbol{x}(t)$ is a solution of (3.48), where $(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0))$ lies in $\mathcal{D}$, then we can expand $\boldsymbol{x}(t)$ into a Taylor series to obtain

$$\begin{aligned}
\boldsymbol{x}(t) &= \boldsymbol{x}_0 + \dot{\boldsymbol{x}}(t_0)(t-t_0) + \ddot{\boldsymbol{x}}(t_0)\frac{(t-t_0)^2}{2} + \cdots \\
&= \boldsymbol{x}_0 + \boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0))(t-t_0) \\
&\quad + \left[\frac{\partial \boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0))}{\partial t} + \sum_{i=1}^{n} \frac{\partial \boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0))}{\partial x_i}\dot{x}_i \right.\\
&\quad \left. + \sum_{j=1}^{m} \frac{\partial \boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0))}{\partial u_j}\dot{u}_j\right]\frac{(t-t_0)^2}{2} + \cdots
\end{aligned}$$

If we stop after the q-th term of the Taylor series solution, then the remainder after q terms of a Taylor series gives us an estimate of the error. We can expect the Taylor series solution to give a fairly accurate approximation of the solution of the state-space equations only in a small interval about t_0. Thus, if $\Delta t = t - t_0$ is some small time interval such that the Taylor series converges for $t \in \left[\, t_0, \quad t_0 + \Delta t \,\right]$, we can use the expansion

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \frac{d\boldsymbol{x}(t_0)}{dt}(t-t_0) + \frac{d^2\boldsymbol{x}(t_0)}{dt^2}\frac{(t-t_0)^2}{2} + \cdots$$

to evaluate $\boldsymbol{x}(t_1) = \boldsymbol{x}(t_0 + \Delta t)$. Then, we can start over again solving $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$ with the new initial condition $\boldsymbol{x}(t_1)$. We illustrate the method with simple examples.

Example 3.17 Consider the first-order differential equation

$$\frac{dx}{dt} = t^2 - x^2, \quad x(1) = 1.$$

Expanding $x = x(t)$ into a Taylor series about $x(1) = 1$ yields

$$x(t) = x(1) + \frac{dx(1)}{dt}(t-1) + \frac{d^2x(1)}{dt^2}\frac{(t-1)^2}{2} + \frac{d^3x(1)}{dt^3}\frac{(t-1)^3}{3!} + \frac{d^4x(1)}{dt^4}\frac{(t-1)^4}{4!}\cdots$$

Because $\frac{dx}{dt} = t^2 - x^2$, we obtain by successive differentiations

$$\begin{aligned}
\frac{d^2x}{dt^2} &= 2t - 2x\frac{dx}{dt}, \\
\frac{d^3x}{dt^3} &= 2 - 2\left(\frac{dx}{dt}\right)^2 - 2x\frac{d^2x}{dt^2}, \\
\frac{d^4x}{dt^4} &= -6\frac{dx}{dt}\frac{d^2x}{dt^2} - 2x\frac{d^3x}{dt^3}, \\
&\vdots
\end{aligned}$$

Evaluating the above at $t = 1$, yields

$$\begin{aligned}
\frac{dx(1)}{dt} &= 0, \\
\frac{d^2x(1)}{dt^2} &= 2, \\
\frac{d^3x(1)}{dt^3} &= -2, \\
\frac{d^4x(1)}{dt^4} &= 4, \\
&\vdots
\end{aligned}$$

Hence,

$$x(t) = 1 + (t-1)^2 - \frac{(t-1)^3}{3} + \frac{(t-1)^4}{6} + \cdots .$$

Example 3.18 In this example, we consider the second-order differential equation

$$\frac{d^2x}{dt^2} + 3\frac{dx}{dt} + 6tx = 0,$$

subject to the initial conditions $x(0) = 1$, and $\frac{dx(0)}{dt} = 1$. Using $\frac{d^2x}{dt^2} = -3\frac{dx}{dt} - 6tx$, we obtain by successive differentiation

$$\begin{aligned}
\frac{d^3x}{dt^3} &= -3\frac{d^2x}{dt^2} - 6t\frac{dx}{dt} - 6x, \\
\frac{d^4x}{dt^4} &= -3\frac{d^3x}{dt^3} - 6t\frac{d^2x}{dt^2} - 12\frac{dx}{dt}, \\
\frac{d^5x}{dt^5} &= -3\frac{d^4x}{dt^4} - 6t\frac{d^3x}{dt^3} - 18\frac{d^2x}{dt^2}, \\
&\vdots
\end{aligned}$$

Taking into account the initial conditions, we evaluate all coefficients at $t = 0$ to get

$$\begin{aligned}
\frac{d^2x(0)}{dt^2} &= -3,\\
\frac{d^3x(0)}{dt^3} &= 3,\\
\frac{d^4x(0)}{dt^4} &= -21,\\
\frac{d^5x(0)}{dt^5} &= 117,\\
&\vdots
\end{aligned}$$

Hence,

$$\begin{aligned}
x(t) &= x(0) + \frac{dx(0)}{dt}t + \frac{d^2x(0)}{dt^2}\frac{t^2}{2} + \frac{d^3x(0)}{dt^3}\frac{t^3}{3!} + \frac{d^4x(0)}{dt^4}\frac{t^4}{4!} + \frac{d^5x(0)}{dt^5}\frac{t^5}{5!} + \cdots\\
&= 1 + t - \frac{3}{2}t^2 + \frac{1}{2}t^3 - \frac{7}{8}t^4 + \frac{117}{120}t^5 + \cdots .
\end{aligned}$$

We now illustrate the method of Taylor series when applied to a system of first-order differential equations.

Example 3.19 We consider the following system of two first-order equations

$$\left.\begin{aligned}
\tfrac{dx_1}{dt} &= f_1(t, x_1, x_2)\\
\tfrac{dx_2}{dt} &= f_2(t, x_1, x_2)
\end{aligned}\right\}$$

The initial conditions at $t = t_0$ are: $x_1(t_0) = x_{10}$, $x_2(t_0) = x_{20}$. Having the initial conditions, we can determine the values $\frac{dx_1(t_0)}{dt}$ and $\frac{dx_2(t_0)}{dt}$ from the original equations. We then differentiate $\frac{dx_1}{dt}$ and $\frac{dx_2}{dt}$ to obtain

$$\begin{aligned}
\frac{d^2x_1}{dt^2} &= \frac{d}{dt}f_1\left(t, x_1, x_2, \frac{dx_1}{dt}, \frac{dx_2}{dt}\right)\\
\frac{d^2x_2}{dt^2} &= \frac{d}{dt}f_2\left(t, x_1, x_2, \frac{dx_1}{dt}, \frac{dx_2}{dt}\right).
\end{aligned}$$

Now, $\frac{d^2x_1}{dt^2}$ and $\frac{d^2x_2}{dt^2}$ can be calculated by substituting known quantities on the right-hand sides of the above equations. By successive differentiation and substitution, we can determine

$\frac{d^3x_1}{dt^3}$ and $\frac{d^3x_2}{dt^3}$ as well as higher derivatives at $t = t_0$. The solution to the given system is then computed as

$$\begin{aligned}
x_1(t) &= x_{10} + \frac{dx_1(t_0)}{dt}(t - t_0) + \frac{d^2x_1(t_0)}{dt^2}\frac{(t - t_0)^2}{2} + \frac{d^3x_1(t_0)}{dt^3}\frac{(t - t_0)^3}{3!} + \cdots \\
x_2(t) &= x_{20} + \frac{dx_2(t_0)}{dt}(t - t_0) + \frac{d^2x_2(t_0)}{dt^2}\frac{(t - t_0)^2}{2} + \frac{d^3x_2(t_0)}{dt^3}\frac{(t - t_0)^3}{3!} + \cdots
\end{aligned}$$

The Taylor series method can be programmed on a computer. However, this is not very efficient method from numerical point of view. In the following, we discuss other methods for numerical solution of the state equations that are related to the method of Taylor series. We first present two simple numerical techniques known as the forward and backward Euler methods.

3.7.2 Forward Euler Method

Suppose that we wish to obtain the solution of the modeling equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$$

in the time interval $\left[\ t_0, \ \ t_f\ \right]$, subject to the initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, and the input vector $\boldsymbol{u}(t)$. We divide the interval $\left[\ t_0, \ \ t_f\ \right]$ into N equal subintervals of width

$$h = \frac{t_f - t_0}{N}.$$

We call h the *step length*. We set

$$t_k = t_0 + kh.$$

To proceed further, we approximate the derivative $\frac{d\boldsymbol{x}(t)}{dt}$ at time t_k by

$$\frac{d\boldsymbol{x}(t_k)}{dt} \approx \frac{\boldsymbol{x}(t_{k+1}) - \boldsymbol{x}(t_k)}{h},$$

and write

$$\frac{\boldsymbol{x}(t_{k+1}) - \boldsymbol{x}(t_k)}{h} = \boldsymbol{f}(t_k, \boldsymbol{x}(t_k), \boldsymbol{u}(t_k)).$$

Hence, for $k = 1, 2, \ldots, N$, we have

$$\boxed{\boldsymbol{x}(t_{k+1}) = \boldsymbol{x}(t_k) + h\boldsymbol{f}(t_k, \boldsymbol{x}(t_k), \boldsymbol{u}(t_k))} \tag{3.49}$$

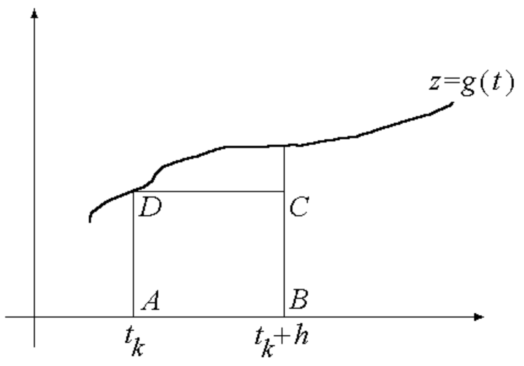


Figure 3.9: Illustration of the rectangular rule of integration.

The above is known as the *forward Euler* algorithm. We see that if h is sufficiently small, we can approximately determine the state at time $t_1 = t_0 + h$ from the initial condition $\boldsymbol{x}_0$ to get

$$\boldsymbol{x}(t_1) = \boldsymbol{x}_0 + h\boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0)).$$

Once we have determined the approximate solution at time t_1, we can determine the approximate solution at time $t_2 = t_1 + h$, and so on. The forward Euler method can also be arrived at when instead of considering the differential equation $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$, we rather start with its equivalent integral representation

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau))d\tau. \tag{3.50}$$

We then use the *rectangular rule* for numerical integration that can be stated as follows. The area under the curve $z = g(t)$ between $t = t_k$ and $t = t_k + h$ is approximately equal to the area of the rectangle $ABCD$ as shown in Figure 3.9, that is,

$$\int_{t_k}^{t_k+h} g(t)dt \approx hg(t_k). \tag{3.51}$$

Applying the rectangular rule of integration to (3.50), we obtain the forward Euler method.

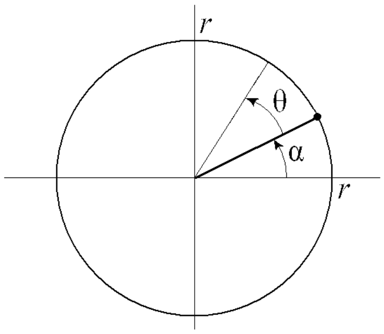


Figure 3.10: Rotating a point mass from α to $\alpha + \theta$.

Example 3.20 We combine basic MATLAB's handle graphics commands with the forward Euler method to write a simple program that animates the motion of a simple pendulum. First, we present a few basic definitions from handle graphics. We start with a description of a handle—it is a floating-point number that MATLAB assigns to every object in the figure window such as a line, tex, label, axes, figure, etc. Thus a handle can be viewed as an object's ID.

Each object has different properties. If we wish to change a property, we need to get hold of the object in question. We use object's handle to get hold of it and then change the object's property using the following commands,

```
get(handle,'PropertyName')
set(handle,'PropertyName','PropertyValue')
```

In our animation of the motion of a pendulum, we use the rotation matrix that we discuss next. Suppose that we have a point mass on a circle of radius r as shown in Figure 3.10. The point mass is connected to the circle center; we call it hinge. The line segment, bar, connecting the hinge and the mass forms the angle α. Therefore, the point mass has the coordinates

$$\boldsymbol{x}_{present} = \begin{bmatrix} r\cos(\alpha) \\ r\sin(\alpha) \end{bmatrix}.$$

Suppose that the bar was rotated by an angle θ from its initial displacement angle α. The new angle of the bar, with the positive x-axis, is then $\alpha+\theta$. We use the following well-known trigonometric identities:

$$\begin{aligned} \cos(\alpha+\theta) &= \cos(\alpha)\cos(\theta) - \sin(\alpha)\sin(\theta) \\ \sin(\alpha+\theta) &= \sin(\alpha)\cos(\theta) + \cos(\alpha)\sin(\theta), \end{aligned}$$

to obtain the mass new coordinates,

$$\begin{aligned}\boldsymbol{x}_{next} &= \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}\boldsymbol{x}_{present}\\ &= \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}r\cos(\alpha)\\ r\sin(\alpha)\end{bmatrix}.\end{aligned}$$

Equipped with the above background results, we are ready to animate the motion of the simple pendulum from Example 2.1. Let

$$x_1 = \theta \quad \text{and} \quad x_2 = \dot{\theta}.$$

Then, we represent the second-order differential equation (2.32) as a system of first-order differential equations,

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -5\sin(x_1).\end{aligned}$$

We approximate the above system of differential equations using Euler's forward method,

$$\begin{aligned}\frac{x_{1next} - x_{1present}}{h} &= x_{2present}\\ \frac{x_{2next} - x_{2present}}{h} &= -5\sin(x_{1present}),\end{aligned}$$

where h is the step size. Combining the above with the handle graphics gives the following simple MATLAB program that animates the simple pendulum motion. This code is based on the code of Example 3 in Pratap [279, p. 207].

```
theta=input('Initial angular displacement of the pendulum in deg= ');
theta=theta*pi/180;           % convert from degrees to radians
tfinal=input('Final time tfinal = ');
data_init=[0 -2]';            % pendulum at rest
R=[cos(theta) -sin(theta);sin(theta) cos(theta)];    % rotation matrix
data=R*data_init;             % initial pendulum position
axis(2*[-1 1 -1 1])
axis('equal')
axis('off')
bar=line('xdata',[0 data(1)],'ydata',[0 data(2)]','linewidth',7,...
    'erase','xor');
```

```
mass=line('xdata',data(1),'ydata',data(2),'linestyle','.',...
    'markersize',50,'erase','xor');
hinge=line('xdata',0,'ydata',0,'linestyle','o','markersize',12,...
    'erase','xor');
h=0.0025;     % step-size for solving differential
t=0;           % initial time
thetadot=0;    % initial angular speed

% solve the differential eqns using the Euler's method
while(t<tfinal)
   t=t+h;
   theta = theta + thetadot*h;
   thetadot=thetadot - 5*sin(theta)*h;
   R=[cos(theta) -sin(theta);sin(theta) cos(theta)];
   datanew=R*data_init;

% change the property values of the bar and hinge objects
   set(bar,'xdata',[0 datanew(1)],'ydata',[0 datanew(2)]);
   set(mass,'xdata',datanew(1),'ydata',datanew(2));
   set(hinge,'xdata',0,'ydata',0)
   drawnow;
end
```

3.7.3 Backward Euler Method

The *backward Euler* method differs from the forward Euler method in the way we approximate the derivative,

$$\frac{d\boldsymbol{x}(t_k)}{dt} \approx \frac{\boldsymbol{x}(t_k) - \boldsymbol{x}(t_{k-1})}{h},$$

This approximation gives the backward Euler formula,

$$\boxed{\boldsymbol{x}(t_{k+1}) = \boldsymbol{x}(t_k) + h\boldsymbol{f}(t_{k+1}, \boldsymbol{x}(t_{k+1}), \boldsymbol{u}(t_{k+1}))}$$

In the above formula, $\boldsymbol{x}(t_{k+1})$ is a function of itself. For this reason, we say that the backward Euler method is an implicit integration algorithm. Implicit integration algorithms require

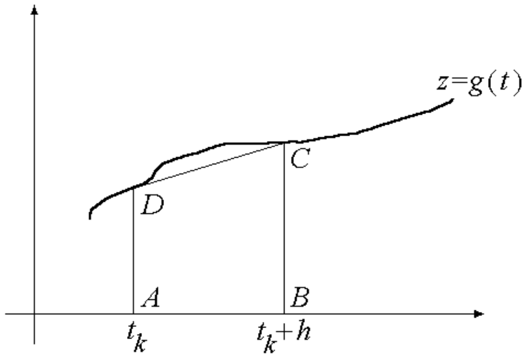


Figure 3.11: The trapezoidal rule of integration.

additional computation to solve for $\boldsymbol{x}(t_{k+1})$. For further discussion of this issue, we refer to Parker and Chua [264].

Euler's methods are rather elementary, and thus, not as accurate as some of the more sophisticated techniques that we are going to discuss next.

3.7.4 Predictor-Corrector Method

The predictor-corrector method, also known as Heun's method, is based on the trapezoidal rule for numerical integration. The trapezoidal rule states that the area under the curve $z = g(t)$ between $t = t_k$ and $t = t_k + h$ is approximately equal to the area of the trapezoid $ABCD$ as depicted in Figure 3.11, that is,

$$\int_{t_k}^{t_k+h} g(t)dt \approx h\frac{g(t_k) + g(t_k + h)}{2}. \tag{3.52}$$

Suppose we wish to find $\boldsymbol{x}(t)$ at $t = t_1 > t_0$. Applying the trapezoidal rule of integration to (3.48), we obtain

$$\boldsymbol{x}(t_1) = \boldsymbol{x}_0 + \frac{h}{2}\left(\boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0)) + \boldsymbol{f}(t_1, \boldsymbol{x}(t_1), \boldsymbol{u}(t_1))\right).$$

The above is an implicit algorithm and it may be viewed as a merging of the forward and backward Euler algorithms. If the function $\boldsymbol{f}$ is nonlinear, then we will, in general, not be able to solve the above equation for $\boldsymbol{x}(t_1)$ exactly. We attempt to obtain $\boldsymbol{x}(t_1)$ iteratively. Let

$$\begin{aligned}
h &= t_{k+1} - t_k, \\
\boldsymbol{x}_1^* &= \boldsymbol{x}_0 + h\boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0)), \\
\boldsymbol{m}_0 &= \boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0)), \\
\boldsymbol{m}_1 &= \boldsymbol{f}(t_1, \boldsymbol{x}_1^*, \boldsymbol{u}(t_1)).
\end{aligned}$$

A first approximation to $\boldsymbol{x}(t_1)$ is

$$\boxed{\boldsymbol{x}_1^* = \boldsymbol{x}_0 + h\boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0))}$$

and is called a *predictor*. The next approximation is

$$\boxed{\boldsymbol{x}(t_1) = \boldsymbol{x}_0 + \frac{h}{2}(\boldsymbol{m}_0 + \boldsymbol{m}_1)}$$

and is called a *corrector*.

Suppose now that we wish to find $\boldsymbol{x}(t)$ at $t = t_f > t_0$. We first divide the interval $\left[\begin{array}{cc} t_0, & t_f \end{array}\right]$ into N equal subintervals using evenly spaced subdivision points to obtain

$$t_0 < t_1 < t_2 < \ldots < t_N = t_f.$$

Once we have found $\boldsymbol{x}(t_1)$, we can apply the same procedure to find $\boldsymbol{x}(t_2)$, and so on.

Example 3.21 We use the predictor-corrector method to find $x(1)$, where the step length $h = 0.5$, for the following differential equation

$$\frac{dx}{dt} = tx^2 - x, \quad x(0) = 1. \tag{3.53}$$

The calculations are summarized in Table 3.1.

We could use $\boldsymbol{x}(t_1)$ obtained from the corrector formula to construct a next approximation to $\boldsymbol{x}(t_1)$. In general, we can use the formula

$$\boldsymbol{x}^{(k)}(t_1) = \boldsymbol{x}_0 + \frac{h}{2}\left(\boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{u}(t_0)) + \boldsymbol{f}(t_1, \boldsymbol{x}^{(k-1)}(t_1), \boldsymbol{u}(t_1)\right)$$

to evaluate $\boldsymbol{x}(t_1)$ until two successive iterations agree to the desired accuracy. This general algorithm is referred to as a second-order predictor-corrector method.

Table 3.1: Solving the differential equation (3.53) using the predictor-corrector method.

t_0	x_0	t_1	m_0	x_1^*	m_1	$m = \frac{m_0+m_1}{2}$	Δx
0	1	0.5	-1.0	0.5	-0.375	-0.688	-0.344
0.5	0.656	1.0	-0.441	0.435	-0.246	-0.344	-0.172
1.0	0.484						

An n-dimensional system of first-order differential equations can be worked analogously. We illustrate the procedure for $n = 2$, where

$$\left.\begin{aligned} \frac{dx}{dt} &= f(t,x,y), \quad x(t_0) = x_0 \\ \frac{dy}{dt} &= g(t,x,y), \quad y(t_0) = y_0 \end{aligned}\right\}$$

Let

$$\begin{aligned} m_0 &= f(t_0, x_0, y_0), \\ n_0 &= g(t_0, x_0, y_0). \end{aligned}$$

Then, the predictor equations are

$$\begin{aligned} x_1^* &= x_0 + m_0\, h, \\ y_1^* &= y_0 + n_0\, h. \end{aligned}$$

Let

$$\begin{aligned} m_1 &= f(t_1, x_1^*, y_1^*), \\ n_1 &= g(t_1, x_1^*, y_1^*). \end{aligned}$$

Then, the corrector equations are

$$\left.\begin{aligned} x(t_1) &= x_0 + \frac{m_0+m_1}{2}\, h \\ y(t_1) &= y_0 + \frac{n_0+n_1}{2}\, h \end{aligned}\right\}$$

Using the above, we can generate the corresponding formulas for t_k.

We now show that the predictor-corrector method yields a solution that agrees with the Taylor series solution through terms of degree two. We consider the scalar case, where f is assumed to be differentiable with respect to its arguments and $\partial^2 f/\partial t \partial x = \partial^2 f/\partial x \partial t$. First observe that $m_1 = f\left(t_1, x_1^*\right)$ can be written as

$$\begin{aligned} f\left(t_1, x_1^*\right) &= f\left(t_0 + h, x_0 + m_0 h\right) \\ &\approx f\left(t_0, x_0\right) + \begin{bmatrix} f_t & f_x \end{bmatrix} \begin{bmatrix} h \\ m_0 h \end{bmatrix} + \frac{1}{2} \begin{bmatrix} h & m_0 h \end{bmatrix} \begin{bmatrix} f_{tt} & f_{tx} \\ f_{xt} & f_{xx} \end{bmatrix} \begin{bmatrix} h \\ m_0 h \end{bmatrix}, \end{aligned}$$

where $f_t = \partial f/\partial t$, $f_x = \partial f/\partial x$, and so on. Substituting the above into the corrector equation yields

$$\begin{aligned} x(t_1) &= x_0 + \frac{h}{2}\left(m_0 + m_1\right) \\ &= x_0 + \frac{h}{2}\left(m_0 + m_0\right) + \frac{h}{2}\left(\begin{bmatrix} f_t & f_x \end{bmatrix}\begin{bmatrix} h \\ m_0 h \end{bmatrix}\right. \\ &\quad \left. + \frac{1}{2}\begin{bmatrix} h & m_0 h \end{bmatrix}\begin{bmatrix} f_{tt} & f_{tx} \\ f_{xt} & f_{xx} \end{bmatrix}\begin{bmatrix} h \\ m_0 h \end{bmatrix}\right) \\ &= x_0 + m_0 h + \left(f_t + f_x m_0\right)\frac{h^2}{2} + \left(f_{tt} + 2 f_{tx} m_0 + f_{xx} m_0^2\right)\frac{h^3}{4}. \end{aligned}$$

The Taylor series expansion, on the other hand, gives

$$\begin{aligned} x\left(t_1\right) &= x_0 + \dot{x}\, h + \ddot{x}\,\frac{h^2}{2} + x^{(3)}\frac{h^3}{6} + \cdots \\ &= x_0 + f(t_0, x_0)\, h + \left(f_t + f_x \dot{x}\right)\frac{h^2}{2} \\ &\quad + \left(f_{tt} + 2 f_{tx}\dot{x} + f_{xx}\left(\dot{x}\right)^2 + f_x \ddot{x}\right)\frac{h^3}{6} + \cdots, \end{aligned}$$

where f_t, f_x, $\dot{x}$, etc. are all evaluated at $t = t_0$. Comparing the right-hand sides of the last two equations, we see that the solution obtained using the predictor-corrector method agrees with the Taylor series expansion of the true solution through terms of degree two.

3.7.5 Runge's Method

Runge's method is an improvement over the predictor-corrector method in the sense that it agrees with the Taylor series solution through terms of degree three. The method depends upon integration by means of Simpson's rule, which says that the area under the curve $x = g(t)$ between t_0 and $t_0 + h$ is given approximately by

$$\int_{t_0}^{t_0+h} g(t)dt \approx \frac{h}{6}\left(x_0 + 4x_1 + x_2\right), \tag{3.54}$$

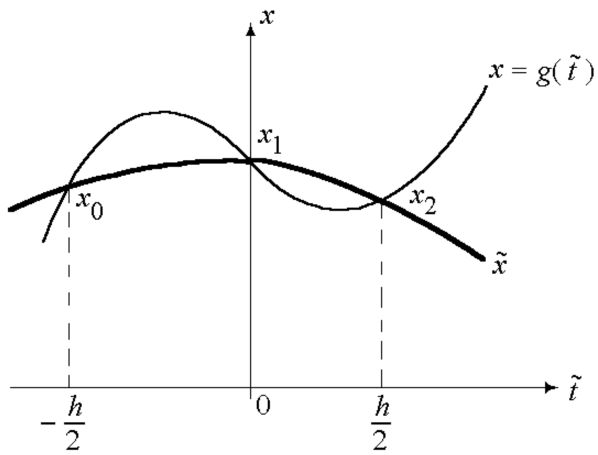


Figure 3.12: Integration using Simpson's rule.

where $x_0 = g(t_0)$, $x_1 = g(t_0 + h/2)$, and $x_2 = g(t_0 + h)$. The above formula is derived by passing a parabola through the three points (t_0, x_0), $(t_0 + h/2, x_1)$, and $(t_0 + h, x_2)$, and assuming that the area under the given curve is approximately equal to the area under the parabola as it is illustrated in Figure 3.12. To verify Simpson's formula, we translate the axes so that in the new coordinates $\left(\tilde{t}, \tilde{x}\right)$ we have $t = t_0 + h/2 + \tilde{t}$, $x = \tilde{x}$. In the new coordinates the three points become

$$(-h/2, x_0), \quad (0, x_1), \quad (h/2, x_2).$$

The parabola through the above points is described by the equation

$$\tilde{x} = a\tilde{t}^2 + b\tilde{t} + c.$$

Hence

$$\begin{aligned} x_0 &= a\frac{h^2}{4} - b\frac{h}{2} + c, \\ x_1 &= c, \\ x_2 &= a\frac{h^2}{4} + b\frac{h}{2} + c. \end{aligned}$$

Using the above three equations, we calculate

$$\begin{aligned} c &= x_1, \\ b &= \frac{x_2 - x_0}{h}, \\ a &= \frac{2}{h^2}(x_0 - 2x_1 + x_2). \end{aligned}$$

Then, the area under the parabola is

$$\begin{aligned} \int_{-h/2}^{h/2} \left(a\tilde{t}^2 + b\tilde{t} + c\right) d\tilde{t} &= \int_{-h/2}^{h/2} \left(\frac{2}{h^2}(x_0 - 2x_1 + x_2)\tilde{t}^2 + \frac{x_2 - x_0}{h}\tilde{t} + x_1\right) d\tilde{t} \\ &= \frac{h}{6}(x_0 + 4x_1 + x_2). \end{aligned}$$

If we are given the differential equation

$$\frac{dx}{dt} = f(t, x), \quad x(t_0) = x_0,$$

we integrate it to obtain

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(s, x(s))ds.$$

Then, we approximate the integral

$$\int_{t_0}^{t_1} f(s, x(s))ds$$

by

$$\frac{h}{6}(m_0 + 4m_1 + m_3),$$

where

$$\begin{aligned} h &= t_1 - t_0, \\ m_0 &= f(t_0, x_0), \\ m_1 &= f\left(t_0 + \frac{h}{2}, x_0 + m_0\frac{h}{2}\right), \\ m_2 &= f(t_0 + h, x_0 + m_0 h), \\ m_3 &= f(t_0 + h, x_0 + m_2 h). \end{aligned}$$

Thus, we may write

$$x(t_1) = x_0 + \Delta x,$$

where $\Delta x = mh$, and $m = \frac{1}{6}(m_0 + 4m_1 + m_3)$.

Table 3.2: Solving the differential equation (3.53) using Runge's method.

t_0	x_0	m_0	m_1	m_2	m_3	m	Δx
0	1	-1	-0.610	-0.375	-0.482	-0.654	-0.327
0.5	0.673	-0.446	-0.325	-0.248	-0.248	-0.332	-0.166
1.0	0.507						

Example 3.22 We use Runge's method to find $x(1)$, where the step length $h = 0.5$, for the differential equation (3.53). We found $x(1)$ using the predictor-corrector method in Example 3.21. The calculations for Runge's method are summarized in Table 3.2.

Runge's method can, of course, be used for systems of first-order differential equations. If, for example,

$$\begin{aligned}\frac{dx}{dt} &= f(t,x,y), \quad x(t_0) = x_0, \\ \frac{dy}{dt} &= g(t,x,y), \quad y(t_0) = y_0,\end{aligned}$$

then

$$\begin{aligned}x(t_1) &= x_0 + \Delta x, \\ y(t_1) &= y_0 + \Delta y,\end{aligned}$$

where

$$\begin{aligned}\Delta x &= \frac{h}{6}\left(m_0 + 4m_1 + m_3\right), \\ \Delta y &= \frac{h}{6}\left(n_0 + 4n_1 + n_3\right).\end{aligned}$$

In the above expressions,

$$
\begin{aligned}
m_0 &= f(t_0, x_0, y_0), \\
m_1 &= f\left(t_0 + h/2, x_0 + m_0\, h/2, y_0 + n_0\, h/2\right), \\
m_2 &= f\left(t_0 + h, x_0 + m_0 h, y_0 + n_0 h\right), \\
m_3 &= f\left(t_0 + h, x_0 + m_2 h, y_0 + n_2 h\right), \\
n_0 &= g(t_0, x_0, y_0), \\
n_1 &= g\left(t_0 + h/2, x_0 + m_0\, h/2, y_0 + n_0\, h/2\right), \\
n_2 &= g\left(t_0 + h, x_0 + m_0 h, y_0 + n_0 h\right), \\
n_3 &= g\left(t_0 + h, x_0 + m_2 h, y_0 + n_2 h\right).
\end{aligned}
$$

3.7.6 Runge-Kutta Method

This method is an improvement over the Runge method, and agrees with the Taylor series solution through terms of degree four—for this reason it is sometimes called the fourth-order Runge-Kutta algorithm. It is based on the use of the formula

$$x(t_1) = x(t_0) + \frac{h}{6}\left(m_0 + 2m_1 + 2m_2 + m_3\right), \tag{3.55}$$

where

$$
\begin{aligned}
m_0 &= f(t_0, x_0), \\
m_1 &= f\left(t_0 + h/2, x_0 + m_0\, h/2\right), \\
m_2 &= f\left(t_0 + h/2, x_0 + m_1\, h/2\right), \\
m_3 &= f\left(t_0 + h, x_0 + h\, m_2\right).
\end{aligned}
$$

Thus, we may write

$$x(t_1) = x_0 + \Delta x,$$

where $\Delta x = mh$, and $m = \frac{1}{6}\left(m_0 + 2m_1 + 2m_2 + m_3\right)$.

Example 3.23 We apply the Runge-Kutta method to find $x(1)$, where the step length $h = 0.5$, for the differential equation (3.53) that we used Examples 3.21 and 3.22. Our calculations for the Runge-Kutta method are summarized in Table 3.3.

Table 3.3: Solving the differential equation (3.53) using the Runge-Kutta method.

t_0	x_0	m_0	m_1	m_2	m_3	m	Δx
0	1	−1.0	−0.609	−0.668	−0.444	−0.666	−0.333
0.5	0.667	−0.444	−0.324	−0.328	−0.250	−0.333	−0.167
1.0	0.500						

The above problem has been worked by the predictor-corrector, Runge's, and the Runge-Kutta methods. This problem can be solved exactly by means of the Taylor series. The Taylor series solution is

$$x(t) = 1 - t + t^2 - t^3 + \cdots$$

The series converges to $1/(1+t)$ for $-1 < t < 1$. A check on the differential equation shows that

$$x(t) = \frac{1}{1+t}$$

is indeed a solution for all $t > -1$. Thus, $x(1) = 0.500$.

The algorithms presented above require only one input point, $x(t_k)$, at each step to compute $x(t_{k+1})$. Such algorithms are referred to as single-step algorithms. In the multi-step algorithms, a number of previous points are used, and the corresponding values of f, to evaluate $x(t_{k+1})$, that is, the multi-step algorithms reuse past information about the trajectory when evaluating its new point. In general, it may be difficult to decide which type of an algorithm is more efficient because the performance of a particular algorithm is problem dependent. Well known multi-step algorithms are: Adams-Bashforth, Adams-Moulton, and Gear. More information about the above mentioned algorithms can be found in [264, Section 4.1].

3.8 Notes

The system of differential equations (3.56) in Exercise 3.11 comes from Boyce and DiPrima [40, page 523]. Al-Khafaji and Tooley [6] provide an introductory treatment of numerical solutions of differential equations. For further discussion of solving differential equations using numerical methods and analyses of their computational efficiency, the reader is referred to Salvadori and Baron [292], Conte and de Boor [65, Chapter 8], or Parker and Chua [264].

3.9 Exercises

Exercise 3.1 Suppose that $\boldsymbol{x} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^\top$ is the representation of the given vector in the natural basis $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$. Find the representation of this vector in the basis $\{\boldsymbol{e}'_1, \boldsymbol{e}'_2, \boldsymbol{e}'_3\}$, where

$$\begin{aligned} \boldsymbol{e}'_1 &= \boldsymbol{e}_1 + \boldsymbol{e}_3 \\ \boldsymbol{e}'_2 &= \boldsymbol{e}_2 - \boldsymbol{e}_3 \\ \boldsymbol{e}'_3 &= \boldsymbol{e}_3. \end{aligned}$$

Exercise 3.2 Prove Theorem 3.1.

Exercise 3.3 Find a similarity transformation that transforms

$$\boldsymbol{A} = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 1 \\ 1 & 0 & \lambda & 0 \\ 0 & 0 & 1 & \lambda \end{bmatrix}$$

into its Jordan form. Do not fix λ; construct a transformation that works for any λ. Verify your calculations using MATLAB's Symbolic Toolbox.

Exercise 3.4 Calculate the solution for the system,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad \boldsymbol{x}(0) = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix},$$

where $u(t)$ is the unit step, that is, $u(t) = 1(t)$. Plot the system trajectories on the time interval $[0,\ 2]$ for $\boldsymbol{x}(0) = \boldsymbol{0}$.

Exercise 3.5 Construct an exact linear discrete state-space model for the system,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{x} \end{aligned}$$

The sampling period is $h = 0.2$ sec. Verify you calculations using MATLAB's function `c2d`. Find the transfer function of the discrete systems and its step response. Plot the step response of the discrete system on the time interval $[0,\ 2\text{ sec}]$.

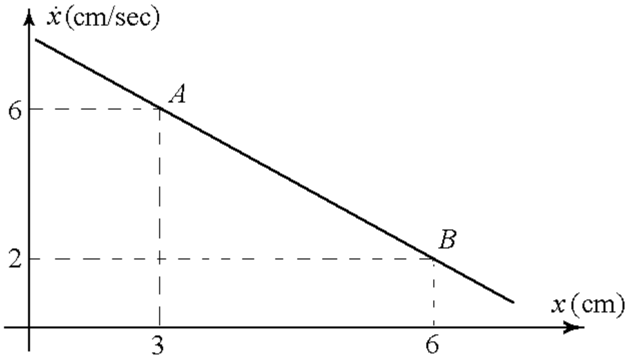


Figure 3.13: A trajectory for Exercise 3.9.

Exercise 3.6 Sketch a phase-plane portrait of the differential equation

$$\ddot{x} - \dot{x} + 2x + 2\text{sign}\,(2x + \dot{x}) = 0.$$

Exercise 3.7 For the system of differential equations

$$\left.\begin{array}{rcl} \dot{x}_1 &=& -\frac{7}{4}x_1 + \frac{1}{4}x_2 \\ \dot{x}_2 &=& \frac{3}{4}x_1 - \frac{5}{4}x_2 \end{array}\right\}$$

find the phase-plane asymptotes.

Exercise 3.8 For a dynamical system modeled by

$$\left.\begin{array}{rcl} \dot{x}_1 &=& x_2 \\ \dot{x}_2 &=& -x_1 \end{array}\right\}$$

find the time T it takes the system trajectory to move from the state $\boldsymbol{x}(0) = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ to the state $\boldsymbol{x}(T) = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$.

Exercise 3.9 Consider a trajectory in the state plane shown in Figure 3.9. Find the time it takes the representative point to move from A to B.

Exercise 3.10 Given the following initial value problem

$$\dot{x} = f(t, x), \quad x(t_0) = x_0.$$

(i) Derive the Euler formula for solving the above initial value problem using a Taylor series.

(ii) Make use of a Taylor series with a remainder to compute the local error formula assuming that the data at the n-th step are correct.

Exercise 3.11 Use Bendixson's theorem to investigate the existence of limit cycles for the system

$$\left.\begin{array}{rcl}\dot{x}_1 & = & x_1 + x_2 - x_1\left(x_1^2 + x_2^2\right) \\ \dot{x}_2 & = & -x_1 + x_2 - x_2\left(x_1^2 + x_2^2\right)\end{array}\right\} \tag{3.56}$$

Sketch a phase-plane portrait for this system.

Exercise 3.12 Use Bendixson's theorem to investigate the existence of limit cycles for the system

$$\left.\begin{array}{rcl}\dot{x}_1 & = & x_2 - x_1\left(x_1^2 + x_2^2 - c\right) \\ \dot{x}_2 & = & -x_1 - x_2\left(x_1^2 + x_2^2 - c\right)\end{array}\right\} \tag{3.57}$$

where c is a constant Sketch phase-plane portraits for this system for $c = -0.2$ and $c = 0.2$. The above nonlinear system was analyzed in [264, page 204].

Exercise 3.13 Verify that the Runge method agrees with the Taylor series solution through terms of degree three.

Exercise 3.14 Verify that the Runge-Kutta method agrees with the Taylor series solution through terms of degree four.

Chapter 4

Linearization

> Early amplifiers were made of crude materials that tended to disintegrate over use, causing the amp to "run away." Not only would an aging relay amplify the phone signal, it would mistakenly compound any tiny deviation from the range it expected until the mushrooming error filled and killed the system. What was needed was Heron's *regula*, a counter signal to rein in the chief signal, to dampen the effect of the perpetual recycling. Black came up with a *negative* feedback loop, which was designated negative in contrast to the snowballing positive loop of the amplifier. Conceptually, the electrical negative feedback loop is a toilet flusher or thermostat. This braking circuit keeps the amplifier honed in on a steady amplification in the same way a thermostat hones in on a steady temperature. But instead of metallic levers, a weak train of electrons talks to itself. [175, page 116]

The replacement of a nonlinear system model by its linear approximation is called *linearization*. The motivation for linearization is that the dynamical behavior of many nonlinear system models can be well approximated within some range of variables by linear system models. Then, we can use well developed techniques for analysis and synthesis of linear systems to analyze a nonlinear system at hand. However, the results of analysis of nonlinear systems using their linearized models should be carefully interpreted in order to avoid unacceptably large errors due to the approximation in the process of linearization. We present two different approaches to linear approximation of nonlinear systems.

4.1 Taylor Linearization

We begin our discussion of linear approximations of nonlinear systems by considering a nonlinear element with a state variable x and an output variable y that are related by the equation

$$y = h(x),$$

where the function $h : \mathbb{R} \to \mathbb{R}$ is continuously differentiable, that is, $h \in \mathcal{C}^1$. Let x_0 be a point of interest that we refer to as an *operating point*. We expand h into a Taylor series about the operating point x_0 to obtain

$$\begin{aligned} y &= h(x) \\ &= h(x_0) + \frac{dh(x_0)}{dx}(x - x_0) + \text{ higher order terms.} \end{aligned}$$

Linearization of $h(x)$ about the point x_0 consists of replacing h by the linear approximation

$$\begin{aligned} y &= h(x_0) + \frac{dh(x_0)}{dx}(x - x_0) \\ &= y_0 + \frac{dh(x_0)}{dx}(x - x_0), \end{aligned} \tag{4.1}$$

where $y_0 = h(x_0)$. Let $\Delta y = y - y_0$ and $\Delta x = x - x_0$. Then, we can represent (4.1) as

$$\Delta y = \frac{dh(x_0)}{dx}\Delta x.$$

Over a small range of Δx, the line (4.1) is a good approximation to the curve $y = h(x)$ in a neighborhood of the operating point x_0—see Figure 4.1 for an illustration of the above statement. We illustrate the process of linearization on the simple pendulum nonlinear model.

Example 4.1 Consider the simple pendulum shown in Figure 4.2. Let g be the gravity constant. The tangential force component, $mg \sin\theta$, acting on the mass m returns the pendulum to its rest, or equilibrium position. By Newton's second law

$$\tau = -mgl \sin\theta, \tag{4.2}$$

where $\tau = I\ddot{\theta}$ is the torque acting on the mass m and $I = ml^2$ is the moment of inertia of the mass m.

We are interested in linearizing the above pendulum model about its rest position, or equilibrium; call it θ_e. At such a point, $\tau(\theta_e) = 0$.

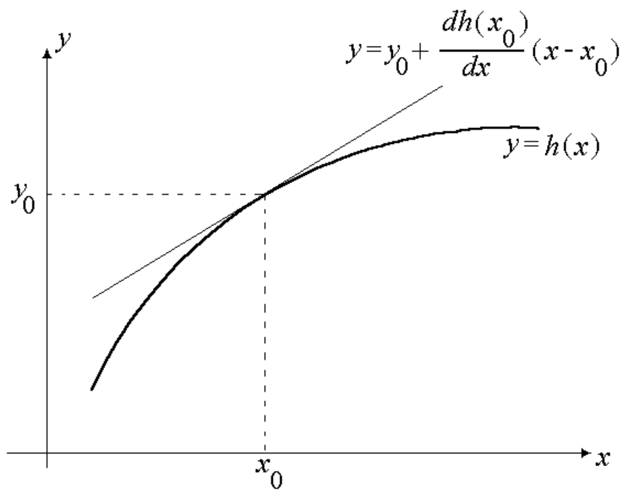


Figure 4.1: Linear approximation to a function $y = h(x)$.

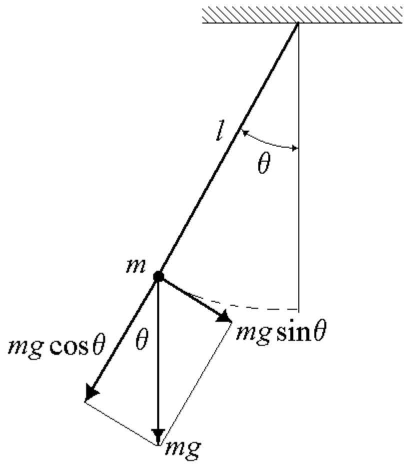


Figure 4.2: Forces acting on the simple pendulum.

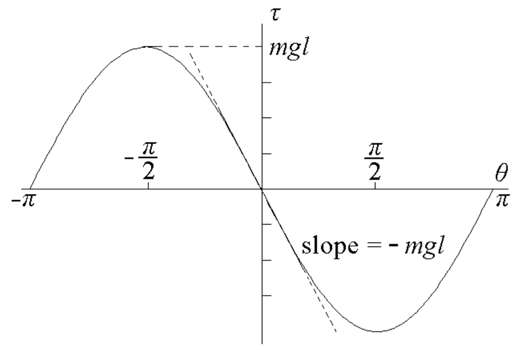


Figure 4.3: Linearizing the equation modeling the simple pendulum.

There are many equilibrium points of the pendulum model (4.2). We select $\theta_e = 0°$. Note that $\tau_e = \tau\,(0°) = 0$. Therefore,

$$\Delta\tau = \tau - \tau_e = \tau \text{ and } \Delta\theta = \theta - \theta_e = \theta.$$

We also have

$$\frac{d\tau}{d\theta}\,(0°) = -mgl\cos 0° = -mgl.$$

Hence, the linearized model, about the equilibrium position $\theta = 0°$ has the form

$$\tau = -mgl\theta.$$

Thus, for small angular displacements θ, the torque τ is proportional to the angular displacements. This approximation is quite accurate for $-\pi/4 \le \theta \le \pi/4$ as can be seen in Figure 4.3.

If $h : \mathbb{R}^n \to \mathbb{R}$, that is, $y = h(x_1, x_2, \ldots, x_n)$ which means that the dependent variable depends upon several variables, the same principle applies. Let

$$\boldsymbol{x}_0 = \begin{bmatrix} x_{10} & x_{20} & \cdots & x_{n0} \end{bmatrix}^\top$$

be the operating point. The Taylor series expansion of h about the operating point $\boldsymbol{x}_0$ yields

$$y - h(\boldsymbol{x}_0) = \nabla h(\boldsymbol{x}_0)^\top(\boldsymbol{x} - \boldsymbol{x}_0) + \text{higher order terms},$$

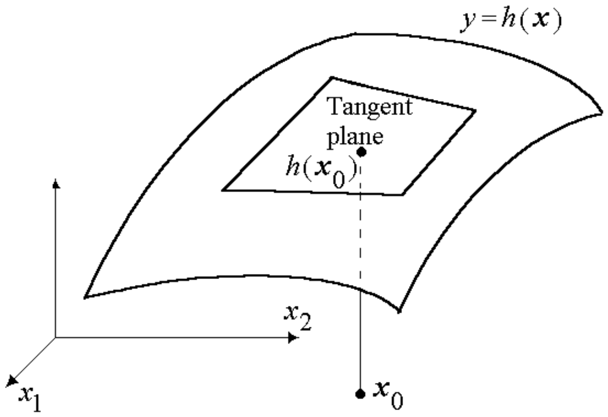


Figure 4.4: Tangent plane as a linear approximation for a function of two variables.

where

$$\nabla h(\boldsymbol{x}_0)^\top = \begin{bmatrix} \left.\frac{\partial h}{\partial x_1}\right|_{\boldsymbol{x}=\boldsymbol{x}_0} & \left.\frac{\partial h}{\partial x_2}\right|_{\boldsymbol{x}=\boldsymbol{x}_0} & \cdots & \left.\frac{\partial h}{\partial x_n}\right|_{\boldsymbol{x}=\boldsymbol{x}_0} \end{bmatrix}.$$

Geometrically, the linearization of h about $\boldsymbol{x}_0$ can be thought of as placing a tangent plane onto the nonlinear surface at the operating point $\boldsymbol{x}_0$ as illustrated in Figure 4.4.

4.2 Linearizing Differential Equations

4.2.1 Time-Invariant Unforced System

We consider a dynamical system modeled by a set of first-order differential equations,

$$\left.\begin{array}{rcl} \frac{dx_1}{dt} & = & f_1(x_1, x_2, \ldots, x_n) \\ \frac{dx_2}{dt} & = & f_2(x_1, x_2, \ldots, x_n) \\ & \vdots & \\ \frac{dx_n}{dt} & = & f_n(x_1, x_2, \ldots, x_n) \end{array}\right\}$$

We assume that the functions f_i, $i = 1, 2, \ldots, n$ are continuously differentiable. We represent the above system of differential equations in vector form as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}). \tag{4.3}$$

Note that the right-hand side of the above modeling equations does not depend explicitly on time. Thus (4.3) is a time-invariant system model. We discuss linearizing time-varying systems in Subsection 4.2.2.

We are interested in constructing a linear model approximation of the above nonlinear model in a vicinity of a point of interest, which is an equilibrium point of the model.

Definition 4.1 *A point $\boldsymbol{x}_e$ is an equilibrium state, or an equilibrium point, of the system modeled by* (4.3) *if it has the property that when the system starts at $\boldsymbol{x}_e$, it will stay at $\boldsymbol{x}_e$ for all future time. The equilibrium points are obtained by solving the algebraic equation*

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}.$$

$\diamond$

A nonlinear system may have a number of equilibrium states.

Example 4.2 The nonlinear system model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 - x_2 - x_1^2 \end{bmatrix},$$

has two equilibrium states

$$\boldsymbol{x}_e^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{x}_e^{(2)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

We are interested here in isolated equilibrium states; these are the equilibrium states that have no other equilibrium states in some neighborhood. In the above example, the equilibrium states are isolated. We can formally define an isolated equilibrium state using notion of the r-neighborhood.

Definition 4.2 *The r-neighborhood is a set of points*

$$\{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}_e\| < r\},$$

where $\|\cdot\|$ can be any p-norm on $\mathbb{R}^n$—see Section A.2 in the Appendix.

$\diamond$

Having the above definition, we formally define an isolated equilibrium state.

Definition 4.3 *An equilibrium state $\boldsymbol{x}_e$ is isolated if there is an $r > 0$ such that the r-neighborhood of $\boldsymbol{x}_e$ contains no equilibrium state other than $\boldsymbol{x}_e$.*

◇

We note that a linear system, $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$, can have only one isolated equilibrium state. Note also that if there two different points, $\boldsymbol{x}_e^{(1)}$ and $\boldsymbol{x}_e^{(2)}$ in $\mathbb{R}^n$, that are equilibrium states of the linear system, then so is any point on the line connecting these two points.

To proceed, we expand the right-hand side of (4.3) into a Taylor series about an isolated equilibrium state $\boldsymbol{x}_e$ to obtain

$$\frac{d}{dt}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}_e) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_e)\,\Delta\boldsymbol{x} + \text{higher-order terms}, \tag{4.4}$$

where $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_e)$ is the Jacobian matrix of $\boldsymbol{f}$ with respect $\boldsymbol{x}$, evaluated at the equilibrium state $\boldsymbol{x}_e$ and $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_e$. Observe that

$$\frac{d}{dt}\boldsymbol{x} = \frac{d}{dt}\boldsymbol{x}_e + \frac{d}{dt}\Delta\boldsymbol{x} = \frac{d}{dt}\Delta\boldsymbol{x}$$

because $\boldsymbol{x}_e$ is constant. Taking the above into account and the fact that $\boldsymbol{f}(\boldsymbol{x}_e) = \boldsymbol{0}$, we obtain a linear approximation of (4.4),

$$\frac{d}{dt}\Delta\boldsymbol{x} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_e)\,\Delta\boldsymbol{x}$$

Let

$$\boldsymbol{A} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_e).$$

Then, a linear model of (4.3) is written as

$$\frac{d}{dt}\Delta\boldsymbol{x} = \boldsymbol{A}\,\Delta\boldsymbol{x}. \tag{4.5}$$

Example 4.3 For the nonlinear dynamical system model,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 2 - x_1 x_2 \\ -4 + x_1^2 \end{bmatrix},$$

- find the equilibrium states;
- construct the corresponding linear models about the equilibrium states found above.

- To find equilibrium states, we solve the algebraic equations, $\dot{x}_1 = \dot{x}_2 = 0$, that is,

$$2 - x_1 x_2 = 0 \quad \text{and} \quad -4 + x_1^2 = 0.$$

  We have two equilibriums states,

$$\boldsymbol{x}_e^{(1)} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{x}_e^{(2)} = \begin{bmatrix} -2 \\ -1 \end{bmatrix}.$$

- A linear model has the form,

$$\frac{d}{dt}\Delta\boldsymbol{x} = \boldsymbol{A}\,\Delta\boldsymbol{x},$$

  where

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -x_2 & -x_1 \\ 2x_1 & 0 \end{bmatrix}$$

  evaluated at the equilibrium state about which we linearize the nonlinear system. We have two linear models:

  1. about $\boldsymbol{x}_e^{(1)}$,

$$\frac{d}{dt}\Delta\boldsymbol{x} = \begin{bmatrix} -1 & -2 \\ 4 & 0 \end{bmatrix} \Delta\boldsymbol{x},$$

  2. about $\boldsymbol{x}_e^{(2)}$,

$$\frac{d}{dt}\Delta\boldsymbol{x} = \begin{bmatrix} 1 & 2 \\ -4 & 0 \end{bmatrix} \Delta\boldsymbol{x}.$$

4.2.2 Time-Varying Forced System

We now consider a dynamical system modeled by a set of nonlinear differential equations

$$\left.\begin{array}{rcl} \frac{dx_1}{dt} & = & f_1(t, x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\ \frac{dx_2}{dt} & = & f_2(t, x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\ & \vdots & \\ \frac{dx_n}{dt} & = & f_n(t, x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \end{array}\right\}$$

The above set of differential equations can be represented in vector form as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}). \tag{4.6}$$

Let $\boldsymbol{u}_e = \begin{bmatrix} u_{1e} & u_{2e} & \cdots & u_{me} \end{bmatrix}^\top$ be a constant input that forces the system (4.6) to settle into a constant, equilibrium, state $\boldsymbol{x}_e = \begin{bmatrix} x_{1e} & x_{2e} & \cdots & x_{ne} \end{bmatrix}^\top$ at time t_0, that is, $\boldsymbol{u}_e$ and $\boldsymbol{x}_e$ satisfy

$$\boldsymbol{f}(t, \boldsymbol{x}_e, \boldsymbol{u}_e) = \boldsymbol{0} \quad \text{for all} \quad t \geq t_0.$$

We say that the pair $(\boldsymbol{x}_e, \boldsymbol{u}_e)$ is the equilibrium pair at t_0. We note that if the pair $(\boldsymbol{x}_e, \boldsymbol{u}_e)$ is an equilibrium pair at t_0, then it is also an equilibrium pair for all $\tilde{t}_0 \geq t_0$.

We perturb the equilibrium state by allowing

$$\boldsymbol{x} = \boldsymbol{x}_e + \Delta \boldsymbol{x}, \quad \boldsymbol{u} = \boldsymbol{u}_e + \Delta \boldsymbol{u}.$$

Taylor's expansion yields

$$\begin{aligned}
\frac{d}{dt}\boldsymbol{x} &= \boldsymbol{f}(t, \boldsymbol{x}_e + \Delta\boldsymbol{x}, \boldsymbol{u}_e + \Delta\boldsymbol{u}) \\
&= \boldsymbol{f}(t, \boldsymbol{x}_e, \boldsymbol{u}_e) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(t, \boldsymbol{x}_e, \boldsymbol{u}_e)\, \Delta\boldsymbol{x} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(t, \boldsymbol{x}_e, \boldsymbol{u}_e)\, \Delta\boldsymbol{u} + \text{higher order terms,}
\end{aligned}$$

where

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(t, \boldsymbol{x}_e, \boldsymbol{u}_e) = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_e \\ \boldsymbol{u}=\boldsymbol{u}_e}} \quad \text{and} \quad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(t, \boldsymbol{x}_e, \boldsymbol{u}_e) = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_e \\ \boldsymbol{u}=\boldsymbol{u}_e}}$$

are the Jacobian matrices of $\boldsymbol{f}$ with respect $\boldsymbol{x}$ and $\boldsymbol{u}$ respectively, evaluated at the equilibrium pair, $\begin{bmatrix} \boldsymbol{x}_e^\top & \boldsymbol{u}_e^\top \end{bmatrix}^\top$. Note that

$$\frac{d}{dt}\boldsymbol{x} = \frac{d}{dt}\boldsymbol{x}_e + \frac{d}{dt}\Delta\boldsymbol{x} = \frac{d}{dt}\Delta\boldsymbol{x}$$

because $\boldsymbol{x}_e$ is constant. Furthermore $\boldsymbol{f}(t, \boldsymbol{x}_e, \boldsymbol{u}_e) = \boldsymbol{0}$ for all $t \geq t_0$. Let

$$\boldsymbol{A}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(t, \boldsymbol{x}_e, \boldsymbol{u}_e) \quad \text{and} \quad \boldsymbol{B}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(t, \boldsymbol{x}_e, \boldsymbol{u}_e).$$

Neglecting higher order terms, we arrive at the linear approximation

$$\frac{d}{dt}\Delta\boldsymbol{x} = \boldsymbol{A}(t)\Delta\boldsymbol{x} + \boldsymbol{B}(t)\Delta\boldsymbol{u}.$$

Similarly, if the outputs of the nonlinear system model are of the form

$$\left.\begin{aligned}
y_1 &= h_1(t, x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
y_2 &= h_2(t, x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
&\vdots \\
y_p &= h_p(t, x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m)
\end{aligned}\right\}$$

or in vector notation

$$\boldsymbol{y} = \boldsymbol{h}(t, \boldsymbol{x}, \boldsymbol{u}),$$

then Taylor's series expansion can again be used to yield the linear approximation of the above output equations. Indeed, let

$$\boldsymbol{y} = \boldsymbol{y}_e + \Delta \boldsymbol{y},$$

where $\boldsymbol{y}_e = \boldsymbol{h}(t, \boldsymbol{x}_e, \boldsymbol{u}_e)$. Then we obtain

$$\Delta \boldsymbol{y} = \boldsymbol{C}(t) \Delta \boldsymbol{x} + \boldsymbol{D}(t) \Delta \boldsymbol{u},$$

where

$$\boldsymbol{C}(t) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}(t, \boldsymbol{x}_e, \boldsymbol{u}_e) = \left. \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial h_p}{\partial x_1} & \cdots & \frac{\partial h_p}{\partial x_n} \end{bmatrix} \right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_e \\ \boldsymbol{u}=\boldsymbol{u}_e}} \in \mathbb{R}^{p \times n},$$

and

$$\boldsymbol{D}(t) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}}(t, \boldsymbol{x}_e, \boldsymbol{u}_e) = \left. \begin{bmatrix} \frac{\partial h_1}{\partial u_1} & \cdots & \frac{\partial h_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial h_p}{\partial u_1} & \cdots & \frac{\partial h_p}{\partial u_m} \end{bmatrix} \right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_e \\ \boldsymbol{u}=\boldsymbol{u}_e}} \in \mathbb{R}^{p \times m}$$

are the Jacobian matrices of $\boldsymbol{h}$ with respect $\boldsymbol{x}$ and $\boldsymbol{u}$ respectively, evaluated at the equilibrium pair $\begin{bmatrix} \boldsymbol{x}_e^\top & \boldsymbol{u}_e^\top \end{bmatrix}^\top$.

Example 4.4 In this example, we linearize the model of Watt's governor that we derived in Section 2.4.1. Recall that the equations modeling the governor have the form:

$$\left. \begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & \frac{1}{2} N^2 x_3^2 \sin 2x_1 - \frac{g}{L} \sin x_1 - \frac{b}{mL} x_2 \\ \dot{x}_3 & = & \frac{\kappa}{I} \cos x_1 - \frac{\tau}{I} \end{array} \right\}$$

We linearize the above model about an equilibrium state of the form

$$\boldsymbol{x}_e = \begin{bmatrix} x_{1e} & 0 & x_{3e} \end{bmatrix}^\top$$

corresponding to the equilibrium control $u_e = \tau$.

Equating the right-hand sides of the above engine-governor equations to zero yields

$$\left. \begin{array}{lcl} x_{2e} & = & 0 \\ N^2 x_{3e}^2 & = & \frac{g}{L \cos x_{1e}} \\ \cos x_{1e} & = & \frac{\tau}{\kappa} \end{array} \right\}$$

The Jacobian matrix of the engine-governor model evaluated at the equilibrium point is

$$\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_e\\ u=u_e}} = \begin{bmatrix} 0 & 1 & 0 \\ N^2 x_{3e}^2 \cos 2x_{1e} - \frac{g}{L}\cos x_{1e} & -\frac{b}{mL} & N^2 x_{3e} \sin 2x_{1e} \\ -\frac{\kappa}{I}\sin x_{1e} & 0 & 0 \end{bmatrix}.$$

We use the following trigonometric identities,

$$\cos 2\alpha = \cos^2\alpha - \sin^2\alpha, \quad \sin 2\alpha = 2\sin\alpha\,\cos\alpha,$$

and the previously obtained relation, $N^2\, x_{3e}^2 = \frac{g}{L\cos x_{1e}}$, to eliminate $N^2\, x_{3e}^2$ from the Jacobian matrix. Performing needed manipulations, we obtain

$$\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_e\\ u=u_e}} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{g\sin^2 x_{1e}}{L\cos x_{1e}} & -\frac{b}{mL} & \frac{2g\sin x_{1e}}{Lx_{3e}} \\ -\frac{\kappa}{I}\sin x_{1e} & 0 & 0 \end{bmatrix}.$$

Hence, the linearized model of the engine-governor system can be represented as

$$\frac{d}{dt}\begin{bmatrix}\Delta x_1\\ \Delta x_2\\ \Delta x_3\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{g\sin^2 x_{1e}}{L\cos x_{1e}} & -\frac{b}{mL} & \frac{2g\sin x_{1e}}{Lx_{3e}} \\ -\frac{\kappa}{I}\sin x_{1e} & 0 & 0 \end{bmatrix}\begin{bmatrix}\Delta x_1\\ \Delta x_2\\ \Delta x_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ -\frac{1}{I}\end{bmatrix}\Delta u. \tag{4.7}$$

4.3 Describing Function Method

The describing function method allows us to apply familiar frequency domain techniques, used in the linear system analysis, to the analysis of a class of nonlinear dynamical systems. The method can be used to predict limit cycles in nonlinear systems. The describing function method can be viewed as a "harmonic linearization" of a nonlinear element. The method provides a "linear approximation" to the nonlinear element based on the assumption that the input to the nonlinear element is a sinusoid of known, constant amplitude. The fundamental harmonic of the element's output is compared with the input sinusoid to determine the steady-state amplitude and phase relation. This relation is the describing function for the nonlinear element. The describing function method is based on the Fourier series. In our review of Fourier series, we use notion of a scalar product over a function space that we discuss next. Our presentation of the scalar product in the context of spaces of functions follows that of Lang [203]. The reader familiar with the Fourier series may go directly to Subsection 4.3.3 where the describing function method analysis is presented.

4.3.1 Scalar Product of Functions

Let $\mathcal{C}([a, b])$ be the set of continuous functions on the interval $[a, b] \subset \mathbb{R}$.

Definition 4.4 *The scalar product of two real-valued functions $u, v \in \mathcal{C}([a, b])$ is*

$$\langle u, v \rangle = \int_a^b u(t)v(t)dt.$$

$\diamond$

Using simple properties of the integral, we can verify that the above scalar product satisfies the following conditions:

1. for all $u, v \in \mathcal{C}([a, b])$, we have $\langle u, v \rangle = \langle v, u \rangle$,
2. if u, v, w are elements of $\mathcal{C}([a, b])$, then $\langle u, v + w \rangle = \langle u, v \rangle + \langle u, w \rangle$,
3. if x is a number, then $\langle xu, v \rangle = x\langle u, v \rangle = \langle u, xv \rangle$,
4. for all $v \in \mathcal{C}([a, b])$, we have $\langle v, v \rangle \geq 0$, and $\langle v, v \rangle > 0$ if $v \neq 0$.

The functions $u, v \in \mathcal{C}([a, b])$ are said to be *orthogonal* if $\langle u, v \rangle = 0$. A set of functions from $\mathcal{C}([a, b])$ is said to be *mutually orthogonal* if each distinct pair of functions in the set is orthogonal.

Example 4.5 The functions $\psi_n(t) = \sin n\omega t$, $\phi_n(t) = \cos n\omega t$, $n = 1, 2, \ldots$, and $\phi_0(t) = 1$ form a mutually orthogonal set of functions on the interval $[t_0, t_0 + 2\pi/\omega]$. This can be verified by direct integration. For example, for positive m and n such that $m \neq n$, we have

$$\begin{aligned}
\langle \psi_m, \psi_n \rangle &= \int_{t_0}^{t_0+2\pi/\omega} \sin m\omega t \sin n\omega t \, dt \\
&= \frac{1}{2} \int_{t_0}^{t_0+2\pi/\omega} (\cos (m-n)\omega t - \cos (m+n)\omega t) \, dt \\
&= \frac{1}{2\omega} \left(\frac{\sin (m-n)\omega t}{m-n} - \frac{\sin (m+n)\omega t}{m+n} \right) \Bigg|_{t_0}^{t_0+2\pi\omega} \\
&= 0
\end{aligned} \tag{4.8}$$

Note that $m + n \neq 0$ because m and n are positive. Using similar computations we can show that for $m \neq n$

$$\langle \phi_m, \phi_n \rangle = 0, \tag{4.9}$$

and for all non-negative m and positive n,

$$\langle \phi_m, \psi_n \rangle = 0. \tag{4.10}$$

We define the *norm* of $v \in \mathcal{C}([a,b])$ to be

$$\|v\| = \sqrt{\langle v, v\rangle}.$$

Example 4.6 Consider $\psi_n(t) = \sin n\omega t$ on the interval $[t_0, t_0 + 2\pi/\omega]$. Then,

$$\begin{aligned}
\|\psi_n\|^2 &= \int_{t_0}^{t_0+2\pi/\omega} \sin n\omega t \sin n\omega t \, dt \\
&= \int_{t_0}^{t_0+2\pi/\omega} \sin^2 n\omega t \, dt \\
&= \frac{1}{2}\int_{t_0}^{t_0+2\pi/\omega} (1 - \cos 2n\omega t) \; dt \\
&= \frac{1}{2}\left(t - \frac{\sin 2n\omega t}{2n\omega}\right)\Bigg|_{t_0}^{t_0+2\pi/\omega} \\
&= \frac{\pi}{\omega}. \qquad (4.11)
\end{aligned}$$

Similarly,

$$\|\phi_n\|^2 = \frac{\pi}{\omega}, \quad n = 1, 2\ldots, \qquad (4.12)$$

and

$$\|\phi_0\|^2 = \frac{2\pi}{\omega}. \qquad (4.13)$$

So far we discussed only continuous functions. In many applications we have to work with more general functions, specifically with piecewise continuous functions.

Definition 4.5 *A function f is said to be* piecewise continuous *on an interval $[a,b]$ if the interval can be partitioned by a finite number of points, say t_i, so that*

1. $a = t_0 < t_1 < \ldots < t_n = b$;
2. *the function f is continuous on each open subinterval (t_{i-1}, t_i),*
3. *the function f approaches a finite limit as the end points of each subinterval are approached from within the subinterval, that is, the limits*

$$\lim_{\substack{h\to 0\\ h>0}} f(t_i - h) \quad \textit{and} \quad \lim_{\substack{h\to 0\\ h>0}} f(t_i + h)$$

both exist.

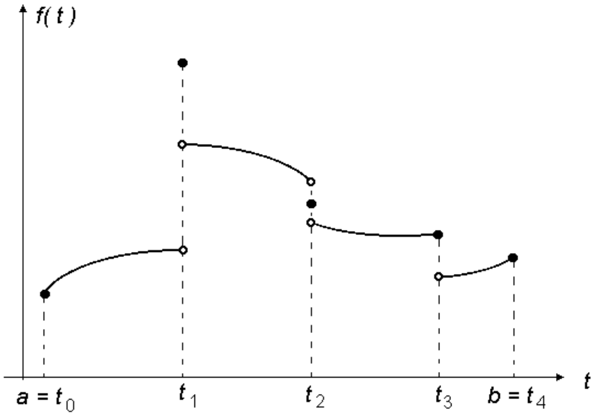


Figure 4.5: A graph of a piecewise continuous function.

$\diamond$

The graph of a piecewise continuous function is shown in Figure 4.5. In further discussion, we assume that two piecewise functions f and g are equal if they have the same values at the points where they are continuous. Thus, the functions shown in Figure 4.6 are considered to be equal, though they differ at the points of discontinuity.

Theorem 4.1 *Let V be the space of functions that are piecewise continuous on the interval $[a, b]$. Let $f \in V$. Then, $\|f\| = 0$ if and only if $f(x) = 0$ for all but finite number of points x in the interval $[a, b]$.*

Proof ($\Leftarrow$) It is clear that if $f(x) = 0$ except for a finite number of $x \in [a, b]$, then

$$\|f\|^2 = \int_a^b f^2(x)dx = 0.$$

($\Rightarrow$) Suppose f is piecewise continuous on $[a, b]$, and let $a = t_0 < t_1 < \ldots < t_n = b$ be a partition of the interval $[a, b]$ such that f is continuous on each subinterval $[t_{i-1}, t_i]$ except possibly at the end points. Suppose that $\|f\| = 0$. Then, also $\|f\|^2 = \langle f, f\rangle = 0$. The above means that

$$\int_a^b f^2(x)dx = 0,$$

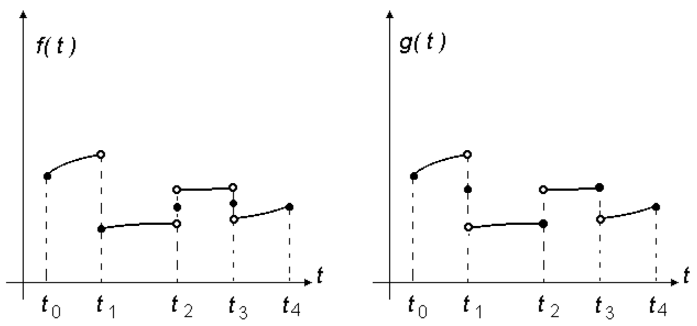


Figure 4.6: Two piecewise continuous functions that we consider to be equal.

where the integral is the sum of the integrals over the intervals $[t_{i-1}, t_i]$, that is,

$$\int_a^b f^2(x)dx = \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} f^2(x)dx = 0.$$

Each integral satisfies

$$\int_{t_{i-1}}^{t_i} f^2(x)dx \geq 0.$$

Hence each such integral must be equal to 0. The function f is continuous on (t_{i-1}, t_i). Therefore,

$$f^2(x) = 0 \quad \text{for} \quad t_{i-1} < x < t_i,$$

which means that

$$f(x) = 0 \quad \text{for} \quad t_{i-1} < x < t_i.$$

Hence, $f(x) = 0$ except at a finite number of points. The proof is complete.

□

The above theorem implies that the scalar product on the space V of piecewise continuous functions is positive definite. In other words, for a piecewise continuous function $f \in V$, we have $\langle f, f \rangle \geq 0$, and $\langle f, f \rangle = 0$ if and only if $f = 0$ for all but a finite number of points in the interval $[a, b]$.

4.3.2 Fourier Series

We begin with the definition of a periodic function.

Definition 4.6 *A function $f : \mathbb{R} \to \mathbb{R}$ is said to be periodic with the period T if the domain of f contains $t + T$ whenever it contains t, and if*

$$f(t+T) = f(t)$$

for every value of t. The smallest positive value of T for which $f(t+T) = f(t)$ is called the fundamental period of f.

$\diamond$

It follows from the above definition that if T is a period of f, then $2T$ is also a period, and so is any integral multiple of T.

A periodic function, satisfying certain assumptions spelled out later, may be represented by the series

$$f(t) = A_0 + \sum_{k=1}^{\infty} \left(A_k \cos k\omega t + B_k \sin k\omega t \right), \tag{4.14}$$

where $\omega = 2\pi/T$, with T being the fundamental period of f. On the set of points where the series (4.14) converges, it defines f, whose value at each point is the sum of the series for that value of t, and we say that the series (4.14) is the Fourier series for f.

Suppose that the series (4.14) converges. We use the results of the previous subsection to find expressions for the coefficients A_k and B_k. We first compute A_k coefficients. Multiplying (4.14) by $\phi_m = \cos m\omega t$, where $m > 0$ is a fixed positive integer, and then integrating both sides with respect to t from t_0 to $t_0 + 2\pi/\omega$ yields

$$\langle \phi_m, f \rangle = A_0 \langle \phi_m, \phi_0 \rangle + \sum_{k=1}^{\infty} A_k \langle \phi_m, \phi_k \rangle + \sum_{k=1}^{\infty} B_k \langle \phi_m, \psi_k \rangle, \tag{4.15}$$

where $\phi_0 = 1$. It follows from (4.8), (4.9), and (4.10) that the only nonvanishing term on the right of (4.15) is the one for which $k = m$ in the first summation. Using (4.12), we obtain

$$\begin{aligned} A_m &= \frac{\langle \phi_m, f \rangle}{\langle \phi_m, \phi_m \rangle} \\ &= \frac{\omega}{\pi} \int_{t_0}^{t_0 + 2\pi/\omega} f(t) \cos m\omega t \, dt, \quad m = 1, 2, \ldots \end{aligned} \tag{4.16}$$

To obtain A_0, we integrate (4.14) and using (4.13), we get

$$A_0 = \frac{\langle \phi_0, f \rangle}{\langle \phi_0, \phi_0 \rangle} = \frac{\omega}{2\pi} \int_{t_0}^{t_0 + 2\pi/\omega} f(t) dt. \tag{4.17}$$

An expression for B_m may be obtained by multiplying (4.14) by $\sin m\omega t$ and then integrating term by term from t_0 to $t_0 + 2\pi/\omega$. Using the orthogonality relations (4.8)–(4.10) yields

$$B_m = \frac{\langle \psi_m, f\rangle}{\langle \psi_m, \psi_m\rangle} = \frac{\omega}{\pi}\int_{t_0}^{t_0+2\pi/\omega} f(t)\sin m\omega t\,dt, \quad m = 1, 2, \ldots \tag{4.18}$$

Suppose now that a function f is given that is periodic with period $2\pi/\omega$ and integrable on the interval $[t_0, t_0+2\pi/\omega]$. Then, coefficients A_k and B_k can be computed using (4.16)–(4.18) and series of the form (4.14) can be formally constructed. However, we do not know whether this series converges for each value of t, and if so, whether its sum is $f(t)$. The answer is given by the theorem that we state after explaining the notation used. We denote the limit of $f(t)$ as t approaches c from the right by $f(c+)$, that is,

$$f(c+) = \lim_{t\to c+} f(t).$$

Similarly, $f(c-) = \lim_{t\to c-} f(t)$ denotes the limit of $f(t)$ as $t \to c$ from the left. The mean value of the right and left hand limits at the point c is

$$\left(f(c+) + f(c-)\right)/2.$$

Note that at any point c where f is continuous, we have

$$f(c+) = f(c-) = f(c).$$

Theorem 4.2 *Assume that f and its derivative are piecewise continuous on the interval $[t_0, t_0 + 2\pi/\omega]$. Furthermore, suppose that f is defined outside the interval $[t_0, t_0 + 2\pi/\omega]$ so that it is periodic with period $2\pi/\omega$. Then, f has a Fourier series of the form (4.14) whose coefficients are given by (4.16)–(4.18). The Fourier series converges to $\left(f(t+) + f(t-)\right)/2$ at all points of the interval $[t_0, t_0 + 2\pi/\omega]$.*

With the above background of the Fourier series we are ready to discuss the describing function method for analyzing a class of nonlinear control systems.

4.3.3 Describing Function in the Analysis of Nonlinear Systems

The describing function method can be considered as an extension of the Nyquist stability criterion to nonlinear systems. In our further discussion, we sometimes use a double box to represent a nonlinear element as illustrated in Figure 4.7.

Consider a feedback system that contains a nonlinear element as shown in Figure 4.8. Assume that the input to the nonlinear element is a sinusoidal signal $x(t) = X \sin \omega t$, where

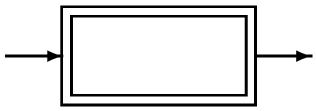


Figure 4.7: A symbol for a nonlinear element.

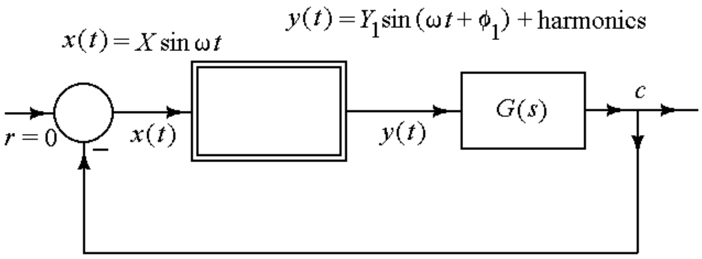


Figure 4.8: A nonlinear feedback system with a nonlinear element in the forward path.

X is the amplitude of the input sinusoid. The output of the nonlinear element is, in general, not sinusoidal in response to a sinusoidal input. Suppose that the nonlinear element output is periodic with the same period as its input and that it may be expanded in the Fourier series,

$$\begin{aligned} y(t) &= A_0 + \sum_{k=1}^{\infty} \left(A_k \cos k\omega t + B_k \sin k\omega t \right) \\ &= A_0 + \sum_{k=1}^{\infty} Y_k \sin \left(k\omega t + \phi_k \right). \end{aligned} \tag{4.19}$$

Thus $y(t)$ consists of a fundamental component

$$Y_1 \sin \left(\omega t + \phi_1 \right),$$

together with a mean level A_0 and harmonic components at frequencies $2\omega, 3\omega, \ldots$. The describing function method relies on the assumption that only the fundamental harmonic is significant. This assumption is called the *filtering hypothesis*. The filtering hypothesis can be expressed as

$$|G(j\omega)| \gg |G(kj\omega)| \quad \text{for} \quad k = 2, 3, \ldots,$$

which requires that the linear element, modeled by the transfer function $G(s)$, be a low-pass filter. The filtering hypothesis is often valid because many, if not most, control systems are low-pass filters. This results in higher harmonics being more attenuated compared with the fundamental harmonic component. The nonlinear element is then approximately modeled by its describing function

$$N(X) = \frac{Y_1}{X} \angle \phi_1 = \frac{\text{fundamental component of output}}{\text{input}}. \tag{4.20}$$

The describing function is, in general, a complex-valued function of the amplitude X of the input sinusoid.

The stability of the nonlinear closed-loop system depicted in Figure 4.8 can be analyzed using Nyquist's criterion applied to the associated linear system shown in Figure 4.9. The associated linear closed-loop system characteristic equation is

$$1 + N(X)G(s) = 0. \tag{4.21}$$

In the $G(j\omega)$ complex plane, the locus traced out by $-1/N$ as X is varied, is being considered as a generalization of the $(-1, j0)$ point in the linear case. The stability of the nonlinear system is determined by computing the encirclements of the $-1/N$ locus by the $G(j\omega)$ plot.

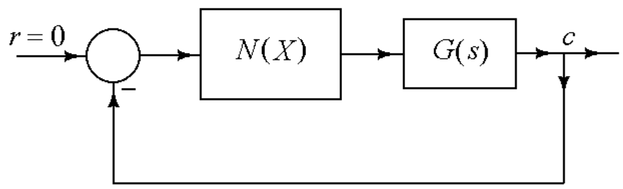


Figure 4.9: Associated linear system used in the describing function analysis of the nonlinear system shown in Figure 4.8.

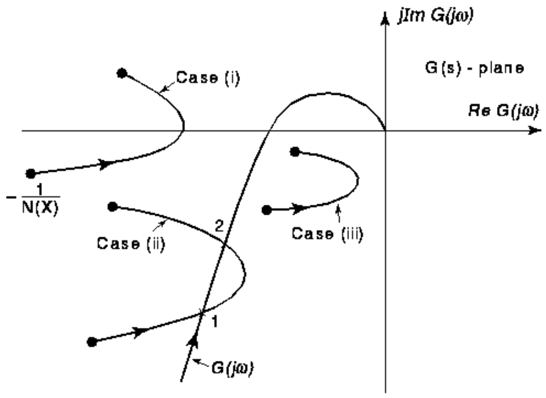


Figure 4.10: Stability analysis using describing function.

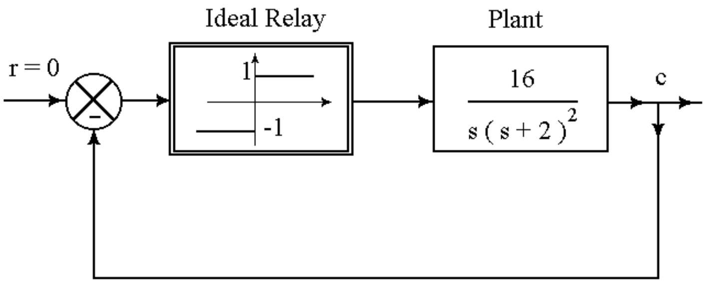


Figure 4.11: Nonlinear feedback control system analyzed in Example 4.7.

We might have situations shown in Figure 4.10. In case (i), the $-1/N$ locus is not encircled by the $G(j\omega)$ plot. Assuming that the poles of $G(s)$ are in the left-half complex plane, the closed-loop system is asymptotically stable. For a limit cycle to exist, we must have an intersection between the polar plot of $G(j\omega)$ and the $-1/N$ locus. We have two intersections between $-1/N$ and $G(j\omega)$ in case (ii). At intersection point 1, any increase in X will cause the operating point to be encircled and consequently X to grow. If there is a decrease in X, then the oscillations will decay. Thus, point 1 represents an unstable limit cycle. At intersection point 2, an increase in X would shift the operating point outside the $G(j\omega)$ plot resulting in the decaying oscillations. Point 2 represents a stable limit cycle. In case (iii), any oscillation will grow without limit.

Example 4.7 Consider the nonlinear control system shown in Figure 4.11 that was analyzed in [278, pages 339–341]. We first derive the describing function of the ideal relay. When the input of the ideal relay is greater than zero, its output is M. When the relay's input is less than zero, its output is $-M$. Thus, the ideal relay can be described as

$$y = \begin{cases} M & \text{for} \quad x > 0 \\ -M & \text{for} \quad x < 0. \end{cases} \tag{4.22}$$

We assume that the input to the relay is a sinusoid, $x(t) = X \sin \omega t$, drawn vertically down from the relay characteristic in Figure 4.12. The output $y(t)$ of the relay is a rectangular

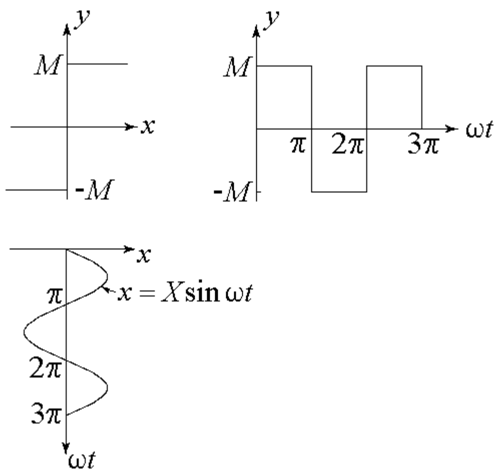


Figure 4.12: Input and output waveforms of an ideal relay.

wave. It is drawn horizontally to the right of the relay characteristic in Figure 4.12. This rectangular wave is an odd function. For an odd function, we have $A_k = 0$, $k = 0, 1, \ldots$, in its Fourier series. Thus, the Fourier series of the relay's output has the form

$$y(t) = \sum_{k=1}^{\infty} B_k \sin k\omega t.$$

The fundamental harmonic of $y(t)$ is

$$B_1 \sin \omega t = Y_1 \sin \omega t,$$

where

$$\begin{aligned} B_1 &= \frac{\omega}{\pi} \int_0^{2\pi/\omega} y(t) \sin \omega t \, dt \\ &= \frac{2\omega}{\pi} \int_0^{\pi/\omega} M \sin \omega t \, dt \\ &= \frac{2M\omega}{\pi} \int_0^{\pi/\omega} \sin \omega t \, dt \\ &= -\frac{2M}{\pi} \cos(\omega t) \Big|_0^{\pi/\omega} \\ &= \frac{4M}{\pi}. \end{aligned} \tag{4.23}$$

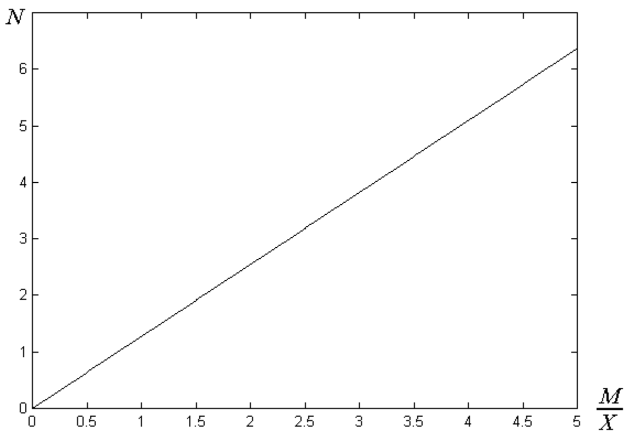


Figure 4.13: A plot of the describing function for the ideal relay.

Hence, the describing function of the ideal relay is

$$N(X) = \frac{Y_1}{X}\angle 0^\circ = \frac{4M}{\pi X}.$$

A normalized plot of the describing function of the ideal relay nonlinearity is depicted in Figure 4.13. In our example, $M = 1$ and thus the describing function of the relay is

$$N(X) = \frac{4}{\pi X}.$$

The sinusoidal transfer function of the linear plant model is

$$G(j\omega) = \frac{16}{j\omega(j\omega+2)^2} = \frac{-64\omega + j16(\omega^2-4)}{\omega(4+\omega^2)^2}.$$

At $\omega = 2$ rad/sec the polar plot of $G(j\omega)$ intersects the plot of $-1/N$, see Figure 4.14, to form a stable limit cycle. The amplitude of the limit cycle is obtained from the relation, $G(j2) = -1/N$, that is,

$$\frac{-64}{(4+4)^2} = -\frac{\pi X}{4}.$$

Hence, the amplitude of the limit cycle is

$$X = \frac{4}{\pi},$$

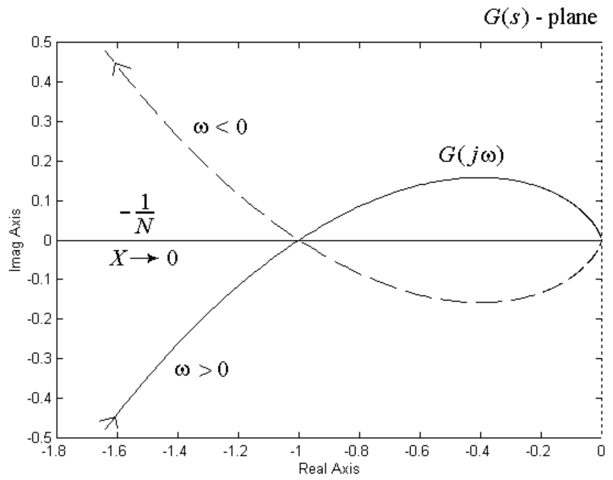


Figure 4.14: Plots of $G(j\omega)$ and $-1/N$ of Example 4.7.

while the amplitude of the fundamental harmonic of the plant output, c, is

$$\frac{4}{\pi}\,|G(j2)| = \frac{4}{\pi}.$$

Note that the filtering hypothesis is satisfied in this example. Indeed, the Fourier series of the output of the relay is

$$\frac{4}{\pi}\left(\sin\omega t + \frac{1}{3}\sin 3\omega t + \frac{1}{5}\sin 5\omega t\right).$$

Thus, for example, the amplitude of the third harmonic of the plant output is

$$\frac{4}{3\pi}\,|G(j6)| = \frac{1}{45}\frac{4}{\pi}.$$

The higher harmonics are even more attenuated compared with the fundamental harmonic.

4.4 Notes

For an in-depth treatment of the describing function method, we refer to Graham and McRuer [118] and Chapters 6 and 7 of Hsu and Meyer [145]. Graham and McRuer [118] make an interesting remark about the origins of the describing function method. They note on page 92 of their 1961 book: "While the notion of representing a nonlinearity by an "equivalent" linear element is quite old and has been used by many writers, the first instance of a major systematic exploitation of the technique was probably by N. M. Kryloff and N. N. Bogoliuboff, *Introduction to Non-linear Mechanics* (a free translation of the Russian edition of 1937, by S. Lefschetz), Princeton University Press, Princeton, N. J., 1943. See also N. Minorsky, *Introduction to Non-linear Mechanics*, J. W. Edwards, Ann Arbor, Mich., 1947." The so-called sinusoidal describing function that we analyzed in the previous section, was developed almost simultaneously in several different countries during and just after World War II. The first to introduce the method in the open literature appears to be A. Tustin of England who published his paper in the Journal of the IEE in 1947.

The behavior of numerous dynamical systems is often modeled by a set of linear, first-order, differential equations. Frequently, however, a linear model is a result of linearization of a nonlinear model. Linear models seem to dominate the controls' literature. Yet nature is nonlinear. "All physical systems are nonlinear and have time-varying parameters in some degree." [118, page 1]. How can one reconcile this paradox? Graham and McRuer [118, page 1] have the following answer: "Where the effect of the nonlinearity is very small, or if the parameters vary only slowly with time, linear constant-parameter methods of analysis can be applied to give an approximate answer which is adequate for engineering purposes. The analysis and synthesis of physical systems, predicated on the study of linear constant-parameter mathematical models, has been, in fact, an outstandingly successful enterprise. A system represented as linear, however, is to some extent a mathematical abstraction that can never be encountered in a real world. Either by design or because of nature's ways it is often true that experimental facts do not, or would not, correspond with any prediction of linear constant-parameter theory. In this case nonlinear or time-varying-parameter theory is essential to the description and understanding of physical phenomena."

4.5 Exercises

Exercise 4.1 Consider the nonlinear mass-spring-damper system depicted in Figure 4.15. The spring coefficient $K = 0.5x^2$ N/m, the damping coefficient $B = 0.1\dot{x}^2$ N·sec/m, and $M = 1$ kg.

(i) Write the second-order differential equation describing the motion of the system.

(ii) Choose $x_1 = x$ and $x_2 = \dot{x}$. Represent the equation of motion derived in (i) in state-

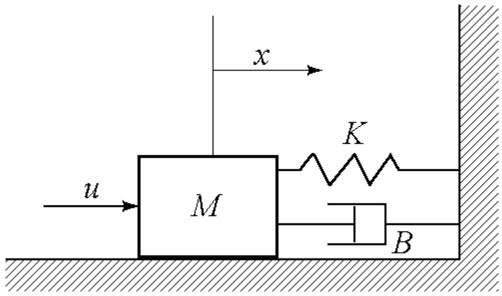


Figure 4.15: The mass-spring-damper system of Exercise 4.1.

space format.

(iii) For the model from (ii), find the equilibrium state corresponding to $u = 0$. Then, linearize the model about the obtained equilibrium point.

Exercise 4.2 (based on [328, pages 29–31]) A cyclotron, whose operation is illustrated in Figure 4.16, accelerates charged particles into high energies so that they can be used in atom-smashing experiments. The operation of the cyclotron can be described as follows. Positive ions enter the cyclotron from the ion source, S, and are available to be accelerated. An electric oscillator establishes an accelerating potential difference across the gap of the two D-shaped objects, called dees. An emerging ion from the ion source, S, finds the dee that is negative. It will accelerate toward this dee and will enter it. Once inside the dee, the ion is screened from electric fields by the metal walls of the dees. The dees are immersed in a magnetic field so that the ion's trajectory bends in a circular path. The radius of the path depends on the ion's velocity. After some time, the ion emerges from the dee on the other side of the ion source. In the meantime, the accelerating potential changes its sign. Thus the ion again faces a negative dee. The ion further accelerates and describes a semicircle of larger radius. This process goes on until the ion reaches the outer edge of one dee, where it is pulled out of the cyclotron by a negatively charged deflector plate. The moving ion is acted upon by a deflecting force of magnitude

$$q(v + \Delta v)(B + \Delta B),$$

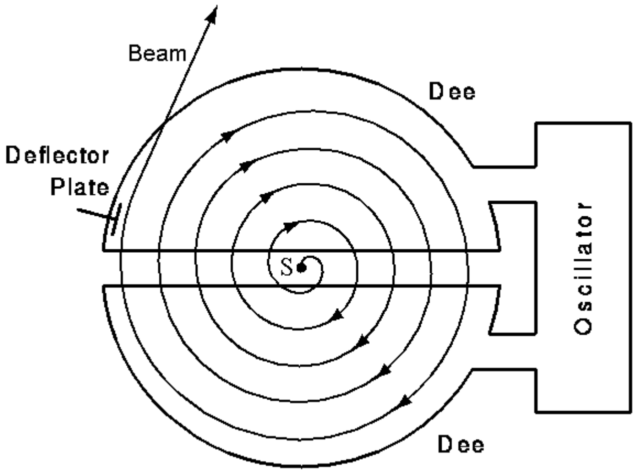


Figure 4.16: Schematic of a cyclotron of Exercise 4.2.

where q is the charge of the ion, v is the ion's speed, Δv is the increase in the ion's speed, B is the magnetic field acting on the ion. The nominal radius of the ion's orbit is r. The change in the magnetic field as the result of a change in the orbit of the ion is ΔB. As the ion speed increases, the relativistic mass m increases by Δm. The ion moving in a circular path undergoes centrifugal force of magnitude

$$\frac{(m+\Delta m)(v+\Delta v)^2}{r+\Delta r}.$$

By Newton's second law, the equation describing the motion of the ion is

$$(m+\Delta m)\frac{d^2}{dt^2}(r+\Delta r) = \frac{(m+\Delta m)(v+\Delta v)^2}{r+\Delta r} - q(v+\Delta v)(B+\Delta B).$$

Linearize the above equation. The motion of the ion along a nominal path is described by

$$m\frac{d^2 r}{dt^2} = \frac{mv^2}{r} - qvB.$$

Find the transfer function relating the change in radius of motion, Δr, of the ion, and the change in the momentum of the ion

$$\Delta p = m\Delta v + \Delta m v.$$

In other words, find the transfer function $\mathcal{L}(\Delta r)/\mathcal{L}(\Delta p)$, where $\mathcal{L}$ is the Laplace transform operator.

Exercise 4.3 For the nonlinear model of the θ-r robot manipulator in Example 2.2, on page 67, find the equilibrium point

$$(\boldsymbol{z}_e, \boldsymbol{u}_e) = \left(\begin{bmatrix} \pi/4 & 0 & 2 & 0 \end{bmatrix}^\top, \boldsymbol{u}_e \right)$$

and linearize the non-linear model about the obtained equilibrium. You may wish to use MATLAB's `jacobian` and `subs` functions to perform linearization.

Exercise 4.4 Linearize the permanent-magnet stepper motor model, analyzed in Subsection 2.4.3, about $x_{1eq} = \theta_{eq} = \text{constant}$.

Exercise 4.5 For the nonlinear model of a double-link robot manipulator in Subsection 2.4.6, find the equilibrium point

$$(\boldsymbol{x}_e, \boldsymbol{u}_e) = \left(\begin{bmatrix} 135^\circ & 0 & 180^\circ & 0 \end{bmatrix}^\top, \boldsymbol{u}_e \right)$$

and linearize the non-linear model about the obtained equilibrium.

Exercise 4.6 Let $v_1, v_2, \ldots, v_n$ be non-zero, mutually orthogonal elements of the space V and let v be an arbitrary element of V.

(i) Find the component of v along v_j.

(ii) Let c_j denote the component of v along v_j. Show that the element

$$v - c_1 v_1 - \cdots c_n v_n$$

is orthogonal to $v_1, v_2, \ldots, v_n$.

(iii) Let c_j be the component of $v \in V$ along v_j. Show that

$$\sum_{i=1}^{n} c_i^2 \leq \|v\|^2.$$

The above relation is called the Bessel inequality.

Exercise 4.7 Consider characteristics of two nonlinear elements that are shown in Figure 4.17. Let

$$N_S(x) = \frac{2}{\pi} \left[\arcsin \frac{1}{x} + \frac{1}{x} \cos \left(\arcsin \frac{1}{x} \right) \right]. \tag{4.24}$$

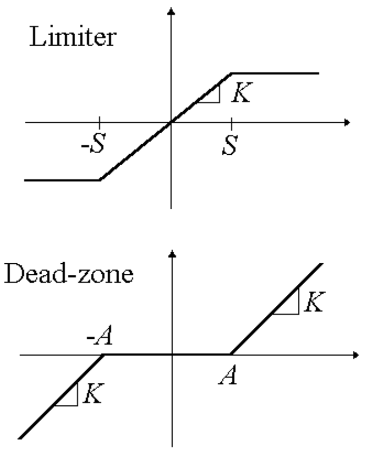


Figure 4.17: Characteristics of a limiter and dead zone nonlinear elements of Exercise 4.7.

Then, the describing function of the limiter is

$$N_l = \begin{cases} K, & M \leq S \\ KN_S\left(\frac{M}{S}\right), & M > S. \end{cases}$$

The describing function of the dead zone is

$$N_{d-z} = \begin{cases} 0, & M \leq A \\ K\left[1 - N_S\left(\frac{M}{A}\right)\right], & M > A. \end{cases}$$

Derive the describing function of the nonlinear element whose characteristic is shown in Figure 4.18.

Exercise 4.8 The describing function of the nonlinear element shown in Figure 4.19, is

$$N_1 = \begin{cases} 0, & M \leq A \\ K\left[1 - N_S\left(\frac{M}{A}\right)\right] + \frac{4B}{\pi M}\sqrt{1 - \left(\frac{A}{M}\right)^2}, & M > A, \end{cases}$$

where $N_S(x)$ is given by (4.24). The describing function of the dead-zone nonlinearity, is given in Exercise 4.7 and its characteristic is shown in Figure 4.17. Derive the describing function of the ideal relay with dead zone whose characteristic is shown in Figure 4.20.

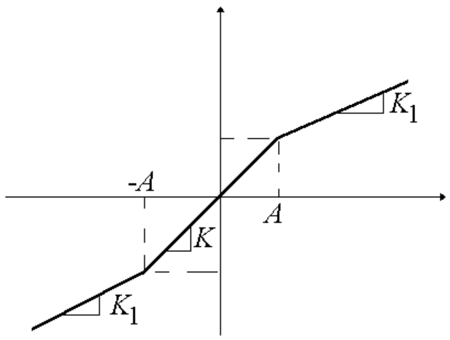


Figure 4.18: Characteristic of a nonlinear element of Exercise 4.7.

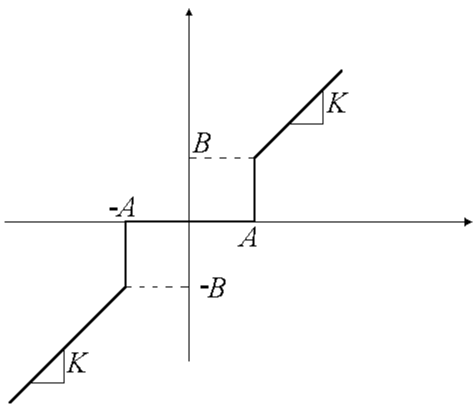


Figure 4.19: Nonlinear element with the describing function N_1 of Exercise 4.8.

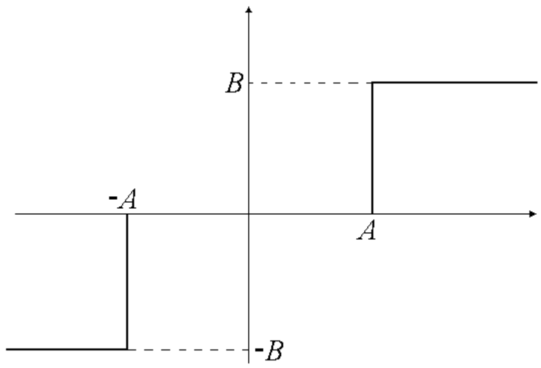


Figure 4.20: Characteristic of the ideal relay with dead zone.

Chapter 5

Linear Time-Invariant (LTI) Systems

> In bear country, wear bells to avoid surprising bruins. Don't approach a bear, but don't run away from one either. Movement may incite a charge, and a bear can outrun you.
> If a bear approaches you, talk to it calmly in a low, firm voice. Slowly back away until the bear is out of sight. If you climb a tree, get at least 12 feet up. If a bear senses that its food may be stolen or that its cub is in danger, it may charge. If this happens, drop to the ground and assume a fetal position, protecting your head with your arms, and play dead. Keep your pack on; it may serve as armor. [281, pages 18–19]

We begin our discussion of linear system theory with a description of the concept of linearity by Mitchell J. Feigenbaum [270, page 3], who is a researcher of nonlinear systems. "Linearity means that the rule that determines what a piece of a system is going to do next is not influenced by what it is doing now. More precisely, this is intended in a differential or incremental sense: For a linear spring, the increase of its tension is proportional to the increment whereby it is stretched, with the ratio of these increments exactly independent of how much it has already been stretched. Such a spring can be stretched arbitrarily far, and in particular will never snap or break. Accordingly, no real spring is linear." The real world is nonlinear. However, there is a number of reasons to investigate linear systems. Linear models are often used to approximate nonlinear systems. Many times, this approximation is sufficient for the controller design for the underlying nonlinear system. Knowledge of linear system theory helps to understand the intricate theory of nonlinear systems.

In this chapter, we acquaint ourselves with basic concepts from linear systems that are used in subsequent chapters.

5.1 Reachability and Controllability

We first consider a linear, time-invariant (LTI), discrete-time system modeled by

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k], \quad \boldsymbol{x}[0] = \boldsymbol{x}_0, \tag{5.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$. The solution to (5.1) is

$$\boldsymbol{x}[i] = \boldsymbol{A}^i\boldsymbol{x}[0] + \sum_{k=0}^{i-1} \boldsymbol{A}^{i-k-1}\boldsymbol{B}\boldsymbol{u}[k],$$

which can be represented as

$$\boldsymbol{x}[i] = \boldsymbol{A}^i\boldsymbol{x}[0] + \begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{i-1}\boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}[i-1] \\ \boldsymbol{u}[i-2] \\ \vdots \\ \boldsymbol{u}[0] \end{bmatrix}. \tag{5.2}$$

Let

$$\boldsymbol{U}_i = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{i-1}\boldsymbol{B} \end{bmatrix}, \quad i = 1, 2, \ldots.$$

Note that $\boldsymbol{U}_i \in \mathbb{R}^{n\times mi}$. If for some positive l,

$$\operatorname{rank} \boldsymbol{U}_l = \operatorname{rank} \boldsymbol{U}_{l+1},$$

then all the columns of $\boldsymbol{A}^l\boldsymbol{B}$ are linearly dependent on those of $\boldsymbol{U}_l$. This, in turn, implies that all the columns of $\boldsymbol{A}^{l+1}\boldsymbol{B}$, $\boldsymbol{A}^{l+2}\boldsymbol{B}, \ldots$ must also be linearly dependent on those of $\boldsymbol{U}_l$. Hence,

$$\operatorname{rank} \boldsymbol{U}_l = \operatorname{rank} \boldsymbol{U}_{l+1} = \operatorname{rank} \boldsymbol{U}_{l+2} = \ldots \tag{5.3}$$

The above implies that the rank of $\boldsymbol{U}_i$ increases by at least one when i is increased by one, until the maximum value of $\operatorname{rank} \boldsymbol{U}_i$ is attained. The maximum value of the rank $\boldsymbol{U}_i$ is guaranteed to be achieved for $i = n$, which means that, we always have

$$\operatorname{rank} \boldsymbol{U}_n = \operatorname{rank} \boldsymbol{U}_{n+1} = \operatorname{rank} \boldsymbol{U}_{n+2} = \ldots \tag{5.4}$$

It follows from (5.3) that if $\operatorname{rank} \boldsymbol{B} = r$, then

$$\operatorname{rank} \begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-r}\boldsymbol{B} \end{bmatrix}. \tag{5.5}$$

Definition 5.1 *We say that the system (5.1) is reachable if for any $\boldsymbol{x}_f$ there exists an integer $q > 0$ and a control sequence, $\{\boldsymbol{u}[i] : i = 0, 1, \ldots, q-1\}$, that transfers $\boldsymbol{x}[0] = \boldsymbol{0}$ to $\boldsymbol{x}[q] = \boldsymbol{x}_f$.*

◇

Theorem 5.1 *The system (5.1) is reachable if and only if*

$$\mathit{rank}\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} = n.$$

Proof Using (5.2) where $\boldsymbol{x}[0] = \boldsymbol{0}$ and $\boldsymbol{x}[q] = \boldsymbol{x}_f$, we obtain

$$\begin{aligned} \boldsymbol{x}_f &= \sum_{k=0}^{q-1} \boldsymbol{A}^{q-k-1}\boldsymbol{B}\boldsymbol{u}[k] \\ &= \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{q-1}\boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}[q-1] \\ \boldsymbol{u}[q-2] \\ \vdots \\ \boldsymbol{u}[0] \end{bmatrix}. \end{aligned} \tag{5.6}$$

It follows from the above that for any $\boldsymbol{x}_f$ there exists a control sequence $\{\boldsymbol{u}[i] : i = 0, 1, \ldots, q-1\}$ that transfers $\boldsymbol{x}[0] = \boldsymbol{0}$ to $\boldsymbol{x}[q] = \boldsymbol{x}_f$ if and only if

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{q-1}\boldsymbol{B} \end{bmatrix} = \operatorname{rank}\boldsymbol{U}_q = n.$$

By (5.4) the maximum value of the rank of $\boldsymbol{U}_q$ is guaranteed to be attained for $q = n$, which proves the theorem.

□

A closely related notion to reachability is the notion of *controllability*.

Definition 5.2 *A system is controllable if it can be transferred from any given state to the origin* $\boldsymbol{0}$ *of the state space by means of an appropriately chosen control law.*

◇

Theorem 5.2 *The system (5.1) is controllable if and only if*

$$\mathit{range}\,(\boldsymbol{A}^n) \subset \mathit{range}\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix}.$$

Proof Using the solution formula for the system (5.1), for the case when $\boldsymbol{x}[0] = \boldsymbol{x}_0$ and $\boldsymbol{x}_f = \boldsymbol{x}[q] = \boldsymbol{0}$, we obtain

$$\begin{aligned} \boldsymbol{A}^q\boldsymbol{x}_0 &= -\sum_{k=0}^{q-1} \boldsymbol{A}^{q-k-1}\boldsymbol{B}\boldsymbol{u}[k] \\ &= -\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{q-1}\boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}[q-1] \\ \boldsymbol{u}[q-2] \\ \vdots \\ \boldsymbol{u}[0] \end{bmatrix}. \end{aligned} \tag{5.7}$$

The system (5.1) is controllable if and only if for some $q > 0$ and arbitrary initial condition $\boldsymbol{x}_0$, the vector $\boldsymbol{A}^q\boldsymbol{x}_0$ is in the range of $\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{q-1}\boldsymbol{B} \end{bmatrix} = \boldsymbol{U}_q$. Taking into account that the maximal range of $\boldsymbol{U}_q$ is guaranteed to be attained for $q = n$, the system (5.1) is controllable if and only if the condition of the theorem is satisfied.

□

Note that the condition of the above theorem can be equivalently represented as

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} = \operatorname{rank}\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} & \boldsymbol{A}^n \end{bmatrix}. \tag{5.8}$$

We illustrate the above discussion with a simple example that we found in [165, page 482].

Example 5.1 Consider a discrete-time system modeled by

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{b}u[k] \\ &= \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix}\boldsymbol{x}[k] + \begin{bmatrix} 1 \\ 0 \end{bmatrix}u[k], \quad a \in \mathbb{R}. \end{aligned} \tag{5.9}$$

The above system model is not reachable because

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{b} & \boldsymbol{Ab} \end{bmatrix} = \operatorname{rank}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = 1 < 2.$$

It is easy to check using (5.6) that, for example, the state $\boldsymbol{x}_f = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$ is not reachable from $\boldsymbol{x}[0] = \boldsymbol{0}$ because $\boldsymbol{x}_f$ does not belong to the range of $\begin{bmatrix} \boldsymbol{b} & \boldsymbol{Ab} \end{bmatrix}$. In fact, nonreachable states form the set,

$$\left\{ c\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ c \in \mathbb{R} \right\}.$$

However, the system (5.9) is controllable because for any $a \in \mathbb{R}$,

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{b} & \boldsymbol{Ab} \end{bmatrix} = \operatorname{rank}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} \boldsymbol{b} & \boldsymbol{Ab} & \boldsymbol{A}^2 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

That is,

$$\operatorname{range}\left(\boldsymbol{A}^2\right) = \operatorname{range}(\boldsymbol{O}) \subset \operatorname{range}\begin{bmatrix} \boldsymbol{b} & \boldsymbol{Ab} \end{bmatrix}.$$

Using (5.7), we can easily see that arbitrary initial state $\boldsymbol{x}[0] \in \mathbb{R}^2$ of the system (5.9) can be transferred to the origin of $\mathbb{R}^2$ using, for example, the zero control sequence, $u[0] = u[1] = 0$. This is because $\boldsymbol{A}^2 = \boldsymbol{O}$ and hence $\boldsymbol{A}^2\boldsymbol{x}[0] = \boldsymbol{0}$ for any $\boldsymbol{x}[0] \in \mathbb{R}^2$.

In summary, for discrete-time linear systems, reachability implies controllability, and the two notions are equivalent if the matrix $\boldsymbol{A}$ of the given discrete system is nonsingular.

We next discuss the notions of reachability and controllability for continuous-time linear systems.

Definition 5.3 *We say that the system $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$, where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, or equivalently, the pair $(\boldsymbol{A}, \boldsymbol{B})$, is reachable if for any $\boldsymbol{x}_f \in \mathbb{R}^n$ there is t_1 and a control law $\boldsymbol{u}(t)$ that transfers $\boldsymbol{x}(t_0) = \boldsymbol{0}$ to $\boldsymbol{x}(t_1) = \boldsymbol{x}_f$.*

$\diamond$

Definition 5.4 *We say that the system $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$, or equivalently, the pair $(\boldsymbol{A}, \boldsymbol{B})$, is controllable if there is a control law $\boldsymbol{u}(t)$ that transfers any initial state $\boldsymbol{x}_0 = \boldsymbol{x}(t_0)$ at any time t_0 to the origin $\boldsymbol{0}$ of the state space at some time $t_1 > t_0$.*

$\diamond$

It turns out that in the case of continuous, time invariant, systems the two notions are equivalent. Indeed, from the solution formula for the controlled system (3.26), for $\boldsymbol{x}(0) = \boldsymbol{0}$ and $\boldsymbol{x}(t_1) = \boldsymbol{x}_f$, we obtain

$$\boldsymbol{x}_f = \int_0^{t_1} e^{\boldsymbol{A}(t_1-t)}\boldsymbol{B}\boldsymbol{u}(t)dt.$$

Premultiplying the above equation by $e^{-\boldsymbol{A}t_1}$ and rearranging, we get

$$\boldsymbol{v} = \int_0^{t_1} e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{u}(t)dt, \tag{5.10}$$

where $\boldsymbol{v} = e^{-\boldsymbol{A}t_1}\boldsymbol{x}_f$. On the other hand, from the solution formula for the controlled system (3.26), for $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and $\boldsymbol{x}(t_1) = \boldsymbol{0}$, we obtain

$$\boldsymbol{0} = e^{\boldsymbol{A}t_1}\boldsymbol{x}_0 + \int_0^{t_1} e^{\boldsymbol{A}(t_1-t)}\boldsymbol{B}\boldsymbol{u}(t)dt.$$

Premultiplying the above equation by $e^{-\boldsymbol{A}t_1}$ and rearranging yields

$$-\boldsymbol{x}_0 = \int_0^{t_1} e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{u}(t)dt. \tag{5.11}$$

Comparing (5.10) and (5.11), we conclude that reachability of the system $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$ is equivalent to its controllability. This is not true in the case of discrete-time systems, where reachability implies controllability, and the two notions are equivalent if the matrix $\boldsymbol{A}$ of the given discrete system is nonsingular.

There are many criteria for checking if a given system is reachable/controllable or not. We present a few of them in this section.

Theorem 5.3 *The following are equivalent:*

(1) The system $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$ *is reachable;*

(2) rank $\begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} = n$;

(3) The matrix

$$\boldsymbol{W}(t_0, t_1) = \int_{t_0}^{t_1} e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{B}^\top e^{-\boldsymbol{A}^\top t} dt$$

is nonsingular for all $t_1 > t_0$*;*

Proof We first prove the implication (1) $\Rightarrow$ (2), that is, if the pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable, then the so called controllability matrix

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} \in \mathbb{R}^{n\times mn}$$

is of full rank n. We prove this statement by contraposition. Thus, we assume that

$$\operatorname{rank} \begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} < n,$$

then, there is a constant n-vector $\boldsymbol{q} \neq \boldsymbol{0}$ so that

$$\boldsymbol{q}^\top\boldsymbol{B} = \boldsymbol{0}^\top, \quad \boldsymbol{q}^\top\boldsymbol{A}\boldsymbol{B} = \boldsymbol{0}^\top, \ldots, \boldsymbol{q}^\top\boldsymbol{A}^{n-1}\boldsymbol{B} = \boldsymbol{0}^\top.$$

By the Cayley-Hamilton theorem the matrix $\boldsymbol{A}$ satisfies its own characteristic equation. Hence

$$\boldsymbol{A}^n = -a_{n-1}\boldsymbol{A}^{n-1} - \cdots - a_1\boldsymbol{A} - a_0\boldsymbol{I}_n.$$

Therefore

$$\boldsymbol{q}^\top\boldsymbol{A}^n\boldsymbol{B} = \boldsymbol{q}^\top\left(-a_{n-1}\boldsymbol{A}^{n-1}\boldsymbol{B} - \cdots - a_1\boldsymbol{A}\boldsymbol{B} - a_0\boldsymbol{B}\right) = \boldsymbol{0}^\top.$$

By induction

$$\boldsymbol{q}^\top\boldsymbol{A}^i\boldsymbol{B} = \boldsymbol{0}^\top \quad \text{for} \quad i = n+1, n+2, \ldots$$

Let now $\boldsymbol{x}(0) = \boldsymbol{0}$ and $\boldsymbol{x}(t_1) = \boldsymbol{q}$. We will show that there is no control law that can transfer the system from $\boldsymbol{x}(0) = \boldsymbol{0}$ to $\boldsymbol{x}(t_1) = \boldsymbol{q}$. From the solution formula for the controlled system, we obtain

$$\boldsymbol{q} = \int_0^{t_1} e^{\boldsymbol{A}(t_1-t)}\boldsymbol{B}\boldsymbol{u}(t)dt.$$

We premultiply the resulting equation by $\boldsymbol{q}^\top$ to get

$$0 \neq \|\boldsymbol{q}\|^2 = \int_0^{t_1} \boldsymbol{q}^\top e^{\boldsymbol{A}(t_1-t)} \boldsymbol{B}\boldsymbol{u}(t)dt = 0$$

because

$$\boldsymbol{q}^\top e^{\boldsymbol{A}(t_1-t)} \boldsymbol{B} = \boldsymbol{q}^\top \left(\boldsymbol{B} + (t_1-t)\boldsymbol{A}\boldsymbol{B} + \frac{(t_1-t)^2}{2!}\boldsymbol{A}^2\boldsymbol{B} + \cdots \right) = \boldsymbol{0}^\top.$$

Thus the system cannot be transferred from $\boldsymbol{x}(0) = \boldsymbol{0}$ to $\boldsymbol{x}(t_1) = \boldsymbol{q}$, and hence the pair $(\boldsymbol{A}, \boldsymbol{B})$ is not reachable.

We next prove, by contraposition, the implication (2) $\Rightarrow$ (3), that is, if the controllability matrix is of full rank, then $\det \boldsymbol{W}(t_0, t_1) \neq 0$ for any $t_1 > t_0$. So assume that $\det \boldsymbol{W}(t_0, t_1) = 0$. Then, there is a nonzero n-vector $\boldsymbol{q} \in \mathbb{R}^n$ such that

$$\boldsymbol{q}^\top \int_{t_0}^{t_1} e^{-\boldsymbol{A}t} \boldsymbol{B}\boldsymbol{B}^\top e^{-\boldsymbol{A}^\top t} dt = \boldsymbol{0}$$

for all $t \in [t_0, t_1]$. We postmultiply the above equation by $\boldsymbol{q}$ to get

$$\int_{t_0}^{t_1} \left(\boldsymbol{q}^\top e^{-\boldsymbol{A}t}\boldsymbol{B}\right)\left(\boldsymbol{B}^\top e^{-\boldsymbol{A}^\top t}\boldsymbol{q}\right) dt = 0.$$

Let $\boldsymbol{v}(t) = \boldsymbol{B}^\top e^{-\boldsymbol{A}^\top t}\boldsymbol{q}$, then the above equation can be written as

$$\int_{t_0}^{t_1} \|\boldsymbol{v}(t)\|^2 dt = 0.$$

However, the above is true if and only if $\boldsymbol{v}(t) = \boldsymbol{0}$, that is, if and only if the rows of $e^{-\boldsymbol{A}t}\boldsymbol{B}$ are linearly dependent over the set of real numbers for all $t \in [t_0, t_1]$. Now, evaluating

$$\boldsymbol{q}^\top e^{-\boldsymbol{A}t}\boldsymbol{B} = \boldsymbol{0}^\top$$

and its successive derivatives with respect to t at $t = 0$, we obtain

$$\boldsymbol{q}^\top\boldsymbol{B} = \boldsymbol{0}^\top, \quad \boldsymbol{q}^\top\boldsymbol{A}\boldsymbol{B} = \boldsymbol{0}^\top, \ldots, \boldsymbol{q}^\top\boldsymbol{A}^{n-1}\boldsymbol{B} = \boldsymbol{0}^\top,$$

which means that $\operatorname{rank}\begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} < n$.

We now prove, constructively, the implication (3) $\Rightarrow$ (1). Let $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ be an arbitrary initial state and $\boldsymbol{x}(t_1) = \boldsymbol{x}_f$ an arbitrary final state. Then, we claim that the control law

$$\boldsymbol{u}(t) = \boldsymbol{B}^\top e^{-\boldsymbol{A}^\top t}\boldsymbol{W}^{-1}(t_0, t_1)\left(e^{-\boldsymbol{A}t_1}\boldsymbol{x}(t_1) - e^{-\boldsymbol{A}t_0}\boldsymbol{x}_0\right) \tag{5.12}$$

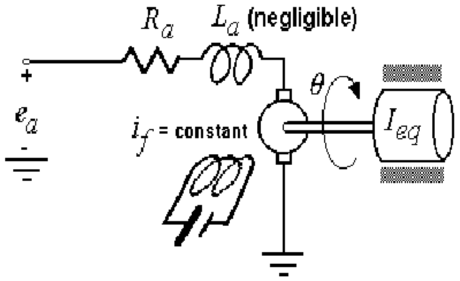


Figure 5.1: Schematic of an armature-controlled DC motor system of Example 5.2.

will transfer the system from $\boldsymbol{x}_0$ to $\boldsymbol{x}_f$. Note that to construct the above control law the matrix $\boldsymbol{W}(t_0, t_1)$ must be invertible. Substituting (5.12) into

$$\boldsymbol{x}(t_1) = e^{\boldsymbol{A}(t_1-t_0)}\boldsymbol{x}_0 + \int_{t_0}^{t_1} e^{\boldsymbol{A}(t_1-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau$$

yields

$$\begin{aligned}\boldsymbol{x}(t_1) &= e^{\boldsymbol{A}(t_1-t_0)}\boldsymbol{x}_0 \\ &\quad + e^{\boldsymbol{A}t_1}\int_{t_0}^{t_1} e^{-\boldsymbol{A}\tau}\boldsymbol{B}\boldsymbol{B}^\top e^{-\boldsymbol{A}^\top\tau}d\tau \boldsymbol{W}^{-1}(t_0,t_1)\left(e^{-\boldsymbol{A}t_1}\boldsymbol{x}(t_1) - e^{-\boldsymbol{A}t_0}\boldsymbol{x}_0\right).\end{aligned}$$

Performing easy manipulations gives $\boldsymbol{x}(t_1) = \boldsymbol{x}(t_1)$, which means that indeed the proposed control law does the required job. Thus, we have established equivalence between the three statements of the theorem.

□

Example 5.2 Consider an armature-controlled DC motor system whose schematic is shown in Figure 5.1, where the system parameters are: $R_a = 1\,\Omega$, $L_a \approx 0\,\mathrm{H}$, $K_b = 5\,\mathrm{V/rad/sec}$, $K_i = 5\,\mathrm{Nm/A}$, and the effective moment of inertia is $I_{eq} = 0.1$ kg·m^2. The gear inertia and friction are negligible.

Applying Kirchhoff's voltage law to the armature circuit gives

$$R_a i_a + K_b\dot{\theta} = e_a. \tag{5.13}$$

The armature-controlled DC motor uses a constant field current, and therefore the motor torque is

$$\tau_m = K_i i_a.$$

The motor torque is equal to the torque delivered to the load. Hence,

$$I_{eq}\ddot{\theta} = K_i i_a. \tag{5.14}$$

Substituting i_a from (5.13) into the above equation and dividing both sides by I_{eq} yields

$$\begin{aligned}\ddot{\theta} &= \frac{K_i}{I_{eq}}\left(\frac{e_a - K_b\dot{\theta}}{R_a}\right)\\ &= -\frac{K_i K_b}{I_{eq}R_a}\dot{\theta} + \frac{K_i}{I_{eq}R_a}e_a.\end{aligned}$$

Let $x_1 = \theta$, $x_2 = \dot{\theta}$, and $u = e_a$. Then taking into account the system parameters, we represent the above equation in a state space format,

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -250\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 50\end{bmatrix}u.$$

We construct a control u that transfers the system from the initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}(0) = \begin{bmatrix}10 & 1\end{bmatrix}^\top$ to the final state $\boldsymbol{x}(2) = \begin{bmatrix}0 & 0\end{bmatrix}^\top$. Recall that the solution of a controlled system has the form

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau.$$

Our goal is to find u so that

$$\boldsymbol{0} = e^{2\boldsymbol{A}}\boldsymbol{x}(0) + \int_0^2 e^{\boldsymbol{A}(2-\tau)}\boldsymbol{b}u(\tau)d\tau.$$

One can verify that one such u has the form

$$u(t) = -\boldsymbol{b}^\top e^{\boldsymbol{A}^\top(2-t)}\left(\int_0^2 e^{\boldsymbol{A}(2-\tau)}\boldsymbol{b}\boldsymbol{b}^\top e^{\boldsymbol{A}^\top(2-\tau)}d\tau\right)^{-1} e^{2\boldsymbol{A}}\boldsymbol{x}(0),$$

where

$$e^{\boldsymbol{A}t} = \begin{bmatrix}1 & \frac{1}{250}\left(1 - e^{-250t}\right)\\ 0 & e^{-250t}\end{bmatrix}.$$

Hence,

$$\begin{aligned}
\int_0^2 e^{\boldsymbol{A}(2-\tau)}\boldsymbol{b}\boldsymbol{b}^\top e^{\boldsymbol{A}^\top(2-\tau)}d\tau &= \int_0^2 \begin{bmatrix} 1 & \frac{1}{250}\left(1-e^{-250(2-\tau)}\right) \\ 0 & e^{-250(2-\tau)} \end{bmatrix} \begin{bmatrix} 0 \\ 50 \end{bmatrix} \begin{bmatrix} 0 & 50 \end{bmatrix} \\
&\quad \times \begin{bmatrix} 1 & 0 \\ \frac{1}{250}\left(1-e^{-250(2-\tau)}\right) & e^{-250(2-\tau)} \end{bmatrix} d\tau \\
&= \begin{bmatrix} 0.07976 & 0.02 \\ 0.02 & 5 \end{bmatrix}.
\end{aligned}$$

The inverse of the above matrix is

$$\begin{bmatrix} 12.5502 & -0.0502 \\ -0.0502 & 0.2002 \end{bmatrix}.$$

Hence,

$$\begin{aligned}
u(t) &= -\begin{bmatrix} 0 & 50 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1}{250}\left(1-e^{-250(2-t)}\right) & e^{-250(2-t)} \end{bmatrix} \\
&\quad \times \begin{bmatrix} 12.5502 & -0.0502 \\ -0.0502 & 0.2002 \end{bmatrix} e^{2\boldsymbol{A}} \begin{bmatrix} 10 \\ 1 \end{bmatrix} \\
&= 50.22e^{(250t-500)} - 25.11.
\end{aligned}$$

The above control strategy transfers the system from $\boldsymbol{x}(0) = \begin{bmatrix} 10 & 1 \end{bmatrix}^\top$ to $\boldsymbol{x}(2) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$. Plots of x_1, x_2 versus time are shown in Figure 5.2. A plot of u versus time is given in Figure 5.3.

We now give a necessary and sufficient condition for the pair $(\boldsymbol{A}, \boldsymbol{B})$ to be nonreachable.

Theorem 5.4 *The pair $(\boldsymbol{A}, \boldsymbol{B})$ is nonreachable if and only if there is a similarity transformationindextransformation!similarity $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x}$ such that*

$$\tilde{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{O} & \boldsymbol{A}_4 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \boldsymbol{T}\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{O} \end{bmatrix},$$

where the pair $(\boldsymbol{A}_1, \boldsymbol{B}_1)$ is reachable, $\boldsymbol{A}_1 \in \mathbb{R}^{r\times r}$, $\boldsymbol{B}_1 \in \mathbb{R}^{r\times m}$, and the rank of the controllability matrix of the pair $(\boldsymbol{A}, \boldsymbol{B})$ equals r.

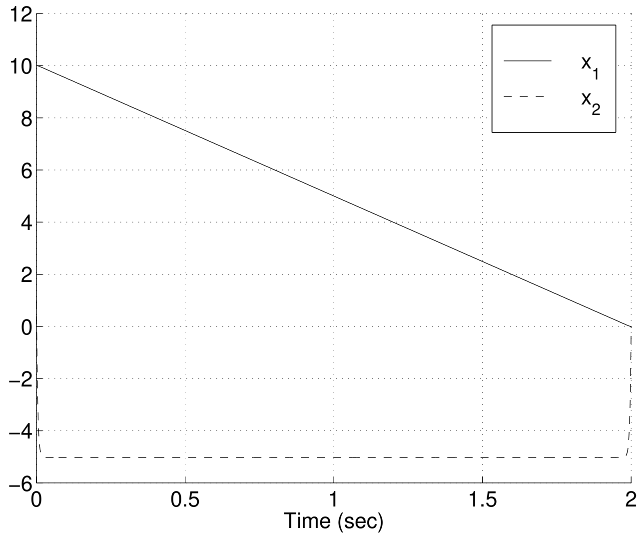


Figure 5.2: Plots of state variables versus time of Example 5.2.

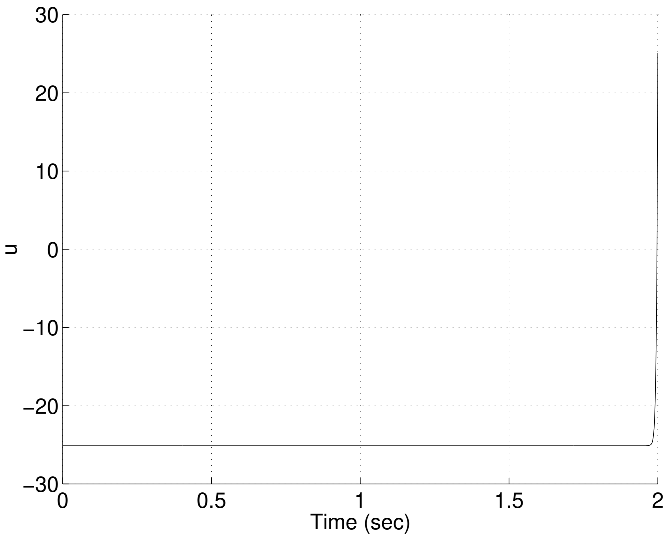


Figure 5.3: A plot of the control action versus time of Example 5.2.

Proof We first prove necessity ($\Rightarrow$). After forming the controllability matrix of the pair $(\boldsymbol{A}, \boldsymbol{B})$, we use elementary row operations to get

$$\begin{aligned}
\boldsymbol{T}\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{TB} & \left(\boldsymbol{TAT}^{-1}\right)(\boldsymbol{TB}) & \cdots & \left(\boldsymbol{TAT}^{-1}\right)^{n-1}(\boldsymbol{TB}) \end{bmatrix} \\
&= \begin{bmatrix} x & x & x & \cdots & x & x & x \\ 0 & x & x & \cdots & x & x & x \\ \vdots & & & \vdots & & & \vdots \\ 0 & 0 & 0 & \cdots & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{A}_1\boldsymbol{B}_1 & \boldsymbol{A}_1^2\boldsymbol{B}_1 & \cdots & \boldsymbol{A}_1^{n-1}\boldsymbol{B}_1 \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} \end{bmatrix},
\end{aligned} \tag{5.15}$$

where the symbol x denotes a "don't care", that is, an unspecified scalar. We now show that indeed

$$\tilde{\boldsymbol{A}} = \boldsymbol{TAT}^{-1} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \boldsymbol{TB} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{O} \end{bmatrix},$$

where $\boldsymbol{A}_3 = \boldsymbol{O}$. It is clear that $\boldsymbol{TB}$ must have the above form. If $\boldsymbol{A}_3$ were not a zero matrix, then we would have

$$\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{A}_1\boldsymbol{B}_1 \\ \boldsymbol{A}_3\boldsymbol{B}_1 \end{bmatrix}.$$

Comparing the above with (5.15), we conclude that $\boldsymbol{A}_3\boldsymbol{B}_1 = \boldsymbol{O}$. Thus,

$$\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{A}_1\boldsymbol{B}_1 \\ \boldsymbol{O} \end{bmatrix}.$$

Next we consider

$$\tilde{\boldsymbol{A}}^2\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{A}_1^2\boldsymbol{B}_1 \\ \boldsymbol{A}_3\boldsymbol{A}_1\boldsymbol{B}_1 \end{bmatrix}.$$

Again, comparing the above with (5.15) gives

$$\boldsymbol{A}_3\boldsymbol{A}_1\boldsymbol{B}_1 = \boldsymbol{O},$$

and thus

$$\tilde{\boldsymbol{A}}^2\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{A}_1^2\boldsymbol{B}_1 \\ \boldsymbol{O} \end{bmatrix}.$$

Continuing in this manner, we conclude that $\boldsymbol{A}_3\boldsymbol{A}_1^{n-1}\boldsymbol{B}_1 = \boldsymbol{O}$. Thus it follows that

$$\boldsymbol{A}_3 \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{A}_1\boldsymbol{B}_1 & \cdots & \boldsymbol{A}_1^{n-1}\boldsymbol{B}_1 \end{bmatrix} = \boldsymbol{O}.$$

Because $\operatorname{rank}\begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{A}_1\boldsymbol{B}_1 & \cdots & \boldsymbol{A}_1^{n-1}\boldsymbol{B}_1 \end{bmatrix} = r$, that is, the controllability matrix of the pair $(\boldsymbol{A}_1, \boldsymbol{B}_1)$ is of full rank, we have to have $\boldsymbol{A}_3 = \boldsymbol{O}$, and the proof of necessity is complete.

To prove sufficiency ($\Leftarrow$), note that if there is a similarity transformation such that

$$\tilde{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{O} & \boldsymbol{A}_4 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \boldsymbol{T}\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{O} \end{bmatrix},$$

then the controllability matrix of the pair $\left(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}\right)$ has the form

$$\begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{A}_1\boldsymbol{B}_1 & \boldsymbol{A}_1^2\boldsymbol{B}_1 & \cdots & \boldsymbol{A}_1^{n-1}\boldsymbol{B}_1 \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} \end{bmatrix}.$$

Hence, the pair $\left(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}\right)$ is nonreachable. Because the similarity transformation preserves the reachability property, the pair $(\boldsymbol{A}, \boldsymbol{B})$ is nonreachable.

$\square$

Example 5.3 For the system model,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 1 & 1 & -1 & 2 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u},$$

construct a state variable transformation so that in the new coordinates the reachable part is separated and distinct from the nonreachable part.

We first form the controllability matrix of the pair $(\boldsymbol{A}, \boldsymbol{B})$ to get

$$\boldsymbol{\mathcal{U}} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \boldsymbol{A}^2\boldsymbol{B} & \boldsymbol{A}^3\boldsymbol{B} \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & 0 & 3 & 4 & -1 & 4 \\ 0 & 0 & 1 & 1 & 0 & 1 & 3 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 3 & 2 & 7 & 7 & 19 \end{bmatrix}.$$

We then use the MATLAB command `[Q,R]=qr(C)` to obtain

$$\boldsymbol{Q} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} -1 & -1 & 1 & 0 & -3 & -4 & 1 & -4 \\ 0 & -1 & -1 & -3 & -2 & -7 & -7 & -19 \\ 0 & 0 & -1 & -1 & 0 & -1 & -3 & -5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ satisfy $\boldsymbol{\mathcal{U}} = \boldsymbol{QR}$, and $\boldsymbol{R}$ is upper row triangular such that $\operatorname{rank}\boldsymbol{R} = \operatorname{rank}\boldsymbol{\mathcal{U}}$. Thus, premultiplying the controllability matrix, $\boldsymbol{\mathcal{U}}$, by $\boldsymbol{Q}^{-1}$ reduces this matrix to an upper row triangular matrix $\boldsymbol{R}$ because $\boldsymbol{Q}^{-1}\boldsymbol{\mathcal{U}} = \boldsymbol{R}$. Let $\boldsymbol{z} = \boldsymbol{Q}^{-1}\boldsymbol{x}$ be the state variable transformation. Then in the new coordinates the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ take the form

$$\boldsymbol{Q}^{-1}\boldsymbol{AQ} = \left[\begin{array}{ccc|c} -1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 4 \end{array}\right] \quad \text{and} \quad \boldsymbol{Q}^{-1}\boldsymbol{B} = \left[\begin{array}{cc} -1 & -1 \\ 0 & -1 \\ 0 & 0 \\ \hline 0 & 0 \end{array}\right].$$

The dimension of the nonreachable part is 1.

Definition 5.5 *If the eigenvalues of the noncontrollable part of a given pair $(\boldsymbol{A}, \boldsymbol{B})$ are all in the open left-hand complex plane, that is, the nonreachable part is asymptotically stable, then the pair $(\boldsymbol{A}, \boldsymbol{B})$ is called stabilizable.*

◇

We now give yet another test for reachability. This test is also known as the Popov-Belevitch-Hautus (PBH) eigenvector test. We denote the set of eigenvalues of a matrix $\boldsymbol{A}$ as $\operatorname{eig}(\boldsymbol{A})$.

Theorem 5.5 *The pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable if and only if*

$$\mathit{rank}\begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} = n \quad \textit{for all} \quad s \in \mathit{eig}(\boldsymbol{A}).$$

Proof We first prove the necessity part ($\Rightarrow$) by contraposition. If there is a vector $\boldsymbol{q} \neq \boldsymbol{0}$ such that for some $s \in \operatorname{eig}(\boldsymbol{A})$,

$$\boldsymbol{q}^\top \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} = \boldsymbol{0}^\top,$$

that is,

$$\boldsymbol{q}^\top \boldsymbol{A} = s\boldsymbol{q}^\top, \quad \boldsymbol{q}^\top \boldsymbol{B} = \boldsymbol{0}^\top,$$

then

$$\boldsymbol{q}^\top \boldsymbol{AB} = s\boldsymbol{q}^\top \boldsymbol{B} = \boldsymbol{0}^\top,$$
$$\boldsymbol{q}^\top \boldsymbol{A}^2\boldsymbol{B} = s\boldsymbol{q}^\top \boldsymbol{AB} = \boldsymbol{0}^\top,$$

and so on, until

$$\boldsymbol{q}^\top \boldsymbol{A}^{n-1}\boldsymbol{B} = s\boldsymbol{q}^\top \boldsymbol{A}^{n-2}\boldsymbol{B} = \boldsymbol{0}^\top.$$

We write the above equations in the form

$$\boldsymbol{q}^\top \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} = \boldsymbol{0}^\top,$$

which means that the pair $(\boldsymbol{A}, \boldsymbol{B})$ is nonreachable.

We now prove the sufficiency part ($\Leftarrow$) also by contraposition. So we assume that rank $\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} = r < n$. By Theorem 5.4, there exists a state variable transformation such that in the new basis the pair $(\boldsymbol{A}, \boldsymbol{B})$ takes the form

$$\tilde{\boldsymbol{A}} = \boldsymbol{TAT}^{-1} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{O} & \boldsymbol{A}_4 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \boldsymbol{TB} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{O} \end{bmatrix}.$$

We choose $\boldsymbol{q}$ of the form

$$\boldsymbol{q}^\top = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{z} \end{bmatrix},$$

where $\boldsymbol{z} \neq \boldsymbol{0}$ is a left eigenvector of $\boldsymbol{A}_4$, that is,

$$\boldsymbol{z}\boldsymbol{A}_4 = s\boldsymbol{z},$$

where $s \in \operatorname{eig}(\boldsymbol{A}_4)$. Thus

$$\begin{aligned} \boldsymbol{q}^\top \tilde{\boldsymbol{A}} &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{z} \end{bmatrix} \tilde{\boldsymbol{A}} \\ &= \begin{bmatrix} \boldsymbol{0} & s\boldsymbol{z} \end{bmatrix} \\ &= s\boldsymbol{q}^\top. \end{aligned}$$

Clearly

$$\boldsymbol{q}^\top \begin{bmatrix} s\boldsymbol{I}_n - \tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}} \end{bmatrix} = \boldsymbol{0}^\top,$$

which means that the rank of the matrix $\begin{bmatrix} s\boldsymbol{I}_n - \tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}} \end{bmatrix}$ is less than n, and therefore the proof of the sufficiency part is complete.

□

5.2 Observability and Constructability

As we have seen in the previous section, system reachability implies the ability to completely control the entire state of the system. Suppose now that the system state is not directly accessible. Instead, we have the output of the system,

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du},$$

where $\boldsymbol{C} \in \mathbb{R}^{p\times n}$, $p \le n$, and $\boldsymbol{D} \in \mathbb{R}^{p\times m}$. However, we still want to know the dynamical behavior of the entire state.

Definition 5.6 *We say that the system*

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} &=& \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &=& \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}, \end{array}\right\}$$

or equivalently the pair $(\boldsymbol{A}, \boldsymbol{C})$*, is observable if there is a finite* $t_1 > t_0$ *such that for arbitrary* $\boldsymbol{u}(t)$ *and resulting* $\boldsymbol{y}(t)$ *over* $[t_0, t_1]$*, we can determine* $\boldsymbol{x}(t_0)$ *from complete knowledge of the system input* $\boldsymbol{u}$ *and output* $\boldsymbol{y}$*.*

◇

Note that once $\boldsymbol{x}(t_0)$ is known, we can determine $\boldsymbol{x}(t)$ from knowledge of $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ over any finite time interval $[t_0, t_1]$. We now show how to constructively determine $\boldsymbol{x}(t_0)$, given $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$. While doing so we establish a sufficiency condition for the system to be observable. The condition is also necessary for the system observability as we show it next. To proceed with constructive determination of the state vector, we note that the solution $\boldsymbol{y}(t)$ is given by

$$\boldsymbol{y}(t) = \boldsymbol{C}e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} \boldsymbol{C}e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau + \boldsymbol{D}\boldsymbol{u}(t).$$

We subtract $\int_{t_0}^{t} \boldsymbol{C}e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau + \boldsymbol{D}\boldsymbol{u}(t)$ from both sides of the above equation. Let

$$\boldsymbol{g}(t) = \boldsymbol{y}(t) - \int_{t_0}^{t} \boldsymbol{C}e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau - \boldsymbol{D}\boldsymbol{u}(t),$$

then we have

$$\boldsymbol{g}(t) = \boldsymbol{C}e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0),$$

where $\boldsymbol{g}$ is known to us. Our goal now is to determine $\boldsymbol{x}(t_0)$. Having $\boldsymbol{x}(t_0)$, we can determine the entire $\boldsymbol{x}(t)$ for all $t \in [t_0, t_1]$ from the formula

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau.$$

Premultiplying both sides of $\boldsymbol{g}(t)$ by $e^{\boldsymbol{A}^\top(t-t_0)}\boldsymbol{C}^\top$ and integrating the resulting expression between the limits t_0 and t_1, we get

$$\int_{t_0}^{t_1} e^{\boldsymbol{A}^\top(t-t_0)}\boldsymbol{C}^\top\boldsymbol{C}e^{\boldsymbol{A}(t-t_0)}dt\, \boldsymbol{x}(t_0) = \int_{t_0}^{t_1} e^{\boldsymbol{A}^\top(t-t_0)}\boldsymbol{C}^\top\boldsymbol{g}(t)dt.$$

Let

$$\boldsymbol{V}(t_0, t_1) = \int_{t_0}^{t_1} e^{\boldsymbol{A}^\top t} \boldsymbol{C}^\top \boldsymbol{C} e^{\boldsymbol{A}t} dt.$$

Then, after performing some manipulations and assuming that the matrix $\boldsymbol{V}(t_0, t_1)$ is invertible, we obtain

$$\boldsymbol{x}(t_0) = e^{\boldsymbol{A}t_0} \boldsymbol{V}^{-1}(t_0, t_1) e^{\boldsymbol{A}^\top t_0} \int_{t_0}^{t_1} e^{\boldsymbol{A}^\top (t-t_0)} \boldsymbol{C}^\top \boldsymbol{g}(t) dt.$$

Knowledge of $\boldsymbol{x}(t_0)$ allows us to reconstruct the entire state $\boldsymbol{x}(t)$ over the interval $[t_0, t_1]$. We conclude that if the matrix $\boldsymbol{V}(t_0, t_1)$ is invertible, then the system is observable.

Example 5.4 For a dynamical system modeled by

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} &=& \begin{bmatrix} -1/2 & 1/2 \\ -1/2 & 1/2 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} -1 \\ 1 \end{bmatrix} u \\ y &=& \begin{bmatrix} 1 & 1 \end{bmatrix} \boldsymbol{x}, \end{array}\right\}$$

we determine $\boldsymbol{x}(t)$ over the time interval $[0, 10]$ knowing that $u(t) = 1$ for $t \geq 0$ and $y(t) = t^2 + 2t$ for $t \geq 0$.

We first find $\boldsymbol{x}(0)$. Knowledge of $\boldsymbol{x}(0)$ will allow us to find the entire state vector for all $t \in [0, 10]$. We have $t_0 = 0$ and $t_1 = 10$. Hence,

$$\begin{aligned} \boldsymbol{x}(t_0) = \boldsymbol{x}(0) &= e^{\boldsymbol{A}t_0} \boldsymbol{V}^{-1}(t_0, t_1) e^{\boldsymbol{A}^\top t_0} \int_{t_0}^{t_1} e^{\boldsymbol{A}^\top (t-t_0)} \boldsymbol{c}^\top g(t) dt \\ &= \boldsymbol{V}^{-1}(0, 10) \int_0^{10} e^{\boldsymbol{A}^\top t} \boldsymbol{c}^\top g(t) dt, \end{aligned}$$

where

$$g(t) = y(t) - \boldsymbol{c} \int_0^t e^{\boldsymbol{A}(t-\tau)} \boldsymbol{b} u(\tau) d\tau.$$

To evaluate the above expression, we first use formula (3.4) to compute,

$$e^{\boldsymbol{A}t} = \mathcal{L}^{-1}\left([s\boldsymbol{I}_2 - \boldsymbol{A}]^{-1}\right) = \mathcal{L}^{-1}\left(\begin{bmatrix} \frac{1}{s} - \frac{1}{2}\frac{1}{s^2} & \frac{1}{2}\frac{1}{s^2} \\ -\frac{1}{2}\frac{1}{s^2} & \frac{1}{s} + \frac{1}{2}\frac{1}{s^2} \end{bmatrix}\right) = \begin{bmatrix} 1 - \frac{1}{2}t & \frac{1}{2}t \\ -\frac{1}{2}t & 1 + \frac{1}{2}t \end{bmatrix}.$$

Next, we compute

$$\begin{aligned} \boldsymbol{V}(0, 10) &= \int_0^{10} e^{\boldsymbol{A}^\top t} \boldsymbol{c}^\top \boldsymbol{c} e^{\boldsymbol{A}t} dt \\ &= \int_0^{10} \left(\begin{bmatrix} 1 - t \\ 1 + t \end{bmatrix} \begin{bmatrix} 1 - t & 1 + t \end{bmatrix} \right) dt = \int_0^{10} \begin{bmatrix} 1 - 2t + t^2 & 1 - t^2 \\ 1 - t^2 & 1 + 2t + t^2 \end{bmatrix} dt \\ &= \left. \begin{bmatrix} t - t^2 + \frac{t^3}{3} & t - \frac{t^3}{3} \\ t - \frac{t^3}{3} & t + t^2 + \frac{t^3}{3} \end{bmatrix} \right|_0^{10} = \begin{bmatrix} 243.333 & -323.333 \\ -323.333 & 443.333 \end{bmatrix}. \end{aligned}$$

The inverse of $\boldsymbol{V}(0,10)$ is

$$\boldsymbol{V}^{-1}(0,10)=\begin{bmatrix}0.133 & 0.097\\ 0.97 & 0.073\end{bmatrix}.$$

We can now compute $g(t)$ to get

$$\begin{aligned}
g(t) &= y(t)-\boldsymbol{c}\int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{b}u(\tau)d\tau\\
&= t^2+2t-\begin{bmatrix}1 & 1\end{bmatrix}\int_0^t\begin{bmatrix}-1+t-\tau\\ 1+t+\tau\end{bmatrix}d\tau = t^2+2t-\int_0^t(2t-2\tau)d\tau\\
&= t^2+2t-2t(\tau)\,\Big|_0^t+\tau^2\,\Big|_0^t = t^2+2t-t^2\\
&= 2t.
\end{aligned}$$

Hence,

$$\int_0^{10} e^{\boldsymbol{A}^\top t}\boldsymbol{c}^\top g(t)dt=\int_0^{10}\begin{bmatrix}(1-t)2t\\ (1+t)2t\end{bmatrix}dt=\begin{bmatrix}t^2-\frac{2}{3}t^3\\ t^2+\frac{2}{3}t^3\end{bmatrix}\Bigg|_0^{10}=\begin{bmatrix}-566.6667\\ 766.6667\end{bmatrix}.$$

Finally, we compute

$$\boldsymbol{x}(0)=\boldsymbol{V}^{-1}(0,10)\int_0^{10}e^{\boldsymbol{A}^\top t}\boldsymbol{c}^\top g(t)dt=\begin{bmatrix}-1\\ 1\end{bmatrix}.$$

Alternatively, we could use the Laplace transform method to find $\boldsymbol{x}(0)$. Knowing $\boldsymbol{x}(0)$, we can find $\boldsymbol{x}(t)$ for $t\geq 0$,

$$\begin{aligned}
\boldsymbol{x}(t) &= e^{\boldsymbol{A}t}\boldsymbol{x}(0)+\int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{b}u(\tau)d\tau\\
&= \begin{bmatrix}-1+t\\ 1+t\end{bmatrix}+\begin{bmatrix}(-1+t)t-\frac{t^2}{2}\\ (1+t)t-\frac{t^2}{2}\end{bmatrix}=\begin{bmatrix}\frac{t^2}{2}-1\\ \frac{t^2}{2}+2t+1\end{bmatrix}.
\end{aligned}$$

We next show that the invertibility of $\boldsymbol{V}(t_0,t_1)$ is also necessary for observability. Hence, if $\boldsymbol{V}(t_0,t_1)$ is singular, there exists a nonzero constant vector, say $\boldsymbol{x}_a$, such that $\boldsymbol{V}(t_0,t_1)\boldsymbol{x}_a=\boldsymbol{0}$. This implies that

$$\boldsymbol{C}e^{\boldsymbol{A}t}\boldsymbol{x}_a=\boldsymbol{0},$$

and therefore we can write

$$\boldsymbol{g}(t) = \boldsymbol{C}e^{\boldsymbol{A}t}\boldsymbol{x}_0 = \boldsymbol{C}e^{\boldsymbol{A}t}\left(\boldsymbol{x}_0 + \boldsymbol{x}_a\right).$$

Thus, $\boldsymbol{x}(0) = \boldsymbol{x}_0 + \boldsymbol{x}_a$ yields the same response as $\boldsymbol{x}(0) = \boldsymbol{x}_0$, which means that we cannot determine the system state, or in other words, the state fails to be observable. In view of the above result, it follows that by repeating essentially the same arguments used in the previous section we can prove the following theorem:

Theorem 5.6 *The following statements are equivalent:*

(1) The pair $(\boldsymbol{A}, \boldsymbol{C})$ *is observable;*

(2) The matrix $\boldsymbol{V}(t_0, t_1)$ *is nonsingular for all* $t_1 > t_0$*;*

(3) The observability matrix

$$\begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{n-1} \end{bmatrix} \in \mathbb{R}^{pn \times n}$$

is of full rank n*.*

(4)

$$rank \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} \\ \boldsymbol{C} \end{bmatrix} = n \quad \text{for all} \quad s \in eig(\boldsymbol{A}).$$

Example 5.5 For the dynamical system model,

$$\left.\begin{array}{rcl} \dot{x} & = & \boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} 2 & 0 \\ 3 & -1 \end{bmatrix} \boldsymbol{x} \\ y & = & \boldsymbol{c}\boldsymbol{x} = \begin{bmatrix} -1 & 1 \end{bmatrix} \boldsymbol{x}, \end{array}\right\}$$

we find a non-zero initial vector $\boldsymbol{x}(0) = \boldsymbol{x}_0$ such that $y(t) = 0$ for all $t \geq 0$.

We have, $\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}_0$, where

$$e^{\boldsymbol{A}t} = \mathcal{L}^{-1}\left((s\boldsymbol{I} - \boldsymbol{A})^{-1}\right) = \begin{bmatrix} e^{2t} & 0 \\ e^{2t} - e^{-t} & e^{-t} \end{bmatrix}.$$

Our objective is to find $\boldsymbol{x}_0$ such that

$$\begin{bmatrix} -1 & 1 \end{bmatrix} e^{\boldsymbol{A}t}\boldsymbol{x}_0 = 0 = y(t), \quad \text{for all} \quad t \geq 0.$$

For example, if

$$\boldsymbol{x}_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top,$$

then $y(t) = 0$ for all $t \geq 0$. Note that we were able to find $\boldsymbol{x}_0 \neq \boldsymbol{0}$ such that $y(t) = 0$ for all $t \geq 0$, because for all $t \geq 0$, the columns of $\boldsymbol{c}e^{\boldsymbol{A}t}$ are linearly dependent over the real numbers. This is because the pair $(\boldsymbol{A}, \boldsymbol{c})$ is nonobservable. The vector $\boldsymbol{x}_0$ is in the null-space of the observability matrix

$$\begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{c}\boldsymbol{A} \end{bmatrix}.$$

We can also prove analogous result to Theorem 5.4 for the given pair $(\boldsymbol{A}, \boldsymbol{C})$, that is, using a similarity transformation we can separate the observable part from the nonobservable.

Definition 5.7 *If the eigenvalues of the nonobservable part of a given pair $(\boldsymbol{A}, \boldsymbol{C})$ are all in the open left-hand complex plane, that is, if the nonobservable part is asymptotically stable, then we say that the pair $(\boldsymbol{A}, \boldsymbol{C})$ is detectable.*

◇

A related notion to observability is the notion of constructability.

Definition 5.8 *We say that the system*

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} & = & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} & = & \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}, \end{array}\right\}$$

or equivalently the pair $(\boldsymbol{A}, \boldsymbol{C})$, is constructable if there is a finite $t_1 > t_0$ such that for arbitrary $\boldsymbol{u}(t)$ and resulting $\boldsymbol{y}(t)$ over $[t_0, t_1]$ we can determine $\boldsymbol{x}(t_1)$ from complete knowledge of the system input $\boldsymbol{u}$ and output $\boldsymbol{y}$.

◇

One can verify that in the case of continuous, time invariant systems, observability is equivalent to constructability. In the case of discrete-time systems observability implies constructability, and the two notions are equivalent if the matrix $\boldsymbol{A}$ of the given discrete system is nonsingular.

5.3 Minimal Realizations

In Section 1.4, we discussed several methods of converting a transfer function into a state-space representation for both continuous and discrete LTI lumped systems. Recall that if $\boldsymbol{G}(s)$ is a proper rational transfer function matrix then a quadruple of constant matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ is a realization of $\boldsymbol{G}(s)$ if $\boldsymbol{G}(s) = \boldsymbol{C}\left(s\boldsymbol{I}_n - \boldsymbol{A}\right)^{-1}\boldsymbol{B} + \boldsymbol{D}$. Recall also that the dimension of a realization is the size of the matrix $\boldsymbol{A}$, that is, if $\boldsymbol{A}$ is an n-by-n matrix then the dimension of the corresponding realization is n.

Different methods may yield realizations of different dimensions as is the case in Examples 1.7 and 1.8. It is easy to see that a proper rational transfer function matrix has infinitely many realizations of different dimensions. A realization with the smallest possible dimension is called a minimal realization. Necessary and sufficient conditions for a realization to be minimal was given by Kalman [169] in 1963.

Theorem 5.7 *A realization $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ of a given transfer function matrix $\boldsymbol{G}(s)$ is minimal if and only if the pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable and the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable.*

We can use Theorem 5.4 to construct a minimal realization using any realization, say $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$, of a given transfer function matrix. Indeed, given a realization $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$. We check if the pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable. If not, we apply Theorem 5.4 to represent $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ in the new coordinates where the reachable part is separated from non-reachable one. In other words, we construct a similarity transformation $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x}$ such that

$$\tilde{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \left[\begin{array}{c|c} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \hline \boldsymbol{O} & \boldsymbol{A}_4 \end{array}\right], \quad \tilde{\boldsymbol{B}} = \boldsymbol{T}\boldsymbol{B} = \left[\begin{array}{c} \boldsymbol{B}_1 \\ \hline \boldsymbol{O} \end{array}\right],$$

where the pair $(\boldsymbol{A}_1, \boldsymbol{B}_1)$ is reachable, $\boldsymbol{A}_1 \in \mathbb{R}^{r\times r}$, $\boldsymbol{B}_1 \in \mathbb{R}^{r\times m}$. The matrix $\boldsymbol{C}$ becomes

$$\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{T}^{-1} = \left[\begin{array}{c|c} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{array}\right],$$

where $\boldsymbol{C}_1 \in \mathbb{R}^{p\times r}$, and $\tilde{\boldsymbol{D}} = \boldsymbol{D}$. The transfer functions of $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ and $(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{C}}, \boldsymbol{D})$ are the same, that is, the transfer function is invariant under similarity transformations. We have

$$
\begin{aligned}
\boldsymbol{G}(s) &= \boldsymbol{C}\left(s\boldsymbol{I}_n - \boldsymbol{A}\right)^{-1}\boldsymbol{B} + \boldsymbol{D} \\
&= \tilde{\boldsymbol{C}}\left(s\boldsymbol{I}_n - \tilde{\boldsymbol{A}}\right)^{-1}\tilde{\boldsymbol{B}} + \boldsymbol{D} \\
&= \left[\begin{array}{c|c} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{array}\right] \left[\begin{array}{c|c} s\boldsymbol{I}_r - \boldsymbol{A}_1 & -\boldsymbol{A}_2 \\ \hline \boldsymbol{O} & s\boldsymbol{I}_{n-r} - \boldsymbol{A}_4 \end{array}\right]^{-1} \left[\begin{array}{c} \boldsymbol{B}_1 \\ \hline \boldsymbol{O} \end{array}\right] + \boldsymbol{D} \\
&= \left[\begin{array}{c|c} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{array}\right] \left[\begin{array}{c|c} [s\boldsymbol{I}_r - \boldsymbol{A}_1]^{-1} & [s\boldsymbol{I}_r - \boldsymbol{A}_1]^{-1}\boldsymbol{A}_2[s\boldsymbol{I}_{n-r} - \boldsymbol{A}_4]^{-1} \\ \hline \boldsymbol{O} & [s\boldsymbol{I}_{n-r} - \boldsymbol{A}_4]^{-1} \end{array}\right] \left[\begin{array}{c} \boldsymbol{B}_1 \\ \hline \boldsymbol{O} \end{array}\right] + \boldsymbol{D} \\
&= \boldsymbol{C}_1[s\boldsymbol{I}_r - \boldsymbol{A}_1]^{-1}\boldsymbol{B}_1 + \boldsymbol{D}
\end{aligned} \tag{5.16}
$$

Thus the quadruple $(\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{C}_1, \boldsymbol{D})$ is a realization of $\boldsymbol{G}(s)$ of lower dimension then the realization $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$.

We proceed to check if the pair $(\boldsymbol{A}_1, \boldsymbol{C}_1)$ is observable or not. If it is, then $(\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{C}_1, \boldsymbol{D})$ is a minimal realization of $\boldsymbol{G}(s)$. If, on the other hand, the pair $(\boldsymbol{A}_1, \boldsymbol{C}_1)$ is not observable, we apply a dual version of Theorem 5.4 to the pair $(\boldsymbol{A}_1, \boldsymbol{C}_1)$ to extract from it an observable part. This observable part along with corresponding input submatrix and the matrix $\boldsymbol{D}$ form a minimal realization of $\boldsymbol{G}(s)$.

We can obtain a minimal realization of a given transfer function matrix using MATLAB's function `ss`. We illustrate the use of this function with a numerical example.

Example 5.6 We obtain a minimal realization of the transfer function matrix considered in Examples 1.7 and 1.8. Recall that

$$
\boldsymbol{G}(s) = \begin{bmatrix} \frac{1}{s(s+2)} & \frac{2s-1}{s(s+2)} & \frac{s-1}{s(s+2)} \\ -\frac{1}{s+2} & \frac{1-s}{s^2+3s+2} & \frac{1}{s^2+3s+2} \end{bmatrix}.
$$

We represent the above transfer function matrix as

```
G=[tf([1],[1 2 0]) tf([2 -1],[1 2 0]) tf([1 -1],[1 2 0]);...
   tf([-1],[1 2]) tf([-1 1],[1 3 2])  tf([1],[1 3 2])]
```

We then type in

```
sys = ss(G,'min')
```

to obtain a fourth dimensional state-space realization of $\boldsymbol{G}(s)$, where

$$\begin{aligned}\boldsymbol{A} &= \begin{bmatrix} -0.2586 & -0.4843 & 0.1328 & -0.3330 \\ 0.1033 & -1.8720 & 0.7555 & 0.3066 \\ -1.1740 & 0.5232 & -1.1520 & 0.6336 \\ 1.0110 & -0.0522 & 1.1670 & -1.7180 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 0.4118 & -0.3924 & -0.4021 \\ 0.6211 & 0.9522 & 0.1655 \\ -0.4053 & -0.9915 & -0.2931 \\ -0.5748 & 1.1860 & 0.8804 \end{bmatrix}, \\ \boldsymbol{C} &= \begin{bmatrix} -0.4876 & -0.2338 & -1.5020 & 0.4572 \\ -0.8541 & -1.1970 & -0.0004 & -0.1651 \end{bmatrix}, \ \boldsymbol{D} = \boldsymbol{O}.\end{aligned}$$

It is easy to check, using, for example, MATLAB's functions `ctrb` and `obsv` that the pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable and the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, thus confirming the fact that the above quadruple is indeed a minimal realization of the given transfer function matrix $\boldsymbol{G}(s)$.

We may have different minimal realizations of a given transfer function matrix. If this is the case, then they are related via a similarity transformation as shown by Kalman [169]. That is, minimal realizations of a given transfer function matrix are equivalent. Indeed, suppose we have two different minimal realizations $(\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{C}_1, \boldsymbol{D}_1)$ and $(\boldsymbol{A}_2, \boldsymbol{B}_2, \boldsymbol{C}_2, \boldsymbol{D}_2)$ of a given proper transfer function matrix $\boldsymbol{G}(s)$. Then

$$\boldsymbol{G}(s) = \boldsymbol{C}_1(s\boldsymbol{I}_n - \boldsymbol{A}_1)^{-1}\boldsymbol{B}_1 + \boldsymbol{D}_1 = \boldsymbol{C}_2(s\boldsymbol{I}_n - \boldsymbol{A}_2)^{-1}\boldsymbol{B}_2 + \boldsymbol{D}_2$$

and hence, we have to have

$$\boldsymbol{D}_1 = \boldsymbol{D}_2$$

and

$$\boldsymbol{C}_1(s\boldsymbol{I}_n - \boldsymbol{A}_1)^{-1}\boldsymbol{B}_1 = \boldsymbol{C}_2(s\boldsymbol{I}_n - \boldsymbol{A}_2)^{-1}\boldsymbol{B}_2.$$

Taking the inverse Laplace transform of both sides yields

$$\boldsymbol{C}_1 e^{\boldsymbol{A}_1 t}\boldsymbol{B}_1 = \boldsymbol{C}_2 e^{\boldsymbol{A}_2 t}\boldsymbol{B}_2.$$

Taking into account that

$$e^{\boldsymbol{A}t} = \boldsymbol{I} + t\boldsymbol{A} + \frac{t^2}{2!}\boldsymbol{A}^2 + \cdots = \sum_{k=0}^{\infty}\frac{t^k}{k!}\boldsymbol{A}^k$$

we obtain

$$\sum_{k=0}^{\infty}\frac{t^k}{k!}\boldsymbol{C}_1\boldsymbol{A}_1^k\boldsymbol{B}_1 = \sum_{k=0}^{\infty}\frac{t^k}{k!}\boldsymbol{C}_2\boldsymbol{A}_2^k\boldsymbol{B}_2,$$

that is,

$$\boldsymbol{C}_1\boldsymbol{A}_1^k\boldsymbol{B}_1 = \boldsymbol{C}_2\boldsymbol{A}_2^k\boldsymbol{B}_2, \quad k = 0, 1, 2, \ldots \tag{5.17}$$

The matrices $\boldsymbol{C}_i\boldsymbol{A}_i^k\boldsymbol{B}_i$ are called *Markov parameters*. Thus Markov parameters of different realizations of a given transfer function matrix are equal.

Using (5.17), we obtain

$$\begin{bmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_1\boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{C}_1\boldsymbol{A}_1^{n-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{A}_1\boldsymbol{B}_1 & \cdots & \boldsymbol{A}_1^{n-1}\boldsymbol{B}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_2 \\ \boldsymbol{C}_2\boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{C}_2\boldsymbol{A}_2^{n-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_2 & \boldsymbol{A}_2\boldsymbol{B}_2 & \cdots & \boldsymbol{A}_2^{n-1}\boldsymbol{B}_2 \end{bmatrix}. \tag{5.18}$$

Let

$$\boldsymbol{\mathcal{U}}_i = \begin{bmatrix} \boldsymbol{B}_i & \boldsymbol{A}_i\boldsymbol{B}_i & \cdots & \boldsymbol{A}_i^{n-1}\boldsymbol{B}_i \end{bmatrix}, \quad i = 1, 2$$

and

$$\boldsymbol{\mathcal{V}}_i = \begin{bmatrix} \boldsymbol{C}_i \\ \boldsymbol{C}_i\boldsymbol{A}_i \\ \vdots \\ \boldsymbol{C}_i\boldsymbol{A}_i^{n-1} \end{bmatrix}, \quad i = 1, 2.$$

Then, we represent (5.18) in a compact format as

$$\boldsymbol{\mathcal{V}}_1\boldsymbol{\mathcal{U}}_1 = \boldsymbol{\mathcal{V}}_2\boldsymbol{\mathcal{U}}_2. \tag{5.19}$$

Pre-multiplying both sides of (5.19) by $\boldsymbol{\mathcal{V}}_1^\top$ gives

$$\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_1\boldsymbol{\mathcal{U}}_1 = \boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_2\boldsymbol{\mathcal{U}}_2. \tag{5.20}$$

Both realizations are minimal, therefore their controllability and observability matrices have full ranks, that is,

$$\operatorname{rank}\boldsymbol{\mathcal{U}}_i = \operatorname{rank}\boldsymbol{\mathcal{V}}_i = n, \quad i = 1, 2.$$

Hence, the $n \times n$ matrix

$$\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_1$$

is invertible. We pre-multiply both sides of (5.20) by

$$\left[\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_1\right]^{-1}$$

to obtain

$$\boldsymbol{\mathcal{U}}_1 = \left[\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_1\right]^{-1}\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_2\boldsymbol{\mathcal{U}}_2. \tag{5.21}$$

Let

$$\boldsymbol{T}_1 = \left[\boldsymbol{\mathcal{V}}_1^\top \boldsymbol{\mathcal{V}}_1\right]^{-1} \boldsymbol{\mathcal{V}}_1^\top \boldsymbol{\mathcal{V}}_2, \tag{5.22}$$

then we represent (5.21) as

$$\boldsymbol{\mathcal{U}}_1 = \boldsymbol{T}_1 \boldsymbol{\mathcal{U}}_2.$$

We re-write the above as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{A}_1\boldsymbol{B}_1 & \cdots & \boldsymbol{A}_1^{n-1}\boldsymbol{B}_1 \end{bmatrix} &= \boldsymbol{T}_1 \begin{bmatrix} \boldsymbol{B}_2 & \boldsymbol{A}_2\boldsymbol{B}_2 & \cdots & \boldsymbol{A}_2^{n-1}\boldsymbol{B}_2 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{T}_1\boldsymbol{B}_2 & \boldsymbol{T}_1\boldsymbol{A}_2\boldsymbol{T}_1^{-1}\boldsymbol{T}_1\boldsymbol{B}_2 & \cdots & \boldsymbol{T}_1\boldsymbol{A}_2^{n-1}\boldsymbol{T}_1^{-1}\boldsymbol{T}_1\boldsymbol{B}_2 \end{bmatrix}. \end{aligned}$$

taking into account that

$$\boldsymbol{T}_1\boldsymbol{A}_2^k\boldsymbol{T}_1^{-1} = \left(\boldsymbol{T}_1\boldsymbol{A}_2\boldsymbol{T}_1^{-1}\right)^k,$$

we conclude that the pairs $(\boldsymbol{A}_1, \boldsymbol{B}_1)$ and $(\boldsymbol{A}_2, \boldsymbol{B}_2)$ are equivalent, that is, they are related via the similarity transformation. In other words,

$$\boldsymbol{A}_1 = \boldsymbol{T}_1\boldsymbol{A}_2\boldsymbol{T}_1^{-1} \quad \text{and} \quad \boldsymbol{B}_1 = \boldsymbol{T}_1\boldsymbol{B}_2.$$

The transformation matrix (5.22) is expressed in terms of the observability matrices. We can express the transformation matrix in terms of the controllability matrices as well. Indeed, post-multiplying both sides of (5.22) by $\boldsymbol{\mathcal{U}}_1^\top$ gives

$$\boldsymbol{\mathcal{V}}_1\boldsymbol{\mathcal{U}}_1\boldsymbol{\mathcal{U}}_1^\top = \boldsymbol{\mathcal{V}}_2\boldsymbol{\mathcal{U}}_2\boldsymbol{\mathcal{U}}_1^\top. \tag{5.23}$$

The $n \times n$ matrix is invertible. We post-multiply both sides of (5.23) by the matrix

$$[\boldsymbol{\mathcal{U}}_1\boldsymbol{\mathcal{U}}_1^\top]^{-1}$$

to obtain

$$\boldsymbol{\mathcal{V}}_1 = \boldsymbol{\mathcal{V}}_2\boldsymbol{\mathcal{U}}_2\boldsymbol{\mathcal{U}}_1^\top[\boldsymbol{\mathcal{U}}_1\boldsymbol{\mathcal{U}}_1^\top]^{-1}. \tag{5.24}$$

Let

$$\boldsymbol{T}_2^{-1} = \boldsymbol{\mathcal{U}}_2\boldsymbol{\mathcal{U}}_1^\top[\boldsymbol{\mathcal{U}}_1\boldsymbol{\mathcal{U}}_1^\top]^{-1}, \tag{5.25}$$

then, we can represent (5.24) as

$$\boldsymbol{\mathcal{V}}_1 = \boldsymbol{\mathcal{V}}_2\boldsymbol{T}_2^{-1}.$$

We re-write the above as

$$\begin{aligned}\begin{bmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_1\boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{C}_1\boldsymbol{A}_1^{n-1}\end{bmatrix} &= \begin{bmatrix} \boldsymbol{C}_2 \\ \boldsymbol{C}_2\boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{C}_2\boldsymbol{A}_2^{n-1}\end{bmatrix}\boldsymbol{T}_2^{-1} \\ &= \begin{bmatrix} \boldsymbol{C}_2\boldsymbol{T}_2^{-1} \\ \boldsymbol{C}_2\boldsymbol{T}_2^{-1}\boldsymbol{T}_2\boldsymbol{A}_2\boldsymbol{T}_2^{-1} \\ \vdots \\ \boldsymbol{C}_2\boldsymbol{T}_2^{-1}\boldsymbol{T}_2\boldsymbol{A}_2^{n-1}\boldsymbol{T}_2^{-1}\end{bmatrix}.\end{aligned}$$

Taking into account that

$$\boldsymbol{T}_2\boldsymbol{A}_2^k\boldsymbol{T}_2^{-1} = \left(\boldsymbol{T}_2\boldsymbol{A}_2\boldsymbol{T}_2^{-1}\right)^k$$

we conclude that the pairs $(\boldsymbol{A}_1, \boldsymbol{C}_1)$ and $(\boldsymbol{A}_2, \boldsymbol{C}_2)$ are equivalent, that is,

$$\boldsymbol{A}_1 = \boldsymbol{T}_2\boldsymbol{A}_2\boldsymbol{T}_2^{-1} \quad \text{and} \quad \boldsymbol{C}_1 = \boldsymbol{C}_2\boldsymbol{T}_2^{-1}.$$

It is easy to verify that

$$\boldsymbol{T}_1 = \boldsymbol{T}_2.$$

Indeed, it is enough to show that

$$\boldsymbol{T}_1\boldsymbol{T}_2^{-1} = \boldsymbol{I}_n.$$

Using (5.22) and (5.25) we obtain,

$$\boldsymbol{T}_1\boldsymbol{T}_2^{-1} = \left[\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_1\right]^{-1}\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_2\boldsymbol{\mathcal{U}}_2\boldsymbol{\mathcal{U}}_1^\top[\boldsymbol{\mathcal{U}}_1\boldsymbol{\mathcal{U}}_1^\top]^{-1}.$$

Substituting into the above (5.19), that is, $\boldsymbol{\mathcal{V}}_2\boldsymbol{\mathcal{U}}_2 = \boldsymbol{\mathcal{V}}_1\boldsymbol{\mathcal{U}}_1$, gives

$$\begin{aligned}\boldsymbol{T}_1\boldsymbol{T}_2^{-1} &= \left[\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_1\right]^{-1}\boldsymbol{\mathcal{V}}_1^\top\boldsymbol{\mathcal{V}}_1\boldsymbol{\mathcal{U}}_1\boldsymbol{\mathcal{U}}_1^\top[\boldsymbol{\mathcal{U}}_1\boldsymbol{\mathcal{U}}_1^\top]^{-1} \\ &= \boldsymbol{I}_n.\end{aligned}$$

Hence,

$$\boldsymbol{T}_1 = \boldsymbol{T}_2 = \boldsymbol{T}.$$

In view of the above, we conclude that different minimal realizations of the same transfer function matrix are equivalent, that is,

$$\boldsymbol{A}_1 = \boldsymbol{T}\boldsymbol{A}_2\boldsymbol{T}^{-1}, \quad \boldsymbol{B}_1 = \boldsymbol{T}\boldsymbol{B}_2, \quad \boldsymbol{C}_1 = \boldsymbol{C}_2\boldsymbol{T}^{-1}, \quad \boldsymbol{D}_1 = \boldsymbol{D}_2, \tag{5.26}$$

where the transformation matrix $\boldsymbol{T}$ can be computed using (5.22) or (5.25).

5.4 Companion Forms

In this section, we present two types of special forms of a given linear dynamical system model represented by a quadruple $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$. These forms are obtained by a change of state variables using similarity transformations. They reveal structural properties of the model. We use the above mentioned forms to construct algorithms for pole placement. The pole placement algorithms are then used to design state-feedback controllers, as well as observers. We begin by introducing the controller companion form, or for short, controller form.

5.4.1 Controller Form

We begin by considering a single-input, continuous-time, LTI model,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t),$$

where the pair $(\boldsymbol{A}, \boldsymbol{b})$ is assumed to be reachable, or equivalently, controllable. This means that

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{b} & \boldsymbol{A}\boldsymbol{b} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b} \end{bmatrix} = n.$$

We select the last row of the inverse of the controllability matrix. Let $\boldsymbol{q}_1$ be that row. Next, we form the matrix

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_1\boldsymbol{A} \\ \vdots \\ \boldsymbol{q}_1\boldsymbol{A}^{n-1} \end{bmatrix}. \tag{5.27}$$

Note that $\boldsymbol{T}$ is invertible. Indeed, because

$$\boldsymbol{T}\begin{bmatrix} \boldsymbol{b} & \boldsymbol{A}\boldsymbol{b} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & x \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & x & \cdots & x & x \end{bmatrix},$$

this implies that $\boldsymbol{T}$ must be nonsingular. In the above, we used the symbol x to denote "don't care", unspecified, scalars in our present discussion. Consider now the state variable transformation, $\tilde{\boldsymbol{x}} = \boldsymbol{T}\boldsymbol{x}$. Then, in the new coordinates, the system model is

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}}(t) &= \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1}\tilde{\boldsymbol{x}}(t) + \boldsymbol{T}\boldsymbol{b}u(t) \\ &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{b}}u(t). \end{aligned}$$

The matrices $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{b}}$ have particular structures that we now discuss. We first analyze the structure of $\tilde{\boldsymbol{b}}$. Because $\boldsymbol{q}_1$ is the last row of the inverse of the controllability matrix, we have

$$\boldsymbol{q}_1\boldsymbol{b} = \boldsymbol{q}_1\boldsymbol{A}\boldsymbol{b} = \cdots = \boldsymbol{q}_1\boldsymbol{A}^{n-2}\boldsymbol{b} = 0,$$

and

$$\boldsymbol{q}_1\boldsymbol{A}^{n-1}\boldsymbol{b} = 1.$$

Hence,

$$\tilde{\boldsymbol{b}} = \boldsymbol{T}\boldsymbol{b} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{b} \\ \boldsymbol{q}_1\boldsymbol{A}\boldsymbol{b} \\ \vdots \\ \boldsymbol{q}_1\boldsymbol{A}^{n-2}\boldsymbol{b} \\ \boldsymbol{q}_1\boldsymbol{A}^{n-1}\boldsymbol{b} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

The structure of $\tilde{\boldsymbol{A}}$ is revealed by considering the relation $\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \tilde{\boldsymbol{A}}$ represented as

$$\boldsymbol{T}\boldsymbol{A} = \tilde{\boldsymbol{A}}\boldsymbol{T}.$$

The left-hand side of the above matrix equation is

$$\boldsymbol{T}\boldsymbol{A} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{A} \\ \boldsymbol{q}_1\boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{q}_1\boldsymbol{A}^{n-1} \\ \boldsymbol{q}_1\boldsymbol{A}^{n} \end{bmatrix}.$$

By the Cayley-Hamilton theorem,

$$\boldsymbol{A}^n = -a_0\boldsymbol{I}_n - a_1\boldsymbol{A} - \cdots - a_{n-1}\boldsymbol{A}^{n-1},$$

and hence

$$\boldsymbol{q}_1\boldsymbol{A}^n = -a_0\boldsymbol{q}_1 - a_1\boldsymbol{q}_1\boldsymbol{A} - \cdots - a_{n-1}\boldsymbol{q}_1\boldsymbol{A}^{n-1}.$$

Comparing both sides of the equation $\boldsymbol{T}\boldsymbol{A} = \tilde{\boldsymbol{A}}\boldsymbol{T}$, and taking into account the Cayley-Hamilton theorem, we get

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix}.$$

Note that the coefficients of the characteristic polynomial of $\boldsymbol{A}$ are immediately apparent by inspecting the last row of $\tilde{\boldsymbol{A}}$. There are other benefits of the pair $(\boldsymbol{A}, \boldsymbol{b})$ being given in its controller form that we discuss in Section 5.5. We illustrate the above algorithm for reducing a single-input system into the controller form with a numerical example.

Example 5.7 Consider the following dynamical system model:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u.$$

The controllability matrix of the pair $(\boldsymbol{A}, \boldsymbol{b})$ is

$$\begin{bmatrix} \boldsymbol{b} & \boldsymbol{A}\boldsymbol{b} & \boldsymbol{A}^2\boldsymbol{b} & \boldsymbol{A}^3\boldsymbol{b} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The pair $(\boldsymbol{A}, \boldsymbol{b})$ is controllable because the controllability matrix is nonsingular. We find the last row of the inverse of the controllability matrix to be $\boldsymbol{q}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$. Hence the transformation matrix is

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_1\boldsymbol{A} \\ \boldsymbol{q}_1\boldsymbol{A}^2 \\ \boldsymbol{q}_1\boldsymbol{A}^3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

The matrices $\boldsymbol{A}$ and $\boldsymbol{b}$ in the new coordinates have the form,

$$\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -3 & 0 \end{bmatrix},$$

and

$$\boldsymbol{T}\boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We generalize the above method of transforming the given pair $(\boldsymbol{A}, \boldsymbol{B})$ into the controller form for multi-input systems. We consider a controllable pair $(\boldsymbol{A}, \boldsymbol{B})$ where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, $m \le n$, and $\operatorname{rank} \boldsymbol{B} = m$. The last assumption we made is just for convenience to simplify further manipulations. This assumption implies that all inputs are mutually independent, which is usually the case in practical applications. We first form the controllability matrix of the pair $(\boldsymbol{A}, \boldsymbol{B})$ and represent it as

$$\begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_m & \boldsymbol{A}\boldsymbol{b}_1 & \boldsymbol{A}\boldsymbol{b}_2 & \cdots & \boldsymbol{A}\boldsymbol{b}_m & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b}_1 & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b}_m \end{bmatrix}, \tag{5.28}$$

where $\boldsymbol{b}_i$, $i = 1, 2, \ldots, m$, is the i-th column of $\boldsymbol{B}$. The above controllability matrix is of full rank because we assumed that the pair $(\boldsymbol{A}, \boldsymbol{B})$ is controllable. Thus, we can select n linearly independent columns from (5.28). We select these columns proceeding from left to right. Then, we rearrange the selected columns to form a $n \times n$ nonsingular matrix $\boldsymbol{L}$ of the form

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{A}\boldsymbol{b}_1 & \cdots & \boldsymbol{A}^{d_1-1}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{A}^{d_2-1}\boldsymbol{b}_2 & \cdots & \boldsymbol{b}_m & \cdots & \boldsymbol{A}^{d_m-1}\boldsymbol{b}_m \end{bmatrix}.$$

We call the m integers d_i the *controllability indices* of the pair $(\boldsymbol{A}, \boldsymbol{B})$. Note that

$$\sum_{i=1}^{m} d_i = n.$$

We call the integer

$$d = \max\{d_i : i = 1, 2, \ldots, m\},$$

the *controllability index* of the pair $(\boldsymbol{A}, \boldsymbol{B})$. Observe that all m columns of $\boldsymbol{B}$ are present in $\boldsymbol{L}$ because we assumed that $\boldsymbol{B}$ is of full rank and the way we selected mutually independent columns from the controllability matrix (5.28) to construct the matrix $\boldsymbol{L}$. Furthermore, if the column $\boldsymbol{A}^k\boldsymbol{b}_j$ is present in $\boldsymbol{L}$, then so is $\boldsymbol{A}^{k-1}\boldsymbol{b}_j$.

To proceed further, we let

$$\sigma_k = \sum_{i=1}^{k} d_i, \quad \text{for} \quad k = 1, 2, \ldots, m.$$

Note that $\sigma_1 = d_1$, $\sigma_2 = d_1 + d_2$, $\ldots$, $\sigma_m = \sum_{i=1}^{m} d_i = n$. We select m rows from $\boldsymbol{L}^{-1}$. We denote the selected rows by $\boldsymbol{q}_k$, where $\boldsymbol{q}_k$ is the σ_k-th row of $\boldsymbol{L}^{-1}$. Next, we form the $n \times n$

matrix

$$
\boldsymbol{T} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_1\boldsymbol{A} \\ \vdots \\ \boldsymbol{q}_1\boldsymbol{A}^{d_1-1} \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_2\boldsymbol{A}^{d_2-1} \\ \vdots \\ \boldsymbol{q}_m \\ \vdots \\ \boldsymbol{q}_m\boldsymbol{A}^{d_m-1} \end{bmatrix}.
$$

We can show that $\boldsymbol{T}$ is nonsingular by computing the product $\boldsymbol{TL}$ and noticing that $\det(\boldsymbol{TL}) = \pm 1$. We define a state-variable transformation $\tilde{\boldsymbol{x}} = \boldsymbol{Tx}$, and then using similar arguments as in the single-input case, we arrive at the special structure of the pair

$$
\left(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}\right) = \left(\boldsymbol{TAT}^{-1}, \boldsymbol{TB}\right),
$$

where

$$
\tilde{\boldsymbol{A}} = \left[\begin{array}{ccccc|cccc|c|cccc}
0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & & 0 & 0 & \cdots & 0 \\
 & & & \ddots & & \vdots & & & \vdots & & \vdots & & & \vdots \\
0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & & 0 & 0 & \cdots & 0 \\
x & x & x & \cdots & x & x & x & \cdots & x & \cdots & x & x & \cdots & x \\
\hline
0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & & & & \vdots & \vdots & & \ddots & \vdots & & \vdots & & & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & & 0 & 0 & \cdots & 0 \\
x & x & x & \cdots & x & x & x & \cdots & x & \cdots & x & x & \cdots & x \\
\hline
\vdots & & & & & & \vdots & & & \ddots & & & \vdots & \\
\hline
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 0 \\
\vdots & & & & \vdots & \vdots & & & \vdots & & \vdots & & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & & 0 & 0 & \cdots & 1 \\
x & x & x & \cdots & x & x & x & \cdots & x & \cdots & x & x & \cdots & x
\end{array}\right].
$$

Note that the m diagonal blocks of $\tilde{\boldsymbol{A}}$ have the structure of the controller form as in the single-input case. The dimensions of the diagonal blocks are $d_i \times d_i$, $i = 1, 2, \ldots, m$. The off-diagonal blocks consist of zero elements except for their respective last rows. Thus, all information about $\tilde{\boldsymbol{A}}$, as well as $\boldsymbol{A}$, can be derived from knowledge of the controllability indices, and the m nonzero rows containing the elements marked using x, of the matrix $\tilde{\boldsymbol{A}}$.

The matrix $\tilde{\boldsymbol{B}}$ has the form,

$$\tilde{\boldsymbol{B}} = \left[\begin{array}{ccccc} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 1 & x & x & x & x \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 1 & x & x & x \\ \hline \vdots & & & & \vdots \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

The nonzero rows of $\tilde{\boldsymbol{B}}$ correspond to the rows of $\tilde{\boldsymbol{A}}$ containing the elements marked using x.

We illustrate the above algorithm for reducing a multi-input system model into the controller form with a numerical example.

Example 5.8 Consider the pair $(\boldsymbol{A}, \boldsymbol{B})$, where

$$\boldsymbol{A} = \begin{bmatrix} -1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 1 & 1 & -1 & 2 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

We transform the above pair into its controller companion form. Following the algorithm, we find,

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{A}\boldsymbol{b}_1 & \boldsymbol{A}\boldsymbol{b}_2 \end{bmatrix} = 4.$$

Hence $d_1 = d_2 = 2$. We next form the matrix $\boldsymbol{L}$,

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{A}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{A}\boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 3 & 0 & 1 \\ 1 & 4 & 0 & 0 \\ 0 & 2 & 1 & 3 \end{bmatrix},$$

and

$$\boldsymbol{L}^{-1} = \begin{bmatrix} 0.20 & 0.60 & -0.40 & -0.20 \\ -0.05 & -0.15 & 0.35 & 0.05 \\ 0.85 & -0.45 & 0.05 & 0.15 \\ -0.25 & 0.25 & -0.25 & 0.25 \end{bmatrix}.$$

Hence

$$\boldsymbol{q}_1 = \begin{bmatrix} -0.05 & -0.15 & 0.35 & 0.05 \end{bmatrix},$$

and

$$\boldsymbol{q}_2 = \begin{bmatrix} -0.25 & 0.25 & -0.25 & 0.25 \end{bmatrix}.$$

The transformation matrix is

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_1\boldsymbol{A} \\ \boldsymbol{q}_2 \\ \boldsymbol{q}_2\boldsymbol{A} \end{bmatrix} = \begin{bmatrix} -0.05 & -0.15 & 0.35 & 0.05 \\ -0.05 & -0.15 & 1.35 & 0.05 \\ -0.25 & 0.25 & -0.25 & 0.25 \\ 0.75 & 0.25 & -1.25 & 0.25 \end{bmatrix}.$$

The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in the new coordinates have the form,

$$\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ -4 & 5 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 3 & -3 & 4 & 1 \end{array}\right] \quad \text{and} \quad \boldsymbol{T}\boldsymbol{B} = \left[\begin{array}{cc} 0 & 0 \\ 1 & 0 \\ \hline 0 & 0 \\ 0 & 1 \end{array}\right].$$

5.4.2 Observer Form

In this subsection, we use duality and the algorithms for reducing reachable systems into controller forms to derive analogous results for observable systems. For the sake of simplicity and without loss of generality, we consider an observable dynamical system model

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}}(t) & = & \boldsymbol{A}\boldsymbol{x}(t) \\ \boldsymbol{y}(t) & = & \boldsymbol{C}\boldsymbol{x}(t), \end{array}\right\} \tag{5.29}$$

or, equivalently, the observable pair $(\boldsymbol{A}, \boldsymbol{C})$, where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{C} \in \mathbb{R}^{p\times n}$, $p \le n$, and $\operatorname{rank} \boldsymbol{C} = p$. We omitted the input variable $\boldsymbol{u}$ for convenience because it plays no role in the subsequent discussion. We next consider the *dual system* model associated with the system (5.29),

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{A}^\top \boldsymbol{z}(t) + \boldsymbol{C}^\top \boldsymbol{v}(t).$$

Because the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, the dual pair $\left(\boldsymbol{A}^\top, \boldsymbol{C}^\top\right)$ is reachable. Therefore, the pair $\left(\boldsymbol{A}^\top, \boldsymbol{C}^\top\right)$ can be transformed into the controller form using the algorithm from the previous subsection. We then take the dual of the result to get

$$\left.\begin{array}{rcl} \dot{\hat{\boldsymbol{x}}}(t) &=& \hat{\boldsymbol{A}}\hat{\boldsymbol{x}} \\ \boldsymbol{y}(t) &=& \hat{\boldsymbol{C}}\hat{\boldsymbol{x}}, \end{array}\right\}$$

where

$$\hat{\boldsymbol{A}} = \begin{bmatrix} \hat{\boldsymbol{A}}_{11} & \hat{\boldsymbol{A}}_{12} & \cdots & \hat{\boldsymbol{A}}_{1p} \\ \hat{\boldsymbol{A}}_{21} & \hat{\boldsymbol{A}}_{22} & \cdots & \hat{\boldsymbol{A}}_{2p} \\ \vdots & & & \vdots \\ \hat{\boldsymbol{A}}_{p1} & \hat{\boldsymbol{A}}_{p2} & \cdots & \hat{\boldsymbol{A}}_{pp} \end{bmatrix}.$$

Each diagonal submatrix $\hat{\boldsymbol{A}}_{ii} \in \mathbb{R}^{\hat{d}_i \times \hat{d}_i}$ of $\hat{\boldsymbol{A}}$ has the form

$$\hat{\boldsymbol{A}}_{ii} = \begin{bmatrix} 0 & 0 & \cdots & 0 & x \\ 1 & 0 & \cdots & 0 & x \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & x \end{bmatrix},$$

which is just the transpose of the controller form for the single-input case. The integers $\hat{d}_i$, $i = 1, 2, \ldots, p$ are the controllability indices of the controllable pair $\left(\boldsymbol{A}^\top, \boldsymbol{C}^\top\right)$. The off-diagonal blocks, $j \ne i$, have the form

$$\hat{\boldsymbol{A}}_{ji} = \begin{bmatrix} 0 & 0 & \cdots & 0 & x \\ 0 & 0 & \cdots & 0 & x \\ \vdots & & & & \vdots \\ 0 & 0 & \cdots & 0 & x \end{bmatrix},$$

that is, they consist of zeros except for their last columns. The matrix $\hat{\boldsymbol{C}}$ consists of p blocks and has the form

$$\hat{\boldsymbol{C}} = \left[\begin{array}{cccc|cccc|c|cccc} 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & x & 0 & 0 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & & & \vdots & & \vdots & & & \vdots \\ 0 & 0 & \cdots & x & 0 & 0 & \cdots & x & \cdots & 0 & 0 & \cdots & 1 \end{array}\right].$$

The p controllability indices of the pair $\left(\boldsymbol{A}^\top, \boldsymbol{C}^\top\right)$ are called the *observability indices* of the pair $(\boldsymbol{A}, \boldsymbol{C})$. Note that

$$\sum_{i=1}^{p} \hat{d}_i = n.$$

Similarly as in the controller form, we define the *observability index* of the pair $(\boldsymbol{A}, \boldsymbol{C})$ to be the integer

$$\hat{d} = \max\{\hat{d}_i : i = 1, 2, \ldots, p\}.$$

We illustrate the above algorithm for transforming an observable pair into the observer form with a numerical example.

Example 5.9 Consider the observable pair $(\boldsymbol{A}, \boldsymbol{C})$,

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & -21 & 5 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

We transform the pair $\left(\boldsymbol{A}^\top, \boldsymbol{C}^\top\right)$ into the controller form. Thus, we first form the controllability matrix for $\left(\boldsymbol{A}^\top, \boldsymbol{C}^\top\right)$, then proceeding from left to right, we select the first $n = 4$ linearly independent columns. They are

$$\boldsymbol{c}_1^\top, \boldsymbol{c}_2^\top, \boldsymbol{A}^\top \boldsymbol{c}_1^\top, \left(\boldsymbol{A}^\top\right)^2 \boldsymbol{c}_1^\top.$$

Therefore, the observability indices are: $\hat{d}_1 = 3$ and $\hat{d}_2 = 1$. Hence,

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{c}_1^\top & \boldsymbol{A}^\top \boldsymbol{c}_1^\top & \left(\boldsymbol{A}^\top\right)^2 \boldsymbol{c}_1^\top & \boldsymbol{c}_2^\top \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$\boldsymbol{L}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The two vectors that we need to form the desired transformation are

$$\boldsymbol{q}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{q}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}.$$

The transformation matrix is

$$\hat{\boldsymbol{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 2 & 3 & -21 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The observer form of the pair $(\boldsymbol{A}, \boldsymbol{C})$ is given by the matrices

$$\hat{\boldsymbol{A}} = \hat{\boldsymbol{T}}^{-T} \boldsymbol{A} \hat{\boldsymbol{T}}^{\top} = \left[\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 3 & 0 \\ \hline 0 & 0 & -105 & 5 \end{array}\right] \quad \text{and} \quad \hat{\boldsymbol{C}} = \boldsymbol{C} \hat{\boldsymbol{T}}^{\top} = \left[\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 0 & 0 & -21 & 1 \end{array}\right].$$

5.5 Linear State-Feedback Control

In the following sections, we discuss methods for a feedback controller design for linear systems. The design process involves three steps. The first step is the design of a state-feedback control law. In this step, we assume that we have all states available to us for feedback purposes. In general, this may not be the case. The assumption that all the states are available allows us to proceed with the state-feedback control law design. We then proceed with the second step that involves the design of an observer to estimate the state vector. The observer should use only the input and output of the plant to be controlled to generate an estimate of the plant state. The last step is to combine the first two steps in that the control law, designed in the first step, uses the state estimate instead of the true state vector. The result of this step is a combined controller-observer compensator. We first discuss the problem of designing a linear state-feedback control law. In the following sections, we present methods for constructing observers and combined controller-observer compensators.

The linear state-feedback control law, for a system modeled by $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$, is the feedback of a linear combination of all the state variables, and has the form

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x} + \boldsymbol{v},$$

where $\boldsymbol{K} \in \mathbb{R}^{m \times n}$ is a constant matrix. Figure 5.4 illustrates the linear state-feedback applied to the linear system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$.

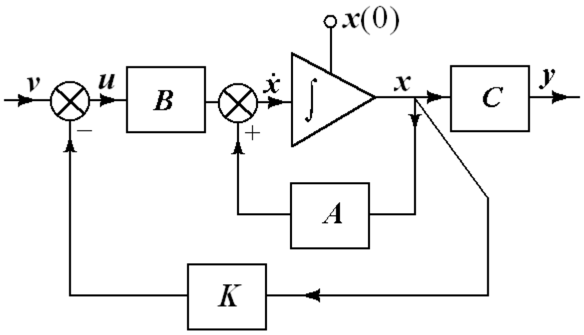


Figure 5.4: Linear state-feedback.

The closed-loop system is described as

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\,\boldsymbol{x} + \boldsymbol{B}\boldsymbol{v}.$$

The poles of the closed-loop system are the roots of the characteristic equation

$$\det(s\boldsymbol{I}_n - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}) = 0.$$

The linear state-feedback control law design consists of selecting the gains

$$k_{ij}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n,$$

so that the roots of the closed-loop characteristic equation

$$\det(s\boldsymbol{I}_n - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}) = 0$$

are in desirable locations in the complex plane. We assume that a designer has made the selection of the desired poles of the closed-loop system and they are:

$$s_1, s_2, \ldots, s_n.$$

The desired closed-loop poles can be real or complex. If they are complex, then they must come in complex conjugate pairs. This is because we use only real gains k_{ij}. Having selected the desired closed-loop poles we can form the desired closed-loop characteristic polynomial,

$$\begin{aligned} \alpha_c(s) &= (s - s_1)(s - s_2)\cdots(s - s_n) \\ &= s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1 s + \alpha_0. \end{aligned}$$

Our goal is to select a feedback matrix $\boldsymbol{K}$ such that

$$\det\left(s\boldsymbol{I}_n - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right) = s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1 s + \alpha_0.$$

The above problem is also referred to as the *pole placement* problem. We first discuss the pole placement problem for the single-input plants. In this case $\boldsymbol{K} = \boldsymbol{k} \in \mathbb{R}^{1\times n}$. The solution to the problem is easily obtained if the pair $(\boldsymbol{A}, \boldsymbol{b})$ is already in the controller companion form. In such a case we have

$$\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 - k_1 & -a_1 - k_2 & -a_2 - k_3 & \cdots & -a_{n-2} - k_{n-1} & -a_{n-1} - k_n \end{bmatrix}.$$

Hence, the desired gains are

$$\begin{aligned} k_1 &= \alpha_0 - a_0, \\ k_2 &= \alpha_1 - a_1, \\ &\vdots \\ k_n &= \alpha_{n-1} - a_{n-1}. \end{aligned}$$

If the pair $(\boldsymbol{A}, \boldsymbol{b})$ is not in the controller companion form, we first transform it into the companion form, and then compute the gain vector $\tilde{\boldsymbol{k}}$ such that

$$\det\left(s\boldsymbol{I}_n - \tilde{\boldsymbol{A}} + \tilde{\boldsymbol{b}}\tilde{\boldsymbol{k}}\right) = s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1 s + \alpha_0.$$

Thus,

$$\tilde{\boldsymbol{k}} = \begin{bmatrix} \alpha_0 - a_0 & \alpha_1 - a_1 & \cdots & \alpha_{n-1} - a_{n-1} \end{bmatrix}.$$

Then,

$$\boldsymbol{k} = \tilde{\boldsymbol{k}}\boldsymbol{T},$$

where $\boldsymbol{T}$ is the transformation that brings the pair $(\boldsymbol{A}, \boldsymbol{b})$ into the controller companion form.

We can represent the above formula for the gain matrix in an alternative way. For this note that

$$\begin{aligned} \tilde{\boldsymbol{k}}\boldsymbol{T} &= \begin{bmatrix} \alpha_0 - a_0 & \alpha_1 - a_1 & \cdots & \alpha_{n-1} - a_{n-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_1\boldsymbol{A} \\ \vdots \\ \boldsymbol{q}_1\boldsymbol{A}^{n-1} \end{bmatrix} \\ &= \boldsymbol{q}_1\left(\alpha_0\boldsymbol{I}_n + \alpha_1\boldsymbol{A} + \cdots + \alpha_{n-1}\boldsymbol{A}^{n-1}\right) - \boldsymbol{q}_1\left(a_0\boldsymbol{I}_n + a_1\boldsymbol{A} + \cdots + a_{n-1}\boldsymbol{A}^{n-1}\right). \end{aligned}$$

By the Cayley-Hamilton theorem,

$$\boldsymbol{A}^n = -\left(a_0\boldsymbol{I}_n + a_1\boldsymbol{A} + \cdots + a_{n-1}\boldsymbol{A}^{n-1}\right).$$

Hence,

$$\boldsymbol{k} = \boldsymbol{q}_1\alpha_c\left(\boldsymbol{A}\right).$$

The above expression for the gain row vector was proposed by Ackermann in 1972, and is now referred to as the *Ackermann's formula* for pole placement.

Example 5.10 For the linear dynamical system

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 1 & -1 \\ 1 & -2 \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}u,$$

we will use the Ackermann's formula to design a state-feedback controller, $u = -\boldsymbol{kx}$, such that the closed-loop poles are located at $\{-1, -2\}$.

To use the Ackermann's formula, we first form the controllability matrix of the pair $(\boldsymbol{A}, \boldsymbol{b})$ and then find the last row of its inverse denoted $\boldsymbol{q}_1$. The controllability matrix is

$$\mathcal{U} = \begin{bmatrix} \boldsymbol{b} & \boldsymbol{Ab} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}.$$

Its inverse is

$$\mathcal{U}^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}.$$

Hence, $\boldsymbol{q}_1 = \begin{bmatrix} 1 & -2 \end{bmatrix}$. The desired closed-loop characteristic polynomial is

$$\alpha_c(s) = (s+1)(s+2) = s^2 + 3s + 2.$$

Therefore,

$$\begin{aligned}
\boldsymbol{k} &= \boldsymbol{q}_1\alpha_c(\boldsymbol{A}) \\
&= \boldsymbol{q}_1\left(\boldsymbol{A}^2 + 3\boldsymbol{A} + 2\boldsymbol{I}_2\right) \\
&= \boldsymbol{q}_1\left(\begin{bmatrix} 1 & -1 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 1 & -2 \end{bmatrix} + 3\begin{bmatrix} 1 & -1 \\ 1 & -2 \end{bmatrix} + 2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \\
&= \boldsymbol{q}_1\left(\begin{bmatrix} 0 & 1 \\ -1 & 3 \end{bmatrix} + \begin{bmatrix} 3 & -3 \\ 3 & -6 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\right) \\
&= \begin{bmatrix} 1 & -2 \end{bmatrix}\begin{bmatrix} 5 & -2 \\ 2 & -1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \end{bmatrix}
\end{aligned}$$

We now present a pole placement algorithm for multi-input systems. If the pair $(\boldsymbol{A}, \boldsymbol{B})$ is already in the controller companion form, then we may proceed as follows. First, represent the matrix $\boldsymbol{B}$ as

$$\boldsymbol{B} = \left[\begin{array}{ccccc} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 1 & x & x & x & x \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 1 & x & x & x \\ \hline \vdots & & & & \vdots \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccccc} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 1 & 0 & 0 & 0 \\ \hline \vdots & & & & \vdots \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} 1 & x & \cdots & x & x \\ 0 & 1 & \cdots & x & x \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & x \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = \hat{\boldsymbol{B}}\boldsymbol{\Gamma},$$

where the square and nonsingular matrix $\boldsymbol{\Gamma}$ consists of the nonzero rows of $\boldsymbol{B}$. Next, let

$$\hat{\boldsymbol{K}} = \boldsymbol{\Gamma}\boldsymbol{K}.$$

Notice that

$$\hat{\boldsymbol{B}}\hat{\boldsymbol{K}} = \left[\begin{array}{ccccc} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ \hat{k}_{11} & \hat{k}_{12} & \cdots & \hat{k}_{1\,n-1} & \hat{k}_{1n} \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ \hat{k}_{21} & \hat{k}_{22} & \cdots & \hat{k}_{2\,n-1} & \hat{k}_{2n} \\ \hline \vdots & & & & \vdots \\ \hline 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ \hat{k}_{m1} & \hat{k}_{m2} & \cdots & \hat{k}_{m\,n-1} & \hat{k}_{mn} \end{array}\right],$$

that is, the nonzero rows in the product $\hat{\boldsymbol{B}}\hat{\boldsymbol{K}}$ match the nonzero rows of the matrix $\boldsymbol{A}$ in the controller companion form. We can then, for example, select the gains k_{ij}, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, so that

$$\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K} = \boldsymbol{A} - \hat{\boldsymbol{B}}\hat{\boldsymbol{K}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-2} & -\alpha_{n-1} \end{bmatrix},$$

where

$$\boldsymbol{K} = \boldsymbol{\Gamma}^{-1}\hat{\boldsymbol{K}}.$$

If the pair $(\boldsymbol{A}, \boldsymbol{B})$ is not in the controller companion form, we first transform it into this form, then compute the gain matrix that allocates the closed-loop poles into the desired locations. The gain matrix $\boldsymbol{K}$ that allocates the closed-loop poles into the prespecified locations for the pair $(\boldsymbol{A}, \boldsymbol{B})$ in the original coordinates is then given by

$$\boldsymbol{K} = \boldsymbol{\Gamma}^{-1}\hat{\boldsymbol{K}}\boldsymbol{T},$$

where $\boldsymbol{T}$ is the transformation matrix that brings the pair $(\boldsymbol{A}, \boldsymbol{B})$ into the controller companion form.

Example 5.11 For the pair

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & -21 & 5 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

we use its controller companion form to find the matrix $\boldsymbol{K} \in \mathbb{R}^{2\times 4}$ so that closed-loop poles are located at

$$-2, -3 + j, -3 - j, -4.$$

We first transform the pair $(\boldsymbol{A}, \boldsymbol{B})$ into its controller companion form. We begin by forming the controllability matrix

$$\begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{A}\boldsymbol{b}_1 & \boldsymbol{A}\boldsymbol{b}_2 & \boldsymbol{A}^2\boldsymbol{b}_1 & \cdots \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & \cdots \\ 0 & 1 & 0 & 5 & 0 & \cdots \end{bmatrix}.$$

We then select, proceeding from left to right, the first four linearly independent columns from the above controllability matrix. We obtain

$$\begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{A}\boldsymbol{b}_1 & \boldsymbol{A}^2\boldsymbol{b}_1 \end{bmatrix}.$$

Thus, the controllability indices are $d_1 = 3$ and $d_2 = 1$. We rearrange the above columns and form the matrix $\boldsymbol{L}$ of the form

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{A}\boldsymbol{b}_1 & \boldsymbol{A}^2\boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} = \boldsymbol{I}_4 = \boldsymbol{L}^{-1}.$$

The two row vectors that we need in constructing the transformation are

$$\boldsymbol{q}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{q}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}.$$

The transformation matrix is

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_1\boldsymbol{A} \\ \boldsymbol{q}_1\boldsymbol{A}^2 \\ \boldsymbol{q}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 1 & 3 & 11 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The pair $(\boldsymbol{A}, \boldsymbol{B})$ in the new coordinates has the form

$$\tilde{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \left[\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 2 & 3 & 0 \\ \hline -21 & 0 & 0 & 5 \end{array}\right] \quad \text{and} \quad \tilde{\boldsymbol{B}} = \boldsymbol{T}\boldsymbol{B} = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hline 0 & 1 \end{array}\right].$$

The desired closed-loop characteristic polynomial is

$$\alpha_c(s) = (s+2)(s+3-j)(s+3+j)(s+4) = s^4 + 12s^3 + 54s^2 + 108s + 80.$$

A possible choice of the gain matrix $\tilde{\boldsymbol{K}}$, from among many, that does the job is $\tilde{\boldsymbol{K}}$ such that

$$\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -80 & -108 & -54 & -12 \end{bmatrix}.$$

Note that $\boldsymbol{\Gamma} = \boldsymbol{I}_2$. Therefore,

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} 1 & 2 & 3 & -1 \\ 59 & 108 & 54 & 17 \end{bmatrix},$$

and thus

$$\boldsymbol{K} = \tilde{\boldsymbol{K}}\boldsymbol{T} = \begin{bmatrix} 3 & 11 & 40 & -1 \\ 54 & 270 & 977 & 17 \end{bmatrix}.$$

The above algorithm for pole placement for multi-input systems is more of a theoretical value rather than practical. This algorithm is not well suited for numerical implementations because the transformation of a given pair $(\boldsymbol{A}, \boldsymbol{B})$ into the controller companion form suffers from poor numerical properties. For a discussion of pole placement numerically stable algorithms for multi-input systems, we refer the reader to Jamshidi, Tarokh, and Shafai [160].

Unlike in the single-input case, a solution to the pole placement problem for a multi-input system is not unique. Therefore, one can use the remaining degrees of freedom to achieve secondary goals. We discuss a method for constructing a linear state-feedback control law that allocates the closed-loop poles into prespecified locations, and at the same time minimizes a quadratic performance index, in Subsection 9.3.3.

Using the results preceding the above discussion, we state and prove a fundamental theorem of linear systems.

Theorem 5.8 *The pole placement problem is solvable for all choices of n desired closed-loop poles, symmetric with respect to the real axis, if and only if the given pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable.*

Proof The proof of necessity ($\Leftarrow$) follows immediately from the discussion preceding Example 5.11. Indeed, the pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable, if and only if it can be transformed into the controller canonical form. Once the transformation is performed, we can easily solve the pole placement problem for the given set of desired closed-loop poles, symmetric with respect to the real axis. We then transform the pair $(\boldsymbol{A}, \boldsymbol{B})$ and the gain matrix back into the original coordinates. Because the similarity transformation does not affect neither controllability nor the characteristic polynomial, the eigenvalues of $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}$ are precisely the desired prespecified closed-loop poles. This completes the proof of the necessity part.

We use a proof by contraposition to prove the sufficiency part ($\Rightarrow$). Assume that the pair $(\boldsymbol{A}, \boldsymbol{B})$ is nonreachable. By Theorem 5.4, there is a similarity transformation $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x}$ such that the pair $(\boldsymbol{A}, \boldsymbol{B})$ in the new coordinates has the form

$$\tilde{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{O} & \boldsymbol{A}_4 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \boldsymbol{T}\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{O} \end{bmatrix},$$

where the pair $(\boldsymbol{A}_1, \boldsymbol{B}_1)$ is reachable, $\boldsymbol{A}_1 \in \mathbb{R}^{r\times r}$, and $\boldsymbol{B}_1 \in \mathbb{R}^{r\times m}$. Let

$$\boldsymbol{u} = -\tilde{\boldsymbol{K}}\boldsymbol{z} = -\begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix}\boldsymbol{z},$$

where $\boldsymbol{K}_1 \in \mathbb{R}^{m\times r}$ and $\boldsymbol{K}_2 \in \mathbb{R}^{m\times(n-r)}$. Then,

$$\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{A}_1 - \boldsymbol{B}_1\boldsymbol{K}_1 & \boldsymbol{A}_2 - \boldsymbol{B}_1\boldsymbol{K}_2 \\ \boldsymbol{O} & \boldsymbol{A}_4 \end{bmatrix}. \tag{5.30}$$

It follows from (5.30) that the nonreachable portion of the system is not affected by the state-feedback, that is, the eigenvalues of $\boldsymbol{A}_4$ cannot be allocated. Hence, the pole placement problem cannot be solved in general if the pair $(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}})$, or equivalently the pair $(\boldsymbol{A}, \boldsymbol{B})$, is nonreachable, which completes the proof of the sufficiency part.

□

For an alternative proof of this important result, the reader may consult Wonham [355].

5.6 Observers

In the previous section, we presented algorithms for solving the pole placement problem using linear state-feedback control law. To implement such a control law, one needs availability of all the state variables. Often, this requirement is not met, either because measuring of all the state variables would require excessive number of sensors, or because the state variables are not accessible for direct measurement. Instead, only a subset of state variables or their combination may be available. To implement the control law constructed in the previous section, we will use estimates of state variables rather than the true states.

We now discuss the problem of constructing observers. Much of the literature refers to observers as "state estimators." For example, Franklin and Powell [106, page 139] argue that estimator is much more descriptive in its function because observer implies a direct measurement. On the other hand, estimator implies non-deterministic approach. Here we use only a deterministic approach. Therefore we follow the original terminology of the observer's inventor.

The first observer was proposed by Luenberger in the early nineteen sixties [221, 222, 223] for the purpose of estimating the state of a dynamical system, referred to as a plant, based on limited measurements of that system. More specifically, an observer is a deterministic dynamical system that can generate an estimate of the plant's state using that plant's input and output signals. A block diagram of a general observer structure is depicted in Figure 5.5. The plant's state estimate can then be employed, in place of the true state, to close the control loop. Observers can be used as "software" or "virtual" sensors as opposed to hardware sensing devices directly measuring physical variables, thus augmenting or replacing sensors in a control system [94]. Observers have been applied in a variety of different areas such

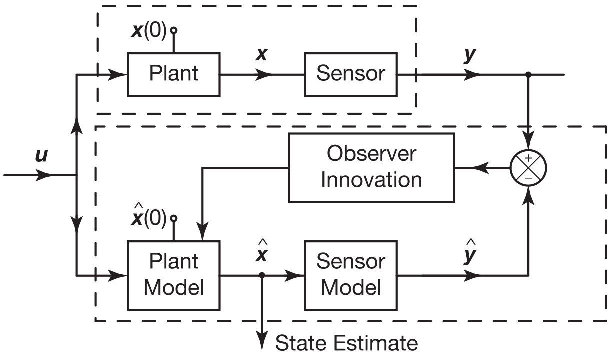


Figure 5.5: General observer architecture.

as secure communication using chaotic synchronization [120, 16, 214, 216, 163, 58], machine vision [59], wind energy systems [296], speed-sensorless control of induction motors [81, 80], or in model-based predictive control (MPC) [240], among many other applications.

We present two types of observers. First, full-order observers are discussed and then a reduced-order observers are presented.

5.6.1 Full-Order Observer

We discuss the problem of constructing a full-order observer for dynamical systems modeled by

$$\left.\begin{array}{rcl}\dot{\boldsymbol{x}}(t) & = & \boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)\\ \boldsymbol{y}(t) & = & \boldsymbol{C}\boldsymbol{x}(t),\end{array}\right\}$$

where $\boldsymbol{C} \in \mathbb{R}^{p\times n}$, $p \leq n$. We assume that the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable. Our goal is to construct a dynamical system that will estimate the state vector $\boldsymbol{x}$ based on the plant input $\boldsymbol{u}$ and the plant output $\boldsymbol{y}$. One could consider constructing a model of the plant dynamics and connecting the resulting dynamical system, referred to as an *open-loop observer*, as in Figure 5.6. An alternative representation of the open-loop observer is depicted in Figure 5.7. This open-loop observer is described by

$$\dot{\tilde{\boldsymbol{x}}}(t) = \boldsymbol{A}\tilde{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t),$$

where $\tilde{\boldsymbol{x}}(t)$ is the estimate of $\boldsymbol{x}(t)$. Let

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \tilde{\boldsymbol{x}}(t)$$

be the estimation error. Then, the dynamics of the observation error are described by

$$\dot{\boldsymbol{e}}(t) = \dot{\boldsymbol{x}}(t) - \dot{\tilde{\boldsymbol{x}}}(t) = \boldsymbol{A}\boldsymbol{e}(t),$$

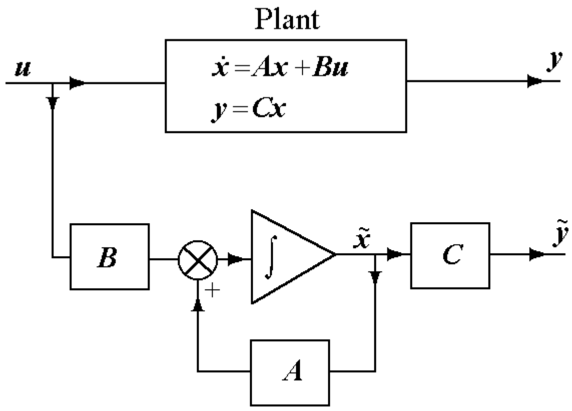


Figure 5.6: Open-loop observer.

with the initial estimation error

$$\boldsymbol{e}(0) = \boldsymbol{x}(0) - \tilde{\boldsymbol{x}}(0).$$

If the eigenvalues of the matrix $\boldsymbol{A}$ are in the open left-hand plane, then the error converges to zero. However, we have no control over the convergence rate. Furthermore, the matrix $\boldsymbol{A}$ does not have to have all its eigenvalues in the open left-hand plane. Thus, the open-loop observer is impractical. We modify this estimator by adding a feedback term to it. The resulting structure, depicted in Figure 5.8, is called the *closed-loop observer*, or the *Luenberger observer*, or the asymptotic full-order observer. An alternative representation of the closed-loop observer is depicted in Figure 5.9. The dynamics of the closed-loop observer are described by

$$\dot{\tilde{\boldsymbol{x}}}(t) = \boldsymbol{A}\tilde{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{L}\left(\boldsymbol{y}(t) - \boldsymbol{C}\tilde{\boldsymbol{x}}(t)\right),$$

and the dynamics of the observation error are governed by

$$\begin{aligned} \dot{\boldsymbol{e}}(t) &= \dot{\boldsymbol{x}}(t) - \dot{\tilde{\boldsymbol{x}}}(t) \\ &= (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\,\boldsymbol{e}(t), \quad \boldsymbol{e}(0) = \boldsymbol{x}(0) - \tilde{\boldsymbol{x}}(0). \end{aligned}$$

The pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, if and only if the dual pair $(\boldsymbol{A}^\top, \boldsymbol{C}^\top)$ is reachable. By assumption, the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, and therefore the pair $(\boldsymbol{A}^\top, \boldsymbol{C}^\top)$ is reachable. By Theorem 5.8, we can solve the pole placement problem for the dual pair $(\boldsymbol{A}^\top, \boldsymbol{C}^\top)$, that is,

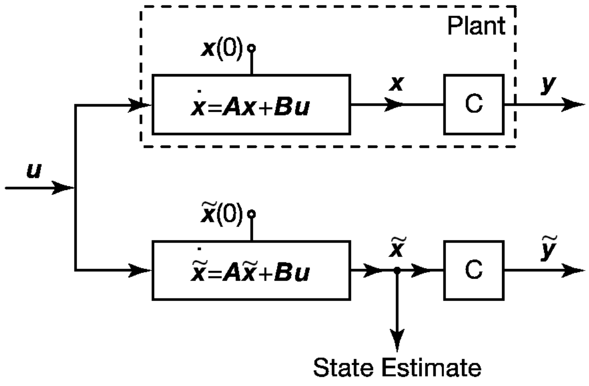


Figure 5.7: Alternative representation of the open-loop observer.

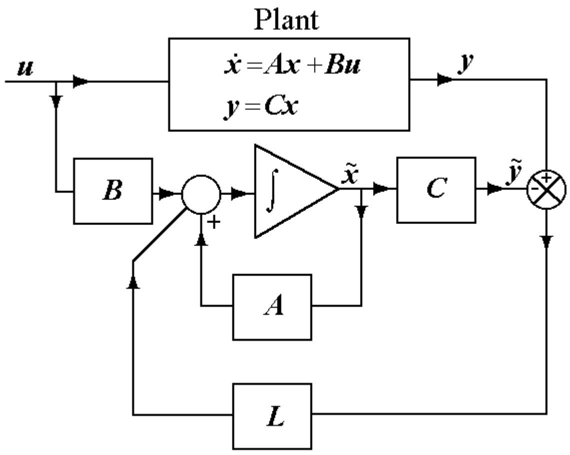


Figure 5.8: Closed-loop observer.

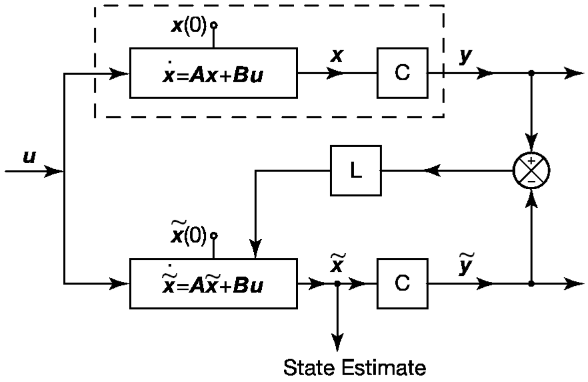


Figure 5.9: Alternative representation of the closed-loop observer.

for any set of prespecified n complex numbers, symmetric with respect to the real axis, there is a matrix, call it $\boldsymbol{L}^\top$, such that the eigenvalues of $\boldsymbol{A}^\top - \boldsymbol{C}^\top \boldsymbol{L}^\top$ and hence of $\boldsymbol{A} - \boldsymbol{LC}$ are in the prespecified locations. It follows from the above that if the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, then in addition to forcing the observation error to converge to zero, we can also control its rate of convergence by appropriately selecting the eigenvalues of the matrix $\boldsymbol{A} - \boldsymbol{LC}$. We also note that the selection of the closed-loop observer gain matrix $\boldsymbol{L}$ can be approached in exactly the same fashion as the construction of the gain matrix $\boldsymbol{K}$ in the linear state-feedback control law design. We illustrate the above point with a numerical example.

Example 5.12 For the given observable pair

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & -21 & 5 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

construct the matrix $\boldsymbol{L} \in \mathbb{R}^{4\times 2}$ so that the eigenvalues of $\boldsymbol{A} - \boldsymbol{LC}$ are located at

$$\{-2, -3 + j, -3 - j, -4\}.$$

In our construction of the gain matrix $\boldsymbol{L}$ we use the observer companion form of the pair $(\boldsymbol{A}, \boldsymbol{C})$ that we computed in Example 5.9. Our goal then is to construct a matrix $\boldsymbol{L}$ so that the characteristic polynomial of the matrix $\boldsymbol{A} - \boldsymbol{LC}$ is

$$\det(s\boldsymbol{I}_4 - \boldsymbol{A} + \boldsymbol{LC}) = s^4 + 12s^3 + 54s^2 + 108s + 80.$$

Using the observer form of the pair $(\boldsymbol{A}, \boldsymbol{C})$, we select $\hat{\boldsymbol{L}}$ so that

$$\hat{\boldsymbol{A}} - \hat{\boldsymbol{L}}\hat{\boldsymbol{C}} = \begin{bmatrix} 0 & 0 & 0 & -80 \\ 1 & 0 & 0 & -108 \\ 0 & 1 & 0 & -54 \\ 0 & 0 & 1 & -12 \end{bmatrix}.$$

We have

$$\hat{\boldsymbol{L}} = \begin{bmatrix} 1681 & 80 \\ 2270 & 108 \\ 1137 & 54 \\ 251 & 17 \end{bmatrix},$$

and hence

$$\boldsymbol{L} = \hat{\boldsymbol{T}}^{\top}\hat{\boldsymbol{L}} = \begin{bmatrix} 1137 & 54 \\ 3955 & 188 \\ 5681 & 270 \\ -23626 & -1117 \end{bmatrix}.$$

We note that there are other possible gain matrices $\boldsymbol{L}$ that could be used to allocate the eigenvalues of $\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}$ into desired locations.

We now present an alternative approach to the closed-loop observer design. This approach will then be used to construct a reduced-order observer that we discuss in the subsequent section. We assume that the dynamics of an observer are given by

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{D}\boldsymbol{z}(t) + \boldsymbol{E}\boldsymbol{u}(t) + \boldsymbol{G}\boldsymbol{y}(t), \tag{5.31}$$

where $\boldsymbol{z}(t) \in \mathbb{R}^n$. To begin with the development, suppose that we would like to obtain

$$\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x},$$

where $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ is invertible. Premultiplying $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$ by $\boldsymbol{T}$ yields

$$\boldsymbol{T}\dot{\boldsymbol{x}}(t) = \boldsymbol{T}\boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{T}\boldsymbol{B}\boldsymbol{u}(t). \tag{5.32}$$

Substituting into (5.31) the relation $\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$ gives

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{D}\boldsymbol{z}(t) + \boldsymbol{E}\boldsymbol{u}(t) + \boldsymbol{G}\boldsymbol{C}\boldsymbol{x}(t). \tag{5.33}$$

Subtracting (5.33) from (5.32) and taking into account that $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x}$ and $\dot{\boldsymbol{z}} = \boldsymbol{T}\dot{\boldsymbol{x}}$ yields

$$\mathbf{0} = (\boldsymbol{TA} - \boldsymbol{DT} - \boldsymbol{GC})\,\boldsymbol{x}(t) + (\boldsymbol{TB} - \boldsymbol{E})\,\boldsymbol{u}(t)$$

for all t and for arbitrary input $\boldsymbol{u}$. Hence, we must have

$$\begin{aligned} \boldsymbol{TA} - \boldsymbol{DT} &= \boldsymbol{GC} \\ \boldsymbol{E} &= \boldsymbol{TB}. \end{aligned}$$

If $\boldsymbol{z} \neq \boldsymbol{T}\boldsymbol{x}$, but the two above equations hold, then subtracting (5.32) from (5.33) and taking into account the above relations yields

$$\begin{aligned} \frac{d}{dt}(\boldsymbol{z} - \boldsymbol{T}\boldsymbol{x}) &= \boldsymbol{D}\boldsymbol{z} - \boldsymbol{TA}\boldsymbol{x} + \boldsymbol{GC}\boldsymbol{x} \\ &= \boldsymbol{D}\boldsymbol{z} - \boldsymbol{DT}\boldsymbol{x} \\ &= \boldsymbol{D}(\boldsymbol{z} - \boldsymbol{T}\boldsymbol{x}). \end{aligned}$$

If the eigenvalues of $\boldsymbol{D}$ all have negative real parts, then

$$\boldsymbol{z} - \boldsymbol{T}\boldsymbol{x} \to \mathbf{0} \quad \text{as} \quad t \to \infty.$$

Note that if we try $\boldsymbol{T} = \boldsymbol{I}_n$, then we have the previously analyzed case, where $\boldsymbol{G} = \boldsymbol{L}$, and

$$\begin{aligned} \boldsymbol{E} &= \boldsymbol{B}, \\ \boldsymbol{A} - \boldsymbol{LC} &= \boldsymbol{D}. \end{aligned}$$

The resulting observer, when $\boldsymbol{T} = \boldsymbol{I}_n$, was called the identity observer by its developer Luenberger [221, 222]. Given the desired eigenvalues, symmetric with respect to the real axis, the equation $\boldsymbol{A} - \boldsymbol{LC} = \boldsymbol{D}$ can be solved for $\boldsymbol{L}$ so that the resulting $\boldsymbol{D}$ has its eigenvalues in the desired locations, if and only if the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable. However, it is not true that given an arbitrary $\boldsymbol{D}$ we can solve for $\boldsymbol{L}$ so that $\boldsymbol{A} - \boldsymbol{LC} = \boldsymbol{D}$ even if the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable. We use the results of the above discussion to construct an observer of reduced order in the following section.

5.6.2 Reduced-Order Observer

The full-order observer possesses a certain degree of redundancy. It estimates all n state variables even though a linear combination of the states in the form of the output, $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$, is already known to us. If the output $\boldsymbol{y}$ is a p-vector with $p < n$, we would like to use the

p outputs to determine p of the x_i's and design an observer of order $n-p$ to estimate the rest of state variables. Specifically, if $\operatorname{rank} \boldsymbol{C} = p$, then we can use the equation $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$ to solve for p of the x_i's in terms of the y_i's and the remaining $n-p$ state variables x_k's. Then, we construct an observer of order $n-p$. We present two methods for constructing reduced-order observers of order $n-p$.

Method 1 for Reduced-Order Observer Construction

To recover all states, we should like to solve, for $\boldsymbol{x}$, the equations

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{T}\boldsymbol{x} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}, \end{aligned}$$

where now $\boldsymbol{T} \in \mathbb{R}^{(n-p)\times n}$. This can be done if and only if

$$\operatorname{rank} \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{C} \end{bmatrix} = n. \tag{5.34}$$

Given $\boldsymbol{D} \in \mathbb{R}^{(n-p)\times(n-p)}$, $\boldsymbol{G} \in \mathbb{R}^{(n-p)\times p}$, and the triple $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$, we can use the Kronecker product to show that the equation

$$\boldsymbol{T}\boldsymbol{A} - \boldsymbol{D}\boldsymbol{T} = \boldsymbol{G}\boldsymbol{C}$$

can be uniquely solved for $\boldsymbol{T}$ if and only if no eigenvalue of $\boldsymbol{D}$ is also an eigenvalue of $\boldsymbol{A}$. Indeed, the above equation is a special case of the matrix equation (A.54) that we analyzed in Section A.5. Note that when choosing the matrices $\boldsymbol{D}$ and $\boldsymbol{G}$, we must make sure that the condition (5.34) is satisfied. The structure of the resulting reduced-order observer is shown in Figure 5.10.

Method 2 for Reduced-Order Observer Construction

This method is presented in Luenberger [223, pp. 304–307]. In this method, we first find a full-rank matrix $\boldsymbol{N} \in \mathbb{R}^{n\times(n-p)}$ such that

$$\boldsymbol{C}\boldsymbol{N} = \boldsymbol{O}.$$

Then, we construct a state-variable transformation

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{N}^\top \\ \boldsymbol{C} \end{bmatrix} = \tilde{\boldsymbol{T}}\boldsymbol{x}.$$

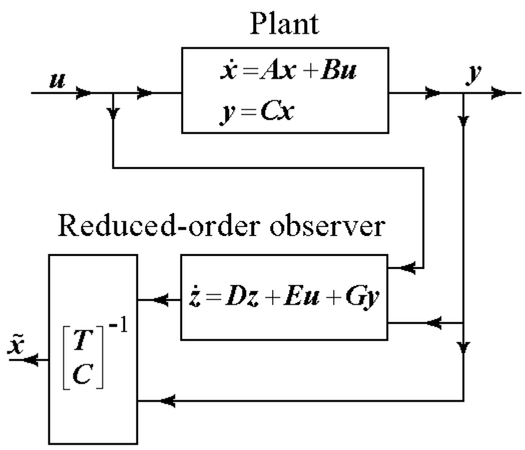


Figure 5.10: Architecture of the reduced-order observer designed using method 1.

Note that by construction, $\det \tilde{\boldsymbol{T}} \neq 0$, and therefore $\tilde{\boldsymbol{T}}$ is invertible. Let

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{N}^\top \boldsymbol{x} \\ \boldsymbol{C}\boldsymbol{x} \end{bmatrix}.$$

The plant model in the new coordinates has the form

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix} &= \tilde{\boldsymbol{T}}\boldsymbol{A}\tilde{\boldsymbol{T}}^{-1}\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix} + \tilde{\boldsymbol{T}}\boldsymbol{B}\boldsymbol{u} \\ &= \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix}\boldsymbol{u} \end{aligned} \tag{5.35}$$

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{C}\tilde{\boldsymbol{T}}^{-1} \\ &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_p \end{bmatrix}\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix}. \end{aligned} \tag{5.36}$$

We represent (5.35) as

$$\dot{\boldsymbol{w}} = \boldsymbol{A}_{11}\boldsymbol{w} + \boldsymbol{A}_{12}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u} \tag{5.37}$$

$$\dot{\boldsymbol{y}} = \boldsymbol{A}_{21}\boldsymbol{w} + \boldsymbol{A}_{22}\boldsymbol{y} + \boldsymbol{B}_2\boldsymbol{u}. \tag{5.38}$$

We can use the PBH test for observability to show that if the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, then so is the pair $(\boldsymbol{A}_{11}, \boldsymbol{A}_{21})$. Therefore, we can construct a matrix $\boldsymbol{L}_r \in \mathbb{R}^{(n-p)\times p}$ such that the matrix $(\boldsymbol{A}_{11} - \boldsymbol{L}_r\boldsymbol{A}_{21})$ has its eigenvalues in pre-specified locations in the open-left half complex plane. Our next step is to premultiply both sides of (5.38) by $\boldsymbol{L}_r$ and subtract the resulting equation from (5.37) to obtain

$$\dot{\boldsymbol{w}} - \boldsymbol{L}_r\dot{\boldsymbol{y}} = (\boldsymbol{A}_{11} - \boldsymbol{L}_r\boldsymbol{A}_{21})\boldsymbol{w} + (\boldsymbol{A}_{12} - \boldsymbol{L}_r\boldsymbol{A}_{22})\boldsymbol{y} + (\boldsymbol{B}_1 - \boldsymbol{L}_r\boldsymbol{B}_2)\boldsymbol{u}$$

We need a term $(\boldsymbol{w} - \boldsymbol{L}_r\boldsymbol{y})$ on the right hand-side of the above equation. We add and subtract the term $(\boldsymbol{A}_{11} - \boldsymbol{L}_r\boldsymbol{A}_{21})\boldsymbol{L}_r\boldsymbol{y}$ and perform simple manipulations to obtain

$$\frac{d}{dt}(\boldsymbol{w}-\boldsymbol{L}_r\boldsymbol{y}) = (\boldsymbol{A}_{11}-\boldsymbol{L}_r\boldsymbol{A}_{21})(\boldsymbol{w}-\boldsymbol{L}_r\boldsymbol{y})+(\boldsymbol{A}_{11}\boldsymbol{L}_r-\boldsymbol{L}_r\boldsymbol{A}_{21}\boldsymbol{L}_r+\boldsymbol{A}_{12}-\boldsymbol{L}_r\boldsymbol{A}_{22})\boldsymbol{y}+(\boldsymbol{B}_1-\boldsymbol{L}_r\boldsymbol{B}_2)\boldsymbol{u}. \tag{5.39}$$

Let

$$\boldsymbol{v} = \boldsymbol{w} - \boldsymbol{L}_r\boldsymbol{y}. \tag{5.40}$$

Then we represent (5.39) as

$$\dot{\boldsymbol{v}} = (\boldsymbol{A}_{11} - \boldsymbol{L}_r\boldsymbol{A}_{21})\boldsymbol{v} + (\boldsymbol{B}_1 - \boldsymbol{L}_r\boldsymbol{B}_2)\boldsymbol{u} + (\boldsymbol{A}_{11}\boldsymbol{L}_r - \boldsymbol{L}_r\boldsymbol{A}_{21}\boldsymbol{L}_r + \boldsymbol{A}_{12} - \boldsymbol{L}_r\boldsymbol{A}_{22})\boldsymbol{y}. \tag{5.41}$$

Let

$$\left.\begin{array}{rcl} \boldsymbol{D} & = & \boldsymbol{A}_{11} - \boldsymbol{L}_r\boldsymbol{A}_{21} \\ \boldsymbol{E} & = & \boldsymbol{B}_1 - \boldsymbol{L}_r\boldsymbol{B}_2 \\ \boldsymbol{G} & = & \boldsymbol{A}_{11}\boldsymbol{L}_r - \boldsymbol{L}_r\boldsymbol{A}_{21}\boldsymbol{L}_r + \boldsymbol{A}_{12} - \boldsymbol{L}_r\boldsymbol{A}_{22}. \end{array}\right\} \tag{5.42}$$

We construct a trivial observer of $\boldsymbol{v}$,

$$\dot{\boldsymbol{z}} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{E}\boldsymbol{u} + \boldsymbol{G}\boldsymbol{y}. \tag{5.43}$$

Note that

$$\frac{d}{dt}(\boldsymbol{v} - \boldsymbol{z}) = (\boldsymbol{A}_{11} - \boldsymbol{L}_r\boldsymbol{A}_{21})(\boldsymbol{v} - \boldsymbol{z}),$$

which means that we fully control the dynamics of the observation error $(\boldsymbol{v} - \boldsymbol{z})$ because we are free to select the gain matrix $\boldsymbol{L}_r$ to place the eigenvalues of $(\boldsymbol{A}_{11} - \boldsymbol{L}_r\boldsymbol{A}_{21})$ at any pre-specified locations. Thus

$$\boldsymbol{z}(t) \to \boldsymbol{v}(t) = \boldsymbol{w} - \boldsymbol{L}_r\boldsymbol{y}.$$

We use $\boldsymbol{z}$ and $\boldsymbol{y}$ to obtain the estimate of $\boldsymbol{x}$, denoted $\tilde{\boldsymbol{x}}$, that is,

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{T}}^{-1}\begin{bmatrix} \hat{\boldsymbol{w}} \\ \boldsymbol{y} \end{bmatrix} = \tilde{\boldsymbol{T}}^{-1}\begin{bmatrix} \boldsymbol{z} + \boldsymbol{L}_r\boldsymbol{y} \\ \boldsymbol{y} \end{bmatrix}. \tag{5.44}$$

The structure of the above observer has the format somewhat similar to the one depicted in Figure 5.10, which is the structure of the reduced-order observer designed using Method 1. However, the observers' matrices are computed quite differently in both methods.

The last step in the control design process involving an observer is to combine the control law, constructed in the previous section, with the observer.

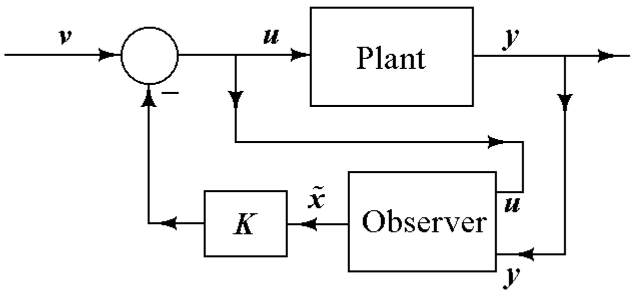


Figure 5.11: Closed-loop system driven by the combined controller-observer compensator.

5.7 Combined Controller-Observer Compensator

We start by combining a control law with a full-order observer. We write the equations that govern the behavior of the dynamical system consisting of the plant model and the full-order observer. The resulting system's order is $2n$, and the equations that describe the system are

$$\left.\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{\tilde{\boldsymbol{x}}}(t) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{LC} & \boldsymbol{A}-\boldsymbol{LC} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \tilde{\boldsymbol{x}}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \tilde{\boldsymbol{x}}(t) \end{bmatrix} \end{aligned}\right\} \tag{5.45}$$

Suppose that we now apply the control law

$$\boldsymbol{u}(t) = -\boldsymbol{K}\tilde{\boldsymbol{x}}(t) + \boldsymbol{v}(t) \tag{5.46}$$

instead of the actual state-feedback control law. In the above, the vector $\boldsymbol{v}$ denotes an external input signal. A schematic of a closed-loop system driven by the combined controller-observer compensator is shown in Figure 5.11. A model of the closed-loop system is obtained by combining equations (5.45) and (5.46),

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{\tilde{\boldsymbol{x}}}(t) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{BK} \\ \boldsymbol{LC} & \boldsymbol{A}-\boldsymbol{LC}-\boldsymbol{BK} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \tilde{\boldsymbol{x}}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{y}(t) &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \tilde{\boldsymbol{x}}(t) \end{bmatrix}. \end{aligned}$$

To analyze the above closed-loop system, it is convenient to perform a change of state variables using the following transformation

$$\begin{bmatrix} \boldsymbol{x}-\tilde{\boldsymbol{x}} \\ \tilde{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \boldsymbol{O} & \boldsymbol{I}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \tilde{\boldsymbol{x}} \end{bmatrix}.$$

Note that

$$\begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \boldsymbol{O} & \boldsymbol{I}_n \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{I}_n \\ \boldsymbol{O} & \boldsymbol{I}_n \end{bmatrix}.$$

In the new coordinates the equations describing the closed-loop system are

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{x}}(t)-\dot{\tilde{\boldsymbol{x}}}(t) \\ \dot{\tilde{\boldsymbol{x}}}(t) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}-\boldsymbol{L}\boldsymbol{C} & \boldsymbol{O} \\ \boldsymbol{L}\boldsymbol{C} & \boldsymbol{A}-\boldsymbol{B}\boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t)-\tilde{\boldsymbol{x}}(t) \\ \tilde{\boldsymbol{x}}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{y}(t) &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t)-\tilde{\boldsymbol{x}}(t) \\ \tilde{\boldsymbol{x}}(t) \end{bmatrix}. \end{aligned}$$

Note that in the above system, the subsystem corresponding to the error component $\boldsymbol{e}(t) = \boldsymbol{x}(t) - \tilde{\boldsymbol{x}}(t)$ is uncontrollable. Furthermore, the $2n$ poles of the closed-loop system are equal to the individual eigenvalues of both $\boldsymbol{A}-\boldsymbol{L}\boldsymbol{C}$ and $\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}$, which means that the design of the observer is separated from the construction of the control law. This is what is known as the *separation principle*. The closed-loop transfer function relating $\boldsymbol{Y}(s)$ and $\boldsymbol{V}(s)$ is

$$\begin{aligned} \boldsymbol{Y}(s) &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} + \boldsymbol{L}\boldsymbol{C} & \boldsymbol{O} \\ -\boldsymbol{L}\boldsymbol{C} & s\boldsymbol{I}_n - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{V}(s) \\ &= \boldsymbol{C}\left(s\boldsymbol{I}_n - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right)^{-1} \boldsymbol{B}\boldsymbol{V}(s). \end{aligned}$$

The above expression is identical to that for the closed-loop system if we applied the state-feedback control law $\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t) + \boldsymbol{v}(t)$. Thus, the combined controller-observer compensator yields the same closed-loop transfer function as the actual state-feedback control law.

We now briefly discuss the issue of the observer pole selection. Franklin, Powell, and Emami-Naeini [107, page 552] recommend that the real parts of the observer poles, that is, the real parts of the eigenvalues of the matrix $\boldsymbol{A}-\boldsymbol{L}\boldsymbol{C}$, be a factor of 2 to 6 times deeper in the open left-half plane than the real parts of the controller poles which are the eigenvalues of the matrix $\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}$. Such a choice ensures a faster decay of the observer error $\boldsymbol{e}(t) = \tilde{\boldsymbol{x}}(t) - \boldsymbol{x}(t)$ compared with the desired controller dynamics. This in turn causes the controller poles to dominate the closed-loop system response. Because the observer poles represent a measure of the speed with which the error $\boldsymbol{e}(t) = \tilde{\boldsymbol{x}}(t) - \boldsymbol{x}(t)$ decays to zero, one would tend to assign observer poles deep in the left-hand plane. However, fast decay requires large gains which may lead to saturation of some signals and unpredictable nonlinear effects. If the observer poles were slower than the controller poles, the closed-loop system response would be dominated by the observer, which is undesirable. As it is usual in engineering practice, the term compromise could be used to describe the process of constructing the final compensator structure.

We next analyze the combined controller-observer compensator that uses the reduced-order observer. The closed-loop system driven by this type of compensator also enjoys the separation property as in the case of the full-order observer. To prove the above statement, we begin with writing the equations describing the closed-loop system,

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}}(t) & = & \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \dot{\boldsymbol{z}}(t) & = & \boldsymbol{D}\boldsymbol{z}(t) + \boldsymbol{T}\boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{G}\boldsymbol{C}\boldsymbol{x}(t) \\ \boldsymbol{y}(t) & = & \boldsymbol{C}\boldsymbol{x}(t) \\ \boldsymbol{u}(t) & = & -\boldsymbol{K}\tilde{\boldsymbol{x}}(t) + \boldsymbol{v}(t), \end{array}\right\}$$

where

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{C} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{y} \end{bmatrix}.$$

Let

$$\begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{C} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{N} \end{bmatrix},$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times(n-p)}$ and $\boldsymbol{N} \in \mathbb{R}^{n\times p}$. Note that

$$\boldsymbol{M}\boldsymbol{T} + \boldsymbol{N}\boldsymbol{C} = \boldsymbol{I}_n.$$

Hence, the control input can be represented as

$$\begin{array}{rcl} \boldsymbol{u}(t) & = & -\boldsymbol{K}\begin{bmatrix} \boldsymbol{M} & \boldsymbol{N} \end{bmatrix}\begin{bmatrix} \boldsymbol{z}(t) \\ \boldsymbol{y}(t) \end{bmatrix} + \boldsymbol{v}(t) \\ & = & -\boldsymbol{K}\boldsymbol{M}\boldsymbol{z}(t) - \boldsymbol{K}\boldsymbol{N}\boldsymbol{y}(t) + \boldsymbol{v}(t). \end{array}$$

Using the above equations, and noting that $\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$, we represent equations that describe the closed-loop system as

$$\begin{aligned}
\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{\boldsymbol{z}}(t) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{GC} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{z}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{TB} \end{bmatrix} (-\boldsymbol{KMz}(t) - \boldsymbol{KNy}(t) + \boldsymbol{v}(t)) \\
&= \begin{bmatrix} \boldsymbol{A} - \boldsymbol{BKNC} & -\boldsymbol{BKM} \\ \boldsymbol{GC} - \boldsymbol{TBKNC} & \boldsymbol{D} - \boldsymbol{TBKM} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{z}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{TB} \end{bmatrix} \boldsymbol{v}(t) \\
\boldsymbol{y}(t) &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{z}(t) \end{bmatrix}.
\end{aligned}$$

To proceed further, we perform a transformation of the state variable of the above system,

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} - \boldsymbol{Tx} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{O} \\ -\boldsymbol{T} & \boldsymbol{I}_{n-p} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}.$$

Note that

$$\begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{O} \\ -\boldsymbol{T} & \boldsymbol{I}_{n-p} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{O} \\ \boldsymbol{T} & \boldsymbol{I}_{n-p} \end{bmatrix}.$$

In the new coordinates the equations describing the closed-loop system become

$$\begin{aligned}
\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{\boldsymbol{z}}(t) - \boldsymbol{T}\dot{\boldsymbol{x}}(t) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} - \boldsymbol{BKNC} - \boldsymbol{BKMT} & -\boldsymbol{BKM} \\ -\boldsymbol{TA} + \boldsymbol{GC} + \boldsymbol{DT} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{z}(t) - \boldsymbol{Tx}(t) \end{bmatrix} \\
&\quad + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{v}(t) \\
\boldsymbol{y}(t) &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{z}(t) - \boldsymbol{Tx}(t) \end{bmatrix}.
\end{aligned}$$

Taking into account the equations

$$\boldsymbol{TA} - \boldsymbol{DT} - \boldsymbol{GC} = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{MT} = \boldsymbol{I}_n - \boldsymbol{NC},$$

we obtain

$$\begin{aligned}
\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{\boldsymbol{z}}(t) - \boldsymbol{T}\dot{\boldsymbol{x}}(t) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} - \boldsymbol{BK} & -\boldsymbol{BKM} \\ \boldsymbol{O} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{z}(t) - \boldsymbol{Tx}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{v}(t) \\
\boldsymbol{y}(t) &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{z}(t) - \boldsymbol{Tx}(t) \end{bmatrix}.
\end{aligned}$$

It is now apparent that the separation principle also holds if we use the reduced-order observer in constructing the compensator, that is, the selection of the gain matrix $\boldsymbol{K}$ that assigns

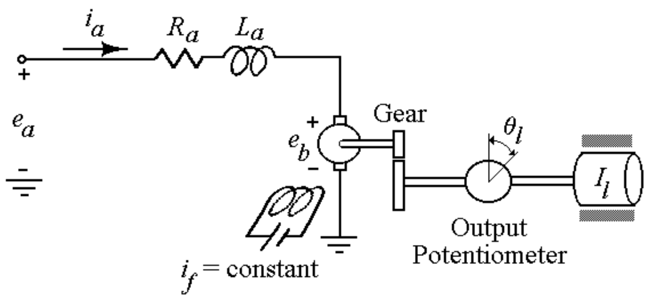


Figure 5.12: Schematic of an armature-controlled DC motor system of Example 5.13.

closed-loop poles is independent, separate, from the selection of the observer gain matrix $\boldsymbol{L}$ that assigns the observer poles. Furthermore, the transfer function matrix relating $\boldsymbol{Y}(s)$ and $\boldsymbol{V}(s)$ in the closed-loop system with the reduced-order observer is exactly the same as the one that would be obtained using the actual state-feedback control law $\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t) + \boldsymbol{v}(t)$.

Example 5.13 Consider a schematic of an armature-controlled DC motor system, shown in Figure 5.12, where the system parameters are: $R_a = 5\ \Omega$, $L_a = 200$ mH, $K_b = 0.1$ V/rad/sec, $K_i = 0.1$ Nm/A, the gear ratio $N_1/N_2 = 1/50$. The armature inertia is $I_{armature} = 2 \times 10^{-3}$ kg·m^2. The load of 10 kg is located at an effective radius of 0.2 m. The gear inertia and friction are negligible.

We first construct a state-space model of the DC motor system. We start with writing the equation relating torque to angular acceleration:

$$T_m = I_{eq}\ddot{\theta}, \tag{5.47}$$

where

$$\begin{aligned}
I_{eq} &= I_{armature} + (N_1/N_2)^2\, I_l \\
&= 2 \times 10^{-3}\,\text{kg·m}^2 + (1/50)^2 \times 10 \times (0.2)^2\,\text{kg·m}^2 \\
&= 2.16 \times 10^{-3}\,\text{kg·m}^2.
\end{aligned}$$

Kirchhoff's voltage law applied to the armature circuit gives

$$R_a i_a + L_a \frac{di_a}{dt} + e_b = e_a,$$

where $e_b = K_b \frac{d\theta}{dt}$. The equation for the developed torque is

$$T_m = K_i i_a. \tag{5.48}$$

Combining (5.47) and (5.48) gives

$$\ddot{\theta} = \frac{K_i}{I_{eq}} i_a.$$

Let $x_1 = i_a$, $x_2 = \theta$, $x_3 = \dot{\theta} = \omega$, $u = e_a$, and $y = \theta_l$. Taking into account the definition of the state variables, we represent the above modeling equations in state-space format:

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} -\frac{R_a}{L_a} & 0 & -\frac{K_b}{L_a} \\ 0 & 0 & 1 \\ \frac{K_i}{I_{eq}} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} u \\
y &= \begin{bmatrix} 0 & \frac{N_1}{N_2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.
\end{aligned}$$

Substituting the given parameter values, we get

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} -25 & 0 & -0.5 \\ 0 & 0 & 1 \\ 46.296 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix} u = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u \\
y &= \begin{bmatrix} 0 & 0.02 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \boldsymbol{c}\boldsymbol{x}.
\end{aligned}$$

Our next step is to design a state-feedback controller $u = -\boldsymbol{k}\boldsymbol{x} + v$, such that the eigenvalues of $\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k}$ are

$$-1 + j, -1 - j, -10.$$

To allocate the poles into the desired locations, we first transform the system model into the controller companion form. We first form the controllability matrix

$$\boldsymbol{\mathcal{U}} = \begin{bmatrix} \boldsymbol{b} & \boldsymbol{A}\boldsymbol{b} & \boldsymbol{A}^2\boldsymbol{b} \end{bmatrix} = \begin{bmatrix} 5 & -125 & 3009.3 \\ 0 & 0 & 231.5 \\ 0 & 231.5 & -5787 \end{bmatrix}.$$

The system is controllable. The last row of the inverse of the controllability matrix is

$$\boldsymbol{q}_1 = \begin{bmatrix} 0 & 0.0043 & 0 \end{bmatrix},$$

and hence the transformation matrix we are seeking has the form

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_1\boldsymbol{A} \\ \boldsymbol{q}_1\boldsymbol{A}^2 \end{bmatrix} = \begin{bmatrix} 0 & 0.0043 & 0 \\ 0 & 0 & 0.0043 \\ 0.2 & 0 & 0 \end{bmatrix}.$$

The system's matrices in the new coordinates are

$$\tilde{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -23.148 & -25 \end{bmatrix},$$

$$\tilde{\boldsymbol{b}} = \boldsymbol{T}\boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

and

$$\tilde{\boldsymbol{c}} = \boldsymbol{c}\boldsymbol{T}^{-1} = \begin{bmatrix} 4.63 & 0 & 0 \end{bmatrix}.$$

The desired closed-loop characteristic polynomial $\alpha_c(s)$ is

$$\alpha_s(s) = (s+1-j)(s+1+j)(s+10) = s^3 + 12s^2 + 22s + 20.$$

We now find $\tilde{\boldsymbol{k}}$ so that

$$\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{b}}\tilde{\boldsymbol{k}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -20 & -22 & -12 \end{bmatrix}.$$

We have

$$\tilde{\boldsymbol{k}} = \begin{bmatrix} 20 & -1.148 & -13 \end{bmatrix},$$

and

$$\boldsymbol{k} = \tilde{\boldsymbol{k}}\boldsymbol{T} = \begin{bmatrix} -2.6 & 0.0864 & -0.005 \end{bmatrix}.$$

Hence,

$$\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k} = \begin{bmatrix} -12 & -0.432 & -0.4752 \\ 0 & 0 & 1 \\ 46.294 & 0 & 0 \end{bmatrix}.$$

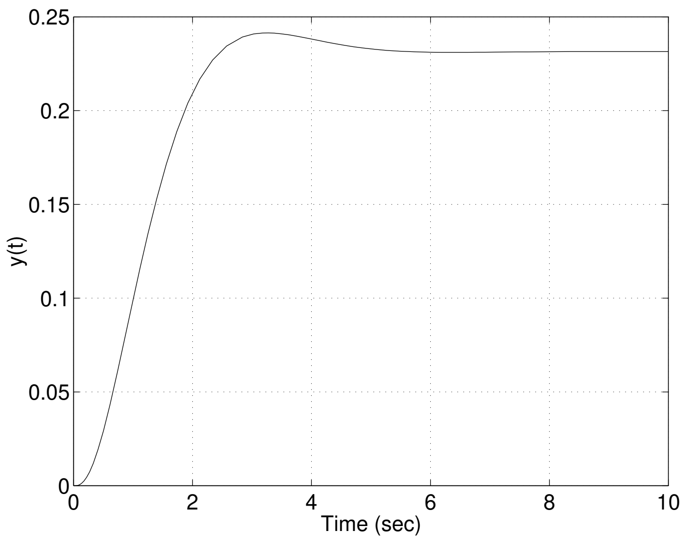


Figure 5.13: Step response of the closed-loop DC motor system with state feedback.

The transfer function of the closed-loop system is

$$\frac{\mathcal{L}(y(t))}{\mathcal{L}(v(t))} = \frac{Y(s)}{V(s)} = \boldsymbol{c}\left(s\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{b}\boldsymbol{k}\right)^{-1}\boldsymbol{b} = \frac{4.6296}{s^3 + 12s^2 + 22s + 20}.$$

The unit-step response of the closed-loop system is shown in Figure 5.13.

Next, we design a full-order observer placing its poles at $\{-4, -5+2j, -5-2j\}$, and then synthesize the combined controller-observer compensator. To compute the observer gain vector $\boldsymbol{l}$, we first transform the pair $(\boldsymbol{A}, \boldsymbol{c})$ into the observer companion form. This is equivalent to transforming the pair $(\boldsymbol{A}^\top, \boldsymbol{c}^\top)$ into the controller companion form. The transformation matrix that brings the pair $(\boldsymbol{A}, \boldsymbol{c})$ into its observer companion form is

$$\hat{\boldsymbol{Q}} = \hat{\boldsymbol{T}}^\top = \begin{bmatrix} 1.1 & -27 & 650 \\ 0 & 0 & 50 \\ 0 & 50 & -1250 \end{bmatrix}.$$

The matrices $\boldsymbol{A}$ and $\boldsymbol{c}$ in the new coordinates have the form:

$$\hat{\boldsymbol{A}} = \hat{\boldsymbol{Q}}^{-1}\boldsymbol{A}\hat{\boldsymbol{Q}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -23.1481 \\ 0 & 1 & -25 \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{c}} = \boldsymbol{c}\hat{\boldsymbol{Q}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

The desired characteristic polynomial of the observer is

$$\det\left(\boldsymbol{I}_3 - \boldsymbol{A} + \boldsymbol{l}\boldsymbol{c}\right) = (s+4)(s+5-2j)(s+5+2j) = s^3 + 14s^2 + 69s + 116.$$

We now find $\hat{\boldsymbol{l}}$ so that

$$\hat{\boldsymbol{A}} - \hat{\boldsymbol{l}}\hat{\boldsymbol{c}} = \begin{bmatrix} 0 & 0 & -116 \\ 1 & 0 & -69 \\ 0 & 1 & -14 \end{bmatrix}.$$

We have

$$\hat{\boldsymbol{l}} = \begin{bmatrix} 116 \\ 45.8519 \\ -11 \end{bmatrix} \quad \text{and therefore} \quad \boldsymbol{l} = \hat{\boldsymbol{Q}}\hat{\boldsymbol{l}} = \begin{bmatrix} -8263 \\ -550 \\ 16043 \end{bmatrix}.$$

The dynamics of the full-order observer are described by

$$\dot{\tilde{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c})\tilde{\boldsymbol{x}} + \boldsymbol{b}u + \boldsymbol{l}y,$$

that is,

$$\dot{\tilde{\boldsymbol{x}}} = \begin{bmatrix} -25 & 165.2544 & -0.5 \\ 0 & 11 & 1 \\ 46.2963 & -320.8519 & 0 \end{bmatrix} \tilde{\boldsymbol{x}} + \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} -8263 \\ -550 \\ 16043 \end{bmatrix} y.$$

We connect the full-order observer to the DC motor system, thus obtaining a closed-loop system with the combined controller-full-order-observer compensator in the loop as in Figure 5.11. In Figure 5.14, we show a plot of x_1 and its estimate for the closed-loop system with the observer in the loop, where $v = 0$, $\boldsymbol{x}(0) = \begin{bmatrix} 1 & 0.2 & -0.1 \end{bmatrix}^\top$, and $\tilde{\boldsymbol{x}}(0) = \boldsymbol{0}$.

We point out that the step response of the closed-loop system with the combined controller-full-order-observer compensator in the loop as in Figure 5.11, with zero initial conditions on the plant and the full-order observer, is identical to that shown in Figure 5.13.

Our next part of this example is to design the reduced-order observer using the design method 1. We choose the reduced-order observer's poles at $\{-5+2j, -5-2j\}$. The reduced-order observer dynamics are given by

$$\dot{\boldsymbol{z}} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{E}u + \boldsymbol{G}y.$$

Let

$$\boldsymbol{G} = \begin{bmatrix} 100 \\ 100 \end{bmatrix}.$$

The observer's poles are supposed to be $-5+2j$ and $-5-2j$. Thus, the desired characteristic polynomial of $\boldsymbol{D}$ should be

$$\det(s\boldsymbol{I}_2 - \boldsymbol{D}) = s^2 + 10s + 29.$$

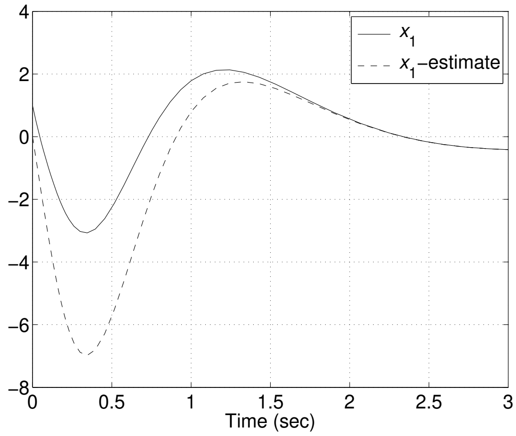


Figure 5.14: A plot of x_1 and its estimate versus time for the closed-loop system with the full-order observer in the loop.

Let

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 \\ -29 & -10 \end{bmatrix}.$$

We now solve the matrix equation,

$$-\boldsymbol{D}\boldsymbol{T} + \boldsymbol{T}\boldsymbol{A} = \boldsymbol{G}\boldsymbol{c},$$

for $\boldsymbol{T}$. We first convert the above equation in the format similar to that given by (A.55) in Section A.5 and then solve the resulting equation. Employing the following MATLAB's commands,

```
Gc=G*c;
t=(-kron(eye(3),D)+kron(A',eye(2)))\Gc(:);
T=[t(1:2) t(3:4) t(5:6)]
```

we obtain

$$\boldsymbol{T} = \begin{bmatrix} -0.5493 & 0.7586 & -0.2507 \\ 2.1271 & -2.0000 & 1.0332 \end{bmatrix}.$$

Note that

$$\det \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{c} \end{bmatrix} = 6.8691 \times 10^{-4} \neq 0,$$

and

$$\begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{c} \end{bmatrix}^{-1} = \begin{bmatrix} -30.0840 & -7.2980 & 411.3200 \\ 0 & 0 & 50 \\ 61.9338 & 15.9921 & -750 \end{bmatrix}.$$

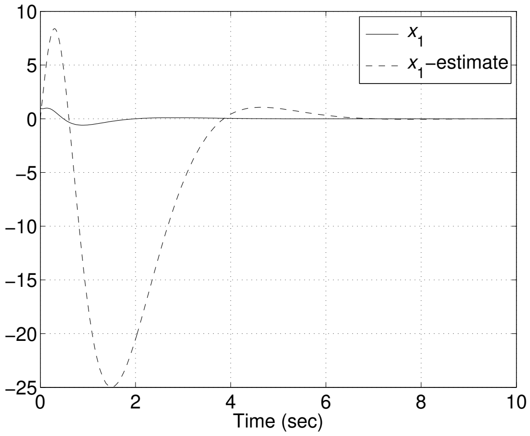


Figure 5.15: A plot of x_1 and its estimate versus time for the closed-loop system with the reduced-order observer, designed using method 1, in the loop.

We also have

$$\boldsymbol{E} = \boldsymbol{T}\boldsymbol{b} = \begin{bmatrix} -2.7463 \\ 10.6357 \end{bmatrix}.$$

Hence, the dynamics of the reduced-order observer are described by

$$\dot{\boldsymbol{z}} = \begin{bmatrix} 0 & 1 \\ -29 & -10 \end{bmatrix} \boldsymbol{z} + \begin{bmatrix} -2.7463 \\ 10.6357 \end{bmatrix} u + \begin{bmatrix} 100 \\ 100 \end{bmatrix} y.$$

An estimate of the entire state can be obtained from

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{c} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{z} \\ y \end{bmatrix} = \begin{bmatrix} -30.0840 & -7.2980 & 411.3200 \\ 0 & 0 & 50 \\ 61.9338 & 15.9921 & -750 \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ y \end{bmatrix}.$$

A plot of x_1 and its estimate versus time for the closed-loop system with the combined controller-reduced-order-observer compensator is depicted in Figure 5.15, where $v = 0$, $\boldsymbol{x}(0) = \begin{bmatrix} 1 & 0.2 & -0.1 \end{bmatrix}^\top$, and $\boldsymbol{z}(0) = \mathbf{0}$.

Our last part of this example is to construct the reduced-order observer employing the design method 2. This reduced-order observer matrices are computed using (5.42). The state estimate is obtained from (5.44). A plot of x_1 and its estimate versus time for the closed-loop system with the combined controller-reduced-order-observer compensator is depicted in Figure 5.16, where, as before, $v = 0$, $\boldsymbol{x}(0) = \begin{bmatrix} 1 & 0.2 & -0.1 \end{bmatrix}^\top$, and $\boldsymbol{z}(0) = \mathbf{0}$.

Of course, for zero initial conditions, the step responses of the closed-loop systems with the full-order, as well as with the reduced-order observers in the loop, are the same as the

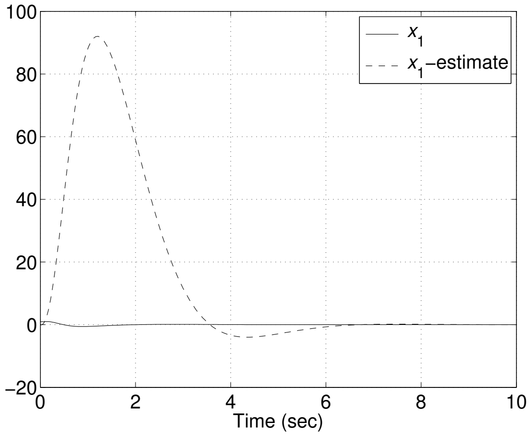


Figure 5.16: A plot of x_1 and its estimate versus time for the closed-loop system with the reduced-order observer, designed using method 2, in the loop.

step response of the closed-loop system with the state-feedback controller $u = -\boldsymbol{kx}+v$. This follows from our analyses of the combined controller-observer compensators.

5.8 Notes

For further reading on the subject of linear system theory, see Brockett [43], Wolovich [352], Barnett [23], Kailath [168], Skelton [300], DeCarlo [75], Rugh [288], Antsaklis and Michel [12], or Chen [55]. For more information on canonical forms, see the papers by Popov [277] and Denham [79].

One may argue that Maxwell's [231] analysis of Watt's governor constitutes the first work in linear system theory: the physical object, its linear model and a method of analysis. For example, MacFarlane [226, page 251] states that "Maxwell's fundamentally important contribution lay in recognizing that the behavior of an automatic feedback control system in the vicinity of an equilibrium condition could be approximated by a linear differential equation, and hence the stability of the control system could be discussed in terms of the location of the roots of an associated algebraic equation." On the other hand, L. Markus in [97, page xxvii] makes the following statement: "But certainly Watt's application of the centrifugal governor for regulating steam engines (1775) can be taken as the start of modern control engineering."

In Kalman's [169] well-known 1963 paper, there is a theorem which states that the state space, $\mathcal{X}$, of a linear system model may be decomposed as a direct sum

$$\mathcal{X} = \mathcal{X}_A \oplus \mathcal{X}_B \oplus \mathcal{X}_C \oplus \mathcal{X}_D,$$

where

$$\begin{aligned}
\mathcal{X}_A &= \text{states that are reachable but not observable}\\
\mathcal{X}_B &= \text{states that are reachable and observable}\\
\mathcal{X}_C &= \text{states that are neither reachable nor observable}\\
\mathcal{X}_D &= \text{states that are not reachable but observable}
\end{aligned}$$

This decomposition became, to a certain extent, an icon of linear systems. For example, the logo on the covers of the old issues of the *International Journal of Systems Science* was the above decomposition. Many authors included Kalman's original decomposition algorithm in their textbooks on linear systems. Yet almost twenty years after Kalman's publication of the decomposition theorem, Boley [37] generated an example which revealed some shortcomings of Kalman's original algorithm. In his reply, Kalman [170] revised his decomposition algorithm and fixed his original proof of the decomposition theorem.

5.9 Exercises

Exercise 5.1 (adapted from [23, page 94]) Consider the system, shown in Figure 5.17, which is composed of two masses moving on a smooth, that is, frictionless, horizontal plane. The springs are linear, that is, they obey Hooke's law,

$$\text{force} = k \times \text{extension},$$

where k is a constant called the force constant of the spring. In addition to the springs, the masses are joined by linear dampers. In a linear damper, the force is proportional to the velocity of one end of the damper relative to the other end. The proportionality constant is called the damping coefficient. In this exercise, damping coefficients are b_1 and b_2, respectively. A force u is applied to the right end mass. Obtain the system's state space model in the form $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u$. Determine under what condition, in terms of k_1, the system model is controllable for $m_1 = m_2 = 1$ kg, $b_1 = b_2 = 1$ Ns/m, and $k_2 = 1/4$ N/m.

Exercise 5.2 In the circuit shown in Figure 5.18, take the state variables to be the voltage across the capacitor and the current through the inductor. The source voltage v_s is the

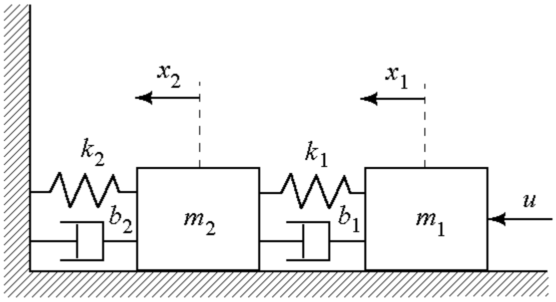


Figure 5.17: A mechanical system for Exercise 5.1.

control variable and the current i is the system output. Obtain the state-space model of the this circuit. Obtain the condition, in terms of R_1, R_2, C, and L, under which the model is uncontrollable and unobservable. Find the transfer function of the circuit model.

Exercise 5.3 In the circuit shown in Figure 5.19, take the state variables to be the voltage across the capacitor C_1, the current through the inductor L_1, and the current through the inductor L_2. The source current i_s is the control variable and the current through the inductor L_2 is the system output. Obtain the state space model of this circuit. Analyze the circuit model controllability and observability properties. Find the transfer function of the circuit model.

Exercise 5.4 For the system

$$\dot{x} = x + u,$$

find the set of points that can be steered to the origin $x = 0$ by an admissible control $|u| \leq 1$.

Exercise 5.5 (a) Find $e^{\boldsymbol{A}t}$ for

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -ab & -a-b \end{bmatrix}.$$

(b) Given a dynamical system modeled by the differential equation

$$\ddot{x} + 2\dot{x} + x = \dot{r} + r,$$

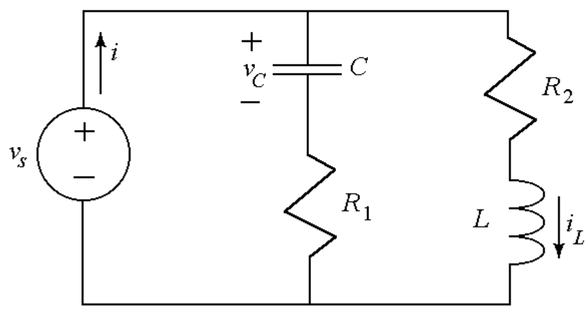


Figure 5.18: Electric circuit for Exercise 5.2.

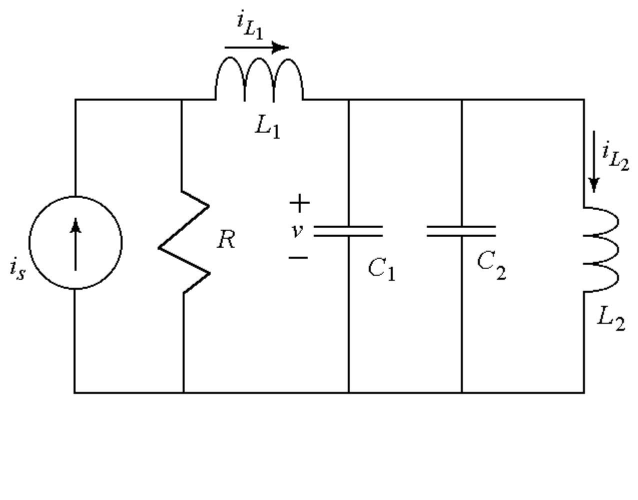


Figure 5.19: Electric circuit for Exercise 5.3.

where $r = r(t) = \sin t$, and the initial conditions are $x(0) = 1$ and $\dot{x}(0) = 0$. Choose the state variables $x_1 = x$ and $x_2 = \dot{x}_1$. Represent the differential equation in state-space format. Solve for $x(t)$ for $t \geq 0$.

In both cases, verify your calculations using MATLAB's Symbolic Toolbox.

Exercise 5.6 Find sufficient conditions, in terms of a, b, c, and d, for reachability of the following pairs:

(a)

$$\boldsymbol{A} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ a & b \\ 0 & 0 \\ c & d \end{bmatrix};$$

(b)

$$\boldsymbol{A} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ a & b \\ 0 & 0 \\ c & d \end{bmatrix}.$$

Exercise 5.7 A system composed of two subsystems connected in cascade is depicted in the block diagram shown in Figure 5.20. Take the state variables of the system to be: $x_1 = y_1 - u_1$ and $x_2 = y_2$. Assume that the system input is $u = u_1$ and the system output is $y = y_2$. Obtain the state space model of the interconnected system. When is the system uncontrollable? Find the transfer function of the uncontrollable system.

Exercise 5.8 Derive a state-space description of a system represented by the block diagram given in Figure 5.21. Construct a state-variable transformation $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x}$ such that the matrix $\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1}$ has a diagonal form. Find the eigenvalue of $\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1}$ corresponding to the noncontrollable state and the eigenvalue corresponding to the nonobservable state. Find the transfer function of the system.

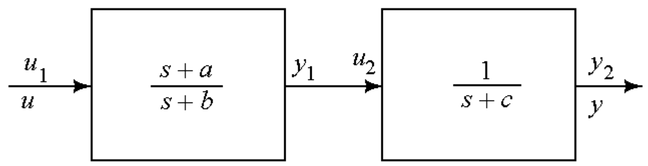


Figure 5.20: A block diagram for Exercise 5.7.

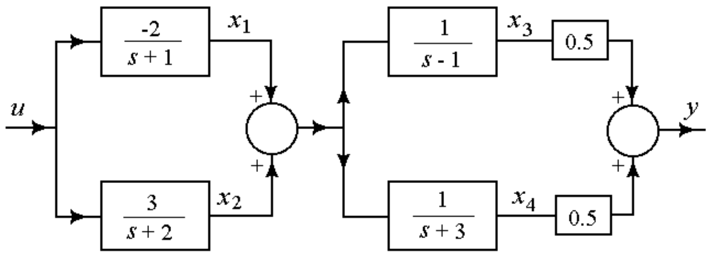


Figure 5.21: A block diagram for Exercise 5.8.

Exercise 5.9 Consider a linear time-invariant (LTI) system model,

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ 1 \\ -1 \\ 1 \\ 1 \\ 0 \\ 2 \end{bmatrix} u(t) \\ \boldsymbol{y}(t) &= \begin{bmatrix} 1 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 3 & 0 & 0 \end{bmatrix} \boldsymbol{x}(t).\end{aligned}$$

(i) Use the PBH test to determine if the system is controllable;

(ii) Use the PBH test to determine if the system is observable.

Exercise 5.10 Examples 1.7 and 5.6 are concerned with constructing realizations of the same transfer function matrix. Apply Theorem 5.4 to extract a minimal realization from the realization given in Example 1.7. You can use MATLAB's function `obsvf` to efficiently solve this part of the problem. Then compute an equivalent transformation relating the obtained minimal realization with the minimal realization given in Example 5.6 as shown in (5.26).

Exercise 5.11 Examples 1.8 and 5.6 are concerned with constructing realizations of the same transfer function matrix. Apply a dual version of Theorem 5.4 to extract a minimal realization from the realization given in Example 1.8. You can use MATLAB's function `ctrbf` to efficiently solve this part of the problem. Then compute an equivalent transformation relating the obtained minimal realization with the minimal realization given in Example 5.6 as shown in (5.26). Finally, construct an equivalent transformation relating the obtained minimal realization in this Exercise with the minimal realization given in Exercise 5.10.

Exercise 5.12 (adapted from Barnett [23, page 95]) A simple representation of a vehicle suspension system is shown in Figure 5.22. Assume that the forces produced by the springs and dampers act at the end points P and Q and x_1 and x_2 are the displacements of these points from equilibrium. The force exerted by a damper is proportional to the velocity of one end of the damper relative to the other end and the springs obey Hooke's law. Assume that the mass of the platform can be neglected so that the spring motions can be regarded as independent. The control force $4u$ is applied one quarter the way along from one end of

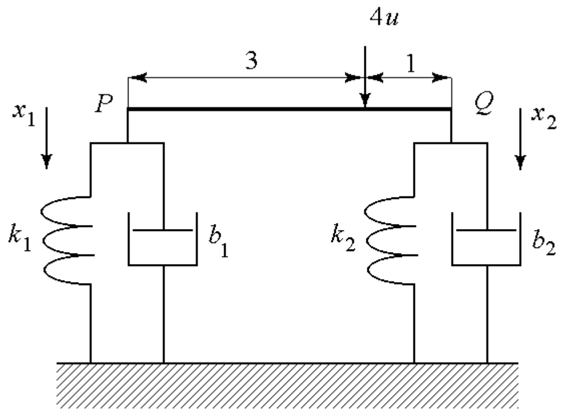


Figure 5.22: A simple representation of a vehicle suspension system of Exercise 5.12.

the platform. Derive the two-dimensional state space model of this simple representation of a vehicle suspension system. Next, suppose that the ends of the platform are each given an initial displacement of 10 units, find a control law that returns the platform to equilibrium at $t = 1$ sec.

Exercise 5.13 For the dynamical system model, $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u$, where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

design a state-feedback controller that stabilizes the system about the operating state corresponding to the constant input $u = 2$. The desired closed-loop poles are -2, -3.

Exercise 5.14 Consider the vehicle model given in Subsection 2.4.2. Let the input signal be $u = \delta_f$—the front wheel steer angle, and $\delta_r = 0$. Compute the transfer function $C(s)/U(s)$. Plot the vehicle response Y versus X for X changing from 0 m to 100 m, when the input δ_f, in the vehicle coordinates, has the form $\delta_f(t) = 0.0872\, h(t)$ rad, where $h(t)$ is the unit step function.

Suppose now that the plot of δ_f versus time, in the car coordinates, has the form shown in Figure 5.23. Note that the values of δ_f are given in deg. The value of δ_f changes form -2.5 deg to 2.5 deg. In your simulations you will need to convert deg to rad. Plot the vehicle response Y versus X for X changing from 0 m to 100 m.

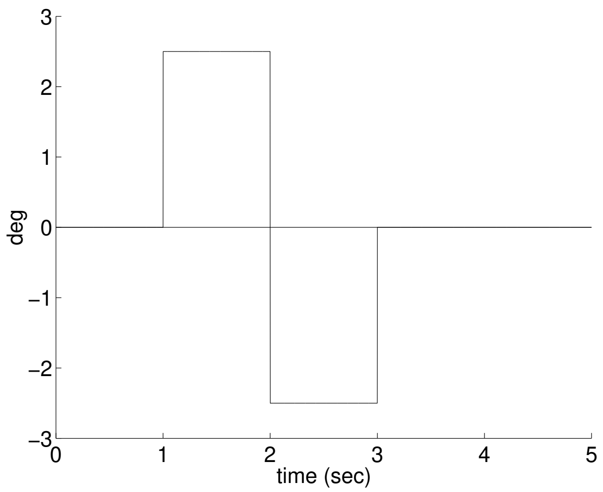


Figure 5.23: Time history of the front steer angle in the car coordinates of Exercise 5.14.

Construct the state-feedback controller $u = -\boldsymbol{k}\boldsymbol{x} + r$ such that $\operatorname{eig}(\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k}) = \{-2, -3+j, -3-j, -4\}$. Plot u and r versus time, and Y versus X, when $r = 0.0872\, h(t)$ rad.

Exercise 5.15 Show that the controllability property is invariant under the following set of transformations:

(i) nonsingular coordinate transformation in the state-space;

(ii) nonsingular transformation of the inputs;

(iii) state feedback;

(iv) output feedback.

Exercise 5.16 Given the following model of a dynamical system

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{x}(t).\end{aligned}$$

(a) Design a full-order asymptotic observer for the above system with the observer poles located at -2 and -3. Write the equations of the observer dynamics. Note that the pair $(\boldsymbol{A}, \boldsymbol{c})$ is already in the observer form, which should facilitate your observer design.

(b) Denote the observer state vector by $\boldsymbol{z}$. Let $u = -\begin{bmatrix} 2 & 3 \end{bmatrix} \boldsymbol{z} + v$. Find the transfer function of the closed-loop system $Y(s)/V(s)$.

Exercise 5.17 Let

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u = \begin{bmatrix} -1 & 1 \\ 0 & 2 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u, \\ y &= \boldsymbol{c}\boldsymbol{x} = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}. \end{aligned}$$

(a) Find the state transformation that transforms the pair $(\boldsymbol{A}, \boldsymbol{c})$ into the observer form.

(b) Find the representation of the dynamical system in the new coordinates.

Exercise 5.18 For the following model of a dynamical system,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x} + 2u, \end{aligned}$$

(a) design a full-order observer with its poles located at -3 and -4. Write the equations of the observer dynamics;

(b) denote the observer state by $\boldsymbol{z}$. Let $u = -\begin{bmatrix} 3 & 2 \end{bmatrix} \boldsymbol{z} + r$. Find the transfer function, $Y(s)/R(s)$, of the closed-loop system.

Exercise 5.19 Given a dynamical system modeled by:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u \\ y &= \boldsymbol{c}\boldsymbol{x}, \end{aligned}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & -3 \\ 4 & 2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 2 & 3 \end{bmatrix}.$$

(a) Design a reduced-order observer

$$\dot{z} = dz + eu + gy,$$

where $d = -5$ and $g = 1$.

(b) Write an expression for the estimated states.

Exercise 5.20 For the plant

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 1 & -1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u \\
y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},
\end{aligned}$$

(a) Design the reduced-order observer for $\boldsymbol{G} = [1]$ and $\boldsymbol{D} = [-2]$.

(b) Write the equations of the estimated state in terms of the plant's output and the observer state.

Exercise 5.21 Consider a dynamical system that consists of a cart with an inverted pendulum (point mass on a massless shaft) attached to it as depicted in Figure 5.24.

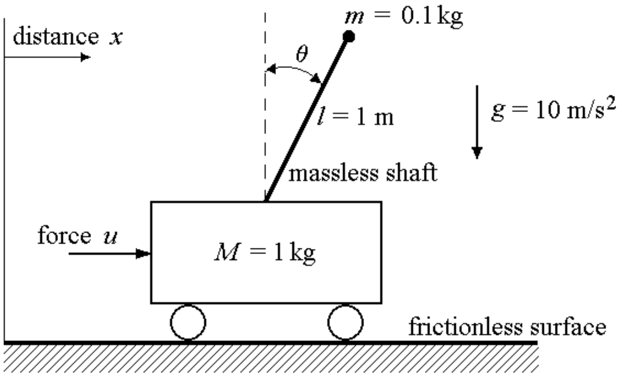


Figure 5.24: An inverted pendulum moving on a cart of Exercise 5.21.

(i) Obtain a nonlinear state-space of the system using the state variables

$$x_1 = x, \quad x_2 = \dot{x}, \quad x_3 = \theta, \quad x_4 = \dot{\theta}.$$

Designate x_1 as the system output.

(ii) Linearize the model about the operating point $\boldsymbol{x} = \mathbf{0}$, $u = 0$.

(iii) Design a state feedback control law $u = -\boldsymbol{k}\boldsymbol{x}$ such that the closed-loop poles are located at

$$-1, -2, -1 \pm j$$

(iv) Write a simple MATLAB script that animates the motion of the linearized closed-loop model of the inverted pendulum on the cart along with animation of the displacement x and the angle θ data. The basic handle graphics of MATLAB can be used to write the script. A script contained in Example 4 in the book [279, page 208] can be helpful here. Alternatively, you can employ SIMULINK.

Exercise 5.22 For the nonlinear model of the one-link robot manipulator of Subsection 2.4.5:

(a) Find the constant value of $u = u_e$ and x_{3e} so that the state $\boldsymbol{x}_e = \begin{bmatrix} 30\pi/180 & 0 & x_{3e} \end{bmatrix}^\top$ becomes the equilibrium state, that is, the obtained $\boldsymbol{x}_e$ and u_e satisfy the equation

$$\boldsymbol{0} = \boldsymbol{f}(\boldsymbol{x}_e, u_e).$$

(b) Linearize the nonlinear model, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u)$, about the operating point $\begin{bmatrix} \boldsymbol{x}_e^\top & u_e \end{bmatrix}^\top$ to obtain

$$\frac{d}{dt}\Delta\boldsymbol{x} = \boldsymbol{A}\Delta\boldsymbol{x} + \boldsymbol{b}\Delta u.$$

where,

$$\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_e, \quad \Delta u = u - u_e,$$

and

$$\boldsymbol{A} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_e, u_e) \quad \text{and} \quad \boldsymbol{b} = \frac{\partial \boldsymbol{f}}{\partial u}(\boldsymbol{x}_e, u_e).$$

(c) Design a state feedback control law $\Delta u = -\boldsymbol{k}\,\Delta\boldsymbol{x}$ such that the closed-loop poles are located at

$$-2, -2 \pm j$$

(d) Implement the obtained control law on the nonlinear model, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u)$. Write a MATLAB script that animates the dynamical behavior of the nonlinear closed-loop system along with animation of the angle θ_p and the input u data.

(e) Design a full-order observer for the linearized model. The observer's poles should be located at

$$-5, -5 \pm j$$

(f) Implement the combined controller-observer compensator on the nonlinear model and write a MATLAB script that animates dynamic behavior of the nonlinear closed-loop system along with animation of the angle θ_p and the input u data.

Exercise 5.23 For the nonlinear model of a double-link robot manipulator in Subsection 2.4.6, use the linearized model of this robot obtained in Exercise 4.5 to

(a) design a state-feedback controller using the linearized model. Write a MATLAB script that animates the behavior of the closed-loop system comprised of the state-feedback controller driving the nonlinear model of the robot for different initial conditions;

(b) use the linearized model to design a Luenberger observer. Implement and perform animation of the closed-loop system comprised of the combined controller-observer compensator driving the nonlinear model of the robot.

Chapter 6

A Glimpse at Linear Time-Varying (LTV) Systems

"As we shall see, the concept of time has no meaning before the beginning of the universe. This was first pointed out by St. Augustine. When asked: What did God do before he created the universe? Augustine didn't reply: He was preparing Hell for people who asked such questions. Instead, he said that time was a property of the universe that God created, and that time did not exist before the beginning of the universe." Stephen W. Hawking [133, page 8]

6.1 Motivating Example

Of considerable importance in many applications are linear systems in which the elements of $\boldsymbol{A}$ and $\boldsymbol{B}$ are continuous functions of time. We present a linear time-varying (LTV) model of a battery as a motivation for our discussion on linear time-varying systems. Our derivation follows that of Chiasson and Vairamohan [60, 61].

The simplest battery model is shown in Figure 6.1, which is called a practical voltage source. This model consists of an ideal voltage source that produces voltage V_{oc}. We refer to V_{oc} as the open-circuit voltage. Thus the open-circuit voltage is obtained by measuring the voltage across the open-circuit, that is, when $I_b = 0$. The internal resistance of the practical voltage source is denoted R_{int}. The battery terminal voltage is labeled as V_b. This model, however, does not take into account the fact that the internal resistance is different under discharge and charge conditions. To account for this, the circuit of Figure 6.1 is modified by introducing two different internal resistances; R_c—the charging resistance and R_d—the

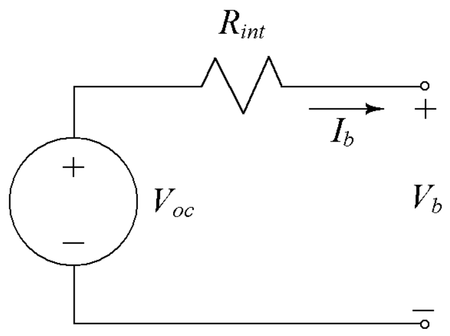


Figure 6.1: Practical voltage source as a simple battery model.

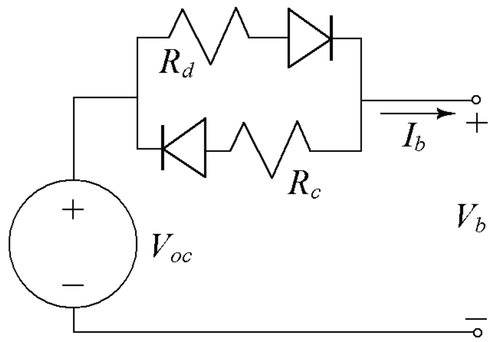


Figure 6.2: Battery model accounting for the different charging, R_c, and discharging, R_d, resistances.

discharging resistance. These resistances are connected to the diodes to model charging and discharging modes of the battery operation. The modified circuit is shown in Figure 6.2. This model, on the other hand, does not take into account the diffusion of the electrolytic through the battery. To model this effect, a polarization capacitor is added to model electrochemical reaction taking place in the battery. The resulting model is depicted in Figure 6.3.

In electric vehicles, the state of charge (SOC) of the battery corresponds to a fuel gauge in a conventional, internal combustion, cars. To define the state of charge, consider a completely charged battery. Let $I_b(t)$ be the discharging current. Then, the charge delivered by the

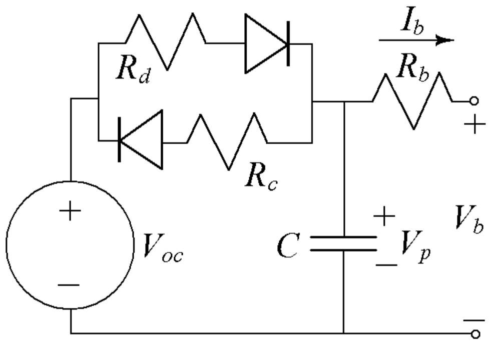


Figure 6.3: Battery model accounting for the different charging, R_c, and discharging, R_d, resistances and polarization capacitance, C.

battery is

$$Q(t) = \int_{t_0}^{t} I_b(\tau)d\tau.$$

Let $Q_0 = \int_{t_0}^{\infty} I_b(\tau)d\tau$ be the total charge that battery can hold. Then the state of charge (SOC) of the battery is

$$\begin{aligned} \text{SOC}(t) &= \frac{Q_0 - Q(t)}{Q_0} \times 100 \\ &= \frac{Q_0 - \int_{t_0}^{t} I_b(\tau)d\tau}{Q_0} \times 100. \end{aligned} \tag{6.1}$$

Direct application of (6.1) to estimate the battery SOC is not accurate. An alternative approach is to use an estimate of the open circuit voltage, V_{oc} of the battery. Then, the SOC can be evaluated from the relation,

$$\text{SOC}(t) = \frac{V_{oc}(t) - a_0}{a_1}, \tag{6.2}$$

where a_0 is the battery terminal voltage when $\text{SOC}(t) = 0\%$ and a_1 is obtained using a_0 and V_{oc} at $\text{SOC}(t) = 100\%$. Thus estimating the SOC is equivalent to estimating the battery open-circuit voltage $V_{oc}(t)$. Now, in order to measure V_{oc}, the battery must be disconnected from its load, which is not possible during the electric vehicle operation that uses the battery whose SOC we wish to estimate. The approach proposed by Chiasson and Vairamohan [60, 61] is to construct the open-circuit voltage estimator. This estimator construction is based on the battery model that we develop next.

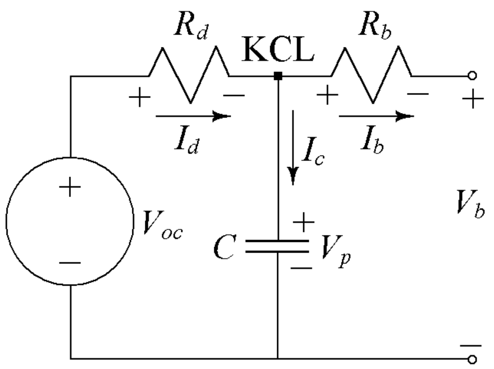


Figure 6.4: A battery model in the discharging mode of operation.

We analyze the condition when the battery is in the discharging mode, that is, when $I_b \geq 0$. The charging mode is analyzed in a similar way. The battery model in the discharging mode is depicted in Figure 6.4. To proceed, we apply Kirchhoff's current law (KCL) at the node between the resistances R_d and R_b to obtain

$$I_c + I_b - I_d = 0,$$

that is,

$$I_c = I_d - I_b.$$

Applying Ohm's law gives

$$I_d = \frac{V_{oc} - V_p}{R_d} \quad \text{and} \quad I_b = \frac{V_p - V_b}{R_b}. \tag{6.3}$$

We also have

$$I_c = C\dot{V}_p.$$

Combining the above equations yields

$$\begin{aligned} C\dot{V}_p &= I_d - I_b \\ &= \frac{V_{oc} - V_p}{R_d} - I_b \\ &= -\frac{V_p}{R_d} + \frac{V_{oc}}{R_d} - I_b. \end{aligned} \tag{6.4}$$

Dividing both sides by C, we obtain

$$\dot{V}_p = -\frac{V_p}{R_d C} + \frac{V_{oc}}{R_d C} - \frac{1}{C} I_b. \tag{6.5}$$

The output equation is obtained from the second of the equations in (6.3),

$$V_b = V_p - R_b I_b \tag{6.6}$$

We define the state variables,

$$x_1 = V_p, \quad x_2 = \frac{1}{R_d C}, \quad x_3 = \frac{V_{oc}}{R_d C}, \quad x_4 = \frac{1}{C}, \quad \text{and} \quad x_5 = R_b.$$

Applying the state variable definitions to (6.5) and (6.6), and taking into account that x_2, x_3, x_4, and x_5 are constant, we obtain a nonlinear time-varying state-space model,

$$\left.\begin{array}{rcl} \dot{x}_1 &=& -x_1 x_2 + x_3 - I_b(t) x_4 \\ \dot{x}_2 &=& 0 \\ \dot{x}_3 &=& 0 \\ \dot{x}_4 &=& 0 \\ \dot{x}_5 &=& 0 \\ y &=& V_b = x_1 - I_b(t) x_5 \end{array}\right\} \tag{6.7}$$

The initial state vector is unknown to us. The state variable x_3 depends on V_{oc}. Recall that the state variables, x_2, x_3, x_4 and x_5 are constant. Our objective is to compute the open-circuit voltage V_{oc}, which is a constant. We note that

$$x_3 = \frac{V_{oc}}{R_d C} = V_{oc} x_2.$$

Therefore

$$V_{oc} = \frac{x_3}{x_2} = \frac{x_{30}}{x_{20}}$$

because V_{oc} is constant. Hence, we can determine V_{oc} if we can estimate the initial values of the state variables x_{20} and x_{30}. We proceed by redefining the state variables as

$$z_1 = x_1, \quad z_2 = x_3, \quad z_3 = x_4, \quad z_4 = x_5.$$

With the above in mind, and using the fact that $x_2 = \text{constant} = x_{20}$, we re-write (6.7) as

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} -x_{20} & 1 & -I_b(t) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \tag{6.8}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & -I_b(t) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}. \tag{6.9}$$

The above is a LTV system because the current $I_b(t)$ is time-varying. In summary, we have a homogenous, or free, or unforced, LTV system model

$$\begin{aligned} \dot{\boldsymbol{z}} &= \boldsymbol{A}(t, x_{20})\boldsymbol{z}, \quad \boldsymbol{z}(t_0) \\ y &= \boldsymbol{c}(t)\boldsymbol{z}. \end{aligned}$$

Our objective is to estimate $\boldsymbol{z}(t_0)$ from knowledge of the output $y(t)$ and the current $I_b(t)$ on some time interval $[t_0,\ t_f]$.

We now outline a solution to the problem of estimating $\boldsymbol{z}(t_0)$. The first step is to compute the state-transition matrix, $\boldsymbol{\Phi}(t, t_0)$, which relates the state at time t to the initial state,

$$\boldsymbol{z}(t) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{z}(t_0).$$

We then use the output equation to obtain

$$\begin{aligned} y(t) &= \boldsymbol{c}\boldsymbol{z}(t) \\ &= \boldsymbol{c}\boldsymbol{\Phi}(t, t_0)\boldsymbol{z}(t_0). \end{aligned}$$

Premultiply both sides of the above by $\boldsymbol{\Phi}(t, t_0)^\top \boldsymbol{c}^\top$ and then integrate both sides to get

$$\int_{t_0}^{t_f} \boldsymbol{\Phi}(t, t_0)^\top \boldsymbol{c}^\top y(t) dt = \int_{t_0}^{t_f} \boldsymbol{\Phi}(t, t_0)^\top \boldsymbol{c}^\top \boldsymbol{c}\boldsymbol{\Phi}(t, t_0) dt\ \boldsymbol{z}(t_0).$$

Let

$$\boldsymbol{V}(t_0, t_f) = \int_{t_0}^{t_f} \boldsymbol{\Phi}(t, t_0)^\top \boldsymbol{c}^\top \boldsymbol{c}\boldsymbol{\Phi}(t, t_0) dt,$$

which, as in the LTI case, we refer to as the observability gramian. If $\boldsymbol{V}(t_0, t_f)$ is invertible then

$$\boldsymbol{z}(t_0) = \boldsymbol{V}^{-1}(t_0, t_f) \int_{t_0}^{t_f} \boldsymbol{\Phi}(t, t_0)^\top \boldsymbol{c}^\top y(t) dt.$$

Having obtained $\boldsymbol{z}(t_0)$ we can compute V_{oc} and then the SOC. We refer to Chiasson and Vairamohan [60, 61] for a complete solution to this problem; see also Pang et al. [262], and McIntyre et al. [238] for alternative solutions.

To be able to solve this problem, we need to acquaint ourselves with the state transition matrix for LTV systems. We start with the LTV uncontrolled system.

6.2 Solving the Uncontrolled LTV System

In general, it is not possible to give explicit expressions for solutions of linear time-varying systems. We obtain some general properties that for some cases will be used to obtain explicit solutions. We first consider the uncontrolled system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{6.10}$$

Theorem 6.1 *If $\boldsymbol{A}(t)$ is continuous for $t \geq t_0$ then the uncontrolled system modeled by* (6.10) *has a unique solution for $t \geq t_0$ given by*

$$\boldsymbol{x}(t) = \boldsymbol{X}(t)\boldsymbol{x}_0,$$

where $\boldsymbol{X}(t)$ is the unique $n \times n$ matrix satisfying

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{A}(t)\boldsymbol{X}, \quad \boldsymbol{X}(t_0) = \boldsymbol{I}_n.$$

Proof See Rugh [287, Chp. 3] or Antsaklis and Michel [13, Section 1.8]. See also Exercise 6.2.

□

Note that it follows from the above theorem that for every initial state $\boldsymbol{x}^{(i)}(t_0)$, $i = 1, 2, \ldots, n$, there exists a unique solution $\boldsymbol{x}^{(i)}(t)$, $i = 1, 2, \ldots, n$. We arrange these n solutions as

$$\boldsymbol{X}(t) = \begin{bmatrix} \boldsymbol{x}^{(1)}(t) & \boldsymbol{x}^{(2)}(t) & \cdots & \boldsymbol{x}^{(n)}(t) \end{bmatrix}.$$

Because every $\boldsymbol{x}^{(i)}(t)$ satisfies (6.10), we have

$$\dot{\boldsymbol{X}}(t) = \boldsymbol{A}(t)\boldsymbol{X}(t).$$

If $\boldsymbol{X}(t_0)$ is nonsingular, then $\boldsymbol{X}(t)$ is called a *fundamental matrix* of $\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x}$. Because the initial states are linearly independent, the fundamental matrix is nonsingular—see Exercise 6.1. The fundamental matrix is not unique because we can choose initial states arbitrarily.

We now analyze a scalar LTV case,

$$\dot{x} = a(t)x, \quad x(0) = x_0. \tag{6.11}$$

Recall that when $a(t) = a =$constant, then the solution has the form

$$x(t) = e^{at}x(0) = e^{\int_0^t a d\tau}x(0).$$

The above formula works for the scalar LTV case, that is,

$$x(t) = e^{\int_0^t a(\tau)d\tau}x(0). \tag{6.12}$$

It is easy to verify that the above is the solution of (6.11) because

$$\begin{aligned}\frac{d}{dt}e^{\int_0^t a(\tau)d\tau} &= e^{\int_0^t a(\tau)d\tau}a(t)\\ &= a(t)e^{\int_0^t a(\tau)d\tau}.\end{aligned}$$

That is, $a(t)$ and $\int_0^t a(\tau)d\tau$ commute.

Example 6.1 We use the solution formula (6.12) to solve the equation,

$$\dot{x} = (\sin t)x, \quad x(0) = x_0.$$

We have

$$\begin{aligned}x(t) &= e^{\int_0^t a(\tau)d\tau}x_0\\ &= x_0 e^{\int_0^t \sin(\tau)d\tau}\\ &= x_0\, e^{-\cos t}\Big|_0^t\\ &= e^{-\cos t+1}x_0\\ &= e\times e^{-\cos t}x_0.\end{aligned}$$

The solution formula (6.12) does not generalize to the multi-dimensional case because, in general,

$$\frac{d}{dt}e^{\int_0^t \boldsymbol{A}(\tau)d\tau} \neq \boldsymbol{A}(t)e^{\int_0^t \boldsymbol{A}(\tau)d\tau}.$$

If, however, $\boldsymbol{A}(t)$ and $\int_0^t \boldsymbol{A}(\tau)d\tau$ commute, then the solution to (6.10) can be written as

$$\boldsymbol{x}(t) = e^{\int_0^t \boldsymbol{A}(\tau)d\tau}\boldsymbol{x}(0).$$

We note that $\boldsymbol{A}(t)$ and $\int_0^t \boldsymbol{A}(\tau)d\tau$ commute if

$$\boldsymbol{A}(t)\boldsymbol{A}(s) = \boldsymbol{A}(s)\boldsymbol{A}(t).$$

We now analyze a general case when t_0 is not necessarily equal to 0. If $\boldsymbol{A}(t) = \boldsymbol{A}$ in (6.10), then the solution is

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0).$$

We re-write the above as

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}e^{-\boldsymbol{A}t_0}\boldsymbol{x}(t_0).$$

Let $\boldsymbol{X}(t) = e^{\boldsymbol{A}t}$, then

$$\boldsymbol{X}(t_0)^{-1} = e^{-\boldsymbol{A}t_0}$$

and we represent the solution as

$$\begin{aligned}\boldsymbol{x}(t) &= \boldsymbol{X}(t)\boldsymbol{X}(t_0)^{-1}\boldsymbol{x}(t_0)\\ &= \boldsymbol{\Phi}(t,t_0)\boldsymbol{x}(t_0),\end{aligned} \tag{6.13}$$

where $\boldsymbol{\Phi}(t,t_0)$ is the state transition matrix.

Definition 6.1 *Let $\boldsymbol{X}(t)$ be a fundamental matrix of* (6.10)*. Then*

$$\boldsymbol{\Phi}(t,t_0) = \boldsymbol{X}(t)\boldsymbol{X}(t_0)^{-1}$$

is the state transition matrix of (6.10)*. The state transition matrix is the unique solution of*

$$\frac{\partial}{\partial t}\boldsymbol{\Phi}(t,t_0) = \boldsymbol{A}(t)\boldsymbol{\Phi}(t,t_0), \quad \boldsymbol{\Phi}(t_0,t_0) = \boldsymbol{I}_n.$$

◇

As in the LTI case, the state transition matrix in the LTV case has the following properties (see Exercise 6.5):

$$\boldsymbol{\Phi}(\tau,\tau) = \boldsymbol{I}_n \tag{6.14}$$

$$\boldsymbol{\Phi}(t,\tau)^{-1} = \boldsymbol{\Phi}(\tau,t) \tag{6.15}$$

$$\boldsymbol{\Phi}(t,t_0) = \boldsymbol{\Phi}(t,t_1)\boldsymbol{\Phi}(t_1,t_0) \tag{6.16}$$

$$\frac{\partial}{\partial\tau}\boldsymbol{\Phi}(t,\tau) = -\boldsymbol{\Phi}(t,\tau)\boldsymbol{A}(\tau), \text{ for all } t,\tau \tag{6.17}$$

The third of the above properties is referred to as the *semigroup property.*

Example 6.2 The state transition matrix of

$$\dot{x} = (\sin t)x, \quad x(t_0) = x_0,$$

has the form

$$\begin{aligned}\Phi(t,t_0) &= X(t)X(t_0)^{-1}\\ &= \frac{e^{-\cos t+1}}{e^{-\cos t_0+1}}.\end{aligned}$$

Example 6.3 We construct the state transition matrix for the LTV system,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & -t \\ 0 & 0 \end{bmatrix} \boldsymbol{x}, \quad \boldsymbol{x}(0) = \boldsymbol{x}(0).$$

The above system consists of two first-order linear differential equations,

$$\dot{x}_1 = -tx_2, \quad \text{and} \quad \dot{x}_2 = 0.$$

We can easily obtain the solution to the second of the above equations,

$$x_2(t) = x_2(0). \tag{6.18}$$

We next obtain the solution to $\dot{x}_1 = -tx_2$ by integrating its both sides,

$$\begin{aligned} x_1(t) &= x_1(0) - \int_0^t \tau x_2(0) d\tau \\ &= x_1(0) - \frac{t^2 x_2(0)}{2}. \end{aligned} \tag{6.19}$$

We can now construct a fundamental matrix. Let

$$\boldsymbol{x}^{(1)}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{x}^{(2)}(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Then using (6.19) and (6.18), we obtain the fundamental matrix corresponding to the above initial conditions is

$$\begin{aligned} \boldsymbol{X}(t) &= \begin{bmatrix} \boldsymbol{x}^{(1)}(t) & \boldsymbol{x}^{(2)}(t) \end{bmatrix} \\ &= \begin{bmatrix} 1 - \frac{t^2}{2} & -\frac{t^2}{2} \\ 1 & 1 \end{bmatrix}. \end{aligned}$$

If we select the initial conditions

$$\boldsymbol{x}^{(1)}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{x}^{(2)}(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

then we obtain a different fundamental matrix,

$$\boldsymbol{X}(t) = \begin{bmatrix} 1 & -\frac{t^2}{2} \\ 0 & 1 \end{bmatrix}.$$

The state transition matrix, however, is unique and has the form

$$\begin{aligned}\boldsymbol{\Phi}(t,t_0) &= \boldsymbol{X}(t)\boldsymbol{X}(t_0)^{-1}\\ &= \begin{bmatrix} 1-\frac{t^2}{2} & -\frac{t^2}{2}\\ 1 & 1\end{bmatrix}\begin{bmatrix} 1-\frac{t_0^2}{2} & -\frac{t_0^2}{2}\\ 1 & 1\end{bmatrix}^{-1}\\ &= \begin{bmatrix} 1-\frac{t^2}{2} & -\frac{t^2}{2}\\ 1 & 1\end{bmatrix}\begin{bmatrix} 1 & \frac{t_0^2}{2}\\ -1 & 1-\frac{t_0^2}{2}\end{bmatrix}\\ &= \begin{bmatrix} 1 & \frac{1}{2}(t_0^2-t^2)\\ 0 & 1\end{bmatrix}.\end{aligned}$$

Example 6.4 We construct the state-transition matrix for the LTV system model found in Luenberger [223, p. 115],

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 0\\ t & \frac{1}{t}\end{bmatrix}\boldsymbol{x}, \quad t \ge 1.$$

Then, for the given $\boldsymbol{x}(2) = \begin{bmatrix} 2 & -1\end{bmatrix}^\top$, we compute $\boldsymbol{x}(1)$.

First, we solve for x_1 to obtain,

$$x_1(t) = x_1(1), \quad t \ge 1$$

Substituting the above into the equation for $\dot{x}_2$ gives

$$\dot{x}_2 = t x_1(t) + \frac{x_2(t)}{t}.$$

Next, we make the substitution

$$z(t) = \frac{x_2(t)}{t}.$$

Computing the derivative of $z(t)$ yields

$$\begin{aligned}\dot{z} &= \frac{\dot{x}_2 t - x_2}{t^2}\\ &= \frac{\dot{x}_2}{t} - \frac{x_2}{t^2}.\end{aligned}$$

Multiplying both sides of the above by t, we obtain

$$\begin{aligned}t\dot{z} &= \dot{x}_2 - \frac{x_2}{t^2}\\ &= \dot{x}_2 - z.\end{aligned} \tag{6.20}$$

We substitute $\dot{x}_2 = tx_1(1) + z$ into 6.20 to obtain

$$t\dot{z} + z = tx_1(1) + z,$$

that is,

$$\dot{z} = x_1(1).$$

Integrating the above from 1 to t gives

$$z(t) = z(1) + x_1(1)(t-1).$$

Therefore,

$$\begin{aligned} z &= \frac{x_2}{t} \\ &= z(1) + x_1(1)(t-1), \end{aligned}$$

and multiplying both sides by t, we obtain

$$x_2(t) = tx_2(1) + tx_1(1)(t-1).$$

We are now ready to generate a fundamental matrix. Let

$$\boldsymbol{x}^{(1)}(1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{x}^{(2)}(1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

then we have

$$\boldsymbol{x}^{(1)}(t) = \begin{bmatrix} 1 \\ t^2 - t \end{bmatrix} \quad \text{and} \quad \boldsymbol{x}^{(2)}(t) = \begin{bmatrix} 0 \\ t \end{bmatrix}.$$

The fundamental matrix corresponding to the above initial conditions is

$$\boldsymbol{X}(t) = \begin{bmatrix} \boldsymbol{x}^{(1)}(t) & \boldsymbol{x}^{(2)}(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ t^2 - t & t \end{bmatrix}.$$

The state transition matrix is

$$\begin{aligned} \boldsymbol{\Phi}(t,\tau) &= \boldsymbol{X}(t)\boldsymbol{X}(\tau)^{-1} \\ &= \begin{bmatrix} 1 & 0 \\ t^2 - t & t \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tau^2 - \tau & \tau \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1 & 0 \\ t^2 - t & t \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\tau + 1 & \frac{1}{\tau} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ t(t-\tau) & \frac{t}{\tau} \end{bmatrix}. \end{aligned}$$

Note that

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t,\tau)\boldsymbol{x}(\tau).$$

Thus, for example, if we are given $\boldsymbol{x}(2) = \begin{bmatrix} 2 & -1 \end{bmatrix}^\top$ and we wish to find $\boldsymbol{x}(1)$, we have,

$$\begin{aligned} \boldsymbol{x}(1) &= \boldsymbol{\Phi}(1,2)\boldsymbol{x}(2) \\ &= \begin{bmatrix} 1 & 0 \\ -1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ -\frac{5}{2} \end{bmatrix}. \end{aligned}$$

6.3 Solving the LTV Controlled System

We now discuss the problem of obtaining the solution to the LTV controlled system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u}, \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{6.21}$$

We claim that, by analogy with the LTI controlled system, the solution to (6.21) is

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t,t_0)\boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\boldsymbol{B}(\tau)\boldsymbol{u}(\tau)d\tau. \tag{6.22}$$

We can prove our claim by verification utilizing Leibniz's rule (see Exercise 6.7),

$$\frac{\partial}{\partial t}\int_{t_0}^{t} f(t,\tau)d\tau = \int_{t_0}^{t}\left(\frac{\partial}{\partial t}f(t,\tau)\right)d\tau + f(t,\tau)|_{\tau=t} \tag{6.23}$$

Example 6.5 (R. A. DeCarlo) The LTV state model,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \begin{bmatrix} \pi e^{-\alpha(t)^2} \\ 0 \end{bmatrix} u(t),$$

has associated state transition matrix

$$\boldsymbol{\Phi}(t,\tau) = e^{\alpha(\tau)^2-\alpha(t)^2}\begin{bmatrix} 1 & 0 \\ \cos(\pi t)-\cos(\pi\tau) & 1 \end{bmatrix},$$

where $\alpha(\cdot)$ is a continuous function.

1. Compute $\boldsymbol{A}(t)$ using the given state transition matrix.

2. Find the zero-state response when the input $u(t)$ is the unit step.

We first compute $\boldsymbol{A}(t)$ using the given state transition matrix. We use the state transition matrix properties to accomplish this. Recall that the state transition matrix satisfies the equation

$$\frac{\partial}{\partial t}\boldsymbol{\Phi}(t,\tau) = \boldsymbol{A}(t)\boldsymbol{\Phi}(t,\tau), \quad \boldsymbol{\Phi}(\tau,\tau) = \boldsymbol{I}_n.$$

Note that, we can also write.

$$\boldsymbol{\Phi}(t,\tau)\big|_{\tau=t} = \boldsymbol{I}_n$$

Taking this into account, we obtain

$$\boldsymbol{A}(t) = \frac{\partial}{\partial t}\boldsymbol{\Phi}(t,\tau)\bigg|_{\tau=t}.$$

To proceed, we first evaluate

$$\begin{aligned} \frac{\partial}{\partial t}e^{\alpha(\tau)^2-\alpha(t)^2}\bigg|_{\tau=t} &= e^{\alpha(\tau)^2-\alpha(t)^2}(-2\alpha(t)\dot{\alpha}(t))\bigg|_{\tau=t} \\ &= -2\alpha(t)\dot{\alpha}(t). \end{aligned}$$

We next evaluate the derivative

$$\begin{aligned} \frac{\partial}{\partial t}\left(e^{\alpha(\tau)^2-\alpha(t)^2}\left(\cos(\pi t)-\cos(\pi\tau)\right)\right)\bigg|_{\tau=t} &= -2\alpha(t)\dot{\alpha}(t)(\cos(\pi t)-\cos(\pi t)) - \pi\sin(\pi t) \\ &= -\pi\sin(\pi t). \end{aligned}$$

Hence

$$\begin{aligned} \boldsymbol{A}(t) &= \frac{\partial}{\partial t}\boldsymbol{\Phi}(t,\tau)\bigg|_{\tau=t} \\ &= \begin{bmatrix} -2\alpha(t)\dot{\alpha}(t) & 0 \\ -\pi\sin(\pi t) & -2\alpha(t)\dot{\alpha}(t) \end{bmatrix}. \end{aligned}$$

We now compute the zero-state response. The formula for the zero-state response is obtained from (6.22) by setting $\boldsymbol{x}_0 = \boldsymbol{0}$,

$$\boldsymbol{x}(t) = \int_{t_0}^{t}\boldsymbol{\Phi}(t,\tau)\boldsymbol{B}(\tau)\boldsymbol{u}(\tau)d\tau.$$

We first compute the integrand

$$\begin{aligned}\boldsymbol{\Phi}(t,\tau)\boldsymbol{B}(\tau) &= e^{\alpha(\tau)^2-\alpha(t)^2}\begin{bmatrix} 1 & 0 \\ \cos(\pi t)-\cos(\pi\tau) & 1\end{bmatrix}\begin{bmatrix}\pi e^{-\alpha(\tau)^2} \\ 0\end{bmatrix} \\ &= \pi e^{-\alpha(t)^2}\begin{bmatrix} 1 \\ \cos(\pi t)-\cos(\pi\tau)\end{bmatrix}.\end{aligned}$$

Therefore, the zero-state response when $u(t)=1(t)$ is

$$\begin{aligned}\boldsymbol{x}(t) &= \int_{t_0}^{t}\boldsymbol{\Phi}(t,\tau)\boldsymbol{B}(\tau)\boldsymbol{u}(\tau)d\tau \\ &= \pi e^{-\alpha(t)^2}\int_0^t\begin{bmatrix} 1 \\ \cos(\pi t)-\cos(\pi\tau)\end{bmatrix}d\tau \\ &= \pi e^{-\alpha(t)^2}\begin{bmatrix} t \\ t\cos(\pi t)-\frac{\sin(\pi\tau)}{\pi}\end{bmatrix}.\end{aligned}$$

6.4 Notes

For additional information on the fundamentals of the LTV systems, we recommend Kamen [173]. An in-depth treatment of the LTV systems can be found in Rugh [288], and in Antsaklis and Michel [13]. For results on observers for LTV systems, see O'Reilly [258]. A novel approach to the analysis and control of LTV systems is proposed by Bourlès and Marinescu [39].

6.5 Exercises

Exercise 6.1 Prove that a fundamental matrix is nonsingular for any t.

Exercise 6.2 Use the contraction map theorem A.19 to prove that (6.10) has a unique solution going through $\boldsymbol{x}_0$ at $t=t_0$, where t_0 is an element of a finite time interval $[a,\ b]$. (Start with the Lipschitz condition.)

Exercise 6.3 **(a)** Find the state-transition matrix $\boldsymbol{\Phi}(t,\tau)$ associated with a homogenous LTI system with

$$\boldsymbol{A}=\boldsymbol{T}\begin{bmatrix} -2 & 0 \\ 0 & 0\end{bmatrix}\boldsymbol{T}^{-1},$$

where

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix}.$$

(b) Find the state-transition matrix $\boldsymbol{\Phi}(t,\tau)$ associated with a homogenous LTV system with

$$\boldsymbol{A}(t) = \boldsymbol{T}\begin{bmatrix} -2 & 0 \\ 0 & \cos t \end{bmatrix}\boldsymbol{T}^{-1},$$

where

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix}.$$

Exercise 6.4 Compute the state transition matrix $\boldsymbol{\Phi}(t,\tau)$ for the linear time-varying (LTV) system model,

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 0 \\ 1 & \frac{1}{t} \end{bmatrix}\boldsymbol{x}(t)$$

defined for $t \geq 1$. Then, for the given $\boldsymbol{x}(2) = \begin{bmatrix} 0 & 4 \end{bmatrix}^{\top}$, compute $\boldsymbol{x}(1)$.

Exercise 6.5 Use the definition of the state transition matrix to prove its properties (6.14), (6.15), (6.16), and (6.17).

Exercise 6.6 **(a)** The value of the state-transition matrix for a homogeneous linear system is

$$\boldsymbol{\Phi}(1,3) = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

Find $\boldsymbol{x}(3)$ when

$$\boldsymbol{x}(1) = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

(b) The values of the state-transition matrix for a homogeneous linear system is

$$\boldsymbol{\Phi}(1,3) = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Phi}(2,3) = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}.$$

If

$$\boldsymbol{x}(1) = \begin{bmatrix} 2 \\ -3 \end{bmatrix},$$

find $\boldsymbol{x}(2)$.

Exercise 6.7 Derive Leibniz's rule given by (6.23).

Exercise 6.8 **(a)** The state-transition matrix of a homogenous LTV system is

$$\boldsymbol{\Phi}(t,\tau) = \begin{bmatrix} e^{\cos t-\cos\tau} & 0 \\ 0 & e^{-\sin t+\sin\tau} \end{bmatrix}.$$

Find the associated $\boldsymbol{A}(t)$-matrix.

(b) A fundamental matrix associated with a homogenous continuous-time LTV state model has the form,

$$\boldsymbol{X}(t) = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t^2-t & t \end{bmatrix}.$$

Compute the state-transition matrix $\boldsymbol{\Phi}(t,\tau)$.

Exercise 6.9 (R. Balakrishnan) For the linear time-varying (LTV) system model,

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & -2t \\ 0 & -t \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ t \end{bmatrix} u(t),$$

- Find the state transition matrix $\boldsymbol{\Phi}(t,\tau)$;
- Let $\boldsymbol{x}(1) = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ and $u(t) = 0$ for $t \geq 0$. Find $\boldsymbol{x}(0)$;
- Let $\boldsymbol{x}(0) = \boldsymbol{0}$ and $u(t) = 1$ for $t \geq 0$. Find $\boldsymbol{x}(t)$ for $t \geq 0$.

Chapter 7

Stability

> With the spread of the centrifugal governor in the early 19th century a number of major problems became apparent. First, because of the absence of integral action, the governor could not remove offset: in the terminology of the time it could not regulate but only moderate. Second, its response to a change in load was slow. And thirdly, (nonlinear) frictional forces in the mechanism could lead to hunting (limit cycling). A number of attempts were made to overcome these problems: for example, the Siemens chronometric governor effectively introduced integral action through differential gearing, as well as mechanical amplification. Other approaches to the design of an isochronous governor (one with no offset) were based on ingenious mechanical constructions, but often encountered problems of stability.—Bissell [36, p. 56]

7.1 An Informal Introduction to Stability

In general, it can be difficult to obtain an explicit solution of nonlinear differential equations modeling a dynamical system. However, often we do not need to know the explicit solution of the modeling equations. Instead, we are interested in the behavior of the solution as time tends to infinity. Furthermore, in many models of physical systems a "small" change in the initial conditions results in a "small" change in the solution. Specifically, if a system is perturbed from its rest, or equilibrium position, then it starts to move. We can roughly say that the equilibrium position is stable if the system does not go far from this position for small initial perturbations. This concept may be illustrated, as in Willems [351], Brogan [44], or in Barnett [23], by means of a ball resting in equilibrium on a sheet of metal bent into

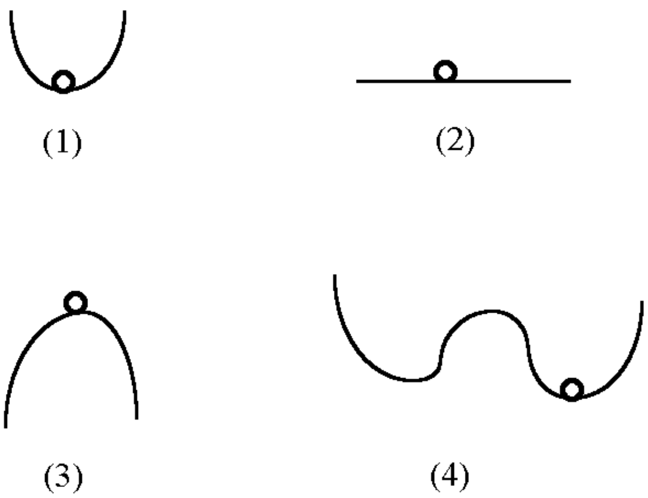


Figure 7.1: Illustration of stable and unstable equilibrium positions.

various shapes with cross-sections depicted in Figure 7.1. This figure illustrates four possible scenarios. If we neglect frictional forces, a small perturbation from the equilibrium leads to:

(1) oscillatory motion about equilibrium irrespective of the magnitude of the initial perturbation. In this case the equilibrium position is globally stable,

(2) the new position is also an equilibrium and the ball will remain in it. Here the equilibrium position is stable,

(3) the ball moving away without returning to the equilibrium. This equilibrium position is unstable,

(4) oscillatory motion about the equilibrium, unless the initial perturbation is so large that the ball is forced to oscillate about the new equilibrium position. This equilibrium position is locally stable.

If friction is taken into account, then the oscillatory motion steadily decreases until the ball returns to the equilibrium positions as in items (1) and (4) above. We refer to such equilibrium positions as being asymptotically stable. In our discussion of stability concepts, we implicitly assume, as in item (4), that there is some “small” region for allowed initial perturbations for which subsequent motions converge to the equilibrium position. If however, as in item (1), any initial perturbation yields a motion that converges to the equilibrium position, then we have global asymptotic stability, which is also referred to as asymptotic stability in the large.

The above informal discussion should serve as a preparation for an in-depth analysis of stability of dynamic systems. We begin our analysis by formally defining the equilibrium state. We then define stability, asymptotic stability, and other relevant notions.

7.2 Basic Definitions of Stability

We consider a class of dynamical systems modeled by the equation:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), \ \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{7.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state vector and $\boldsymbol{f} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued function with the components

$$f_i(t, x_1, x_2, \ldots, x_n) : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}, \ i = 1, 2, \ldots, n.$$

We assume that f_i's are continuous and have continuous first partial derivatives. Under these assumptions, the solution to $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), \ \boldsymbol{x}(t_0) = \boldsymbol{x}_0$, exists and is unique. We denote this solution as $\boldsymbol{x}(t) = \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$. If the f_i's do not depend explicitly on t, then the system represented by $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)) = \boldsymbol{f}(\boldsymbol{x}(t))$ is called *autonomous* or *time-invariant*, otherwise it is called *non-autonomous* or *time-varying*.

A point $\boldsymbol{x}_e$ is an *equilibrium point*, or state, at time 0 if for all $t \geq 0$,

$$\boldsymbol{f}(t, \boldsymbol{x}_e) = \boldsymbol{0}.$$

We note that if $\boldsymbol{x}_e$ is an equilibrium state at time 0, then it is also an equilibrium state for all initial times greater than or equal 0.

We also note that an equilibrium state $\boldsymbol{x}_e$ is a constant solution to (7.1) for $t \geq 0$.

Observe that for the time-invariant system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x})$$

a point is an equilibrium at some time if and only if is is an equilibrium at all times.

If $\boldsymbol{x}_e$ is an equilibrium point, we can arrange for this equilibrium point to be transferred to the origin of $\mathbb{R}^n$ by introducing the new variable

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_e.$$

We assume that this has been done for an equilibrium point under consideration. Therefore, we have

$$\boldsymbol{f}(t, \mathbf{0}) = \mathbf{0} \quad \text{for all} \quad t \geq t_0.$$

There may be several equilibrium points for a given system. We assume that there is no other equilibrium point in a neighborhood of the origin of $\mathbb{R}^n$. Thus, the origin is an isolated equilibrium point. We formally define isolated equilibrium in Definition 4.3.

We now introduce basic definitions of stability in the sense of A. M. Lyapunov[1] (1857-1918). In the following considerations we use the standard Euclidean norm of a vector. Thus, if $\boldsymbol{x} \in \mathbb{R}^n$ then

$$\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = (\boldsymbol{x}^\top \boldsymbol{x})^{\frac{1}{2}}.$$

Definition 7.1 *An equilibrium state $\boldsymbol{x}_e$ is said to be stable if for any given $t_0 \geq 0$ and any positive scalar ε, there exists a positive scalar $\delta = \delta(t_0, \varepsilon)$ such that if $\|\boldsymbol{x}(t_0) - \boldsymbol{x}_e\| < \delta$, then $\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0) - \boldsymbol{x}_e\| < \varepsilon$ for all $t \geq t_0$.*

◇

Friedland [108, page 43] makes the following comment regarding the above definition of stability: "If this "epsilonics" definition leaves you cold, think of a contest between you, the system designer, and an adversary (nature?). The adversary picks a region in state space of radius ε and challenges you to find another region, of radius δ, such that if the initial state starts out inside your region, it remains inside his region. If you can do this, your system design is stable." We add that you have to find an appropriate δ for any choice of ε for the equilibrium state to be stable.

In summary, an equilibrium state, $\boldsymbol{x}_e$, is stable if whenever the initial state is near $\boldsymbol{x}_e$, the state trajectory, $\boldsymbol{x}(t)$, remains near the equilibrium, perhaps even tending toward this equilibrium state as time increases. Note that Lyapunov's stability does not imply that $\boldsymbol{x}(t)$ converges to $\boldsymbol{x}_e$ as $t \to \infty$. The states are only required to "hover" around the equilibrium state $\boldsymbol{x}_e$. However, the stability condition bounds the amount of wiggling room for $\boldsymbol{x}(t)$ in the sense that small perturbations in the initial conditions produce small perturbations in the solution $\boldsymbol{x}(t)$.

Definition 7.2 *An equilibrium state $\boldsymbol{x}_e$ is said to be* convergent, *or* attractive, *if for any given $t_0 \geq 0$ there is a positive scalar $\delta_1 = \delta_1(t_0)$ such that if $\|\boldsymbol{x}(t_0) - \boldsymbol{x}_e\| < \delta_1$, then*

$$\lim_{t \to \infty} \boldsymbol{x}(t; t_0, \boldsymbol{x}_0) = \boldsymbol{x}_e.$$

[1] Sometimes transliterated in the literature as Liapunov or Liapunoff.

◇

We can restate Definition 7.2 as follows. An equilibrium state $\boldsymbol{x}_e$ is convergent if for any $t_0 \geq 0$ there is a positive scalar $\delta_1 = \delta_1(t_0)$ such that if $\|\boldsymbol{x}(t_0) - \boldsymbol{x}_e\| < \delta_1$, then for any given $\varepsilon_1 > 0$ there exists a $T = T(\varepsilon_1, t_0, \boldsymbol{x}_0)$ such that $\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0) - \boldsymbol{x}_e\| < \varepsilon_1$ for all $t > t_0 + T$.

Definition 7.3 *An equilibrium state $\boldsymbol{x}_e$ is asymptotically stable if it is stable and convergent.*

◇

We note that for linear time-invariant (LTI) systems asymptotic stability is equivalent with convergence. That is, in a convergent LTI system, stability condition is automatically satisfied. In general, however, it is not sufficient that just

$$\boldsymbol{x}(t; t_0, \boldsymbol{x}_0) \to \boldsymbol{x}_e \quad \text{as} \quad t \to \infty$$

for asymptotic stability. We need stability too! Why? Hahn [129], see also [12, p. 451], constructed a nonlinear system example 7.2 where convergence does not mean asymptotic stability as in the case of linear time-invariant systems. In this example the origin is attractive but not asymptotically stable.

Example 7.1 This is a second-order nonlinear system described by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{x_1^2(x_2 - x_1) + x_2^5}{(x_1^2 + x_2^2)(1 + (x_1^2 + x_2^2)^2)} \\ \frac{x_2^2(x_2 - 2x_1)}{(x_1^2 + x_2^2)(1 + (x_1^2 + x_2^2)^2)} \end{bmatrix}. \tag{7.2}$$

A phase portrait of the above system is shown in Figure 7.2. All trajectories, solutions, are approaching the equilibrium state $\boldsymbol{x}_e = \boldsymbol{0}$, without this equilibrium being asymptotically stable.

Definition 7.4 *An equilibrium state $\boldsymbol{x}_e$ is said to be uniformly stable if the positive scalar $\delta = \delta(t_0, \varepsilon)$ introduced in Definition 7.1 can be taken independent of the initial time t_0.*

◇

Example 7.2 The following example that comes from Rugh [288, page 100] nicely illustrates the notion of uniform stability. The solution to the differential equation,

$$\dot{x}(t) = (4t \sin t - 2t)x(t), \quad x(t_0) = x_0 \tag{7.3}$$

is

$$x(t) = x_0 \exp\left(4 \sin t - 4t \cos t - t^2 - 4 \sin t_0 + 4t_0 \cos t_0 + t_0^2\right). \tag{7.4}$$

One can verify this using MATLAB's Symbolic Toolbox by typing the command,

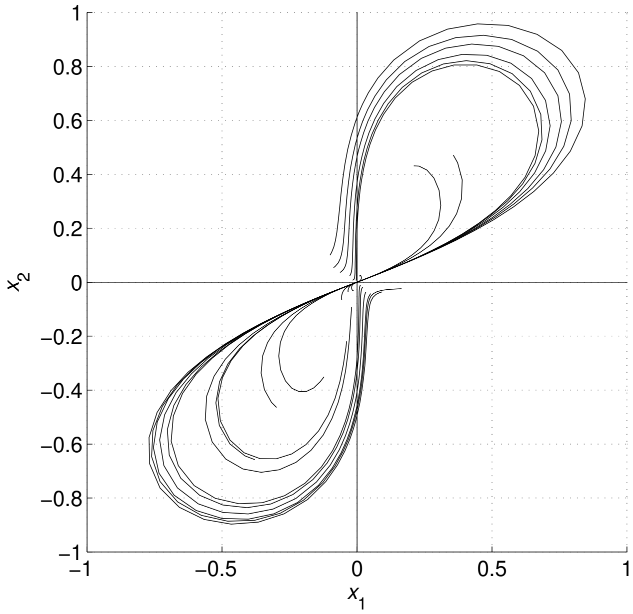


Figure 7.2: A phase portrait of system (7.2) where the origin is attractive but not asymptotically stable.

```
x=dsolve('Dx=(4*t*sin(t)-2*t)*x, x(t_0)=x_0','t')
```

It is easy to show that for fixed t_0 there is a γ such that (7.4) is bounded by $\gamma|x_0|$ for all $t \geq t_0$. This is because the term $(-t^2)$ dominates the exponent as t increases. Therefore, for a given ε, it is enough to choose x_0 so that $\gamma|x_0| < \varepsilon$, or equivalently $\delta < \varepsilon/\gamma$, to show that the equilibrium solution $x_e = 0$ is stable. However, the equilibrium solution $x_e = 0$ is not uniformly stable. To show this, fix initial state x_0 and consider a sequence of initial times $t_0 = 2k\pi$, where $k = 0, 1, \ldots$, and the values of the respective solutions at times π units later:

$$x(2k\pi + \pi) = x_0 \exp\left((4k+1)\pi(4-\pi)\right).$$

One can see that there is no bound on the exponential factor and hence a candidate bound γ increases as k increases, where increasing k corresponds to the increasing initial time.

Definition 7.5 *An equilibrium state $\boldsymbol{x}_e$ is said to be uniformly convergent, or uniformly attractive, if the positive scalars $\delta_1 = \delta_1(t_0)$ and $T = T(\varepsilon_1, t_0, \boldsymbol{x}_0)$ introduced in Definition 7.2 are independent of t_0.*

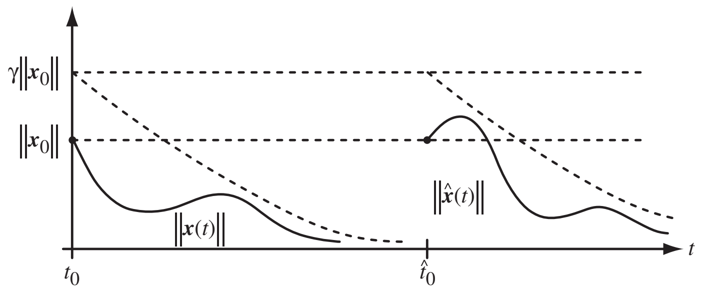


Figure 7.3: Illustration of uniform exponential stability: a decaying-exponential bound is independent of the initial time t_0.

◇

The last two definitions allow us to introduce the notion of uniform asymptotic stability.

Definition 7.6 *An equilibrium state $\boldsymbol{x}_e$ is said to be uniformly asymptotically stable if it is uniformly stable and uniformly convergent.*

◇

Notice that the solution $\boldsymbol{x}(t) = \boldsymbol{x}_e$ must first be stable in order to qualify to be asymptotically stable. This is to prevent a system trajectory from straying arbitrarily far from the equilibrium state before converging toward it. Note also that the concepts of Lyapunov stability and asymptotic stability are local concepts. They apply to the system's behavior in the small or in a "small" neighborhood about the equilibrium point (Hsu and Meyer [145]).

A special case of uniform asymptotic stability is *uniform exponential stability* that implies uniform stability, and imposes an additional requirement that all solutions originating "close" to the given equilibrium, $\boldsymbol{x}_e = \mathbf{0}$, approach this equilibrium exponentially as $t \to \infty$. More rigorously, we say that $\boldsymbol{x}_e = \mathbf{0}$ is uniformly exponentially stable if it is stable and there exist finite positive constants γ and λ such that for any t_0 and $\boldsymbol{x}_0$, any solution starting in a neighborhood of $\mathbf{0}$ satisfies

$$\|\boldsymbol{x}(t)\| \leq \gamma e^{-\lambda(t-t_0)} \|\boldsymbol{x}_0\| .$$

Note that γ is no less than unity, and uniform implies that γ and λ are independent of t_0—see Figure 7.3.

An equilibrium state is unstable if it is not stable. An ε-δ definition of instability is given below.

Definition 7.7 *An equilibrium state $\boldsymbol{x}_e$ is said to be unstable if there is an $\varepsilon > 0$ so that for any $\delta > 0$ there exists $\boldsymbol{x}(t_0)$ such that if $\|\boldsymbol{x}(t_0) - \boldsymbol{x}_e\| < \delta$, then $\|\boldsymbol{x}(t_1) - \boldsymbol{x}_e\| \geq \varepsilon$ for some $t_1 > t_0$.*

◇

A two dimensional geometric interpretation of the above definitions is shown in Figure 7.4. We assume that $\boldsymbol{x}_e = \mathbf{0}$ is the equilibrium state. If the equilibrium state is stable, as in

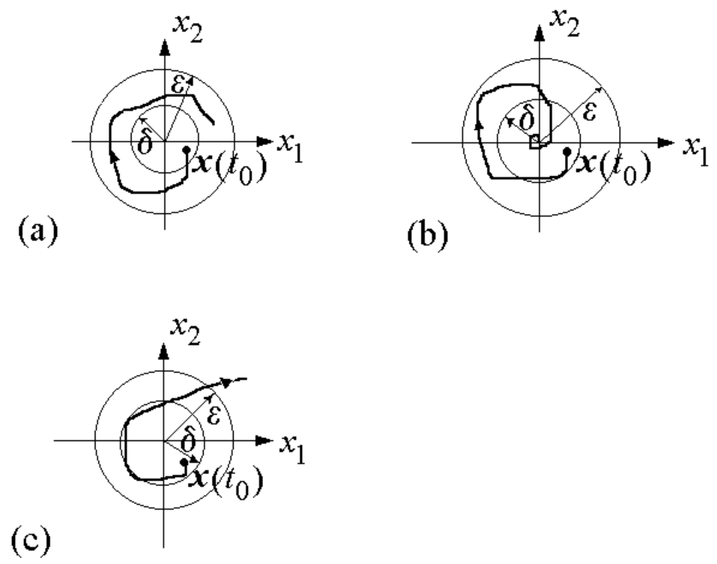
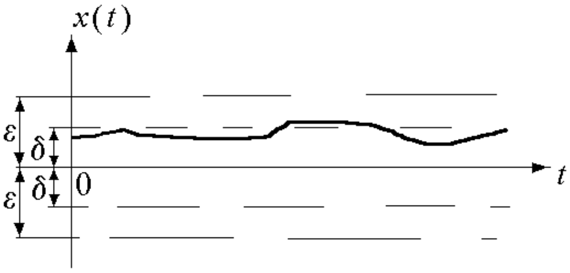
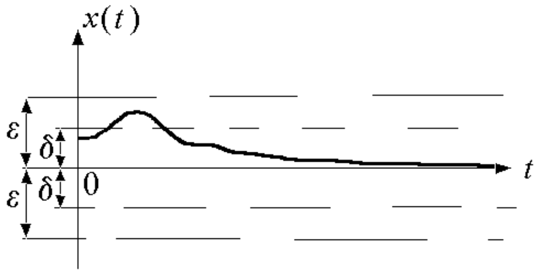


Figure 7.4: Illustration of stability concepts in two dimensions: (a) stable equilibrium state in the sense of Lyapunov, (b) asymptotically stable equilibrium state, (c) unstable equilibrium state.

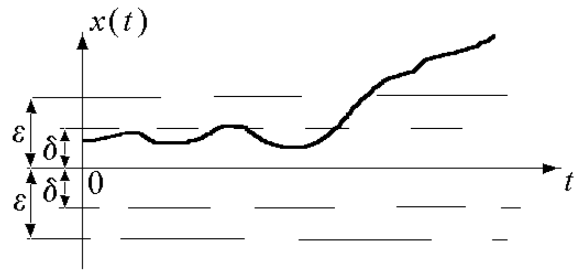
Figure 7.4(a), then given an outer circle of radius ε, there exists an inner circle of radius δ such that trajectories starting within the δ-circle never leave the ε-circle. If the equilibrium state is asymptotically stable, then trajectories tend to the equilibrium state as t tends to ∞. This situation is depicted in Figure 7.4(b). Finally, the equilibrium state is unstable, as in Figure 7.4(c), then there exists an ε-circle such that for every δ-circle there exists a trajectory starting within it that leaves the ε-circle at some later time. A geometric interpretation of the above described scenarios in one dimension is given in Figure 7.5.



(a)



(b)



(c)

Figure 7.5: Illustration of stability concepts in one dimension: (a) stable equilibrium state in the sense of Lyapunov, (b) asymptotically stable equilibrium state, (c) unstable equilibrium state.

7.3 Stability of Continuous-Time LTI Systems

It follows from the discussion in the previous sections that a linear time-invariant (LTI) dynamical system modeled by the equation

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{7.5}$$

is asymptotically stable if and only if, the solution to equation (7.5) decays to zero as $t \to \infty$ for any initial state $\boldsymbol{x}_0$. We can view the vector $\boldsymbol{x}(t) \in \mathbb{R}^n$ as defining the coordinates of a point moving in an n-dimensional state space $\mathbb{R}^n$. In an asymptotically stable system modeled by equation (7.5), this point converges to the origin of $\mathbb{R}^n$. We can argue that if a trajectory is converging to the origin of the state space, then it should be possible to find a family of nested surfaces described by $V(x_1, x_2, \ldots, x_n) = c$, $c \geq 0$, such that monotonically decreasing values of c give surfaces progressively shrinking in on the origin with the limiting surface $V(x_1, x_2, \ldots, x_n) = 0$ being the origin $\boldsymbol{x} = \boldsymbol{0}$. Furthermore, along every trajectory of (7.5), the parameter c should steadily decrease. We now introduce some notation and definitions that we use in the ensuing discussion.

Let $V(\boldsymbol{x}) = V(x_1, x_2, \ldots, x_n) : \mathbb{R}^n \to \mathbb{R}$ be a real-valued function and let $\mathcal{S} \subseteq \mathbb{R}^n$ be a compact region containing the origin $\boldsymbol{x} = \boldsymbol{0}$ in its interior.

Definition 7.8 *We say that the function $V = V(\boldsymbol{x})$ is* positive semi-definite *(p.s.d.) in $\mathcal{S}$, with respect to $\boldsymbol{x} = \mathbf{0}$, if*

1. *V is continuously differentiable, that is, $V \in \mathcal{C}^1$,*
2. *$V(\mathbf{0}) = 0$,*
3. *$V(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathcal{S}$.*

◇

Definition 7.9 *We say that the function $V = V(\boldsymbol{x})$ is* positive definite *(p.d.) in $\mathcal{S}$, with respect to $\boldsymbol{x} = \mathbf{0}$, if*

1. *$V \in \mathcal{C}^1$,*
2. *$V(\mathbf{0}) = 0$,*
3. *$V(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathcal{S} \setminus \{\mathbf{0}\}$.*

◇

The symbol $\mathcal{S} \setminus \{\mathbf{0}\}$ in item 3 above denotes the set $\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{S}, \boldsymbol{x} \neq \mathbf{0}\}$.

Negative semi-definite (n.s.d.) and *negative definite* (n.d.) functions are defined in a similar fashion by reversing inequality signs in item 3 in both definitions above.

There are many functions that satisfy the above definitions. In the stability analysis of dynamic systems one class is particularly useful. This is the positive definite quadratic form, a label attached to a nested family of ellipses and their extensions to higher dimensional spaces—ellipsoids for $n = 3$, and hyper-ellipsoids for $n \geq 4$. Quadratic forms are reviewed in Section A.4. We consider the following quadratic form,

$$V(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{x},$$

where $\boldsymbol{P}$ is a real symmetric $n \times n$ matrix. The time derivative of $V(\boldsymbol{x}(t))$ evaluated on a solution of $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ is

$$\begin{aligned} \frac{dV(\boldsymbol{x}(t))}{dt} &= \dot{V}(\boldsymbol{x}(t)) \\ &= \dot{\boldsymbol{x}}(t)^\top \boldsymbol{P} \boldsymbol{x}(t) + \boldsymbol{x}(t)^\top \boldsymbol{P} \dot{\boldsymbol{x}}(t). \end{aligned}$$

Substituting $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ and $\dot{\boldsymbol{x}}^\top = \boldsymbol{x}^\top \boldsymbol{A}^\top$ into the above equation yields

$$\begin{aligned} \dot{V} &= \boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{A} \boldsymbol{x} \\ &= \boldsymbol{x}^\top (\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A}) \boldsymbol{x}. \end{aligned}$$

Let

$$\boldsymbol{Q} = -(\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}).$$

Then,

$$\begin{aligned}\dot{V} &= \frac{dV(\boldsymbol{x})}{dt} \\ &= -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}. \end{aligned} \tag{7.6}$$

We are ready to state and prove the theorem concerning the stability of the system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$, or equivalently, the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.

Theorem 7.1 (A. M. Lyapunov, 1892) *The system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$, $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, is asymptotically stable if and only if for any given real symmetric positive definite (r.s.p.d.) matrix $\boldsymbol{Q}$ the solution $\boldsymbol{P}$ to the continuous Lyapunov matrix equation*

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q} \tag{7.7}$$

is also (real symmetric) positive definite.

Proof ($\Rightarrow$) We prove necessity, for asymptotic stability, by contradiction. We have $-\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} < 0$ for all $\boldsymbol{x} \neq \boldsymbol{0}$. We assume that $\boldsymbol{A}$ is an asymptotically stable matrix and that for some nonzero $\boldsymbol{x}_0$ we have $\boldsymbol{x}_0^\top \boldsymbol{P}\boldsymbol{x}_0 \leq 0$. From the discussion before the theorem, and in particular from (7.6), it follows that

$$\frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) = -\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t) < 0.$$

The above means that $V(\boldsymbol{x}(t)) = \boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)$ decreases on the trajectories of the system $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t)$ as $t \to \infty$. On the other hand, because $\boldsymbol{A}$ is asymptotically stable, we have

$$\lim_{t\to\infty} \boldsymbol{x}(t) = \boldsymbol{0}.$$

Hence,

$$\lim_{t\to\infty} \boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t) = \lim_{\boldsymbol{x}(t)\to\boldsymbol{0}} \boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t) = 0.$$

Thus, we arrived at a contradiction. This completes the proof of the first part of the theorem that states that asymptotic stability of $\boldsymbol{A}$ implies that for any r.s.p.d matrix $\boldsymbol{Q}$ the solution $\boldsymbol{P}$ to the equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$ is r.s.p.d..

($\Leftarrow$) We prove sufficiency, also by contradiction. We assume that

$$\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} > 0 \quad \text{and} \quad -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} < 0 \quad \text{for all} \quad \boldsymbol{x} \neq \boldsymbol{0},$$

and that the matrix $\boldsymbol{A}$ is not asymptotically stable. This means that the solution to $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t)$, $\boldsymbol{x}_0 \neq \boldsymbol{0}$, which has the form $\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}_0$, is bounded away from the origin $\boldsymbol{x} = \boldsymbol{0}$. Thus, there is a constant $b > 0$ such that

$$\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t) > b > 0 \quad \text{for all } t > t_0.$$

Integrating $\frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) = -\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t)$ yields

$$\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t) = \boldsymbol{x}(t_0)^\top \boldsymbol{P}\boldsymbol{x}(t_0) + \int_{t_0}^{t} (-\boldsymbol{x}(s)^\top \boldsymbol{Q}\boldsymbol{x}(s))ds.$$

Taking into account that $-\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t) < -b < 0 \quad \text{for all } t > t_0$, we obtain for $t > t_0$,

$$\begin{aligned} \boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t) &= \boldsymbol{x}(t_0)^\top \boldsymbol{P}\boldsymbol{x}(t_0) + \int_{t_0}^{t} (-\boldsymbol{x}(s)^\top \boldsymbol{Q}\boldsymbol{x}(s))ds \\ &< \boldsymbol{x}(t_0)^\top \boldsymbol{P}\boldsymbol{x}(t_0) - b(t - t_0). \end{aligned}$$

The above implies that for a sufficiently large t we would have $\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t) < 0$, which contradicts the assumption that $\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t) > 0$. This means that $\boldsymbol{A}$ must be asymptotically stable, and the proof is complete.

□

Example 7.3 Let

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix},$$

and $\boldsymbol{Q} = \boldsymbol{I}_2$. We solve the Lyapunov equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$ for $\boldsymbol{P}$. Assume $\boldsymbol{P}$ to have the form,

$$\boldsymbol{P} = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} = \boldsymbol{P}^\top.$$

Then, we have

$$\begin{aligned} \boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} &= \begin{bmatrix} 0 & -1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \\ &= \begin{bmatrix} -2p_2 & -p_3 + p_1 - 2p_2 \\ -p_3 + p_1 - 2p_2 & 2p_2 - 4p_3 \end{bmatrix} \\ &= -\boldsymbol{Q} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned}$$

Solving the corresponding linear equations yields

$$\boldsymbol{P} = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}.$$

The solution matrix $\boldsymbol{P}$ is positive definite. Hence, the matrix $\boldsymbol{A}$ must have its eigenvalues in the open left-half plane. Indeed, the eigenvalues of $\boldsymbol{A}$ are both located at -1.

Definition 7.10 *A positive definite function $V(\boldsymbol{x})$ whose time derivative evaluated on the solutions of the system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ is negative definite (or negative semi-definite) is called a Lyapunov function for this system.*

◇

Lyapunov's theorem gives us a necessary and sufficient condition for asymptotic stability of linear time-invariant dynamical systems. To check if a given matrix $\boldsymbol{A}$ is asymptotically stable or not, it is enough to take an arbitrary r.s.p.d. matrix $\boldsymbol{Q}$, solve the equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$ for real symmetric $\boldsymbol{P}$ and check if the obtained $\boldsymbol{P}$ is positive definite, or not. If $\boldsymbol{P} > 0$, then $\boldsymbol{A}$ is asymptotically stable. If $\boldsymbol{P}$ is not positive definite, then $\boldsymbol{A}$ cannot be asymptotically stable. Because $\boldsymbol{Q}$ can be an arbitrary r.s.p.d. matrix, $\boldsymbol{Q} = \boldsymbol{I}_n$, the $n \times n$ identity matrix, is a good choice. It is important to note that there would be no use, as the following example shows, in first choosing a matrix $\boldsymbol{P}$ and then calculating $\boldsymbol{Q}$ from the equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$.

Example 7.4 Let

$$\boldsymbol{A} = \begin{bmatrix} -1 & 3 \\ 0 & -1 \end{bmatrix},$$

which is asymptotically stable. Let

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{I}_2 > 0.$$

Then,

$$\boldsymbol{Q} = -(\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}) = \begin{bmatrix} 2 & -3 \\ -3 & 2 \end{bmatrix}$$

which is indefinite. Thus, unless $\boldsymbol{Q}$ turns out to be definite, nothing can be inferred about asymptotic stability from the Lyapunov theorem if we start with $\boldsymbol{P}$ and then calculate $\boldsymbol{Q}$.

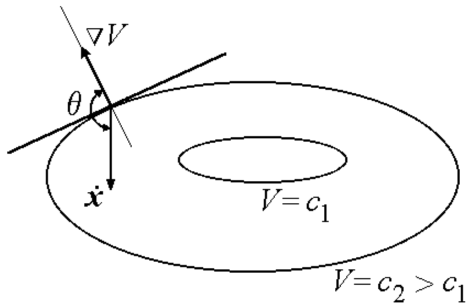


Figure 7.6: Geometric interpretation of the Lyapunov theorem.

We now give a geometric interpretation of the Lyapunov's theorem. First, note that the function

$$\dot{V} = \frac{dV(\boldsymbol{x}(t))}{dt}$$

can be expressed, using the chain rule, as

$$\dot{V} = \nabla V^\top \dot{\boldsymbol{x}} = \|\nabla V\| \, \|\dot{\boldsymbol{x}}\| \cos\theta,$$

where θ is the angle between the vectors ∇V and $\dot{\boldsymbol{x}}$. If $\|\dot{\boldsymbol{x}}\| = 1$, then $\nabla V^\top \dot{\boldsymbol{x}}$ can be viewed as the rate of increase of V at $\boldsymbol{x}$ in the direction $\dot{\boldsymbol{x}}$. Thus, $\dot{V} < 0$ if and only if, $90\,\text{deg} < \theta < 270\,\text{deg}$. In such a case the velocity vector $\dot{\boldsymbol{x}}(t)$ points in the direction of decreasing V. We illustrate the above discussion in Figure 7.6.

Our next theorem is very useful in numerical verification of stability; it is a variant of Theorem 7.1.

Theorem 7.2 *A matrix $\boldsymbol{A}$ is asymptotically stable if and only if for any $\boldsymbol{Q} = \boldsymbol{C}^\top \boldsymbol{C}$, $\boldsymbol{C} \in \mathbb{R}^{p\times n}$, such that the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, the solution $\boldsymbol{P}$ to the Lyapunov matrix equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{C}^\top \boldsymbol{C}$ is positive definite.*

Proof We first prove the necessity condition ($\Rightarrow$) for asymptotic stability. If $\boldsymbol{A}$ is asymptotically stable, that is, $\Re\lambda_i(\boldsymbol{A}) < 0$, then we will show that

$$\boldsymbol{P} = \int_0^\infty e^{\boldsymbol{A}^\top t}\boldsymbol{C}^\top \boldsymbol{C} e^{\boldsymbol{A}t} dt \tag{7.8}$$

is the symmetric positive definite solution to the Lyapunov equation. Indeed, the integrand of (7.8) is a sum of terms of the form

$$t^k e^{\lambda_i t},$$

where λ_i is the i-th eigenvalue of $\boldsymbol{A}$. Since $\Re\lambda_i(\boldsymbol{A}) < 0$, the integral in (7.8) exists because the terms in $\boldsymbol{P}$ do not diverge when $t \to \infty$. Indeed, let $-\alpha_i = \Re\lambda_i$, where $\alpha_i > 0$, then

$$\lim_{t\to\infty} t^k e^{-\alpha_i t} = \lim_{t\to\infty} \frac{t^k}{e^{\alpha_i t}},$$

Differentiating the numerator and denominator of the above k times and using L'Hospital's rule gives

$$\begin{aligned}
\lim_{t\to\infty} \frac{t^k}{e^{\alpha_i t}} &= \lim_{t\to\infty} \frac{k t^{k-1}}{\alpha_i e^{\alpha_i t}} \\
&= \lim_{t\to\infty} \frac{k(k-1) t^{k-2}}{\alpha_i^2 e^{\alpha_i t}} \\
&\ \vdots \\
&= \lim_{t\to\infty} \frac{k!}{\alpha_i^k e^{\alpha_i t}} \\
&= 0.
\end{aligned} \tag{7.9}$$

Thus, by (7.9), the formula for $\boldsymbol{P}$ given by (7.8) is well defined. Next, note that $\boldsymbol{P} = \boldsymbol{P}^\top$. The matrix $\boldsymbol{P}$ is also positive definite because the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable. Substituting the expression for $\boldsymbol{P}$ into the Lyapunov equation and performing manipulations yields

$$\begin{aligned}
\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} &= \int_0^\infty \boldsymbol{A}^\top e^{\boldsymbol{A}^\top t}\boldsymbol{C}^\top \boldsymbol{C} e^{\boldsymbol{A}t} dt + \int_0^\infty e^{\boldsymbol{A}^\top t}\boldsymbol{C}^\top \boldsymbol{C} e^{\boldsymbol{A}t}\boldsymbol{A} dt \\
&= \int_0^\infty \frac{d}{dt}\left(e^{\boldsymbol{A}^\top t}\boldsymbol{C}^\top \boldsymbol{C} e^{\boldsymbol{A}t}\right) dt \\
&= e^{\boldsymbol{A}^\top t}\boldsymbol{C}^\top \boldsymbol{C} e^{\boldsymbol{A}t}\Big|_0^\infty \\
&= -\boldsymbol{C}^\top \boldsymbol{C}.
\end{aligned}$$

The sufficiency part ($\Leftarrow$) can be proven using the arguments of the proof of the sufficiency part of the Lyapunov theorem and noting that the observability of the pair $(\boldsymbol{A}, \boldsymbol{C})$ implies that $\boldsymbol{x}(t)^\top \boldsymbol{C}^\top \boldsymbol{C}\boldsymbol{x}(t)$ is not identically zero along any nonzero solution of $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t)$. □

We now show that if $\boldsymbol{A}$ is asymptotically stable, then the Lyapunov equation (7.7) has a unique solution for every $\boldsymbol{Q}$. Uniqueness can be proven in many ways. We use the Kronecker product in our proof. Using the results of Section A.5, we represent the Lyapunov equation (7.7) as

$$\left(\boldsymbol{I}_n \otimes \boldsymbol{A}^\top + \boldsymbol{A}^\top \otimes \boldsymbol{I}_n\right) \operatorname{vec}(\boldsymbol{P}) = -\operatorname{vec}(\boldsymbol{Q}). \tag{7.10}$$

The condition for a solution $s(\boldsymbol{P})$ of the above equation to be unique is

$$\lambda_i(\boldsymbol{A}^\top) + \lambda_j(\boldsymbol{A}) \neq 0, \quad i, j = 1, 2, \ldots, n.$$

Because $\lambda_i(\boldsymbol{A}^\top) = \lambda_i(\boldsymbol{A})$, and $\boldsymbol{A}$ is asymptotically stable by assumption, the condition $\lambda_i(\boldsymbol{A}^\top) + \lambda_j(\boldsymbol{A}) \neq 0$ is met, and hence the Lyapunov equation has a unique solution.

7.4 Evaluating Quadratic Indices

In this Section, we apply the Lyapunov theory to evaluate the quadratic indices

$$J_r = \int_0^\infty t^r \boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t)dt, \quad r = 0, 1, \ldots$$

subject to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$

where $\boldsymbol{Q} = \boldsymbol{C}^\top \boldsymbol{C}$ such that the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, and the matrix $\boldsymbol{A}$ is asymptotically stable. We reformulate the above problem as follows. Evaluate the quadratic performance indices

$$J_r = \int_0^\infty t^r \boldsymbol{y}(t)^\top \boldsymbol{y}(t)dt, \quad r = 0, 1, \ldots$$

subject to

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t), \end{aligned}$$

where the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable, and the matrix $\boldsymbol{A}$ is asymptotically stable.

To evaluate

$$J_0 = \int_0^\infty \boldsymbol{y}(t)^\top \boldsymbol{y}(t)dt = \int_0^\infty \boldsymbol{x}(t)^\top \boldsymbol{C}^\top \boldsymbol{C}\boldsymbol{x}(t)dt = \int_0^\infty \boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t)dt,$$

we note that

$$\frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) = -\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t), \tag{7.11}$$

where $\boldsymbol{P}$ and $\boldsymbol{Q} = \boldsymbol{C}^\top \boldsymbol{C}$ satisfy the Lyapunov matrix equation, $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$. Hence,

$$J_0 = -\int_0^\infty \frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) dt. \tag{7.12}$$

Integrating both sides of (7.11) with respect to t yields

$$\begin{aligned} J_0 &= -\int_0^\infty \frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) dt \\ &= -\left(\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right)\Big|_0^\infty \\ &= \lim_{t\to\infty}\left(-\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) + \boldsymbol{x}(0)^\top \boldsymbol{P}\boldsymbol{x}(0) \\ &= \boldsymbol{x}(0)^\top \boldsymbol{P}\boldsymbol{x}(0) \end{aligned}$$

because by assumption $\boldsymbol{A}$ is asymptotically stable and therefore, $\lim_{t\to\infty}\boldsymbol{x}(t) = \boldsymbol{0}$ for all $\boldsymbol{x}(0)$. Furthermore, $\boldsymbol{P} = \boldsymbol{P}^\top > 0$, and thus $J_0 > 0$ for all $\boldsymbol{x}(0) \neq \boldsymbol{0}$.

Repeating the above arguments leads to a similar expression for J_1. First observe that

$$\begin{aligned} \frac{d}{dt}\left(t\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) &= \boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t) + t\frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) \\ &= \boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t) - t\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t), \end{aligned}$$

where $\boldsymbol{P}$ satisfies the Lyapunov matrix equation, $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$. Hence,

$$\begin{aligned} J_1 &= \int_0^\infty t\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t)dt \\ &= \int_0^\infty \boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)dt - \int_0^\infty \frac{d}{dt}\left(t\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) dt. \end{aligned} \tag{7.13}$$

Because $\boldsymbol{P} = \boldsymbol{P}^\top > 0$, we can solve

$$\boldsymbol{A}^\top \boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A} = -\boldsymbol{P}$$

for $\boldsymbol{P}_1 = \boldsymbol{P}_1^\top > 0$. The above equation can also be represented as

$$\frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}_1\boldsymbol{x}(t)\right) = -\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t). \tag{7.14}$$

Substituting (7.14) into (7.13) yields

$$\begin{aligned} J_1 &= -\int_0^\infty \frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}_1\boldsymbol{x}(t)\right) dt - \left(t\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right)\Big|_0^\infty \\ &= \lim_{t\to\infty}\left(-\boldsymbol{x}(t)^\top \boldsymbol{P}_1\boldsymbol{x}(t)\right) + \boldsymbol{x}(0)^\top \boldsymbol{P}_1\boldsymbol{x}(0) - 0 + 0 \\ &= \boldsymbol{x}(0)^\top \boldsymbol{P}_1\boldsymbol{x}(0). \end{aligned}$$

Using similar arguments, we can evaluate J_r, $r > 1$, which involves successive solutions of the Lyapunov matrix equations.

In many control problems a dynamical system model is given in terms of a transfer function rather than in terms of a system of differential equations. We discuss a method of converting a given rational function into a system of first order differential equations. After this process is completed, we use the Lyapunov theory to evaluate a performance index of interest. Suppose we are given a rational function of the form

$$E(s) = \frac{b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}, \tag{7.15}$$

where $E(s) = \mathcal{L}(e(t))$ is the Laplace transform of $e(t)$. We show that $e(t)$ can be generated as the solution of the differential equation

$$\frac{d^n e(t)}{dt^n} + a_{n-1}\frac{d^{n-1}e(t)}{dt^{n-1}} + \cdots + a_1\frac{de(t)}{dt} + a_0 e(t) = 0, \tag{7.16}$$

subject to a set of initial conditions that we derive next. We follow the method of Power and Simpson [278, pages 220–221]. We first take the Laplace transform of the differential equation and then rearrange the resulting terms in an appropriate matrix equation. Recall that

$$\mathcal{L}\left(\frac{d^i e(t)}{dt^i}\right) = s^i E(s) - s^{i-1}e(0) - s^{i-2}\frac{de(0)}{dt} - \cdots - \frac{d^{i-1}e(0)}{dt^{i-1}}. \tag{7.17}$$

Note that (7.16) in the s-domain for zero initial conditions takes the form

$$\left(s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0\right) E(s) = 0. \tag{7.18}$$

We apply (7.17) to (7.16), take into account the above relation, and compare coefficients of like powers to obtain

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_{n-1} & 1 \\ a_2 & a_3 & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n-1} & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} e(0) \\ \frac{de(0)}{dt} \\ \vdots \\ \frac{d^{n-2}e(0)}{dt^{n-2}} \\ \frac{d^{n-1}e(0)}{dt^{n-1}} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix}. \tag{7.19}$$

The coefficient matrix in the above equation is nonsingular. Thus, there is a unique solution to this equation, that is, we can uniquely determine the initial conditions. We define the state vector

$$\boldsymbol{x} = \begin{bmatrix} e & \frac{de}{dt} & \cdots & \frac{d^{n-1}e}{dt^{n-1}} \end{bmatrix}^\top,$$

and represent the differential equation (7.16) as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x},$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix}.$$

The vector of initial conditions is

$$\boldsymbol{x}(0) = \begin{bmatrix} e(0) \\ \frac{de(0)}{dt} \\ \vdots \\ \frac{d^{n-2}e(0)}{dt^{n-2}} \\ \frac{d^{n-1}e(0)}{dt^{n-1}} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n-1} & 1 \\ a_2 & a_3 & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n-1} & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix}.$$

Note that $e = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \boldsymbol{x}$.

The above procedure for converting a rational function description of a process into its equivalent state-space representation shows a relation between the zeros of the given rational function and the initial conditions of the associated state-space realization.

We use the above method of constructing a state-space realization of a given rational function in the following example that involves evaluation of a quadratic index.

Example 7.5 Consider a block diagram of a space telescope control system model shown in Figure 7.7. This system comes from Dorf and Bishop [85, page 333]. The values of the parameters are $K_1 = 0.5$, and $K_1K_2K_p = 2.5$. We use the Lyapunov theory to select the gain K_3 that minimizes the effect of the disturbance $u(t) = \mathcal{L}^{-1}(U(s))$ in the sense of the integral of the square error (ISE) criterion. In other words, we wish to find K_3 that leads to minimization of the performance index

$$J_0 = \int_0^\infty c(t)^2 dt,$$

where $c(t) = \mathcal{L}^{-1}(C(s))$. We assume a unit step disturbance and $R(s) = 0$.

We first find a rational function that models $C(s)$. From the block diagram shown in Figure 7.7, we obtain

$$C(s) = U(s) - \frac{K_2}{s}\frac{K_1}{s + K_1K_3}K_pC(s).$$

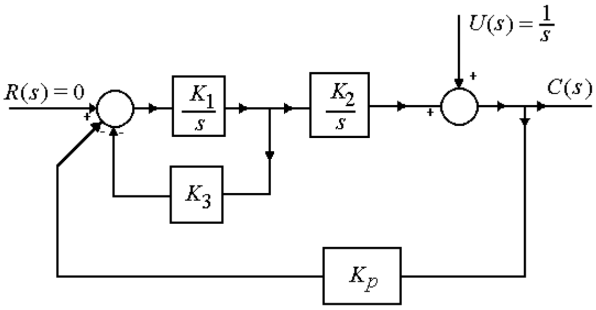


Figure 7.7: Block diagram of the optimized control system model of Example 7.5.

Hence

$$C(s) = \frac{s + 0.5K_3}{s^2 + 0.5K_3 s + 2.5}.$$

The next step is to determine the state-space representation corresponding to $C(s)$. We use the method given before this example. We first form the differential equation for $c(t)$,

$$\frac{d^2c(t)}{dt^2} + 0.5K_3\frac{dc(t)}{dt} + 2.5c(t) = 0$$

with the equation for the initial conditions

$$\begin{bmatrix} 0.5K_3 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} c(0) \\ \frac{dc(0)}{dt} \end{bmatrix} = \begin{bmatrix} 0.5K_3 \\ 1 \end{bmatrix}$$

obtained by applying (7.19). Solving the above equation yields

$$\begin{bmatrix} c(0) \\ \frac{dc(0)}{dt} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Let $x_1 = c$, $x_2 = \frac{dc}{dt}$. The differential equation in vector-matrix form corresponding to $C(s)$ is

$$\left.\begin{aligned} \dot{\boldsymbol{x}}(t) &= \begin{bmatrix} 0 & 1 \\ -2.5 & -0.5K_3 \end{bmatrix}\boldsymbol{x}(t), \quad \boldsymbol{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ c(t) &= \boldsymbol{c}\boldsymbol{x}(t), \end{aligned}\right\}$$

where $\boldsymbol{c} = \begin{bmatrix} 1 & 0 \end{bmatrix}$. We evaluate the performance index J_0,

$$J_0 = \int_0^\infty c(t)^2 dt = \int_0^\infty \boldsymbol{x}(t)^\top \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}(t) dt = \int_0^\infty \boldsymbol{x}(t)^\top \boldsymbol{c}^\top \boldsymbol{c}\, \boldsymbol{x}(t) dt,$$

The pair $(\boldsymbol{A}, \boldsymbol{c})$ is observable. Therefore, we can solve the Lyapunov equation

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{c}^\top \boldsymbol{c}$$

for $\boldsymbol{P}$ to obtain

$$\boldsymbol{P} = \begin{bmatrix} \frac{K_3}{10} + \frac{1}{K_3} & \frac{1}{5} \\ \frac{1}{5} & \frac{2}{5K_3} \end{bmatrix}.$$

Hence

$$J_0 = \boldsymbol{x}(0)^\top \boldsymbol{P} \boldsymbol{x}(0) = \frac{K_3}{10} + \frac{1}{K_3}.$$

Differentiating J_0 and equating the result to zero gives

$$\frac{dJ_0}{dK_3} = \frac{1}{10} - \frac{1}{K_3^2} = 0.$$

Selecting

$$K_3 = \sqrt{10},$$

yields a positive definite $\boldsymbol{P}$. Note that because

$$\left.\frac{d^2 J_0}{dK_3^2}\right|_{K_3=\sqrt{10}} = \left.\frac{2}{K_3^3}\right|_{K_3=\sqrt{10}} > 0,$$

that is, the second order sufficiency condition for the minimum is satisfied, we conclude that $K_3 = \sqrt{10} \approx 3.16$ minimizes J_0.

Next, we wish to determine the gain K_3 that leads to minimization of the effect of the disturbance $u(t)$ in the sense of the integral of time multiplied by the squared error (ITSE) criterion. That is, we wish to select K_3 to minimize the performance index

$$J_1 = \int_0^\infty t c(t)^2 dt.$$

We first solve the Lyapunov equation,

$$\boldsymbol{A}^\top \boldsymbol{P}_1 + \boldsymbol{P}_1 \boldsymbol{A} = -\boldsymbol{P},$$

to get

$$\boldsymbol{P}_1 = \begin{bmatrix} \frac{2}{K_3^2} + \frac{K_3^2}{100} & \frac{1}{5}\left(\frac{1}{K_3} + \frac{K_3}{10}\right) \\ \frac{1}{5}\left(\frac{1}{K_3} + \frac{K_3}{10}\right) & \frac{4}{5K_3^2} + \frac{1}{25} \end{bmatrix}.$$

Hence

$$J_1 = \boldsymbol{x}(0)^\top \boldsymbol{P}_1 \boldsymbol{x}(0) = \frac{2}{K_3^2} + \frac{K_3^2}{100}.$$

Differentiating J_1 and equating the result to zero gives

$$\frac{dJ_1}{dK_3} = -\frac{4}{K_3^3} + \frac{K_3}{50} = 0.$$

We select $K_3 = \sqrt[4]{200}$ to keep $\boldsymbol{P}_1$ positive definite. For this choice of K_3, the second order sufficiency condition for the minimum of J_1 is satisfied. Thus, $K_3 \approx 3.76$ minimizes J_1.

7.5 Discrete-Time Lyapunov Equation

In this Section, we use the Lyapunov approach to the stability analysis of the discrete linear time-invariant (LTI) systems modeled by

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k], \quad \boldsymbol{x}[k_0] = \boldsymbol{x}_0. \tag{7.20}$$

By analogy with the continuous-time case, consider a quadratic positive definite form,

$$V(\boldsymbol{x}[k]) = \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k].$$

We evaluate the first difference of V, defined $\Delta V = V(\boldsymbol{x}[k+1]) - V(\boldsymbol{x}[k])$, on the trajectories of (7.20) to obtain

$$\begin{aligned} \Delta V(\boldsymbol{x}[k]) &= V(\boldsymbol{x}[k+1]) - V(\boldsymbol{x}[k]) \\ &= \boldsymbol{x}[k+1]^\top \boldsymbol{P}\boldsymbol{x}[k+1] - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] \\ &= \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] \\ &= \boldsymbol{x}[k]^\top \left(\boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\right) \boldsymbol{x}[k], \end{aligned} \tag{7.21}$$

where we replaced $\boldsymbol{x}[k+1]$ in the above with $\boldsymbol{A}\boldsymbol{x}[k]$. For the solutions of (7.20) to decrease the "energy function" V, at each $k > k_0$, it is necessary and sufficient that $\Delta V(\boldsymbol{x}[k]) < 0$, or equivalently, that for some positive definite $\boldsymbol{Q}$,

$$\Delta V(\boldsymbol{x}[k]) = -\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k]. \tag{7.22}$$

Combining (7.20) and (7.22), we obtain the so called *discrete Lyapunov matrix equation*,

$$\boxed{\boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P} = -\boldsymbol{Q}} \tag{7.23}$$

Theorem 7.3 *The matrix $\boldsymbol{A}$ has its eigenvalues in the open unit disk if and only if for any positive definite $\boldsymbol{Q}$ the solution $\boldsymbol{P}$ to (7.23) is positive definite.*

Proof ($\Rightarrow$) We can use the Kronecker product, and in particular relation (A.62) to represent the discrete-time Lyapunov equation as

$$\left(\boldsymbol{I}_{n^2} - \boldsymbol{A}^\top \otimes \boldsymbol{A}^\top\right) \operatorname{vec}(\boldsymbol{P}) = \operatorname{vec}(\boldsymbol{Q}), \tag{7.24}$$

where $\otimes$ denotes the Kronecker product and $\operatorname{vec}(\cdot)$ is the stacking operator. It is easy to see that the above equation has a unique solution if and only if

$$1 - \lambda_i \lambda_j \neq 0, \quad \text{for all} \quad i, j = 1, 2, \ldots, n,$$

where $\{\lambda_i\}$ is the set of eigenvalues of $\boldsymbol{A}$. By assumption, $|\lambda_i| < 1$, which is a sufficient condition for (7.24) and hence (7.23) to have a unique solution. The solution $\boldsymbol{P}$ can be expressed as

$$\boldsymbol{P} = \sum_{k=0}^{\infty} \left(\boldsymbol{A}^\top\right)^k \boldsymbol{Q} \boldsymbol{A}^k. \tag{7.25}$$

The above expression is well defined because all eigenvalues of $\boldsymbol{A}$ have magnitude (strictly) less than unity. Substituting (7.25) into (7.23) gives

$$\begin{aligned}
\boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{A} - \boldsymbol{P} &= \boldsymbol{A}^\top \left(\sum_{k=0}^{\infty} \left(\boldsymbol{A}^\top\right)^k \boldsymbol{Q} \boldsymbol{A}^k \right) \boldsymbol{A} - \sum_{k=0}^{\infty} \left(\boldsymbol{A}^\top\right)^k \boldsymbol{Q} \boldsymbol{A}^k \\
&= \sum_{k=1}^{\infty} \left(\boldsymbol{A}^\top\right)^k \boldsymbol{Q} \boldsymbol{A}^k - \sum_{k=1}^{\infty} \left(\boldsymbol{A}^\top\right)^k \boldsymbol{Q} \boldsymbol{A}^k - \boldsymbol{Q} \\
&= -\boldsymbol{Q}.
\end{aligned}$$

Thus, indeed (7.25) satisfies the discrete-time Lyapunov equation. Note that if $\boldsymbol{Q}$ is symmetric so is $\boldsymbol{P}$. If $\boldsymbol{Q}$ is positive definite, so is $\boldsymbol{P}$.

($\Leftarrow$) Let λ_i be an eigenvalue of $\boldsymbol{A}$ and $\boldsymbol{v}_i$ be a corresponding eigenvector, that is, $\boldsymbol{A}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$. Observe that λ_i and $\boldsymbol{v}_i$ may be complex. We denote by $\boldsymbol{v}_i^*$ the complex conjugate transpose of $\boldsymbol{v}_i$. Note that $\boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i$ is a scalar. We show that its complex conjugate equals itself, which means that $\boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i$ is real. Indeed,

$$\overline{\boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i} = \boldsymbol{v}_i^* \boldsymbol{P}^* \boldsymbol{v}_i = \boldsymbol{v}_i^* \boldsymbol{P}^\top \boldsymbol{v}_i = \boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i,$$

where we used the fact that the complex conjugate transpose of a real matrix reduces to its transpose. Thus, $\boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i$ is real for any complex $\boldsymbol{v}_i$.

Premultiplying now (7.23) by $\boldsymbol{v}_i^*$ and postmultiplying by $\boldsymbol{v}_i$ gives

$$\boldsymbol{v}_i^* \left(\boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{A} - \boldsymbol{P} \right) \boldsymbol{v}_i = -\boldsymbol{v}_i^* \boldsymbol{Q} \boldsymbol{v}_i. \tag{7.26}$$

The left-hand side of the above equality evaluates to

$$\begin{aligned} \boldsymbol{v}_i^* \boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{A} \boldsymbol{v}_i - \boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i &= \bar{\lambda}_i \boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i \lambda_i - \boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i \\ &= \left(|\lambda_i|^2 - 1 \right) \boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i. \end{aligned}$$

Comparing the above with the right-hand side of (7.26) yields

$$\left(1 - |\lambda_i|^2 \right) \boldsymbol{v}_i^* \boldsymbol{P} \boldsymbol{v}_i = \boldsymbol{v}_i^* \boldsymbol{Q} \boldsymbol{v}_i > 0$$

because $\boldsymbol{Q}$ is positive definite. Since $\boldsymbol{P}$ is positive definite, we have to have $|\lambda_i| < 1$. The proof is complete.

□

We now concern ourselves with evaluating performance indices on the trajectories of (7.20). Specifically, we wish to evaluate

$$K_r = \sum_{k=0}^{\infty} k^r \boldsymbol{x}[k]^\top \boldsymbol{Q} \boldsymbol{x}[k], \quad r = 0, 1, \tag{7.27}$$

where $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$ subject to (7.20). We assume that the matrix $\boldsymbol{A}$ has its eigenvalues in the open unit disk, that is, the matrix $\boldsymbol{A}$ is *convergent*. We first consider the case when $r = 0$. For this case, K_r becomes

$$K_0 = \sum_{k=0}^{\infty} \boldsymbol{x}[k]^\top \boldsymbol{Q} \boldsymbol{x}[k]. \tag{7.28}$$

Because the matrix $\boldsymbol{A}$ is convergent, the solution $\boldsymbol{P}$ to (7.23) is positive definite. Using (7.21), (7.22) and (7.20), we write

$$\begin{aligned} \boldsymbol{x}[k]^\top \boldsymbol{Q} \boldsymbol{x}[k] &= -\boldsymbol{x}[k]^\top \left(\boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{A} - \boldsymbol{P} \right) \boldsymbol{x}[k] \\ &= \boldsymbol{x}[k]^\top \boldsymbol{P} \boldsymbol{x}[k] - \boldsymbol{x}[k+1]^\top \boldsymbol{P} \boldsymbol{x}[k+1] \end{aligned} \tag{7.29}$$

Summing both sides of the above form $k = 0$ to $k = \infty$ and using the fact that $\boldsymbol{x}[\infty] = \boldsymbol{0}$ because the matrix $\boldsymbol{A}$ is convergent, we obtain

$$K_0 = \boldsymbol{x}[0]^\top \boldsymbol{P} \boldsymbol{x}[0]. \tag{7.30}$$

We now evaluate

$$K_1 = \sum_{k=0}^{\infty} k\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k],$$

subject to (7.20). To proceed, we multiply both sides of (7.29) by k to get

$$k\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] = k\boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] - k\boldsymbol{x}[k+1]^\top \boldsymbol{P}\boldsymbol{x}[k+1].$$

Summing the above from $k = 0$ to $k = \infty$ gives

$$\begin{aligned} K_1 &= \boldsymbol{x}[1]^\top \boldsymbol{P}\boldsymbol{x}[1] - \boldsymbol{x}[2]^\top \boldsymbol{P}\boldsymbol{x}[2] + 2\boldsymbol{x}[2]^\top \boldsymbol{P}\boldsymbol{x}[2] - 2\boldsymbol{x}[3]^\top \boldsymbol{P}\boldsymbol{x}[3] + \cdots \\ &= \sum_{k=0}^{\infty} \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] - \boldsymbol{x}[0]^\top \boldsymbol{P}\boldsymbol{x}[0]. \end{aligned} \tag{7.31}$$

Using the method we employed to find the value of K_0, we evaluate the first term on the right-hand side of (7.31) to get

$$\sum_{k=0}^{\infty} \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] = \boldsymbol{x}[0]^\top \boldsymbol{P}_1\boldsymbol{x}[0],$$

where $\boldsymbol{P}_1$ satisfies

$$\boldsymbol{A}^\top \boldsymbol{P}_1 \boldsymbol{A} - \boldsymbol{P}_1 = -\boldsymbol{P}.$$

Taking the above into account, we represent (7.31) as

$$K_1 = \boldsymbol{x}[0]^\top \left(\boldsymbol{P}_1 - \boldsymbol{P}\right) \boldsymbol{x}[0].$$

Proceeding in a similar manner, we can find expressions for K_r, where $r > 1$.

7.6 Constructing Robust Linear Controllers

In this section, we use the Lyapunov theorem to construct robust state-feedback controllers. One of the basic issues in the control of dynamical systems is the effect of uncertainties, or neglected nonlinearities, on the stability of the closed-loop system. A controller is viewed robust if it maintains stability for all uncertainties in an expected range. We consider a class of dynamical systems modeled by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{f}(t, \boldsymbol{x}(t)) + \boldsymbol{B}\boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{7.32}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$, and the functions $\boldsymbol{h}$ and $\boldsymbol{f}$ model uncertainties, or nonlinearities, in the system. We refer to $\boldsymbol{h}$ as the *matched uncertainty* because it affects the system dynamics via the input matrix $\boldsymbol{B}$ in the same fashion as the input $\boldsymbol{u}$ does. In other words, the uncertainty $\boldsymbol{h}$ matches the system input $\boldsymbol{u}$. The vector $\boldsymbol{f}$ models the *unmatched uncertainty*. We assume that the uncertain elements $\boldsymbol{h}$ and $\boldsymbol{f}$ satisfy the following norm bounds:

1. $\|\boldsymbol{h}(t, \boldsymbol{x}, \boldsymbol{u})\| \leq \gamma_h \|\boldsymbol{u}\| + \alpha_h \|\boldsymbol{x}\|$,
2. $\|\boldsymbol{f}(t, \boldsymbol{x})\| \leq \alpha_f \|\boldsymbol{x}\|$,

where γ_h, α_h, and α_f are known nonnegative constants. We further assume that the matrix $\boldsymbol{A}$ is asymptotically stable. If this is not the case, we apply a preliminary state-feedback controller $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$ such that $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}$ is asymptotically stable. Such a feedback exists provided that the pair $(\boldsymbol{A}, \boldsymbol{B})$ is stabilizable. Our goal is to construct a linear state-feedback controller that would make the closed-loop system asymptotically stable for arbitrary $\boldsymbol{f}$ and $\boldsymbol{h}$ that satisfy the above norm bounds.

Theorem 7.4 *Suppose that $\boldsymbol{A}$ is asymptotically stable, and $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ is the solution to the Lyapunov matrix equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{Q}$ for some $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$. Suppose also that*

$$\alpha_f < \frac{\lambda_{\min}(\boldsymbol{Q})}{\lambda_{\max}(\boldsymbol{P})} \quad \text{and} \quad \gamma_h < 1.$$

Then, the state-feedback controller $\boldsymbol{u} = -\gamma \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{x}$, where

$$\gamma > \frac{\alpha_h^2}{4\left(\lambda_{\min}(\boldsymbol{Q}) - \alpha_f \lambda_{\max}(\boldsymbol{P})\right)(1 - \gamma_h)}$$

stabilizes the uncertain system model (7.32) for arbitrary $\boldsymbol{f}$ and $\boldsymbol{h}$ that satisfy the norm bounds.

Proof The time derivative of the positive definite function $V = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}$ evaluated on a generic trajectory of the closed-loop system is

$$\begin{aligned}\dot{V} &= 2\boldsymbol{x}^\top \boldsymbol{P}\dot{\boldsymbol{x}} \\ &= -2\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - 2\gamma \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{x} + 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{f} + 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{h}.\end{aligned}$$

Our goal is to determine $\tilde{\gamma} > 0$ such that if $\gamma > \tilde{\gamma}$, then $\dot{V} < 0$, which in turn implies the asymptotic stability of the closed-loop system. To proceed further, recall that if $\boldsymbol{Q}$ is a symmetric matrix, then

$$\lambda_{\min}(\boldsymbol{Q})\|\boldsymbol{x}\|^2 \leq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} \leq \lambda_{\max}(\boldsymbol{Q})\|\boldsymbol{x}\|^2,$$

and therefore

$$-\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} \le -\lambda_{\min}(\boldsymbol{Q})\|\boldsymbol{x}\|^2.$$

Furthermore,

$$\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\right)^\top = \|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\|^2.$$

For a symmetric positive definite matrix $\boldsymbol{P}$, its induced 2-norm is

$$\|\boldsymbol{P}\| = \lambda_{\max}(\boldsymbol{P}).$$

Taking the above relations into account, using the norm bounds on the uncertain elements, and performing some manipulations yields

$$\begin{aligned} \dot{V} \;\le\; & -2\lambda_{\min}(\boldsymbol{Q})\|\boldsymbol{x}\|^2 - 2\gamma\|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\|^2 + 2\alpha_f\lambda_{\max}(\boldsymbol{P})\|\boldsymbol{x}\|^2 \\ & +2\|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\|\left(\alpha_h\|\boldsymbol{x}\| + \gamma_h\gamma\|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\|\right). \end{aligned} \tag{7.33}$$

Let

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix} \lambda_{\min}(\boldsymbol{Q}) - \alpha_f\lambda_{\max}(\boldsymbol{P}) & -\frac{\alpha_h}{2} \\ -\frac{\alpha_h}{2} & \gamma(1-\gamma_h) \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{7.34}$$

Then, we can represent (7.33) as

$$\dot{V} \le -2\begin{bmatrix} \|\boldsymbol{x}\| & \|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\| \end{bmatrix} \tilde{\boldsymbol{Q}} \begin{bmatrix} \|\boldsymbol{x}\| \\ \|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\| \end{bmatrix}. \tag{7.35}$$

By Theorem A.6 on page 969, for $\dot{V}$ to be negative it is enough that the leading principal minors of the matrix $\tilde{\boldsymbol{Q}}$ given by (7.34) be positive. The first-order leading principal minor is positive by assumption. For the second-order leading principal minor to be positive it is sufficient that $\gamma_h < 1$ and

$$\gamma > \frac{\alpha_h^2}{4\left(\lambda_{\min}(\boldsymbol{Q}) - \alpha_f\lambda_{\max}(\boldsymbol{P})\right)(1-\gamma_h)}.$$

The proof is now complete.

□

Note that if $\gamma_h = 0$, then the closed-loop system is asymptotically stable if

$$\alpha_f < \frac{\lambda_{\min}(\boldsymbol{Q})}{\lambda_{\max}(\boldsymbol{P})} \quad \text{and} \quad \gamma > \frac{\alpha_h^2}{4\left(\lambda_{\min}(\boldsymbol{Q}) - \alpha_f\lambda_{\max}(\boldsymbol{P})\right)}.$$

Example 7.6 Consider a dynamic system model

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}(u + h(\boldsymbol{x}, u)) \\ &= \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} (u + h(\boldsymbol{x}, u)).\end{aligned}$$

Let $\boldsymbol{P}$ be the solution of the matrix Lyapunov equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{I}_2$. We design a linear state-feedback controller $u = -\gamma \boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}$ so that the origin of the closed-loop system is uniformly asymptotically stable for any $h(\boldsymbol{x}, u)$ such that

$$|h(\boldsymbol{x}, u)| \leq 4\|\boldsymbol{x}\| + \frac{1}{2}|u|. \tag{7.36}$$

In particular, we will find a bound $\tilde{\gamma} > 0$ so that for any $\gamma > \tilde{\gamma}$ the closed-loop system is uniformly asymptotically stable in the face of uncertainties satisfying the norm bound (7.36).

Solving the Lyapunov equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{I}_2$ yields

$$\boldsymbol{P} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix}.$$

Hence,

$$u = -\gamma \boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x} = -\gamma \left(\frac{1}{2} x_1 + x_2 \right).$$

We obtain a bound on γ by evaluating $\frac{d}{dt}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}\right)$ on the trajectories of the closed-loop system. First, we evaluate

$$\begin{aligned}\frac{d}{dt}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}\right) &= 2\boldsymbol{x}^\top \boldsymbol{P}\dot{\boldsymbol{x}} \\ &= 2\boldsymbol{x}^\top \boldsymbol{P}\left(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}(u + h(\boldsymbol{x}, u))\right) \\ &= -2\boldsymbol{x}^\top \boldsymbol{x} - 2\gamma \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x} + 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}h.\end{aligned}$$

Taking the norms and performing manipulations yields

$$\begin{aligned}\frac{d}{dt}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}\right) &\leq -2\boldsymbol{x}^\top \boldsymbol{x} - 2\gamma \left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right|^2 + 2\left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right| \left(4\|\boldsymbol{x}\| + \frac{1}{2}\gamma \left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right| \right) \\ &= -2\|\boldsymbol{x}\|^2 - \gamma \left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right|^2 + 8\left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right| \|\boldsymbol{x}\| \\ &= -2 \begin{bmatrix} \|\boldsymbol{x}\| & \left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right| \end{bmatrix} \begin{bmatrix} 1 & -2 \\ -2 & \gamma/2 \end{bmatrix} \begin{bmatrix} \|\boldsymbol{x}\| \\ \left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right| \end{bmatrix}.\end{aligned}$$

For $\frac{d}{dt}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}\right)$ to be negative definite it is enough that the matrix

$$\begin{bmatrix} 1 & -2 \\ -2 & \gamma/2 \end{bmatrix}$$

be positive definite. This is the case whenever

$$\gamma > \tilde{\gamma} = 8.$$

7.7 Hurwitz and Routh Stability Criteria

In this section, we present two tests for determining if the zeros of an n-th degree polynomial are located in the open left-half complex plane. In deriving these tests, we use the Lyapunov matrix equation given by (7.7). We consider the n-th degree polynomial

$$p(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0 \tag{7.37}$$

with real coefficients. Associated with the polynomial $p(s)$ is the $n \times n$ real Hurwitz matrix,

$$\boldsymbol{H} = \begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} & a_{n-7} & \cdots \\ a_n & a_{n-2} & a_{n-4} & a_{n-6} & \cdots \\ 0 & a_{n-1} & a_{n-3} & a_{n-5} & \cdots \\ 0 & a_n & a_{n-2} & a_{n-4} & \cdots \\ \vdots & & & \vdots & \ddots \end{bmatrix}. \tag{7.38}$$

In our further discussion we assume, without loss of generality, that $a_n = 1$. The leading principal minors of the Hurwitz matrix are

$$H_1 = a_{n-1},\ H_2 = \det \begin{bmatrix} a_{n-1} & a_{n-3} \\ 1 & a_{n-2} \end{bmatrix},\ H_3 = \det \begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} \\ 1 & a_{n-2} & a_{n-4} \\ 0 & a_{n-1} & a_{n-3} \end{bmatrix}, \ldots, H_n = \det \boldsymbol{H}. \tag{7.39}$$

We next define

$$b_1 = H_1, \quad b_2 = \frac{H_2}{H_1}, \quad b_3 = \frac{H_3}{H_2 H_1},$$

and

$$b_k = \frac{H_{k-3} H_k}{H_{k-2} H_{k-1}} \quad \text{for} \quad k = 4, 5, \ldots, n.$$

Thus, for example, $b_4 = \frac{H_1 H_4}{H_2 H_3}$. Let

$$\boldsymbol{B} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -b_n & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -b_{n-1} & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -b_3 & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -b_2 & -b_1 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

For example, for $n = 4$, we have

$$\boldsymbol{B} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -b_4 & 0 & 1 & 0 \\ 0 & -b_3 & 0 & 1 \\ 0 & 0 & -b_2 & -b_1 \end{bmatrix}.$$

Let

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Lemma 7.1 *The matrices* $\boldsymbol{B}$ *and* $\boldsymbol{A}$ *are similar, that is, there is a nonsingular matrix* $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ *such that*

$$\boldsymbol{B} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}. \tag{7.40}$$

Proof The similarity matrix is lower triangular and has the form

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ t_{31} & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & t_{42} & 0 & 1 & 0 & 0 & \cdots & 0 \\ t_{51} & 0 & t_{53} & 0 & 1 & 0 & \cdots & 0 \\ 0 & t_{62} & 0 & t_{64} & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & & \vdots \end{bmatrix}. \tag{7.41}$$

To determine the nonfixed coefficients t_{ij} of the matrix $\boldsymbol{T}$, we represent (7.40) as

$$\boldsymbol{T}\boldsymbol{B} = \boldsymbol{A}\boldsymbol{T}, \tag{7.42}$$

and compare both sides of (7.42) to determine the coefficients t_{ij}. We obtain, for example, $t_{31} = -b_n$, $t_{42} = -b_n - b_{n-1}$, etc.

□

Let $\boldsymbol{P}$ be an $n \times n$ diagonal matrix defined as

$$\boldsymbol{P} = \text{diag}\begin{bmatrix} b_1 b_2 \times \cdots \times b_n & b_1 b_2 \times \cdots \times b_{n-1} & \cdots & b_1 b_2 & b_1 \end{bmatrix}, \tag{7.43}$$

and $\boldsymbol{Q}$ be an $n \times n$ diagonal matrix defined as

$$\boldsymbol{Q} = \text{diag}\begin{bmatrix} 0 & 0 & \cdots & 0 & 2b_1^2 \end{bmatrix}.$$

We represent the matrix $\boldsymbol{Q}$ as $\boldsymbol{Q} = \boldsymbol{c}^\top \boldsymbol{c}$, where

$$\boldsymbol{c} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \sqrt{2} b_1 \end{bmatrix}.$$

Note that the pair $(\boldsymbol{B}, \boldsymbol{c})$ is observable. The matrices $\boldsymbol{B}$, $\boldsymbol{P}$, and $\boldsymbol{Q}$ satisfy the Lyapunov matrix equation,

$$\boldsymbol{B}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{B} = -\boldsymbol{Q}. \tag{7.44}$$

By Theorem 7.2, on page 325, applied to (7.44), the matrix $\boldsymbol{B}$ is asymptotically stable if and only if the symmetric matrix $\boldsymbol{P}$ is positive definite. By Theorem A.6, the matrix $\boldsymbol{P}$ is positive definite if and only if its leading principal minors are positive. Using the above results we can prove the following theorem of Hurwitz:

Theorem 7.5 (Hurwitz) *The real polynomial $p(s)$ of degree n with positive coefficients a_i, has its zeros in the open left-half plane if and only if the leading principal minors of the Hurwitz matrix are positive.*

Proof The polynomial $p(s)$ has its zeros in the open left plane if and only if the matrix $\boldsymbol{B}$ is asymptotically stable. This in turn is true if and only if the diagonal elements of $\boldsymbol{P}$, given by (7.43), are positive. We have

$$b_1 = H_1, \quad b_1 b_2 = H_2,$$

and for $k = 3, \ldots, n$,

$$b_1 b_2 \times \cdots \times b_k = \frac{H_k}{H_{k-2}}.$$

Thus, the diagonal elements of $\boldsymbol{P}$ are all positive, if and only if the leading principal minors of the Hurwitz matrix are positive, which completes the proof.

□

We now prove the Routh necessary and sufficient condition for the real polynomial (7.37) to have its zeros in the open left-half plane. We first use the polynomial coefficients to form the following $(n+1)$-row Routh array:

$$\begin{array}{cc|ccccc} s^n & \text{row} & a_n & a_{n-2} & a_{n-4} & a_{n-6} & \cdots \\ s^{n-1} & \text{row} & a_{n-1} & a_{n-3} & a_{n-5} & a_{n-7} & \cdots \\ s^{n-2} & \text{row} & r_{21} & r_{22} & r_{23} & r_{24} & \cdots \\ s^{n-3} & \text{row} & r_{31} & r_{32} & r_{33} & r_{34} & \cdots \\ \vdots & & \vdots & & & \vdots & \\ s^0 & \text{row} & r_{n1} & & & & \end{array} \tag{7.45}$$

The elements that comprise the s^n and s^{n-1} rows are appropriately ordered coefficients of $p(s)$. The elements of the s^{n-2} row are formed from its preceding two rows as follows,

$$\begin{aligned} r_{21} &= -\frac{1}{a_{n-1}} \det \begin{bmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{bmatrix} = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}} \\ r_{22} &= -\frac{1}{a_{n-1}} \det \begin{bmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{bmatrix} = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}} \\ r_{23} &= -\frac{1}{a_{n-1}} \det \begin{bmatrix} a_n & a_{n-6} \\ a_{n-1} & a_{n-7} \end{bmatrix} = \frac{a_{n-1}a_{n-6} - a_n a_{n-7}}{a_{n-1}} \\ \vdots & \qquad \vdots \end{aligned}$$

The elements of the s^{n-3} row of the Routh array are generated using its preceding two rows,

$$\begin{aligned} r_{31} &= -\frac{1}{r_{21}} \det \begin{bmatrix} a_{n-1} & a_{n-3} \\ r_{21} & r_{22} \end{bmatrix} = \frac{r_{21}a_{n-3} - r_{22}a_{n-1}}{r_{21}} \\ r_{32} &= -\frac{1}{r_{21}} \det \begin{bmatrix} a_{n-1} & a_{n-5} \\ r_{21} & r_{23} \end{bmatrix} = \frac{r_{21}a_{n-5} - r_{23}a_{n-1}}{r_{21}} \\ r_{33} &= -\frac{1}{r_{21}} \det \begin{bmatrix} a_{n-1} & a_{n-7} \\ r_{21} & r_{24} \end{bmatrix} = \frac{r_{21}a_{n-7} - r_{24}a_{n-1}}{r_{21}} \\ \vdots & \qquad \vdots \end{aligned}$$

Subsequent rows of the array are formed from their preceding rows in an analogous fashion. Undefined positions can be filled with zeros. The last two rows of the array will have one term each. We now state the stability test that in some texts (for example, in [283] or [353]) is also called the Routh-Hurwitz criterion.

Theorem 7.6 (Routh) *The zeros of the polynomial* $p(s) = a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0$, *whose coefficients are all non-zero and have the same sign, are all in the open left-half plane if and only if the elements of the first column of the associated Routh array all have the same sign and are nonzero.*

Proof Without loss of generality, we assume that the polynomial coefficients are all positive. If they are negative, we can multiply both sides of the equation $p(s) = 0$ by -1 thus making the coefficients positive. We then perform elementary row operations on the associated Hurwitz matrix, $\boldsymbol{H}$. These operations are equivalent to premultiplying the Hurwitz matrix by lower triangular matrices with ones on the diagonal. For example,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ -a_n/a_{n-1} & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} & a_{n-7} \\ a_n & a_{n-2} & a_{n-4} & a_{n-6} \\ 0 & a_{n-1} & a_{n-3} & a_{n-5} \\ 0 & a_n & a_{n-2} & a_{n-4} \\ \vdots & & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} & a_{n-7} \\ 0 & r_{21} & r_{22} & r_{23} \\ 0 & a_{n-1} & a_{n-3} & a_{n-5} \\ 0 & a_n & a_{n-2} & a_{n-4} \\ \vdots & & \vdots & \ddots \end{bmatrix}.$$

Let $\boldsymbol{U}$ be the product of the matrices corresponding to the above described elementary operations on $\boldsymbol{H}$. Then,

$$\boldsymbol{U} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ u_{12} & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & 1 \end{bmatrix},$$

where, for example, $u_{12} = -a_n/a_{n-1}$. Premultiplying the Hurwitz matrix (7.38) by the matrix $\boldsymbol{U}$ given above yields

$$\boldsymbol{UH} = \begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} & \cdots & \cdots \\ 0 & r_{21} & r_{22} & \cdots & r_{2,n-1} \\ 0 & 0 & r_{31} & \cdots & r_{3,n-2} \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & r_{n1} \end{bmatrix}. \tag{7.46}$$

Because the matrix $\boldsymbol{U}$ is lower triangular with the ones on the diagonal, the leading principal minors of $\boldsymbol{H}$ and $\boldsymbol{UH}$ are the same, where

$$H_1 = a_{n-1}, \quad H_2 = a_{n-1}r_{21}, \quad \ldots \quad , H_i = a_{n-1}r_{21} \times \cdots \times r_{i1}, \quad i = 1, 2, \ldots, n.$$

Hence,

$$a_{n-1} = H_1, \quad \text{and} \quad r_{i1} = \frac{H_i}{H_{i-1}} \quad \text{for} \quad i = 2, \ldots, n.$$

By the Hurwitz theorem, the zeros of the polynomial $p(s)$ are in the open left-half plane if and only if the coefficients r_{i1}, $i = 1, 2 \ldots, n$, which are the elements of the first column of the Routh array, are all positive. The proof is complete.

□

It is easy to verify that a necessary and sufficient condition for the first- and second-order polynomials to have their zeros in the open left-half plane is that all their coefficients be nonzero and have the same sign. However, this condition is only necessary and not sufficient for third and higher order polynomials to have all their zeros in the open left-half plane.

It is possible to extend the above Routh stability criterion so that it can be used to find the number of polynomial zeros with positive real parts. The number of zeros of $p(s)$ with positive real parts is equal to the number of sign changes in the first column of the Routh array. For a proof of this statement using the law of inertia of quadratic forms see, for example, Kaczorek [164, Section 8.4.3].

Example 7.7 We use the Routh stability criterion to analyze stability of the linearized model of the flyball governor. The governor's linearized model is given by (4.7) on page 193. The model's characteristic equation is

$$\begin{aligned} \det \begin{bmatrix} s & -1 & 0 \\ \frac{g \sin^2 x_{1e}}{\cos x_{1e}} & s + \frac{b}{m} & -\frac{2g \sin x_{1e}}{x_{3e}} \\ \frac{\kappa}{I} \sin x_{1e} & 0 & s \end{bmatrix} &= s^3 + \frac{b}{m} s^2 + \frac{g \sin^2 x_{1e}}{\cos x_{1e}} s + \frac{2g\kappa \sin^2 x_{1e}}{I x_{3e}} \\ &= s^3 + a_2 s^2 + a_1 s + a_0. \end{aligned} \tag{7.47}$$

The coefficients of the above third order polynomial are all positive. Therefore, the necessary condition for stability is satisfied. We next construct the Routh array for (7.47) to get,

$$\begin{array}{c|cc} s^3 & 1 & a_1 \\ s^2 & a_2 & a_0 \\ s^1 & \frac{1}{a_2}(a_2 a_1 - a_0) & \\ s^0 & a_0 & \end{array} \tag{7.48}$$

Because the characteristic polynomial coefficients are all positive, the linearized model is asymptotically stable if and only if

$$a_2 a_1 - a_0 = \frac{b}{m} \frac{g \sin^2 x_{1e}}{\cos x_{1e}} - \frac{2g\kappa \sin^2 x_{1e}}{I x_{3e}} > 0. \tag{7.49}$$

Let

$$\nu = \frac{x_{3e}}{2\kappa \cos x_{1e}}.$$

Using the above notation, we express (7.49) as

$$\frac{bI}{m}\nu > 1. \tag{7.50}$$

It follows from condition (7.50) that

1. an increase of the mass m of the weights of the governor has a deteriorating effect on the governor's stability;
2. a decrease of the friction coefficient b has also a harmful effect on stability;
3. a decrease of the moment of inertia I of the flywheel harms stability;
4. a decrease of the coefficient ν harms stability.

The above deductions are attributed by Pontryagin [276, page 219] to the Russian engineer Vyshnegradskiy. Vyshnegradskiy used these observations to propose, in 1876, design rules for governors to regulate engines without engine speeds fluctuating.

Example 7.8 This example comes from Chen [55, p. 139]. Its purpose is to show that in an LTV system, $\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x}$, even if the eigenvalues of $\boldsymbol{A}(t)$ are defined at every t and in the open left-hand complex plane, the system does not have to be stable; in fact it may be unstable. That is, the concept of eigenvalues is not useful in the stability investigation in LTV systems. The LTV system we consider is modeled by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} = \begin{bmatrix} -1 & e^{2t} \\ 0 & -1 \end{bmatrix}\boldsymbol{x}.$$

It is easy to see that the matrix $\boldsymbol{A}(t)$ has a repeated eigenvalue at -1 for all t. In the LTI case this would imply asymptotic stability. One can verify that the state transition matrix, $\boldsymbol{\Phi}(t,0)$, for the above LTV system has the form

$$\boldsymbol{\Phi}(t,0) = \begin{bmatrix} e^{-t} & 0.5\left(e^{t} - e^{-t}\right) \\ 0 & e^{-t} \end{bmatrix}.$$

The $(1,2)$-th element of the state transition matrix is unbounded, which means that the system is unstable!

7.8 Stability of Nonlinear Systems

In this section, we are concerned with stability analysis of nonlinear time-varying dynamical system models described by (7.1), that is, models of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0,$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state vector, and $\boldsymbol{f}$ is a vector-valued function.

To proceed, we need the following lemma. Let $B_q(\mathbf{0})$ denote an open ball with the center at $\mathbf{0}$ and radius q, and let $\mathcal{S} = \bar{B}_R(\mathbf{0}) \subset \mathbb{R}^n$ be a closed ball, a compact set, that contains the origin $\boldsymbol{x} = \mathbf{0}$ in its interior.

Lemma 7.2 *For a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$, where $V(\mathbf{0}) = 0$, we have $V(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathcal{S} \setminus \{\mathbf{0}\}$, if and only if there is a continuous, strictly increasing, function $\psi : \mathbb{R} \to \mathbb{R}$ such that $\psi(0) = 0$, $\psi(r) > 0$ for $r > 0$, and $V(\boldsymbol{x}) \geq \psi(r)$, where $r = \|\boldsymbol{x}\|$ and $r \leq R$.*

Proof Sufficiency ($\Leftarrow$) is clear.

To prove necessity ($\Rightarrow$), we define, following Willems [351, page 21], the function $\kappa : \mathbb{R} \to \mathbb{R}$,

$$\kappa(y) = \min_{\substack{\boldsymbol{x} \in \mathcal{S} \\ \|\boldsymbol{x}\| = y}} V(\boldsymbol{x}) \tag{7.51}$$

Note that the function κ is well defined because the set

$$\{\boldsymbol{x} : \|\boldsymbol{x}\| = y\} \cap \mathcal{S} \tag{7.52}$$

is compact and V is continuous. Therefore, by the Weierstrass theorem on page 985, the function V achieves its infimum over the set (7.52).

We next show that κ is continuous on $\mathcal{S} = \bar{B}_R(\mathbf{0})$. Then, we construct an appropriate function ψ satisfying the hypotheses of the lemma. To show that κ is continuous on $\mathcal{S}$, we use the fact that V is uniformly continuous in $\mathcal{S}$. To this end let $\boldsymbol{x}^{(1)} \in \mathcal{S}$ be such that $\|\boldsymbol{x}^{(1)}\| = y$ and $\kappa(y) = V\left(\boldsymbol{x}^{(1)}\right)$. Next, let Δy be an increment of y and let $\boldsymbol{x}^{(2)} \in \mathcal{S}$ be a point, a vector, such that $y + \Delta y = \|\boldsymbol{x}^{(2)}\|$. For example, we can take

$$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} + \frac{\boldsymbol{x}^{(1)}}{\|\boldsymbol{x}^{(1)}\|} \Delta y.$$

Observe that

$$V\left(\boldsymbol{x}^{(2)}\right) \geq \kappa(y + \Delta y).$$

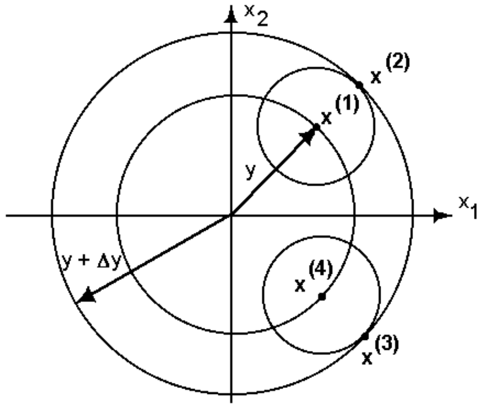


Figure 7.8: A two-dimensional illustration of the proof of the continuity of the function κ given by (7.51).

Choose now a vector $\boldsymbol{x}^{(3)}$ such that

$$\|\boldsymbol{x}^{(3)}\| = y + \Delta y \quad \text{and} \quad V\left(\boldsymbol{x}^{(3)}\right) = \kappa(y + \Delta y).$$

Note that for any vector, say $\boldsymbol{x}^{(4)}$, such that $\|\boldsymbol{x}^{(4)}\| = y$ we have $V\left(\boldsymbol{x}^{(4)}\right) \geq \kappa(y)$—see Figure 7.8. We can take

$$\boldsymbol{x}^{(4)} = \boldsymbol{x}^{(3)} - \frac{\boldsymbol{x}^{(3)}}{\|\boldsymbol{x}^{(3)}\|}\Delta y.$$

By the uniform continuity of V, for any $\epsilon > 0$ there exists a $\delta > 0$ such that if

$$\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)}\| < \delta \quad \text{and} \quad \|\boldsymbol{x}^{(3)} - \boldsymbol{x}^{(4)}\| < \delta,$$

then

$$\left| V\left(\boldsymbol{x}^{(1)}\right) - V\left(\boldsymbol{x}^{(2)}\right) \right| < \epsilon \quad \text{and} \quad \left| V\left(\boldsymbol{x}^{(3)}\right) - V\left(\boldsymbol{x}^{(4)}\right) \right| < \epsilon.$$

From $\left| V\left(\boldsymbol{x}^{(1)}\right) - V\left(\boldsymbol{x}^{(2)}\right) \right| < \epsilon$, we obtain

$$-\epsilon < V\left(\boldsymbol{x}^{(1)}\right) - V\left(\boldsymbol{x}^{(2)}\right) < \epsilon.$$

Thus

$$V\left(\boldsymbol{x}^{(1)}\right) > V\left(\boldsymbol{x}^{(2)}\right) - \epsilon.$$

From $\left| V\left(\boldsymbol{x}^{(3)}\right) - V\left(\boldsymbol{x}^{(4)}\right) \right| < \epsilon$, we get

$$V\left(\boldsymbol{x}^{(3)}\right) > V\left(\boldsymbol{x}^{(4)}\right) - \epsilon.$$

From the definition of the function κ it follows that

$$V\left(\boldsymbol{x}^{(2)}\right) \geq V\left(\boldsymbol{x}^{(3)}\right) \quad \text{and} \quad V\left(\boldsymbol{x}^{(4)}\right) \geq V\left(\boldsymbol{x}^{(1)}\right).$$

Because $V\left(\boldsymbol{x}^{(1)}\right) > V\left(\boldsymbol{x}^{(2)}\right) - \epsilon$, we obtain

$$V\left(\boldsymbol{x}^{(1)}\right) > V\left(\boldsymbol{x}^{(3)}\right) - \epsilon$$

and therefore

$$-\epsilon < V\left(\boldsymbol{x}^{(1)}\right) - V\left(\boldsymbol{x}^{(3)}\right). \tag{7.53}$$

Because $V\left(\boldsymbol{x}^{(3)}\right) > V\left(\boldsymbol{x}^{(4)}\right) - \epsilon$, we have

$$V\left(\boldsymbol{x}^{(3)}\right) > V\left(\boldsymbol{x}^{(1)}\right) - \epsilon,$$

and thus

$$V\left(\boldsymbol{x}^{(1)}\right) - V\left(\boldsymbol{x}^{(3)}\right) < \epsilon. \tag{7.54}$$

Combining (7.53) and (7.54), we get

$$\left| V\left(\boldsymbol{x}^{(1)}\right) - V\left(\boldsymbol{x}^{(3)}\right) \right| < \epsilon,$$

or equivalently,

$$|\kappa(y) - \kappa(y + \Delta y)| < \epsilon$$

if

$$\Delta y < \delta,$$

which proves the continuity of κ.

We are now ready to construct a function ψ that satisfies the hypotheses of the lemma. This function comes from Willems [351, page 22] and has the form

$$\boxed{\psi(r) = r \min_{r \leq y \leq R} \frac{\kappa(y)}{y}} \tag{7.55}$$

Note that ψ is continuous, $\psi(0) = 0$, and it is positive and strictly increasing—see Figure 7.9 for a graphical illustration of the way ψ is constructed from κ. The proof of the lemma is now complete.

□

We can characterize a positive definite function V in another way as in the following lemma.

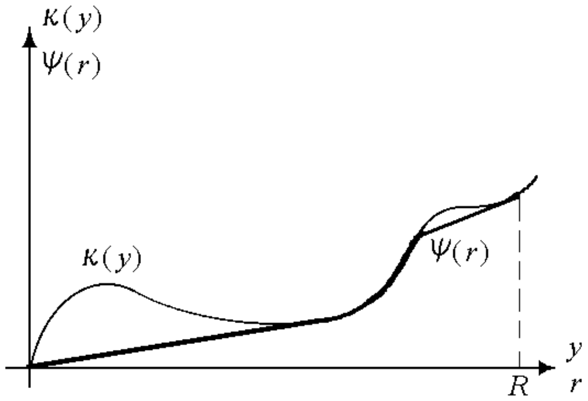


Figure 7.9: Constructing the function ψ given by (7.55).

Lemma 7.3 *A continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is positive definite in $\mathcal{S} = \bar{B}_R(\mathbf{0}) \subset \mathbb{R}^n$, with respect to $\boldsymbol{x} = \mathbf{0}$, if and only if there exists a continuous nondecreasing function $\alpha : \mathbb{R} \to \mathbb{R}$, such that $\alpha(0) = 0$, $\alpha(r) > 0$ for $r > 0$, and $V(\boldsymbol{x}) \geq \alpha(r)$, where $r = \|\boldsymbol{x}\|$ and $r \leq R$.*

A possible choice for $\alpha(\cdot)$ is

$$\alpha(r) = \min_{r \leq \|\boldsymbol{x}\| \leq R} V(\boldsymbol{x}).$$

Note that the function $\alpha(\cdot)$ is nondecreasing because as r increases, the minimum is taken over a smaller region. Furthermore, $\alpha(r) > 0$ whenever $r > 0$ because $V(\boldsymbol{x}) > 0$ for $\boldsymbol{x} \in \mathcal{S} \setminus \{\mathbf{0}\}$. The function α is continuous because V is continuous.

The main difference between the above two characterizations of positive definite functions is that in Lemma 7.2 the function ψ is strictly increasing rather than just nondecreasing. In our further discussion, we will exploit extensively the existence of a such a function ψ associated with a positive definite function V.

So far we only considered positive definite, or negative definite functions, that are time-invariant. We now extend our definition of positive definiteness to time-varying functions of the form $W(t, \boldsymbol{x})$. We first define a time-varying positive semi-definite function. As before, we let $\mathcal{S}$ to denote a compact subset of $\mathbb{R}^n$ containing the origin $\boldsymbol{x} = \mathbf{0}$ in its interior.

Definition 7.11 *A real-valued function $W : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ is positive semi-definite, with respect to $\boldsymbol{x} = \boldsymbol{0}$, in a region $\mathcal{S} \subseteq \mathbb{R}^n$ if for all t and all $\boldsymbol{x}$ in $\mathcal{S}$*

(i) W is continuous and has continuous partial derivatives with respect to all of its arguments, that is, $W \in \mathcal{C}^1$,

(ii) $W(t, \boldsymbol{0}) = 0$,

(iii) $W(t, \boldsymbol{x}) \geq 0$.

◇

Definition 7.12 *A real-valued function $W : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ is positive definite, with respect to $\boldsymbol{x} = \boldsymbol{0}$, in a region $\mathcal{S} \subseteq \mathbb{R}^n$ if for all t and all $\boldsymbol{x}$ in $\mathcal{S}$*

(i) $W \in \mathcal{C}^1$,

(ii) $W(t, \boldsymbol{0}) = 0$,

(iii) there exists a positive definite time-invariant function V such that $W(t, \boldsymbol{x}) \geq V(\boldsymbol{x})$.

◇

We next introduce a notion of a *decrescent* function.

Definition 7.13 *A time-varying function $W : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ is decrescent in a region $\mathcal{S}$ if a positive definite function $\tilde{V}$ exists such that for all $\boldsymbol{x}$ in $\mathcal{S}$ and all t*

$$|W(t, \boldsymbol{x})| \leq \tilde{V}(\boldsymbol{x}),$$

that is, W tends to zero uniformly with respect to t as $\|\boldsymbol{x}\| \to 0$.

◇

Consider now an equilibrium state of the dynamic system modeled by (7.1), that is, a constant state $\boldsymbol{x}_e$ such that $\boldsymbol{f}(t, \boldsymbol{x}_e) = \boldsymbol{0}$ for all t. Let $W(t, \boldsymbol{x})$ be a time-dependent real-valued function whose time derivative along the solution $\boldsymbol{x}(t) = \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$ of the given differential equation, is

$$\frac{dW(t, \boldsymbol{x})}{dt} = \dot{W}(t, \boldsymbol{x}) = \frac{\partial W}{\partial t} + \nabla W^\top \dot{\boldsymbol{x}}.$$

Then, we have the following theorem concerning the stability of the equilibrium point in the sense of Lyapunov.

Theorem 7.7 *If in some neighborhood $\mathcal{S}$ of the equilibrium state $\boldsymbol{x}_e$, there is a positive definite function W, with respect to $\boldsymbol{x}_e$, such that its time derivative along any solution of $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t))$ is negative semi-definite in $\mathcal{S}$, then the equilibrium state $\boldsymbol{x}_e$ is stable. If W is also decrescent, then the equilibrium state is uniformly stable.*

Proof Without loss of generality, we assume that $\boldsymbol{x}_e = \boldsymbol{0}$ and follow the arguments of Willems [351, pages 29, 30]. Because W is positive definite, by Definition 7.12 and Lemma 7.2 there is a strictly increasing function $\psi_1(r)$, $\psi(0) = 0$, such that

$$W(t, \boldsymbol{x}) \geq \psi_1(\|\boldsymbol{x}\|),$$

where $\|\boldsymbol{x}\| \leq R$, and R is the radius of the largest ball contained in $\mathcal{S}$. Let

$$\psi_2(t_0, r) = \max_{\|\boldsymbol{x}\| \leq r} W(t_0, \boldsymbol{x}).$$

Then

$$\psi_1(r) \leq \psi_2(t_0, r).$$

Thus, for any $\epsilon > 0$ there is a positive δ that depends on ϵ and t_0, such that

$$\psi_2(t_0, \delta) < \min\{\psi_1(\epsilon), \psi_1(R)\}.$$

Figure 7.10 illustrates how to determine a δ given an ϵ. By assumption $dW(t, \boldsymbol{x})/dt \leq 0$ along any solution of the system. Hence $W(t, \boldsymbol{x})$ is a non-increasing function and therefore

$$W(t, \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)) \leq W(t_0, \boldsymbol{x}_0) \qquad \text{for all} \qquad t \geq t_0.$$

If $\|\boldsymbol{x}_0\| < \delta$, then for $t \geq t_0$ we have

$$\psi_1(\epsilon) > \psi_2(t_0, \delta) \geq W(t_0, \boldsymbol{x}_0) \geq W(t, \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)) \geq \psi_1(\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\|).$$

Therefore,

$$\psi_1(\epsilon) > \psi_1(\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\|).$$

Because ψ_1 is strictly increasing, we have to have

$$\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| < \epsilon$$

for all $t \geq t_0$. This means that the null solution is (locally) stable, which completes the proof of the first part of the theorem.

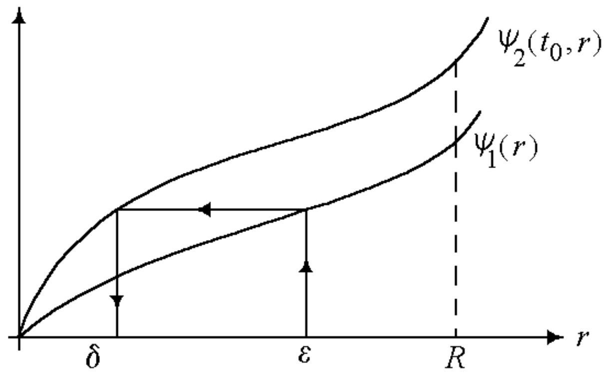


Figure 7.10: Determining a δ given an ϵ.

If W is also decrescent, then there is a nondecreasing function $\tilde{\psi}_2 : \mathbb{R} \to \mathbb{R}$, $\tilde{\psi}_2(0) = 0$, and

$$W(t, \boldsymbol{x}) \leq \tilde{\psi}_2(\|\boldsymbol{x}\|)$$

for all $\boldsymbol{x}$ such that $\|\boldsymbol{x}\| \leq R$ and all t. Indeed, because $W(t, \boldsymbol{x})$ is positive definite and decrescent, there exists a positive definite $V(\boldsymbol{x})$ such that $W(t, \boldsymbol{x}) \leq V(\boldsymbol{x})$. Then, we can take $\tilde{\psi}_2$ defined as

$$\tilde{\psi}_2(r) = \max_{\|\boldsymbol{x}\| \leq r} V(\boldsymbol{x}).$$

Let

$$\tilde{\psi}_2(\delta) < \min\{\psi_1(\epsilon), \psi_1(R)\}.$$

Note that δ depends now only on ϵ, and not on t_0 as before. To show the uniform stability of the null solution, we repeat the argument above with $\tilde{\psi}_2$ replacing ψ_2. This completes the proof of the theorem.

□

The next theorem gives us a sufficiency condition for an equilibrium point to be uniformly asymptotically stable.

Theorem 7.8 *If in some neighborhood $\mathcal{S}$ of the given equilibrium point $\boldsymbol{x}_e$ there is a positive definite, with respect to $\boldsymbol{x}_e$, and decrescent function W such that its time derivative along* *any solution of the system model (7.1) is negative definite in $\mathcal{S}$, then the equilibrium state is uniformly asymptotically stable.*

Proof As before, without loss of generality, we assume that $\boldsymbol{x}_e = \mathbf{0}$ and then follow the arguments of Willems [351, pages 30, 31]. The previous theorem ensures uniform stability of the equilibrium state $\boldsymbol{x}_e$. Thus, only the uniform convergence remains to be shown. From the assumptions of the theorem it follows that there exist strictly increasing functions ψ_1 and ψ_3, and a non-decreasing function ψ_2 such that

$$\begin{aligned}\psi_1(0) &= \psi_2(0) = \psi_3(0) = 0,\\ \psi_1(\|\boldsymbol{x}\|) &\leq W(t, \boldsymbol{x}) \leq \psi_2(\|\boldsymbol{x}\|),\\ \psi_3(\|\boldsymbol{x}\|) &\leq -\dot{W}(t, \boldsymbol{x}),\end{aligned}$$

for all t and all $\boldsymbol{x}$ such that $\|\boldsymbol{x}\| \leq R$, where R is the radius of the largest ball contained in $\mathcal{S}$. First, select δ_1 so that

$$\psi_2(\delta_1) < \psi_1(R). \tag{7.56}$$

Next, given an $\epsilon_1 > 0$, define positive constants δ_2 and T as follows:

$$\psi_2(\delta_2) < \min\{\psi_1(\epsilon_1), \psi_2(\delta_1)\}, \tag{7.57}$$

$$T = \psi_1(R)/\psi_3(\delta_2). \tag{7.58}$$

Note that δ_1 and T are independent of t_0.

We prove the theorem in two stages. First, we show that for some $t_2 \in [t_0, t_0 + T]$,

$$\|\boldsymbol{x}(t_2; t_0, \boldsymbol{x}_0)\| < \delta_2$$

whenever $\|\boldsymbol{x}_0\| < \delta_1$. Then, we show that once $\|\boldsymbol{x}(t_2; t_0, \boldsymbol{x}_0)\| < \delta_2$, then

$$\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| < \epsilon_1 \quad \text{for all} \quad t \geq t_2.$$

We begin with proving the claim that for some $t_2 \in [t_0, t_0 + T]$,

$$\|\boldsymbol{x}(t_2; t_0, \boldsymbol{x}_0)\| < \delta_2$$

whenever $\|\boldsymbol{x}_0\| < \delta_1$. We prove this claim by contradiction. We assume that for all $t \in [t_0, t_0 + T]$,

$$\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| \geq \delta_2 \quad \text{if} \quad \|\boldsymbol{x}_0\| < \delta_1. \tag{7.59}$$

Using (7.59), (7.58), and (7.56), we obtain

$$
\begin{aligned}
0 < \psi_1(\delta_2) &\leq \psi_1(\|\boldsymbol{x}(t_0+T;t_0,\boldsymbol{x}_0)\|) \\
&\leq W[t_0+T,\boldsymbol{x}(t_0+T;t_0,\boldsymbol{x}_0)] \\
&= W(t_0,\boldsymbol{x}_0) + \int_{t_0}^{t_0+T} \dot{W}[t,\boldsymbol{x}(t;t_0,\boldsymbol{x}_0)]dt \\
&\leq W(t_0,\boldsymbol{x}_0) - \int_{t_0}^{t_0+T} \psi_3(\|\boldsymbol{x}(t;t_0,\boldsymbol{x}_0)\|)dt \\
&\leq W(t_0,\boldsymbol{x}_0) - T\psi_3(\delta_2) \\
&\leq \psi_2(\|\boldsymbol{x}_0\|) - T\psi_3(\delta_2) \\
&< \psi_2(\delta_1) - \psi_1(R) \\
&< 0,
\end{aligned}
$$

which is a contradiction, and thus we proved the first part of the theorem. We now proceed with the second part. It follows from the first part that if $\|\boldsymbol{x}_0\| < \delta_1$, then for some $t_2 \in [t_0, t_0+T]$, we have $\|\boldsymbol{x}(t_2;t_0,\boldsymbol{x}_0)\| < \delta_2$. Hence, for $t \geq t_2$,

$$\psi_1(\|\boldsymbol{x}(t;t_0,\boldsymbol{x}_0)\|) \leq W\left(t,\boldsymbol{x}(t;t_0,\boldsymbol{x}_0)\right) \leq W\left(t_2,\boldsymbol{x}(t_2;t_0,\boldsymbol{x}_0)\right) < \psi_2(\delta_2) < \psi_1(\epsilon_1).$$

Therefore,

$$\psi_1(\|\boldsymbol{x}(t;t_0,\boldsymbol{x}_0)\|) < \psi_1(\epsilon_1),$$

and because ψ_1 is strictly increasing,

$$\|\boldsymbol{x}(t;t_0,\boldsymbol{x}_0)\| < \epsilon_1$$

for all $t \geq t_2$ which proves the uniform convergence. The proof of the theorem is thus complete.

□

Example 7.9 Given a dynamical system model

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} -0.75 & 0.3536 \\ 0.3536 & -0.5 \end{bmatrix} \boldsymbol{x}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0.$$

The solution $\boldsymbol{P} = \boldsymbol{P}^\top$ to the Lyapunov matrix equation $\boldsymbol{A}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$, where $\boldsymbol{Q} = 2\boldsymbol{I}_2$, is

$$\boldsymbol{P} = \begin{bmatrix} 2 & \sqrt{2} \\ \sqrt{2} & 3 \end{bmatrix}.$$

The eigenvalues of $\boldsymbol{P}$ are

$$\lambda_1 = \lambda_{\min}(\boldsymbol{P}) = 1, \quad \lambda_2 = \lambda_{\max}(\boldsymbol{P}) = 4.$$

In general, for a symmetric matrix $\boldsymbol{P}$,

$$\lambda_{\min}(\boldsymbol{P})\|\boldsymbol{x}\|^2 \le \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} \le \lambda_{\max}(\boldsymbol{P})\|\boldsymbol{x}\|^2.$$

Therefore, we can take

$$\psi_1(\|\boldsymbol{x}\|) = \lambda_{\min}(\boldsymbol{P})\|\boldsymbol{x}\|^2, \quad \psi_2(\|\boldsymbol{x}\|) = \lambda_{\max}(\boldsymbol{P})\|\boldsymbol{x}\|^2, \quad \text{and} \quad \psi_3(\|\boldsymbol{x}\|) = \lambda_{\min}(\boldsymbol{Q})\|\boldsymbol{x}\|^2.$$

In our example, $\lambda_{\min}(\boldsymbol{Q}) = 2$.

- Given $\varepsilon = 4$; we find $\delta > 0$ such that if $\|\boldsymbol{x}_0\| \le \delta$, then $\|\boldsymbol{x}(t)\| \le \varepsilon$ for $t \ge 0$. We obtain $\delta = \delta(\varepsilon)$ by solving the equation,

$$\psi_1(\varepsilon) = \psi_2(\delta),$$

to get

$$\delta = \sqrt{\frac{\lambda_{\min}(\boldsymbol{P})}{\lambda_{\max}(\boldsymbol{P})}}\varepsilon = 2.$$

- Given $\boldsymbol{x}_0 = \begin{bmatrix} 8 & 6 \end{bmatrix}^\top$ and $\varepsilon_1 = 2$; we find T such that $\|\boldsymbol{x}(t)\| \le \varepsilon_1$ for all $t \ge T$. We have $\|\boldsymbol{x}_0\|_2 = 10$. Let $\delta_1 = 10$. We then compute R, δ_2, and T from the equations:

$$\begin{aligned} \psi_2(\delta_1) &= \psi_1(R) \\ \psi_2(\delta_2) &= \min\left(\psi_1(\varepsilon_1), \psi_2(\delta_1)\right) \\ T &= \frac{\psi_1(R)}{\psi_3(\delta_2)}. \end{aligned}$$

Using the first equation, we find

$$R = \sqrt{\frac{\lambda_{\max}(\boldsymbol{P})}{\lambda_{\min}(\boldsymbol{P})}}\delta_1 = 20.$$

We then solve for $\delta_2 \le \delta_1$ to get

$$\delta_2 = \sqrt{\frac{\lambda_{\min}(\boldsymbol{P})}{\lambda_{\max}(\boldsymbol{P})}}\varepsilon_1 = 1.$$

Hence,

$$T = \frac{\lambda_{\min}(\boldsymbol{P})R^2}{\lambda_{\min}(\boldsymbol{Q})\delta_2^2} = \frac{\lambda_{\min}(\boldsymbol{P})\frac{\lambda_{\max}(\boldsymbol{P})}{\lambda_{\min}(\boldsymbol{P})}\delta_1^2}{\lambda_{\min}(\boldsymbol{Q})\frac{\lambda_{\min}(\boldsymbol{P})}{\lambda_{\max}(\boldsymbol{P})}\varepsilon_1^2} = \frac{\lambda_{\max}^2(\boldsymbol{P})\delta_1^2}{\lambda_{\min}(\boldsymbol{Q})\lambda_{\min}(\boldsymbol{P})\varepsilon_1^2} = 200.$$

Proposition 7.1 *If in some neighborhood* $\mathcal{S}$ *of* $\boldsymbol{x}_e = \mathbf{0}$ *the positive definite and decrescent function* W *satisfies*

$$c_1\|\boldsymbol{x}\|^2 \le W(t,\boldsymbol{x}) \le c_2\|\boldsymbol{x}\|^2$$

for some positive constants c_1 *and* c_2 *and its time derivative along any solution of the system (7.1) is negative definite in* $\mathcal{S}$ *and such that for some positive constant* c_3*,*

$$\dot{W}(t,\boldsymbol{x}) \le -c_3\|\boldsymbol{x}\|^2,$$

then the equilibrium state $\boldsymbol{x}_e = \mathbf{0}$ *is uniformly exponentially stable.*

Proof We have

$$\begin{aligned}\dot{W}(t,\boldsymbol{x}) &\le -c_3\|\boldsymbol{x}\|^2\\ &= -\frac{c_3}{c_2}\left(c_2\|\boldsymbol{x}\|^2\right)\\ &\le -\frac{c_3}{c_2}W(t,\boldsymbol{x}).\end{aligned}$$

Applying to the above Theorem A.20, which is called the comparison theorem, we obtain

$$W(t,\boldsymbol{x}) \le W(t_0,\boldsymbol{x}_0)\exp\left(-\frac{c_3}{c_2}(t-t_0)\right).$$

Hence,

$$\begin{aligned}\|\boldsymbol{x}\| &\le \sqrt{\frac{W(t,\boldsymbol{x})}{c_1}}\\ &\le \sqrt{\frac{W(t_0,\boldsymbol{x}_0)\exp\left(-\frac{c_3}{c_2}(t-t_0)\right)}{c_1}}\\ &\le \left(\frac{c_2}{c_1}\right)^{\frac{1}{2}}\|\boldsymbol{x}(t_0)\|\exp\left(-\frac{c_3}{2c_2}(t-t_0)\right),\end{aligned}$$

which means that the equilibrium state $\boldsymbol{x}_e = \mathbf{0}$ is uniformly exponentially stable.

□

The above theorems were concerned with local stability. We now proceed to prove sufficiency conditions for global stability. For this we need the following definitions. We first define a *radially unbounded* or *coercive* function.

Definition 7.14 *A time-varying function $W : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ is radially unbounded, or coercive, if as $\|\boldsymbol{x}\| \to \infty$, the function W tends to infinity uniformly with respect to t, that is, for any given positive real number M, there is a positive real number N so that $W(t, \boldsymbol{x}) > M$ for all t and all $\boldsymbol{x}$ whenever $\|\boldsymbol{x}\| > N$.*

◇

The next two definitions are concerned with *Lagrange stable equilibrium states.*

Definition 7.15 *The equilibrium state $\boldsymbol{x}_e$ is called Lagrange stable, or bounded, if for every real number $d > 0$ there exists a positive real number $b = b(t_0, d)$ such that if*

$$\|\boldsymbol{x}_0 - \boldsymbol{x}_e\| < d,$$

then

$$\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0) - \boldsymbol{x}_e\| < b$$

for all $t \geq t_0$.

◇

Definition 7.16 *The equilibrium state $\boldsymbol{x}_e$ is uniformly bounded if the number b in Definition 7.15 can be taken independent of the initial time t_0.*

◇

We now define *uniformly asymptotically stable in the large* equilibrium state.

Definition 7.17 *If the equilibrium state $\boldsymbol{x}_e$ is uniformly stable, uniformly bounded, and all trajectories are uniformly convergent to $\boldsymbol{x}_e$, then this equilibrium state is called uniformly asymptotically stable in the large, or globally uniformly asymptotically stable.*

◇

The following theorem gives a sufficiency condition for an equilibrium state to be globally uniformly asymptotically stable.

Theorem 7.9 *If there is a function W that is positive definite for all $\boldsymbol{x} \neq \boldsymbol{x}_e$, $W(t, \boldsymbol{x}_e) = 0$, decrescent, radially unbounded, and if its time derivative along any solution of the system is negative definite for all $\boldsymbol{x}$, then the equilibrium state $\boldsymbol{x}_e$ is globally uniformly asymptotically stable.*

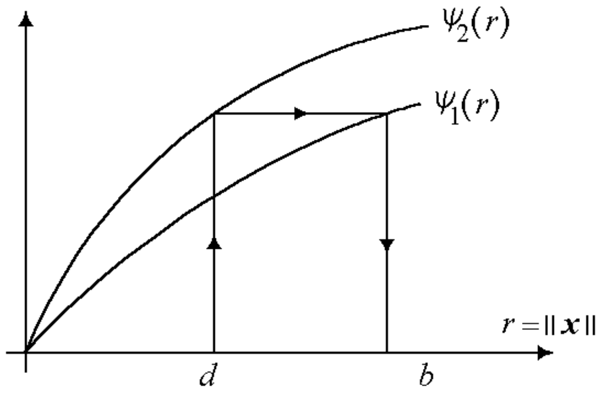


Figure 7.11: Determining a constant $b > 0$ given $d > 0$ in the proof of Theorem 7.9.

Proof We assume, for simplicity, that $\boldsymbol{x}_e = \boldsymbol{0}$ and follow the arguments of Willems [351, pages 31, 32]. By Theorem 7.7, the equilibrium state $\boldsymbol{x}_e$ is uniformly stable. Similarly as in Theorem 7.8, we construct the functions ψ_1, ψ_2, and ψ_3. Since, by assumption, W is radially unbounded, $\psi_1(\|\boldsymbol{x}\|) \to \infty$ as $\|\boldsymbol{x}\| \to \infty$. Therefore, for any constant $d > 0$, there is a constant $b > 0$ such that $\psi_1(b) > \psi_2(d)$. See Figure 7.11 for an illustration of the procedure that can be used to find the constant b. If $\|\boldsymbol{x}_0\| < d$, then $\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| < b$ for all $t \geq t_0$. Indeed, because

$$\psi_1(b) > \psi_2(d) \geq W(t_0, \boldsymbol{x}_0) \geq W(t, \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)) \geq \psi_1(\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\|),$$

then

$$\psi_1(b) > \psi_1(\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\|).$$

The above implies that

$$\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| < b$$

because ψ_1 is strictly increasing. Hence all trajectories of the system are uniformly bounded. To prove their convergence to the equilibrium state, we define δ_2, T, and b, given d and ϵ_1, as

$$\begin{aligned}
\psi_2(d) &< \psi_1(b), \\
\psi_2(\delta_2) &< \psi_1(\epsilon_1), \\
T &= \psi_1(b)/\psi_3(\delta_2).
\end{aligned}$$

If $\|\boldsymbol{x}_0\| < d$, then for all $t > t_0 + T$, we have

$$\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| < \epsilon_1,$$

which can be shown using the same arguments as in the proof of Theorem 7.8 with b playing the role of R, and d playing the role of δ_1. This proves that all trajectories of the system converge uniformly to the equilibrium state. Hence, the equilibrium state is globally uniformly asymptotically stable.

□

We now give a theorem that can be used as a test for checking the instability of nonlinear dynamic systems. This instability test is illustrated in Figure 7.12.

Theorem 7.10 *Let $\boldsymbol{x}_e$ be an equilibrium state of the system $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$, that is, $\boldsymbol{f}(t, \boldsymbol{x}_e) = \mathbf{0}$. If there exists a function $W : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ such that in the region $\mathcal{S} = \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}_e\| \leq R\}$*

(i) $W \in \mathcal{C}^1$,

(ii) $W(t, \boldsymbol{x}_e) = 0$,

(iii) $\dot{W}(t, \boldsymbol{x}) > 0$ *for all* $\boldsymbol{x} \in \mathcal{S} \setminus \{\boldsymbol{x}_e\}$,

(iv) in every neighborhood of $\boldsymbol{x}_e$ in $\mathcal{S}$, for any t_0 there is at least one point $\boldsymbol{x}$ at which $W(t_0, \boldsymbol{x}) > 0$, and

(v) $W(t, \boldsymbol{x})$ is bounded in $\mathcal{S}$,

then the equilibrium state $\boldsymbol{x}_e$ is unstable.

Proof Without loss of generality, assume that $\boldsymbol{x}_e = \mathbf{0}$. By assumption *(iv)*, for any t_0 and any δ, $0 < \delta < R$, there exists $\boldsymbol{x}_\delta$, $0 < \|\boldsymbol{x}_\delta\| < \delta$, such that $W(t_0, \boldsymbol{x}_\delta) > 0$. The system trajectory $\boldsymbol{x}(t) = \boldsymbol{x}(t; t_0, \boldsymbol{x}_\delta)$ cannot approach the equilibrium state $\boldsymbol{x}_e = \mathbf{0}$ as $t \to \infty$ because $W(t, \mathbf{0}) = 0$ but $W(t_0, \boldsymbol{x}_\delta) > 0$ and $\dot{W}(t, \boldsymbol{x}) > 0$ for $\boldsymbol{x} \neq \boldsymbol{x}_e$. Since $\dot{W}$ is continuous, because $W \in \mathcal{C}^1$, and the trajectory is bounded away from the equilibrium state, there is a constant $m > 0$ such that for all $t \geq t_0$,

$$\dot{W}(t, \boldsymbol{x}(t)) \geq m > 0,$$

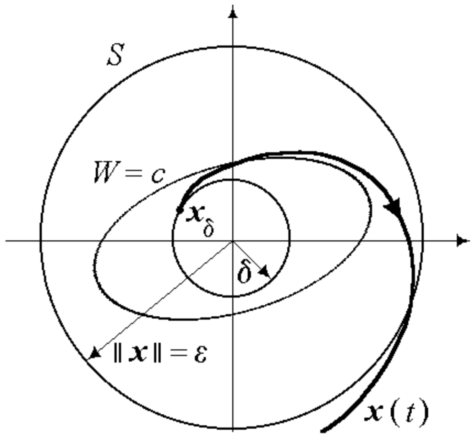


Figure 7.12: An illustration of Theorem 7.10.

as long as

$$\|\boldsymbol{x}(t)\| \leq R, \qquad \boldsymbol{x} \neq \mathbf{0}.$$

Hence,

$$W(t, \boldsymbol{x}) = W(t_0, \boldsymbol{x}_\delta) + \int_{t_0}^{t} \dot{W}(s, \boldsymbol{x}(s))ds \geq W(t_0, \boldsymbol{x}_\delta) + m(t - t_0). \tag{7.60}$$

By condition (v), the function W is bounded in $\mathcal{S}$.. However, the right hand side of (7.60) is unbounded. Therefore the trajectory $\boldsymbol{x}(t; t_0, \boldsymbol{x}_\delta)$ cannot remain infinitely long in $\mathcal{S}$. Thus, for some $t_1 > t_0$ the trajectory reaches the boundary of the region $\|\boldsymbol{x}\| \leq R$. Hence, given any ϵ, $0 < \epsilon < R$, then for any $\delta > 0$, no matter how small, there is at least one trajectory such that $\|\boldsymbol{x}_\delta\| < \delta$ but $\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_\delta)\| \geq \epsilon$ for some $t_1 > t_0$. Therefore, the equilibrium state is unstable.

□

Corollary 7.1 *If there exists a function $W : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ such that in the region $\mathcal{S} = \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}_e\| \leq R\}$*

(i) $W \in \mathcal{C}^1$,

(ii) $W(t, \boldsymbol{x}_e) = 0$,

(iii) $\dot{W}(t, \boldsymbol{x}) < 0$ *for all* $\boldsymbol{x} \in \mathcal{S} \setminus \{\boldsymbol{x}_e\}$,

(iv) in every neighborhood of $\boldsymbol{x}_e$ in $\mathcal{S}$, for any t_0 there is at least one point $\boldsymbol{x}$ at which $W(t_0, \boldsymbol{x}) < 0$, and

(v) $W(t, \boldsymbol{x})$ is bounded in $\mathcal{S}$,

then the equilibrium state $\boldsymbol{x}_e$ is unstable.

Note that if $W(t, \boldsymbol{x}) = W(\boldsymbol{x})$, that is, if W is time-invariant, then Condition (v) of Theorem 7.10 and Corollary 7.1 is not needed because $W = W(\boldsymbol{x})$ being continuous on the compact set $\mathcal{S}$ must be bounded in $\mathcal{S}$.

7.9 Lyapunov's Indirect Method

In this section, we are concerned with the problem of investigating stability properties of an equilibrium state of a nonlinear system based on its linearization about the given equilibrium. We devise a method that allows one to determine whether the equilibrium of the nonlinear system is asymptotically stable or unstable based on the dynamics of the linearized system about this equilibrium. The method is sometimes referred to as the *Lyapunov's first method* or *Lyapunov's indirect method*. For simplicity of notation, we assume that the equilibrium point to be tested for stability is the origin. We consider a nonlinear system model,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{7.61}$$

where $\boldsymbol{f} : \mathcal{D} \to \mathbb{R}^n$ is a continuously differentiable function from a domain $\mathcal{D} \subseteq \mathbb{R}^n$ into $\mathbb{R}^n$. We assume that the origin $\boldsymbol{x} = \boldsymbol{0}$ is in the interior of $\mathcal{D}$ and is an equilibrium state of (7.61), that is, $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$. Let

$$\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{0}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{\boldsymbol{x}=\boldsymbol{0}} \tag{7.62}$$

be the Jacobian matrix of $\boldsymbol{f}$ evaluated at $\boldsymbol{x} = \boldsymbol{0}$. Then, applying Taylor's theorem to the right hand side of (7.61) yields

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}), \tag{7.63}$$

where the function $\boldsymbol{g}$ represents higher-order terms and has the property,

$$\lim_{\|\boldsymbol{x}\| \to 0} \frac{\|\boldsymbol{g}(\boldsymbol{x})\|}{\|\boldsymbol{x}\|} = 0. \tag{7.64}$$

The above suggests that in a "small" neighborhood of the origin, we can approximate the nonlinear system (7.61) with its linearized system about the origin, $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$. The following theorems give conditions under which we can infer about stability properties of the origin of the nonlinear system based on stability properties of the linearized system.

Theorem 7.11 *Let the origin, $\boldsymbol{x} = 0$, be an equilibrium state of the nonlinear system, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$. Then, the origin is an asymptotically stable equilibrium of the nonlinear system if $\boldsymbol{A}$, the Jacobian matrix of $\boldsymbol{f}$ evaluated at the origin, has all its eigenvalues in the open left-half plane.*

Proof Our proof is based on that of Khalil [178, pages 127–128]. To begin with, we suppose that all the eigenvalues of $\boldsymbol{A}$ are in the open left-half plane. Then, by Theorem 7.1, for any positive definite symmetric matrix $\boldsymbol{Q}$, the solution $\boldsymbol{P}$ to the Lyapunov equation (7.7) is also positive definite. We employ the positive definite quadratic form $V(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}$ as a Lyapunov function candidate for the nonlinear system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$. We use (7.63) and the Lyapunov matrix equation (7.7) when evaluating the time derivative of $V(\boldsymbol{x})$ on the trajectories of the nonlinear system. We obtain

$$\begin{aligned} \dot{V}(\boldsymbol{x}) &= 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{f}(\boldsymbol{x}) \\ &= 2\boldsymbol{x}^\top \boldsymbol{P}\left(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x})\right) \\ &= \boldsymbol{x}^\top \left(\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}\right)\boldsymbol{x} + 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{g}(\boldsymbol{x}) \\ &= -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{g}(\boldsymbol{x}). \end{aligned} \tag{7.65}$$

The first term on the right hand side of the above equation is negative definite, while the second term is, in general, indefinite. The function $\boldsymbol{g}$ satisfies

$$\frac{\|\boldsymbol{g}(\boldsymbol{x})\|_2}{\|\boldsymbol{x}\|_2} \to 0 \quad \text{as} \quad \|\boldsymbol{x}\|_2 \to 0.$$

Therefore, for any $\gamma > 0$ there exists $r > 0$ such that

$$\|\boldsymbol{g}(\boldsymbol{x})\|_2 < \gamma \|\boldsymbol{x}\|_2 \quad \text{for all} \quad \|\boldsymbol{x}\|_2 < r. \tag{7.66}$$

Substituting the above into (7.65) gives

$$\begin{aligned} \dot{V}(\boldsymbol{x}) &< -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\gamma \|\boldsymbol{P}\|_2 \|\boldsymbol{x}\|_2^2 \\ &\le -\left(\lambda_{\min}(\boldsymbol{Q}) - 2\gamma\lambda_{\max}(\boldsymbol{P})\right)\|\boldsymbol{x}\|_2^2 \quad \text{for all} \quad \|\boldsymbol{x}\|_2 < r \end{aligned} \tag{7.67}$$

Thus, selecting

$$\gamma < \frac{\lambda_{\min}(\boldsymbol{Q})}{2\lambda_{\max}(\boldsymbol{P})}$$

ensures that $\dot{V}(\boldsymbol{x})$ is negative definite. By Theorem 7.8, we conclude that the origin, $\boldsymbol{x} = \mathbf{0}$, is an asymptotically stable equilibrium state of the nonlinear system, which completes the proof.

□

Theorem 7.12 *Let the origin, $\boldsymbol{x} = \mathbf{0}$, be an equilibrium state of the nonlinear system, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$. Then, the origin is unstable if $\boldsymbol{A}$ has one or more eigenvalues in the right-half plane.*

Proof Our proof follows that of Khalil [178, pages 128–130]. We first consider the special case when $\boldsymbol{A}$ has no eigenvalues on the imaginary axis. Suppose that the eigenvalues cluster into a group of eigenvalues in the open right-plane and a group of eigenvalues in the open left-half plane. Then, we can find a nonsingular matrix $\boldsymbol{T}$ such that

$$\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} -\boldsymbol{A}_1 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2 \end{bmatrix},$$

where both $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ have their eigenvalues in the open left-half plane. A possible candidate for $\boldsymbol{T}$ would be the matrix that transforms $\boldsymbol{A}$ into its real Jordan form. Now, let

$$\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix},$$

where the partition of $\boldsymbol{z}$ is compatible with the dimensions of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. The change of variables, $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x}$, transforms the nonlinear system (7.63) into the form

$$\left.\begin{array}{rcl} \dot{\boldsymbol{z}}_1 & = & -\boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{g}_1(\boldsymbol{z}) \\ \dot{\boldsymbol{z}}_2 & = & \boldsymbol{A}_2\boldsymbol{z}_2 + \boldsymbol{g}_2(\boldsymbol{z}) \end{array}\right\} \tag{7.68}$$

where the functions $\boldsymbol{g}_i$, $i = 1, 2$, have the property that for any $\gamma > 0$, there exists $r > 0$ such that for $i = 1, 2$, we have

$$\|\boldsymbol{g}_i(\boldsymbol{z})\|_2 \leq \gamma \|\boldsymbol{z}\|_2 \quad \text{for all} \quad \|\boldsymbol{z}\|_2 \leq r. \tag{7.69}$$

The origin $\boldsymbol{z} = \mathbf{0}$ is an equilibrium state for the nonlinear system (7.68). Since the matrix $\boldsymbol{T}$ is nonsingular, stability properties of $\boldsymbol{z} = \mathbf{0}$ in the $\boldsymbol{z}$-coordinates are the same as the stability properties of $\boldsymbol{x} = \mathbf{0}$ in the $\boldsymbol{x}$-coordinates. We show that $\boldsymbol{z} = \mathbf{0}$ is an unstable equilibrium of the system (7.68), which implies that $\boldsymbol{x} = \mathbf{0}$ is an unstable equilibrium of the system (7.61). We use Theorem 7.10, which entails constructing a function V satisfying the hypotheses of that theorem. Let $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ be positive definite symmetric matrices of the dimensions of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, respectively. Because $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ have their eigenvalues in the open left-half plane, the Lyapunov matrix equations,

$$\boldsymbol{A}_i^\top \boldsymbol{P}_i + \boldsymbol{P}_i \boldsymbol{A}_i = -\boldsymbol{Q}_i, \quad i = 1, 2,$$

have unique positive definite solutions $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. Let

$$V(\boldsymbol{z}) = \boldsymbol{z}_1^\top \boldsymbol{P}_1 \boldsymbol{z}_1 - \boldsymbol{z}_2^\top \boldsymbol{P}_2 \boldsymbol{z}_2 = \boldsymbol{z}^\top \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{P}_2 \end{bmatrix} \boldsymbol{z}. \tag{7.70}$$

In the subspace, $\left\{ \boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_1^\top & \boldsymbol{z}_2^\top \end{bmatrix}^\top : \boldsymbol{z}_2 = \boldsymbol{0} \right\}$, $V(\boldsymbol{z}) > 0$ in every neighborhood of the origin. Let

$$\mathcal{U} = \left\{ \boldsymbol{z} \in \mathbb{R}^n : \|\boldsymbol{z}\|_2 \le r \right\}.$$

The time derivative of V evaluated on the trajectories of the system (7.68) is

$$\begin{aligned}
\dot{V}(\boldsymbol{z}) &= 2\boldsymbol{z}_1^\top \boldsymbol{P}_1 \dot{\boldsymbol{z}}_1 - 2\boldsymbol{z}_2^\top \boldsymbol{P}_2 \dot{\boldsymbol{z}}_2 \\
&= 2\boldsymbol{z}_1^\top \boldsymbol{P}_1 \left(-\boldsymbol{A}_1 \boldsymbol{z}_1 + \boldsymbol{g}_1(\boldsymbol{z}) \right) - 2\boldsymbol{z}_2^\top \boldsymbol{P}_2 \left(\boldsymbol{A}_2 \boldsymbol{z}_2 + \boldsymbol{g}_2(\boldsymbol{z}) \right) \\
&= \boldsymbol{z}_1^\top \boldsymbol{Q}_1 \boldsymbol{z}_1 + 2\boldsymbol{z}_1^\top \boldsymbol{P}_1 \boldsymbol{g}_1(\boldsymbol{z}) + \boldsymbol{z}_2^\top \boldsymbol{Q}_2 \boldsymbol{z}_2 - 2\boldsymbol{z}_2^\top \boldsymbol{P}_2 \boldsymbol{g}_2(\boldsymbol{z}) \\
&\ge \lambda_{\min}(\boldsymbol{Q}_1) \|\boldsymbol{z}_1\|_2^2 + \lambda_{\min}(\boldsymbol{Q}_2) \|\boldsymbol{z}_2\|_2^2 + 2\boldsymbol{z}^\top \begin{bmatrix} \boldsymbol{P}_1 \boldsymbol{g}_1(\boldsymbol{z}) \\ -\boldsymbol{P}_2 \boldsymbol{g}_2(\boldsymbol{z}) \end{bmatrix} \\
&\ge \lambda_{\min}(\boldsymbol{Q}_1) \|\boldsymbol{z}_1\|_2^2 + \lambda_{\min}(\boldsymbol{Q}_2) \|\boldsymbol{z}_2\|_2^2 \\
&\quad -2 \|\boldsymbol{z}\|_2 \sqrt{\|\boldsymbol{P}_1\|_2^2 \|\boldsymbol{g}_1(\boldsymbol{z})\|_2^2 + \|\boldsymbol{P}_2\|_2^2 \|\boldsymbol{g}_2(\boldsymbol{z})\|_2^2}.
\end{aligned} \tag{7.71}$$

Let

$$\alpha = \min \left(\lambda_{\min}(\boldsymbol{Q}_1), \lambda_{\min}(\boldsymbol{Q}_2) \right) \quad \text{and} \quad \beta = \max \left(\|\boldsymbol{P}_1\|_2, \|\boldsymbol{P}_2\|_2 \right).$$

With (7.69) and the above taken into account, (7.71) evaluates to

$$\dot{V}(\boldsymbol{z}) \ge \left(\alpha - 2\sqrt{2}\beta\gamma \right) \|\boldsymbol{z}\|_2^2 \quad \text{for all} \quad \boldsymbol{z} \in \mathcal{U}. \tag{7.72}$$

If we select

$$\gamma < \frac{\alpha}{2\sqrt{2}\beta},$$

then $\dot{V}(\boldsymbol{z}) > 0$ in $\mathcal{U} \setminus \{\boldsymbol{0}\}$. Thus, all hypotheses of Theorem 7.10 are satisfied and hence the origin is an unstable equilibrium of the system (7.61).

In the general case when $\boldsymbol{A}$ has eigenvalues on the imaginary axis, in addition to eigenvalues in the open right-half plane, we shift the imaginary axis in order to reduce this case to the previous one. Specifically, suppose that $\boldsymbol{A}$ has m eigenvalues in the open right-half plane with $\Re\lambda_i > \delta > 0$. Then, the matrix

$$\boldsymbol{A} - \frac{\delta}{2}\boldsymbol{I}$$

still has m eigenvalues in open right-half plane, but no eigenvalues on the imaginary axis. We can proceed similarly as in the previous case to conclude that the origin is an unstable equilibrium of the corresponding nonlinear system (7.61). For details of this part of the proof, we refer to Khalil [178, page 130].

□

In summary, the above two theorems provide us with a simple method for determining asymptotic stability or instability of an equilibrium state of the nonlinear system (7.61). We compute the Jacobian matrix (7.62) of the nonlinear system, evaluate it at the equilibrium state of interest, and test its eigenvalues. If $\Re\lambda_i < 0$ for all i, then the equilibrium state is asymptotically stable. If for some i, $\Re\lambda_i > 0$, then the equilibrium is unstable. Finally, if $\Re\lambda_i \leq 0$ for all i with $\Re\lambda_i = 0$ for some i, linearization fails to determine asymptotic stability of the equilibrium of the nonlinear system. In this case, the stability of the nonlinear system depends on the nature of the higher-order terms.

Example 7.10 For the nonlinear system model,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1.1x_1 + g_2(x_2) \\ g_1(x_1) - 1.1x_2 \end{bmatrix}, \tag{7.73}$$

where

$$g_i(x_i) = \frac{2}{\pi}\arctan(0.7\pi x_i), \quad i = 1, 2,$$

we find its equilibrium states and determine their stability properties using the linearization method. We mention here that the above system was analyzed in [140, 241].

To find the system equilibrium states, we solve the algebraic equations $\dot{x}_1 = 0$ and $\dot{x}_2 = 0$, that is,

$$\left.\begin{array}{rcl} -1.1x_1 + \frac{2}{\pi}\arctan(0.7\pi x_2) & = & 0 \\ \frac{2}{\pi}\arctan(0.7\pi x_1) - 1.1x_2 & = & 0 \end{array}\right\}$$

We obtain three equilibrium states:

$$\boldsymbol{x}^{(1)} = \begin{bmatrix} 0.454 \\ 0.454 \end{bmatrix}, \quad \boldsymbol{x}^{(2)} = \begin{bmatrix} -0.454 \\ -0.454 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{x}^{(3)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Let

$$\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^{(i)}, \quad i = 1, 2, 3.$$

Then, a linearized model about an equilibrium state has the form

$$\frac{d}{dt}\Delta\boldsymbol{x} = \boldsymbol{A}\Delta\boldsymbol{x},$$

where

$$\boldsymbol{A}^{(i)} = \left.\begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_1(\boldsymbol{x})}{\partial x_2} \\ \frac{\partial f_2(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_2(\boldsymbol{x})}{\partial x_2} \end{bmatrix}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}} = \left.\begin{bmatrix} -1.1 & \frac{1.4}{1+(0.7\pi x_2)^2} \\ \frac{1.4}{1+(0.7\pi x_1)^2} & -1.1 \end{bmatrix}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}. \tag{7.74}$$

We have three linearized models:

(i) about $\boldsymbol{x}^{(1)}$,

$$\frac{d}{dt}\Delta\boldsymbol{x} = \begin{bmatrix} -1.1 & 0.7073 \\ 0.7073 & -1.1 \end{bmatrix}\Delta\boldsymbol{x},$$

(ii) about $\boldsymbol{x}^{(2)}$,

$$\frac{d}{dt}\Delta\boldsymbol{x} = \begin{bmatrix} -1.1 & 0.7073 \\ 0.7073 & -1.1 \end{bmatrix}\Delta\boldsymbol{x},$$

(iii) about $\boldsymbol{x}^{(3)}$,

$$\frac{d}{dt}\Delta\boldsymbol{x} = \begin{bmatrix} -1.1 & 1.4 \\ 1.4 & -1.1 \end{bmatrix}\Delta\boldsymbol{x}.$$

By Theorem 7.11, the states $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ are asymptotically stable because the corresponding linearized models have their eigenvalues in the open left-half plane: -1.8073 and -0.3927. By Theorem 7.12, the equilibrium state $\boldsymbol{x}^{(3)}$ is unstable because the linearized model has an eigenvalue in the open right-half plane, at 0.3 (the other eigenvalue is at -2.5). In Figure 7.13, we show a phase portrait of the above nonlinear system, where one can visually confirm the stability properties of the equilibrium states.

Example 7.11 In this example, we employ the Lyapunov's indirect method in the analysis of the hypothalamic-pituitary-adrenal (HPA) axis model presented in Section 2.4.8. Recall that the HPA axis is the body's primary stress management system. In our analysis of the HPA axis, we use plots of equilibrium values of the state variables as a function of the external stressor, d. These plots are shown in Figures 7.14 and 7.15.

In a chronically stressed individual, cortisol concentration, x_4, is very low. Thus a healthy individual subjected to a prolonged extreme stress, $d > 0.168$, would settle down in a stable

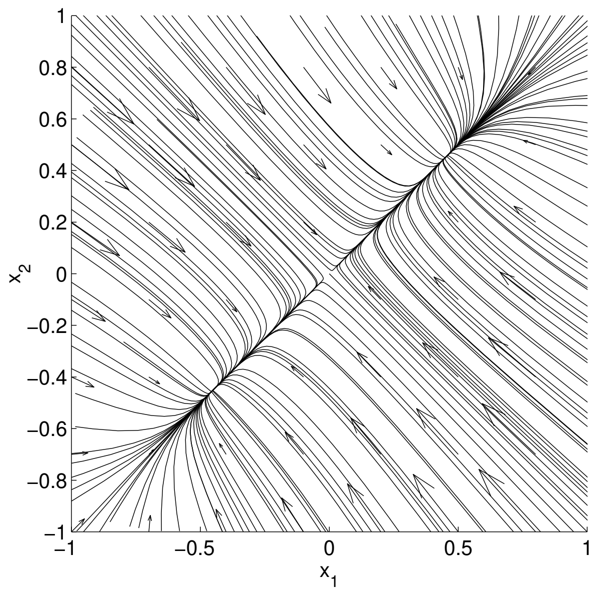


Figure 7.13: Phase portrait of the nonlinear system of Example (7.10).

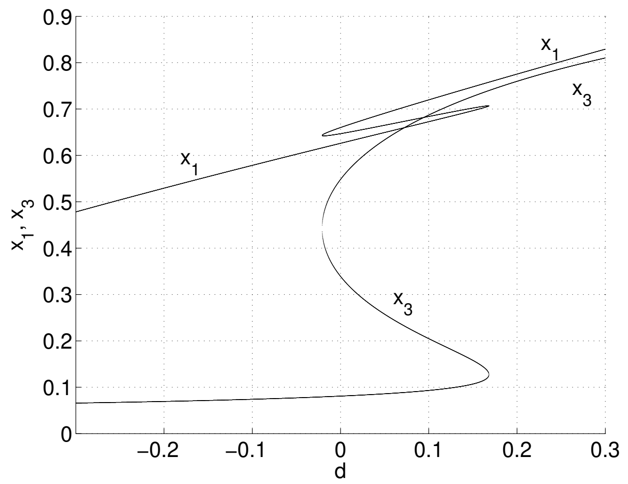


Figure 7.14: Plots of the equilibrium values of x_1 and x_3 versus d for $u = 0$.

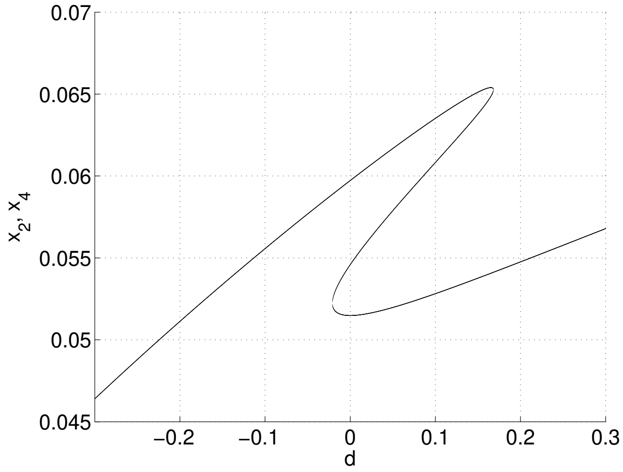


Figure 7.15: Plots of the equilbrium values of x_2 and x_4 versus d for $u = 0$.

equilibrium state corresponding to depressed cortisol concentration, x_4, corresponding to the lower branch of the curve in Figure 7.15. When the stress subsides, that is, $d = 0$, the individual will stay in the new equilibrium state corresponding to a depressed cortisol concentration. This is because the equilibrium corresponding to $d = 0$ is asymptotically stable and so states "close" to it are attracted by this asymptotically stable low cortisol equilibrium. We note that the middle branch in Figure 7.15, for the values of the stressor d where there are three equilibria, corresponds to unstable equilibria. As an illustration of the above statement, we show the equilibrium states corresponding to $d = 0.1$; we have three equilibrium states:

$$\boldsymbol{x}_e^{(1)} = \begin{bmatrix} 0.6727 \\ 0.0635 \\ 0.0931 \\ 0.0635 \end{bmatrix}, \quad \boldsymbol{x}_e^{(2)} = \begin{bmatrix} 0.6840 \\ 0.0608 \\ 0.2048 \\ 0.0608 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{x}_e^{(3)} = \begin{bmatrix} 0.7198 \\ 0.0528 \\ 0.6871 \\ 0.0528 \end{bmatrix}.$$

The corresponding linear models are;

$$\frac{d}{dt}\Delta\boldsymbol{x} = \boldsymbol{A}^{(i)}\Delta\boldsymbol{x} + \boldsymbol{b}_2^{(i)}\Delta d, \quad i = 1, 2, 3,$$

where

$$\boldsymbol{A}^{(1)} = \begin{bmatrix} -1.0000 & 0 & 0 & -4.1141 \\ 0.9442 & -10.0000 & -0.3809 & -0.5582 \\ 0 & 0 & -0.1988 & 1.0276 \\ 0 & 1.0000 & 0 & -1.0000 \end{bmatrix}, \quad \boldsymbol{b}_2^{(1)} = \begin{bmatrix} 0.6116 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\boldsymbol{A}^{(2)} = \begin{bmatrix} -1.0000 & 0 & 0 & -4.2530 \\ 0.8892 & -10.0000 & -0.3290 & -1.1077 \\ 0 & 0 & 0.2355 & 3.8235 \\ 0 & 1.0000 & 0 & -1.0000 \end{bmatrix}, \quad \boldsymbol{b}_2^{(2)} = \begin{bmatrix} 0.6218 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

and

$$\boldsymbol{A}^{(3)} = \begin{bmatrix} -1.0000 & 0 & 0 & -4.7104 \\ 0.7337 & -10.0000 & -0.2047 & -2.6627 \\ 0 & 0 & -0.1859 & 9.2897 \\ 0 & 1.0000 & 0 & -1.0000 \end{bmatrix}, \quad \boldsymbol{b}_2^{(3)} = \begin{bmatrix} 0.6544 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The eigenvalues of $\boldsymbol{A}^{(i)}$, $i = 1, 2, 3$, are:

$$\begin{aligned} \operatorname{eig}(\boldsymbol{A}^{(1)}) &= \{-9.990413, -0.989890, -0.989890, -0.228634\}, \\ \operatorname{eig}(\boldsymbol{A}^{(2)}) &= \{-9.937246, -0.992395, -0.992395, 0.157563\}, \\ \operatorname{eig}(\boldsymbol{A}^{(3)}) &= \{-9.763819, -1.051757, -1.051757, -0.318609\}. \end{aligned}$$

The second linear model is unstable because one of the eigenvalues of $\boldsymbol{A}^{(2)}$ is positive.

Ben-Zvi et al. [29] proposed a treatment strategy whereby the chronically stressed individual is moved to the healthy state corresponding to the normal cortisol concentration, by moving the individual into a negative stress region, after which internal regulatory processes translates the individual into the healthy equilibrium which is asymptotically stable and maintains it there. Effective treatment can be accomplished when the states as well as stress levels are available. Thus a method to estimate state variables as well as the stress level are highly desirable. This subject is discussed in Chapter 12.

7.10 Discontinuous Robust Controllers

In this section, we consider the problem of constructing stabilizing state-feedback controllers for nonlinear dynamical systems whose mathematical models contain uncertainties. The design of the controllers is based on the Lyapunov theory presented in the previous sections. The class of dynamical systems considered here is modeled by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)) + \boldsymbol{G}(t, \boldsymbol{x}(t)) \left(\boldsymbol{u}(t) + \boldsymbol{\xi}(t, \boldsymbol{x}(t)) \right), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{7.75}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$, and $\boldsymbol{\xi}(t, \boldsymbol{x}(t)) \in \mathbb{R}^m$ represents the lumped uncertainties. We assume that the uncertain element $\boldsymbol{\xi}$ is bounded by a known continuous non-negative function ρ, that is,

$$\|\boldsymbol{\xi}(t, \boldsymbol{x}(t))\| \leq \rho(t, \boldsymbol{x}(t)).$$

Stabilizing State-Feedback Controller

We assume that the equilibrium state $\boldsymbol{x} = \mathbf{0}$ of the uncontrolled nominal system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)) \tag{7.76}$$

is globally uniformly asymptotically stable, and that we were able to construct a Lyapunov function $W : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ corresponding to this equilibrium state. The Lyapunov function W is assumed to satisfy the following conditions: there exist three continuous and strictly increasing functions ψ_i, $i = 1, 2, 3$, such that

$$\left.\begin{array}{rcl} \psi_i(0) & = & 0, \quad i = 1, 2, 3, \\ \lim_{r\to\infty} \psi_1(r) & = & \infty, \\ \psi_1(\|\boldsymbol{x}\|) & \leq & W(t, \boldsymbol{x}) \leq \psi_2(\|\boldsymbol{x}\|), \\ \psi_3(\|\boldsymbol{x}\|) & \leq & -\dot{W}(t, \boldsymbol{x}) \end{array}\right\} \tag{7.77}$$

for all t and all $\boldsymbol{x}$. We show that if we apply the control strategy

$$\boldsymbol{u}(t, \boldsymbol{x}) = \begin{cases} -\frac{\boldsymbol{\mu}(t,\boldsymbol{x})}{\|\boldsymbol{\mu}(t,\boldsymbol{x})\|}\rho(t, \boldsymbol{x}) & \text{if} \quad (t, \boldsymbol{x}) \notin \Omega \\ \mathbf{0} & \text{if} \quad (t, \boldsymbol{x}) \in \Omega, \end{cases} \tag{7.78}$$

where $\boldsymbol{\mu}(t, \boldsymbol{x}) = \boldsymbol{G}^\top(t, \boldsymbol{x})\nabla_{\boldsymbol{x}} W(t, \boldsymbol{x})$, and $\Omega = \{(t, \boldsymbol{x}) : \boldsymbol{\mu} = \mathbf{0}\}$, then the equilibrium state $\boldsymbol{x} = \mathbf{0}$ is a globally uniformly asymptotically stable solution of the closed-loop system. For this, consider the Lyapunov function W for the uncontrolled nominal system. The time derivative of W evaluated on the solutions of the closed-loop system is

$$\begin{aligned} \dot{W}(t, \boldsymbol{x}) &= \frac{\partial W}{\partial t} + \nabla_{\boldsymbol{x}} W^\top \left(\boldsymbol{f}(t, \boldsymbol{x}) + \boldsymbol{G}(t, \boldsymbol{x})\left(\boldsymbol{u} + \boldsymbol{\xi}(t, \boldsymbol{x})\right)\right) \\ &\leq -\psi_3(\|\boldsymbol{x}\|) + \nabla_{\boldsymbol{x}} W^\top \boldsymbol{G}(t, \boldsymbol{x})\boldsymbol{u} + \nabla_{\boldsymbol{x}} W^\top \boldsymbol{G}(t, \boldsymbol{x})\boldsymbol{\xi}(t, \boldsymbol{x}) \\ &\leq -\psi_3(\|\boldsymbol{x}\|) + \boldsymbol{\mu}^\top(t, \boldsymbol{x})\boldsymbol{u} + \|\boldsymbol{\mu}(t, \boldsymbol{x})\|\rho(t, \boldsymbol{x}). \end{aligned} \tag{7.79}$$

For $(t, \boldsymbol{x}) \notin \Omega$,

$$\begin{aligned}
\dot{W}(t, \boldsymbol{x}) &\leq -\psi_3(\|\boldsymbol{x}\|) - \boldsymbol{\mu}^\top(t, \boldsymbol{x}) \frac{\boldsymbol{\mu}(t, \boldsymbol{x})}{\|\boldsymbol{\mu}(t, \boldsymbol{x})\|} \rho(t, \boldsymbol{x}) + \|\boldsymbol{\mu}(t, \boldsymbol{x})\| \rho(t, \boldsymbol{x}) \\
&= -\psi_3(\|\boldsymbol{x}\|) - \frac{\|\boldsymbol{\mu}(t, \boldsymbol{x})\|^2}{\|\boldsymbol{\mu}(t, \boldsymbol{x})\|} \rho(t, \boldsymbol{x}) + \|\boldsymbol{\mu}(t, \boldsymbol{x})\| \rho(t, \boldsymbol{x}) \\
&= -\psi_3(\|\boldsymbol{x}\|) - \|\boldsymbol{\mu}(t, \boldsymbol{x})\| \rho(t, \boldsymbol{x}) + \|\boldsymbol{\mu}(t, \boldsymbol{x})\| \rho(t, \boldsymbol{x}) \\
&= -\psi_3(\|\boldsymbol{x}\|) \\
&< 0. \qquad (7.80)
\end{aligned}$$

For $(t, \boldsymbol{x}) \in \Omega$, we have $\boldsymbol{\mu}(t, \boldsymbol{x}) = \mathbf{0}$, and using (7.79), we obtain

$$\dot{W}(t, \boldsymbol{x}) \leq -\psi_3(\|\boldsymbol{x}\|) < 0. \qquad (7.81)$$

Thus, W is also a Lyapunov function for the closed-loop system, which means that the uncertain system (7.75) driven by the control law (7.78) is indeed globally uniformly asymptotically stable.

Example 7.12 For the uncertain system model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\xi}_1(\boldsymbol{x}) + \boldsymbol{b}\left(u + \xi_2(\boldsymbol{x}, u)\right),$$

where

$$\boldsymbol{A} = \begin{bmatrix} -2 & 1 \\ 0 & -3 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \|\boldsymbol{\xi}_1(\boldsymbol{x})\| \leq 1.75\|\boldsymbol{x}\|, \quad \text{and} \quad |\xi_2(\boldsymbol{x}, u)| \leq 10\|\boldsymbol{x}\| + 0.5|u|,$$

we design stabilizing control strategy of the form (7.78).

The uncontrolled nominal system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ is asymptotically stable. Let

$$\boldsymbol{Q} = \begin{bmatrix} 4 & -1 \\ -1 & 6 \end{bmatrix}.$$

The eigenvalues of $\boldsymbol{Q}$ are $\lambda_{\min}(\boldsymbol{Q}) = 3.5858$ and $\lambda_{\max}(\boldsymbol{Q}) = 6.4142$. Solving the Lyapunov matrix equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$ yields

$$\boldsymbol{P} = \boldsymbol{I}_2.$$

The proposed controller has the form

$$u = -\eta(\boldsymbol{x}) \frac{\mu(\boldsymbol{x})}{|\mu(\boldsymbol{x})|} = -\eta(\boldsymbol{x}) \frac{\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}}{|\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}|} = -\eta(\boldsymbol{x}) \frac{\boldsymbol{b}^\top \boldsymbol{x}}{|\boldsymbol{b}^\top \boldsymbol{x}|} = -\eta(\boldsymbol{x}) \operatorname{sign}\left(\boldsymbol{b}^\top \boldsymbol{x}\right). \qquad (7.82)$$

Our goal is to determine a bound $\eta^*(\boldsymbol{x}) > 0$ such that for any $\eta(\boldsymbol{x}) \geq \eta^*(\boldsymbol{x})$ the uncertain closed-loop system is globally uniformly asymptotically stable. To begin, consider

$$V(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{x}$$

as the Lyapunov function candidate for the closed-loop system. Evaluating the time derivative of V on the trajectories of the closed-loop system yields

$$\begin{aligned}\frac{d}{dt}V(\boldsymbol{x}) &= 2\boldsymbol{x}^\top \dot{\boldsymbol{x}} \\ &= 2\boldsymbol{x}^\top \left(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\xi}_1(\boldsymbol{x}) + \boldsymbol{b}\left(u + \xi_2(\boldsymbol{x}, u)\right)\right) \\ &= -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{x}^\top \boldsymbol{\xi}_1(\boldsymbol{x}) + 2\boldsymbol{x}^\top \boldsymbol{b}u(\boldsymbol{x}) + 2\boldsymbol{x}^\top \boldsymbol{b}\xi_2(\boldsymbol{x}, u). \end{aligned} \tag{7.83}$$

Substituting into the above the expression for the controller given by (7.82), taking the norms, and using bounds on the uncertain elements, we obtain

$$\begin{aligned}\frac{d}{dt}V(\boldsymbol{x}) &\leq -\lambda_{\min}(\boldsymbol{Q})\|\boldsymbol{x}\|^2 + 2\|\boldsymbol{x}\|\|\boldsymbol{\xi}_1(\boldsymbol{x})\| - 2\eta(\boldsymbol{x})\frac{\left|\boldsymbol{x}^\top \boldsymbol{b}\right|^2}{|\boldsymbol{x}^\top \boldsymbol{b}|} + 2\left|\boldsymbol{x}^\top \boldsymbol{b}\right| |\xi_2(\boldsymbol{x}, u)| \\ &\leq (-3.5858 + 3.5)\|\boldsymbol{x}\|^2 - 2\eta(\boldsymbol{x})\left|\boldsymbol{x}^\top \boldsymbol{b}\right| + 2\left(10\|\boldsymbol{x}\| + 0.5\eta(\boldsymbol{x})\right)\left|\boldsymbol{x}^\top \boldsymbol{b}\right|. \end{aligned} \tag{7.84}$$

In the above, we used the fact that

$$|u| = \eta(\boldsymbol{x}).$$

Collecting the terms in (7.84) yields

$$\frac{d}{dt}V(\boldsymbol{x}) \leq (-3.5858 + 3.5)\|\boldsymbol{x}\|^2 + (20\|\boldsymbol{x}\| - \eta(\boldsymbol{x}))\left|\boldsymbol{x}^\top \boldsymbol{b}\right|. \tag{7.85}$$

Therefore, $\frac{dV}{dt} < 0$ for any

$$\eta(\boldsymbol{x}) \geq \eta^*(\boldsymbol{x}) = 20\|\boldsymbol{x}\|. \tag{7.86}$$

In summary, the uncertain closed-loop system will be globally uniformly asymptotically stable if the controller gain $\eta(\boldsymbol{x})$ is chosen to satisfy (7.86).

Model-Reference Tracking Controller

We now use the Lyapunov approach to design a model-reference tracking controller. A block diagram of the model-reference control system is shown in Figure 7.16. Model-reference control can be used for plants with nonlinearities and time-varying parameters. One of the

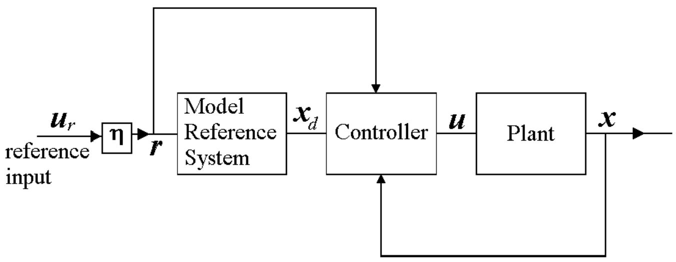


Figure 7.16: Model-reference control system block diagram with input scaling.

assumptions that is made while designing a model-reference controller is that a plant model is in a controller companion form. Here, we propose a method for designing model-reference controllers for a class of multi-input systems that does not require a plant model to be in a specific form. The crux of our method is in an appropriate selection of the model-reference system. Consider a plant model of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \tag{7.87}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times m}$. The model-reference signal generator is described by

$$\dot{\boldsymbol{x}}_d = \boldsymbol{A}_d\boldsymbol{x}_d + \boldsymbol{B}_d\boldsymbol{r} \tag{7.88}$$

where $\boldsymbol{A}_d \in \mathbb{R}^{n\times n}$, $\boldsymbol{B}_d \in \mathbb{R}^{n\times m}$, and $\boldsymbol{r} \in \mathbb{R}^m$ is a command signal that is chosen so that $\boldsymbol{x}_d$ represents a desired state trajectory of the plant. The matrix $\boldsymbol{A}_d$ of the reference system is chosen from the set

$$\left\{\boldsymbol{A}_d : \boldsymbol{A}_d = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_d, \boldsymbol{K}_d \in \mathbb{R}^{m\times n}\right\}, \tag{7.89}$$

and $\boldsymbol{B}_d = \boldsymbol{B}$. Thus, the reference system is state-feedback equivalent with the plant model (7.87). We assume that the pair $(\boldsymbol{A}, \boldsymbol{B})$ is controllable. Therefore, we can select $\boldsymbol{A}_d$ to have its eigenvalues in prescribed locations in the open-left hand plane. To proceed further, we define the tracking error $\boldsymbol{e}$ as

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_d(t). \tag{7.90}$$

Our goal is to construct a state-feedback control law so that

$$\lim_{t\to\infty} \|\boldsymbol{e}(t)\| = 0. \tag{7.91}$$

We form the differential equation that describes the error dynamics. For this we evaluate the time derivative of the error vector to get

$$\begin{aligned}\dot{\boldsymbol{e}}(t) &= \dot{\boldsymbol{x}}(t) - \dot{\boldsymbol{x}}_d(t)\\ &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} - \boldsymbol{A}_d\boldsymbol{x}_d - \boldsymbol{B}\boldsymbol{r}\\ &= \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}(\boldsymbol{u} - \boldsymbol{r}).\end{aligned} \tag{7.92}$$

Recall that $\boldsymbol{A}_d = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_d$, hence $\boldsymbol{A} = \boldsymbol{A}_d + \boldsymbol{B}\boldsymbol{K}_d$. Substituting this expression for $\boldsymbol{A}$ into the error equation (7.92), we obtain

$$\begin{aligned}\dot{\boldsymbol{e}}(t) &= (\boldsymbol{A}_d + \boldsymbol{B}\boldsymbol{K}_d)\boldsymbol{e} + \boldsymbol{B}(\boldsymbol{u} + -\boldsymbol{r})\\ &= \boldsymbol{A}_d\boldsymbol{e}(t) + \boldsymbol{B}(\boldsymbol{u} + \boldsymbol{K}_d\boldsymbol{x} - \boldsymbol{r}).\end{aligned} \tag{7.93}$$

We use a Lyapunov function approach in our further analysis that to obtain a control law forcing the error $\boldsymbol{e}(t)$ to go asymptotically to zero. As a Lyapunov function candidate for (7.93), we take

$$V(\boldsymbol{e}) = \boldsymbol{e}^\top \boldsymbol{P}\boldsymbol{e}, \tag{7.94}$$

where $\boldsymbol{P}$ solves the Lyapunov matrix equation, $\boldsymbol{A}_d^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_d = -\boldsymbol{Q}$, for some $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$. Because by construction $\boldsymbol{A}_d$ is asymptotically stable, the solution to the Lyapunov matrix equation is positive definite, that is, $\boldsymbol{P} = \boldsymbol{P}^\top > 0$. We next evaluate the time derivative of $V(\boldsymbol{e}(t))$ on the trajectories of (7.93),

$$\begin{aligned}\dot{V}(\boldsymbol{e}) &= 2\boldsymbol{e}^\top\boldsymbol{P}\dot{\boldsymbol{e}}\\ &= 2\boldsymbol{e}^\top\boldsymbol{P}(\boldsymbol{A}_d\boldsymbol{e} + \boldsymbol{B}(\boldsymbol{u} + \boldsymbol{K}_d\boldsymbol{x} - \boldsymbol{r}))\\ &= \boldsymbol{e}^\top(\boldsymbol{A}_d^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_d)\boldsymbol{e} + 2\boldsymbol{e}^\top\boldsymbol{P}\boldsymbol{B}(\boldsymbol{u} + \boldsymbol{K}_d\boldsymbol{x} - \boldsymbol{r}).\end{aligned} \tag{7.95}$$

Using the Lyapunov matrix equation, $\boldsymbol{A}_d^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_d = -\boldsymbol{Q}$, and taking the norms of the above yields

$$\dot{V}(\boldsymbol{e}) \le -\boldsymbol{e}^\top\boldsymbol{Q}\boldsymbol{e} + 2\boldsymbol{e}^\top\boldsymbol{P}\boldsymbol{B}\boldsymbol{u} + 2\left\|\boldsymbol{e}^\top\boldsymbol{P}\boldsymbol{B}\right\|\|\boldsymbol{K}_d\boldsymbol{x}\| + 2\left\|\boldsymbol{e}^\top\boldsymbol{P}\boldsymbol{B}\right\|\|\boldsymbol{r}\|. \tag{7.96}$$

If $\boldsymbol{e}$ is such that $\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{e} = \boldsymbol{0}$, then

$$\dot{V}(\boldsymbol{e}) \le -\boldsymbol{e}^\top\boldsymbol{Q}\boldsymbol{e} < 0. \tag{7.97}$$

We next analyze the case when $\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{e} \neq \boldsymbol{0}$. Let the proposed control law be

$$\boldsymbol{u} = -\rho(\boldsymbol{x}, \boldsymbol{r})\frac{\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{e}}{\left\|\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{e}\right\|}, \tag{7.98}$$

where $\rho(\boldsymbol{x}, \boldsymbol{r}) = \|\boldsymbol{K}_d\boldsymbol{x}\| + \|\boldsymbol{r}\|$. Substituting (7.98) into (7.96), we obtain

$$\begin{aligned} \dot{V}(\boldsymbol{e}) &\leq -\boldsymbol{e}^\top \boldsymbol{Q}\boldsymbol{e} - 2\left\|\boldsymbol{e}^\top \boldsymbol{P}\boldsymbol{B}\right\|(\rho(\boldsymbol{x}, r) - \|\boldsymbol{K}_d\boldsymbol{x}\| - \|\boldsymbol{r}\|) \\ &= -\boldsymbol{e}^\top \boldsymbol{Q}\boldsymbol{e} \\ &< 0. \end{aligned}$$

Since for all $\boldsymbol{e} \in \mathbb{R}^n$, we have $V(\boldsymbol{e}) > 0$ and $\dot{V}(\boldsymbol{e}) < 0$, the equilibrium point $\boldsymbol{e} = \boldsymbol{0}$ of the error equation (7.93) is globally uniformly asymptotically stable, which means that $\boldsymbol{x}(t) \to \boldsymbol{x}_d(t)$.

Example 7.13 We apply the above tracking controller to the vehicle model derived in the Subsection 2.4.2, where $\delta_r = 0$. That is, we wish to construct a tracking controller for a front wheel steered vehicle. The first step is to select the poles of the model reference system. We choose the model reference system to have the following set of asymptotically stable poles:

$$\{-4 - 2j, -4 + 2j, -4.5, -5.0\}. \tag{7.99}$$

The pair $(\boldsymbol{A}, \boldsymbol{b})$ is controllable, hence we can find $\boldsymbol{k}_d$ such that the eigenvalues of the matrix $\boldsymbol{A}_d = \boldsymbol{A} - \boldsymbol{b}\boldsymbol{k}_d$ are at the above selected locations. The feedback gain is

$$\boldsymbol{k}_d = \begin{bmatrix} 0.0329 & 2.2854 & 0.2429 & -0.1653 \end{bmatrix}, \tag{7.100}$$

and the resulting model reference plant matrix is

$$\boldsymbol{A}_d = \begin{bmatrix} -6.6638 & -107.1281 & -29.6350 & 7.7479 \\ 0 & 0 & 1 & 0 \\ -0.9205 & -65.8195 & -10.8362 & 4.7603 \\ -1 & -18.3 & 0 & 0 \end{bmatrix}. \tag{7.101}$$

The next step is to find $\boldsymbol{P}$ that solves the Lyapunov equation, $\boldsymbol{A}_d^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_d = -\boldsymbol{Q}$. We take $\boldsymbol{Q} = \boldsymbol{I}_4$. Solving the Lyapunov equation, we get

$$\boldsymbol{P} = \begin{bmatrix} 0.1145 & 0.7530 & -0.1809 & -0.0966 \\ 0.7530 & 34.6174 & -0.2883 & -3.3437 \\ -0.1809 & -0.2883 & 0.5144 & 0.0522 \\ -0.0966 & -3.3437 & 0.0522 & 0.6216 \end{bmatrix}. \tag{7.102}$$

Now that we have $\boldsymbol{P}$, we are ready to construct the controller. We have

$$
\begin{aligned}
u &= -\rho(\boldsymbol{x}, r)\frac{\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{e}}{|\,\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{e}\,|} \\
&= -(|\boldsymbol{k}_d\boldsymbol{x}| + |r|)\mathrm{sign}(\boldsymbol{b}^\top \boldsymbol{P}(\boldsymbol{x} - \boldsymbol{x}_d)) \\
&= -\left(\left|\begin{bmatrix} 0.0329 \\ 2.2854 \\ 0.2429 \\ -0.1653 \end{bmatrix}^\top \boldsymbol{x}\right| + |r|\right)\mathrm{sign}\left(\begin{bmatrix} 0.1571 \\ 26.9931 \\ 6.3324 \\ -3.0250 \end{bmatrix}^\top (\boldsymbol{x} - \boldsymbol{x}_d)\right).
\end{aligned}
\tag{7.103}
$$

Our goal is to have the reference input signal u_r to correspond to a desired vehicle path. This requires scaling the reference input. The scaling factor is

$$
\eta = \left(\lim_{s\to 0} \boldsymbol{c}(s\boldsymbol{I}_4 - \boldsymbol{A}_d)^{-1}\boldsymbol{b}\right)^{-1}, \tag{7.104}
$$

where $\boldsymbol{c} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$. Note that η is well defined because the matrix $\boldsymbol{A}_d$ is asymptotically stable. In our example, $\eta = -0.1653$. Thus, we have $r = \eta u_r$. The reference input is $u_r = 41(t - 1\,\mathrm{sec})$, where $1(t)$ is the unit step function.

The performance of the controller, for zero initial conditions, is shown in Figures 7.17 and 7.18. In Figure 7.17, the closed-loop system output is shown in the upper subplot, while the desired output is shown in the bottom subplot. The vehicle tracks the desired path with no noticeable steady-state error. The control action is shown in Figure 7.18.

The proposed tracking controller is discontinuous in the state, which results in its high activity referred to as chattering. Chattering can be reduced by using a continuous version of the above controller. The continuous controller is obtained by introducing a so-called boundary layer that smooths out the discontinuous controller. We discuss the problem of constructing boundary layer controllers in the following section.

7.11 Uniform Ultimate Boundedness

A drawback to the discontinuous controllers, discussed in the previous section, is that rapid switching tends to excite high frequency modes of the plant and this may degrade the performance of the controller. These problems can be alleviated by adding a so called *boundary*

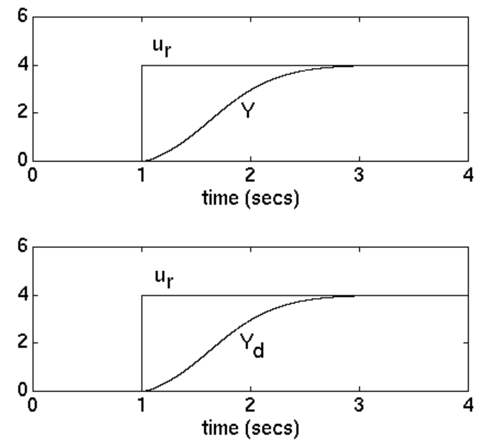


Figure 7.17: Plots of the actual output of the closed-loop system and the desired output in Example 7.13.

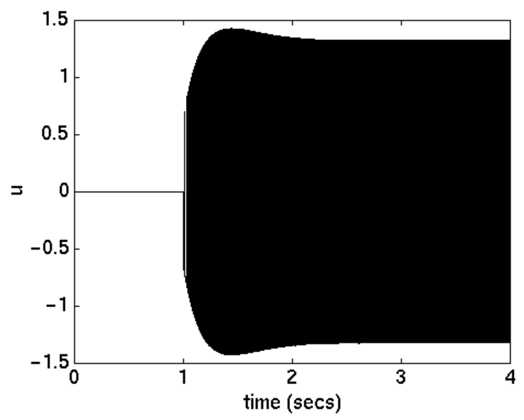


Figure 7.18: Plot of the controller action versus time in Example 7.13.

layer to the discontinuous controller resulting in a continuous control strategy. However, there is a price for approximating a discontinuous controller with a continuous one. The closed-loop system driven by a boundary layer controller may only be *uniformly ultimately bounded* rather than asymptotically stable. The notions we are concerned with in this section are described in the following three definitions.

Definition 7.18 *The closed ball $\bar{B}_q(\mathbf{0}) = \{\boldsymbol{x} : \|\boldsymbol{x}\| \leq q\}$ is uniformly stable if for any $\varepsilon > q$, there is a $\delta = \delta(\varepsilon) > 0$ so that if $\|\boldsymbol{x}(t_0)\| \leq \delta$, then $\|\boldsymbol{x}(t)\| \leq \varepsilon$ for $t \geq t_0$.*

◇

For the benefit of the reader, we now re-phrase Definition 7.16 of uniform boundedness from Section 7.8.

Definition 7.19 *The solutions of $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$ are uniformly bounded if for any given $d > 0$, there exists $b = b(d) < \infty$ so that for each solution $\boldsymbol{x}(t) = \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$ with $\|\boldsymbol{x}_0\| \leq d$, we have $\|\boldsymbol{x}(t)\| \leq b$ for all $t \geq t_0$.*

◇

We next define the notion of uniform ultimate boundedness (UUB) of a system.

Definition 7.20 *The solutions of $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$ are uniformly ultimately bounded (UUB) with respect to the ball $\bar{B}_q(\mathbf{0})$ if for every $d > 0$, there exists $T(q, d)$ such that for each solution $\boldsymbol{x}(t) = \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$ with $\|\boldsymbol{x}_0\| \leq d$, we have $\boldsymbol{x}(t) \in \bar{B}_q(\mathbf{0})$ for $t \geq t_0 + T(q, d)$.*

◇

Recall a dynamical system model (7.75),

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)) + \boldsymbol{G}(t, \boldsymbol{x}(t)) \left(\boldsymbol{u}(t) + \boldsymbol{\xi}(t, \boldsymbol{x}(t))\right), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0,$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$, and $\boldsymbol{\xi}(t, \boldsymbol{x}(t)) \in \mathbb{R}^m$ represents the lumped uncertainties. We assume that the uncertain element $\boldsymbol{\xi}$ is bounded by a known continuous non-negative function ρ, that is, $\|\boldsymbol{\xi}(t, \boldsymbol{x}(t))\| \leq \rho(t, \boldsymbol{x}(t))$. As in Section 7.10, we assume that the equilibrium state $\boldsymbol{x} = \mathbf{0}$ of the uncontrolled nominal system, $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t))$, is globally uniformly asymptotically stable, and that we were able to construct a Lyapunov function $W : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ satisfying the conditions (7.77) for all t and all $\boldsymbol{x}$.

We showed in the previous section that if we apply the discontinuous control strategy

$$\boldsymbol{u}(t, \boldsymbol{x}) = \begin{cases} -\frac{\boldsymbol{\mu}(t, \boldsymbol{x})}{\|\boldsymbol{\mu}(t, \boldsymbol{x})\|} \rho(t, \boldsymbol{x}) & \text{if} \quad \boldsymbol{\mu}(t, \boldsymbol{x}) \neq \mathbf{0} \\ \mathbf{0} & \text{if} \quad \boldsymbol{\mu}(t, \boldsymbol{x}) = \mathbf{0}, \end{cases}$$

where $\boldsymbol{\mu}(t,\boldsymbol{x}) = \boldsymbol{G}(t,\boldsymbol{x})^\top \nabla_{\boldsymbol{x}} W(t,\boldsymbol{x})$, then the equilibrium state $\boldsymbol{x} = \boldsymbol{0}$ is a globally uniformly asymptotically stable solution of the closed-loop system. However, the above controller is discontinuous in $\boldsymbol{x}$. Following Corless and Leitmann [69], we approximate it with a continuous controller by introducing a boundary layer. The proposed boundary layer controller is

$$\boldsymbol{u}(t,\boldsymbol{x}) = \begin{cases} -\frac{\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})}{\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\|}\rho(t,\boldsymbol{x}) & \text{if } \ \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| > \nu \\ \boldsymbol{p}(t,\boldsymbol{x}), \ \|\boldsymbol{p}(t,\boldsymbol{x})\| \le \rho(t,\boldsymbol{x}) & \text{if } \ \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| \le \nu, \end{cases} \tag{7.105}$$

where ν is the thickness of the boundary layer about the switching surface $\{(t,\boldsymbol{x}) : \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| = 0\}$, and $\boldsymbol{p}$ is a continuous function such that

$$\boldsymbol{p} = -\frac{\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})}{\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\|}\rho(t,\boldsymbol{x}) \quad \text{if} \quad \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| = \nu.$$

A particular example of the above type of a controller is

$$\boldsymbol{u}(t,\boldsymbol{x}) = \begin{cases} -\frac{\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})}{\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\|}\rho(t,\boldsymbol{x}) & \text{if } \ \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| > \nu \\ -\frac{\boldsymbol{\mu}(t,\boldsymbol{x})}{\nu}\rho^2(t,\boldsymbol{x}) & \text{if } \ \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| \le \nu. \end{cases} \tag{7.106}$$

We show that the closed-loop system driven by the boundary layer controller (7.105) is UUB with respect to a neighborhood of the zero state. This neighborhood can be made arbitrarily small by letting $\nu \to 0$. Before we state the results, we first compute the time derivative of a Lyapunov function candidate on the trajectories of the closed-loop system. Consider the Lyapunov function W for the uncontrolled nominal system. The time derivative of W evaluated on the solutions of the closed-loop system is

$$\begin{aligned} \dot{W}(t,\boldsymbol{x}) &= \frac{\partial W}{\partial t} + \nabla_{\boldsymbol{x}} W^\top \left(\boldsymbol{f}(t,\boldsymbol{x}) + \boldsymbol{G}(t,\boldsymbol{x})\left(\boldsymbol{u} + \boldsymbol{\xi}(t,\boldsymbol{x})\right)\right) \\ &\le -\psi_3(\|\boldsymbol{x}\|) + \nabla_{\boldsymbol{x}} W^\top \boldsymbol{G}(t,\boldsymbol{x})\boldsymbol{u} + \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\|. \end{aligned}$$

We now consider two cases. The first case is when $\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| > \nu$. Taking into account the controller structure for this case yields

$$\begin{aligned} \dot{W}(t,\boldsymbol{x}) &\le -\psi_3(\|\boldsymbol{x}\|) - \nabla_{\boldsymbol{x}} W^\top \boldsymbol{G}(t,\boldsymbol{x}) \frac{\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})}{\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\|}\rho(t,\boldsymbol{x}) + \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| \\ &= -\psi_3(\|\boldsymbol{x}\|) - \frac{\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\|^2}{\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\|} + \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| \\ &= -\psi_3(\|\boldsymbol{x}\|). \end{aligned}$$

If, on the other hand, $\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| \leq \nu$, then using (7.105), we obtain

$$\begin{aligned}\dot{W}(t,\boldsymbol{x}) &\leq -\psi_3(\|\boldsymbol{x}\|) + \nabla_{\boldsymbol{x}} W^\top \boldsymbol{G}(t,\boldsymbol{x})\boldsymbol{u} + \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| \\ &\leq -\psi_3(\|\boldsymbol{x}\|) + \|\nabla_{\boldsymbol{x}} W^\top \boldsymbol{G}(t,\boldsymbol{x})\|\,\|\boldsymbol{u}\| + \|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| \\ &\leq -\psi_3(\|\boldsymbol{x}\|) + 2\|\boldsymbol{\mu}(t,\boldsymbol{x})\rho(t,\boldsymbol{x})\| \\ &\leq -\psi_3(\|\boldsymbol{x}\|) + 2\nu.\end{aligned}$$

We conclude that for all t and all $\boldsymbol{x}$, we have

$$\boxed{\dot{W}(t,\boldsymbol{x}) \leq -\psi_3(\|\boldsymbol{x}\|) + 2\nu} \tag{7.107}$$

To proceed further, we need an assumption on the function ψ_3; namely if ψ_3 is bounded and

$$\lim_{r\to\infty} \psi_3(r) = l < \infty, \tag{7.108}$$

then we assume

$$2\nu < l. \tag{7.109}$$

Proposition 7.2 *The solutions of the system (7.75) driven by the boundary layer controller (7.105) are uniformly bounded.*

Proof Let R be such that

$$\psi_3(R) = 2\nu.$$

Note that R is well defined by virtue of (7.109) and the fact that ψ_3 is strictly increasing. Because ψ_3 is strictly increasing, we can write

$$R = \psi_3^{-1}(2\nu).$$

Figure 7.19 illustrates the above construction.

For a given $d > 0$, where $\|\boldsymbol{x}_0\| \leq d$, let

$$\tilde{d} = \max\{d, R\}.$$

We claim that

$$b = b(d) = \psi_1^{-1}(\psi_2(\tilde{d})),$$

that is, the solutions of (7.75) driven by the boundary layer controller (7.105) are uniformly bounded. In other words, if $\|\boldsymbol{x}_0\| \leq d$ then $\|\boldsymbol{x}(t;t_0,\boldsymbol{x}_0)\| \leq b$ for $t \geq t_0$. First note that

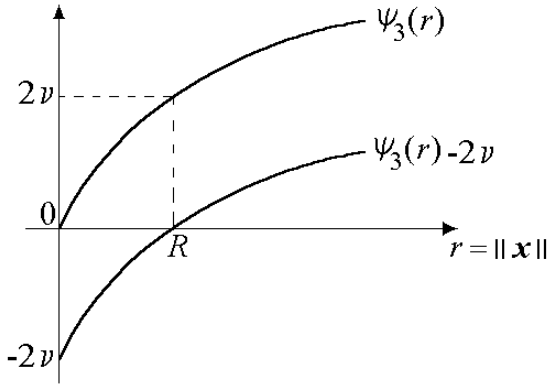


Figure 7.19: Finding R for a given ν in the proof of Proposition 7.2.

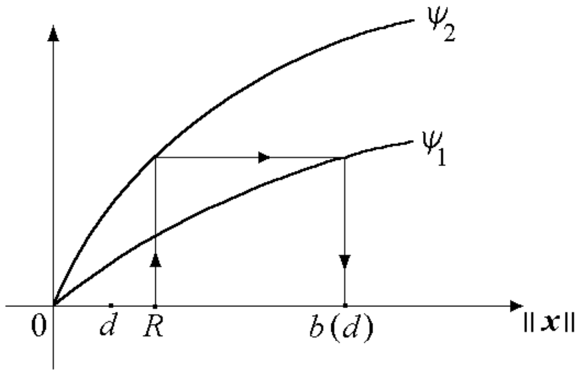


Figure 7.20: Finding b for a given d in the proof of Proposition 7.2.

$\|\boldsymbol{x}_0\| \leq \tilde{d}$ and that $\psi_1(b) = \psi_2(\tilde{d})$. Hence, $\tilde{d} \leq b$—see Figure 7.20 for an illustration of this construction, and thus

$$\begin{aligned} \dot{W}(t, \boldsymbol{x}(t)) &\leq -\psi_3(\|\boldsymbol{x}(t)\|) + 2\nu \\ &= -\psi_3(\|\boldsymbol{x}(t)\|) + \psi_3(R). \end{aligned} \tag{7.110}$$

Suppose first that $d > R$. Then for $\|\boldsymbol{x}_0\| > R$ and $\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| > R$, $\dot{W} < 0$ and therefore

$$\begin{aligned} \psi_1(b) &= \psi_2(\tilde{d}) \\ &\geq W(t_0, \boldsymbol{x}(t_0; t_0, \boldsymbol{x}_0)) \\ &\geq W(t, \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)) \\ &\geq \psi_1(\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\|), \end{aligned}$$

which implies that $\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| \leq b$ for $t \geq t_0$.

If, on the other hand, $\|\boldsymbol{x}_0\| \leq R$, then because $\dot{W}(t, \boldsymbol{x}(t)) \leq -\psi_3(\|\boldsymbol{x}(t)\|) + 2\nu$, the solution may reach the sphere $\|\boldsymbol{x}\| = R$ at some time $t_2 \geq t_0$. However, by (7.110), $\dot{W} < 0$ outside $\|\boldsymbol{x}\| = R$. Therefore, $\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| \leq b$. Thus the solution is uniformly bounded.

□

We illustrate Proposition 7.2 with Figure 7.21. The initial condition in Figure 7.21 is such that $\|\boldsymbol{x}_0\| < d$ and $d < R$. The trajectory is confined to stay within the ball of radius $b = \psi_1^{-1}(\psi_2(R))$.

We now show, using arguments similar to those of Corless and Leitmann [69], that the solutions of the closed-loop system are UUB with respect to any ball of radius $\tilde{b} > \psi_1^{-1}(\psi_2(R))$.

Theorem 7.13 *Let $\boldsymbol{x}(t) = \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$ be a solution of the closed-loop system for $[t_0, \infty)$ such that (7.107) is satisfied. If $\|\boldsymbol{x}_0\| \leq d$, then for a given*

$$\tilde{b} > \psi_1^{-1}(\psi_2(R)), \tag{7.111}$$

$\|\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\| \leq \tilde{b}$ for all $t \geq t_0 + T(\tilde{b}, d)$, where

$$T(\tilde{b}, d) = \begin{cases} 0 & \text{if} \quad d \leq \tilde{R} \\ \frac{\psi_2(d) - \psi_1(\tilde{R})}{\psi_3(\tilde{R}) - 2\nu} & \text{if} \quad d > \tilde{R}, \end{cases}$$

and

$$\psi_2\left(\tilde{R}\right) = \psi_1(\tilde{b}). \tag{7.112}$$

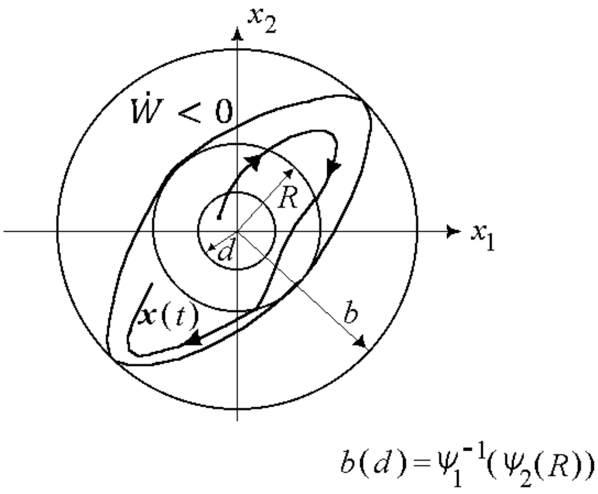


Figure 7.21: A two-dimensional illustration of an uniformly ultimately bounded (UUB) solution.

Proof From assumption (7.111),

$$\psi_1(\tilde{b}) > \psi_2(R),$$

and by (7.112),

$$\psi_1(\tilde{b}) = \psi_2(\tilde{R}).$$

Hence, by (7.77), where we assumed that ψ_2 is strictly increasing,

$$\tilde{R} > R. \tag{7.113}$$

We consider two cases. The first case is when $d \leq \tilde{R}$. By uniform boundedness, for all $t \geq t_0$,

$$\|\boldsymbol{x}(t)\| \leq \tilde{b},$$

for any $\tilde{b}$ such that $\psi_1(\tilde{b}) > \psi_2(\tilde{R})$. Hence in this case $T(\tilde{b}, d) = 0$.

We now consider the case when $d > \tilde{R}$. We show, by contradiction, that for some $t_1 \in [t_0, t_0 + T(\tilde{b}, d)]$,

$$\|\boldsymbol{x}(t_1)\| \leq \tilde{R}.$$

From this we shall conclude the result by referring to the previous case. So suppose that $\|\boldsymbol{x}(t)\| > \tilde{R}$ for all $t \in \left[t_0, t_0 + T(\tilde{b}, d)\right]$. Then,

$$\begin{aligned}
\psi_1\left(\left\|\boldsymbol{x}\left(t_0 + T(\tilde{b}, d); t_0, \boldsymbol{x}_0\right)\right\|\right) &\leq W\left(t_0 + T(\tilde{b}, d), \boldsymbol{x}\left(t_0 + T(\tilde{b}, d); t_0, \boldsymbol{x}_0\right)\right) \\
&= W(t_0, \boldsymbol{x}_0) + \int_{t_0}^{t_0 + T(\tilde{b}, d)} \dot{W} dt \\
&\leq W(t_0, \boldsymbol{x}_0) + \int_{t_0}^{t_0 + T(\tilde{b}, d)} \left(-\psi_3(\tilde{R}) + 2\nu\right) dt \\
&\leq \psi_2(d) - \int_{t_0}^{t_0 + T(\tilde{b}, d)} \left(\psi_3(\tilde{R}) - 2\nu\right) dt.
\end{aligned}$$

Using the fact that $\psi_3(\tilde{R}) - 2\nu > 0$ because by (7.113), $\tilde{R} > R$, we obtain

$$\begin{aligned}
\psi_1\left(\left\|\boldsymbol{x}\left(t_0 + T(\tilde{b}, d); t_0, \boldsymbol{x}_0\right)\right\|\right) &\leq \psi_2(d) - \left(\psi_3(\tilde{R}) - 2\nu\right) T(\tilde{b}, d) \\
&= \psi_2(d) - \left(\psi_3(\tilde{R}) - 2\nu\right) \frac{\psi_2(d) - \psi_1(\tilde{R})}{\psi_3(\tilde{R}) - 2\nu} \\
&= \psi_1\left(\tilde{R}\right).
\end{aligned}$$

Thus, because ψ_1 is strictly increasing,

$$\left\|\boldsymbol{x}\left(t_0 + T(\tilde{b}, d); t_0, \boldsymbol{x}_0\right)\right\| \leq \tilde{R},$$

which contradicts our assumption that for all $t \in \left[t_0, t_0 + T(\tilde{b}, d)\right]$ it should have been $\|\boldsymbol{x}(t)\| > \tilde{R}$. Hence, there exists $t_1 \in \left[t_0, t_0 + T(\tilde{b}, d)\right]$ such that $\|\boldsymbol{x}(t_1)\| \leq \tilde{R}$. Then using the same argument as in the previous case we can show that

$$\|\boldsymbol{x}(t)\| \leq \tilde{b}$$

for all $t \geq t_1$ and the proof is complete.

□

We have the following consequence of the above theorem.

Corollary 7.2 *The ball*

$$\left\{\boldsymbol{x} : \|\boldsymbol{x}\| \leq \psi_1^{-1}(\psi_2(R))\right\}$$

is uniformly stable.

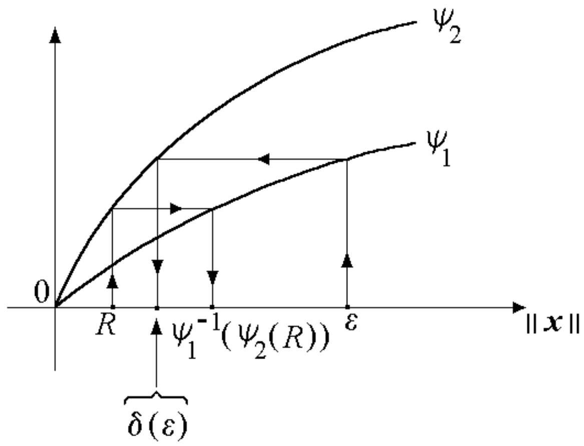


Figure 7.22: Finding δ given ε for an uniformly stable ball in the proof of Corollary 7.2.

Proof Let

$$\varepsilon > \psi_1^{-1}(\psi_2(R)),$$

and consider $\delta = \delta(\varepsilon)$ obtained from

$$\psi_1(\varepsilon) = \psi_2(\delta).$$

Obviously $\delta \geq R$. Figure 7.22 illustrates the above relations. Hence, by uniform boundedness if $\|\boldsymbol{x}_0\| \leq \delta$, then $\|\boldsymbol{x}(t)\| \leq \varepsilon$ for all $t \geq t_0$ which means that the ball is uniformly stable.

$\square$

Example 7.14 Consider a dynamical system model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}(u + h(t, \boldsymbol{x})),$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \text{and} \quad |h(t, \boldsymbol{x})| \leq 2.$$

Let $\boldsymbol{P}$ be the solution of the matrix Lyapunov equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{I}_2$, and let

$$u = \begin{cases} -2\,\mathrm{sign}(\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}) & \text{if} \quad |\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}| \geq \nu \\ -2\dfrac{\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}}{\nu} & \text{if} \quad |\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}| < \nu, \end{cases}$$

where $\nu = 1$. Suppose that the initial condition, $\boldsymbol{x}(0)$, is such that $\|\boldsymbol{x}(0)\|_2 \leq 10$. Find the radius, b, of a ball such that $\|\boldsymbol{x}(t)\|_2 \leq b$ for all $t \geq 0$.

Solving the Lyapunov equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{I}_2$ yields

$$\boldsymbol{P} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}.$$

We next evaluate $\frac{d}{dt}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}\right)$ on the trajectories of the closed-loop system. For the case when $\left|\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}\right| \geq \nu$, we get

$$\frac{d}{dt}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}\right) \leq -2\|\boldsymbol{x}\|^2.$$

For the case when $\left|\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}\right| < \nu$, we obtain

$$\begin{aligned} \frac{d}{dt}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}\right) &= -2\|\boldsymbol{x}\|^2 + 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}(u+h) \\ &\leq -2\|\boldsymbol{x}\|^2 - 4\frac{\left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right|^2}{\nu} + 4\left|\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\right| \\ &\leq -2\|\boldsymbol{x}\|^2 + 4\nu \\ &= -2\|\boldsymbol{x}\|^2 + 4. \end{aligned}$$

Hence, $\frac{d}{dt}\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}\right) < 0$ outside of the ball

$$\bar{B}_R(\boldsymbol{0}) = \{\boldsymbol{x} : \|\boldsymbol{x}\| \leq R = \sqrt{2}\}.$$

To find the radius, b, of a ball, such that $\|\boldsymbol{x}(t)\|_2 \leq b$ for all $t \geq 0$ and $\|\boldsymbol{x}(0)\|_2 \leq 10 = d$, we first observe that

$$\lambda_{\min}(\boldsymbol{P})\|\boldsymbol{x}\|^2 = \psi_1(\|\boldsymbol{x}\|) \leq \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} \leq \psi_2(\|\boldsymbol{x}\|) = \lambda_{\max}(\boldsymbol{P})\|\boldsymbol{x}\|^2.$$

Let $\tilde{d} = \max\{d, R\} = 10$. Then, the radius b is computed by solving the equation

$$\psi_1(b) = \psi_2(\tilde{d}),$$

that is,

$$\lambda_{\min}(\boldsymbol{P})b^2 = \lambda_{\max}(\boldsymbol{P})\tilde{d}^2.$$

We obtain

$$b = \tilde{d}\sqrt{\frac{\lambda_{\max}(\boldsymbol{P})}{\lambda_{\min}(\boldsymbol{P})}} = 10\sqrt{\frac{(5+\sqrt{5})/2}{(5-\sqrt{5})/2}} = 16.18.$$

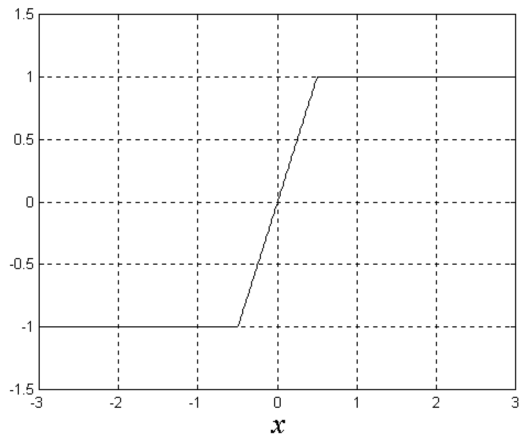


Figure 7.23: Plot of the linear saturation function described by (7.115).

Example 7.15 In this example, we investigate the performance of a boundary layer tracking controller on the tracking problem considered in Example 7.13. Specifically, we approximate the discontinuous controller given by (7.103) with a boundary layer continuous controller obtained by replacing the sign nonlinearity with a linear saturation nonlinearity. A linear saturation nonlinearity, denoted sat_ν, can be described as

$$\text{sat}_\nu(x) = \frac{1}{2\nu}\left(|x+\nu| - |x-\nu|\right), \tag{7.114}$$

where the parameter ν controls the width of the boundary layer. In other words,

$$\text{sat}_\nu(x) = \begin{cases} 1 & \text{if} \quad x \geq \nu \\ \frac{1}{\nu}x & \text{if} \quad -\nu < x < \nu \\ -1 & \text{if} \quad x \leq -\nu. \end{cases} \tag{7.115}$$

A plot of (7.114) for $\nu = 0.5$ is shown in Figure 7.23. A boundary layer approximation of

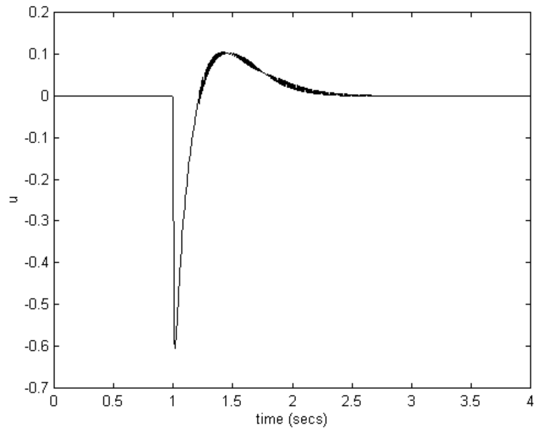


Figure 7.24: Plot of the boundary layer controller action versus time in Example 7.15.

the discontinuous tracking controller (7.103) is

$$\begin{aligned} u &= -\rho(\boldsymbol{x}, r)\mathrm{sat}_\nu\left(\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{e}\right) \\ &= -(|\boldsymbol{k}_d\boldsymbol{x}| + |r|)\mathrm{sat}_\nu(\boldsymbol{b}^\top \boldsymbol{P}(\boldsymbol{x} - \boldsymbol{x}_d)) \\ &= -\left(\left|\begin{bmatrix} 0.0329 \\ 2.2854 \\ 0.2429 \\ -0.1653 \end{bmatrix}^\top \boldsymbol{x}\right| + |r|\right)\mathrm{sat}_\nu\left(\begin{bmatrix} 0.1571 \\ 26.9931 \\ 6.3324 \\ -3.0250 \end{bmatrix}^\top (\boldsymbol{x} - \boldsymbol{x}_d)\right). \end{aligned} \tag{7.116}$$

The plot of the actual output, Y, of the closed-loop system driven by (7.116) for $\nu = 0.5$ and zero initial conditions was indistinguishable from that shown in Figure 7.17 when the discontinuous controller was used. The boundary layer controller action versus time is shown in Figure 7.24. Chattering is basically eliminated in this case.

7.12 Lyapunov Like Analysis

Usually, it is challenging to analyze the asymptotic stability of time-varying systems because it is very difficult to find Lyapunov functions with negative definite Lyapunov derivatives.

One may ask the following question: when can we conclude asymptotic stability when the Lyapunov derivative is only negative semi-definite?

In this section, we present a number of results that attempt to answer our question. In addition these results are very useful in the analysis and design of adaptive controllers. To proceed, we review some properties of time-varying functions. In particular, we need a condition that guarantees that $\dot{f}(t) \to 0$ as $t \to \infty$.

We present some basic properties of a function $f : \mathbb{R} \to \mathbb{R}$ in the following list,

- Recall that $\dot{f}(t) \to 0$ does not imply that $f(t)$ has a limit as $t \to \infty$. On the other hand, $f(t)$ has a limit as $t \to \infty$ does not imply that $\dot{f}(t) \to 0$.

- A well-known result from calculus states that if $f(t)$ is lower bounded and decreasing, that is, $\dot{f}(t) \leq 0$, then it converges to a limit. But we do not know whether $\dot{f}(t) \to 0$ or not as $t \to \infty$.

- Recall also that a function $f(t)$ is continuous if small changes in t result in small changes in its value $f(t)$. Intuitively, a continuous function is a function whose graph can be drawn without lifting the pencil from the paper.

- Continuity is a local property of a function, that is, a function f is continuous, or not, at a particular point t.

- A function being continuous on an interval means only that it is continuous at each point of the interval.

- A function f is uniformly continuous if it is continuous and, in addition, the size of the changes in $f(t)$ depends only on the size of the changes in t but not on t itself.

- The slope of an uniformly continuous function is bounded, that is, $\dot{f}$ is bounded.

- Uniform continuity is a global property of a function.

- Every uniformly continuous function is continuous, but the converse is not true.

- A function is uniformly continuous, or not, on an entire interval.

- A function may be continuous at each point of an interval without being uniformly continuous on the entire interval.

Example 7.16 The function $f(t) = \sin(t)$ is uniformly continuous. Note that the slope of the above function is bounded.

Example 7.17 The function $f(t) = 1/t$ is continuous but not uniformly continuous on positive real numbers. Note that as t approaches 0, the changes in $f(t)$ become unbounded.

In the next example from Slotine and Li [301, p. 124], we show that in an asymptotically stable LTI system with bounded input, the output is uniformly continuous.

Example 7.18 We consider an asymptotically stable LTI system

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{u}\\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}.\end{aligned}$$

The state $\boldsymbol{x}(t)$ is bounded because $\boldsymbol{u}$ is bounded and $\boldsymbol{A}$ is asymptotically stable. Because $\boldsymbol{x}$ is bounded and $\boldsymbol{u}$ is bounded, we have that $\dot{\boldsymbol{x}}$ is bounded as well. We now consider the time derivative of the output,

$$\dot{\boldsymbol{y}} = \boldsymbol{C}\dot{\boldsymbol{x}}.$$

The time derivative of $\boldsymbol{y}$ is bounded because $\dot{\boldsymbol{x}}$ is bounded. Hence, $\boldsymbol{y}$ is uniformly continuous.

With the above background, we are now ready to state and prove the following lemma.

Lemma 7.4 (Barbalat) *Given a differentiable function $f = f(t) : \mathbb{R} \to \mathbb{R}$ such that $\dot{f}$ is uniformly continuous. If $\lim_{t\to\infty} f(t)$ exists, then*

$$\lim_{t\to\infty} \dot{f}(t) = 0.$$

Proof We prove the lemma by contraposition. We assume that $\dot{f}(t)$ does not converge to 0 as $t \to \infty$. Therefore, there exists an $\varepsilon > 0$ such that for every $T > 0$ we can find $t_i \geq T$ so that $|\dot{f}(t_i)| \geq \varepsilon$. By assumption $\dot{f}$ is uniformly continuous. Hence, given $\varepsilon/2 > 0$ there is $\delta_{\varepsilon/2} > 0$ so that for every t_i and every $\tau \in \left[0, \delta_{\varepsilon/2}\right]$, we have

$$\left|\dot{f}(t_i + \tau) - \dot{f}(t_i)\right| < \frac{\varepsilon}{2}. \tag{7.117}$$

Hence, for all $t \in \left[t_i, t_i + \delta_{\varepsilon/2}\right]$, we have

$$\begin{aligned}
\left|\dot{f}(t)\right| &= \left|\dot{f}(t) - \dot{f}(t_i) + \dot{f}(t_i)\right| \\
&\geq \left|\dot{f}(t_i)\right| - \left|\dot{f}(t) - \dot{f}(t_i)\right| \\
&> \varepsilon - \frac{\varepsilon}{2} \\
&= \frac{\varepsilon}{2}.
\end{aligned}$$

Suppose that $\dot{f}(t_i) > 0$. (The other case is when $\dot{f}(t_i) < 0$ and its analysis is similar to the case we analyze now.) It follows from the above that for any $t \in \left[t_i, t_i + \delta_{\varepsilon/2}\right]$,

$$\dot{f}(t) > \frac{\varepsilon}{2}.$$

Using the above, we obtain

$$\int_{t_i}^{t_i+\delta_{\varepsilon/2}} \dot{f}(t)dt > \int_{t_i}^{t_i+\delta_{\varepsilon/2}} \frac{\varepsilon}{2}dt = \frac{\varepsilon}{2}\delta_{\varepsilon/2}. \tag{7.118}$$

However,

$$\int_{t_i}^{t_i+\delta_{\varepsilon/2}} \dot{f}(t)dt = f(t_i + \delta_{\varepsilon/2}) - f(t_i). \tag{7.119}$$

Combining (7.118) and(7.119) yields

$$f(t_i + \delta_{\varepsilon/2}) - f(t_i) > \frac{\varepsilon}{2}\delta_{\varepsilon/2} \quad \text{for every} \quad i,$$

which contradicts the assumption that $f(t)$ has a finite limit as $t \to \infty$. The proof of the lemma is complete.

□

A simple sufficient condition for a differentiable function to be uniformly continuous is boundedness of its derivative. Thus if a function f is twice differentiable, then its derivative $\dot{f}$ is uniformly continuous if its second derivative $\ddot{f}$ is bounded.

The following lemma, called by Slotine and Li [301, page 125] the "Lyapunov-like lemma", follows immediately from Barbalat's lemma.

Lemma 7.5 ("Lyapunov-Like Lemma") *Given a real-valued function $W(t, \boldsymbol{x})$ such that*

1. *$W(t, \boldsymbol{x})$ is bounded below;*

2. *$\dot{W}(t, \boldsymbol{x})$ is negative semi-definite;*

3. *$\dot{W}(t, \boldsymbol{x})$ is uniformly continuous in time,*

then

$$\dot{W}(t, \boldsymbol{x}) \to 0 \quad as \quad t \to \infty.$$

Example 7.19 We investigate stability of the following dynamical system model:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2x_1 + x_2 u \\ -x_1 u \end{bmatrix}, \tag{7.120}$$

where the input u is bounded.

Consider the Lyapunov function candidate

$$V = \|\boldsymbol{x}\|^2 = x_1^2 + x_2^2.$$

Note that V is bounded below. The time derivative of V evaluated on the trajectories of the system model (7.120) is

$$\dot{V} = 2x_1\dot{x}_1 + 2x_2\dot{x}_2 = 2x_1\left(-2x_1 + x_2 u\right) - 2x_2 x_1 u = -4x_1^2 \leq 0.$$

Thus, the system modeled by (7.120) is uniformly stable, and hence x_1 and x_2 are bounded. We use Lemma 7.5 to show that in addition, $x_1(t) \to 0$ as $t \to \infty$. To use this lemma, we have to show that $\dot{V}$ is uniformly continuous in t. For this it is enough to show that the derivative of $\dot{V}$ is bounded. The derivative of $\dot{V}$ is

$$\ddot{V} = -8x_1\dot{x}_1 = -8x_1\left(-2x_1 + x_2 u\right),$$

which is bounded because u is bounded by assumption and we showed that x_1 and x_2 are also bounded. Hence, $\dot{V}$ is uniformly continuous. The function V satisfies all the conditions of Lemma 7.5 and therefore $x_1(t) \to 0$ as $t \to \infty$.

Example 7.20 Consider a dynamical system model,

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} & = & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u \\ e & = & \boldsymbol{c}\boldsymbol{x} \end{array}\right\} \tag{7.121}$$

where

$$u = \boldsymbol{\phi}^\top \boldsymbol{\xi} \tag{7.122}$$

The components $\phi_i = \phi_i(t)$ of the vector-valued function $\boldsymbol{\phi} = \boldsymbol{\phi}(t) \in \mathbb{R}^K$ are adjustable parameters according to the adaptation law

$$\dot{\boldsymbol{\phi}} = -\boldsymbol{\Gamma} e \boldsymbol{\xi}, \tag{7.123}$$

where $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^\top > 0$ is a design parameter matrix and

$$e = \boldsymbol{c}\boldsymbol{x}.$$

The vector-valued function $\boldsymbol{\xi} = \boldsymbol{\xi}(t) \in \mathbb{R}^K$ is bounded for all t. The matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is asymptotically stable, and $\boldsymbol{b} \in \mathbb{R}^n$. The row vector $\boldsymbol{c} \in \mathbb{R}^{1\times n}$ satisfies the equation

$$\boldsymbol{c} = \boldsymbol{b}^\top \boldsymbol{P}, \tag{7.124}$$

where $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ is the solution of the matrix Lyapunov equation

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{Q}$$

for some $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, where the scalar 2 that multiplies $\boldsymbol{Q}$ is just for convenience to simplify manipulations.

Using the Lyapunov function candidate,

$$V(\boldsymbol{x}, \boldsymbol{\phi}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} + \frac{1}{2}\boldsymbol{\phi}^\top \boldsymbol{\Gamma}^{-1}\boldsymbol{\phi} \tag{7.125}$$

and the Lyapunov-like lemma, we show that $\boldsymbol{x}$ and $\boldsymbol{\phi}$ are both bounded and that

$$\lim_{t\to\infty} \boldsymbol{x}(t) = \boldsymbol{0}.$$

We first evaluate the time derivative of the Lyapunov function candidate $V = V(\boldsymbol{x}, \boldsymbol{\phi})$ on the trajectories of the closed-loop system,

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} &=& \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}(\boldsymbol{\phi}^\top \boldsymbol{\xi}) \\ \dot{\boldsymbol{\phi}} &=& -\boldsymbol{\Gamma} e \boldsymbol{\xi} \end{array}\right\} \tag{7.126}$$

Then, we have

$$\begin{aligned} \dot{V} &= \boldsymbol{x}^\top \boldsymbol{P}\dot{\boldsymbol{x}} + \boldsymbol{\phi}^\top \boldsymbol{\Gamma}^{-1}\dot{\boldsymbol{\phi}} \\ &= \boldsymbol{x}^\top \boldsymbol{P}\left(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\left(\boldsymbol{\phi}^\top \boldsymbol{\xi}\right)\right) - \boldsymbol{\phi}^\top e \boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}\boldsymbol{\xi} \\ &= -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \left(e\boldsymbol{\phi}^\top - e\boldsymbol{\phi}^\top\right)\boldsymbol{\xi} \\ &= -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}, \end{aligned}$$

where we used the fact that $\boldsymbol{c} = \boldsymbol{b}^\top \boldsymbol{P}$ and hence $e = \boldsymbol{c}\boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{c}^\top = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}$. Thus, $\dot{V}$ is negative semi-definite in the $\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\phi} \end{bmatrix}$-space, which implies that $\boldsymbol{x}$ and $\boldsymbol{\phi}$ are bounded.

To show the convergence of $\boldsymbol{x}$, we use the Lyapunov-like lemma. For this, we first show that $\dot{V}$ is uniformly continuous. Indeed, because

$$\ddot{V} = -2\boldsymbol{x}^\top \boldsymbol{Q}\dot{\boldsymbol{x}}$$

is bounded, since $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are bounded, $\dot{V}$ is uniformly continuous. Hence, $\dot{V} \to 0$ as $t \to \infty$. Since $\dot{V} = -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$ and $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, we have to have $\boldsymbol{x}(t) \to \boldsymbol{0}$ as $t \to \infty$.

We can re-state Barbalat's lemma in alternative form as follows (see, for example, [178, page 192]).

Lemma 7.6 *Let $\phi : \mathbb{R} \to \mathbb{R}$ be a uniformly continuous function on $[0, \infty)$. Suppose that $\lim_{t\to\infty} \int_0^t \phi(\tau)d\tau$ exists and is finite. Then,*

$$\phi(t) \to 0 \quad \text{as} \quad t \to \infty.$$

7.13 Input-Output Stability of LTI Systems

The response of a linear time-invariant (LTI) system can be decomposed as the zero-state response and the zero-input response. In the previous sections we studied methods that can be used to investigate the stability properties of the zero-input response of linear systems. Here, we devote ourselves to investigating the stability properties of the zero-state response. To proceed, we need a definition of a bounded signal.

Definition 7.21 *A signal $u(t)$ is bounded if there exists a constant $B < \infty$ such that*

$$|u(t)| \le B \quad \text{for all} \quad t \ge 0.$$

$\diamond$

Using the above definition, we define the bounded-input bounded-output (BIBO) stability of an LTI causal system. This type of stability is concerned with a particular property of the zero-state response of the system.

Definition 7.22 *A system is BIBO stable if every bounded input produces a bounded output.*

◇

Example 7.21 Consider an LTI system whose transfer function is

$$\frac{Y(s)}{U(s)} = G(s) = \frac{1}{s},$$

which is an integrator. A block diagram of this system is shown in Figure 7.25. The

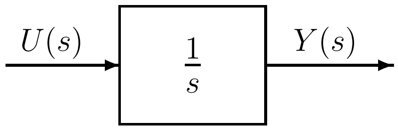


Figure 7.25: System of Example 7.21. This system is not BIBO stable.

integrator is not BIBO stable. Indeed, suppose that the system's input is the unit step, that is, $u(t) = 1(t)$. The Laplace transform of the unit step is $U(s) = 1/s$. Then the Laplace transform of the system's output is

$$Y(s) = \frac{1}{s^2},$$

and therefore

$$y(t) = \mathcal{L}^{-1}\left(\frac{1}{s^2}\right) = t1(t),$$

which is an unbounded signal.

Theorem 7.14 *A single-input single-output (SISO) continuous LTI causal system is BIBO stable if and only if its impulse response, $g(t)$, is absolutely integrable on $[0, \infty)$, that is,*

$$\int_0^\infty |g(t)|dt \leq M < \infty$$

for some constant M.

Proof We prove that the absolute integrability of the impulse response is sufficient for BIBO stability. For a proof that the absolute integrability of the impulse response is also necessary for BIBO stability, we refer to Chen [55, p. 122].

So assume that $g(t)$ is absolutely integrable and let the system's input $u(t)$ be bounded on $[0, \infty)$. Then, using the formula for an LTI system response given by (1.10) on page 21, we obtain

$$\begin{aligned}|y(t)| &= \left|\int_0^t g(\tau)u(t-\tau)d\tau\right| \\ &\leq \int_0^t |g(\tau)||u(t-\tau)|d\tau \\ &\leq B\int_0^t |g(\tau)|d\tau \\ &\leq BM,\end{aligned}$$

which means that the output $y(t)$ is bounded for any bounded input $u(t)$.

□

Example 7.22 The system whose block diagram is depicted in Figure 7.26 is BIBO stable. Indeed, the system's impulse response is

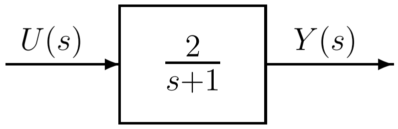


Figure 7.26: System of Example 7.22. This system is BIBO stable.

$$\begin{aligned}g(t) &= \mathcal{L}^{-1}(G(s)) \\ &= \mathcal{L}^{-1}\left(\frac{2}{s+1}\right) \\ &= 2e^{-t}1(t).\end{aligned}$$

The above impulse response is absolutely integrable because

$$\int_0^\infty \left|2e^{-t}\right| dt \leq 2 < \infty.$$

Therefore, the system in this example is BIBO stable.

In the case of a discrete SISO LTI causal relaxed at $k = 0$ system, its output can be expressed as given by (1.24) on page 26, that is,

$$\begin{aligned} y[k] &= \sum_{m=0}^{k} g[k-m]u[m] \\ &= \sum_{m=0}^{k} g[m]u[k-m], \end{aligned}$$

where $g[k]$ is the impulse response sequence.

By analogy with the continuous case, an input sequence $u[k]$ is bounded if for some constant $B < \infty$, we have

$$|u[k]| \leq B \quad \text{for all} \quad k = 0, 1, 2, \ldots$$

Theorem 7.15 *A single-input single-output (SISO) discrete LTI causal system is BIBO stable if and only if its impulse response, $g[k]$, is absolutely summable, that is,*

$$\sum_{k=0}^{\infty} |g[k]| \leq M < \infty$$

for some constant M.

Example 7.23 The system whose block diagram is depicted in Figure 7.26 is not BIBO stable. The system's impulse response is

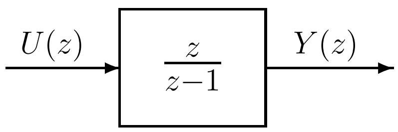


Figure 7.27: System of Example 7.23. This system is not BIBO stable.

$$g[k] = 1 \quad \text{for} \quad k = 0, 1, 2, \ldots$$

The above impulse response is not absolutely summable because

$$\sum_{k=0}^{\infty} |g[k]| = \sum_{k=0}^{\infty} 1^k = \infty.$$

Therefore, the system in this example is not BIBO stable.

Example 7.24 The system with the impulse response,

$$g[k] = (-0.5)^k, \quad k = 0, 1, 2, \ldots$$

is BIBO stable because this impulse response is summable. Indeed, it is easy to verify that

$$\sum_{k=0}^{\infty} |g[k]| = \sum_{k=0}^{\infty} (-0.5)^k = \frac{2}{3}.$$

The above tests for checking if a given system is BIBO stable or not are not particulary convenient. We give now tests for BIBO stability that are very easy to use.

Theorem 7.16 *A continuous SISO LTI system with a proper rational transfer function is BIBO stable if and only if every pole of the transfer function has a negative real part.*

Using the above theorem, we can by inspection determine that the system of Example 7.21 is not BIBO stable while the system of Example 7.22 is BIBO stable.

Theorem 7.17 *A discrete SISO LTI system with a proper rational transfer function is BIBO stable if and only if every pole of the transfer function has a magnitude less than 1. Equivalently, all poles of the transfer function are located in the open unit disc in the z-plane.*

Using the above theorem, we can by inspection determine that the system of Example 7.23 is not BIBO stable.

Theorem 7.18 *A continuous MIMO LTI system with a proper rational transfer function matrix, $\boldsymbol{G}(s)$, is BIBO stable if and only if every pole of $G_{ij}(s)$ has a negative real part, where*

$$\boldsymbol{G}(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{p1}(s) & G_{p2}(s) & \cdots & G_{pm}(s) \end{bmatrix}.$$

Note that Theorem 7.16 is a special case of Theorem 7.18.

Theorem 7.19 *A discrete MIMO LTI system with a proper rational transfer function matrix, $\boldsymbol{G}(z)$, is BIBO stable if and only if every pole of $G_{ij}(z)$ is located in the open unit disc in the z-plane, where*

$$\boldsymbol{G}(z) = \begin{bmatrix} G_{11}(z) & G_{12}(z) & \cdots & G_{1m}(z) \\ \vdots & \vdots & \ddots & \vdots \\ G_{p1}(z) & G_{p2}(z) & \cdots & G_{pm}(z) \end{bmatrix}.$$

Note that Theorem 7.19 is a generalization of Theorem 7.17.

7.14 Input-Output Stability of LTV Systems

Consider a SISO LTV relaxed at t_0, causal system, whose response is expressed by (1.6) on page 19, that is,

$$y(t) = \int_{t_0}^{t} g(t,\tau)u(\tau)d\tau.$$

A block diagram of LTV system is shown in Figure 7.28. A necessary and sufficient condition

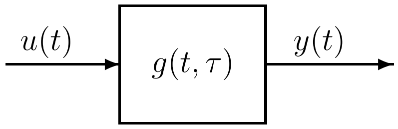


Figure 7.28: Linear time-varying system with the impulse response $g(t,\tau)$.

for the above system to be BIBO stable is that

$$\int_{t_0}^{t} |g(t,\tau)|d\tau \le M < \infty \tag{7.127}$$

for all t and t_0 with $t \ge t_0$. The proof is similar to that for the LTI case.

For the MIMO case the output of an LTV relaxed at t_0, causal system is given by (1.7) on page 19, that is,

$$\boldsymbol{y}(t) = \int_{t_0}^{t} \boldsymbol{G}(t,\tau)\boldsymbol{u}(\tau)d\tau,$$

where

$$\boldsymbol{G}(t,\tau) = \begin{bmatrix} g_{11}(t,\tau) & g_{12}(t,\tau) & \cdots & g_{1m}(t,\tau) \\ g_{21}(t,\tau) & g_{22}(t,\tau) & \cdots & g_{2m}(t,\tau) \\ \vdots & \vdots & \ddots & \vdots \\ g_{p1}(t,\tau) & g_{p2}(t,\tau) & \cdots & g_{pm}(t,\tau) \end{bmatrix}.$$

The MIMO LTV system is BIBO stable if and only if every entry of the impulse response matrix $\boldsymbol{G}(t,\tau)$ satisfies the condition given by (7.127). This condition can be expressed more compactly in an equivalent fashion using norms. Chen [55, p. 137] suggests using the infinite-norms because of their convenience in BIBO stability study,

$$\|\boldsymbol{u}\|_\infty = \max_i |u_i| \quad \text{and} \quad \|\boldsymbol{G}\|_\infty = \text{largest row absolute sum}.$$

Then, a necessary and sufficient condition for the MIMO LTV system to be BIBO stable is that

$$\int_{t_0}^{t} \|\boldsymbol{G}(t,\tau)\|_\infty d\tau \le M < \infty \tag{7.128}$$

for all t and t_0 with $t \ge t_0$.

7.15 Notes

A classical text on stability of dynamical systems is by Hahn [128]. Stability of discrete dynamical systems is presented by Hahn in [127] and LaSalle [204]. Extensive coverage of stability issues can also be found in Willems [351], Slotine and Li [301], Vidyasagar [336], Krstić, Kanellakopoulos, and Kokotović [195], and Khalil [178]. Lyapunov-based controller design methods for a class of uncertain systems are presented in [213]. For more information on the subject of input-output stability, we recommend Antsaklis and Michel [13, Sections 4.7 and 4.85].

There is an interesting story behind the origins of the Routh test. The problem of investigating the stability of a linear dynamical system in terms of the location of the roots of its characteristic equation was posed by Maxwell [231]. On page 271 of his 1868 paper, Maxwell writes, "This condition is mathematically equivalent to the condition that all the possible roots, and all the possible parts of the impossible roots, of a certain equation shall be negative." By impossible roots, Maxwell means complex roots, and, by possible parts, means real parts in today's terminology. He continues, "I have not been able completely to determine these conditions for equations of a higher degree than the third; but I hope that the subject will obtain the attention of mathematicians." A complete general solution to the problem was published by Routh in 1877. When Maxwell and Routh were classmates at Cambridge University in 1854, Routh was placed first in their final math examination ahead of Maxwell. A story says (Smith [305, page 55]) that apparently Maxwell was so sure of coming in first on this exam that he did not even bother to rise early to hear the exam results read out in the Senate House. Maxwell sent his servant instead. When the servant came back, Maxwell asked, "Well, tell me who's second!". Obviously, he did not expect to receive the reply, "You are, sir". In 1877, Routh received the prestigious Adams Prize for solving the problem posed by Maxwell in 1868. According to an account by Truxal [329, page 527], Routh presented his solution with the opening remark, "It has recently come to my attention that my good friend James Clerk Maxwell has had difficulty with a rather trivial problem... ."

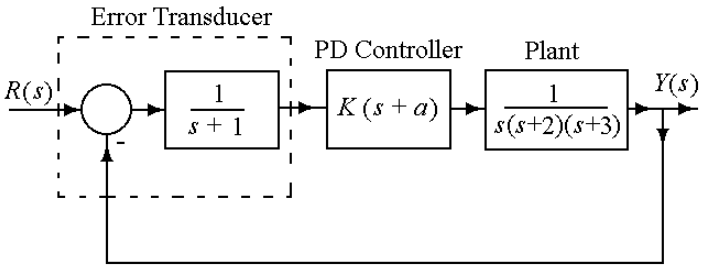


Figure 7.29: A closed-loop feedback system of Exercise 7.1.

7.16 Exercises

Exercise 7.1 **(a)** Consider a negative unity feedback closed-loop system whose open-loop transfer function is

$$G(s) = \frac{6s^3 + 11s^2}{s^4 + 6s + K}.$$

For what values of the parameter K the closed-loop system is asymptotically stable?

(b) Figure 7.29 shows a closed-loop system in which the error, measured by a transducer that has a time constant of 1 sec, is applied to a proportional-plus-derivative (PD) controller. The output of the controller drives a third-order plant. Use the Routh-Hurwitz criterion to find the region of asymptotic stability, of the closed-loop feedback system, in the (a, K)-plane.

Exercise 7.2 **(a)** Determine if the polynomial

$$P(s) = s^7 + 2s^6 + 5s^5 + 7s^4 + 2s^3 - 5s^2 - 8s - 4$$

is Hurwitz. Use the Routh-Hurwitz criterion to find the number of the polynomial RHP zeros and determine the $j\omega$-axis zeros.

(b) Consider a negative unity feedback system with the forward loop transfer function

$$G(s) = \frac{K}{s(s+3)(s+10)}.$$

Use the Routh-Hurwitz criterion to find the interval of K for which the closed-loop system is asymptotically stable.

(c) Consider a negative unity feedback system with the forward loop transfer function

$$G(s) = \frac{K}{s(Ts+1)(0.5\,s+1)}.$$

Use the Routh-Hurwitz criterion to derive the domain of asymptotic stability of the closed-loop system in the (T, K)-plane, and plot it using MATLAB.

Exercise 7.3 For a dynamical system model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 - x_2 \sin x_1 \end{bmatrix},$$

(a) find an equilibrium point for this system;

(b) linearize the system about the found equilibrium point;

(c) determine if the linearized system is stable, asymptotically stable, or unstable.

Exercise 7.4 Consider the following model of a dynamical system,

$$\left.\begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & -x_1^5 - x_2^3 \end{array}\right\} \tag{7.129}$$

(a) Find the equilibrium state of the system. Then, consider the following Lyapunov function candidate,

$$V(x_1, x_2) = \frac{1}{2}x_2^2 + \int_0^{x_1} s^5 ds.$$

Evaluate $\frac{d}{dt}V(x_1, x_2)$ on the trajectories of the system (7.129).

(b) Use the results obtained in (a) to decide whether the equilibrium state is stable, asymptotically stable, or unstable.

Exercise 7.5 For a dynamical system modeled by $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\zeta}(t, \boldsymbol{x})$, where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1/2 & -3/2 \end{bmatrix}, \text{ and } \|\boldsymbol{\zeta}(t, \boldsymbol{x})\| \leq k\|\boldsymbol{x}\|,$$

find $k^* > 0$ such that for any $k < k^*$ the system is globally uniformly asymptotically stable.

Exercise 7.6 Is the system,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} -2 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & -2 & -2 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} \boldsymbol{x} - 2u, \end{aligned}$$

BIBO stable?

Exercise 7.7 Determine the stability of the system

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -2 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & -2 & -2 \end{bmatrix} \boldsymbol{x}$$

in the sense of Lyapunov by solving the Lyapunov matrix equation using the Kronecker product. Take $\boldsymbol{Q} = \boldsymbol{I}$. Verify your solution using the MATLAB function `lyap`.

Exercise 7.8 Is the system,

$$\begin{aligned} \boldsymbol{x}[k+1] &= \begin{bmatrix} -0.8 & 0 & 0 \\ 0.4 & 0 & 0.4 \\ 0 & -0.8 & -0.8 \end{bmatrix} \boldsymbol{x}[k] + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u[k] \\ y &= \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} \boldsymbol{x}[k] - 2u[k], \end{aligned}$$

BIBO stable?

Exercise 7.9 Determine the stability of the system

$$\boldsymbol{x}[k+1] = \begin{bmatrix} -0.8 & 0 & 0 \\ 0.4 & 0 & 0.4 \\ 0 & -0.8 & -0.8 \end{bmatrix} \boldsymbol{x}[k]$$

in the sense of Lyapunov by solving the Lyapunov matrix equation using the Kronecker product. Take $\boldsymbol{Q} = \boldsymbol{I}$. Verify your solution using the MATLAB function `dlyap`.

Exercise 7.10 (J. Hu) For the linear time-invariant (LTV) system model,

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} -e^{-t} & \frac{1}{t+1} \\ 0 & -e^{-t} \end{bmatrix} \boldsymbol{x}(t),$$

- Find the state transition matrix $\boldsymbol{\Phi}(t, \tau)$,
- Is the system asymptotically stable?
- Let $\boldsymbol{x}(0) = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top$. Will the system trajectory $\boldsymbol{x}(t)$ stay bounded as $t \to \infty$?

Exercise 7.11 **(a)** Find the value of

$$J_0 = 2 \int_0^\infty \|\boldsymbol{x}(t)\|^2 dt$$

subject to

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \boldsymbol{x}, \quad \boldsymbol{x}(0) = \begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

(b) Find the value of

$$K_0 = \sum_{k=0}^{\infty} \|\boldsymbol{x}[k]\|^2$$

subject to

$$\boldsymbol{x}[k+1] = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/4 \end{bmatrix} \boldsymbol{x}[k], \quad \boldsymbol{x}[0] = \begin{bmatrix} 3 \\ 0 \end{bmatrix}.$$

Exercise 7.12 Determine the stability or instability of the zero solution of the system,

$$\left.\begin{array}{rcl} \dot{x}_1 &=& x_2 \\ \dot{x}_2 &=& x_1 - x_2 \sin x_1 \end{array}\right\}$$

using the second method of Lyapunov and compare the result of your analysis with the solution to Exercise 7.3.

Exercise 7.13 (based on [328, page 329]) Construct a Lyapunov function for the system modeled by

$$\left.\begin{array}{rcl} \dot{x}_1 &=& x_2 \\ \dot{x}_2 &=& -x_2 - x_1^3. \end{array}\right\}$$

Show that the origin is a globally stable equilibrium state of the system.

Exercise 7.14 Investigate the stability of the following dynamical system model:

$$\left.\begin{array}{rcl}\dot{x}_1 & = & x_2 - 2x_1x_2^2 \\ \dot{x}_2 & = & -x_1 - x_2 - 2x_2^3\end{array}\right\}$$

Is the null solution, $x_1(t) = x_2(t) = 0$, $t \geq 0$, locally stable, globally stable, locally asymptotically stable, globally asymptotically stable, or unstable? Justify your answer.

Exercise 7.15 Use the second method of Lyapunov to estimate the region of asymptotic stability of the equilibrium state $\left[\begin{array}{cc} 2/3 & 2/3 \end{array}\right]^\top$ of the dynamical system model

$$\left.\begin{array}{rcl}\dot{x}_1 & = & (1 - x_1 - 0.5x_2)\, x_1 \\ \dot{x}_2 & = & (1 - 0.5x_1 - x_2)\, x_2\end{array}\right\}$$

Exercise 7.16 For the nonlinear system model, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$, where

$$\boldsymbol{f}(\boldsymbol{x}) = \left[\begin{array}{c} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{array}\right] = \left[\begin{array}{c} 4x_1 - 2x_1^2x_2 \\ -16x_2 + 4x_1^2x_2 \end{array}\right], \tag{7.130}$$

find its equilibrium points and determine their stability properties using the linearization method.

Exercise 7.17 Consider a dynamical system modeled by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. The eigenvalues λ_i, $i = 1, 2, \ldots, n$, of the matrix $\boldsymbol{A}$ are all negative, and the matrix $\boldsymbol{A}$ is diagonalizable, that is, there exists a nonsingular matrix $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ such that

$$\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \boldsymbol{D} = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots \lambda_n\}.$$

Let

$$V(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{T}^\top\boldsymbol{T}\boldsymbol{x} = \boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{x}.$$

(a) Find $\dot{V}$, that is, find the derivative of $V(\boldsymbol{x}(t))$ with respect to time along the solutions of the system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$.

(b) Find the upper bound on $\dot{V}$ such that

$$\dot{V}(\boldsymbol{x}) \leq c\|\boldsymbol{x}\|,$$

where the constant $c < 0$ depends on an eigenvalue of $\boldsymbol{A}$ and an eigenvalue of $\boldsymbol{P}$.

Exercise 7.18 Consider a system of linear equations of the form,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $m \geq n$, $\operatorname{rank}(\boldsymbol{A}) = n$, and $\boldsymbol{x}$ and $\boldsymbol{b}$ are of appropriate dimensions. Consider also an associated iterative algorithm,

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \alpha \boldsymbol{A}^\top \left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}[k]\right), \quad \boldsymbol{x}[0] = \boldsymbol{x}_0,$$

where $\alpha > 0$. Treat the above algorithm as a discrete-time dynamical system.

(a) Find the equilibrium state of the algorithm;

(b) Use the conditions for asymptotic stability of discrete-time systems to find the largest range of values of α for which the algorithm converges to its equilibrium state;

(c) Show that the algorithm, for the range of α found in **(b)**, minimizes the objective function

$$J(\boldsymbol{x}) = \left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\right)^\top \left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\right).$$

Exercise 7.19 Consider a system of linear equations of the form,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $m \geq n$, $\operatorname{rank}(\boldsymbol{A}) = n$, and $\boldsymbol{x}$ and $\boldsymbol{b}$ are of appropriate dimensions. Consider also associated objective function

$$J(\boldsymbol{x}) = \left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\right)^\top \left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\right).$$

Write down an iterative steepest descent algorithm, in terms of $\boldsymbol{A}$ and $\boldsymbol{b}$, for minimizing the above objective function. Carefully derive an expression for the step size α_k in terms of $\nabla J\left(\boldsymbol{x}^{(k)}\right)$ and $\boldsymbol{A}$ only.

Exercise 7.20 **(a)** Given two symmetric positive definite matrices $\boldsymbol{P}$ and $\tilde{\boldsymbol{P}}$. Show, by contradiction, that if for all $\boldsymbol{x}$,

$$\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} \geq \boldsymbol{x}^\top \tilde{\boldsymbol{P}}\boldsymbol{x},$$

then

$$\lambda_{\max}\left(\boldsymbol{P}\right) \geq \lambda_{\max}\left(\tilde{\boldsymbol{P}}\right).$$

(b) Let $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ be the unique solution to

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{Q}, \tag{7.131}$$

where $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, and let $\hat{\boldsymbol{P}} = \hat{\boldsymbol{P}}^\top > 0$ be the unique solution to

$$\boldsymbol{A}^\top \hat{\boldsymbol{P}} + \hat{\boldsymbol{P}}\boldsymbol{A} = -2\hat{\boldsymbol{Q}}, \tag{7.132}$$

where $\hat{\boldsymbol{Q}} = q\boldsymbol{Q}$, $q > 0$. Show that

$$\frac{\lambda_{\min}\left(\hat{\boldsymbol{Q}}\right)}{\lambda_{\max}\left(\hat{\boldsymbol{P}}\right)} = \frac{\lambda_{\min}(\boldsymbol{Q})}{\lambda_{\max}(\boldsymbol{P})}.$$

(c) Prove the following lemma of Patel and Toda [265].

Lemma 7.7 *Let $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ be the unique solution to*

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{Q},$$

where $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$. Then, the ratio

$$\frac{\lambda_{\min}(\boldsymbol{Q})}{\lambda_{\max}(\boldsymbol{P})}$$

is maximized if $\boldsymbol{Q} = \boldsymbol{I}$.

Exercise 7.21 Consider the Lyapunov matrix equation,

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{I}_n,$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is asymptotically stable.

(a) Let $\sigma_{\min}$ be the smallest singular value of $\boldsymbol{A}$, that is,

$$\sigma_{\min} = \sqrt{\lambda_{\min}\left(\boldsymbol{A}\boldsymbol{A}^\top\right)},$$

where $\lambda_{\min}\left(\boldsymbol{A}\boldsymbol{A}^\top\right)$ denotes the smallest eigenvalue of $\boldsymbol{A}\boldsymbol{A}^\top$. Show that $\sigma_{\min} > 0$.

(b) Let $\lambda_{\max}(\boldsymbol{P})$ be the largest eigenvalue of $\boldsymbol{P}$, the solution of the above Lyapunov equation. Show that

$$\lambda_{\max}(\boldsymbol{P}) \geq \frac{1}{\sigma_{\min}}.$$

Exercise 7.22 (based on Theorem 1 in [43]) Consider an asymptotically stable strictly proper transfer function

$$T(s) = \frac{n(s)}{d(s)}.$$

Suppose that the triple $(\boldsymbol{c}, \boldsymbol{A}, \boldsymbol{b})$ is a controllable and observable realization of the transfer function $T(s)$. Thus $\boldsymbol{A}$ must be asymptotically stable. This and the observability assumption guarantees that we can solve the Lyapunov equation

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{c}^\top \boldsymbol{c}$$

for $\boldsymbol{P} = \boldsymbol{P}^\top > 0$. Let

$$q(s) = d(s)\boldsymbol{b}^\top \boldsymbol{P} (s\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{b}.$$

Show that

$$d(s)q(-s) + q(s)d(-s) = n(s)n(-s).$$

Exercise 7.23 Consider a dynamical system model,

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$

where $\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$ and $\|\boldsymbol{x}_0\| \leq 10$. Use the Lyapunov approach to find $T \geq 0$, given $\varepsilon_1 = 20$, such that

$$\|\boldsymbol{x}(t; 0, \boldsymbol{x}_0)\| \leq \varepsilon_1 \quad \text{for all} \quad t \geq T.$$

You can use, for example,

$$\boldsymbol{Q} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

when solving the matrix Lyapunov equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$.

Exercise 7.24 The role of an automotive suspension is to support the vehicle body on the axles. A simple linear model of the suspension assembly associated with a single wheel, called the quarter-car model, is shown in Figure 7.30. We represent the vehicle body by the "sprung" mass, while the tire and axle are represented by the "unsprung" mass. We model the tire as a spring with spring coefficient k_t, and assume its damping to be negligible. The suspension consists of the spring, the shock absorber, and the variable force element. The role of the variable force element is to generate a force that compensates for the uneven road which is the source of the system disturbances. We assume that the variable force element F_a can instantaneously provide any desired force; thus, we have a linear active suspension. The

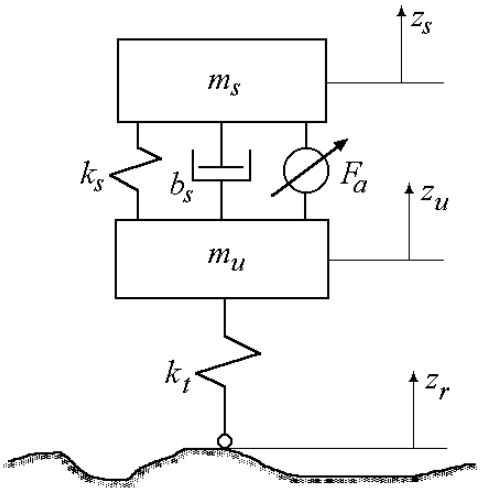


Figure 7.30: Linear quarter-car model

equations modeling the dynamics of the two degree of freedom system depicted in Figure 7.30 are derived by applying Newton's law to each mass of the system, and by taking into account the positive direction of the z coordinates. We obtain

$$\begin{aligned} m_s\ddot{z}_s &= -k_s(z_s - z_u) - b_s(\dot{z}_s - \dot{z}_u) + F_a, \\ m_u\ddot{z}_u &= k_s(z_s - z_u) + b_s(\dot{z}_s - \dot{z}_u) - F_a - k_t(z_u - z_r). \end{aligned}$$

Define the following state variables:

$$\begin{aligned} x_1 &= z_s - z_u = \text{suspension stroke}, \\ x_2 &= \dot{z}_s = \text{sprung mass velocity}, \\ x_3 &= z_u - z_r = \text{tire deflection}, \\ x_4 &= \dot{z}_u = \text{unsprung mass velocity}. \end{aligned}$$

Let $u = F_a$ be the system input, and let $\delta(t) = \dot{z}_r(t)$ be the disturbance input due to the road roughness.

(a) Represent the active suspension model in state-space format.

(b) Substitute the vehicle data, given in Table 7.1, into the modeling equations.

Table 7.1: Data for a vehicle suspension taken from [314].

m_s	355 kg
m_u	46 kg
k_s	14384 N/m
b_s	1860 N/m/sec
k_t	300000 N/m

(c) Our goal is to isolate the sprung mass from road disturbances. Specifically, we wish to design a control law such that the sprung mass velocity $\dot{z}_s(t) = x_2$ is as close to 0 as possible in spite of road disturbances. Thus, we assume that we can measure the sprung mass velocity, which means that the output of the plant is

$$y(t) = \boldsymbol{c}\boldsymbol{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}(t).$$

To investigate the effect of road disturbances on the vertical velocity of the sprung mass, set $u = 0$ and $\delta(t) = d\sin(6t)$. Generate a plot of y versus t for $d = 0.5$.

(d) Construct a state-feedback Lyapunov-based controller of Theorem 7.4. Generate a plot of y of the closed-loop system versus t for $d = 0.5$. Discuss your strategy of the above state-feedback controller design that would minimize the influence of the disturbance $\delta(t)$ on the sprung mass velocity y.

Exercise 7.25 Consider a dynamical system model

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} (u + h(t, \boldsymbol{x})),$$

where

$$|h(t, \boldsymbol{x})| \leq 3.$$

Let $\boldsymbol{P}$ be the solution of the matrix Lyapunov equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -2\boldsymbol{I}_2$, and let

$$u = \begin{cases} -3\,\mathrm{sign}(\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}) & \text{if } \ |\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}| \geq \nu \\ -3\dfrac{\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}}{\nu} & \text{if } \ |\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{x}| < \nu, \end{cases}$$

where $\nu = 0.5$. Find the radius of the ball, in $\mathbb{R}^2$, that is uniformly stable for the closed-loop system.

Chapter 8

A Computational Tool for Control Design

> Henry Ford offered customers the Model T Ford "in any color the customer wants, as long as it is black." Henry Ford's policy resulted from subordination to the assembly line constraint that could not keep up with market demand. This was a resource constraint. To maximize throughput it was important for Ford not to change colors during production. This policy assured maximum profit and value for Ford by subordinating the whole system (including customers) to production.
> Henry Ford's failure was that he stuck to his policy even when new competitors entered the market; then the resource constraint became a market constraint resulting in a steep drop in sales and market share. The right step would have been to realize that there was now a market constraint and subordinate the system to the constraint. Competition and customers' needs should have driven Ford to produce a variety of cars: different styles, colors, or amenities.—Ronen and Pass [284, p. 84]

8.1 Motivation

Many control design problems for linear and nonlinear systems can be formulated as *convex optimization* problems. To define a convex optimization problem, we need definitions of a convex set and a convex function.

Definition 8.1 *A set $\boldsymbol{\Theta} \subseteq \mathbb{R}^n$ is convex if for any $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\boldsymbol{\Theta}$, the line segment between $\boldsymbol{x}$ and $\boldsymbol{y}$ lies in $\boldsymbol{\Theta}$, that is,*

$$\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y} \in \boldsymbol{\Theta} \quad \textit{for any} \quad \alpha \in (0,1).$$

$\diamond$

Examples of convex sets include the empty set, $\mathbb{R}^n$, a subspace, a line, a line segment.

Definition 8.2 *A real-valued function*

$$f : \Omega \to \mathbb{R}$$

defined on a convex set $\boldsymbol{\Theta} \subseteq \mathbb{R}^n$ is convex if for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ and all $\alpha \in (0,1)$,

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}).$$

$\diamond$

Thus a real-valued function on n variables is convex over a convex set if for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ the points on the line segment connecting $(\boldsymbol{x}, f(\boldsymbol{x}))$ and $(\boldsymbol{y}, f(\boldsymbol{y}))$ line on or above the graph of f. For more information on the subject of convex function, we recommend [62, Section 21.2].

A convex optimization problem is the one where the objective function to be minimized or maximized is convex and the constraint set, over which we optimize the objective function, is a convex set.

Semidefinite programming (SDP) is a subfield of convex optimization concerned with minimizing or maximizing a linear objective function subject to a matrix inequality. The matrix inequality constraint is convex and defines a convex feasible set over which the linear objective function is being optimized. Semidefinite programming can be viewed as an extension of linear programming, where the componentwise inequalities on vectors are replaced by matrix inequalities. For further reading on the subject of semidefinite programming, we recommend an excellent survey paper by Vandenberghe and Boyd [334].

8.2 Linear Matrix Inequality and Its Properties

Consider $n+1$ real symmetric matrices

$$\boldsymbol{F}_i = \boldsymbol{F}_i^\top \in \mathbb{R}^{m \times m}, \quad i = 0, 1, \ldots, n$$

and a vector

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top \in \mathbb{R}^n.$$

Then,

$$\begin{aligned}\boldsymbol{F}(\boldsymbol{x}) &= \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \ldots + x_n\boldsymbol{F}_n \\ &= \boldsymbol{F}_0 + \sum_{i=1}^{n} x_i\boldsymbol{F}_i\end{aligned}$$

is an affine function of $\boldsymbol{x}$, rather than a linear function of $\boldsymbol{x}$ because $\boldsymbol{F}(\boldsymbol{x})$ is composed of a linear term, $\sum_{i=1}^{n} x_i\boldsymbol{F}_i$, and a constant term, $\boldsymbol{F}_0$.

Consider now an expression of the form

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \cdots + x_n\boldsymbol{F}_n \geq 0.$$

The above is to be interpreted as follows: find a set of vectors $\boldsymbol{x}$ such that

$$\boldsymbol{z}^\top \boldsymbol{F}(\boldsymbol{x})\boldsymbol{z} \geq 0 \text{ for all } \boldsymbol{z} \in \mathbb{R}^m,$$

that is, $\boldsymbol{F}(\boldsymbol{x})$ is positive definite. Recall that $\boldsymbol{F}_i$s are constant matrices, $\boldsymbol{x}$ is unknown, and $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x})^\top$ is an affine function $\boldsymbol{x}$.

The expression, $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \cdots + x_n\boldsymbol{F}_n \geq 0$, is referred to as in the literature as the *linear matrix inequality* (LMI) although the term the *a*ffine matrix inequality would be a correct term to use.

It is easy to verify, see Exercise 8.1, that the set, $\{\boldsymbol{x} : \boldsymbol{F}(\boldsymbol{x}) \geq 0\}$, is a convex set.

A system of LMIs,

$$\boldsymbol{F}_1(\boldsymbol{x}) \geq 0,\ \boldsymbol{F}_2(\boldsymbol{x}) \geq 0, \ldots, \boldsymbol{F}_k(\boldsymbol{x}) \geq 0,$$

can be represented as one single LMI,

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{F}_1(\boldsymbol{x}) & & & \\ & \boldsymbol{F}_2(\boldsymbol{x}) & & \\ & & \ddots & \\ & & & \boldsymbol{F}_k(\boldsymbol{x}) \end{bmatrix} \geq 0.$$

A linear matrix inequality, involving an m-by-n constant matrix $\boldsymbol{A}$, of the form,

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$$

can be represented as m LMIs

$$b_i - \boldsymbol{a}_i^\top \boldsymbol{x} \geq 0, \quad i = 1, 2, \ldots, m,$$

where $\boldsymbol{a}_i^\top$ is the i-th row of the matrix $\boldsymbol{A}$. We can view each scalar inequality as an LMI. We then represent m LMIs as one LMI,

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} b_1 - \boldsymbol{a}_1^\top \boldsymbol{x} & & & \\ & b_2 - \boldsymbol{a}_2^\top \boldsymbol{x} & & \\ & & \ddots & \\ & & & b_m - \boldsymbol{a}_m^\top \boldsymbol{x} \end{bmatrix} \geq 0.$$

With the above facts as a background, we can now give an example of semidefinite programming,

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{F}(\boldsymbol{x}) \geq 0. \end{array}$$

The matrix property that we discuss next is useful when converting some nonlinear matrix inequalities into linear inequalities. We start with a simple observation. Let $\boldsymbol{P} = \boldsymbol{P}^\top$ be a nonsingular n-by-n matrix and let $\boldsymbol{x} = \boldsymbol{M}\boldsymbol{z}$, where $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ such that $\det \boldsymbol{M} \neq 0$. Then, we have

$$\boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{x} \geq 0 \;\text{ if and only if }\; \boldsymbol{z}^\top \boldsymbol{M}^\top \boldsymbol{P} \boldsymbol{M} \boldsymbol{z} \geq 0,$$

that is,

$$\boldsymbol{P} \geq 0 \;\text{ if and only if }\; \boldsymbol{M}^\top \boldsymbol{P} \boldsymbol{M} \geq 0.$$

Suppose that we have a square block matrix

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix}.$$

Then

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix} \geq 0 \;\text{ if and only if }\; \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{O} \end{bmatrix} \geq 0,$$

where $\boldsymbol{I}$ is an identity matrix of appropriate dimension. In other words,

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix} \geq 0 \;\text{ if and only if }\; \begin{bmatrix} \boldsymbol{D} & \boldsymbol{B}^\top \\ \boldsymbol{B} & \boldsymbol{A} \end{bmatrix} \geq 0.$$

Consider now a square block matrix of the form

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix},$$

where $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{22}$ are square submatrices. Let us now suppose that the matrix $\boldsymbol{A}_{11}$ is invertible. Then we have

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Delta} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix},$$

where

$$\boldsymbol{\Delta} = \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}.$$

The matrix $\boldsymbol{\Delta}$ is called the Schur complement of $\boldsymbol{A}_{11}$. For the case when $\boldsymbol{A}_{12} = \boldsymbol{A}_{21}^\top$, we have

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{21}^\top \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Delta} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{21}^\top \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix},$$

where

$$\boldsymbol{\Delta} = \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{21}^\top.$$

Hence,

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{21}^\top \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} > 0 \text{ if and only if } \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Delta} \end{bmatrix} > 0,$$

that is,

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{21}^\top \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} > 0 \text{ if and only if } \boldsymbol{A}_{11} > 0 \text{ and } \boldsymbol{\Delta} > 0.$$

Many problems of optimization, control design, and signal processing can be formulated in terms of LMIs. To test whether or not there exist solutions $\boldsymbol{x}$ to $\boldsymbol{F}(\boldsymbol{x}) > 0$ is called a feasibility problem. We say that the LMI is non-feasible if no solution exists.

Example 8.1 We present an example illustrating the LMI feasibility problem. It is well known that a constant square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ has its eigenvalues in the open left half-complex plane if and only if for any real, symmetric, positive definite $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ the solution $\boldsymbol{P} = \boldsymbol{P}^\top$ to the Lyapunov matrix equation

$$\boldsymbol{A}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q},$$

is positive definite. That is, real parts of the eigenvalues of $\boldsymbol{A}$ are all negative if and only if there exists a real symmetric positive definite matrix $\boldsymbol{P}$ such that

$$\boldsymbol{A}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} < 0,$$

or equivalently,

$$-\boldsymbol{A}^\top\boldsymbol{P} - \boldsymbol{P}\boldsymbol{A} > 0.$$

Thus, the location of all eigenvalues of $\boldsymbol{A}$ in the open left half-complex plane is equivalent to feasibility of the following LMI,

$$\begin{bmatrix} \boldsymbol{P} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{A}^\top \boldsymbol{P} - \boldsymbol{P}\boldsymbol{A} \end{bmatrix} \geq 0,$$

that is, the existence of a symmetric positive definite $\boldsymbol{P}$ such that the above inequality holds.

Remark 8.1 *Most of the optimization solvers do not handle strict inequalities. Therefore, the operator $>$ is the same as $\geq$, and so $>$ implements the non-strict inequality $\geq$.*

◇

We now show that finding $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ such that $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} < 0$ is indeed an LMI. For this let

$$\boldsymbol{P} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ x_2 & x_{n+1} & \cdots & x_{2n-1} \\ \vdots & & & \vdots \\ x_n & x_{2n-1} & \cdots & x_q \end{bmatrix},$$

where

$$q = \frac{n(n+1)}{2}.$$

We next define the following matrices,

$$\boldsymbol{P}_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

$$\boldsymbol{P}_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

$$\vdots$$

$$
\boldsymbol{P}_q = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.
$$

Note that $\boldsymbol{P}_i$ has only non-zero elements corresponding to x_i in $\boldsymbol{P}$. Let

$$
\boldsymbol{F}_i = -\boldsymbol{A}^\top \boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{A}, \quad i = 1, 2, \ldots, q.
$$

We then write

$$
\begin{aligned}
\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} &= x_1 \left(\boldsymbol{A}^\top \boldsymbol{P}_1 + \boldsymbol{P}_1 \boldsymbol{A}\right) + x_2 \left(\boldsymbol{A}^\top \boldsymbol{P}_2 + \boldsymbol{P}_2 \boldsymbol{A}\right) + \cdots + x_q \left(\boldsymbol{A}^\top \boldsymbol{P}_q + \boldsymbol{P}_q \boldsymbol{A}\right) \\
&= -x_1 \boldsymbol{F}_1 - x_2 \boldsymbol{F}_2 - \cdots - x_q \boldsymbol{F}_q \\
&< 0.
\end{aligned}
$$

Let

$$
\boldsymbol{F}(\boldsymbol{x}) = x_1 \boldsymbol{F}_1 + x_2 \boldsymbol{F}_2 + \cdots + x_q \boldsymbol{F}_q,
$$

then

$$
\boldsymbol{P} = \boldsymbol{P}^\top > 0 \quad \text{and} \quad \boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} < 0
$$

if and only if

$$
\boldsymbol{F}(\boldsymbol{x}) > 0.
$$

More examples of linear matrix inequalities in system and control theory can be found in the book by Boyd et al. [41].

8.3 LMI Solvers

The LMI $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_0 + x_1 \boldsymbol{F}_1 + \cdots + x_n \boldsymbol{F}_n \geq 0$ is called the *canonical representation* of an LMI. The LMIs in the canonical form are very inefficient from a storage view-point as well as from the efficiency of the LMI solvers view-point. The LMI solvers use a structured representation of LMIs.

One can use MATLAB's toolbox to efficiently solve LMIs. This toolbox has three types of LMI solvers that we discuss next.

8.3.1 Solving the Feasibility Problem

This solver computes a feasible solution, that is, is solves the feasible problem defined by a given system of LMI constraints. Using this solver, we can solve any system of LMIs of the form

$$\boldsymbol{N}^\top \boldsymbol{\mathcal{L}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)\boldsymbol{N} < \boldsymbol{M}^\top \boldsymbol{\mathcal{R}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)\boldsymbol{M},$$

where $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k$ are matrix variables, $\boldsymbol{N}$ is the left outer factor, $\boldsymbol{M}$ is the right outer factor, $\boldsymbol{\mathcal{L}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$ is the left inner factor and $\boldsymbol{\mathcal{R}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$ is the right inner factor. The matrices $\boldsymbol{\mathcal{L}}(\cdot)$ and $\boldsymbol{\mathcal{R}}(\cdot)$ are, in general, symmetric block matrices. We note that the term "left-hand side" refers to what is on the "smaller" side of the inequality $0 < \boldsymbol{X}$. Thus in $\boldsymbol{X} > 0$, the matrix $\boldsymbol{X}$ is still on the right-hand side because it is on the "larger " side of the inequality.

We now provide a description of an approach that can be used to solve the given LMI system feasibility problem. To initialize the LMI system description, we type `setlmis([])`. Then we declare matrix variables using the command `lmivar`. The command `lmiterm` allows us to specify LMIs that constitute the LMI system under consideration. Next, we need to obtain an internal representation using the command `getlmis`. We next compute a feasible solution to the given LMI system using the command `feasp`. After that we extract matrix variable values with the command `dec2mat`. In summary, a general structure of a MATLAB program for finding a feasible solution to the set of LMIs could have the form,

```
setlmis([])
lmivar
lmiterm
.
.
.
lmiterm
getlmis
feasp
dec2mat
```

We now analyze the above commands in some detail so that the reader can write simple MATLAB programs for solving LMIs after finishing reading this section. First, to create a new matrix-valued variable, say, $\boldsymbol{X}$, in the given LMI system, we use the command

```
X=lmivar(type,structure)
```

The input `type` specifies the structure of the variable `X`. There may be three structures of matrix variables. When `type=1`, we have a symmetric block diagonal matrix variable. The input `type=1` refers to a full rectangular matrix variable. Finally, `type=3` refers to other cases. The second input `structure` gives additional information on the structure of the matrix variable `X`. For example the matrix variable `X` could have the form

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{D}_2 & \cdots & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{D}_r \end{bmatrix},$$

where each $\boldsymbol{D}_i$ is a square symmetric matrix. For the above example we would use `type=1`. The above matrix variable has r blocks. The input `structure` is then an $r \times 2$ matrix whose i-th row describes the i-th block, where the first component of each row gives the corresponding block size, while the second element of each row specifies the block type. For example,

```
X=lmivar(1,[3 1])
```

specifies a full symmetric 3×3 matrix variable. On the other hand,

```
X=lmivar(2,[2 3])
```

specifies a rectangular 2×3 matrix variable. Finally, the matrix variable $\boldsymbol{S}$ of the form

$$\boldsymbol{S} = \left[\begin{array}{cc|cc} s_1 & 0 & 0 & 0 \\ 0 & s_1 & 0 & 0 \\ -- & -- & -- & -- \\ 0 & 0 & s_2 & s_3 \\ 0 & 0 & s_3 & s_4 \end{array}\right],$$

can be declared as follows:

```
S=lmivar(1,[2 0;2 1])
```

Note that in the above the second component of the first row of the second input has the value of zero, that is, `structure(1,2)=0`. This describes a scalar block matrix of the form

$$\boldsymbol{D}_1 = s_1 \boldsymbol{I}_2.$$

Note that the second block is a 2×2 symmetric full block.

We next take a closer look at the next command whose purpose is to specify the terms of the LMI system of interest. This command has the form

```
lmiterm(termid,A,B,flag)
```

We briefly describe each of the four inputs of this command. The first input, `termid`, is a row with four elements that specify the terms of each LMI of the LMI system. We have `termid(1)=n` to specify the left-hand side of the n-th LMI. We use `termid(1)=-n` to specify the right-hand side of the n-th LMI. The middle two elements of the input `termid` specify the block location. Thus `termid(2,3)=[i j]` refers to the term that belongs to the (i, j) block of the LMI specified by the first component. Finally, `termid(4)=0` for the constant term, `termid(4)=X` for the variable term in the form $\boldsymbol{AXB}$, while `termid(4)=-X` for the variable term in the form $\boldsymbol{AX}^\top\boldsymbol{B}$. The third and the fourth inputs of the command `lmiterm` give the values of the left and right outer factors, that is, `A` and `B` give the value of the constant outer factors in the variable terms, $\boldsymbol{AXB}$ or in $\boldsymbol{AX}^\top\boldsymbol{B}$. Finally the fourth input to `lmiterm` serves as a compact way to specify the expression

$$\boldsymbol{AXB} + (\boldsymbol{AXB})^\top.$$

Thus `flag='s'` can be used to denote a symmetrized expression. We illustrate the above command on the following LMI,

$$\boldsymbol{PA} + (\boldsymbol{PA})^\top < 0.$$

We have one LMI with two terms. We could use the following description of this single LMI,

```
lmiterm([1 1 1 P],1,A)
lmiterm([1 1 1 -P],A',1)
```

On the other hand, we can compactly describe the above LMI with the use of the `flag` as follows,

```
lmiterm([1 1 1 P],1,A,'s')
```

Now to solve the feasibility problem we could have typed

```
[tmin,xfeas]=feas(lmis)
```

In general, for a given LMI feasibility problem of the form

$$\begin{aligned} \text{find} \quad & \boldsymbol{x} \\ \text{such that} \quad & \boldsymbol{L}(\boldsymbol{x}) < \boldsymbol{R}(\boldsymbol{x}) \end{aligned}$$

the command `feasp` solves the auxiliary convex problem

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \boldsymbol{L}(\boldsymbol{x}) < \boldsymbol{R}(\boldsymbol{x}) + t\boldsymbol{I}. \end{array}$$

The system of LMIs is feasible if the minimal t is negative. We add that the current value of t is displayed by `feasp` at each iteration.

Finally we convert the output of the LMI solver into matrix variables using the command

```
P=dec2mat(lmis,xfeas,P).
```

8.3.2 Minimization of a Linear Objective Under LMI Constraints

This solver is invoked using the function `mincx`. It solves the convex problem

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}(\boldsymbol{x}) < \boldsymbol{B}(\boldsymbol{x}). \end{array}$$

The notation $\boldsymbol{A}(\boldsymbol{x}) < \boldsymbol{B}(\boldsymbol{x})$ is a shorthand notation for general structured LMI systems.

Thus to solve a `mincx` problem, in addition to specifying the LMI constraints as in the `feasp` problem, we also declare the linear objective function. Then we invoke the function `mincx`. We illustrate both the `feasp` and `mincx` solvers in the following example.

Example 8.2 We consider the optimization problem

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}, \end{array}$$

where

$$\begin{aligned} \boldsymbol{c}^\top &= \begin{bmatrix} 4 & 5 \end{bmatrix}, \\ \boldsymbol{A} &= \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 2 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 8 \\ 18 \\ 14 \end{bmatrix}. \end{aligned}$$

We first solve the feasibility problem, that is, we find an $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$, using the `feasp` solver. After that we solve the above minimization problem using the `mincx` solver. A simple MATLAB code accomplishing the above tasks is shown below.

```
% Enter problem data
A =[1 1;1 3;2 1];
b=[8 18 14]';
c=[-4 -5]';
setlmis([]);
X=lmivar(2,[2 1]);
lmiterm([1 1 1 X],A(1,:),1);
lmiterm([1 1 1 0],-b(1));
lmiterm([1 2 2 X],A(2,:),1);
lmiterm([1 2 2 0],-b(2));
lmiterm([1 3 3 X],A(3,:),1);
lmiterm([1 3 3 0],-b(3));
lmis=getlmis;
%------------------------------------
disp('-------------feasp result---------------')
[tmin,xfeas]=feasp(lmis);
x_feasp=dec2mat(lmis,xfeas,X)
disp('-------------mincx result---------------')
[objective,x_mincx]=mincx(lmis,c,[0.0001 1000 0 0 1])
```

The `feasp` produces

$$\boldsymbol{x}_{feasp} = \begin{bmatrix} -64.3996 \\ -25.1712 \end{bmatrix}.$$

The `mincx` produces

$$\boldsymbol{x}_{mincx} = \begin{bmatrix} 3.0000 \\ 5.0000 \end{bmatrix}.$$

In the next example we discuss the function `defcx` that we can use to construct the vector $\boldsymbol{c}$ used by the LMI solver `mincx`.

Example 8.3 Suppose that we wish to solve the optimization problem

$$\begin{array}{rl} \text{minimize} & \operatorname{trace}(\boldsymbol{P}) \\ \text{subject to} & \boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} < 0, \end{array}$$

where $\boldsymbol{P} = \boldsymbol{P}^\top > 0$.

We can use the function `mincx` to solve the above problem. However, to use the function `mincx`, we need a vector $\boldsymbol{c}$ such that

$$\boldsymbol{c}^\top \boldsymbol{x} = \operatorname{trace}(\boldsymbol{P}).$$

After specifying the LMIs and obtaining their internal representation using, for example, the command `lmisys=getlmis`, we can obtain the desired $\boldsymbol{c}$ with the following MATLAB code,

```
q=decnbr(lmisys);
c=zeros(q,1);
for j=1:q
    Pj=defcx(lmisys,j,P);
    c(j)=trace(Pj);
end
```

Having obtained the vector $\boldsymbol{c}$, we can use the function `mincx` to solve the optimization problem.

8.3.3 Generalized Eigenvalue Minimization Problem

This problem can be stated as

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \\ & \begin{array}{rcl} \boldsymbol{C}(\boldsymbol{x}) & < & \boldsymbol{D}(\boldsymbol{x}) \\ 0 & < & \boldsymbol{B}(\boldsymbol{x}) \\ \boldsymbol{A}(\boldsymbol{x}) & < & \lambda \boldsymbol{B}(\boldsymbol{x}). \end{array} \end{array}$$

In the above, we need to distinguish between standard LMI constraints of the form $\boldsymbol{C}(\boldsymbol{x}) < \boldsymbol{D}(\boldsymbol{x})$ and the linear-fractional LMIs of the form $\boldsymbol{A}(\boldsymbol{x}) < \lambda \boldsymbol{B}(\boldsymbol{x})$ that are concerned with the generalized eigenvalue λ. For more information on this type of the LMI solver, we refer to the LMI Lab in the MATLAB's Robust Control Toolbox user's guide.

8.4 Notes

A quick introduction to the LMI toolbox is the tutorial that can be accessed by typing in `lmidem`. It is very instructive and we recommend to try it.

We mention that in addition to the MATLAB's LMI Toolbox there is another toolbox for solving LMIs called LMITOOL, a built in software package in Scilab toolbox, developed at INRIA in France. Scilab offers free software for numerical optimization. There is a version of LMITOOL for MATLAB that can be uploaded from the website of the Scilab Consortium.

Yet Another LMI Package (YALMIP) for solving LMIs was developed in Switzerland in the Automatic Control Laboratory at ETH. YALMIP is an "intuitive and flexible modelling language for solving optimization problems in MATLAB."

8.5 Exercises

Exercise 8.1 Show that the set

$$\{\boldsymbol{x} : \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \cdots + x_n\boldsymbol{F}_n \geq 0\}$$

is convex.

Exercise 8.2 Consider the following symmetric matrix,

$$\boldsymbol{Q} = \left[\begin{array}{cc|c} 2 & 2 & 1 \\ 2 & 3 & 1 \\ \hline 1 & 1 & 2 \end{array}\right] = \left[\begin{array}{c|c} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \hline \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{array}\right].$$

(i) Compute the Schur complement, Δ_{11}, of $\boldsymbol{Q}_{11}$;

(ii) Compute the Schur complement, Δ_{22}, of $\boldsymbol{Q}_{22}$.

Exercise 8.3 Use the Schur complement to show that solving the discrete Lyapunov matrix equation

$$\boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P} < 0$$

for $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ is equivalent to solving the LMI,

$$\begin{bmatrix} \boldsymbol{P} & \boldsymbol{A}^\top \boldsymbol{P} \\ \boldsymbol{P}\boldsymbol{A} & \boldsymbol{P} \end{bmatrix} > 0$$

Exercise 8.4 Write a MATLAB program that finds a positive definite $\boldsymbol{P}$ such that

$$\begin{aligned} \boldsymbol{A}_1^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_1 &< 0 \\ \boldsymbol{A}_2^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_2 &< 0, \end{aligned}$$

and

$$P < 5I_2,$$

where

$$A_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} -2 & 0 \\ 1 & -1 \end{bmatrix}.$$

Exercise 8.5 It is known, see Example 15.7, that for the matrices

$$A_1 = \begin{bmatrix} -1 & 4 \\ 0 & -2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -1 & 0 \\ 4 & -2 \end{bmatrix}$$

there is no common positive definite P such that $A_i^\top P + PA_i < 0$ for $i = 1, 2$, that is, there is no $P = P^\top > 0$ such that

$$A_1^\top P + PA_1 < 0 \quad \text{and} \quad A_2^\top P + PA_2 < 0.$$

Verify this using LMIs.

Exercise 8.6 Formulate LMIs that find a state feedback controller

$$u = -Kx$$

that stabilizes each individual system for a family of linear systems,

$$\dot{x} = A_i x + B_i u, \quad i = 1, \ldots, N.$$

One way to solve this is to formulate LMIs that solve the problem of finding $P = P^\top > 0$ such that

$$(A_i - B_i K)^\top P + P(A - B_i K) < 0, \quad i = 1, \ldots, N.$$

Chapter 9

Optimal Control

Nobel laureate H. A. Simon revolutionized management by identifying a managerial phenomenon that causes decision-making failures. Simon (1957) claimed that executives, engineers, and decision makers are trying to become optimizers while making decisions.

Optimizer: *Executive, engineer, or decision maker who wants to make the best possible decision, without consideration of time constraints.*

Simon proposes an alternative approach and suggests that decision makers behave as "satisficres," that is, aspire to reach a "satisfactory solution," a "good enough solution," and not necessarily a perfect one (the "optimal" one).

Satisficer: *An executive or decision maker who is satisfied with a reasonable solution that will significantly improve the system and does not strive for the optimal solution.*

The satisficer sets a level of aspiration, a threshold he aspires to achieve. The objective is no longer to maximize or minimize some performance measure, but to achieve a solution that will improve the measure beyond the predefined level of aspiration.—Ronen and Pass [284, pp.15–16]

One of the essential elements of the control problem, discussed in Section 1.1, is a means of testing the performance of any proposed control law. Whenever we use the term "best" or "optimal" to describe the effectiveness of a given control strategy, we do so with respect to some numerical index of performance called the performance index. We assume that the value of the performance index decreases as the quality of the given admissible control law increases. The admissible controller that ensures the completion of the system objective and at the same time minimizes the performance index is called an *optimal controller* for the system. In this chapter, we discuss various methods of constructing optimal controllers.

9.1 Performance Indices

Constructing a performance index, that is, choosing a means to measure the system performance, can be considered as a part of the system modeling. Here, we discuss some typical choices of performance indices.

First, suppose that the objective is to control a dynamical system modeled by the equations,

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 && (9.1) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t), && (9.2) \end{aligned}$$

on a fixed interval $[t_0, t_f]$ so that the components of the state vector are "small". A suitable performance index to be minimized would be

$$J_1 = \int_{t_0}^{t_f} \boldsymbol{x}(t)^\top \boldsymbol{x}(t) dt. \tag{9.3}$$

Obviously, if J_1 is small, then the state vector norm, $\|\boldsymbol{x}(t)\|$, is small in the sense of the above performance index.

If the objective is to control the system so that the components of the output, $\boldsymbol{y}(t)$, are to be small, then we could use the performance index

$$\begin{aligned} J_2 &= \int_{t_0}^{t_f} \boldsymbol{y}(t)^\top \boldsymbol{y}(t) dt \\ &= \int_{t_0}^{t_f} \boldsymbol{x}(t)^\top \boldsymbol{C}^\top \boldsymbol{C}\boldsymbol{x}(t) dt \\ &= \int_{t_0}^{t_f} \boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t) dt, && (9.4) \end{aligned}$$

where the weight matrix $\boldsymbol{Q} = \boldsymbol{C}^\top \boldsymbol{C}$ is symmetric positive semi-definite.

In the case when we wish to control the system in such a manner that the components of the input, $\boldsymbol{u}(t)$, are "not too large", a suitable performance index to be minimized is

$$J_3 = \int_{t_0}^{t_f} \boldsymbol{u}(t)^\top \boldsymbol{u}(t) dt, \tag{9.5}$$

or

$$J_4 = \int_{t_0}^{t_f} \boldsymbol{u}(t)^\top \boldsymbol{R} \boldsymbol{u}(t) dt, \tag{9.6}$$

where the weight matrix $\boldsymbol{R}$ is symmetric positive definite. There is no loss of generality in assuming the weight matrix $\boldsymbol{R}$ to be symmetric. If $\boldsymbol{R}$ was not symmetric, we could represent the quadratic term $\boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u}$ equivalently as

$$\boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u} = \boldsymbol{u}^\top \left(\frac{\boldsymbol{R} + \boldsymbol{R}^\top}{2} \right) \boldsymbol{u},$$

where the matrix $\frac{1}{2}\left(\boldsymbol{R} + \boldsymbol{R}^\top\right)$ is symmetric.

As pointed out by Owens [260, page 186], we cannot simultaneously minimize the performance indices (9.3) and (9.5) because minimization of (9.3) requires large control signals, while minimization of (9.5) requires small control signals. To solve the dilemma, we could compromise between the two conflicting objectives by minimizing the performance index that is a convex combination of J_1 and J_3,

$$\begin{aligned} J &= \lambda J_1 + (1-\lambda) J_3 \\ &= \int_{t_0}^{t_f} \left(\lambda \boldsymbol{x}(t)^\top \boldsymbol{x}(t) + (1-\lambda) \boldsymbol{u}(t)^\top \boldsymbol{u}(t) \right) dt, \end{aligned} \tag{9.7}$$

where λ is a parameter in the range $[0,1]$. If $\lambda = 1$, then $J = J_1$ and if $\lambda = 0$, then $J = J_3$. By trial and error, we could select λ from the interval $[0,1]$ to compromise between the two extremes. A generalization of the performance index (9.7) is

$$J = \int_{t_0}^{t_f} \left(\boldsymbol{x}(t)^\top \boldsymbol{Q} \boldsymbol{x}(t) + \boldsymbol{u}(t)^\top \boldsymbol{R} \boldsymbol{u}(t) \right) dt.$$

In certain applications, we may wish the final state $\boldsymbol{x}(t_f)$ to be as close as possible to $\mathbf{0}$. Then, a possible performance measure to be minimized is

$$\boldsymbol{x}(t_f)^\top \boldsymbol{F} \boldsymbol{x}(t_f), \tag{9.8}$$

where $\boldsymbol{F}$ is a symmetric positive definite matrix.

We can combine the performance measures (9.4), (9.6), and (9.8) when our control aim is to keep the state "small", the control "not too large", and the final state as near to $\mathbf{0}$ as possible. The resulting performance index is

$$J = \frac{1}{2}\boldsymbol{x}(t_f)^\top \boldsymbol{F}\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left(\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}(t)^\top \boldsymbol{R}\boldsymbol{u}(t)\right) dt, \tag{9.9}$$

where the factor 1/2 is placed to simplify subsequent algebraic manipulations. Minimizing (9.9) subject to (9.1) is called the *linear quadratic regulator* problem, or the LQR problem for short.

In some cases, the controller's goal is to force the system state to track a desired state trajectory, $\boldsymbol{x}_d(t)$, throughout the interval $[t_0, t_f]$ while maintaining the deviations of the actual state $\boldsymbol{x}(t)$ "small" from the desired trajectory with the control effort $\boldsymbol{u}(t)$ "not too large" and the final state $\boldsymbol{x}(t_f)$ being as near as possible to some desired state $\boldsymbol{x}_d(t_f)$. A suitable performance index to be minimized in this case might be

$$\begin{aligned} J \;=\; & \frac{1}{2}\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_d(t_f)\right)^\top \boldsymbol{F}\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_d(t_f)\right) \\ & + \frac{1}{2}\int_{t_0}^{t_f} \Big(\left(\boldsymbol{x}(t) - \boldsymbol{x}_d(t)\right)^\top \boldsymbol{Q}\left(\boldsymbol{x}(t) - \boldsymbol{x}_d(t)\right) + \boldsymbol{u}(t)^\top \boldsymbol{R}\boldsymbol{u}(t)\Big)\, dt. \end{aligned} \tag{9.10}$$

9.2 A Glimpse at the Calculus of Variations

The goal of this section is to provide a brief introduction to the subject of the *calculus of variations*. The calculus of variations deals with minimization or maximization of *functionals*. A functional is a mapping or a transformation that depends on one or more functions and the values of the functional are numbers. Examples of functionals are performance indices that we introduced in the previous section. We now give two more examples of functionals.

Example 9.1 The length l of the arc connecting two given points $A(x_0, y_0)$ and $B(x_1, y_1)$ in the x-y plane, as depicted in Figure 9.1, is an example of a functional. We know that

$$l = l(y(x)) = \int_{x_0}^{x_1} \sqrt{1 + (dy(x)/dx)^2}\,dx,$$

subject to the constraints $y(x_0) = y_0$ and $y(x_1) = y_1$.

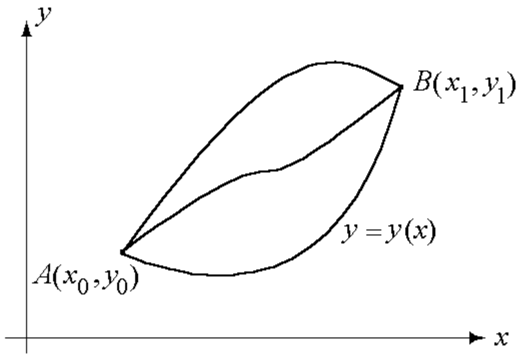


Figure 9.1: The length of arcs joining two given points in the plane is an example of a functional.

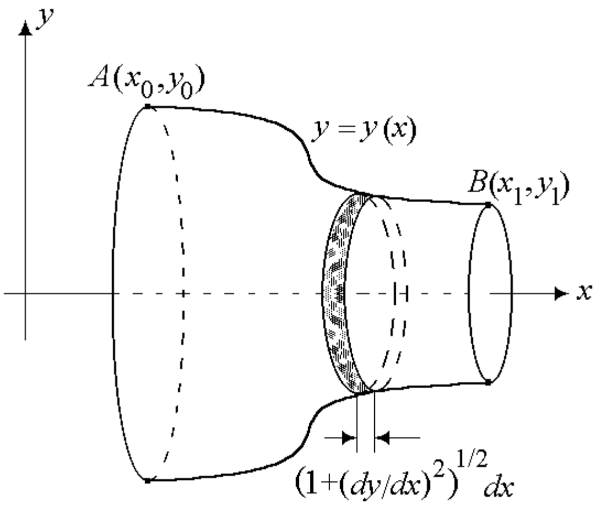


Figure 9.2: The area of the surface of revolution connecting two circular disks is an example of a functional.

Example 9.2 Another example of a functional is the area of the surface of revolution formed by a surface connecting two circular disks as illustrated in Figure 9.2. In particular, the surface of revolution can be formed by a soap film clinging to two circular disks. When a wire circle is dipped into a soap solution and withdrawn, a circular disk of soap film bounded by the circle is formed. If a second smaller circle is made to touch the disk and then moved away, the two circles will be joined by a surface of film that is a surface of revolution in the particular case when the circles are parallel and have their centers on the same axis perpendicular to their planes. If we denote the arc of intersection with the x-y plane by $y = y(x)$, then the area of the surface of revolution is

$$v = v\left(y(x)\right) = 2\pi \int_{x_0}^{x_1} y(x)\sqrt{1 + \left(dy(x)/dx\right)^2}dx,$$

subject to the constraints $y(x_0) = y_0$, $y(x_1) = y_1$. One may then want to minimize $v = v\left(y(x)\right)$ to find $y = y(x)$ resulting in the surface of revolution of minimum area.

The problem of optimization of functionals dates back to Johann Bernoulli. In 1696 he published a letter in which he posed the problem, which is now known as the *brachistochrone problem*. Brachistochrone translates from Greek as the shortest time. The brachistochrone problem can be described as follows. A particle of mass m is to slide down along a frictionless track connecting two given points. The motion of the particle is to be constrained only by gravity and the shape of the curve of the track. Find the curve of the track that minimizes the time travel between the two given points. We illustrate the problem in Figure 9.3. The brachistochrone problem was solved by Johann Bernoulli himself, his brother Jacob, Isaac Newton, the Marquis de L'Hospital, Gottfried Leibniz and Tschirnhaus. A solution to the brachistochrone problem is presented in Example 9.23. In 1697 Johann Bernoulli posed and solved the *geodesics* problem which can be formulated as follows. Find a line of shortest length that connects two given points on a given surface—see Figure 9.4 for an illustration of the problem. The curves of the least length connecting pairs of points on a surface are called *geodesics*. A general procedure for solving the above type of problems, using the calculus of variations, was given by Leonhard Euler (who flourished between 1707 and 1783).

To acquaint ourselves with the methods of the calculus of variations, we need first to introduce concepts concerning functionals.

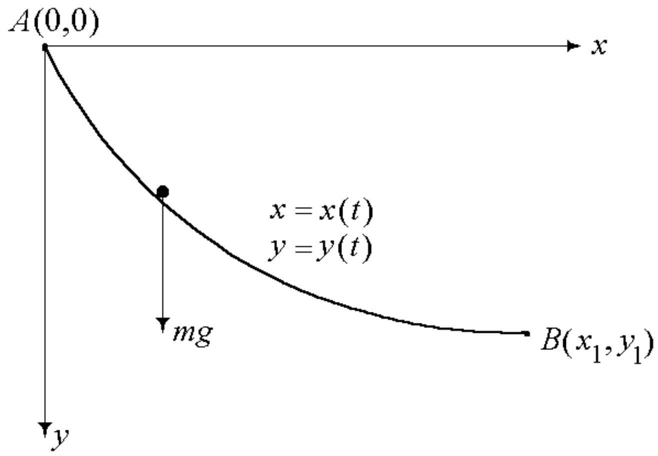


Figure 9.3: Illustration of the brachistochrone problem.

9.2.1 Variation and Its Properties

A *function* can be viewed as a rule, or a mapping, that assigns to each element of some set a unique element of a possibly different set. In particular, a function $x : \mathbb{R} \to \mathbb{R}$ of a real variable t, assigns to each real number a unique real number. An increment of the argument of a function of one variable t is $\Delta t = t - t_1$.

Similarly, a *functional* is a mapping that assigns to each function, from some class of functions, a unique number. Thus, we can say that a functional is a "function of a function." Let $x : \mathbb{R} \to \mathbb{R}$ be an argument of a functional. By a *variation* δx of an argument x of a functional v we mean the difference of two functions

$$\delta x = x - x_1 .$$

We assume that x can change in an arbitrary way in some class of functions.

Recall, that a function $x : \mathbb{R} \to \mathbb{R}$ is continuous if a small change of its argument t corresponds to a small change of the value $x(t)$ of the function. Similarly, a functional v is said to be continuous if a "small" change of its argument x corresponds to a small change of the value of the functional.

A functional v is called linear if

$$v\left(a x_1 + x_2\right) = a v\left(x_1\right) + v\left(x_2\right),$$

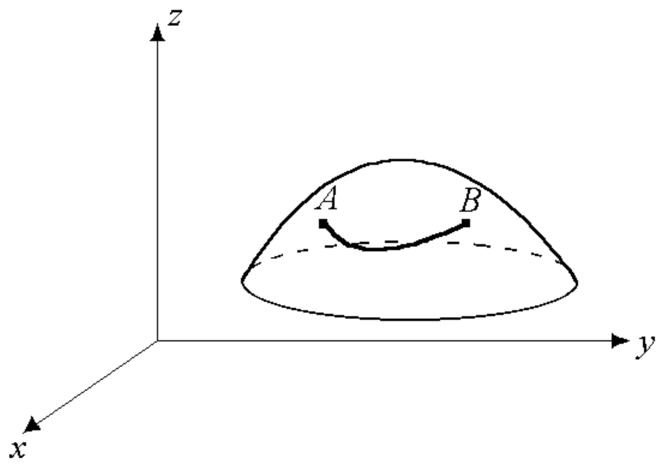


Figure 9.4: The problem of finding geodesics.

where a is a constant.

We will now introduce the notion of the *variation of a functional*. This notion is analogous to the notion of a function differential. To connect the two, we first consider the case of a function of one variable. Let x be a differentiable function defined on an open interval U, and let $t \in U$. The derivative of x at t is defined as

$$x'(t) = \lim_{\Delta t \to 0} \frac{x(t + \Delta t) - x(t)}{\Delta t}. \tag{9.11}$$

Let

$$\varphi(\Delta t) = \frac{x(t + \Delta t) - x(t)}{\Delta t} - x'(t). \tag{9.12}$$

The function φ is not defined at $\Delta t = 0$, however

$$\lim_{\Delta t \to 0} \varphi(\Delta t) = 0.$$

We rewrite (9.12) as

$$x(t + \Delta t) - x(t) = x'(t)\Delta t + \varphi(\Delta t)\Delta t. \tag{9.13}$$

The above relation has meaning only when $\Delta t \neq 0$. To make it hold at $\Delta t = 0$, we define $\varphi(\Delta t)|_{\Delta t=0} = 0$. To proceed, let

$$\begin{aligned} \beta(\Delta t) &= \varphi(\Delta t) \quad \text{if} \quad \Delta t > 0 \\ \beta(\Delta t) &= -\varphi(\Delta t) \quad \text{if} \quad \Delta t < 0. \end{aligned} \tag{9.14}$$

Then, if x is differentiable, there exists a function β such that

$$\begin{aligned} x(t+\Delta t)-x(t)=\Delta x &= x'(t)\Delta t+\beta(\Delta t)|\Delta t| \\ &= L(t,\Delta t)+\beta(\Delta t)|\Delta t|, \end{aligned} \tag{9.15}$$

where $\lim_{\Delta t\to 0}\beta(\Delta t)=0$, and $L(t,\Delta t)=x'(t)\Delta t$ is a linear function in Δt.

For a real-valued function $f=f(\boldsymbol{x}):\mathbb{R}^n\to\mathbb{R}$, relation (9.15) takes the form

$$\begin{aligned} f(\boldsymbol{x}+\Delta\boldsymbol{x})-f(\boldsymbol{x})=\Delta f &= Df(\boldsymbol{x})\Delta\boldsymbol{x}+\beta(\Delta\boldsymbol{x})\|\Delta\boldsymbol{x}\| \\ &= L(\boldsymbol{x},\Delta\boldsymbol{x})+\beta(\Delta\boldsymbol{x})\|\Delta\boldsymbol{x}\|, \end{aligned} \tag{9.16}$$

where $\lim_{\Delta\boldsymbol{x}\to\boldsymbol{0}}\beta(\Delta\boldsymbol{x})=0$, and

$$L(\boldsymbol{x},\Delta\boldsymbol{x})=Df(\boldsymbol{x})\Delta\boldsymbol{x}=\nabla f(\boldsymbol{x})^\top\Delta\boldsymbol{x}$$

is a linear function in $\Delta\boldsymbol{x}$. Relation (9.16) holds for operators defined over Banach spaces with scalars being either real or complex numbers—see Definition A.4 of Banach space on page 994. The functionals we consider are operators over a Banach space of continuous functions, $\mathcal{C}([t_0,t_1])$, with the norm given by (A.99) on page 993.

If an increment $\Delta v=v\,(x+\delta x)-v\,(x)$ of a functional v can be represented as

$$\Delta v=L\,(x,\delta x)+\beta\,(\delta x)\,\|\delta x\|,$$

where $L\,(x,\delta x)$ is a linear functional with respect to δx, the term $\|\delta x\|=\max_{t\in[t_0,t_1]}|\delta x|$ denotes the maximal value of $|\delta x|$, and $\beta\,(\delta x)\to 0$ if $\|\delta x\|\to 0$, then the linear part L of Δv is called the *variation of the functional* and is denoted δv, that is,

$$\delta v=L\,(x,\delta x)\,.$$

Example 9.3 We find the variation of the functional

$$v=\int_0^1\left(2x^2(t)+x(t)\right)dt.$$

For this we first calculate its increment to get

$$\begin{aligned} \Delta v &= v\,(x+\delta x)-v\,(x) \\ &= \int_0^1\left(2(x+\delta x)^2+(x+\delta x)\right)dt-\int_0^1\left(2x^2+x\right)dt \\ &= \int_0^1\left(2x^2+4x\delta x+2(\delta x)^2+x+\delta x-2x^2-x\right)dt \\ &= \int_0^1(4x+1)\delta x dt+2\int_0^1(\delta x)^2dt. \end{aligned}$$

The linear part of Δv is

$$\delta v = \int_0^1 (4x+1)\delta x dt,$$

which is the variation of the given functional.

To proceed, note that the linear part on the right-hand side of (9.16) can be computed as

$$Df(\boldsymbol{x})\Delta\boldsymbol{x} = L(\boldsymbol{x},\Delta\boldsymbol{x}) = \frac{d}{d\alpha} f(\boldsymbol{x}+\alpha\Delta\boldsymbol{x})\bigg|_{\alpha=0} \tag{9.17}$$

As it was mentioned before, the above also holds for operators over Banach spaces other than Euclidean spaces. Using this observation, we state and prove a lemma that allows us to calculate the variation of a given functional more efficiently than the method we used in Example 9.3.

Lemma 9.1

$$\delta v = \frac{d}{d\alpha} v\left(x+\alpha\delta x\right)\bigg|_{\alpha=0}.$$

Proof Suppose that for a given functional v there exists its variation. This means that we can represent Δv as

$$\Delta v = v\left(x+\alpha\delta x\right) - v\left(x\right) = L(x,\alpha\delta x) + \beta(\alpha\delta x)|\alpha|\,\|\delta x\|.$$

Then, the derivative of $v\left(x+\alpha\delta x\right)$ with respect to α evaluated at $\alpha=0$ is equal to

$$\begin{aligned}
\lim_{\alpha\to 0}\frac{\Delta v}{\alpha} &= \lim_{\alpha\to 0}\frac{L(x,\alpha\delta x)+\beta(\alpha\delta x)|\alpha|\,\|\delta x\|}{\alpha} \\
&= \lim_{\alpha\to 0}\frac{L(x,\alpha\delta x)}{\alpha} + \lim_{\alpha\to 0}\frac{\beta(\alpha\delta x)|\alpha|\,\|\delta x\|}{\alpha} \\
&= L(x,\delta x),
\end{aligned}$$

because $L(\cdot,\cdot)$ is linear with respect to the second argument and hence,

$$L(x,\alpha\delta x) = \alpha L(x,\delta x).$$

Furthermore,

$$\lim_{\alpha\to 0}\frac{\beta(\alpha\delta x)|\alpha|\,\|\delta x\|}{\alpha} = \lim_{\alpha\to 0}\beta(\alpha\delta x)\|\delta x\| = 0.$$

This completes the proof.

□

Example 9.4 We use Lemma 9.1 to find the variation of the functional considered in Example 9.3. We have

$$\begin{aligned}
\delta v &= \frac{d}{d\alpha} v\left(x + \alpha\delta x\right)|_{\alpha=0} \\
&= \frac{d}{d\alpha}\left(\int_0^1 \left(2(x+\alpha\delta x)^2 + (x+\alpha\delta x)\right) dt\right)\Bigg|_{\alpha=0} \\
&= \int_0^1 \left(4(x+\alpha\delta x)\delta x + \delta x\right) dt\Bigg|_{\alpha=0} \\
&= \int_0^1 (4x+1)\delta x dt,
\end{aligned}$$

which is the variation of the given functional in Example 9.3.

Example 9.5 We find the variation of the functional

$$v(x_1, x_2) = v(x_1(t), x_2(t)) = \int_0^1 x_1 e^{-x_2} dt,$$

using the formula

$$\delta v = \frac{d}{d\alpha} v(\boldsymbol{x} + \alpha\delta\boldsymbol{x})\Bigg|_{\alpha=0}.$$

We have

$$\begin{aligned}
\delta v &= \frac{d}{d\alpha}\left(\int_0^1 \left(x_1 + \alpha\delta x_1\right) e^{-(x_2+\alpha\delta x_2)} dt\right)\Bigg|_{\alpha=0} \\
&= \int_0^1 \frac{d}{d\alpha}\left(\left(x_1 + \alpha\delta x_1\right) e^{-(x_2+\alpha\delta x_2)} dt\right)\Bigg|_{\alpha=0} \\
&= \int_0^1 \left(\delta x_1 e^{-(x_2+\alpha\delta x_2)} - \delta x_2 e^{-(x_2+\alpha\delta x_2)}(x_1 + \alpha\delta x_1)\right) dt\Bigg|_{\alpha=0} \\
&= \int_0^1 \left(e^{-x_2}\delta x_1 - x_1 e^{-x_2}\delta x_2\right) dt.
\end{aligned}$$

Thus,

$$\delta v = \int_0^1 \left(e^{-x_2}\delta x_1 - x_1 e^{-x_2}\delta x_2\right) dt.$$

Definition 9.1 *A functional v is maximal on some function x_0 if the value of the functional on any function x close to x_0 is less than or equal to that on x_0, that is,*

$$\Delta v = v(x) - v(x_0) \leq 0.$$

◇

We now state a first order necessary condition for a given functional to achieve a maximum or minimum on a curve.

Theorem 9.1 *If for a functional v there exists a variation and the functional reaches a maximum or a minimum on a curve x_0, then*

$$\delta v(x_0) = 0.$$

Proof Suppose that x_0 and δx are fixed. Then

$$v(x_0 + \alpha \delta x) = \Phi(\alpha),$$

that is, $v(x_0 + \alpha\delta x)$ is a function of α that for $\alpha = 0$ achieves its maximum or minimum. This implies that $\frac{d}{d\alpha}\Phi = 0$, or equivalently,

$$\left.\frac{d}{d\alpha} v(x_0 + \alpha\delta x)\right|_{\alpha=0} = 0.$$

Hence, by Lemma 9.1, we have $\delta v = 0$. Therefore, a functional that achieves its maximum or a minimum on a function x_0, has its variation evaluated at x_0 equal to zero.

□

The above results are used in the next section to determine conditions that a curve has to satisfy to be a maximizer or a minimizer, that is, to be an extremal of a given functional.

9.2.2 Euler-Lagrange Equation

In this section, we consider the following optimization problem.

Problem 1 Suppose that we are given two points (t_0, x_0) and (t_1, x_1) in the (t, x)-plane. We wish to find a curve, trajectory, joining the given points such that the functional

$$v(x) = \int_{t_0}^{t_1} F(t, x, \dot{x})\, dt \tag{9.18}$$

along this trajectory can achieve its extremal, that is, maximal or minimal, value. The problem is illustrated in Figure 9.5. In solving Problem 1, we use the following lemma.

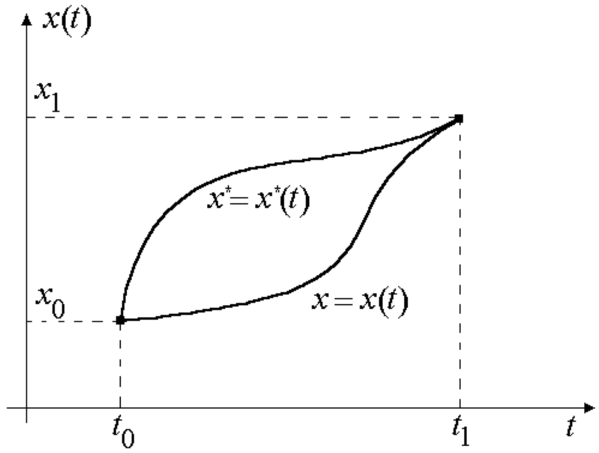


Figure 9.5: Illustration of Problem 1.

Lemma 9.2 *Suppose that a function $\phi : \mathbb{R} \to \mathbb{R}$ is continuous on the interval $[t_0, t_1]$. Then,*

$$\int_{t_0}^{t_1} \phi(t)\delta x(t)dt = 0$$

if and only if $\phi(t) = 0$ at every point of the interval $[t_0, t_1]$.

Proof Sufficiency is obvious because we can satisfy $\int_{t_0}^{t_1} \phi(t)\delta x(t)dt = 0$ by choosing ϕ that vanishes at every point of the interval $[t_0, t_1]$.

We prove necessity by contraposition. We thus prove that if ϕ is nonzero on some subinterval of $[t_0, t_1]$, then $\int_{t_0}^{t_1} \phi(t)\delta x(t)dt \neq 0$. Specifically, assume, without loss of generality, that ϕ is positive on the interval $[a, b]$, where $t_0 \leq a \leq b \leq t_1$. Note that if ϕ were nonzero at only one point, then because it is continuous, it would have to be nonzero in some neighborhood of that point. Hence, we can assume that $a < b$. Consider a variation given by

$$\delta x = \delta x(t) = \begin{cases} k^2(t-a)^2(t-b)^2 & \text{for} \quad a \leq t \leq b \\ 0 & \text{elsewhere.} \end{cases} \tag{9.19}$$

If we assumed ϕ to be negative on the interval $[a, b]$, then we would use δx to be negative of (9.19). We illustrate δx in Figure 9.6. Because δx is positive in $[a, b]$ and zero elsewhere

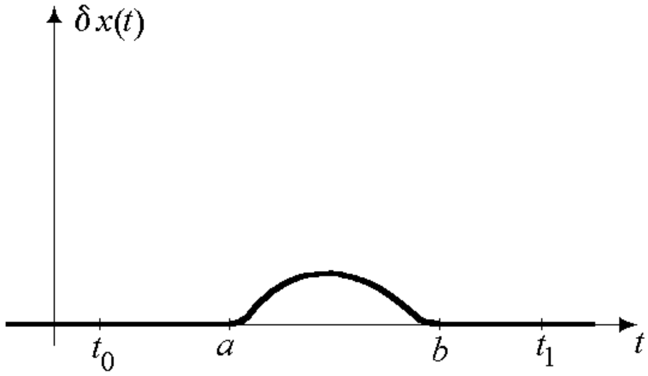


Figure 9.6: Illustration of the proof of Lemma 9.2.

in the interval $[t_0,\, t_1]$, the integrand of $\int_{t_0}^{t_1} \phi(t)\delta x(t)dt$ will have the same sign as ϕ in $[a,\, b]$ and will be zero outside this subinterval. Thus

$$\int_{t_0}^{t_1} \phi(t)\delta x(t)dt = \int_a^b \phi(t)\delta x(t)dt \neq 0,$$

and the proof is complete.

□

Solution to Problem 1 We assume that there is an optimal trajectory $x = x(t)$ joining the points (t_0, x_0) and (t_1, x_1) such that $\delta v(x(t)) = 0$. Let us then consider an arbitrary acceptable curve $x^* = x^*(t)$ that is close to x and from a family of curves

$$x(t, \alpha) = x(t) + \alpha\,(x^*(t) - x(t)) = x(t) + \alpha\delta x(t).$$

Note that for $\alpha = 0$ we get $x(t)$, and for $\alpha = 1$ we obtain $x^*(t)$. Furthermore,

$$\frac{d}{dt}(\delta x(t)) = \frac{d}{dt}\,(x^*(t) - x(t)) = \delta x',$$

and analogously

$$\frac{d^2}{dt^2}(\delta x(t)) = \frac{d^2}{dt^2}\,(x^*(t) - x(t)) = \delta x''.$$

We are now ready to investigate the behavior of the functional v given by (9.18) on the curves from the family $x(t, \alpha)$. Note that the functional v can be considered as a function of α, that is, $v(x(t, \alpha)) = \Phi(\alpha)$. It follows from the first order necessary condition for a point to be an extremizer of a function of one variable that $\alpha = 0$ is a candidate to be an extremizer of Φ if

$$\left.\frac{d\Phi(\alpha)}{d\alpha}\right|_{\alpha=0} = 0,$$

where

$$\Phi(\alpha) = \int_{t_0}^{t_1} F(t, x(t, \alpha), x'(t, \alpha))\, dt,$$

and $x'(t, \alpha) = \frac{d}{dt}x(t, \alpha)$. We evaluate $\frac{d\Phi(\alpha)}{d\alpha}$ to obtain

$$\frac{d\Phi(\alpha)}{d\alpha} = \int_{t_0}^{t_1} \left(F_x \frac{d}{d\alpha} x(t, \alpha) + F_{x'} \frac{d}{d\alpha} x'(t, \alpha) \right) dt,$$

where $F_x = \frac{\partial}{\partial x} F(t, x(t, \alpha), x'(t, \alpha))$, and $F_{x'} = \frac{\partial}{\partial x'} F(t, x(t, \alpha), x'(t, \alpha))$. Because

$$\frac{d}{d\alpha} x(t, \alpha) = \frac{d}{d\alpha}(x(t) + \alpha \delta x(t)) = \delta x(t),$$

and

$$\frac{d}{d\alpha} x'(t, \alpha) = \frac{d}{d\alpha}(x'(t) + \alpha \delta x'(t)) = \delta x'(t),$$

we can write

$$\frac{d}{d\alpha}\Phi(\alpha) = \int_{t_0}^{t_1} (F_x(t, x(t, \alpha), x'(t, \alpha))\delta x(t) + F_{x'}(t, x(t, \alpha), x'(t, \alpha))\delta x'(t))\, dt.$$

By Lemma 9.1,

$$\frac{d}{d\alpha}\Phi(0) = \delta v.$$

Therefore,

$$\delta v = \int_{t_0}^{t_1} (F_x \delta x + F_{x'} \delta x')\, dt,$$

where for convenience we dropped the argument t. Integrating by parts the last term, and taking into account that $\delta x' = (\delta x)'$ yields

$$\delta v = (F_{x'}\delta x)|_{t_0}^{t_1} + \int_{t_0}^{t_1} \left(F_x - \frac{d}{dt} F_{x'} \right) \delta x dt.$$

The end points (t_0, x_0) and (t_1, x_1) are fixed. This implies that

$$\delta x|_{t=t_0} = x^*(t_0) - x(t_0) = 0,$$

and

$$\delta x|_{t=t_1} = x^*(t_1) - x(t_1) = 0.$$

Hence

$$\delta v = \int_{t_0}^{t_1} \left(F_x - \frac{d}{dt} F_{x'} \right) \delta x dt.$$

By Lemma 9.2,

$$\delta v = 0 \quad \text{if and only if} \quad F_x - \frac{d}{dt} F_{x'} = 0.$$

The equation,

$$\boxed{F_x - \frac{d}{dt} F_{x'} = 0} \tag{9.20}$$

is known as the Euler-Lagrange equation. The curves $x = x(t, C_1, C_2)$ that are solutions to the Euler-Lagrange equations are called *extremals*. The constants C_1 and C_2 are the integration constants. Thus, a functional can only achieve its extremum on the extremals.

The Euler-Lagrange equation $F_x - \frac{d}{dt} F_{x'} = 0$ can also be represented as

$$F_x - F_{x't} - F_{x'x} x' - F_{x'x'} x'' = 0.$$

Example 9.6 We find the extremal of the functional

$$v = \int_0^{\frac{\pi}{2}} \left((x')^2 - x^2 \right) dt, \tag{9.21}$$

where $x(0) = 0$, and $x\left(\frac{\pi}{2}\right) = 1$. The Euler-Lagrange equation for the problem takes the form

$$\begin{aligned} F_x - \frac{d}{dt} F_{x'} &= \frac{\partial}{\partial x} \left((x')^2 - x^2 \right) - \frac{d}{dt} \left(\frac{\partial}{\partial x'} \left((x')^2 - x^2 \right) \right) \\ &= -2x - \frac{d}{dt} (2x') \\ &= -2x - 2x'' \\ &= 0. \end{aligned}$$

Equivalently, the Euler-Lagrange equation for the functional (9.21) can be represented as

$$x'' + x = 0.$$

The characteristic equation of the above differential equation is $s^2 + 1 = 0$. Therefore, the general solution of the Euler-Lagrange equation has the form

$$x(t) = C_1 \cos t + C_2 \sin t.$$

Taking into account the end-point conditions gives

$$\begin{aligned} x(0) &= C_1 = 0, \\ x\left(\frac{\pi}{2}\right) &= C_2 = 1. \end{aligned}$$

Thus, we have one extremal

$$x(t) = \sin t.$$

The functional (9.21) can be minimized or maximized only on the extremal $x(t) = \sin t$.

Example 9.7 We find the extremal of the functional

$$v = \int_0^1 \big((x')^2 + 12xt\big)\, dt, \tag{9.22}$$

where $x(0) = 0$, and $x(1) = 1$. The Euler-Lagrange equation for the this functional is

$$\begin{aligned} F_x - \frac{d}{dt}F_{x'} &= \frac{\partial}{\partial x}\big((x')^2 + 12xt\big) - \frac{d}{dt}\left(\frac{\partial}{\partial x'}\big((x')^2 + 12xt\big)\right) \\ &= 12t - \frac{d}{dt}(2x') \\ &= 12t - 2x'' \\ &= 0. \end{aligned}$$

Equivalently, the Euler-Lagrange equation for the functional (9.22) can be represented as

$$x'' = 6t.$$

The general solution to the above differential equation is

$$x(t) = C_1 t + C_2 + t^3.$$

From the end-point conditions, we get

$$\begin{aligned} x(0) &= C_2 = 0, \\ x(1) &= C_1 + 1 = 1. \end{aligned}$$

Hence, $C_1 = 0$, and the extremal has the form

$$x(t) = t^3.$$

Suppose now that $A = A(t_0, x_{10}, \dots, x_{n0})$ and $B = B(t_1, x_{11}, \dots, x_{n1})$ are two given points in the $(n+1)$-dimensional space of $(n+1)$-tuples, $(t, x_1, \dots, x_n)$, along with the functional

$$v = v(x_1, \dots, x_n) = \int_{t_0}^{t_1} F(t, x_1, \dots, x_n, \dot{x}_1, \dots, \dot{x}_n) dt.$$

To find a trajectory along which $\delta v = 0$, we consider only the variation in one coordinate, say x_i, while other coordinates are fixed. We perform this process for each coordinate. The functional v will depend only upon x_i, and we write

$$v(x_1, \dots, x_i, \dots, x_n) = \tilde{v}(x_i).$$

Thus, we reduced the given n-dimensional problem to n one-dimensional problems. Proceeding as before, we conclude that the functional $v = v(x_1, \dots, x_n)$ can only be extremized on the trajectories that satisfy n Euler-Lagrange equations,

$$F_{x_i} - \frac{d}{dt} F_{\dot{x}_i} = 0, \quad i = 1, \dots, n. \tag{9.23}$$

Observe that if in (9.23) we set $F = L = K - U$, where L is the Lagrangian, $K = K(\dot{q}_1, \dots, \dot{q}_n)$ is the kinetic energy of the particle, $U = U(q_1, \dots, q_n)$ is its potential energy, and $x_i = q_i$, then (9.23) becomes

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) = 0, \quad i = 1, 2, \dots, n,$$

which are the Lagrange equations of motion for a particle. Referring to the solution to Problem 1 on page 443, we conclude that a particle moves along a path on which the functional

$$J = \int_{t_1}^{t_2} L dt,$$

is extremized, which agrees with the Hamilton principle saying that a particle moves along the path that minimizes the integral J.

In our discussion up to now, both end-points were fixed. We would like to solve a more general optimization problem, where the end-points need not be fixed. For simplicity, we assume first that one of the end-points is fixed, while the other is free. After we solve this problem, we can easily generalize to the case when both end-points are free.

Problem 2 In the (t, x)-plane the left-end point A is fixed, while the right end-point B is free—see Figure 9.7. It is required to find a trajectory $x = x(t)$ starting at A on which the functional $v = \int_{t_0}^{t_1} F(t, x, x') dt$ can achieve its extremal value.

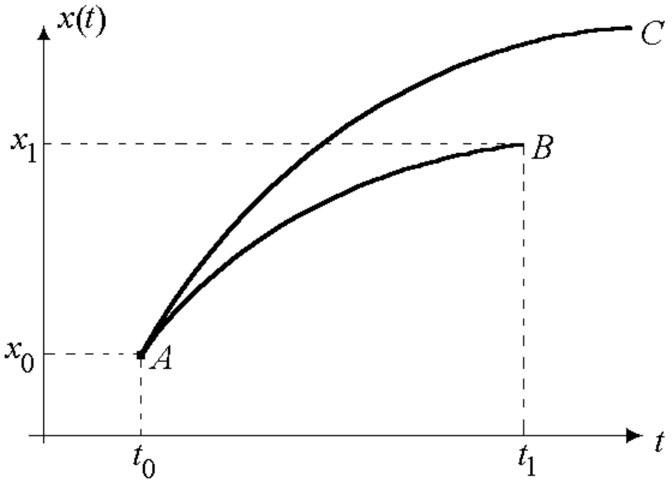


Figure 9.7: Illustration of Problem 2.

Solution to Problem 2 Following Elsgolc [96, pages 64–72], we begin by assuming that a solution to Problem 2 exists, that is, there is a trajectory $x = x(t)$ joining (t_0, x_0) and (t_1, x_1) such that $\delta v = 0$. This trajectory must satisfy the Euler-Lagrange equation. Consider an adjoining trajectory specified by $x + \delta x$ joining (t_0, x_0) and $(t_1 + \delta t_1, x_1 + \delta x_1)$. Then,

$$\begin{aligned}
\Delta v &= v(x+\delta x) - v(x) \\
&= \int_{t_0}^{t_1+\delta t_1} F(t, x+\delta x, x'+\delta x')dt - \int_{t_0}^{t_1} F(t, x, x')dt \\
&= \int_{t_1}^{t_1+\delta t_1} F(t, x+\delta x, x'+\delta x')dt + \int_{t_0}^{t_1} \left(F(t, x+\delta x, x'+\delta x') - F(t, x, x')\right) dt.
\end{aligned}$$

Our goal is to extract from Δv its linear portion in δx, which we know is the variation of the given functional. We begin by approximating the first integral as

$$\int_{t_1}^{t_1+\delta t_1} F(t, x+\delta x, x'+\delta x')dt \approx F(t, x, x')\big|_{t=t_1} \delta t_1.$$

Expanding the integrand of the second integral into a Taylor series and retaining only first-order terms yields

$$\int_{t_0}^{t_1} (F(t, x + \delta x, x' + \delta x') - F(t, x, x'))\, dt = \int_{t_0}^{t_1} (F_x(t, x, x')\delta x + F_{x'}(t, x, x')\delta x')\, dt.$$

Integrating the second term by parts and combining with the first term gives

$$\int_{t_0}^{t_1} (F_x \delta x + F_{x'} \delta x')\, dt = (F_{x'} \delta x)\, |_{t_0}^{t_1} + \int_{t_0}^{t_1} \left(F_x - \frac{d}{dt} F_{x'} \right) \delta x dt,$$

where for simplicity of notation we suppressed the arguments of F_x and $F_{x'}$. We assumed that a solution to the problem exists. Hence, an extremal must satisfy the Euler-Lagrange equation $F_x - \frac{d}{dt} F_{x'} = 0$. Because the left end-point is fixed, we obtain

$$\delta x|_{t=t_0} = 0.$$

Taking into account the results of the above manipulations yields

$$\Delta v \approx F(t, x, x')|_{t=t_1} \delta t_1 + (F_{x'} \delta x)|_{t=t_1}. \tag{9.24}$$

The final step in obtaining δv is to evaluate $(F_{x'} \delta x)\, |_{t_1}$. The problem is two dimensional so we can illustrate it graphically as shown in Figure 9.8. Referring to Figure 9.8, we obtain

$$\begin{aligned} DB &= \delta x|_{t=t_1}, \\ FC &= \delta x_1, \\ EC &\approx x'(t_1)\delta t_1, \\ DB &= FC - EC. \end{aligned}$$

Hence,

$$\delta x|_{t=t_1} \approx \delta x_1 - x'(t_1)\delta t_1.$$

Substituting the above into (9.24) gives

$$\begin{aligned} \delta v &= F(t, x, x')|_{t=t_1} \delta t_1 + F_{x'}|_{t=t_1} (\delta x_1 - x'(t_1)\delta t_1) \\ &= (F - x' F_{x'})\, |_{t=t_1} \delta t_1 + F_{x'}|_{t=t_1} \delta x_1. \end{aligned}$$

If the variations δt_1 and δx_1 are arbitrary, we obtain $\delta v = 0$ if in addition to the Euler-Lagrange equation the following end-point conditions are also satisfied,

$$\boxed{(F - x' F_{x'})\, |_{t=t_1} = 0 \quad \text{and} \quad F_{x'}|_{t=t_1} = 0}$$

The above end-point relations are called the *transversality conditions*. In conclusion, the optimal trajectory must satisfy both the Euler-Lagrange equation and the transversality conditions.

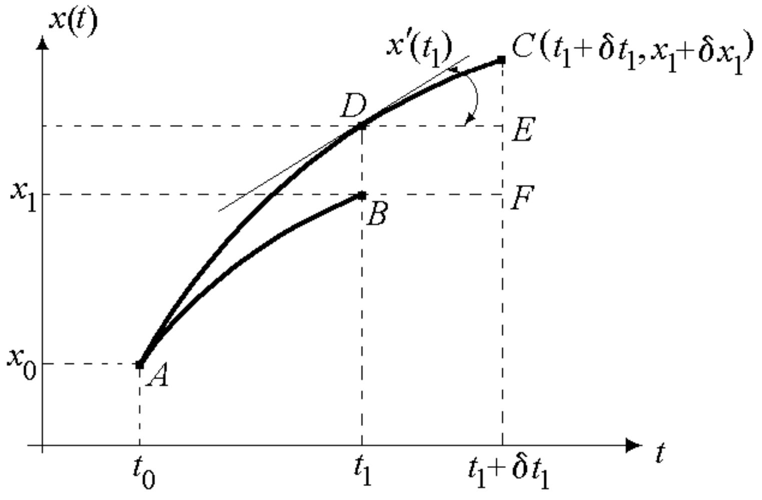


Figure 9.8: A geometric interpretation of calculating the variation of the functional for Problem 2.

Example 9.8 Consider the functional of Example 9.7, where the left end-point is fixed as before, that is, $x(0) = 0$, while the right end-point $x(1)$ is free. Here, we have $t_1 = 1$. Because $\delta t_1 = 0$ and δx_1 is arbitrary, the transversality conditions for the right end-point are reduced to one condition,

$$F_{x'}|_{t=t_1} = 0.$$

Taking into account the solution to the Euler-Lagrange equation obtained in Example 9.7, and then using the fact that the left end-point is fixed, we obtain

$$\begin{aligned} 0 &= F_{x'}|_{t=t_1} \\ &= 2x'(t_1) \\ &= 2(C_1 + 3t^2)|_{t=1} \\ &= 2(C_1 + 3). \end{aligned}$$

Hence, $C_1 = -3$ and the extremal for this case is

$$x(t) = -3t + t^3.$$

In Figure 9.9, we show a plot of the above extremal.

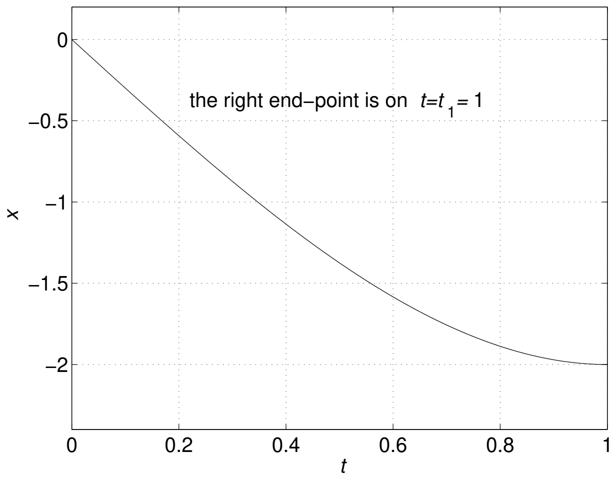


Figure 9.9: A plot of the extremal, $x(t) = -3t + t^3$, of Example 9.8.

Example 9.9 We consider again the same functional as in Example 9.7, where, this time, the left end-point is fixed, as before, while at the right-end point t_1 is free and $x(t_1) = 1$. Hence, in this case, $\delta x_1 = 0$ while δt_1 is arbitrary. The transversality conditions reduce to one condition,

$$(F - x' F_{x'})\,|_{t=t_1} = 0.$$

We perform simple calculations to obtain,

$$F - x' F_{x'} = 12xt - (x')^2,$$

where

$$x' = \left(C_1 t + t^3\right)' = C_1 + 3t^2.$$

Note that because $x(t_1) = C_1 t_1 + t_1^3 = 1$, we have

$$C_1 = \frac{1 - t_1^3}{t_1}.$$

Taking into account the above and the fact that $x(t_1) = 1$, we perform simple algebraic manipulations to arrive at the following equation for t_1,

$$(F - x' F_{x'})\,|_{t=t_1} = \left(12x(t_1)t_1 - (x'(t_1))^2\right) = -4t_1^6 + 8t_1^3 - 1 = 0.$$

Solving the above equation, we obtain the following two solutions,

$$t_1 = \left(1 \pm \sqrt{\frac{3}{4}}\right)^{1/3}.$$

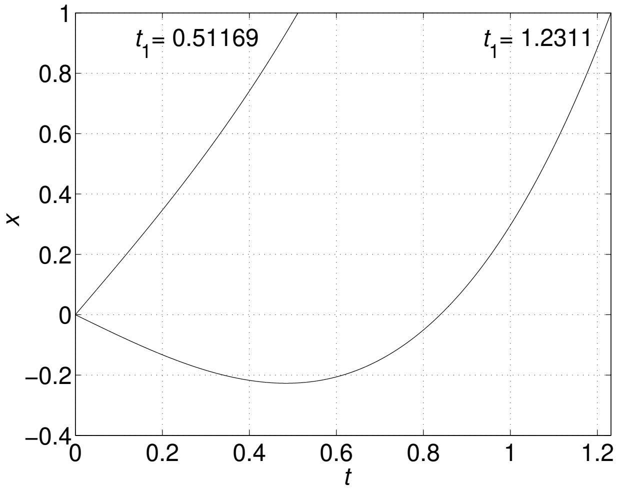


Figure 9.10: Plots of the extremals of Example 9.9.

Plots of the two extremals for this example are shown in Figure 9.10.

If the position of the end point B is restricted to a curve $x = g_1(t)$, then

$$\delta x_1 \approx g_1'(t_1)\delta t_1,$$

and hence

$$\delta v = \left(F + \left(g_1' - x'\right) F_{x'}\right) \delta t_1 = 0$$

if

$$\left(F + \left(g_1' - x'\right) F_{x'}\right) |_{t=t_1} = 0. \tag{9.25}$$

The above is the transversality condition for the case when the end-point B is restricted to the curve $x = g_1(t)$.

In the next section, we discuss a specific problem of minimizing a functional subject to constraints in the form of differential equations.

9.3 Linear Quadratic Regulator

9.3.1 Algebraic Riccati Equation (ARE)

Consider a linear system modeled by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$

and the associated performance index

$$J=\int_0^\infty \left(\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}+\boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}\right)dt,$$

where $\boldsymbol{Q}=\boldsymbol{Q}^\top \ge 0$ and $\boldsymbol{R}=\boldsymbol{R}^\top > 0$. Our goal is to construct a stabilizing linear state-feedback controller of the form $\boldsymbol{u}=-\boldsymbol{K}\boldsymbol{x}$ that minimizes the performance index J. We denote such a linear control law by $\boldsymbol{u}^*$.

First, we assume that a linear state-feedback optimal controller exists such that the optimal closed-loop system

$$\dot{\boldsymbol{x}}=(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K})\boldsymbol{x}$$

is asymptotically stable. The above assumption implies that there is a Lyapunov function $V=\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{x}$ for the closed-loop system, that is, for some positive definite matrix $\boldsymbol{P}$ the time derivative dV/dt evaluated on the trajectories of the closed-loop system is negative definite.

Theorem 9.2 *If the state-feedback controller $\boldsymbol{u}^*=-\boldsymbol{K}\boldsymbol{x}$ is such that*

$$\min_{\boldsymbol{u}}\left(\frac{dV}{dt}+\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{u}^\top\boldsymbol{R}\boldsymbol{u}\right)=0, \tag{9.26}$$

for some $V=\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{x}$, then the controller is optimal.

Proof We can rewrite the condition of the theorem as

$$\left.\frac{dV}{dt}\right|_{\boldsymbol{u}=\boldsymbol{u}^*}+\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{u}^{*\top}\boldsymbol{R}\boldsymbol{u}^*=0.$$

Hence,

$$\left.\frac{dV}{dt}\right|_{\boldsymbol{u}=\boldsymbol{u}^*}=-\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}-\boldsymbol{u}^{*\top}\boldsymbol{R}\boldsymbol{u}^*.$$

Integrating both sides of the resulting equation with respect to time from 0 to ∞, we obtain

$$V(\boldsymbol{x}(\infty))-V(\boldsymbol{x}(0))=-\int_0^\infty\left(\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{u}^{*\top}\boldsymbol{R}\boldsymbol{u}^*\right)dt.$$

Because by assumption the closed-loop system is asymptotically stable, $\boldsymbol{x}(\infty)=\boldsymbol{0}$, and therefore

$$V(\boldsymbol{x}(0))=\boldsymbol{x}_0^\top\boldsymbol{P}\boldsymbol{x}_0=\int_0^\infty\left(\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{u}^{*\top}\boldsymbol{R}\boldsymbol{u}^*\right)dt.$$

Thus, we showed that if a linear state-feedback controller satisfies the assumption of the theorem, then the value of the performance index for such a controller is

$$J\left(\boldsymbol{u}^*\right)=\boldsymbol{x}_0^\top\boldsymbol{P}\boldsymbol{x}_0.$$

To show that such a controller is indeed optimal, we use a proof by contradiction. We assume that (9.26) holds and that $\boldsymbol{u}^*$ is not optimal. Suppose that $\tilde{\boldsymbol{u}}$ yields a smaller value of J, that is,

$$J(\tilde{\boldsymbol{u}}) < J(\boldsymbol{u}^*).$$

It follows from (9.26) that

$$\left.\frac{dV}{dt}\right|_{\boldsymbol{u}=\tilde{\boldsymbol{u}}} + \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \tilde{\boldsymbol{u}}^\top \boldsymbol{R}\tilde{\boldsymbol{u}} \geq 0,$$

that is,

$$\left.\frac{dV}{dt}\right|_{\boldsymbol{u}=\tilde{\boldsymbol{u}}} \geq -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \tilde{\boldsymbol{u}}^\top \boldsymbol{R}\tilde{\boldsymbol{u}}.$$

Integrating the above with respect to time from 0 to ∞ yields

$$V(\boldsymbol{x}(0)) \leq \int_0^\infty \left(\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \tilde{\boldsymbol{u}}^\top \boldsymbol{R}\tilde{\boldsymbol{u}}\right) dt$$

implying that

$$J(\boldsymbol{u}^*) \leq J(\tilde{\boldsymbol{u}}),$$

which is a contradiction, and the proof is complete.

$\square$

It follows from the above theorem that the synthesis of the optimal control law involves finding an appropriate Lyapunov function, or equivalently, the matrix $\boldsymbol{P}$. The appropriate $\boldsymbol{P}$ is found by minimizing

$$f(\boldsymbol{u}) = \frac{dV}{dt} + \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}, \tag{9.27}$$

We first apply to (9.27) the necessary condition for unconstrained minimization,

$$\left.\frac{\partial}{\partial \boldsymbol{u}}\left(\frac{dV}{dt} + \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}\right)\right|_{\boldsymbol{u}=\boldsymbol{u}^*} = \boldsymbol{0}.$$

Differentiating the above yields

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{u}}\left(\frac{dV}{dt} + \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}\right) &= \frac{\partial}{\partial \boldsymbol{u}}\left(2\boldsymbol{x}^\top \boldsymbol{P}\dot{\boldsymbol{x}} + \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}\right) \\
&= \frac{\partial}{\partial \boldsymbol{u}}\left(2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{u} + \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}\right) \\
&= 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B} + 2\boldsymbol{u}^\top \boldsymbol{R} \\
&= \boldsymbol{0}.
\end{aligned}$$

Hence, the candidate for an optimal control law has the form

$$\boldsymbol{u}^* = -\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{x} = -\boldsymbol{K}\boldsymbol{x}, \tag{9.28}$$

where $\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}$. Note that

$$\begin{aligned}
\frac{\partial^2}{\partial \boldsymbol{u}^2}\left(\frac{dV}{dt} + \boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top\boldsymbol{R}\boldsymbol{u}\right) &= \frac{\partial^2}{\partial \boldsymbol{u}^2}\left(2\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{B}\boldsymbol{u} + \boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top\boldsymbol{R}\boldsymbol{u}\right)\\
&= \frac{\partial}{\partial \boldsymbol{u}}\left(2\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{B} + 2\boldsymbol{u}^\top\boldsymbol{R}\right)\\
&= 2\boldsymbol{R}\\
&> 0.
\end{aligned}$$

Thus the second order sufficiency condition for $\boldsymbol{u}^*$ to minimize (9.27) is satisfied.

We next turn our attention to finding an appropriate matrix $\boldsymbol{P}$. The optimal closed-loop system has the form

$$\dot{\boldsymbol{x}} = \left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}\right)\boldsymbol{x}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0.$$

Our optimal controller satisfies the equation

$$\left.\frac{dV}{dt}\right|_{\boldsymbol{u}=\boldsymbol{u}^*} + \boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^{*\top}\boldsymbol{R}\boldsymbol{u}^* = 0,$$

that is,

$$2\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{B}\boldsymbol{u}^* + \boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^{*\top}\boldsymbol{R}\boldsymbol{u}^* = 0.$$

We substitute the expression for $\boldsymbol{u}^*$, given by (9.28), into the above equation and represent it as

$$\boldsymbol{x}^\top\left(\boldsymbol{A}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}\right)\boldsymbol{x} - 2\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{x} + \boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{x} = 0.$$

Factoring out $\boldsymbol{x}$ yields

$$\boldsymbol{x}^\top\left(\boldsymbol{A}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}\right)\boldsymbol{x} = 0.$$

The above equation should hold for any $\boldsymbol{x}$. For this to be true we have to have

$$\boxed{\boldsymbol{A}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P} = \boldsymbol{O}} \tag{9.29}$$

The above equation is referred to as the *algebraic Riccati equation* (ARE). In conclusion, the synthesis of the optimal linear state feedback controller minimizing the performance index

$$J = \int_0^\infty \left(\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top\boldsymbol{R}\boldsymbol{u}\right)dt$$

subject to

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0$$

requires solving the ARE given by (9.29).

Example 9.10 Consider the following model of a dynamical system:

$$\dot{x} = 2u_1 + 2u_2, \qquad x(0) = 3,$$

and the associated performance index

$$J = \int_0^\infty \left(x^2 + ru_1^2 + ru_2^2\right) dt,$$

where $r > 0$ is a parameter.

(i) We first find the solution to the ARE corresponding to the linear state-feedback optimal controller. We have

$$\boldsymbol{A} = 0, \quad \boldsymbol{B} = \begin{bmatrix} 2 & 2 \end{bmatrix}, \quad \boldsymbol{Q} = 1, \quad \boldsymbol{R} = r\boldsymbol{I}_2.$$

The ARE for this problem is

$$\boldsymbol{O} = \boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P} = 1 - \frac{8}{r}p^2,$$

whose solution is

$$p = \sqrt{\frac{r}{8}}.$$

(ii) We write the equation of the closed-loop system driven by the optimal controller,

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{x} = -\frac{1}{\sqrt{2r}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} x.$$

The closed-loop optimal system is described by

$$\dot{x} = \begin{bmatrix} 2 & 2 \end{bmatrix} \boldsymbol{u} = -\frac{4}{\sqrt{2r}} x.$$

(iii) Finally, we find the value of J for the optimal closed-loop system. We have

$$J = \boldsymbol{x}(0)^\top \boldsymbol{P}\boldsymbol{x}(0) = \frac{9}{2}\sqrt{\frac{r}{2}}.$$

We can directly verify that $J = \frac{9}{2}\sqrt{\frac{r}{2}}$ as follows:

$$\begin{aligned} \min J &= \min \int_0^\infty \left(x^2 + ru_1^2 + ru_2^2\right) dt \\ &= \int_0^\infty 9 \exp\left(-\frac{8t}{\sqrt{2r}}\right)(1+1)\, dt = 9\frac{\sqrt{2r}}{4} = \frac{9}{2}\sqrt{\frac{r}{2}}. \end{aligned}$$

A major difficulty when solving the ARE is that it is a nonlinear equation. In the next section, we present an efficient method for solving the ARE.

9.3.2 Solving the ARE Using the Eigenvector Method

We now present a method for solving the ARE referred to as the MacFarlane and Potter method, or the eigenvector method. In our discussion of the method, we follow the development of Power and Simpson [278]. We begin by representing the ARE in the form

$$\begin{bmatrix} \boldsymbol{P} & -\boldsymbol{I}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \\ -\boldsymbol{Q} & -\boldsymbol{A}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_n \\ \boldsymbol{P} \end{bmatrix} = \boldsymbol{O}. \tag{9.30}$$

The $2n \times 2n$ matrix in the middle is called the *Hamiltonian matrix*. We use the symbol $\boldsymbol{H}$ to denote the Hamiltonian matrix, that is,

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \\ -\boldsymbol{Q} & -\boldsymbol{A}^\top \end{bmatrix}.$$

Thus, the ARE can be represented as

$$\begin{bmatrix} \boldsymbol{P} & -\boldsymbol{I}_n \end{bmatrix} \boldsymbol{H} \begin{bmatrix} \boldsymbol{I}_n \\ \boldsymbol{P} \end{bmatrix} = \boldsymbol{O}.$$

We premultiply the above equation by $\boldsymbol{X}^{-1}$, and then postmultiply it by $\boldsymbol{X}$, where $\boldsymbol{X}$ is a nonsingular $n \times n$ matrix,

$$\begin{bmatrix} \boldsymbol{X}^{-1}\boldsymbol{P} & -\boldsymbol{X}^{-1} \end{bmatrix} \boldsymbol{H} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{P}\boldsymbol{X} \end{bmatrix} = \boldsymbol{O}. \tag{9.31}$$

Observe that if we can find matrices $\boldsymbol{X}$ and $\boldsymbol{P}\boldsymbol{X}$ such that

$$\boldsymbol{H} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{P}\boldsymbol{X} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{P}\boldsymbol{X} \end{bmatrix} \Lambda,$$

then equation (9.31) becomes

$$\begin{bmatrix} \boldsymbol{X}^{-1}\boldsymbol{P} & -\boldsymbol{X}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{P}\boldsymbol{X} \end{bmatrix} \Lambda = \boldsymbol{O}.$$

We have thus reduced the problem of solving the ARE to that of constructing appropriate matrices $\boldsymbol{X}$ and $\boldsymbol{PX}$. To proceed further, let $\boldsymbol{v}_i$ be an eigenvector of $\boldsymbol{H}$ and s_i the corresponding eigenvalue, then

$$\boldsymbol{H}\boldsymbol{v}_i = s_i\boldsymbol{v}_i.$$

In our further discussion, we assume that $\boldsymbol{H}$ has at least n distinct real eigenvalues among its $2n$ eigenvalues. (The results obtained can be generalized for the case when the eigenvalues of $\boldsymbol{H}$ are complex or non-distinct.) Thus, we can write

$$\boldsymbol{H}\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix}\begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & s_n \end{bmatrix}.$$

Let

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{PX} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix},$$

and

$$\boldsymbol{\Lambda} = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & s_n \end{bmatrix}.$$

The above choice of $\boldsymbol{X}$ and $\boldsymbol{PX}$ constitutes a possible solution to the equation

$$\begin{bmatrix} \boldsymbol{X}^{-1}\boldsymbol{P} & -\boldsymbol{X}^{-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{PX} \end{bmatrix}\Lambda = \boldsymbol{O}.$$

To construct $\boldsymbol{P}$, we partition the $2n \times n$ eigenvector matrix $\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix}$ into two $n \times n$ submatrices as follows

$$\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{W} \\ \boldsymbol{Z} \end{bmatrix}.$$

Thus,

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{PX} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W} \\ \boldsymbol{Z} \end{bmatrix}.$$

Taking $\boldsymbol{X} = \boldsymbol{W}$, $\boldsymbol{PX} = \boldsymbol{Z}$, and assuming that $\boldsymbol{W}$ is invertible we obtain

$$\boxed{\boldsymbol{P} = \boldsymbol{Z}\boldsymbol{W}^{-1}} \tag{9.32}$$

We now have to decide what set of n eigenvalues should be chosen from those of $\boldsymbol{H}$ to construct the particular $\boldsymbol{P}$. In the case when all $2n$ eigenvalues of $\boldsymbol{H}$ are distinct, the number of different matrices $\boldsymbol{P}$ generated by the above described method is

$$\frac{(2n)!}{(n!)^2}.$$

Let $\boldsymbol{Q} = \boldsymbol{C}^\top \boldsymbol{C}$ be a full rank factorization of $\boldsymbol{Q}$, as defined on page 970. It follows from Theorem 3 of Kučera [196, page 346] that the Hamiltonian matrix $\boldsymbol{H}$ has n eigenvalues in the open left-half complex plane and n in the open right-half plane if and only if the pair $(\boldsymbol{A}, \boldsymbol{B})$ is stabilizable and the pair $(\boldsymbol{A}, \boldsymbol{C})$ is detectable. The matrix $\boldsymbol{P}$ that we seek corresponds to the asymptotically stable eigenvalues of $\boldsymbol{H}$. The eigenvalues of the Hamiltonian matrix come in pairs $\pm s_i$ (see Exercise 9.15). The characteristic polynomial of $\boldsymbol{H}$ contains only even powers of s. With such constructed matrix $\boldsymbol{P}$, we have the following result:

Theorem 9.3 *The poles of the closed-loop system*

$$\dot{\boldsymbol{x}}(t) = \left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\right) \boldsymbol{x}(t)$$

are those eigenvalues of $\boldsymbol{H}$ having negative real parts.

Proof Because $\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{W} \\ \boldsymbol{Z} \end{bmatrix}$, we can write

$$\begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \\ -\boldsymbol{Q} & -\boldsymbol{A}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{W} \\ \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W} \\ \boldsymbol{Z} \end{bmatrix} \boldsymbol{\Lambda}.$$

Performing appropriate multiplication of $n \times n$ block matrices yields

$$\boldsymbol{A}\boldsymbol{W} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{Z} = \boldsymbol{W}\boldsymbol{\Lambda},$$

or

$$\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{Z}\boldsymbol{W}^{-1} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^{-1},$$

because $\boldsymbol{P} = \boldsymbol{Z}\boldsymbol{W}^{-1}$. Thus, the matrix $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}$ is similar to the matrix $\boldsymbol{\Lambda}$ whose eigenvalues are the asymptotically stable eigenvalues of $\boldsymbol{H}$. The proof is complete.

□

Example 9.11 Consider the following model of a dynamical system

$$\dot{x} = 2x + u,$$

and the associated performance index

$$J = \int_0^\infty (x^2 + ru^2)\, dt.$$

Find the value of r such that the optimal closed-loop system has its pole at -3.

We form the associated Hamiltonian matrix

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \\ -\boldsymbol{Q} & -\boldsymbol{A}^\top \end{bmatrix} = \begin{bmatrix} 2 & -\frac{1}{r} \\ -1 & -2 \end{bmatrix}.$$

The characteristic equation of $\boldsymbol{H}$ is

$$\det(s\boldsymbol{I}_2 - \boldsymbol{H}) = s^2 - 4 - \frac{1}{r} = 0.$$

Hence,

$$r = \frac{1}{5}$$

results in the optimal closed-loop system having its pole located at -3.

We mention again that the eigenvector method for solving the ARE can be generalized to the case when the eigenvalues of the Hamiltonian matrix $\boldsymbol{H}$ are not necessarily distinct or real.

9.3.3 Optimal Systems With Prescribed Poles

We can always find a state-feedback $u = -\boldsymbol{K}\boldsymbol{x}$ such that the closed-loop system $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}$ will have its poles in prescribed locations (symmetric with respect to the real axis) if and only if the pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable. In the multi-input systems the gain matrix $\boldsymbol{K}$ that shifts the poles to the prescribed locations is not unique. We would like to use the remaining degrees of freedom to construct a gain matrix $\boldsymbol{K}$ that shifts the system poles to the desired locations and at the same time minimizes a performance index. Specifically, we are given a system model,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t),$$

the desired set of closed-loop poles

$$\{s_1, s_2, \ldots, s_n\},$$

and a $m \times m$ positive definite matrix $\boldsymbol{R} = \boldsymbol{R}^\top$. We wish to construct a state-feedback control law $\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t)$ and an $n \times n$ positive semi-definite matrix $\boldsymbol{Q} = \boldsymbol{Q}^\top$ such that the closed-loop system has its poles in the desired locations, and at the same time the performance index

$$J = \int_0^\infty \left(\boldsymbol{x}(t)^\top \boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}(t)^\top \boldsymbol{R}\boldsymbol{u}(t)\right) dt$$

is minimized. In other words, we are interested in designing an optimal control system with prescribed poles. We illustrate the problem with a very simple example.

Example 9.12 For a dynamical system modeled by

$$\begin{aligned} \dot{x} &= 2x + u \\ &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \end{aligned}$$

and the performance index

$$\begin{aligned} J &= \int_0^\infty (qx^2 + u^2) dt \\ &= \int_0^\infty \left(\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}\right) dt, \end{aligned}$$

find q so that the eigenvalue of

$$\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}$$

is equal to -3.

The corresponding Hamiltonian matrix is

$$\boldsymbol{H} = \begin{bmatrix} 2 & -1 \\ -q & -2 \end{bmatrix}.$$

The characteristic polynomial of $\boldsymbol{H}$ is

$$\det[s\boldsymbol{I}_2 - \boldsymbol{H}] = \det \begin{bmatrix} s-2 & 1 \\ q & s+2 \end{bmatrix} = s^2 - 4 - q.$$

We want

$$\det[s\boldsymbol{I}_2 - \boldsymbol{H}] = (s+3)(s-3) = s^2 - 9.$$

Hence, the desired value of q is $q = 5$.

We now present a method for designing optimal control systems with prescribed poles. In constructing the gain matrix $\boldsymbol{K}$ and the weight matrix $\boldsymbol{Q}$, we use repeatedly a state variable transformation. To proceed, we analyze the system model, the performance index, and the Hamiltonian matrix in a new basis. Let

$$\boldsymbol{x} = \boldsymbol{T}\tilde{\boldsymbol{x}},$$

where $\det \boldsymbol{T} \neq 0$. Then, the system model in the new coordinates has the form

$$\begin{aligned}\dot{\tilde{\boldsymbol{x}}}(t) &= \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\tilde{\boldsymbol{x}}(t) + \boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{u}(t)\\ &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{B}}\boldsymbol{u}(t).\end{aligned}$$

The performance index J, after state variable transformation, takes the form

$$\tilde{J} = \int_0^\infty \left(\tilde{\boldsymbol{x}}(t)^\top \boldsymbol{T}^\top \boldsymbol{Q}\boldsymbol{T}\tilde{\boldsymbol{x}}(t) + \boldsymbol{u}(t)^\top \boldsymbol{R}\boldsymbol{u}(t)\right) dt.$$

The Hamiltonian matrix is transformed to

$$\begin{aligned}\tilde{\boldsymbol{H}} &= \begin{bmatrix} \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} & -\boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{T}^{-\top} \\ -\boldsymbol{T}^T\boldsymbol{Q}\boldsymbol{T} & -\boldsymbol{T}^\top\boldsymbol{A}^\top\boldsymbol{T}^{-\top} \end{bmatrix}\\ &= \begin{bmatrix} \tilde{\boldsymbol{A}} & -\tilde{\boldsymbol{B}}\boldsymbol{R}^{-1}\tilde{\boldsymbol{B}}^\top \\ -\tilde{\boldsymbol{Q}} & -\tilde{\boldsymbol{A}}^\top \end{bmatrix},\end{aligned}$$

where $\boldsymbol{T}^{-\top} = \left(\boldsymbol{T}^{-1}\right)^\top = \left(\boldsymbol{T}^\top\right)^{-1}$. Observe that

$$\tilde{\boldsymbol{H}} = \begin{bmatrix} \boldsymbol{T}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}^\top \end{bmatrix} \boldsymbol{H} \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}^{-\top} \end{bmatrix},$$

that is, $\tilde{\boldsymbol{H}}$ and $\boldsymbol{H}$ are similarity matrices because

$$\begin{bmatrix} \boldsymbol{T}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}^{-\top} \end{bmatrix} = \boldsymbol{I}_{2n}.$$

Hence, the eigenvalues of $\boldsymbol{H}$ and $\tilde{\boldsymbol{H}}$ are identical. Our next step is to compute the characteristic equation of $\tilde{\boldsymbol{H}}$ as a function of $\tilde{\boldsymbol{Q}}$. Computing $\det\left(s\boldsymbol{I}_{2n} - \tilde{\boldsymbol{H}}\right)$ is facilitated after we perform some manipulations on the characteristic matrix $\left[s\boldsymbol{I}_{2n} - \tilde{\boldsymbol{H}}\right]$ that do not change its determinant. Specifically, we premultiply the characteristic matrix $\left[s\boldsymbol{I}_{2n} - \tilde{\boldsymbol{H}}\right]$ by a $2n \times 2n$

matrix whose determinant equals 1. Then, the determinant of the product is the same as $\det\left(s\boldsymbol{I}_{2n}-\tilde{\boldsymbol{H}}\right)$, but it is much easier to evaluate. We have

$$
\begin{aligned}
&\begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} \\ -\tilde{\boldsymbol{Q}}\left(s\boldsymbol{I}_n-\tilde{\boldsymbol{A}}\right)^{-1} & \boldsymbol{I}_n \end{bmatrix}
\begin{bmatrix} s\boldsymbol{I}_n-\tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}}\boldsymbol{R}^{-1}\tilde{\boldsymbol{B}}^\top \\ \tilde{\boldsymbol{Q}} & s\boldsymbol{I}_n+\tilde{\boldsymbol{A}}^\top \end{bmatrix} \\
&\quad = \begin{bmatrix} s\boldsymbol{I}_n-\tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}}\boldsymbol{R}^{-1}\tilde{\boldsymbol{B}}^\top \\ \boldsymbol{0} & s\boldsymbol{I}_n+\tilde{\boldsymbol{A}}^\top-\tilde{\boldsymbol{Q}}\left(s\boldsymbol{I}_n-\tilde{\boldsymbol{A}}\right)^{-1}\tilde{\boldsymbol{B}}\boldsymbol{R}^{-1}\tilde{\boldsymbol{B}}^\top \end{bmatrix}.
\end{aligned}
$$

Let

$$
\boldsymbol{F}=\tilde{\boldsymbol{B}}\boldsymbol{R}^{-1}\tilde{\boldsymbol{B}}^\top,
$$

then

$$
\begin{aligned}
\det\left(s\boldsymbol{I}_{2n}-\tilde{\boldsymbol{H}}\right) &= \det\begin{bmatrix} s\boldsymbol{I}_n-\tilde{\boldsymbol{A}} & \boldsymbol{F} \\ \boldsymbol{0} & s\boldsymbol{I}_n+\tilde{\boldsymbol{A}}^\top-\tilde{\boldsymbol{Q}}\left(s\boldsymbol{I}_n-\tilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{F} \end{bmatrix} \\
&= \det\left(s\boldsymbol{I}_n-\tilde{\boldsymbol{A}}\right)\det\left(s\boldsymbol{I}_n+\tilde{\boldsymbol{A}}^\top-\tilde{\boldsymbol{Q}}\left(s\boldsymbol{I}_n-\tilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{F}\right).
\end{aligned}
$$

Suppose that the matrix $\boldsymbol{A}$ has n linearly independent eigenvectors, hence it is diagonalizable. Let

$$
\boldsymbol{T}=\begin{bmatrix} \boldsymbol{t}_1 & \boldsymbol{t}_2 & \cdots & \boldsymbol{t}_n \end{bmatrix}
$$

be a matrix composed of the eigenvectors of $\boldsymbol{A}$. Therefore,

$$
\tilde{\boldsymbol{A}}=\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}=\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.
$$

Note that $\tilde{\boldsymbol{A}}^\top=\tilde{\boldsymbol{A}}$. Assume that

$$
\tilde{\boldsymbol{Q}}=\tilde{\boldsymbol{Q}}_j=\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{q}_{jj} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix},
$$

that is, $\tilde{\boldsymbol{Q}}_j$ has only one non-zero element $\tilde{q}_{jj}$. Under the above assumptions the matrix $\left[s\boldsymbol{I}_n + \tilde{\boldsymbol{A}}^\top - \tilde{\boldsymbol{Q}}_j\left(s\boldsymbol{I}_n - \tilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{F}\right]$ takes the form

$$
\begin{aligned}
s\boldsymbol{I}_n + \tilde{\boldsymbol{A}}^\top - \tilde{\boldsymbol{Q}}_j\left(s\boldsymbol{I}_n - \tilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{F} &= \begin{bmatrix} s+\lambda_1 & 0 & \cdots & 0 \\ 0 & s+\lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s+\lambda_n \end{bmatrix} \\
&- \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \tilde{q}_{jj} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{s-\lambda_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{s-\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{s-\lambda_n} \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix} \\
&= \begin{bmatrix} s+\lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & s+\lambda_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & s+\lambda_j & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & s+\lambda_n \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\tilde{q}_{jj}f_{j1}}{s-\lambda_j} & \frac{\tilde{q}_{jj}f_{j2}}{s-\lambda_j} & \frac{\tilde{q}_{jj}f_{jj}}{s-\lambda_j} & \cdots & \frac{\tilde{q}_{jj}f_{jn}}{s-\lambda_j} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \\
&= \begin{bmatrix} s+\lambda_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & s+\lambda_2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ -\frac{\tilde{q}_{jj}f_{j1}}{s-\lambda_j} & -\frac{\tilde{q}_{jj}f_{j2}}{s-\lambda_j} & \cdots & \cdots & s+\lambda_j - \frac{\tilde{q}_{jj}f_{jj}}{s-\lambda_j} & -\frac{\tilde{q}_{jj}f_{jn}}{s-\lambda_j} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & s+\lambda_n \end{bmatrix}.
\end{aligned}
$$

Hence,

$$
\det\left(s\boldsymbol{I}_n + \tilde{\boldsymbol{A}}^\top - \tilde{\boldsymbol{Q}}_j\left(s\boldsymbol{I}_n - \tilde{\boldsymbol{A}}\right)^{-1}\boldsymbol{F}\right) = \left(\prod_{\substack{i=1\\ i\neq j}}^{n}(s+\lambda_i)\right)\left(s+\lambda_j - \frac{\tilde{q}_{jj}f_{jj}}{s-\lambda_j}\right).
$$

We can now evaluate the determinant of the Hamiltonian matrix,

$$\begin{aligned}\det(s\boldsymbol{I}_{2n}-\boldsymbol{H}) &= \det(s\boldsymbol{I}_{2n}-\tilde{\boldsymbol{H}})\\ &= \left(\prod_{i=1}^{n}(s-\lambda_i)\right)\left(\prod_{\substack{i=1\\ i\neq j}}^{n}(s+\lambda_i)\right)\left(s+\lambda_j-\frac{\tilde{q}_{jj}f_{jj}}{s-\lambda_j}\right)\\ &= \prod_{\substack{i=1\\ i\neq j}}^{n}(s+\lambda_i)(s-\lambda_i)\left((s+\lambda_j)(s-\lambda_j)-\tilde{q}_{jj}f_{jj}\right).\end{aligned}$$

From the form of the determinant of the Hamiltonian matrix, we conclude that selecting an appropriate value of $\tilde{q}_{jj}$ allows us to shift the j-th pole of the closed-loop system into a prespecified location while leaving the other poles in their original positions. To see this we represent the term that involves $\tilde{q}_{jj}$ as

$$\begin{aligned}(s+\lambda_j)(s-\lambda_j)-\tilde{q}_{jj}f_{jj} &= s^2-\lambda_j^2-\tilde{q}_{jj}f_{jj}\\ &= s^2-\left(\lambda_j^2+\tilde{q}_{jj}f_{jj}\right)\\ &= (s-s_j)(s+s_j),\end{aligned}$$

where

$$s_j=\pm\sqrt{\lambda_j^2+\tilde{q}_{jj}f_{jj}}.$$

From the above expression for s_j, we calculate $\tilde{q}_{jj}$ that results in the j-th pole being shifted to $-\sqrt{\lambda_j^2+\tilde{q}_{jj}f_{jj}}$. We obtain

$$\boxed{\tilde{q}_{jj}=\frac{s_j^2-\lambda_j^2}{f_{jj}}} \tag{9.33}$$

Note that while selecting the desired location for s_j, we have to make sure that $\tilde{q}_{jj}>0$ to ensure positive semidefiniteness of the weight matrix $\tilde{\boldsymbol{Q}}_j$. Having $\tilde{\boldsymbol{Q}}_j$, we compute $\boldsymbol{Q}_j=\boldsymbol{T}^{-\top}\tilde{\boldsymbol{Q}}_j\boldsymbol{T}^{-1}$ and solve the Riccati equation

$$\boldsymbol{A}^\top\boldsymbol{P}_j+\boldsymbol{P}_j\boldsymbol{A}+\boldsymbol{Q}_j-\boldsymbol{P}_j\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}_j=\boldsymbol{O} \tag{9.34}$$

for $\boldsymbol{P}_j$. We compute $\boldsymbol{u}=-\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}_j\boldsymbol{x}+\boldsymbol{u}_j=-\boldsymbol{K}_j\boldsymbol{x}+\boldsymbol{u}_j$ that shifts only one pole of the closed-loop system

$$\begin{aligned}\dot{\boldsymbol{x}} &= (\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_j)\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}_j\\ &= \boldsymbol{A}_j\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}_j\end{aligned}$$

to its specified position. We then start with the new system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_j \boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}_j$$

and shift the next pole to a specified location, and then combine the obtained feedback gains. The above method is based on the result, established by Solheim [307], that different feedback matrices $\boldsymbol{K}_j$ and the corresponding weight matrices $\boldsymbol{Q}_j$ can be combined to give the final optimal gain $\boldsymbol{K}$ and the corresponding weight matrix $\boldsymbol{Q}$. Indeed, suppose we apply the above method to the system $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_j)\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}_j$. After finding an appropriate weight matrix $\boldsymbol{Q}_{j+1}$, we solve the Riccati equation

$$(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_j)^\top \boldsymbol{P}_{j+1} + \boldsymbol{P}_{j+1}(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_j) + \boldsymbol{Q}_{j+1} - \boldsymbol{P}_{j+1}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}_{j+1} = \boldsymbol{O} \tag{9.35}$$

for $\boldsymbol{P}_{j+1}$ to compute the gain $\boldsymbol{K}_{j+1}$. Adding the two Riccati equations (9.34) and (9.35), and performing simple manipulations yields

$$\boldsymbol{A}^\top(\boldsymbol{P}_j + \boldsymbol{P}_{j+1}) + (\boldsymbol{P}_j + \boldsymbol{P}_{j+1})\boldsymbol{A} + \boldsymbol{Q}_j + \boldsymbol{Q}_{j+1} - (\boldsymbol{P}_j + \boldsymbol{P}_{j+1})\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top(\boldsymbol{P}_j + \boldsymbol{P}_{j+1}) = \boldsymbol{O}.$$

Let $\boldsymbol{P} = \boldsymbol{P}_j + \boldsymbol{P}_{j+1}$ and $\boldsymbol{Q} = \boldsymbol{Q}_j + \boldsymbol{Q}_{j+1}$, then

$$\begin{aligned} \boldsymbol{K} &= \boldsymbol{K}_j + \boldsymbol{K}_{j+1} \\ &= \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}_j + \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}_{j+1}. \end{aligned}$$

In a similar manner we can show that the method works for more than just two iterations.

Example 9.13 We consider a model of one-link robot manipulator presented in Subsection 2.4.5. Applying Kirchhoff's voltage law to the armature circuit yields

$$L_a \frac{di_a}{dt} + R_a i_a + K_b N \frac{d\theta_p}{dt} = u,$$

where K_b is the back emf constant. We assume now that $L_a \approx 0$. Hence,

$$u = R_a i_a + K_b N \frac{d\theta_p}{dt}, \tag{9.36}$$

We next compute i_a from (9.36) and substitute the result into (2.81) to get

$$ml^2 \frac{d^2\theta_p}{dt^2} = mgl \sin\theta_p + NK_m \left(\frac{u}{R_a} - \frac{K_b N \frac{d\theta_p}{dt}}{R_a} \right). \tag{9.37}$$

We construct a reduced-order state-space model of the one-link robot manipulator. For this we choose the following state and output variables:

$$x_1 = \theta_p, \quad x_2 = \frac{d\theta_p}{dt} = \omega_p, \quad \text{and} \quad y = x_1.$$

Then, using (9.37), we obtain the following simple state-space model of the robot manipulator:

$$\begin{aligned}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} &= \begin{bmatrix} x_2 \\ \frac{g}{l}\sin x_1 - \frac{K_b K_m N^2}{ml^2 R_a} x_2 + \frac{N K_m}{ml^2 R_a} u\end{bmatrix}\\ y &= x_1.\end{aligned}$$

Reasonable parameters for the robot are: $l = 1\,\mathrm{m}$, $m = 1\,\mathrm{kg}$, $N = 10$, $K_m = 0.1\,\mathrm{Nm/A}$, $K_b = 0.1\,\mathrm{Vs/rad}$, $R_a = 1\,\Omega$. With the above parameters the robot model takes the form:

$$\begin{aligned}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} &= \begin{bmatrix} x_2 \\ 9.8\,\sin x_1 - x_2 + u\end{bmatrix}\\ y &= x_1.\end{aligned}$$

Time histories of state trajectories of the uncontrolled nonlinear system, $u = 0$, are shown in Figure 9.11 for the initial conditions are $x_1(0) = 1$ and $x_2(0) = 0$. A phase portrait

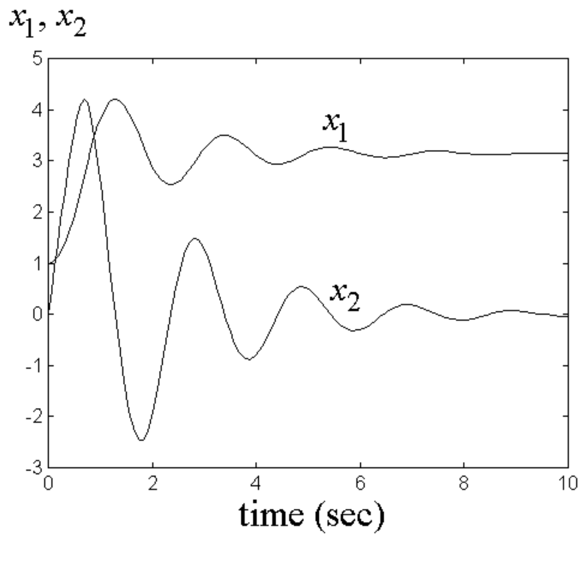


Figure 9.11: Plots of $y = x_1$ and x_2 versus time of the uncontrolled nonlinear system of Example 9.13.

of the uncontrolled nonlinear system is shown in Figure 9.12. The linearized model about

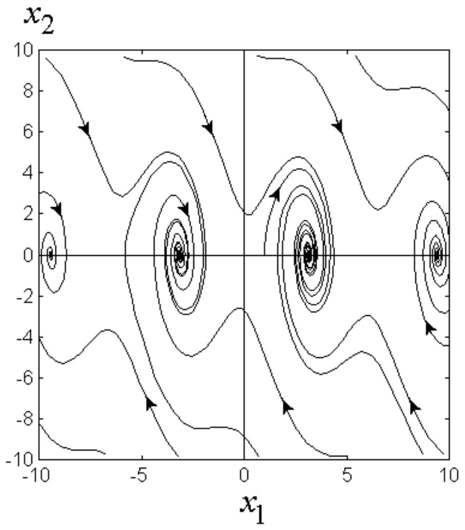


Figure 9.12: A phase portrait of the uncontrolled nonlinear system of Example 9.13.

$\boldsymbol{x} = \mathbf{0}, u = 0$ has the form

$$\left.\begin{array}{rcl} \frac{d}{dt}\Delta\boldsymbol{x} & = & \begin{bmatrix} 0 & 1 \\ 9.8 & -1 \end{bmatrix} \Delta\boldsymbol{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Delta u \\ \Delta y & = & \begin{bmatrix} 1 & 0 \end{bmatrix} \Delta\boldsymbol{x} \end{array}\right\} \tag{9.38}$$

Plots of $y = x_1$ and x_2 versus time are shown in Figure 9.13. A phase portrait of the linearized system is shown in Figure 9.14. Let

$$J = \int_0^\infty \left(y^2 + u^2\right) dt.$$

We find the linear state-feedback control law $u = -\boldsymbol{k}\boldsymbol{x}$ that minimizes J subject to the equations given by (9.38). We have

$$\boldsymbol{Q} = \boldsymbol{c}^\top \boldsymbol{c} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{R} = [1].$$

Solving the associated Riccati equation yields

$$\boldsymbol{P} = \begin{bmatrix} 72.3371 & 19.6509 \\ 19.6509 & 5.3484 \end{bmatrix}. \tag{9.39}$$

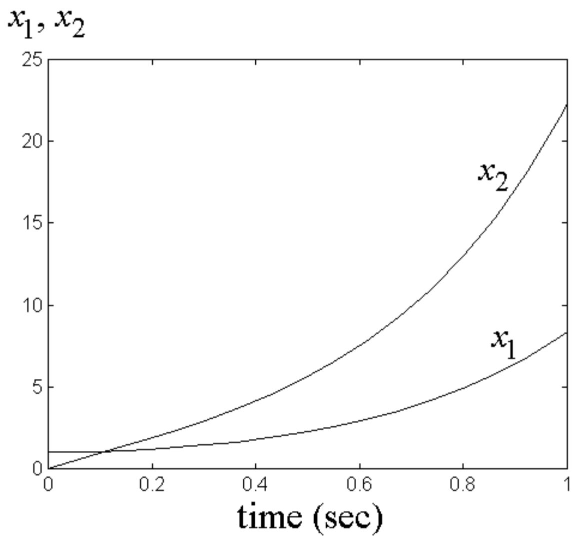


Figure 9.13: Plots of x_1 and x_2 versus time of the linearized system of Example 9.13.

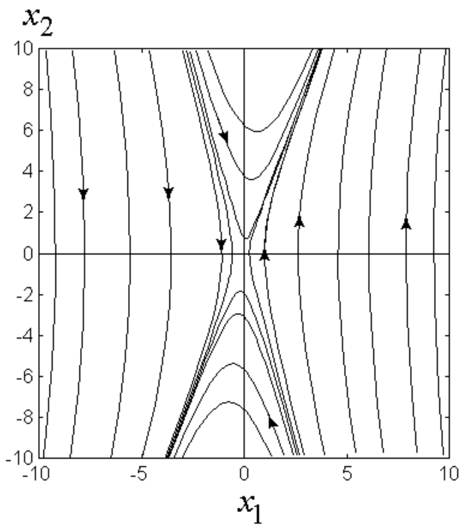


Figure 9.14: A phase portrait of the linearized system of Example 9.13.

Hence,

$$\boldsymbol{k} = \begin{bmatrix} 19.6509 & 5.3484 \end{bmatrix}, \tag{9.40}$$

and

$$\boldsymbol{A}_c = \begin{bmatrix} 0 & 1 \\ -9.8509 & -6.3484 \end{bmatrix}.$$

Plots of x_1 and x_2 versus time of the closed-loop linear system

$$\begin{aligned} \dot{\boldsymbol{x}} &= (\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k})\,\boldsymbol{x} = \boldsymbol{A}_c\boldsymbol{x} \\ y &= x_1, \end{aligned}$$

when the initial conditions are $x_1(0) = 1$ and $x_2(0) = 0$ are shown in Figure 9.15. A phase

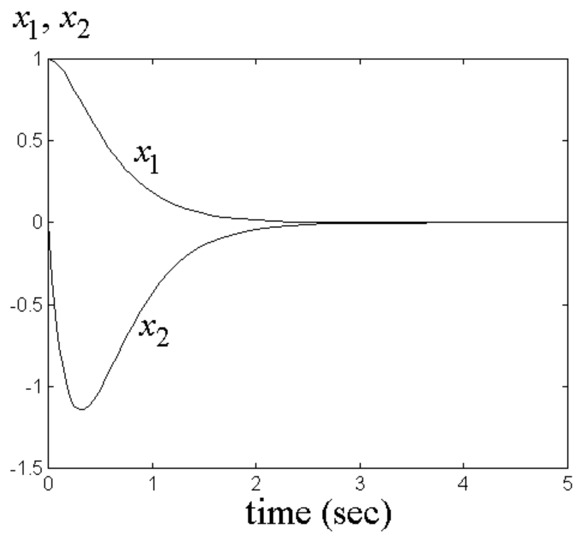


Figure 9.15: Plots of x_1 and x_2 versus time of the linear closed-loop system of Example 9.13.

portrait of the closed-loop linear system is shown in Figure 9.16. We now apply the above optimal controller to the nonlinear model. Plots of x_1 and x_2 versus time of the closed-loop nonlinear systems are shown in Figure 9.17. A phase portrait of the nonlinear closed-loop system is shown in Figure 9.18. The closed-loop poles of the linearized system, that is, the eigenvalues of $\boldsymbol{A}_c$ are $\lambda_1 = -2.7003$ and $\lambda_2 = -3.6481$.

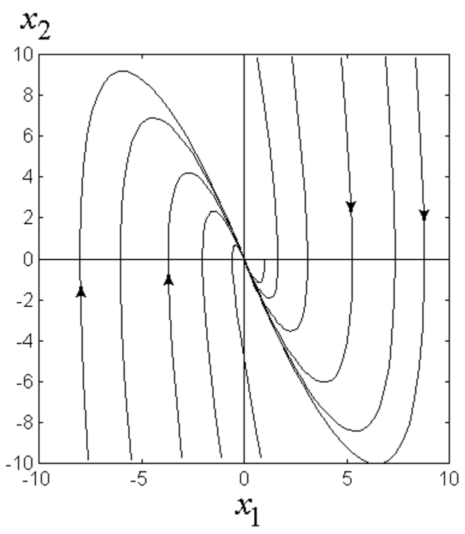


Figure 9.16: A phase portrait of the linear closed-loop system of Example 9.13.

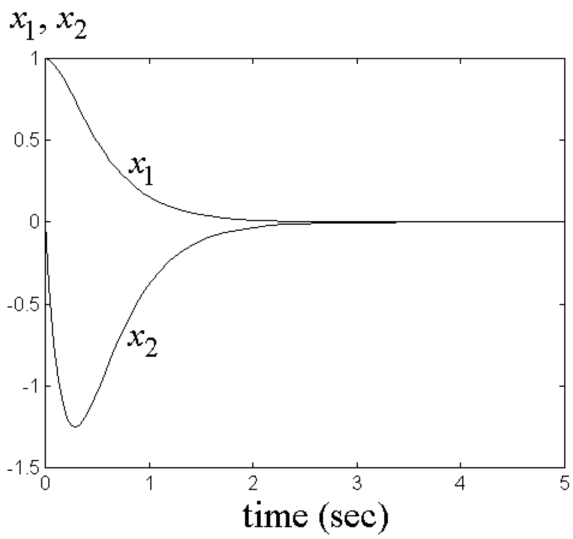


Figure 9.17: Plots of x_1 and x_2 of the nonlinear closed-loop system of Example 9.13.

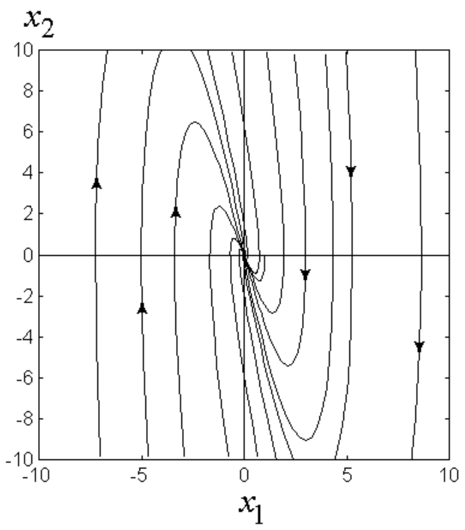


Figure 9.18: A phase portrait of the nonlinear closed-loop system of Example 9.13.

Example 9.14 We use the data from the previous example to find the weight matrix $\boldsymbol{Q}_2$ and the corresponding $\boldsymbol{k}_2$ so that λ_2 is shifted to $s_2 = -5$, that is, the new poles of the closed-loop system

$$\dot{\boldsymbol{x}} = (\boldsymbol{A}_c - \boldsymbol{b}\boldsymbol{k}_2)\,\boldsymbol{x}$$

are $s_1 = \lambda_1$ and $s_2 = -5$. For the control law $u = -\boldsymbol{k}\boldsymbol{x} + r$, the linear closed-loop system is

$$\begin{aligned}\dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 \\ -9.8509 & -6.3484 \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ &= \boldsymbol{A}_c\boldsymbol{x} + \boldsymbol{b}r,\end{aligned}$$

where for simplicity we used $\boldsymbol{x}$ rather than $\Delta\boldsymbol{x}$, and u instead of Δu. The eigenvalues of $\boldsymbol{A}_c$ are $\lambda_1 = -2.7003$ and $\lambda_2 = -3.6481$. We have $\lambda_1 \neq \lambda_2$ and thus the matrix $\boldsymbol{A}_c$ is diagonalizable. Note that different eigenvalues are only a sufficient condition for diagonalizability. We construct the similarity transformation using the eigenvectors of $\boldsymbol{A}_c$ The eigenvectors of $\boldsymbol{A}_c$ are

$$\boldsymbol{v}_1 = \begin{bmatrix} 0.3473 \\ -0.9378 \end{bmatrix} \quad \text{and} \quad \boldsymbol{v}_2 = \begin{bmatrix} -0.2644 \\ -0.9644 \end{bmatrix}.$$

Let $\boldsymbol{x} = \boldsymbol{T}\boldsymbol{z}$, where $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 \end{bmatrix}$. Then, $\boldsymbol{A}_c$ in the new coordinates has the form

$$\tilde{\boldsymbol{A}}_c = \boldsymbol{T}^{-1}\boldsymbol{A}_c\boldsymbol{T} = \begin{bmatrix} -2.7003 & 0 \\ 0 & -3.6481 \end{bmatrix},$$

and

$$\tilde{\boldsymbol{b}} = \boldsymbol{T}^{-1}\boldsymbol{b} = \begin{bmatrix} 3.0383 \\ 3.9912 \end{bmatrix}.$$

Next, we compute

$$\boldsymbol{T}^{-1}\boldsymbol{b}\boldsymbol{R}^{-1}\boldsymbol{b}^\top\boldsymbol{T}^{-\top} = \begin{bmatrix} 9.2312 & 12.1264 \\ 12.1264 & 15.9296 \end{bmatrix}.$$

Hence,

$$\tilde{q}_{22} = \frac{s_2^2 - \lambda_2^2}{15.9296} = 0.7339.$$

Therefore,

$$\tilde{\boldsymbol{Q}}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0.7339 \end{bmatrix} = \boldsymbol{T}^\top\boldsymbol{Q}_2\boldsymbol{T},$$

and

$$\boldsymbol{Q}_2 = \boldsymbol{T}^{-\top}\tilde{\boldsymbol{Q}}_2\boldsymbol{T}^{-1} = \begin{bmatrix} 85.2450 & 31.5686 \\ 31.5686 & 11.6907 \end{bmatrix}.$$

Solving the ARE with $\boldsymbol{A}_c$, $\boldsymbol{b}$, $\boldsymbol{Q}_2$, and $\boldsymbol{R}$ yields

$$\boldsymbol{P}_2 = \begin{bmatrix} 9.8572 & 3.6504 \\ 3.6504 & 1.3518 \end{bmatrix},$$

and

$$\boldsymbol{k}_2 = \begin{bmatrix} 3.6504 & 1.3518 \end{bmatrix}.$$

As a check, if we take

$$\boldsymbol{Q}_{new} = \boldsymbol{Q} + \boldsymbol{Q}_2 = \begin{bmatrix} 86.2450 & 31.5686 \\ 31.5686 & 11.6907 \end{bmatrix},$$

where $\boldsymbol{Q} = \boldsymbol{c}^\top\boldsymbol{c}$ as in Example 9.13, and $\boldsymbol{R}_{new} = \boldsymbol{R}$, then solving the associated ARE using $\boldsymbol{A}$, rather then $\boldsymbol{A}_c$, we obtain

$$\boldsymbol{P}_{new} = \begin{bmatrix} 82.1943 & 23.3013 \\ 23.3013 & 6.7002 \end{bmatrix} = \boldsymbol{P} + \boldsymbol{P}_2,$$

where $\boldsymbol{P}$ is given by (9.39), and

$$\boldsymbol{k}_{new} = \begin{bmatrix} 23.3013 & 6.7002 \end{bmatrix} = \boldsymbol{k} + \boldsymbol{k}_2,$$

where $\boldsymbol{k}$ is the optimal gain given by (9.40).

Example 9.15 For the linearized model given by (9.38) and $\boldsymbol{R} = [2]$, we determine $\boldsymbol{Q}$ and the corresponding $\boldsymbol{k}$ that result in the closed-loop system $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k})\,\boldsymbol{x}$ with poles located at $s_1 = -4$ and $s_2 = -5$, where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 9.8 & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

We shift poles into the desired locations one by one. We first diagonalize $\boldsymbol{A}$ using similarity transformation constructed from eigenvectors of $\boldsymbol{A}$. The eigenvalues of $\boldsymbol{A}$ are $\lambda_1 = 2.6702$ and $\lambda_2 = -3.6702$. We denote the corresponding eigenvectors by $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, respectively. We obtain

$$\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} = \begin{bmatrix} 2.6702 & 0 \\ 0 & -3.6702 \end{bmatrix},$$

where

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} 0.3507 & -0.2629 \\ 0.9365 & 0.9648 \end{bmatrix}.$$

We construct $\boldsymbol{k}_1$ so that λ_1 is shifted to $s_1 = -4$, while λ_2 remains unchanged, that is, the eigenvalues of $\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k}_1$ become s_1 and λ_2. For this, we first compute

$$\boldsymbol{F} = \boldsymbol{T}^{-1}\boldsymbol{b}\boldsymbol{R}^{-1}\boldsymbol{b}^\top\boldsymbol{T}^{-\top} = \begin{bmatrix} 0.1011 & 0.1349 \\ 0.1349 & 0.1800 \end{bmatrix}.$$

Hence, $f_{11} = 0.1011$,

$$\tilde{q}_{11} = \frac{4^2 - (2.6702)^2}{0.1011} = 87.7352,$$

and

$$\tilde{\boldsymbol{Q}}_1 = \begin{bmatrix} 87.7352 & 0 \\ 0 & 0 \end{bmatrix} = \boldsymbol{T}^\top\boldsymbol{Q}_1\boldsymbol{T}.$$

Hence,

$$\boldsymbol{Q}_1 = \boldsymbol{T}^{-\top}\tilde{\boldsymbol{Q}}_1\boldsymbol{T}^{-1} = \begin{bmatrix} 239.0024 & 65.1202 \\ 65.1202 & 17.7431 \end{bmatrix}.$$

Solving the associated ARE, using $\boldsymbol{R} = [2]$, yields

$$\boldsymbol{P}_1 = \begin{bmatrix} 179.7014 & 48.9626 \\ 48.9626 & 13.3407 \end{bmatrix}.$$

Therefore,

$$\boldsymbol{k}_1 = \begin{bmatrix} 24.4813 & 6.6703 \end{bmatrix},$$

and

$$\boldsymbol{A}_1 = \boldsymbol{A} - \boldsymbol{b}\boldsymbol{k}_1 = \begin{bmatrix} 0 & 1 \\ -14.6813 & -7.6703 \end{bmatrix}.$$

We construct $\boldsymbol{k}_2$ that shifts λ_2 to s_2 leaving s_1 unchanged. We first diagonalize $\boldsymbol{A}_1$ using its eigenvectors. Note that $\boldsymbol{A}_1$ and $\boldsymbol{A}$ share the common eigenvector $\boldsymbol{v}_2$. The other eigenvector of $\boldsymbol{A}_1$ corresponding to its eigenvalue s_1 is $\boldsymbol{u}_1 = \begin{bmatrix} -0.2425 & 0.9701 \end{bmatrix}^\top$. We have

$$\boldsymbol{T}^{-1}\boldsymbol{A}_1\boldsymbol{T} = \begin{bmatrix} s_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

where

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} -0.2425 & 0.2629 \\ 0.9701 & -0.9648 \end{bmatrix}.$$

We next compute

$$\boldsymbol{T}^{-1}\boldsymbol{b}\boldsymbol{R}^{-1}\boldsymbol{b}^\top\boldsymbol{T}^{-\top} = \begin{bmatrix} 78.0607 & 72.0157 \\ 72.0157 & 66.4389 \end{bmatrix}.$$

Thus, $\tilde{f}_{22} = 66.4389$,

$$\tilde{q}_{22} = \frac{5^2 - (-3.6702)^2}{66.4389} = 0.1735,$$

and

$$\tilde{\boldsymbol{Q}}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0.1735 \end{bmatrix} = \boldsymbol{T}^\top\boldsymbol{Q}_2\boldsymbol{T}.$$

From the above, we obtain

$$\boldsymbol{Q}_2 = \boldsymbol{T}^{-\top}\tilde{\boldsymbol{Q}}_2\boldsymbol{T}^{-1} = \begin{bmatrix} 368.9003 & 92.2211 \\ 92.2211 & 23.0543 \end{bmatrix}.$$

Solving the associated ARE, we get

$$\boldsymbol{P}_2 = \begin{bmatrix} 42.5495 & 10.6369 \\ 10.6369 & 2.6591 \end{bmatrix},$$

and

$$\boldsymbol{k}_2 = \begin{bmatrix} 5.3185 & 1.3296 \end{bmatrix}.$$

The final weight matrix $\boldsymbol{Q}$ that results in shifting poles into the desired locations is

$$\boldsymbol{Q} = \boldsymbol{Q}_1 + \boldsymbol{Q}_2 = \begin{bmatrix} 607.9027 & 157.3413 \\ 157.3413 & 40.7974 \end{bmatrix}.$$

The corresponding $\boldsymbol{P}$ is

$$\boldsymbol{P} = \boldsymbol{P}_1 + \boldsymbol{P}_2 = \begin{bmatrix} 222.2509 & 59.5996 \\ 59.5996 & 15.9998 \end{bmatrix}.$$

The corresponding optimal gain vector $\boldsymbol{k}$ is

$$\boldsymbol{k} = \boldsymbol{k}_1 + \boldsymbol{k}_2 = \begin{bmatrix} 29.7998 & 7.9999 \end{bmatrix}.$$

9.3.4 Optimal Saturating Controllers

Asymptotic stability of an equilibrium state of a nonlinear system is a local property. This means that the system's trajectories starting in a neighborhood of the equilibrium state will approach this equilibrium asymptotically. Often, we are interested in determining how far from the equilibrium state the system's trajectories can start and still converge to the equilibrium asymptotically. The region around an equilibrium state within which all trajectories approach the equilibrium is called a *region of attraction*, or *region of asymptotic stability* abbreviated as RAS. Finding the exact RAS analytically can be a challenging task. A tool often used to estimate the exact RAS is a Lyapunov function. Specifically, if $V = V(\boldsymbol{x})$ is a Lyapunov function of the equilibrium state of interest, that is, V is positive definite with respect to the equilibrium state in a domain $\mathcal{S}$ and the time derivative of V evaluated on the trajectories of the system is negative definite in $\mathcal{S}$, then the set

$$\Omega_c = \left\{ \boldsymbol{x} : \boldsymbol{x} \in \mathcal{S}, V(\boldsymbol{x}) \leq c, \dot{V}(\boldsymbol{x}) < 0 \right\}$$

is an estimate of the exact RAS. In other words, a region of attraction is a set of states $\boldsymbol{x}$ for which $\dot{V}(\boldsymbol{x}) < 0$ and $V(\boldsymbol{x}) \leq c$ for some positive constant c. The set Ω_c may be a conservative estimate, often a crude underestimate, of the true RAS. In order to obtain the least conservative estimate of the domain of attraction, corresponding to the given Lyapunov

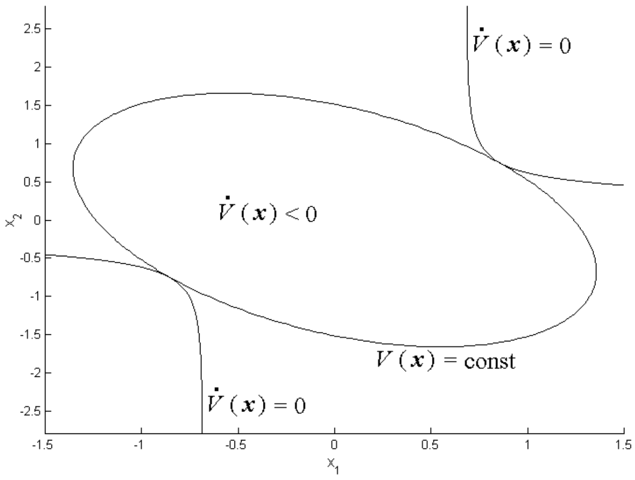


Figure 9.19: Estimating the region of asymptotic stability.

function, we could try to maximize the size of Ω_c by finding the largest V contour that can just fit on the equilibrium side of the surface described by $\dot{V}(\boldsymbol{x}) = 0$. This can be accomplished by solving the optimization problem for c such that

$$c = \min_{\dot{V}(\boldsymbol{x})=0} V(\boldsymbol{x}). \tag{9.41}$$

Solving the above optimization problem corresponds to maximizing the size of the region $\{\boldsymbol{x} : V(\boldsymbol{x}) \leq c\}$ in which $\dot{V}(\boldsymbol{x}) < 0$ by making the boundary of $\{\boldsymbol{x} : V(\boldsymbol{x}) \leq c\}$ to be tangent to the boundary of $\{\boldsymbol{x} : \dot{V}(\boldsymbol{x}) < 0\}$. This is illustrated in Figure 9.19.

We next estimate the RAS for single-input linear dynamical systems with saturating linear optimal control. Our development follows that of Friedland [108, Section 5.1]. We will use such controllers in Chapter 18 to stabilize chaotic systems. We consider a single-input linear system model,

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\, U_{max}\text{sat}\left(u/U_{max}\right), \tag{9.42}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and

$$\text{sat}(z) = \begin{cases} 1 & \text{if} \quad z \geq 1 \\ z & \text{if} \quad |z| < 1 \\ -1 & \text{if} \quad z \leq -1. \end{cases}$$

In the linear region, $|u| \le U_{max}$, (9.42) becomes $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u$. We consider using the control law of the form

$$u = U_{max}\mathrm{sat}\left(-\boldsymbol{k}\boldsymbol{x}/U_{max}\right). \tag{9.43}$$

When $|\boldsymbol{k}\boldsymbol{x}| \le U_{max}$, the control law given by (9.43) becomes a linear state-feedback controller, $u = -\boldsymbol{k}\boldsymbol{x}$. For simplicity of notation, we can assume that $U_{max} = 1$. Indeed, we can always scale the input vector $\boldsymbol{b}$ so that $U_{max} = 1$. The gain vector $\boldsymbol{k}$ is obtained by solving the associated LQR problem, with the performance index,

$$J = \int_0^\infty \left(\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + ru^2\right) dt,$$

where we assume that the pair $(\boldsymbol{A}, \boldsymbol{b})$ is reachable, $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$ and $r > 0$. Solving the algebraic Riccati equation,

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^\top \boldsymbol{P}/r = \boldsymbol{O}, \tag{9.44}$$

we use its solution, $\boldsymbol{P} = \boldsymbol{P}^\top > 0$, to construct the gain vector,

$$\boldsymbol{k} = \boldsymbol{b}^\top \boldsymbol{P}/r. \tag{9.45}$$

We then proceed with evaluating of $\frac{d}{dt}V = \frac{d}{dt}\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}$ on the trajectories of the closed-loop system,

$$\begin{aligned} \dot{V} &= 2\boldsymbol{x}^\top \boldsymbol{P}\dot{\boldsymbol{x}} \\ &= \boldsymbol{x}^\top(\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A})\boldsymbol{x} - 2\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{b}\,\mathrm{sat}(\boldsymbol{k}\boldsymbol{x}). \end{aligned} \tag{9.46}$$

By (9.45),

$$\boldsymbol{b}^\top \boldsymbol{P} = r\boldsymbol{k}. \tag{9.47}$$

Using the above and (9.44) gives

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = r\boldsymbol{k}^\top \boldsymbol{k} - \boldsymbol{Q}. \tag{9.48}$$

Substituting (9.48) and (9.47) into (9.46) yields

$$\boxed{\dot{V} = -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - 2r\boldsymbol{k}\boldsymbol{x}\,\mathrm{sat}(\boldsymbol{k}\boldsymbol{x}) + r(\boldsymbol{k}\boldsymbol{x})^2} \tag{9.49}$$

For $\boldsymbol{x}$ such that $|\boldsymbol{k}\boldsymbol{x}| \le 1$, the above becomes

$$\dot{V} = -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - r(\boldsymbol{k}\boldsymbol{x})^2,$$

which is negative definite. Thus, the origin is (locally) asymptotically stable for the closed-loop system.

Recall that an estimate of the RAS is a set of states bounded by the surface $V =$ constant for which $\dot{V} < 0$. To find as large an estimate as possible, we attempt to find the surface $\dot{V} = 0$ and find $c = \min V(\boldsymbol{x})$ on that surface. Then, the region bounded by $V(\boldsymbol{x}) = c$ is an estimate of the RAS. Solving the above problem amounts to solving the optimization problem,

$$\min_{\boldsymbol{x}} \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}$$

$$\text{subject to} \quad -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - 2r\boldsymbol{k}\boldsymbol{x}\,\mathrm{sat}(\boldsymbol{k}\boldsymbol{x}) + r(\boldsymbol{k}\boldsymbol{x})^2 = 0. \tag{9.50}$$

Solving the above problem is not a simple task. Instead, we solve an easier problem resulting in a more conservative estimate of the RAS. We first rewrite (9.50) as

$$y^2 - 2y\,\mathrm{sat}(y) - \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}/r = 0, \tag{9.51}$$

where $y = \boldsymbol{k}\boldsymbol{x}$. We investigate the case when $|y| \geq 1$ because for $|y| < 1$, we have $\dot{V} < 0$ and we are interested in the solution to the above when $\dot{V} = 0$. When $y \geq 1$, (9.51) becomes

$$y^2 - 2y - \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}/r = 0. \tag{9.52}$$

because for $y \geq 1$, $\mathrm{sat}(y) = 1$. The solution to (9.52) for $y \geq 1$ is

$$y = 1 + \sqrt{1 + \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}/r} > 2, \tag{9.53}$$

while for $y \leq -1$, the solution to (9.51) has the form

$$y = -1 - \sqrt{1 + \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}/r} < -2. \tag{9.54}$$

The above solutions satisfy the inequality,

$$|y| \geq 2.$$

Thus, we have $\dot{V} < 0$ for $|y| = |\boldsymbol{k}\boldsymbol{x}| \leq 2$ and hence the RAS includes the region

$$\{\boldsymbol{x} : V(\boldsymbol{x}) \leq \tilde{c}\},$$

where $\tilde{c}$ is obtained by solving the optimization problem

$$\boxed{\tilde{c} = \max_{|\boldsymbol{k}\boldsymbol{x}| \leq 2} \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}}$$

We use the Lagrange multipliers method to solve the above constrained optimization problem. We solve the equivalent problem,

$$\max \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}$$

$$\text{subject to} \quad (\boldsymbol{k}\boldsymbol{x})^2 = 4.$$

The Lagrangian for the above problem is

$$l(\boldsymbol{x}, \lambda) = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} + \lambda\left(4 - (\boldsymbol{k}\boldsymbol{x})^2\right).$$

Applying to the Lagrangian l the first-order necessary conditions for unconstrained minimization gives

$$D_{\boldsymbol{x}} l(\boldsymbol{x}, \lambda) = 2\boldsymbol{x}^\top \boldsymbol{P} - 2\lambda(\boldsymbol{k}\boldsymbol{x})\boldsymbol{k} = \boldsymbol{0}, \tag{9.55}$$

$$D_\lambda l(\boldsymbol{x}, \lambda) = 4 - (\boldsymbol{k}\boldsymbol{x})^2 = 0. \tag{9.56}$$

From (9.55),

$$\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} = \lambda(\boldsymbol{k}\boldsymbol{x})^2.$$

Because $(\boldsymbol{k}\boldsymbol{x})^2 = 4$,

$$\lambda = \frac{1}{4}\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x},$$

and therefore,

$$\tilde{c} = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} = 4\lambda.$$

Again, using (9.55), we obtain

$$\boldsymbol{x}^\top \boldsymbol{P} = \lambda(\boldsymbol{k}\boldsymbol{x})\boldsymbol{k}.$$

Postmultiplying both sides of the above by $\boldsymbol{P}^{-1}\boldsymbol{k}^\top$ gives

$$\boldsymbol{x}^\top \boldsymbol{k}^\top = \boldsymbol{k}\boldsymbol{x} = \lambda(\boldsymbol{k}\boldsymbol{x})\boldsymbol{k}\boldsymbol{P}^{-1}\boldsymbol{k}^\top. \tag{9.57}$$

Note that $\boldsymbol{k}\boldsymbol{x} \neq 0$ because $(\boldsymbol{k}\boldsymbol{x})^2 = 4$. Hence, from (9.57),

$$\lambda = \frac{1}{\boldsymbol{k}\boldsymbol{P}^{-1}\boldsymbol{k}^\top}.$$

Substituting (9.45) into the above gives

$$\lambda = \frac{1}{\boldsymbol{k}\boldsymbol{P}^{-1}\boldsymbol{k}^\top} = \frac{r^2}{\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{b}}.$$

Hence,

$$\boxed{\tilde{c} = 4\lambda = \frac{4r^2}{\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{b}}}$$

9.3.5 Linear Quadratic Regulator for Discrete Systems on an Infinite Time Interval

Consider a discrete linear plant model,

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k], \quad k = 0, 1, 2, \ldots, \tag{9.58}$$

with a specified initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, where $\boldsymbol{x}[k] \in \mathbb{R}^n$ and $\boldsymbol{u}[k] \in \mathbb{R}^m$. We assume that the pair $(\boldsymbol{A}, \boldsymbol{B})$ is reachable. Our objective is to construct a stabilizing linear state-feedback controller,

$$\boldsymbol{u}[k] = -\boldsymbol{K}\boldsymbol{x}[k]$$

that minimizes the quadratic performance index

$$J(\boldsymbol{u}) = \sum_{k=0}^{\infty} \left\{ \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}[k]^\top \boldsymbol{R}\boldsymbol{u}[k] \right\}, \tag{9.59}$$

where $\boldsymbol{Q} = \boldsymbol{Q}^\top \geq 0$, and $\boldsymbol{R} = \boldsymbol{R}^\top > 0$. We assume that the components of the control vector are unconstrained. An optimal control law will be denoted $\boldsymbol{u}^*$.

Because we seek the controller among the ones that stabilize the system (9.58), the closed-loop system,

$$\boldsymbol{x}[k+1] = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\,\boldsymbol{x}[k], \tag{9.60}$$

is asymptotically stable, that is, the eigenvalues of the matrix $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}$ are in the open unit disk. This implies that there exists a Lyapunov function, $V(\boldsymbol{x}[k]) = \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k]$, for the system (9.60). Therefore the first forward difference, $\Delta V(\boldsymbol{x}[k]) = V(\boldsymbol{x}[k+1]) - V(\boldsymbol{x}[k])$ evaluated on the trajectories of (9.60) is negative definite. We now state and prove a sufficient condition for $\boldsymbol{u}[k] = -\boldsymbol{K}\boldsymbol{x}[k]$ to be optimal.

Theorem 9.4 *If the state feedback controller, $\boldsymbol{u}[k] = -\boldsymbol{K}\boldsymbol{x}[k]$, is such that*

$$\min_{\boldsymbol{u}} \left(\Delta V(\boldsymbol{x}[k]) + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}[k]^\top \boldsymbol{R}\boldsymbol{u}[k] \right) = 0, \tag{9.61}$$

then it is optimal.

Proof We can represent (9.61) as

$$\Delta V(\boldsymbol{x}[k])|_{\boldsymbol{u}=\boldsymbol{u}^*} + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}^*[k]^\top \boldsymbol{R}\boldsymbol{u}^*[k] = 0.$$

Hence,

$$\Delta V(\boldsymbol{x}[k])|_{\boldsymbol{u}=\boldsymbol{u}^*} = (V(\boldsymbol{x}[k+1]) - V(\boldsymbol{x}[k])|_{\boldsymbol{u}=\boldsymbol{u}^*} = -\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] - \boldsymbol{u}^*[k]^\top \boldsymbol{R}\boldsymbol{u}^*[k].$$

We sum both sides from $k=0$ to $k=\infty$ to get

$$V(\boldsymbol{x}[\infty]) - V(\boldsymbol{x}[0]) = -\sum_{k=0}^{\infty}\left(\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}^*[k]^\top \boldsymbol{R}\boldsymbol{u}^*[k]\right).$$

Because by assumption the closed-loop system is asymptotically stable, we have $\boldsymbol{x}[\infty]=\boldsymbol{0}$. Hence,

$$V(\boldsymbol{x}[0]) = \boldsymbol{x}[0]^\top \boldsymbol{P}\boldsymbol{x}[0] = \sum_{k=0}^{\infty}\left(\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}^*[k]^\top \boldsymbol{R}\boldsymbol{u}^*[k]\right).$$

Thus, if a linear stabilizing controller satisfies the hypothesis of the theorem, then the value of the performance index (9.59) resulting from applying this controller is

$$J = \boldsymbol{x}[0]^\top \boldsymbol{P}\boldsymbol{x}[0].$$

To show that such a controller is an optimal one, we use a proof by contradiction. We assume that the hypothesis of the theorem holds but the controller $\boldsymbol{u}^*$ that satisfies this hypothesis is not optimal, that is, there is a controller $\tilde{\boldsymbol{u}}$ such that

$$J(\tilde{\boldsymbol{u}}) < J(\boldsymbol{u}^*). \tag{9.62}$$

The hypothesis of the theorem implies that

$$\Delta V(\boldsymbol{x}[k])|_{\boldsymbol{u}=\tilde{\boldsymbol{u}}} + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \tilde{\boldsymbol{u}}[k]^\top \boldsymbol{R}\tilde{\boldsymbol{u}}[k] \geq 0. \tag{9.63}$$

We represent (9.63) as

$$\Delta V(\boldsymbol{x}[k])|_{\boldsymbol{u}=\tilde{\boldsymbol{u}}} \geq -\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] - \tilde{\boldsymbol{u}}[k]^\top \boldsymbol{R}\tilde{\boldsymbol{u}}[k].$$

Summing the above from $k=0$ to $k=\infty$ yields

$$V(\boldsymbol{x}[0]) \leq \sum_{k=0}^{\infty}\left(\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \tilde{\boldsymbol{u}}[k]^\top \boldsymbol{R}\tilde{\boldsymbol{u}}[k]\right),$$

that is,

$$J(\tilde{\boldsymbol{u}}) \geq J(\boldsymbol{u}^*),$$

which contradicts (9.62). The proof is complete.

$\square$

Finding an optimal controller involves finding an appropriate quadratic Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}$, which then is used to construct the optimal controller. We first find $\boldsymbol{u}^*$ that minimizes the function

$$f = f(\boldsymbol{u}[k]) = \Delta V(\boldsymbol{x}[k]) + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}[k]^\top \boldsymbol{R}\boldsymbol{u}[k]. \tag{9.64}$$

To proceed, we perform preliminary manipulations on the function f. In particular, we use the definition of ΔV and substitute into (9.64) the system equation (9.58) to obtain

$$\begin{aligned} f(\boldsymbol{u}[k]) &= \boldsymbol{x}[k+1]^\top \boldsymbol{P}\boldsymbol{x}[k+1] - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}[k]^\top \boldsymbol{R}\boldsymbol{u}[k] \\ &= (\boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k])^\top \boldsymbol{P}(\boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k]) - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] \\ &\quad + \boldsymbol{u}[k]^\top \boldsymbol{R}\boldsymbol{u}[k]. \end{aligned} \tag{9.65}$$

Applying to the above function the first-order necessary condition for a relative minimum gives

$$\begin{aligned} \frac{\partial f(\boldsymbol{u}[k])}{\partial \boldsymbol{u}[k]} &= 2(\boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k])^\top \boldsymbol{P}\boldsymbol{B} + 2\boldsymbol{u}[k]^\top \boldsymbol{R} \\ &= 2\boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B} + 2\boldsymbol{u}[k]^\top \left(\boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B}\right) \\ &= \boldsymbol{0}. \end{aligned}$$

The matrix $\boldsymbol{R}+\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{B}$ is positive definite because $\boldsymbol{R}$ is, and therefore the matrix $\boldsymbol{R}+\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{B}$ is invertible. Hence,

$$\boldsymbol{u}^*[k] = -\left(\boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B}\right)^{-1} \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] = -\boldsymbol{K}\boldsymbol{x}[k], \tag{9.66}$$

where

$$\boldsymbol{K} = \left(\boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B}\right)^{-1} \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}.$$

Let

$$\boldsymbol{S} = \boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B}.$$

Then, we represent (9.66) as

$$\boldsymbol{u}^*[k] = -\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k]. \tag{9.67}$$

We next check if $\boldsymbol{u}^*$ satisfies the second-order sufficient condition for a relative minimizer of the function (9.64). We have,

$$\begin{aligned} &\frac{\partial^2}{\partial \boldsymbol{u}^2(k)} \left(\boldsymbol{x}[k+1]^\top \boldsymbol{P}\boldsymbol{x}[k+1] - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}[k]^\top \boldsymbol{R}\boldsymbol{u}[k]\right) \\ &= \frac{\partial}{\partial \boldsymbol{u}[k]} \left(2\boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B} + 2\boldsymbol{u}[k]^\top \left(\boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B}\right)\right) \\ &= 2\left(\boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B}\right) \\ &> 0, \end{aligned} \tag{9.68}$$

that is, $\boldsymbol{u}^*$ satisfies the second-order sufficient condition for a relative minimizer of the function (9.64).

The optimal controller (9.67) can be constructed if we have found an appropriate positive definite matrix $\boldsymbol{P}$. Our next task is to devise a method that would allow us to compute the desired matrix $\boldsymbol{P}$. For this, we first find the equation describing the closed-loop system driven by the optimal controller given by (9.67),

$$\boldsymbol{x}[k+1] = \left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\right)\boldsymbol{x}[k]. \tag{9.69}$$

Our controller satisfies the hypothesis of Theorem 9.4, that is,

$$\boldsymbol{x}[k+1]^\top \boldsymbol{P}\boldsymbol{x}[k+1] - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}^*[k]^\top \boldsymbol{R}\boldsymbol{u}^*[k] = 0.$$

Substituting into the above (9.69), the expression for $\boldsymbol{u}^*$ given by (9.67), and performing simple manipulations gives

$$\begin{aligned}
&\boldsymbol{x}[k]^\top \left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\right)^\top \boldsymbol{P}\left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\right)\boldsymbol{x}[k] - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] \\
&\quad + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] \\
&\quad = \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] - \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] \\
&\qquad - \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] \\
&\qquad - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] \\
&\quad = \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] \\
&\qquad - 2\boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] \\
&\qquad + \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\left(\boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B}\right)\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] \\
&\quad = \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] - \boldsymbol{x}[k]^\top \boldsymbol{P}\boldsymbol{x}[k] + \boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] \\
&\qquad - 2\boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] \\
&\qquad + \boldsymbol{x}[k]^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{x}[k] \\
&\quad = \boldsymbol{x}[k]^\top \left(\boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P} + \boldsymbol{Q} - \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A}\right)\boldsymbol{x}[k] \\
&\quad = 0.
\end{aligned} \tag{9.70}$$

The above equation should hold for any $\boldsymbol{x}$. For this to be true we have to have

$$\boxed{\boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P} + \boldsymbol{Q} - \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{A} = \boldsymbol{O}} \tag{9.71}$$

The above equation is called the *discrete algebraic Riccati equation.*

9.4 Dynamic Programming

The objective of this section is to introduce the reader to the method of dynamic programming invented by Richard E. Bellman (1920–1984) in 1957. Dynamic programming can be used to synthesize optimal controllers for nonlinear, time-varying dynamical systems. We first present dynamic programming for discrete-time systems and then discuss the continuous version of dynamic programming.

9.4.1 Discrete-Time Systems

Dynamic programming is based on the principle of optimality (PO), that was formulated by Bellman as follows:

> *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

In other words, an optimal trajectory has the property that at an intermediate point, no matter how it was reached, the rest of the trajectory must coincide with an optimal trajectory as computed from this intermediate point as the initial point. We justify the PO by contradiction. Suppose that we are given an optimal trajectory starting from initial state $\boldsymbol{x}_0 = \boldsymbol{x}(t_0)$ at time t_0 and terminating at the final state $\boldsymbol{x}(t_f)$ at time t_f. Let $\boldsymbol{x}(t)$ be an intermediate state on the optimal trajectory and let $J^*(t, \boldsymbol{x}(t))$ be the optimal cost of traveling from $\boldsymbol{x}(t)$ to the final state $\boldsymbol{x}(t_f)$. Thus, $J^*(t_0, \boldsymbol{x}(t_0))$ is the optimal cost of traveling from $\boldsymbol{x}_0 = \boldsymbol{x}(t_0)$ to $\boldsymbol{x}(t_f)$. It then follows that the portion of the optimal trajectory from $\boldsymbol{x}(t)$ to $\boldsymbol{x}(t_f)$ is also optimal. For if some other path, indicated by dashed curve in Figure 9.20, connecting $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t_f)$ resulted in a smaller cost than the solid path, then this would provide a less expensive route from $\boldsymbol{x}_0$ to $\boldsymbol{x}(t_f)$, thus contradicting the optimality of the original trajectory.

We note that the PO seems to be already known to Johann Bernoulli who stated in 1706 that "... all curves, which should give a maximum, preserve also in all their sections the laws of the same maximum". Here, the reader should interpret maximum as optimum.

Bellman's principle of optimality serves to "limit the number of potentially optimal control strategies that must be investigated." [210, page 315]. The optimal control strategy is determined, using the PO, by working backward from the final stage.

We illustrate this "backwardness" nature of the PO on an example of a sequential decision making process. We define a decision as the choice of alternative paths leaving a given node.

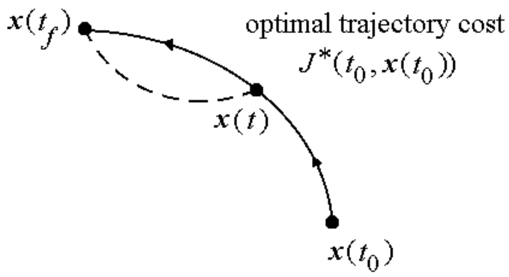


Figure 9.20: Illustration of the Principle of Optimality.

Example 9.16 We use the PO to find the minimum cost path that starts at the node **a** and ends at any one of the nodes **k**, **l**, **m** in the routing network depicted in Figure 9.21. The travel costs are shown beside each path segment. The line segments can only be traversed from left to right. Applying the PO to this routing problem, we find the optimal path by performing a backward pass through the network. Let J_q^* be the minimum cost from a generic node **q** to any one of the three possible final nodes. Then, we have

$$J_g^* = 16, \quad J_h^* = 8, \quad J_i^* = 7, \quad J_j^* = 6.$$

The next stage, yields

$$J_d^* = \min\{3 + J_g^*, 11 + J_h^*\} = \min\{19, 19\} = 19.$$

Thus, in this case, we have two optimal paths from node **d**. We can take either **dgk** or **dhk** paths. We then compute optimal paths from nodes **e** and **f**. We have

$$J_e^* = \min\{6 + J_h^*, 10 + J_i^*\} = \min\{6 + 8, 10 + 7\} = 14,$$

and

$$J_f^* = \min\{4 + J_i^*, 12 + J_j^*\} = \min\{4 + 7, 12 + 6\} = 11.$$

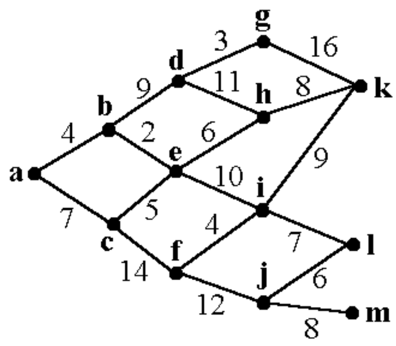


Figure 9.21: A routing network.

Thus, the optimal path emanating from node **e** is **ehk**, and from node **f** is **fil**. The next stage yields

$$J_b^* = \min\{9 + J_d^*, 2 + J_e^*\} = \min\{9 + 19, 2 + 14\} = 16,$$

and

$$J_c^* = \min\{5 + J_e^*, 14 + J_f^*\} = \min\{5 + 14, 14 + 11\} = 19.$$

Therefore, the optimal path starting from **b** is **behk**, and starting from **c** the optimal path is **cehk**. Finally, the initial, and hence final, stage yields

$$J_a^* = \min\{4 + J_b^*, 7 + J_c^*\} = \min\{4 + 16, 7 + 19\} = 20.$$

Hence, the optimal, minimum total cost, route from **a** to any one of the three possible termination nodes **k**, **l**, or **m** is **abehk**. The total cost of this path is 20.

We next consider a class of discrete-time dynamical systems modeled by the difference equation

$$\boldsymbol{x}[k+1] = \boldsymbol{f}\left(k, \boldsymbol{x}[k], \boldsymbol{u}[k]\right), \tag{9.72}$$

with a given initial condition $\boldsymbol{x}[k_0] = \boldsymbol{x}_0$, and the associated performance index

$$J = J_0 = \Phi(N, \boldsymbol{x}[N]) + \sum_{k=0}^{N-1} F(k, \boldsymbol{x}[k], \boldsymbol{u}[k]).$$

Let $J_k^*(\boldsymbol{x}[k])$ denote the minimum cost of transferring the system from $\boldsymbol{x}[k]$ to the terminal point $\boldsymbol{x}[N]$. Because $\boldsymbol{x}[N]$ is the terminal point, therefore $J_N^*(\boldsymbol{x}[N]) = \Phi(N, \boldsymbol{x}[N])$. Using the PO, we obtain the cost of transfer from $\boldsymbol{x}[N-1]$ to the terminal point,

$$\begin{aligned} J_{N-1}^*(\boldsymbol{x}[N-1]) &= \min_{\boldsymbol{u}[N-1]} \{F(N-1, \boldsymbol{x}[N-1], \boldsymbol{u}[N-1]) + J_N^*(\boldsymbol{x}[N])\} \\ &= \min_{\boldsymbol{u}[N-1]} \{F(N-1, \boldsymbol{x}[N-1], \boldsymbol{u}[N-1]) + \Phi(N, \boldsymbol{x}[N])\}. \end{aligned}$$

When applying the PO to solving optimal control problems for discrete-time dynamical systems, we usually perform two stage-by-stage passes through the time stages. We begin with a backward pass. For this, we use (9.72) to eliminate $\boldsymbol{x}[N]$. Next, we carry out the minimization to find $\boldsymbol{u}^*[N-1]$. We then repeat the process. A typical step of the backward pass is described as,

$$\begin{aligned} J_k^*(\boldsymbol{x}[k]) &= \min_{\boldsymbol{u}[k]} \left\{F(k, \boldsymbol{x}[k], \boldsymbol{u}[k]) + J_{k+1}^*(\boldsymbol{x}[k])\right\} \\ &= \min_{\boldsymbol{u}[k]} \left\{F(k, \boldsymbol{x}[k], \boldsymbol{u}[k]) + J_{k+1}^*\left(\boldsymbol{f}(k, \boldsymbol{x}[k], \boldsymbol{u}[k])\right)\right\}. \end{aligned}$$

The backward pass is completed when the initial time k_0 is reached. Now, because $\boldsymbol{x}_0 = \boldsymbol{x}[k_0]$ is known, we find $\boldsymbol{u}_0 = \boldsymbol{u}[k_0]$ in terms of this state. We can then proceed with the *forward pass*. We use $\boldsymbol{x}_0, \boldsymbol{u}_0$, and (9.72) to compute $\boldsymbol{x}[k_0+1]$. The state $\boldsymbol{x}[k_0+1]$ is then used to compute $\boldsymbol{u}[k_0+1]$ from $\boldsymbol{x}[k_0+2] = \boldsymbol{f}(k_0+1, \boldsymbol{x}[k_0+1], \boldsymbol{u}[k_0+1])$, etc.

Example 9.17 Consider a scalar discrete-time dynamical system modeled by

$$x[k+1] = 2x[k] - 3u[k], \quad x[0] = 4,$$

and the performance index

$$J = J_0 = (x[2] - 10)^2 + \frac{1}{2}\sum_{k=0}^{1}\left(x[k]^2 + u[k]^2\right).$$

Note that the final state, in this example, is free. We use the PO to find optimal $u[0]$ and $u[1]$. There are no constraints on $u[k]$. We begin with the backward pass. We have

$$J^*(x[2]) = (x[2] - 10)^2.$$

Then,

$$\begin{aligned} J^*(x[1]) &= \min_{u[1]} \left\{ \frac{1}{2}\left(x[1]^2 + u[1]^2\right) + J^*(x[2]) \right\} \\ &= \min_{u[1]} \left\{ \frac{1}{2}\left(x[1]^2 + u[1]^2\right) + (2x[1] - 3u[1] - 10)^2 \right\}. \end{aligned}$$

Because there are no constraints on $u[1]$, we find optimal $u[1]$ as a function of $x[1]$ by applying the first-order necessary condition for unconstrained optimization to get

$$\frac{\partial}{\partial u[1]} \left\{ \frac{1}{2}\left(x[1]^2 + u[1]^2\right) + (2x[1] - 3u[1] - 10)^2 \right\} = 0.$$

Hence,

$$u[1] = \frac{1}{19}\left(12x[1] - 60\right),$$

and therefore

$$J^*(x[1]) = \frac{1}{2}\left(x[1]^2 + \left(\frac{12x[1]-60}{19}\right)^2\right) + \left(2x[1] - 3\frac{12x[1]-60}{19} - 10\right)^2.$$

Next, we compute $u[0]$. For this observe that

$$J^*(x[0]) = \min_{u[0]} \left\{ \frac{1}{2}\left(x[0]^2 + u[0]^2\right) + J^*(x[1]) \right\}.$$

Taking into account that

$$x[1] = 2x[0] - 3u[0], \quad x[0] = 4,$$

and the fact that there are no constraints on $u[0]$, we again use the first-order necessary condition for unconstrained optimization to find optimal $u[0]$. We compute

$$\frac{\partial}{\partial u[0]} \left\{ \frac{1}{2}\left(x[0]^2 + u[0]^2\right) + J^*(x[1]) \right\} = 0$$

to get

$$\begin{aligned} &\frac{\partial}{\partial u[0]} \left\{ 8 + \frac{1}{2}u[0]^2 + \frac{1}{2}(8 - 3u[0])^2 + \frac{1}{2}\left(\frac{12(8-3u[0])-60}{19}\right)^2 \right\} \\ &+ \frac{\partial}{\partial u[0]} \left\{ 2(8 - 3u[0]) - 3\frac{12(8-3u[0])-60}{19} - 10 \right\}^2 = 0. \end{aligned}$$

Performing some manipulations yields $13.7895u[0] = 27.7895$. Hence, $u[0] = 2.0153$. This completes the backward pass. We are now ready for the forward pass. We use the system difference equation, $x[0]$, and $u[0]$ to compute $x[1] = 1.9541$. Using the above, we find

$$u[1] = \frac{12x[1] - 60}{19} = -1.9237.$$

Therefore,

$$x[2] = 2x[1] - 3u[1] = 9.6794,$$

which completes the forward pass.

9.4.2 Discrete Linear Quadratic Regulator Problem

In Example 9.17 above, we solved a simple discrete linear quadratic regulator problem where we found the open-loop optimal control strategy. We apply now the PO to find the optimal control of a linear state-feedback form for the discrete linear quadratic regulator problem. This problem is stated as follows. Given a discrete linear plant model,

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k], \quad 0 \le k \le N-1,$$

with a specified initial condition $\boldsymbol{x}[0] = \boldsymbol{x}_0$. We wish to calculate the optimal control sequence,

$$\{\boldsymbol{u}^*[0], \boldsymbol{u}^*[1], \ldots, \boldsymbol{u}^*[N-1]\},$$

that minimizes the quadratic performance index

$$J = J_0 = \frac{1}{2}\boldsymbol{x}[N]^\top \boldsymbol{H}_N \boldsymbol{x}[N] + \frac{1}{2}\sum_{k=0}^{N-1} \left\{\boldsymbol{x}[k]^\top \boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}[k]^\top \boldsymbol{R}\boldsymbol{u}[k]\right\},$$

where $\boldsymbol{H}_N = \boldsymbol{H}_N^\top \ge 0$, $\boldsymbol{Q} = \boldsymbol{Q}^\top \ge 0$, and $\boldsymbol{R} = \boldsymbol{R}^\top > 0$. We assume that the components of the control vector are unconstrained. To solve this problem we use the PO. To begin, we note that

$$J^*(\boldsymbol{x}[N]) = \frac{1}{2}\boldsymbol{x}[N]^\top \boldsymbol{H}_N \boldsymbol{x}[N].$$

The above is the penalty for being in a state $\boldsymbol{x}[N]$ at a time N.

We now decrement k to $N-1$ to get

$$J_{N-1}(\boldsymbol{x}[N-1]) = \frac{1}{2}\boldsymbol{x}[N]^\top \boldsymbol{H}_N \boldsymbol{x}[N] + \frac{1}{2}\boldsymbol{x}[N-1]^\top \boldsymbol{Q}\boldsymbol{x}[N-1] + \frac{1}{2}\boldsymbol{u}[N-1]^\top \boldsymbol{R}\boldsymbol{u}[N-1].$$

We use the state equation to eliminate $\boldsymbol{x}[N]$ from J_{N-1} to obtain

$$\begin{aligned} J_{N-1} &= \frac{1}{2}\left(\boldsymbol{A}\boldsymbol{x}[N-1] + \boldsymbol{B}\boldsymbol{u}[N-1]\right)^\top \boldsymbol{H}_N \left(\boldsymbol{A}\boldsymbol{x}[N-1] + \boldsymbol{B}\boldsymbol{u}[N-1]\right) \\ &\quad + \frac{1}{2}\boldsymbol{x}[N-1]^\top \boldsymbol{Q}\boldsymbol{x}[N-1] + \frac{1}{2}\boldsymbol{u}[N-1]^\top \boldsymbol{R}\boldsymbol{u}[N-1]. \end{aligned}$$

Because there are no constraints on control, we can apply the first-order necessary condition from static optimization to find $\boldsymbol{u}^*[N-1]$ as a function of $\boldsymbol{x}[N-1]$,

$$\begin{aligned}\frac{\partial J_{N-1}}{\partial \boldsymbol{u}[N-1]} &= \left(\boldsymbol{A}\boldsymbol{x}[N-1]+\boldsymbol{B}\boldsymbol{u}[N-1]\right)^\top \boldsymbol{H}_N\boldsymbol{B}+\boldsymbol{u}[N-1]^\top\boldsymbol{R}\\ &= \boldsymbol{0}.\end{aligned}$$

Solving for the control satisfying the first-order necessary condition, we obtain

$$\boldsymbol{u}^*[N-1] = -\left(\boldsymbol{R}+\boldsymbol{B}^\top\boldsymbol{H}_N\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{H}_N\boldsymbol{A}\boldsymbol{x}[N-1].$$

Let

$$\boldsymbol{K}_{N-1} = \left(\boldsymbol{R}+\boldsymbol{B}^\top\boldsymbol{H}_N\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{H}_N\boldsymbol{A},$$

then

$$\boldsymbol{u}^*[N-1] = -\boldsymbol{K}_{N-1}\boldsymbol{x}[N-1].$$

Computing the optimal cost of transferring the system from $N-1$ to N, yields

$$\begin{aligned}J^*_{N-1} &= \frac{1}{2}\boldsymbol{x}[N-1]^\top(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_{N-1})^\top\boldsymbol{H}_N(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_{N-1})\boldsymbol{x}[N-1]\\ &\quad+\frac{1}{2}\boldsymbol{x}[N-1]^\top\left(\boldsymbol{K}_{N-1}^\top\boldsymbol{R}\boldsymbol{K}_{N-1}+\boldsymbol{Q}\right)\boldsymbol{x}[N-1].\end{aligned}$$

Let

$$\boldsymbol{H}_{N-1} = (\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_{N-1})^\top\boldsymbol{H}_N(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_{N-1})+\boldsymbol{K}_{N-1}^\top\boldsymbol{R}\boldsymbol{K}_{N-1}+\boldsymbol{Q},$$

then

$$J^*_{N-1} = \frac{1}{2}\boldsymbol{x}[N-1]^\top\boldsymbol{H}_{N-1}\boldsymbol{x}[N-1].$$

Decrementing k to $N-2$ yields

$$J_{N-2} = \frac{1}{2}\boldsymbol{x}[N-1]^\top\boldsymbol{H}_{N-1}\boldsymbol{x}[N-1]+\frac{1}{2}\boldsymbol{x}[N-2]^\top\boldsymbol{Q}\boldsymbol{x}[N-2]+\frac{1}{2}\boldsymbol{u}[N-2]^\top\boldsymbol{R}\boldsymbol{u}[N-2].$$

Note that J_{N-2} has the same form as J_{N-1}. Thus, we obtain optimal feedback gain where N is being replaced with $N-1$. Continuing in this fashion, we get the following results for each $k = N-1, N-2, \ldots, 0$:

$$\begin{aligned}\boldsymbol{K}_k &= \left(\boldsymbol{R}+\boldsymbol{B}^\top\boldsymbol{H}_{k+1}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{H}_{k+1}\boldsymbol{A},\\ \boldsymbol{u}^*[k] &= -\boldsymbol{K}_k\boldsymbol{x}[k],\\ \boldsymbol{H}_k &= (\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_k)^\top\boldsymbol{H}_{k+1}(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_k)+\boldsymbol{K}_k^\top\boldsymbol{R}\boldsymbol{K}_k+\boldsymbol{Q},\end{aligned}$$

and

$$J^*_k = \frac{1}{2}\boldsymbol{x}[k]^\top\boldsymbol{H}_k\boldsymbol{x}[k].$$

The above control scheme can be implemented by computing the sequence of gain matrices $\{\boldsymbol{K}_k\}$ off-line and stored. Then, we can implement the controller $\boldsymbol{u}^*[k] = -\boldsymbol{K}_k\boldsymbol{x}[k]$. Note that the time-varying nature of the feedback controller may make an implementation of the control strategy quite complicated, especially for large n and N. However, as we go to the limit as $N \to \infty$, we can simplify the solution to the problem. Specifically, let us consider a (reachable) system model

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k], \quad k \geq 0, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$

and the performance index

$$J = \frac{1}{2}\sum_{k=0}^{\infty}\left\{\boldsymbol{x}[k]^\top\boldsymbol{Q}\boldsymbol{x}[k] + \boldsymbol{u}[k]^\top\boldsymbol{R}\boldsymbol{u}[k]\right\}$$

obtained from the previously considered performance index by letting $N \to \infty$ and setting $\boldsymbol{H}_N = \boldsymbol{0}$. Then, the optimal controller takes the form

$$\boldsymbol{u}^*[k] = -\boldsymbol{K}_\infty\boldsymbol{x}[k], \quad k \geq 0,$$

where

$$\boldsymbol{K}_\infty = \left(\boldsymbol{R} + \boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{A} \tag{9.73}$$

and $\boldsymbol{H}_\infty$ is defined as the limit obtained by continuing the recursion

$$\boldsymbol{H}_k = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_k)^\top\boldsymbol{H}_{k+1}(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_k) + \boldsymbol{K}_k^\top\boldsymbol{R}\boldsymbol{K}_k + \boldsymbol{Q} \tag{9.74}$$

back to $k = -\infty$, that is,

$$\lim_{k\to-\infty}\boldsymbol{H}_k = \boldsymbol{H}_\infty.$$

Alternatively, letting $k \to -\infty$ in (9.74), we obtain

$$\boldsymbol{H}_\infty = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_\infty)^\top\boldsymbol{H}_\infty(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_\infty) + \boldsymbol{K}_\infty^\top\boldsymbol{R}\boldsymbol{K}_\infty + \boldsymbol{Q}, \tag{9.75}$$

where $\boldsymbol{K}_\infty$ is given by (9.73). Substituting (9.73) into (9.75) yields

$$\begin{aligned}
&\boldsymbol{H}_\infty \\
&= \left(\boldsymbol{A} - \boldsymbol{B}\left(\boldsymbol{R} + \boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{A}\right)^\top\boldsymbol{H}_\infty\left(\boldsymbol{A} - \boldsymbol{B}\left(\boldsymbol{R} + \boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{A}\right) \\
&\quad + \left(\left(\boldsymbol{R} + \boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{A}\right)^\top\boldsymbol{R}\left(\left(\boldsymbol{R} + \boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{H}_\infty\boldsymbol{A}\right) + \boldsymbol{Q}.
\end{aligned} \tag{9.76}$$

Thus, we can compute $\boldsymbol{H}_\infty$ by solving the above algebraic equation. We note that the above formidable looking equation is just the discrete algebraic Riccati equation. Indeed, setting $\boldsymbol{P} = \boldsymbol{H}_\infty$, using the notation,

$$\boldsymbol{S} = \boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{H}_\infty \boldsymbol{B} = \boldsymbol{R} + \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{B},$$

and performing simple manipulations, we obtain (9.71).

We next discuss a continuous form of dynamic programming. We begin our discussion by analyzing the minimum time regulator problem. Then, we focus our attention on a more general optimal control problem.

9.4.3 Continuous Minimum Time Regulator Problem

In this subsection, we consider a control system modeled by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$$

with a given initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$. We restrict the controller $\boldsymbol{u}(t)$ to be restrained by the compact set $U \subset \mathbb{R}^m$. For example, we may restrict the components u_i, $i = 1, 2, \ldots, m$ of the control vector $\boldsymbol{u}$ to satisfy the amplitude constraints

$$|u_i| \leq k_i,$$

where k_i are fixed real numbers. A piecewise continuous control vector $\boldsymbol{u}$ satisfying the above constraints is called admissible. The set of admissible controllers is denoted U. Our goal is to determine an admissible control law $\boldsymbol{u}^*$ that steers the system's trajectory from a given initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ to a specified final state $\boldsymbol{x}_f$ in the shortest possible time. This is equivalent to minimizing the performance index

$$J = \int_{t_0}^{t_f} dt = t_f - t_0.$$

We call such a controller *time optimal*. We assume that for any $\boldsymbol{x} \neq \boldsymbol{x}_f$ there exists admissible, time optimal control transferring the system from $\boldsymbol{x}$ to $\boldsymbol{x}_f$ in minimum time. We denote by $J^*(t, \boldsymbol{x}) = J^*(t, \boldsymbol{x}(t))$ the time of the optimal transfer from $\boldsymbol{x}(t)$ to the specified $\boldsymbol{x}_f$. Observe that $J^*(t, \boldsymbol{x}) = J^*(t, x_1, x_2 \ldots, x_n)$, that is,

$$J^* : \mathbb{R}^{n+1} \to \mathbb{R},$$

which means that J^* is a real valued function of $n+1$ variables. We assume that the function J^* is continuous and has continuous partial derivatives everywhere except at the point $\boldsymbol{x}_f$.

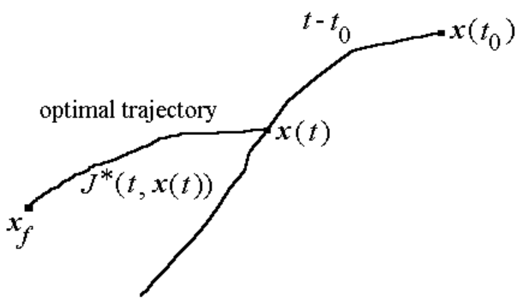


Figure 9.22: Illustration of the derivation of the condition that a controller must satisfy when transferring a dynamical system from an initial state to the final state in an optimal manner.

Now, let $\boldsymbol{x}_0 = \boldsymbol{x}(t_0) \neq \boldsymbol{x}_f$ be an arbitrary initial state in $\mathbb{R}^n$. Assume that we applied some admissible $\boldsymbol{u}_0 \in U$ on the time interval $[t_0, t]$. As a result of this control action, the system is transferred from $\boldsymbol{x}(t_0)$ to $\boldsymbol{x}(t)$. The time spent on the motion from $\boldsymbol{x}(t_0)$ to $\boldsymbol{x}(t)$ is $t - t_0$. We then apply an admissible optimal control that transfers the system in minimum time $J^*(t, \boldsymbol{x}(t))$ from $\boldsymbol{x}(t)$ to $\boldsymbol{x}_f$. The above considerations are illustrated in Figure 9.22. Denoting the optimal time of transferring the system from $\boldsymbol{x}(t_0)$ to $\boldsymbol{x}_f$ by $J^*(t_0, \boldsymbol{x}(t_0))$, we obtain

$$(t - t_0) + J^*(t, \boldsymbol{x}(t)) \geq J^*(t_0, \boldsymbol{x}(t_0)).$$

Rearranging the terms of the above equation and dividing by $(t - t_0)$ yields

$$1 + \frac{J^*(t, \boldsymbol{x}(t)) - J^*(t_0, \boldsymbol{x}(t_0))}{t - t_0} \geq 0.$$

Letting $t \to t_0$, and taking into account that $\dot{x}_i = f_i(t, \boldsymbol{x}, \boldsymbol{u})$, $i = 1, 2, \ldots, n$, we obtain

$$\begin{aligned} 1 + \frac{\partial J^*(t, \boldsymbol{x})}{\partial t} + \sum_{i=1}^{n} \frac{\partial J^*(t, \boldsymbol{x})}{\partial x_i} \dot{x}_i &= 1 + \frac{\partial J^*(t, \boldsymbol{x})}{\partial t} + \sum_{i=1}^{n} \frac{\partial J^*(t, \boldsymbol{x})}{\partial x_i} f_i \\ &= 1 + \frac{\partial J^*}{\partial t} + \frac{\partial J^*}{\partial \boldsymbol{x}} \boldsymbol{f} \\ &\geq 0. \end{aligned}$$

We conclude that for any $\boldsymbol{x}_0 \neq \boldsymbol{x}_f$ and for any $\boldsymbol{u} \in U$,

$$1 + \frac{\partial J^*}{\partial t} + \frac{\partial J^*}{\partial \boldsymbol{x}} \boldsymbol{f} \geq 0.$$

We next define the Hamiltonian function for the time optimal control problem to be

$$H(t, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) = 1 + \boldsymbol{p}^\top \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}),$$

where

$$\boldsymbol{p} = \begin{bmatrix} \frac{\partial J^*}{\partial x_1} & \frac{\partial J^*}{\partial x_2} & \cdots & \frac{\partial J^*}{\partial x_n} \end{bmatrix}^\top.$$

Thus, we can say that for any $\boldsymbol{x}_0 \neq \boldsymbol{x}_f$ and for any $\boldsymbol{u} \in U$,

$$\frac{\partial J^*(t, \boldsymbol{x})}{\partial t} + H(t, \boldsymbol{x}, \boldsymbol{u}) \geq 0.$$

We will use the PO to show that

$$\frac{\partial J^*(t, \boldsymbol{x})}{\partial t} + H(t, \boldsymbol{x}, \boldsymbol{u}) = 0$$

for any optimal transfer, that is,

$$\frac{\partial J^*(t, \boldsymbol{x})}{\partial t} + \min_{\boldsymbol{u} \in U} H(t, \boldsymbol{x}, \boldsymbol{u}) = 0 \quad \text{for any} \quad \boldsymbol{x} \neq \boldsymbol{x}_f.$$

Indeed, suppose that $\boldsymbol{u}(t) \in U$ is an optimal controller that transfers the system from the initial state $\boldsymbol{x}(t_0) \neq \boldsymbol{x}_f$ to the given final state $\boldsymbol{x}_f$ in a time t_f. Thus, we have $J^*(t_0, \boldsymbol{x}(t_0)) = t_f - t_0$. The transfer from $\boldsymbol{x}(t)$ to $\boldsymbol{x}_f$ requires $(t_f - t)$ time units. See Figure 9.23 for an illustration of the arguments. Hence,

$$t_f - t = J^*(t_0, \boldsymbol{x}(t_0)) - (t - t_0),$$

and it is not possible to go from $\boldsymbol{x}(t)$ to $\boldsymbol{x}_f$ in a time less than $J^*(t_0, \boldsymbol{x}(t_0)) - (t - t_0)$. For if such a more rapid motion were to exist, indicated by the dashed curve in Figure 9.23, then moving from $\boldsymbol{x}_0$ to $\boldsymbol{x}(t)$ during the time $t - t_0$, and then from $\boldsymbol{x}(t)$ to $\boldsymbol{x}_f$ during a time less than $J^*(t_0, \boldsymbol{x}(t_0)) - (t - t_0)$, we would accomplish the transfer from $\boldsymbol{x}_0$ to $\boldsymbol{x}_f$ during a time less than $J^*(t_0, \boldsymbol{x}(t_0))$, which is impossible. Therefore, we can write

$$\begin{aligned} t_f - t &= J^*(t, \boldsymbol{x}(t)) \\ &= J^*(t_0, \boldsymbol{x}(t_0)) - (t - t_0). \end{aligned}$$

Rearranging terms, dividing by $t - t_0$, and letting $t \to t_0$, yields

$$\frac{\partial J^*(t, \boldsymbol{x})}{\partial t} + \min_{\boldsymbol{u} \in U} H(t, \boldsymbol{x}, \boldsymbol{u}) = 0 \quad \text{for any} \quad \boldsymbol{x} \neq \boldsymbol{x}_f.$$

We summarize our considerations in the following theorem.

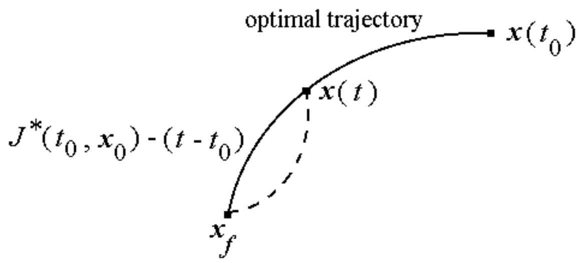


Figure 9.23: Illustration of the derivation of the Hamilton-Jacobi-Bellman equation.

Theorem 9.5 *Let $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$ be a control system model, U a set of admissible controllers, and $\boldsymbol{x}_f$ a prescribed final state. Suppose that*

(i) for any $\boldsymbol{x} \neq \boldsymbol{x}_f$ there exists an admissible optimal control transferring $\boldsymbol{x}$ to $\boldsymbol{x}_f$,

(ii) the function $J^*(t, \boldsymbol{x}(t))$, which is the time of optimal transfer of the system from $\boldsymbol{x}$, at time t, to $\boldsymbol{x}_f$, is continuously differentiable except at the point $\boldsymbol{x}_f$,

then

$$\begin{aligned} \frac{\partial J^*(t, \boldsymbol{x})}{\partial t} + H(t, \boldsymbol{x}, \boldsymbol{u}) &\geq 0 \quad \text{for any} \quad \boldsymbol{u} \in U \quad \text{and} \quad \boldsymbol{x} \neq \boldsymbol{x}_f, \\ \frac{\partial J^*(t, \boldsymbol{x})}{\partial t} + \min_{\boldsymbol{u} \in U} H(t, \boldsymbol{x}, \boldsymbol{u}) &= 0 \quad \text{for any optimal transfer, where} \quad \boldsymbol{x} \neq \boldsymbol{x}_f. \end{aligned}$$

The above theorem is a special case of a more general result known as the continuous form of the dynamic programming. We now use the PO to derive a continuous form of dynamic programming for a class of problems with more general performance indices.

9.4.4 The Hamilton-Jacobi-Bellman (HJB) Equation

We consider a control process model of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{9.77}$$

and we consider the performance index that we wish to minimize

$$J(t_0, \boldsymbol{x}(t_0), \boldsymbol{u}) = \Phi(t_f, \boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} F(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau))d\tau. \tag{9.78}$$

Define

$$J(t, \boldsymbol{x}(t), \boldsymbol{u}(\tau)) = \Phi(t_f, \boldsymbol{x}(t_f)) + \int_{t}^{t_f} F(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau))d\tau$$

with $t \leq \tau \leq t_f$, and let

$$J^*(t, \boldsymbol{x}(t)) = \min_{\boldsymbol{u}} J(t, \boldsymbol{x}(t), \boldsymbol{u}(\tau)).$$

We subdivide the interval $[t, t_f]$ as

$$[t, t_f] = [t, t + \Delta t] \cup [t + \Delta t, t_f].$$

Then, we write

$$J^*(t, \boldsymbol{x}(t), \boldsymbol{u}(\tau)) = \min_{\boldsymbol{u}} \left\{ \int_t^{t+\Delta t} F d\tau + \int_{t+\Delta t}^{t_f} F d\tau + \Phi(t_f, \boldsymbol{x}(t_f)) \right\}.$$

By the PO the trajectory on the interval $[t + \Delta t, t_f]$ must be optimal. Hence,

$$J^*(t, \boldsymbol{x}(t)) = \min_{\boldsymbol{u}} \left\{ \int_t^{t+\Delta t} F d\tau + J^*(t + \Delta t, \boldsymbol{x}(t + \Delta t)) \right\}.$$

We next expand $J^*(t + \Delta t, \boldsymbol{x}(t + \Delta t))$ in a Taylor series about the point $(t, \boldsymbol{x}(t))$ to get

$$J^*(t, \boldsymbol{x}(t)) = \min_{\boldsymbol{u}} \left\{ \int_t^{t+\Delta t} F d\tau + J^* + \frac{\partial J^*}{\partial t}\Delta t + \frac{\partial J^*}{\partial \boldsymbol{x}} (\boldsymbol{x}(t + \Delta t) - \boldsymbol{x}(t)) + \text{H.O.T.} \right\},$$

where H.O.T. stands for higher order terms. Because $J^*(t, \boldsymbol{x}(t))$ is independent of $\boldsymbol{u}$, we cancel the terms $J^*(t, \boldsymbol{x}(t))$ out and use the fact that $\boldsymbol{x}(t + \Delta t) - \boldsymbol{x}(t) \approx \dot{\boldsymbol{x}}\Delta t$ to obtain

$$0 = \min_{\boldsymbol{u}} \left\{ \int_t^{t+\Delta t} F d\tau + \frac{\partial J^*}{\partial t}\Delta t + \frac{\partial J^*}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}\Delta t + \text{H.O.T.} \right\}.$$

By assumption Δt is small, hence we can write

$$0 = \min_{\boldsymbol{u}} \left\{ F\Delta t + \frac{\partial J^*}{\partial t}\Delta t + \frac{\partial J^*}{\partial \boldsymbol{x}}\boldsymbol{f}\Delta t + \text{H.O.T.} \right\}.$$

Dividing the above equation by Δt and letting $\Delta t \to 0$ yields

$$0 = \frac{\partial J^*}{\partial t} + \min_{\boldsymbol{u}} \left\{ F + \frac{\partial J^*}{\partial \boldsymbol{x}}\boldsymbol{f} \right\}. \tag{9.79}$$

Let $H = F + \frac{\partial J^*}{\partial \boldsymbol{x}} \boldsymbol{f}$ denote the Hamiltonian function. Then, we write the partial differential equation (9.79) as

$$\boxed{0 = \frac{\partial J^*}{\partial t} + \min_{\boldsymbol{u}} H} \tag{9.80}$$

subject to the boundary condition

$$J^*(t_f, \boldsymbol{x}(t_f)) = \Phi(t_f, \boldsymbol{x}(t_f)).$$

The partial differential equation (9.80) for the optimal cost $J^*(t, \boldsymbol{x}(t))$ is called the *Hamilton-Jacobi-Bellman* (HJB) equation. It provides the solution to the optimal control problem for general nonlinear dynamical systems. However, analytical solution to the HJB equation is difficult to obtain in most cases. We now illustrate, with three examples, how the HJB equation can be used to construct optimal controllers.

Example 9.18 We consider a general class of dynamical systems modeled by the scalar differential equation

$$\dot{x} = ax + bu, \quad x(t_0) = x_0,$$

with the associated cost index

$$J(t_0) = \frac{1}{2} f x(t_f)^2 + \frac{1}{2} \int_{t_0}^{t_f} \left(q x(t)^2 + r u(t)^2 \right) dt,$$

where the initial time $t_0 = 0$, the final time $t_f < \infty$ is fixed, and the final state $x(t_f)$ is free. Our goal is to construct the control u^* that minimizes the cost index J.

We first form the Hamiltonian function for the problem,

$$H\left(t, x, u, \frac{\partial J^*}{\partial x}\right) = \frac{1}{2} q x^2 + \frac{1}{2} r u^2 + \frac{\partial J^*}{\partial x}(ax + bu).$$

Because there are no constraints on the control u, we can determine the form of the optimal control by applying the first order necessary condition for static optimization. We compute

$$\frac{\partial H}{\partial u} = ru + b\frac{\partial J^*}{\partial x} = 0.$$

The optimal control, therefore, has the form

$$u^* = -\frac{1}{r} b \frac{\partial J^*}{\partial x}.$$

This control indeed minimizes the Hamiltonian function H because

$$\frac{\partial^2 H}{\partial u^2} = r > 0,$$

that is, the second order sufficient condition for static minimization is satisfied. We thus reduced the problem of constructing the optimal control to that of finding $\frac{\partial J^*}{\partial x}$. Substituting u^* into the HJB equation yields

$$\begin{aligned}
&\frac{\partial J^*}{\partial t}+\frac{1}{2}qx^2+\frac{b^2}{2r}\left(\frac{\partial J^*}{\partial x}\right)^2+ax\frac{\partial J^*}{\partial x}-\frac{b^2}{r}\left(\frac{\partial J^*}{\partial x}\right)^2\\
&\quad=\frac{\partial J^*}{\partial t}+\frac{1}{2}qx^2-\frac{b^2}{2r}\left(\frac{\partial J^*}{\partial x}\right)^2+ax\frac{\partial J^*}{\partial x}\\
&\quad=0.
\end{aligned}$$

Let us assume that J^* is a quadratic function of the state for all $t \le t_f$, that is,

$$J^*=\frac{1}{2}p(t)x^2,$$

where $p(t)$ is a scalar function of time to be determined. Partial derivatives of J^* are

$$\frac{\partial J^*}{\partial t}=\frac{1}{2}\dot{p}(t)x^2 \quad \text{and} \quad \frac{\partial J^*}{\partial x}=p(t)x.$$

Substituting the above into the HJB equation and performing some manipulations gives

$$\left(\frac{1}{2}\dot{p}(t)+\frac{1}{2}q+ap(t)-\frac{b^2p(t)^2}{2r}\right)x^2=0.$$

The above holds for all $x=x(t)$ if and only if

$$\frac{1}{2}\dot{p}(t)+\frac{1}{2}q+ap(t)-\frac{b^2p(t)^2}{2r}=0,$$

or equivalently

$$\dot{p}(t)+q+2ap(t)-\frac{b^2p(t)^2}{r}=0 \tag{9.81}$$

subject to the boundary condition $p(t_f)=f$ that we obtain by comparing $J^*=\frac{1}{2}p(t)x^2$ and $J(t_f)=\frac{1}{2}fx(t_f)^2$. To solve the above equation we assume that

$$p(t)=\rho\frac{\dot{w}(t)}{w(t)},$$

where the parameter ρ is to be determined. Note that

$$\dot{p}(t)=\rho\frac{\ddot{w}(t)w(t)-\dot{w}(t)^2}{w(t)^2}.$$

Substituting the above into (9.81), we obtain

$$\begin{aligned}
&\dot{p}+q+2ap-\frac{b^2p^2}{r}\\
&=\rho\frac{\ddot{w}w-\dot{w}^2}{w^2}+q\frac{w^2}{w^2}+2a\rho\frac{\dot{w}}{w}-\frac{b^2\rho^2\dot{w}^2/r}{w^2}\\
&=\frac{\rho\ddot{w}w-\rho\dot{w}^2+qw^2+2a\rho\dot{w}w-b^2\rho^2\dot{w}^2/r}{w^2}\\
&=0.
\end{aligned}$$

We choose ρ so that the terms nonlinear in $\dot{w}$ cancel out. For this it is enough that

$$-\rho\dot{w}^2-b^2\rho^2\dot{w}^2/r=0.$$

Hence, if we set

$$\rho=-\frac{r}{b^2},$$

then the equation to be solved takes the form

$$\frac{\left(-\frac{r}{b^2}\ddot{w}-\frac{2ar}{b^2}\dot{w}+qw\right)w}{w^2}=0.$$

The above is satisfied if

$$-\frac{r}{b^2}\ddot{w}-\frac{2ar}{b^2}\dot{w}+qw=0,$$

or equivalently

$$\ddot{w}+2a\dot{w}-\frac{qb^2}{r}w=0,$$

which is just a second order linear differential equation. Its characteristic equation has two real roots:

$$s_{1,2}=-a\pm\sqrt{a^2+\frac{qb^2}{r}}.$$

Hence

$$w(t)=C_1e^{s_1t}+C_2e^{s_2t},$$

and

$$p(t)=\rho\frac{\dot{w}(t)}{w(t)}=\rho\frac{s_1C_1e^{s_1t}+s_2C_2e^{s_2t}}{C_1e^{s_1t}+C_2e^{s_2t}}.$$

Using the boundary value $p(t_f)=f$, we express C_1 in terms of C_2 to obtain

$$C_1=\frac{\rho s_2C_2e^{s_2t_f}-fC_2e^{s_2t_f}}{fe^{s_1t_f}-\rho s_1e^{s_1t_f}}.$$

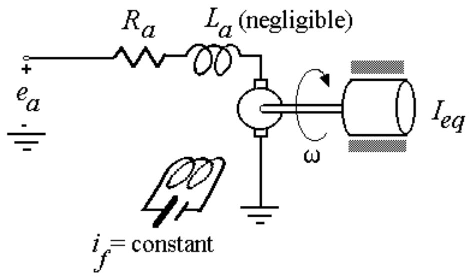


Figure 9.24: Schematic of an armature-controlled DC motor system of Example 9.19.

We substitute the above expression for C_1 into $p(t)$ and cancel out C_2. Then, we use the obtained result to construct the optimal state-feedback controller

$$\boxed{u^* = -\frac{1}{r}b\frac{\partial J^*}{\partial x} = -\frac{1}{r}bp(t)x}$$

Example 9.19 Consider the schematic of an armature controlled DC motor, given in Figure 9.24, where the system parameters are: $R_a = 2\,\Omega$, $L_a \approx 0\,\mathrm{H}$, $K_b = 2\,\mathrm{V/rad/sec}$, $K_i = 2\,\mathrm{Nm/A}$, and the equivalent moment of inertia referred to the motor shaft is $I_{eq} = 1$ kg·m^2. The friction is assumed to be negligible.

(i) Construct the first-order differential equation modeling the DC motor;

(ii) Use the Hamilton-Jacobi-Bellman equation to find the optimal state feedback controller, $e_a = k(t)\omega$, that minimizes the performance index

$$J = \frac{1}{2}\omega(10)^2 + \int_0^{10} R_a i_a(t)^2 dt,$$

where i_a is the armature current. The final state is free. There are no constraints on e_a. Assume $J^* = \frac{1}{2}p(t)\omega^2$.

Applying Kirchhoff's voltage law to the armature circuit yields

$$R_a i_a + K_b\omega = e_a.$$

The torque developed by the motor, T_m, is

$$I_{eq}\dot{\omega} = T_m = K_i i_a.$$

Substituting i_a from the first equation into the second and dividing both sides by I_{eq} yields

$$\dot{\omega} = -\frac{K_i K_b}{R_a I_{eq}}\omega + \frac{K_i}{I_{eq}R_a}e_a.$$

Substituting into the above the parameter values, we obtain

$$\dot{\omega} = -2\omega + e_a. \tag{9.82}$$

We represent the performance index, J, in terms of ω and e_a. Applying Ohm's law to the armature circuit, we get

$$e_a - K_b\omega = R_a i_a.$$

Hence,

$$J = \frac{1}{2}\omega(10)^2 + \int_0^{10} \frac{(e_a - K_b\omega)^2}{R_a} dt.$$

The Hamiltonian function is

$$H = \frac{(e_a - K_b\omega)^2}{R_a} + \frac{\partial J^*}{\partial \omega}\left(-\frac{K_i K_b}{R_a I_{eq}}\omega + \frac{K_i}{I_{eq}R_a}e_a\right) = \frac{(e_a - 2\omega)^2}{2} + \frac{\partial J^*}{\partial \omega}\left(-2\omega + e_a\right).$$

Because there are no constraints on e_a, we can find the optimal control by solving the equation

$$0 = \frac{\partial H}{\partial e_a} = \frac{2\left(e_a - K_b\omega\right)}{R_a} + \frac{K_i}{I_{eq}R_a}\frac{\partial J^*}{\partial \omega}.$$

Hence,

$$e_a^* = 2\omega - \frac{\partial J^*}{\partial \omega}. \tag{9.83}$$

Substituting (9.83) into the HJB equation yields

$$0 = \frac{\partial J^*}{\partial t} + \frac{1}{2}\left(-\frac{\partial J^*}{\partial \omega}\right)^2 + \frac{\partial J^*}{\partial \omega}\left(-\frac{\partial J^*}{\partial \omega}\right) = \frac{\partial J^*}{\partial t} - \frac{1}{2}\left(\frac{\partial J^*}{\partial \omega}\right)^2.$$

By assumption $J^* = \frac{1}{2}p(t)\omega^2$. Hence,

$$\frac{\partial J^*}{\partial t} = \frac{1}{2}\dot{p}\omega^2 \quad \text{and} \quad \frac{\partial J^*}{\partial \omega} = p\omega.$$

Substituting the above into the HJB equation, we get

$$\frac{1}{2}\left(\dot{p} - p^2\right)\omega^2 = 0.$$

We next solve the nonlinear differential equation

$$\dot{p} - p^2 = 0. \tag{9.84}$$

We assume that $p = \rho\frac{\dot{w}}{w}$. Therefore,

$$\dot{p} = \rho\frac{\ddot{w}w - \dot{w}^2}{w^2}.$$

We substitute the expressions for p and $\dot{p}$ into (9.84) to get

$$\frac{\rho\ddot{w}w - \rho\dot{w}^2 - \rho^2\dot{w}^2}{w^2} = 0.$$

Selecting $\rho = -1$ eliminates the nonlinear terms in the numerator and we are left with the equation

$$\ddot{w} = 0,$$

whose solution is

$$w = w(t) = C_1 t + C_2,$$

where C_1 and C_2 are integration constants. Hence,

$$p = -\frac{\dot{w}}{w} = -\frac{C_1}{C_1 t + C_2}.$$

Because

$$J^*(10) = \frac{1}{2}\omega(10)^2 = \frac{1}{2}p(10)\omega(10)^2,$$

we conclude that $p(10) = 1$. We use this information to eliminate one of the integration constants. We obtain

$$p(10) = 1 = -\frac{C_1}{10C_1 + C_2}.$$

Hence,

$$C_2 = -11C_1,$$

and

$$p = p(t) = -\frac{C_1}{C_1 t + C_2} = -\frac{C_1}{C_1 t - 11C_1} = -\frac{1}{t-11}.$$

Therefore,

$$\boxed{e_a^* = 2\omega - \frac{\partial J^*}{\partial \omega} = 2\omega - p\omega = \left(2 + \frac{1}{t-11}\right)\omega}$$

In the above examples, the final time t_f was fixed and $t_f < \infty$. In the following example, we consider the linear quadratic regulator design for $t_f = \infty$ using the HJB equation.

Example 9.20 For the dynamical system model,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u,$$

and the performance index

$$J = \frac{1}{2} \int_0^\infty \left(x_1^2 + x_2^2 + u^2\right) dt,$$

we will use the HJB equation to find the optimal state-feedback controller. We first form the Hamiltonian function

$$H = \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right) + \frac{\partial J^*}{\partial x_1} x_2 + \frac{\partial J^*}{\partial x_2}(-x_2 + u).$$

We assume that J^* has the form

$$J^* = \frac{1}{2}\left(\alpha x_1^2 + 2\beta x_1 x_2 + \gamma x_2^2\right).$$

Hence,

$$\frac{\partial J^*}{\partial t} = 0,$$

and the HJB equation takes the form

$$\min_u H = 0,$$

with the boundary condition

$$J^*(\boldsymbol{x}(t_f)) = J^*(\boldsymbol{x}(\infty)) = 0.$$

Because there are no constraints on u, we apply the first-order necessary condition for unconstrained optimization to find u minimizing H,

$$\frac{\partial H}{\partial u} = 0.$$

We denote by u^* the solution to the above equation. Differentiating, we obtain

$$u^* + \frac{\partial J^*}{\partial x_2} = 0.$$

Hence,

$$u^* = -\frac{\partial J^*}{\partial x_2}.$$

We substitute u^* into the HJB equation to get

$$\frac{1}{2}\left(x_1^2 + x_2^2 + \left(\frac{\partial J^*}{\partial x_2}\right)^2\right) + \frac{\partial J^*}{\partial x_1}x_2 - \frac{\partial J^*}{\partial x_2}x_2 - \left(\frac{\partial J^*}{\partial x_2}\right)^2 = 0.$$

Note that

$$\frac{\partial J^*}{\partial x_1} = \alpha x_1 + \beta x_2 \quad \text{and} \quad \frac{\partial J^*}{\partial x_2} = \beta x_1 + \gamma x_2.$$

Substituting the above into the HJB equation yields

$$\frac{1}{2}\left(x_1^2 + x_2^2 + (\beta x_1 + \gamma x_2)^2\right) + (\alpha x_1 + \beta x_2)x_2 - (\beta x_1 + \gamma x_2)x_2 - (\beta x_1 + \gamma x_2)^2 = 0.$$

We rearrange the above equation to obtain

$$\frac{1}{2}x_1^2(1-\beta^2) + x_1x_2(\alpha - \beta - \beta\gamma) + \frac{1}{2}x_2^2(1 - 2\gamma - \gamma^2 + 2\beta) = 0.$$

For this equation to hold, we have to have

$$\left.\begin{array}{lcl} 1-\beta^2 & = & 0 \\ \alpha - \beta - \beta\gamma & = & 0 \\ 1 - 2\gamma - \gamma^2 + 2\beta & = & 0. \end{array}\right\}$$

There are three solutions to the above system of equations:

1. $\alpha = 2$, $\beta = \gamma = 1$;

2. $\alpha = -2$, $\beta = 1$, $\gamma = -3$;

3. $\alpha = 0$, $\beta = \gamma = -1$.

Thus, we have three candidate solutions to the problem:

$$\begin{aligned} u^{(1)} &= -x_1 - x_2, \\ u^{(2)} &= -x_1 + 3x_2, \\ u^{(3)} &= x_1 + x_2. \end{aligned}$$

However, only $u^{(1)}$ stabilizes the plant. Hence, the optimal control law for the problem is

$$u^* = -x_1 - x_2.$$

Note, that only for this controller the boundary condition, $J^*(\boldsymbol{x}(t_f)) = J^*(\boldsymbol{x}(\infty)) = 0$, is satisfied.

9.5 Pontryagin's Minimum Principle

9.5.1 Optimal Control With Constraints on Inputs

In this section, we provide a derivation of the minimum principle of Pontryagin, which is a generalization of the Euler-Lagrange equations that also includes problems with constraints on the control inputs. Only a special case of the minimum principle is stated. However, this special case covers a large class of control problems. We consider the problem of minimizing the performance index given by

$$J = \Phi(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t))dt \tag{9.85}$$

subject to

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad \text{and} \quad \boldsymbol{x}(t_f) = \boldsymbol{x}_f. \tag{9.86}$$

We consider two cases: fixed final state and free final state.

To proceed we define the Hamiltonian function $H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p})$ as

$$H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) = H = F + \boldsymbol{p}^\top \boldsymbol{f},$$

where the costate vector $\boldsymbol{p}$ will be determined in the analysis to follow.

We discuss the case when the final time t_f is fixed. We follow the development of Luenberger [223, Section 11.1]. We "adjoin" to J additional terms that sum up to zero. We note that because the state trajectories must satisfy the equation, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$, we have

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \dot{\boldsymbol{x}} = \boldsymbol{0}. \tag{9.87}$$

We introduce the modified objective performance index,

$$\tilde{J} = J + \int_{t_0}^{t_f} \boldsymbol{p}(t)^\top (\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \dot{\boldsymbol{x}})\, dt. \tag{9.88}$$

Because for any state trajectory, (9.87) is satisfied, for any choice of $\boldsymbol{p}(t)$ the value of $\tilde{J}$ is the same as that of J. Therefore, we can express $\tilde{J}$ as

$$\begin{aligned} \tilde{J} &= \Phi(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t))\, dt + \int_{t_0}^{t_f} \boldsymbol{p}(t)^\top (\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \dot{\boldsymbol{x}})\, dt \\ &= \Phi(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} (H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) - \boldsymbol{p}^\top \dot{\boldsymbol{x}})\, dt. \end{aligned} \tag{9.89}$$

Let $\boldsymbol{u}(t)$ be a nominal control strategy; it determines a corresponding state trajectory $\boldsymbol{x}(t)$. If we apply another control strategy, say $\boldsymbol{v}(t)$, that is "close" to $\boldsymbol{u}(t)$, then $\boldsymbol{v}(t)$ will produce a state trajectory close to the nominal trajectory. This new state trajectory is just a perturbed version of $\boldsymbol{x}(t)$ and it can be represented as

$$\boldsymbol{x}(t) + \delta\boldsymbol{x}(t).$$

The change in the state trajectory yields a corresponding change in the modified performance index. We represent this change as $\delta\tilde{J}$; it has the form,

$$\delta\tilde{J} = \Phi(\boldsymbol{x}(t_f) + \delta\boldsymbol{x}(t_f)) - \Phi(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} (H(\boldsymbol{x} + \delta\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{p}) - H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) - \boldsymbol{p}^\top \delta\dot{\boldsymbol{x}})\, dt. \tag{9.90}$$

We evaluate the above integral. We first perform an integration by parts to obtain

$$\int_{t_0}^{t_f} \boldsymbol{p}^\top \delta\dot{\boldsymbol{x}}\, dt = \boldsymbol{p}(t_f)^\top \delta\boldsymbol{x}(t_f) - \boldsymbol{p}(t_0)^\top \delta\boldsymbol{x}(t_0) - \int_{t_0}^{t_f} \dot{\boldsymbol{p}}^\top \delta\boldsymbol{x}\, dt. \tag{9.91}$$

Note that $\delta\boldsymbol{x}(t_0) = \boldsymbol{0}$ because a change in the control strategy does not change the initial state. Taking into account the above, we represent (9.90) as

$$\begin{aligned} \delta\tilde{J} &= \Phi(\boldsymbol{x}(t_f) + \delta\boldsymbol{x}(t_f)) - \Phi(\boldsymbol{x}(t_f)) - \boldsymbol{p}(t_f)^\top \delta\boldsymbol{x}(t_f) \\ &\quad + \int_{t_0}^{t_f} (H(\boldsymbol{x} + \delta\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{p}) - H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) + \dot{\boldsymbol{p}}^\top \delta\boldsymbol{x})\, dt. \end{aligned}$$

We next replace $(\Phi(\boldsymbol{x}(t_f)+\delta\boldsymbol{x}(t_f))-\Phi(\boldsymbol{x}(t_f)))$ with its first-order approximation, and add and subtract the term $H(\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})$ under the integral to obtain

$$\begin{aligned}\delta\tilde{J} &= \left(\nabla_{\boldsymbol{x}}\Phi|_{t=t_f}-\boldsymbol{p}(t_f)\right)^\top \delta\boldsymbol{x}(t_f)\\ &\quad+\int_{t_0}^{t_f}(H(\boldsymbol{x}+\delta\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})-H(\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})+H(\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})-H(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p})+\dot{\boldsymbol{p}}^\top\delta\boldsymbol{x})\,dt.\end{aligned}$$

Replacing $(H(\boldsymbol{x}+\delta\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})-H(\boldsymbol{x},\boldsymbol{v},\boldsymbol{p}))$ with its first-order approximation,

$$H(\boldsymbol{x}+\delta\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})-H(\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})=\frac{\partial H}{\partial\boldsymbol{x}}\delta\boldsymbol{x},$$

gives

$$\begin{aligned}\delta\tilde{J} &= \left(\nabla_{\boldsymbol{x}}\Phi|_{t=t_f}-\boldsymbol{p}(t_f)\right)^\top \delta\boldsymbol{x}(t_f)\\ &\quad+\int_{t_0}^{t_f}\left(\left(\frac{\partial H}{\partial\boldsymbol{x}}+\dot{\boldsymbol{p}}^\top\right)\delta\boldsymbol{x}+H(\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})-H(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p})\right)\,dt.\end{aligned}$$

Selecting in the above $\boldsymbol{p}$ as the solution to the differential equation

$$\frac{\partial H}{\partial\boldsymbol{x}}+\dot{\boldsymbol{p}}^\top=\boldsymbol{0}^\top$$

with the final condition

$$\boldsymbol{p}(t_f)=\nabla_{\boldsymbol{x}}\Phi|_{t=t_f},$$

reduces $\delta\tilde{J}$ to

$$\delta\tilde{J}=\int_{t_0}^{t_f}(H(\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})-H(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p}))\,dt.$$

If the original control $\boldsymbol{u}(t)$ is optimal, then we should have $\delta\tilde{J}\geq 0$, that is, $H(\boldsymbol{x},\boldsymbol{v},\boldsymbol{p})\geq H(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p})$. We summarize our development in the following theorem.

Theorem 9.6 *Necessary conditions for $\boldsymbol{u}\in U$ to minimize (9.85) subject to (9.86) are:*

$$\dot{\boldsymbol{p}}=-\left(\frac{\partial H}{\partial\boldsymbol{x}}\right)^\top,$$

where $H=H(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p})=F(\boldsymbol{x},\boldsymbol{u})+\boldsymbol{p}^\top\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})$,

$$H(\boldsymbol{x}^*,\boldsymbol{u}^*,\boldsymbol{p}^*)=\min_{\boldsymbol{u}\in U}H(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p}).$$

If the final state, $\boldsymbol{x}(t_f)$, is free, then in addition to the above conditions it is required that the following end-point condition is satisfied,

$$\boldsymbol{p}(t_f)=\nabla_{\boldsymbol{x}}\Phi|_{t=t_f}.$$

The equation,

$$\dot{\boldsymbol{p}} = -\left(\frac{\partial H}{\partial \boldsymbol{x}}\right)^\top \tag{9.92}$$

is called in the literature the *adjoint* or *costate* equation.

We illustrate the above theorem with the following well-known example.

Example 9.21 We consider a controlled object modeled by

$$\begin{aligned}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} &= \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u\\ &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u\end{aligned}$$

with the performance index

$$J = \int_0^{t_f} dt.$$

The control is required to satisfy

$$|u(t)| \leq 1$$

for all $t \in [0, t_f]$. This constraint means that the control must have magnitude no greater than 1. Our objective is to find admissible control that minimizes J and transfers the system from a given initial state $\boldsymbol{x}_0$ to the origin.

We begin by finding the Hamiltonian function for the problem,

$$\begin{aligned}H &= 1 + \boldsymbol{p}^\top(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u)\\ &= 1 + \begin{bmatrix}p_1 & p_2\end{bmatrix}\left(\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u\right)\\ &= 1 + p_1x_2 + p_2u.\end{aligned}$$

The costate equations are

$$\begin{bmatrix}\dot{p}_1\\ \dot{p}_2\end{bmatrix} = -\begin{bmatrix}\frac{\partial H}{\partial x_1}\\ \frac{\partial H}{\partial x_2}\end{bmatrix} = \begin{bmatrix}0\\ -p_1\end{bmatrix}.$$

Solving the costate equations yields

$$p_1 = d_1 \quad \text{and} \quad p_2 = -d_1t + d_2,$$

where d_1 and d_2 are integration constants. We now find an admissible control minimizing the Hamiltonian,

$$\begin{aligned}\arg_u \min H &= \arg_u \min(1 + p_1x_2 + p_2u)\\ &= \arg_u \min(p_2u)\\ &= \begin{cases}u(t) = 1 & \text{if } \ p_2 < 0\\ u(t) = -1 & \text{if } \ p_2 > 0\end{cases}.\end{aligned}$$

Hence,

$$u^*(t) = -\mathrm{sign}(p_2^*) = -\mathrm{sign}(-d_1 t + d_2),$$

where

$$\mathrm{sign}(z) = \begin{cases} 1 & \text{if} \quad z > 0 \\ -1 & \text{if} \quad z < 0. \end{cases}$$

Thus, the optimal control law is piecewise constant taking the values 1 or -1. This control law has at most two intervals of constancy because the argument of the sign function is a linear function, $-d_1 t + d_2$, that changes its sign at most once. This type of control is called a *bang-bang control* because it switches back and forth between its extreme values. System trajectories for $u = 1$ and $u = -1$ are families of parabolas and their equations were derived in Example 3.12. A phase portrait for $u = 1$ is shown in Figure 3.5, while for $u = -1$ in Figure 3.6. Note that only one parabola from each family passes through the specified terminal point $\boldsymbol{x} = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$ in the state plane. Segments of the two parabolas through the origin form the switching curve,

$$x_1 = -\frac{1}{2} x_2^2 \,\mathrm{sign}(x_2).$$

This means that if an initial state is above the switching curve, then $u = -1$ is used until the switching curve is reached. Then, $u = 1$ is used to reach the origin. For an initial state below the switching curve, the control $u = 1$ is used first to reach the switching curve, and then the control is switched to $u = -1$. We implement the above control action as

$$u = -\mathrm{sign}(v) = \begin{cases} 1 & \text{if} \quad v < 0 \\ -1 & \text{if} \quad v > 0, \end{cases}$$

where $v = v(x_1, x_2) = x_1 + \frac{1}{2} x_2^2 \mathrm{sign}(x_2) = 0$ is the equation describing the switching curve. We can use the above equation to synthesize a closed-loop system such that starting at an arbitrary initial state in the state plane, the trajectory will always be moving in an optimal fashion towards the origin. Once the origin is reached, the trajectory will stay there. A phase portrait of the closed-loop time optimal system is given in Figure 9.25.

Example 9.22 Consider the schematic of an armature-controlled DC motor, given in Figure 9.24. Use the minimum principle to find the armature voltage $e_a(t)$ such that the motor

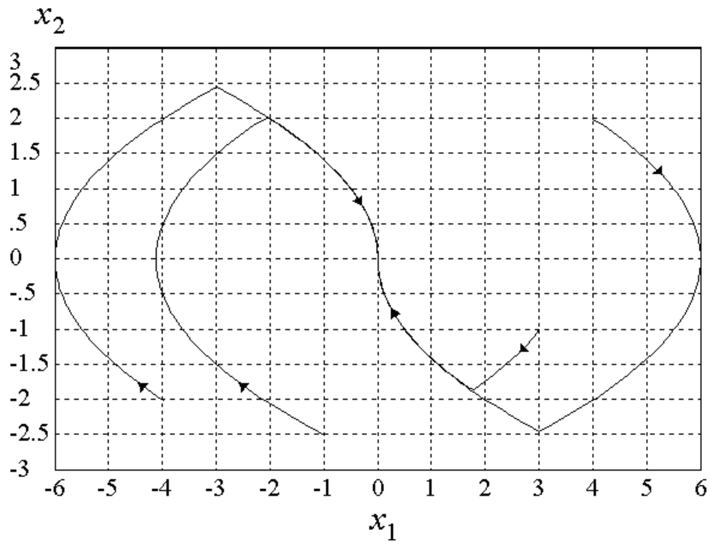


Figure 9.25: A phase-plane portrait of the time optimal closed-loop system of Example 9.21.

angular velocity ω changes from $\omega(0) = 0\,\text{rad/sec}$ at $t = 0$ to $\omega(1) = 10\,\text{rad/sec}$ while minimizing the energy dissipated in the armature resistor, R_a. There are no constraints on e_a. Note that the energy dissipated in the armature resistor is

$$J = \int_0^1 R_a i_a(t)^2 dt,$$

where i_a is the armature current.

The modeling equation of the DC motor was derived in Example 9.19. It is given by (9.82) on page 502. We represent the expression for the energy dissipated in the armature resistor in terms of ω and e_a. Applying Ohm's law to the armature circuit, we get

$$e_a - K_b\omega = R_a i_a.$$

Hence,

$$J = \int_0^1 R_a i_a(t)^2 dt = \int_0^1 \frac{(e_a - K_b\omega)^2}{R_a} dt.$$

The Hamiltonian function is

$$H = \frac{(e_a - K_b\omega)^2}{R_a} + p\left(-\frac{K_i K_b}{R_a I_{eq}}\omega + \frac{K_i}{I_{eq} R_a} e_a\right).$$

Because there are no constraints on e_a, we can find the optimal control by solving the equation

$$0 = \frac{\partial H}{\partial e_a} = \frac{2\left(e_a - K_b\omega\right)}{R_a} + \frac{K_i}{I_{eq}R_a}p.$$

Hence,

$$e_a = 2\omega - p. \tag{9.93}$$

The costate equation is

$$\dot{p} = -\frac{\partial H}{\partial \omega} = \frac{2K_b\left(e_a - K_b\omega\right)}{R_a} + \frac{K_iK_b}{R_aI_{eq}}p = 2(e_a - 2\omega) + 2p. \tag{9.94}$$

We substitute (9.93) into (9.94) to obtain

$$\dot{p} = 0.$$

Hence,

$$p(t) = \text{constant} = c.$$

Therefore,

$$e_a = 2\omega - c. \tag{9.95}$$

Substituting (9.95) into (9.82), we get

$$\dot{\omega} = -c.$$

Solving the above yields

$$\omega(t) = -ct + \omega(0) = -ct,$$

because $\omega(0) = 0$. We next use the boundary condition $\omega(1) = 10$ to find $c = -10$. Hence, the optimal armature voltage is

$$\boxed{e_a^*(t) = 20t + 10}$$

Example 9.23 In this example, we will solve the Johann Bernoulli's brachistochrone problem, introduced in Section 9.2, using the minimum principle. We first recall the problem as re-stated by Johann Bernoulli in 1697.

"Among the infinitely many curves which join the two given points ... choose the one such that, if the curve is replaced by a thin tube or groove, and a small sphere placed in it and released, then this will pass from one point to the other in the shortest time." quoted from W. Dunham [89, page 200].

It is difficult to resist not to mention about the reward that one to obtain for solving the problem:

"Let who can seize quickly the prize which we have promised to the solver. Admittedly this prize is neither of gold nor silver, for these appeal only to base and venal souls. ... Rather, since virtue itself is its own most desirable reward and a fame is a powerful incentive, we offer the prize fitting for the man of noble blood, compounded of honor, praise, and approbation..." quoted from W. Dunham [89, page 200]. We add that Johann Bernoulli wrote the above after he already knew the solution to the problem!

Model and the Performance Index

The field is conservative, hence the sums of the kinetic and potential energy is constant,

$$\frac{1}{2}mv(t)^2 - mgy(t) = \frac{1}{2}mv^2(t_0) - mgy(t_0) = 0,$$

where we assumed that the particle is at rest at t_0. Using the above, we can find the particle's speed at any time $t \geq t_0$,

$$v(t) = \sqrt{2gy(t)}.$$

Then, the brachistochrone problem can be formulated as an optimal control problem of the form,

$$\begin{array}{ll} \text{minimize} & J(t_0) = \displaystyle\int_{t_0}^{T} dt \\ \text{subject to} & \left\{ \begin{array}{lcl} \dot{x} & = & v\cos\theta \\ \dot{y} & = & v\sin\theta, \end{array} \right. \end{array}$$

where θ is the angle that the particle velocity vector forms with the positive x-axis, while $\dot{x}$ and $\dot{y}$ are the horizontal and vertical components of the particle velocity vector. The path angle θ is considered as the control input to be determined.

Particle Trajectory

To proceed, we form the Hamiltonian function,

$$H = 1 + p_1 v\cos\theta + p_2 v\sin\theta, \tag{9.96}$$

where p_1 and p_2 are the costate variables. The costate equations are:

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = -\begin{bmatrix} \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{g}{v}(p_1\cos\theta + p_2\sin\theta) \end{bmatrix}.$$

It follows from the above that $p_1 = K =$ constant. Because there are no constraints on the control input θ, the necessary condition for the control optimality is

$$\frac{\partial H}{\partial \theta} = 0,$$

where

$$\frac{\partial H}{\partial \theta} = -p_1 v \sin\theta + p_2 v \cos\theta.$$

Hence,

$$p_2 = p_1 \tan\theta = K \tan\theta.$$

We next find an expression for $\dot{\theta}$. First note that

$$\dot{p}_2 = \frac{\partial p_2}{\partial \theta}\dot{\theta} = \frac{\partial (K\tan\theta)}{\partial \theta}\dot{\theta} = \frac{K}{\cos^2\theta}\dot{\theta}. \tag{9.97}$$

On the other hand,

$$\begin{aligned}
\dot{p}_2 &= -\frac{g}{v}(p_1\cos\theta + p_2\sin\theta)\\
&= -\frac{g}{v}(K\cos\theta + K\tan\theta\sin\theta)\\
&= -\frac{Kg}{v}(\cos\theta + \frac{\sin^2\theta}{\cos\theta})\\
&= -\frac{Kg}{v}\frac{\cos^2\theta + \sin^2\theta}{\cos\theta})\\
&= -\frac{Kg}{v\cos\theta}.
\end{aligned}\tag{9.98}$$

Combining (9.97) and (9.98) yields

$$\dot{\theta} = -\frac{g}{v}\cos\theta. \tag{9.99}$$

We now use the boundary conditions to express x and y as functions of time and θ. First, using the above we can write

$$\dot{y} = \frac{dy}{d\theta}\dot{\theta} = \frac{dy}{d\theta}\left(-\frac{g}{v}\cos\theta\right).$$

On the other hand,

$$\dot{y} = v\sin\theta.$$

Equating the right-hand sides of the above two equations yields

$$\frac{dy}{v^2} = \frac{dy}{2gy} = -\frac{1}{g}\tan\theta d\theta.$$

Integrating both sides gives

$$\frac{1}{2g}\int \frac{dy}{y} = -\frac{1}{g}\int \tan\theta d\theta,$$

that is,

$$\frac{1}{2g}\ln|y| = \frac{1}{g}\ln|\cos\theta|.$$

Performing manipulations, we obtain

$$y = C\cos^2\theta,$$

where C is an integration constant. Evaluating the above at $t = T$ gives

$$y(T) = y_1 = C\cos^2\theta(T).$$

Hence,

$$C = \frac{y_1}{\cos^2\theta(T)}$$

and

$$\boxed{y(t) = \frac{y_1}{\cos^2\theta(T)}\cos^2\theta(t)} \tag{9.100}$$

In a similar way, we can obtain an expression for x as a function of time and θ. Indeed, on the one hand,

$$\dot{x} = \frac{dx}{d\theta}\dot{\theta} = -\frac{dx}{d\theta}\frac{g}{v}\cos\theta,$$

on the other hand,

$$\dot{x} = v\cos\theta.$$

Comparing the right-hand sides of the above two equations gives

$$\frac{dx}{d\theta} = -\frac{v^2}{g} = -2y = -2\frac{y_1}{\cos^2\theta(T)}\cos\theta.$$

We represent the above as

$$dx = -2y = -2\frac{y_1}{\cos^2\theta(T)}\cos\theta d\theta.$$

Integrating both sides, using the identity, $\cos^2\theta = \frac{1}{2}(1+\cos 2\theta)$, yields

$$x = -2\frac{y_1}{\cos^2\theta(T)}\left(\frac{\theta}{2} + \frac{\sin 2\theta}{4}\right) + \tilde{C},$$

where $\tilde{C}$ is the integration constant. Evaluating the above at $t = T$ gives

$$x_1 = -2\frac{y_1}{\cos^2\theta(T)}\left(\frac{\theta(T)}{2} + \frac{\sin 2\theta(T)}{4}\right) + \tilde{C}.$$

Hence,

$$\tilde{C} = x_1 + \frac{y_1}{\cos^2\theta(T)}\left(\theta(T) + \frac{\sin 2\theta(T)}{2}\right).$$

Using the above, we obtain

$$\boxed{x(t) = x_1 + \frac{y_1}{2\cos^2\theta(T)}\left(2(\theta(T) - \theta(t)) + \sin 2\theta(T) - \sin 2\theta(t)\right)} \tag{9.101}$$

We point out that the equation for $x(t)$, $y(t)$ and $\dot{\theta}$ specify an implicit feedback control law in the sense that the control θ can be calculated given the state $(x(t), y(t))$.

We use the boundary conditions to find the description of the time-optimal closed loop system. We assume, without loss of generality, that $x(t_0) = y(t_0) = x(0) = y(0) = 0$. Then, using the expression for $y(t)$ we note that $y(t_0) = 0 = \frac{y_1}{\cos^2\theta(T)}\cos^2\theta(t_0)$. Thus, the angle of departure is $\theta(t_0) = 90°$. With the above initial path angle, y, and hence v, is increased at the outset as quickly as possible. Using this data, we next evaluate x at $t = t_0$ to form an equation that can be used to find the angle of arrival $\theta(T)$. The equation is

$$2x_1\cos^2\theta(T) + y_1\left(2\theta(T) - \pi + \sin 2\theta(T)\right) = 0.$$

The above equation can be solved easily using MATLAB's Symbolic Toolbox or the MATLAB's `fsolve` function. Once $\theta(T)$ is found, we use the expression for y to find $\cos\theta$, where

$$\cos\theta(t) = \sqrt{\frac{y(t)}{y_1}}\cos\theta(T).$$

Hence,

$$\sin\theta(t) = \sqrt{1 - \cos^2\theta(t)}.$$

We substitute the above expression into $\dot{x} = v\cos\theta$ and $\dot{y} = v\sin\theta$ to obtain the state-space model of the time optimal motion of the particle moving under the force of gravity from a given point to another specified point. The equations are,

$$\begin{cases} \dot{x}(t) &= \sqrt{2gy(t)}\sqrt{\frac{y(t)}{y_1}}\cos\theta(T) \\ \dot{y}(t) &= \sqrt{2gy(t)}\sqrt{1 - \frac{y(t)}{y_1}\cos^2\theta(T)} \end{cases}$$

subject to the boundary conditions $\begin{bmatrix} x(t_0) & y(t_0) \end{bmatrix} = \begin{bmatrix} x_0 & y_0 \end{bmatrix}$ and $\begin{bmatrix} x(T) & y(T) \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \end{bmatrix}$.

The Time of Quickest Descent

We derive an expression that can be used to compute the time of the particle travel. We first establish an auxiliary result that we will use in our derivation. Specifically, we will show that $\dot{\theta}(t)$ is constant on the interval $[t_0, T]$. Differentiating both sides of equation (9.99) gives

$$\begin{aligned}
\frac{d\dot{\theta}}{dt} &= \frac{d}{dt}\left(-\frac{g}{\sqrt{2gy}}\cos\theta\right)\\
&= -\frac{g}{\sqrt{2g}}\left(\frac{d}{dt}\left(\frac{1}{\sqrt{y}}\cos\theta\right)\right)\\
&= -\frac{g}{\sqrt{2g}}\left(-\frac{1}{2}\frac{\dot{y}\cos\theta}{y^{3/2}}-\frac{\sin\theta}{\sqrt{y}}\dot{\theta}\right)\\
&= \frac{g}{\sqrt{2g}}\left(\frac{1}{2}\frac{\dot{y}\cos\theta}{y^{3/2}}+\frac{\sin\theta}{\sqrt{y}}\dot{\theta}\right).
\end{aligned}$$

Substituting into the above the expressions for $\dot{\theta}$ and $\dot{y}$ gives

$$\begin{aligned}
\frac{d\dot{\theta}}{dt} &= \frac{g}{\sqrt{2g}}\left(\frac{1}{2}\frac{\sqrt{2gy}\sin\theta\cos\theta}{y^{3/2}}-g\frac{\sin\theta\cos\theta}{\sqrt{y}\sqrt{2gy}}\right)\\
&= \frac{g}{\sqrt{2g}}\left(\frac{\sqrt{g}}{y\sqrt{2}}\sin\theta\cos\theta-\frac{\sqrt{g}}{y\sqrt{2}}\sin\theta\cos\theta\right)\\
&= 0.
\end{aligned}$$

Because $\dot{\theta}$ is constant on the interval $[t_0, T]$, we use (9.99) to write

$$\frac{d\theta}{dt} = -\frac{g}{\sqrt{2gy(T)}}\cos\theta(T).$$

Separating the variables yields

$$d\theta = -\frac{g}{\sqrt{2gy(T)}}\cos\theta(T)dt.$$

Integrating both sides of the above, we obtain

$$\int_{\theta(t)}^{\theta(T)} d\theta = -\frac{g}{\sqrt{2gy(T)}}\cos\theta(T)\int_{t_0}^{T} dt.$$

Hence,

$$\theta(T)-\theta(t) = -\frac{g}{\sqrt{2gy(T)}}\cos\theta(T)(T-t_0).$$

To obtain an expression for the particle travel time, we set in the above equation $t = t_0 = 0$ and perform simple manipulations to obtain

$$\boxed{T = \frac{\pi/2-\theta(T)}{\cos\theta(T)}\sqrt{2y(T)g}}$$

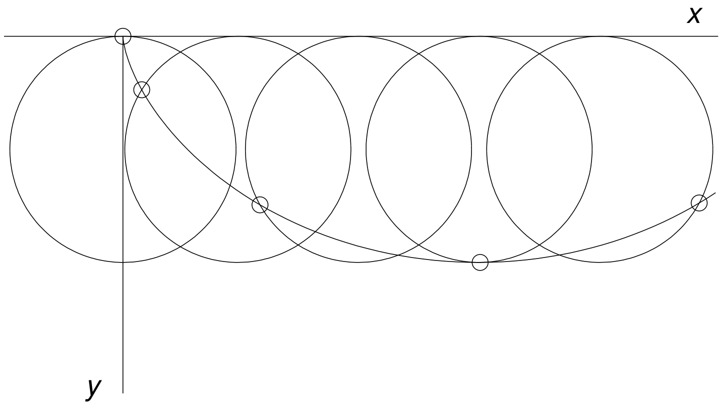


Figure 9.26: Constructing an upside-down cycloid—the solution to the brachistochrone problem.

It's an Upside-Down Cycloid!

The path of quickest descent is an upside-down cycloid. A cycloid is a curve described by a point on the circumference of a circle that rolls without slipping—see Figure 9.26. The equations describing a cycloid are

$$\left.\begin{array}{rcl} x(\phi) & = & R(\phi - \sin\phi) \\ y(\phi) & = & R(1 - \cos\phi) \end{array}\right\}$$

We will now show that equations (9.101) and (9.100) describe a cycloid. To proceed, we define

$$R = \frac{y_1}{2\cos^2\theta(T)} \quad \text{and} \quad \phi(t) = \pi - 2\theta(t). \tag{9.102}$$

Using the above notation, and taking into account that $\sin(\pi-\phi) = \sin\phi$, we re-write (9.101) as

$$x(t) - x_1 + R(\phi(T) - \sin\phi(T)) = R(\phi(t) - \sin\phi(t)).$$

Using the notation given in (9.102) and the fact that $\cos 2\phi = 2\cos^2\phi - 1$, we represent (9.100) as

$$\begin{array}{rcl} y(t) & = & \dfrac{y_1}{\cos^2\theta(T)}\cos^2\theta(t) \\ & = & \dfrac{y_1}{\cos^2\theta(T)}\dfrac{\cos 2\theta(t) + 1}{2} \\ & = & R(1 + \cos(\pi - \phi(t)) \\ & = & R(1 - \cos\phi(t)), \end{array}$$

because $\cos(\pi - \phi(t)) = -\cos\phi$.

In the following, we discuss a general minimum time regulator problem for a class of multi-input dynamical systems modeled by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}.$$

The goal is to transfer the system from some initial state to the origin in minimum time subject to

$$|u_i| \le k_i, \quad k_i > 0, \quad i = 1, 2, \ldots, m.$$

Thus, the performance index for the problem is $J = \int_{t_0}^{t_f} dt$. We begin our analysis by forming the associated Hamiltonian function,

$$\begin{aligned} H &= 1 + \boldsymbol{p}^\top(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}) \\ &= 1 + \boldsymbol{p}^\top \boldsymbol{A}\boldsymbol{x} + \sum_{i=1}^{m} \left(\boldsymbol{p}^\top \boldsymbol{b}_i\right) u_i, \end{aligned}$$

where $\boldsymbol{b}_i$ is the i-th column of $\boldsymbol{B}$. Let

$$s_i(t) = \boldsymbol{p}^{*\top}(t)\boldsymbol{b}_i.$$

Then, applying Pontryagin's minimum principle we obtain the following form of the optimal control,

$$u_i^*(t) = -k_i \operatorname{sign}(s_i(t)), \quad i = 1, 2, \ldots, m.$$

Thus, the optimal control for the i-th control channel has the bang-bang form, $u_i^* = \pm k_i$. Note that $s_i(t) = \boldsymbol{p}^{*\top}(t)\boldsymbol{b}_i = 0$ is the switching curve for the i-th control channel. We compute the costate vector from the costate equation,

$$\begin{aligned} \dot{\boldsymbol{p}}^{*\top} &= -\frac{\partial H}{\partial \boldsymbol{x}} \\ &= -\frac{\partial}{\partial \boldsymbol{x}} \left(\boldsymbol{p}^{*\top} \boldsymbol{A}\boldsymbol{x}\right) \\ &= -\boldsymbol{p}^{*\top} \boldsymbol{A}, \end{aligned}$$

or

$$\dot{\boldsymbol{p}}^* = -\boldsymbol{A}^\top \boldsymbol{p}^*.$$

The solution to the costate equation is

$$\boldsymbol{p}^*(t) = e^{-\boldsymbol{A}^\top t} \boldsymbol{p}^*(0).$$

Thus, the i-th switching curve can be represented as

$$s_i(t) = \boldsymbol{p}^*(0)^\top e^{-\boldsymbol{A}t}\boldsymbol{b}_i = 0.$$

Observe that if $s_i(t) = 0$ on some time interval, then $u_i^*(t)$ is indefinite on this interval. We show that under some mild assumptions there is no time interval on which $s_i(t) = 0$, that is, there is no time interval on which $u_i^*(t)$ is indefinite. First, we assume that $\boldsymbol{b}_i \neq \boldsymbol{0}$. Note that in our problem here, $\min_{\boldsymbol{u}\in U} H(\boldsymbol{x}, \boldsymbol{u}) = 0$ because $J^*(t, \boldsymbol{x}) = J^*(\boldsymbol{x})$ and therefore $\frac{\partial J^*}{\partial t} = 0$. Hence, $1 + \boldsymbol{p}^*(t)^\top(\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{B}\boldsymbol{u}^*) = 0$ for all t for the optimal process. Thus, the optimal costate vector $\boldsymbol{p}^*(t)$ cannot be zero for any value of t. Finally, if $s_i = \boldsymbol{p}^{*\top}\boldsymbol{b}_i$ is zero on some time interval, that is, s_i is constant on a time interval, then this implies that on the above time interval,

$$\dot{s}_i = \dot{\boldsymbol{p}}^{*\top}\boldsymbol{b}_i = -\boldsymbol{p}^{*\top}\boldsymbol{A}\boldsymbol{b}_i = 0.$$

Similarly, higher derivatives of $s_i(t) = \boldsymbol{p}^{*\top}\boldsymbol{b}_i$ will also be zero on the time interval on which $s_i(t) = \boldsymbol{p}^{*\top}\boldsymbol{b}_i$ is zero, that is, $s_i^{(j)}(t) = (-1)^j\boldsymbol{p}^{*\top}\boldsymbol{A}^j\boldsymbol{b}_i = 0$, $j = 1, 2, \ldots$. We represent the above relations as

$$\boldsymbol{p}^{*\top}\begin{bmatrix} \boldsymbol{b}_i & \boldsymbol{A}\boldsymbol{b}_i & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b}_i & \cdots \end{bmatrix} = \boldsymbol{0}.$$

If the system is controllable by the i-th input acting alone, then

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{b}_i & \boldsymbol{A}\boldsymbol{b}_i & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b}_i & \cdots \end{bmatrix} = n,$$

and we would have to have $\boldsymbol{p}^* = \boldsymbol{p}^*(t) = \boldsymbol{0}$. However, we already ruled out this case and thus $s_i(t) = \boldsymbol{p}^{*\top}\boldsymbol{b}_i$ cannot be zero on any time interval. In conclusion, if the system is controllable by each input acting alone, then there is no time interval on which optimal control vector is indefinite.

9.5.2 A Two-Point Boundary-Value Problem

We consider the problem of minimizing the quadratic performance index

$$J = \frac{1}{2}\boldsymbol{x}(t_f)^\top\boldsymbol{F}\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f}\left(\boldsymbol{x}(t)^\top\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}(t)^\top\boldsymbol{R}\boldsymbol{u}(t)\right)dt \tag{9.103}$$

subject to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad \boldsymbol{x}(t_f) \quad \text{is free.} \tag{9.104}$$

We assume that there are no constraints on the control input $\boldsymbol{u}$. The weight matrices $\boldsymbol{F}$ and $\boldsymbol{Q}$ are real symmetric positive semi-definite, and the matrix $\boldsymbol{R}$ is real symmetric positive definite. The Hamiltonian function for the above linear quadratic control problem is

$$H = \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \frac{1}{2}\boldsymbol{u}^\top\boldsymbol{R}\boldsymbol{u} + \boldsymbol{p}^\top(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}), \tag{9.105}$$

where $\boldsymbol{p} \in \mathbb{R}^n$ is the costate vector. It follows from Theorem 9.6 that the optimal controller must minimize the Hamiltonian function. Because the control vector is unconstrained, the necessary condition for optimality of the control $\boldsymbol{u}$ is

$$\frac{\partial H}{\partial \boldsymbol{u}} = \boldsymbol{0}. \tag{9.106}$$

Evaluating (9.106) yields

$$\frac{\partial H}{\partial \boldsymbol{u}} = \boldsymbol{u}^\top \boldsymbol{R} + \boldsymbol{p}^\top \boldsymbol{B} = \boldsymbol{0}. \tag{9.107}$$

Hence, the optimal controller has the form

$$\boldsymbol{u} = -\boldsymbol{R}^{-1} \boldsymbol{B}^\top \boldsymbol{p}. \tag{9.108}$$

Substituting the above into (9.104), we obtain

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{p}, \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{9.109}$$

We next find the costate equation,

$$\begin{aligned} \dot{\boldsymbol{p}} &= -\left(\frac{\partial H}{\partial \boldsymbol{x}}\right)^\top \\ &= -\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{p}. \end{aligned} \tag{9.110}$$

Combining (9.109) and the above yields $2n$ linear differential equations

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \\ -\boldsymbol{Q} & -\boldsymbol{A}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{p} \end{bmatrix}. \tag{9.111}$$

By Theorem 9.6, the state vector at time t_f must satisfy

$$\boldsymbol{p}(t_f) = \frac{1}{2}\left(\nabla \boldsymbol{x}^\top \boldsymbol{F} \boldsymbol{x}\right)\Big|_{t=t_f} = \boldsymbol{F}\boldsymbol{x}(t_f). \tag{9.112}$$

In summary, we reduced the linear quadratic control problem to solving $2n$ linear differential equations with mixed boundary conditions, which is an example of a *two-point boundary value problem*, or TPBVP for short. We solve the above TPBVP. If we had the initial conditions $\boldsymbol{x}(t_0)$ and $\boldsymbol{p}(t_0)$, then the solution to (9.111) would have the form

$$\begin{bmatrix} \boldsymbol{x}(t_f) \\ \boldsymbol{p}(t_f) \end{bmatrix} = e^{\boldsymbol{H}(t_f - t_0)} \begin{bmatrix} \boldsymbol{x}(t_0) \\ \boldsymbol{p}(t_0) \end{bmatrix}, \tag{9.113}$$

where

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \\ -\boldsymbol{Q} & -\boldsymbol{A}^\top \end{bmatrix}.$$

However, in this particular TPBVP, $\boldsymbol{p}(t_0)$ is unknown and instead we are given $\boldsymbol{p}(t_f)$. To proceed, let

$$\begin{bmatrix} \boldsymbol{x}(t_f) \\ \boldsymbol{p}(t_f) \end{bmatrix} = e^{\boldsymbol{H}(t_f-t)} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{p}(t) \end{bmatrix}, \tag{9.114}$$

where

$$e^{\boldsymbol{H}(t_f-t)} = \begin{bmatrix} \boldsymbol{\Phi}_{11}(t_f,t) & \boldsymbol{\Phi}_{12}(t_f,t) \\ \boldsymbol{\Phi}_{21}(t_f,t) & \boldsymbol{\Phi}_{22}(t_f,t) \end{bmatrix}.$$

Each of the blocks $\boldsymbol{\Phi}_{ij}(t_f,t)$, $i,j=1,2$, is $n \times n$. Thus we have

$$\begin{bmatrix} \boldsymbol{x}(t_f) \\ \boldsymbol{p}(t_f) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{11}(t_f,t) & \boldsymbol{\Phi}_{12}(t_f,t) \\ \boldsymbol{\Phi}_{21}(t_f,t) & \boldsymbol{\Phi}_{22}(t_f,t) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{p}(t) \end{bmatrix}. \tag{9.115}$$

The above matrix equation can be represented as

$$\boldsymbol{x}(t_f) = \boldsymbol{\Phi}_{11}(t_f,t)\boldsymbol{x}(t) + \boldsymbol{\Phi}_{12}(t_f,t)\boldsymbol{p}(t), \tag{9.116}$$

$$\boldsymbol{p}(t_f) = \boldsymbol{\Phi}_{21}(t_f,t)\boldsymbol{x}(t) + \boldsymbol{\Phi}_{22}(t_f,t)\boldsymbol{p}(t). \tag{9.117}$$

Using the boundary condition (9.112) and (9.116) gives

$$\boldsymbol{p}(t_f) = \boldsymbol{F}\boldsymbol{x}(t_f) = \boldsymbol{F}\boldsymbol{\Phi}_{11}(t_f,t)\boldsymbol{x}(t) + \boldsymbol{F}\boldsymbol{\Phi}_{12}(t_f,t)\boldsymbol{p}(t). \tag{9.118}$$

Subtracting (9.117) from (9.118), we obtain

$$\boldsymbol{0} = \left(\boldsymbol{F}\boldsymbol{\Phi}_{11}(t_f,t) - \boldsymbol{\Phi}_{21}(t_f,t)\right)\boldsymbol{x}(t) + \left(\boldsymbol{F}\boldsymbol{\Phi}_{12}(t_f,t) - \boldsymbol{\Phi}_{22}(t_f,t)\right)\boldsymbol{p}(t). \tag{9.119}$$

Using the above, we compute

$$\begin{aligned} \boldsymbol{p}(t) &= \left(\boldsymbol{\Phi}_{22}(t_f,t) - \boldsymbol{F}\boldsymbol{\Phi}_{12}(t_f,t)\right)^{-1}\left(\boldsymbol{F}\boldsymbol{\Phi}_{11}(t_f,t) - \boldsymbol{\Phi}_{21}(t_f,t)\right)\boldsymbol{x}(t) \\ &= \boldsymbol{P}(t)\boldsymbol{x}(t), \end{aligned} \tag{9.120}$$

where,

$$\boldsymbol{P}(t) = \left(\boldsymbol{\Phi}_{22}(t_f,t) - \boldsymbol{F}\boldsymbol{\Phi}_{12}(t_f,t)\right)^{-1}\left(\boldsymbol{F}\boldsymbol{\Phi}_{11}(t_f,t) - \boldsymbol{\Phi}_{21}(t_f,t)\right). \tag{9.121}$$

Substituting (9.120) into (9.108) yields the optimal state-feedback controller

$$\boldsymbol{u}(t) = -\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P}(t)\boldsymbol{x}(t). \tag{9.122}$$

We show that $\boldsymbol{P}(t)$ satisfies a matrix differential equation. Indeed, differentiating (9.120) gives

$$\dot{\boldsymbol{P}}\boldsymbol{x} + \boldsymbol{P}\dot{\boldsymbol{x}} - \dot{\boldsymbol{p}} = \boldsymbol{0}. \tag{9.123}$$

Substituting into the above (9.109), (9.110), and (9.120) yields

$$\left(\dot{\boldsymbol{P}} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P} + \boldsymbol{Q} + \boldsymbol{A}^\top\boldsymbol{P}\right)\boldsymbol{x} = \boldsymbol{0}. \tag{9.124}$$

The preceding must hold throughout $t_0 \le t \le t_f$. Hence, $\boldsymbol{P}$ must satisfy

$$\boxed{\dot{\boldsymbol{P}} = \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\boldsymbol{P} - \boldsymbol{A}^\top\boldsymbol{P} - \boldsymbol{P}\boldsymbol{A} - \boldsymbol{Q}} \tag{9.125}$$

subject to the boundary condition

$$\boldsymbol{P}(t_f) = \boldsymbol{F}. \tag{9.126}$$

Equation (9.125) is called the matrix *Riccati differential equation*. Note that because $\boldsymbol{F}$ is symmetric, so is $\boldsymbol{P} = \boldsymbol{P}(t)$. Observe that $\boldsymbol{P} = \boldsymbol{P}(t)$ given by (9.121) is a solution to (9.125) subject to the boundary condition (9.126).

The Riccati differential equation (9.125) associated with a class of one-dimensional plants was solved in Example 9.18.

9.6 Notes

Sussmann and Willems [315] made the following statement, in 1997, regarding the origins of optimal control: "Optimal control was born in 1697—300 years ago—in Groningen, a university town in the north of The Netherlands, when Johann Bernoulli, professor of mathematics at the local university from 1695 to 1705, published his solution of the *brachystochrone problem*."

For a serious student of optimal control, the seminal text by Lee and Markus [205] is recommended. Less advanced than Lee and Markus are Kirk [181], Bryson and Ho [45], Owens [260], and Pierre [275]. The classical text by Kwakernaak and Sivan [199] is also worth perusing because of its clarity and its treatment of the stochastic aspects of control problems. Informative, and at the same time entertaining, accounts of the origins of the calculus of variations are given by Dunham [89, Chapter 8], and Sussmann and Willems [315]. Elsgolc [96] provides a lucid introduction, illustrated with many examples, to the calculus of variations. Applications of the calculus of variations in physics and engineering are provided by Weinstock [347]. Interrelations between the calculus of variations and optimal control are discussed by Hestenes [136]. Frequency-domain aspects of optimal control and robust optimal control design is discussed by Burl in [48]. Our presentation of the continuous minimum time regulator problem in Subsection 9.4.3, was based on Boltyanskii [38]. For more numerical examples related to optimal control, see Lewis and Syrmos [210], and Bryson [46]. Exciting applications of optimal control to a number of control problems of technology and science can be found in Lee and Markus [205] .

9.7 Exercises

Exercise 9.1 Find the variations of the functionals:

(i) $v(x_1, x_2) = \int_0^{\pi/2} \left(x_1^4 + 2x_1^2x_2^2 + x_2^2\right) dt$,

(ii) $v(x_1, x_2) = \int_0^1 \left(e^{x_1 - x_2} - e^{x_1 + x_2}\right) dt$.

Exercise 9.2 Find the equations of the curves that are extremals of the functional

$$v(x) = \int_0^{t_f} \left(\frac{1}{4}\dot{x}^2 - t\dot{x} - x + \frac{1}{2}x\dot{x}\right) dt$$

for the boundary conditions

(i) $x(0) = 1$, $t_f = 2$, and $x(2) = 10$;

(ii) $x(0) = 1$, $t_f = 10$, and $x(10)$ is free.

Exercise 9.3 Find the equation of the curve that is an extremal of the functional

$$v(x) = \int_0^{t_f} \left(\frac{1}{2}\dot{x}^2 - t\dot{x} + 2x^2\right) dt$$

subject to the boundary conditions

$$x(0) = 3/4,\ t_f = 1, \text{ and } x(1) \text{ is free.}$$

Exercise 9.4 Given a dynamical system model

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 2 & 0 \end{bmatrix} \boldsymbol{x}, \end{aligned}$$

and the associated performance index

$$J = \int_0^{\infty} \left(y^2 + u^2\right) dt.$$

Assuming that $u = -\boldsymbol{k}\boldsymbol{x}$ is such that J is minimized, find the poles of the closed-loop system.

Exercise 9.5 Given a dynamical system model

$$\begin{aligned}\dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 2 & 0 \end{bmatrix}\boldsymbol{x},\end{aligned}$$

and the performance index

$$J = \int_0^\infty \left(y^2 + u^2\right) dt.$$

Use the Kronecker product to represent the associated algebraic Riccati equation as the nonlinear vector algebraic equation. Solve it, and construct the optimal state feedback control law $u = -\boldsymbol{k}\boldsymbol{x}$. Assume that

$$\boldsymbol{x}(0) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Find the optimal value of J.

Exercise 9.6 Find a state feedback control law u so as to minimize

$$J = \int_0^1 \left(x^2 + u^2\right) dt$$

subject to

$$\dot{x} = -x + u, \quad x(0) = 2.$$

Assume $J^* = p(t)x^2$.

Exercise 9.7 Use the principle of optimality to find the set of numbers $x_1, x_2, \ldots, x_n$ satisfying $x_1 + x_2 + \cdots + x_n = c$, where c is a positive constant such that the product $x_1 x_2 \cdots x_n$ is maximized.

Exercise 9.8 Use dynamic programming to find the maximum cost path between node **a** and node **t** in the routing network depicted in Figure 9.27, where the travel costs are shown beside each segment and toll charges are shown by each node. The network can only be traversed from left to right.

Exercise 9.9 Use dynamic programming to find $u[0]$ and $u[1]$ that minimize

$$J = (x[2] - 2)^2 + \sum_{k=0}^{1} u[k]^2$$

subject to

$$x[k+1] = x[k] + u[k], \quad x[0] = 1.$$

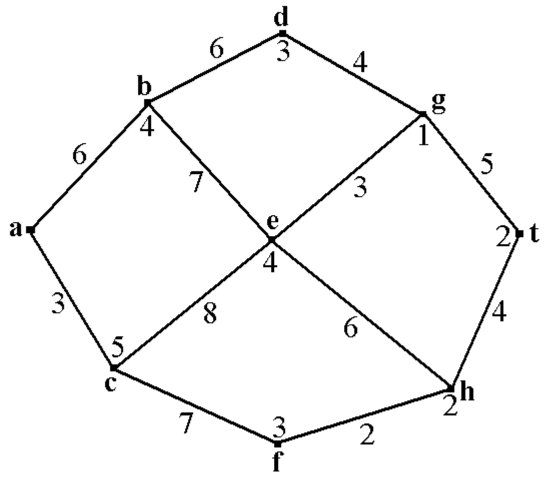


Figure 9.27: A routing network of Exercise 9.8.

Exercise 9.10 Consider the following optimal control problem:

$$\begin{array}{ll} \text{minimize} & J_2(u) = \dfrac{1}{2}\displaystyle\sum_{k=0}^{1} u[k]^2 \\ \text{subject to} & x[k+1] = x[k] + 2u[k], \quad x[0] = 3, \quad x[2] = 0. \end{array}$$

Convert the above optimal control problem into a quadratic optimization problem with an equality constraint using the composite input vector

$$\boldsymbol{z} = \begin{bmatrix} u[0] \\ u[1] \end{bmatrix}.$$

Then, employ Lagrange's multiplier theorem to find the optimal sequence $\{u^*[0], u^*[1]\}$.

Exercise 9.11 Consider the following optimal control problem:

$$\begin{array}{ll} \text{minimize} & J_2(u) = \dfrac{1}{2}x[2]^2 + \dfrac{1}{2}\displaystyle\sum_{k=0}^{1} \left(x[k]^2 + u[k]^2\right) \\ \text{subject to} & x[k+1] = x[k] + 2u[k], \quad x[0] = 3. \end{array}$$

Convert the above optimal control problem into a quadratic optimization problem with an equality constraint using the composite input-state vector

$$\boldsymbol{z} = \begin{bmatrix} u[0] \\ x[1] \\ u[1] \\ x[2] \end{bmatrix}.$$

Then, employ Lagrange's multiplier theorem to find the optimal sequence $\{u^*[0], u^*[1]\}$.

Exercise 9.12 Find $u = u(t)$ so as to minimize

$$J = \int_0^1 u^2 dt$$

subject to

$$\dot{x} = -2x + u, \quad x(0) = 10, \quad x(1) = 0.$$

Exercise 9.13 For the circuit shown in Figure 9.28, find the input voltage, $u(t)$, that changes the current in the inductor form $i = 10$ A at $t = 0$ to $i = 0$ at $t = 1$ sec while minimizing the performance index,

$$J = \int_0^1 u(t)^2 dt.$$

Exercise 9.14 Use the minimum principle to find the input voltage $u(t)$ that charges the capacitor, in Figure 9.29, from $x(0) = 2\,V$ at $t = 0$ to $x(10) = 12\,V$ while minimizing the energy dissipated in the resistor. There are no constraints on $u(t)$. Note that, by Kirchhoff's voltage law,

$$Ri(t) + x(t) = u(t),$$

where $i(t) = C\frac{dx(t)}{dt}$, and the energy dissipated is

$$J = \int_0^{10} Ri(t)^2 dt.$$

Exercise 9.15 Show that if μ is an eigenvalue of the Hamiltonian matrix $\boldsymbol{H}$, then so is $-\mu$.

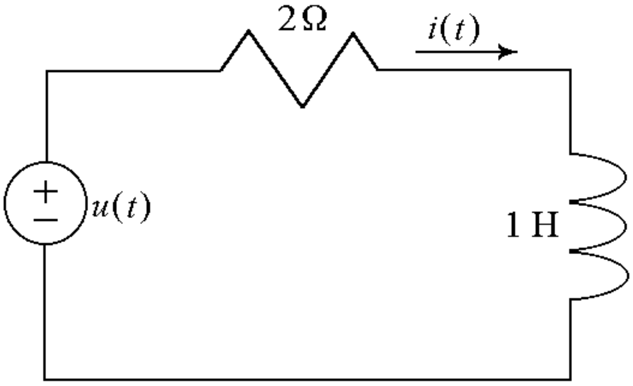


Figure 9.28: A circuit for Exercise 9.13.

Exercise 9.16 Consider the Hamiltonian function defined by

$$H = F + \boldsymbol{p}^\top \boldsymbol{f}.$$

Show that if the functions $\boldsymbol{f}$ and F do not explicitly depend upon t, then

$$H|_{\boldsymbol{u}=\boldsymbol{u}^*} = \text{constant}, \quad t_0 \le t \le t_f.$$

Exercise 9.17 Given the Hamiltonian matrix corresponding to a dynamical system and its associated performance index. The Hamiltonian matrix is

$$\boldsymbol{H} = \begin{bmatrix} 2 & -5 \\ -1 & -2 \end{bmatrix}.$$

The MATLAB command `[V,D] = eig(H)` resulted in the following data:

$$\boldsymbol{V} = \begin{bmatrix} 0.9806 & 0.7071 \\ -0.1961 & 0.7071 \end{bmatrix}, \qquad \boldsymbol{D} = \begin{bmatrix} 3 & 0 \\ 0 & -3 \end{bmatrix}.$$

(i) Write the equation of the closed loop system driven by the optimal controller $u = -kx$.

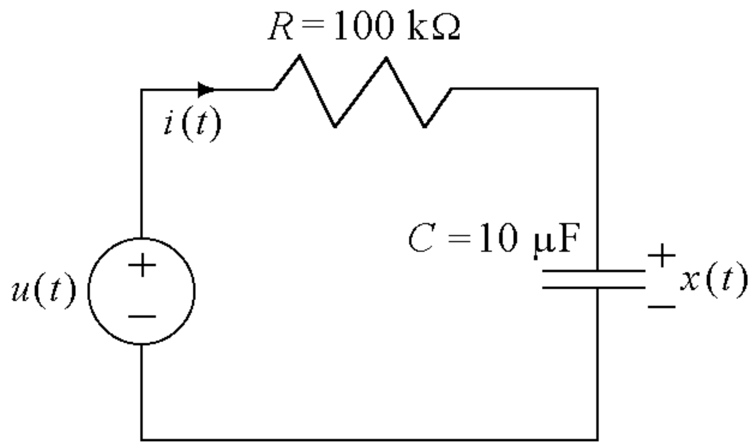


Figure 9.29: A capacitor charging circuit of Exercise 9.14.

(ii) Find the solution to the algebraic Riccati equation corresponding to the optimal controller.

Exercise 9.18 **(i)** Suppose that s_i is an eigenvalue of the matrix

$$\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P},$$

and $\boldsymbol{v}_i \neq \boldsymbol{0}$ is the corresponding eigenvector, where $\boldsymbol{P}$ is the symmetric positive definite solution of the algebraic Riccati equation. Show that

$$\begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{P}\boldsymbol{v}_i \end{bmatrix}$$

is an eigenvector of the associated Hamiltonian matrix corresponding to the eigenvalue s_i.

(ii) Show that if $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, then the real parts of the eigenvalues of the matrix

$$\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{P}$$

are all negative, where $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ is the solution of the associated algebraic Riccati equation.

Exercise 9.19 Given the following model of a dynamical system:

$$\begin{aligned}\dot{x}_1 &= x_2 + 5\\ \dot{x}_2 &= u,\end{aligned}$$

where

$$|u| \leq 1.$$

The performance index to be minimized is

$$J = \int_0^{t_f} dt.$$

Find the state-feedback control law $u = u(x_1, x_2)$ that minimizes J and drives the system from a given initial condition $\boldsymbol{x}(0) = [x_1(0), x_2(0)]^\top$ to the final state $\boldsymbol{x}(t_f) = \mathbf{0}$. Proceed as indicated below.

(a) Derive the equations of the optimal trajectories.

(b) Derive the equation of the switching curve.

(c) Write the expression for the optimal state-feedback controller.

Exercise 9.20 (based on [289]) Given a dynamical system model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u,$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0\\ 0 & 0 & 1 & \cdots & 0 & 0\\ \vdots & & & \cdots & & \vdots\\ 0 & 0 & 0 & \cdots & 0 & 1\\ -a_0 & -a_1 & -a_2 & \cdots & a_{n-2} & a_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0\\ 0\\ \vdots\\ 0\\ 1 \end{bmatrix},$$

and the associated performance index

$$J = \frac{1}{2}\int_0^\infty \left(\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + u^2\right) dt, \quad \boldsymbol{Q} = \boldsymbol{Q}^\top \geq 0.$$

We label the above problem as $(\boldsymbol{A}, \boldsymbol{b})$. Consider now the dynamical system model

$$\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{b}}\tilde{u},$$

where $\boldsymbol{b} = \tilde{\boldsymbol{b}}$,

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & \cdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix},$$

and the associated performance index has the form

$$\tilde{J} = \frac{1}{2}\int_0^\infty \left(\tilde{\boldsymbol{x}}^\top \tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}} + \tilde{u}^2\right) dt,$$

where $\tilde{\boldsymbol{Q}} = \boldsymbol{Q} + \boldsymbol{D}\tilde{\boldsymbol{A}} + \tilde{\boldsymbol{A}}^\top \boldsymbol{D} + \boldsymbol{D}\boldsymbol{b}\boldsymbol{b}^\top \boldsymbol{D}$, and

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & -a_0 \\ 0 & 0 & 1 & \cdots & 0 & -a_1 \\ \vdots & & & \cdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -a_{n-2} \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix}.$$

We refer to the second problem as $\left(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}}\right)$.

(a) Show that the symmetric matrix $\boldsymbol{P}$ is a solution to the $(\boldsymbol{A}, \boldsymbol{b})$ problem if, and only if, the symmetric matrix $\tilde{\boldsymbol{P}} = \boldsymbol{P} - \boldsymbol{D}$ is a solution to the $\left(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}}\right)$ problem.

(b) Show that $\boldsymbol{A} - \boldsymbol{b}\boldsymbol{b}^\top \boldsymbol{P} = \tilde{\boldsymbol{A}} - \boldsymbol{b}\boldsymbol{b}^\top \tilde{\boldsymbol{P}}$.

Hint: Notice that $\tilde{\boldsymbol{A}} = \boldsymbol{A} - \boldsymbol{b}\boldsymbol{b}^\top \boldsymbol{D}$, and $\tilde{\boldsymbol{b}} = \boldsymbol{b}$. Write down the algebraic Riccati equation for the $\left(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}}\right)$ problem and perform appropriate substitutions.

Exercise 9.21 Consider the following variation of the brachistochrone problem: A particle of mass m is to slide down a frictionless track connecting two given points that are not located on the vertical line. The particle's initial speed is v_0. The motion of the particle is constrained only by gravity and the shape of the track. Find the curve of the track that minimizes the time travel between the two given points. Solve this brachistochrone problem using the minimum principle of Pontryagin. Illustrate your derivations with a numerical example and simulations using MATLAB.

Exercise 9.22 **(a)** Consider the ground vehicle model described in Subsection 2.4.2. Let the input signal be $u = \delta_f$—the front wheel steer angle, and $\delta_r = 0$. Write a MATLAB script that animates the lateral motion Y versus the longitudinal motion X of the vehicle, the steer angle δ_f, and the heading angle ψ. Use the steer angle shown in Figure 9.30.

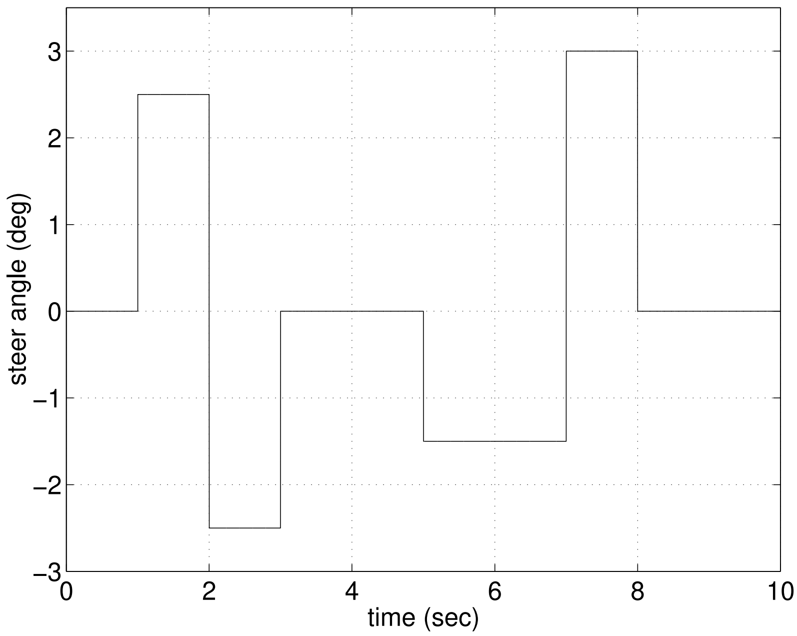


Figure 9.30: Plot of the steer angle δ_f versus time for Exercise 9.22.

(b) Use the linear model and the data from Table 2.1 to design a linear quadratic regulator (LQR). In this part, the control has the form

$$\delta_f = -\boldsymbol{k}\boldsymbol{x} + Y_{ref},$$

where Y_{ref} is the command input shown in Figure 9.31. That is, the closed-loop system here has the form,

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k})\boldsymbol{x} + \boldsymbol{b}Y_{ref}.$$

You as a designer, will select the weights $\boldsymbol{Q}$ and r. See how the linear vehicle tracks the command input. Observe the sign difference in the command input and the heading angle. Your LQR gain should be calculated for each chosen v_x. In other words, if in your interactive simulation, a specific value of v_x is chosen, then your program should calculate the corresponding gain for this particular value of v_x. Implement the LQR state-feedback employing the vehicle model that uses the nonlinear cornering

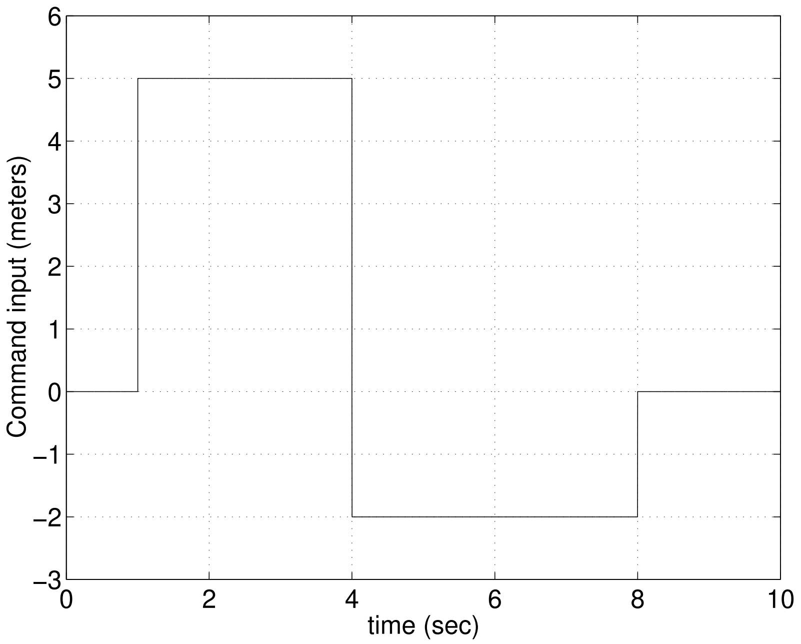


Figure 9.31: Plot of the command input Y_{ref} versus time for Exercise 9.22.

characteristics shown in Figure 2.10 and test the performance of the closed-loop system utilizing the command input shown in Figure 9.31. You may need to impose constraints on δ_f, for example, $|\delta_f| \leq 20°$. (Do not forget about converting degrees into radians.) Write a MATLAB script that animates the lateral motion Y versus the longitudinal motion X of the vehicle, the command input, and the heading angle.

Exercise 9.23 For the one-link manipulator from Exercise 5.22, use the linearized model from Part **(b)** of that exercise to

(a) design the optimal controller that minimizes the performance index

$$J = \frac{1}{2}\Delta\boldsymbol{x}(t_f)^\top \boldsymbol{F}\Delta\boldsymbol{x}(t_f) + \frac{1}{2}\int_0^{t_f} \left(\Delta\boldsymbol{x}(t)^\top \boldsymbol{Q}\Delta\boldsymbol{x}(t) + \Delta u(t)^2\right) dt,$$

where $t_f = 2$ sec, $\boldsymbol{F} = 3\boldsymbol{I}_3$, and $\boldsymbol{Q} = 2\boldsymbol{I}_3$. (Hint: Use the method presented in Subsection 9.5.2);

(b) Implement and perform animation of the closed-loop system comprised of the optimal state-feedback controller driving the nonlinear model of the robot;

(c) Implement and perform animation of the closed-loop system comprised of the combined optimal controller-observer compensator driving the nonlinear model of the robot.

Chapter 10

Sliding Modes

> As philosopher of science Karl Popper has emphasized, a good theory is characterized by the fact that it makes a number of predictions that could in principle be disproved or falsified by observation. Each time new experiments are observed to agree with the predictions the theory survives, and our confidence in it is increased; but if ever a new observation is found to disagree, we have to abandon or modify the theory. At least that is what is supposed to happen, but you can always question the competence of the person who carried out the observation. [133, page 130]

10.1 Simple Variable Structure Systems

A *variable structure system* is a dynamical system whose structure changes in accordance with the current value of its state. A variable structure system can be viewed as a system composed of independent structures together with a switching logic between each of the structures. With appropriate switching logic, a variable structure system can exploit the desirable properties of each of the structures the system is composed of. Even more, a variable structure system may have a property that is not a property of any of its structures. We illustrate the above ideas with two numerical examples.

Example 10.1 This example was adapted from Utkin [331]. We consider a dynamical system model

$$\left\{ \begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & -ux_1, \end{array} \right. \tag{10.1}$$

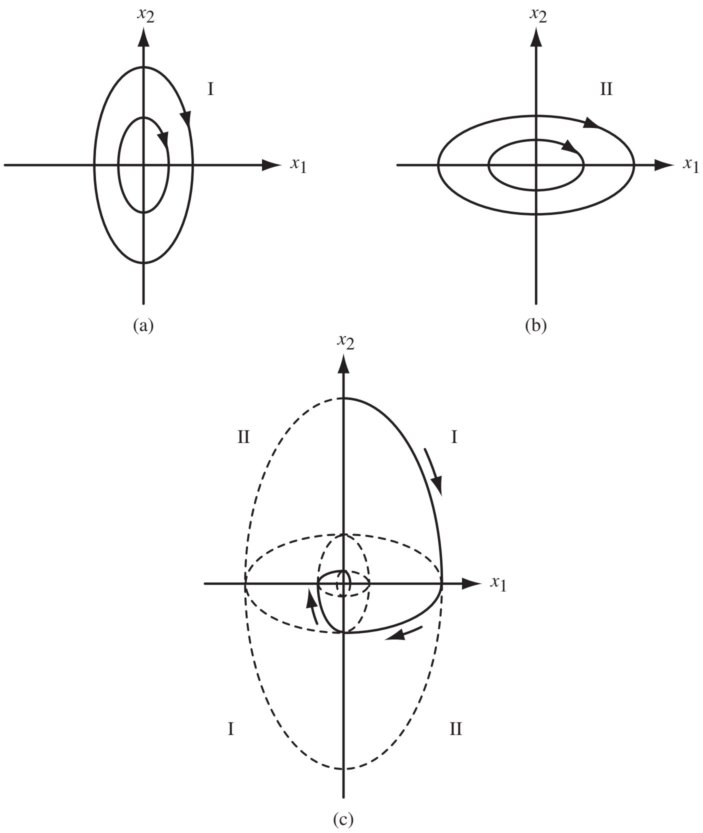


Figure 10.1: Figures (a) and (b) are phase-plane portraits of structures making up a variable structure system and Figure (c) is a phase-plane portrait of the variable structure system itself, where the structure of the system is changed any time the system's trajectory crosses either of the coordinate axis. This figure was adapted from the survey paper by Utkin [331].

having two structures corresponding to $u = 1/a$ and $u = a$, where a is a positive constant greater than 1; in our simulations we take $a = 5$. The phase-plane portraits of the structures are shown in Figures 10.1 (a) and (b). The phase-plane portraits of the individual structures are families of ellipses. Neither of the structures is asymptotically stable—they are only stable. However, by choosing a suitable switching logic between the structures, we can make the resulting variable structure system asymptotically stable. Indeed, suppose the structure of the system is changed any time the system's trajectory crosses either of the coordinate axes of the state plane, that is,

$$u = \begin{cases} 1/5 & \text{if} \quad x_1 x_2 < 0 \\ 5 & \text{if} \quad x_1 x_2 > 0. \end{cases} \tag{10.2}$$

A phase plane portrait of the system (10.1) with the switching logic specified by (10.2) is

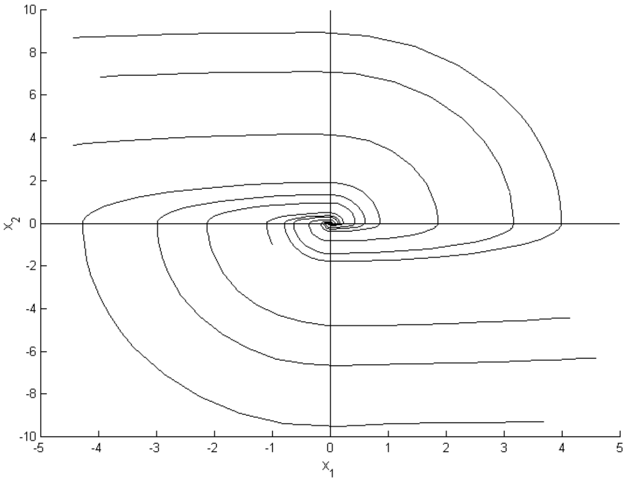


Figure 10.2: Phase-plane portrait of the variable structure system described by (10.1) and (10.2).

shown in Figure 10.2. We add that the trajectories spiral clockwise towards the origin.

Example 10.2 Consider the double integrator

$$\ddot{x} = u,$$

whose state-space model is

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u. \end{cases} \tag{10.3}$$

Let $u = \pm 2$. Typical trajectories of the above system are shown in Figure 10.3. We have two independent structures, neither of them being stable. Let us use the following switching logic between each of the structures:

$$u = -2\,\mathrm{sign}\,(x_1 + x_2)\,. \tag{10.4}$$

With the above switching logic, the closed-loop system changes its structure anytime the state trajectory crosses the line

$$x_1 + x_2 = 0.$$

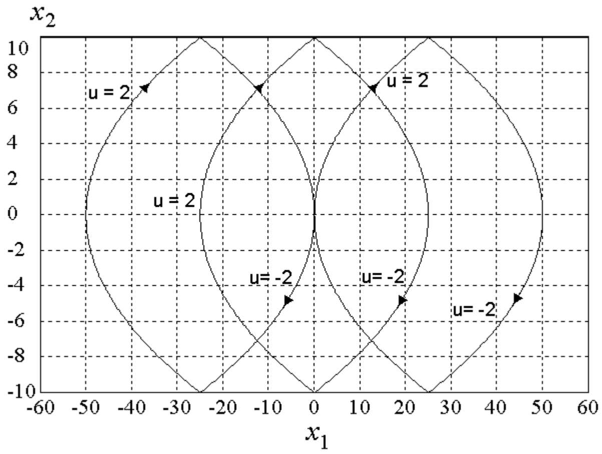


Figure 10.3: Typical trajectories of the double integrator system.

We introduce the notation,

$$\sigma(\boldsymbol{x}) = \sigma(x_1, x_2) = x_1 + x_2.$$

Then, we can describe the switching logic action as

$$u = \begin{cases} -2 & \text{if} \quad \sigma(\boldsymbol{x}) > 0 \\ 2 & \text{if} \quad \sigma(\boldsymbol{x}) < 0. \end{cases} \tag{10.5}$$

Typical trajectories of the closed-loop system (10.3), (10.4) for the initial conditions located "close" to the origin of the state-plane are shown in Figure 10.4. Close to the origin, once the trajectories intercept the switching line they are confined to it thereafter. If we increase the integration step size, then we can observe oscillations, or *chattering*, of the trajectory about the switching surface as it is explained in Figure 10.5. A plot of two trajectories with an increased integration time is shown in Figure 10.6.

10.2 Sliding Mode

The motion of the system while confined to the switching line or a surface is referred to as *sliding*. A sliding mode will exist if in the vicinity of the switching surface the state velocity vectors are directed towards the surface. In such a case, the switching surface attracts trajectories when they are in its vicinity and once a trajectory intersects the switching surface it will stay on it thereafter.

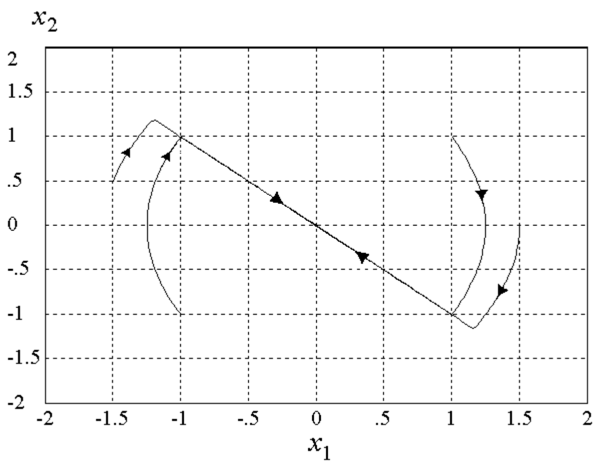


Figure 10.4: Typical trajectories of the closed-system (10.3), (10.4) for initial conditions "close" to the origin of the state-plane. The applied control law is given by (10.4).

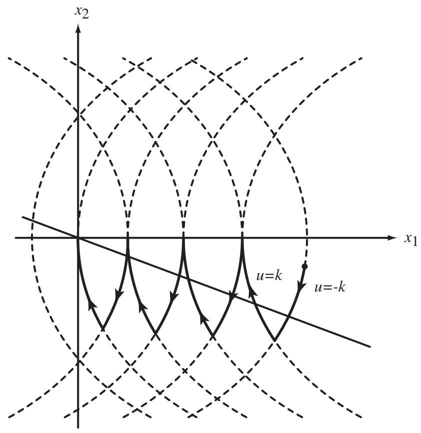


Figure 10.5: Chattering in a variable structure sliding mode system—a magnified view.

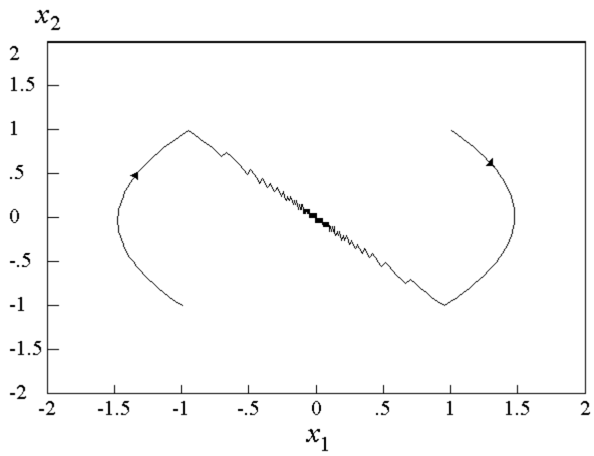


Figure 10.6: An illustration of chattering in a variable structure sliding mode system consisting of the double integrator and the control law, $u = -\text{sign}(x_1 + x_2)$. Two trajectories starting from two different initial conditions are plotted.

A surface $\sigma(\boldsymbol{x}) = 0$ is attractive if

1. any trajectory starting on the surface remains there, and
2. any trajectory starting outside the surface tends to it at least asymptotically.

Thus, for a sliding motion to occur we need

$$\lim_{\sigma \to 0^+} \dot{\sigma} < 0 \quad \text{and} \quad \lim_{\sigma \to 0^-} \dot{\sigma} > 0.$$

The above conditions are illustrated in Figure 10.7. They ensure that the motion of the state trajectory $\boldsymbol{x}$ on either side of the switching surface $\sigma(\boldsymbol{x}) = 0$ is towards the switching surface. The two conditions may be combined to give

$$\sigma\dot{\sigma} < 0 \tag{10.6}$$

in the neighborhood of the switching surface. Note, however, that the closed-loop system is now modeled by differential equations with discontinuous right-hand sides. The modeling differential equations are discontinuous on the switching surface $\sigma(\boldsymbol{x}) = 0$. This is the reason we often refer to the switching surface as the *discontinuity surface*. A rigorous analysis of such systems cannot be carried out using methods of classical theory of differential equations.

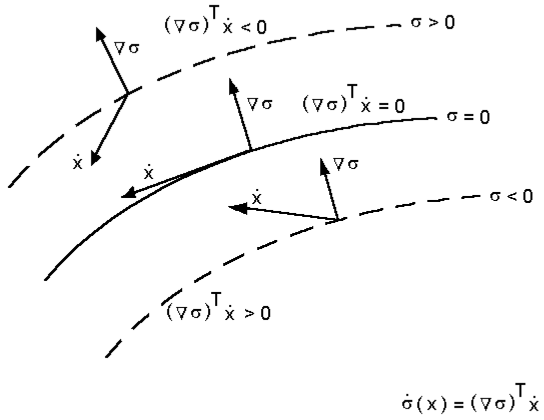


Figure 10.7: An illustration of the sufficient condition of the attractivity of the switching surface.

A solution to the differential equation $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$ with a continuous right-hand side is a function $\boldsymbol{x}(t)$ that has a derivative and satisfies the differential equation everywhere on a given interval. In the case when the right-hand side of $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$ is discontinuous, we have to use more general methods such as one, for example, developed by Filippov [103]. We illustrate these points with a numerical example taken from Filippov [103, page 1].

Example 10.3 Consider the following differential equation with the discontinuous right-hand side,

$$\dot{x} = 1 - 2\text{sign}(x). \tag{10.7}$$

For $x < 0$, we have $\dot{x} = 3$, while for $x > 0$, $\dot{x} = -1$. In Figure 10.8, plots of x versus time for different initial conditions are given. As t increases, the solution x of (10.7) reaches the line $x = 0$ and stays there. The function $x(t)$, however, does not satisfy (10.7) in the usual sense because if $\dot{x}(t) = 0$ and $x(t) = 0$, the right hand side of (10.7) becomes

$$1 - 2\text{sign}(0) = 1 \neq 0,$$

where we used the usual definition of the sign function, where $\text{sign}(0) = 0$. To circumvent the above difficulties, the defintion of the sign function is generalized by defininig its value at zero as a point-to-set mapping, that is,

$$\text{sign}(0) \in [-1,\ 1]$$

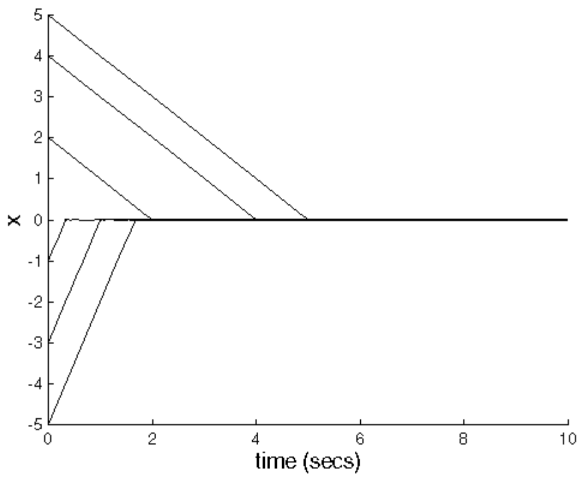


Figure 10.8: Plots of x versus time for different initial conditions for the differential equation (10.7).

and as before

$$\operatorname{sign}(z) = \begin{cases} 1 & \text{if } z > 0 \\ -1 & \text{if } z < 0. \end{cases}$$

Filippov [103, pages 2-3] recommends that the generalized solution to an ordinary differential equation with a discontinuous right-hand side should meet the following criteria:

(i) for a differential equation with a continuous right-hand side, the generalized solution should be equivalent with the usual one;

(ii) for the equation $\dot{x} = f(t)$, the generalized solution should be of the form

$$x(t) = \int f(t)dt + \text{constant};$$

(iii) for any initial condition $x(t_0) = x_0$, the solution must exist at least for $t > t_0$ and should be continuable;

(iv) the generalized solution should be applicable to a broad class of equations that model dynamical systems;

(v) the limit of an uniformly convergent sequence of solutions should be a solution;

(vi) after applying to a solution a common change of variables, the result of the transformation should be a solution.

We next describe the proposed generalized solution in the sense of Filippov. We begin with an example. Suppose that we are given a single-input dynamical system model, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u)$ along with a switching surface $\sigma(\boldsymbol{x}) = 0$. Let $\boldsymbol{f}^- = \boldsymbol{f}(\boldsymbol{x}, u^-)$ and $\boldsymbol{f}^+ = \boldsymbol{f}(\boldsymbol{x}, u^+)$, where $u = u^-$ if $\sigma < 0$ and $u = u^+$ if $\sigma > 0$. We join the velocity vectors $\boldsymbol{f}^-$ and $\boldsymbol{f}^+$. The resulting velocity vector $\boldsymbol{f}^0$ is obtained by drawing the line tangent to the switching surface that intersects the line segment joining the velocity vectors $\boldsymbol{f}^-$ and $\boldsymbol{f}^+$. Thus,

$$\boldsymbol{f}^0 = \alpha \boldsymbol{f}^- + (1 - \alpha)\boldsymbol{f}^+,$$

where $\alpha \in [0, 1]$. The above construction is illustrated in Figure 10.9. Observe that $\boldsymbol{f}^0$ belongs to the smallest convex set containing $\boldsymbol{f}^-$ and $\boldsymbol{f}^+$. Recall that a set $\mathcal{S}$ is said to be convex if for every $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$ and every real number α, $0 < \alpha < 1$, the point $\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y} \in \mathcal{S}$.

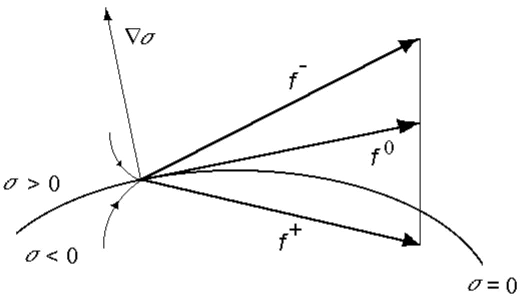


Figure 10.9: Constructing a solution in the sense of Filippov in $\mathbb{R}^2$.

Following Itkis [159], we give a simple interpretation of the above construction. The number α may be interpreted as the time the system's trajectory remains below the switching surface, while $(1-\alpha)$ may be interpreted as the time the trajectory remains above the switching surface during the trajectory oscillation period about the switching surface. Specifically, if the trajectory performs one oscillation about $\sigma = 0$ in a "small" time interval $\Delta\tau$, then it will traverse a distance $\|\boldsymbol{f}^-\|\alpha\Delta\tau$ in time $\alpha\Delta\tau$ with velocity $\boldsymbol{f}^-$. During the remaining time of oscillation the trajectory will traverse a distance $\|\boldsymbol{f}^+\|(1-\alpha)\Delta\tau$ in time $(1-\alpha)\Delta\tau$ with velocity $\boldsymbol{f}^+$. The average displacement of the trajectory during the time $\Delta\tau$ will then be $\|\alpha\boldsymbol{f}^- + (1-\alpha)\boldsymbol{f}^+\|\Delta\tau$ with the average velocity $\boldsymbol{f}^0$.

Suppose that there are m discontinuity surfaces of the form

$$\sigma_i = \{\boldsymbol{x} : \sigma_i(\boldsymbol{x}) = 0\}, \quad i = 1, 2, \ldots, m.$$

Let

$$H = \bigcup_{i=1}^{m} \sigma_i.$$

Then, we take a solution of the differential equation $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ with the discontinuous right-hand side to be a solution of the differential inclusion

$$\dot{\boldsymbol{x}}(t) \in G(\boldsymbol{x}(t)),$$

where for each $\boldsymbol{x}$, the set $G(\boldsymbol{x})$ is the smallest closed convex set containing the cluster values, limits, of the function $\boldsymbol{f}(\boldsymbol{y})$ as $\boldsymbol{y} \to \boldsymbol{x}$, $\boldsymbol{y} \notin H$, that is, a solution of the dynamical system

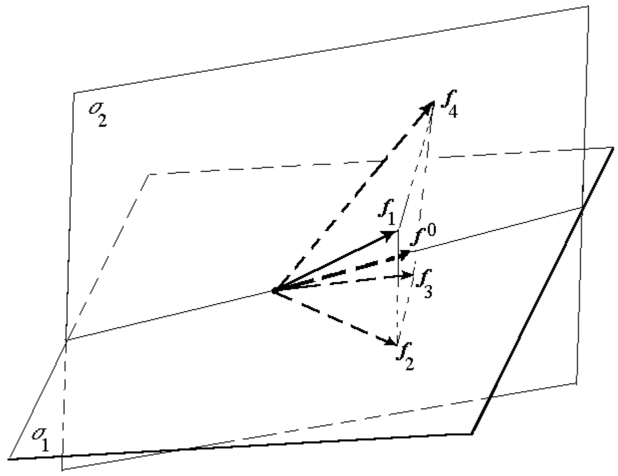


Figure 10.10: Constructing a solution in the sense of Filippov in $\mathbb{R}^3$.

$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ is an absolutely continuous function $\boldsymbol{x}(t)$, in the sense of Definition A.2, defined on an interval, or segment, L for which $\dot{\boldsymbol{x}}(t) \in G(\boldsymbol{x}(t))$ almost everywhere on L (Filippov [103]). Observe that if the function $\boldsymbol{f}$ is continuous at a point $\boldsymbol{x}$, that is, $\boldsymbol{x} \notin H$, then $G(\boldsymbol{x})$ consists of a single point, namely $\boldsymbol{f}(\boldsymbol{x})$, and the solution satisfies $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ in the usual sense. However, if $\boldsymbol{x} \in H$, then $\boldsymbol{x}$ lies on one or more of the surfaces $\sigma_1, \sigma_2, \ldots, \sigma_m$. Suppose that $\boldsymbol{x}$ lies on the surfaces $\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k}$. It then follows that for a sufficiently small $\delta > 0$ the surfaces $\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k}$ partition the δ-neighborhood of $\boldsymbol{x}$ into $\ell \leq 2^k$ regions $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_\ell$, in each of which the function $\boldsymbol{f}$ is continuous. Let $\boldsymbol{f}_j$ denote the function $\boldsymbol{f}$ restricted to the region $\mathcal{R}_j$. Then, $G(\boldsymbol{x})$ is the smallest closed convex set containing the set of points

$$\left\{ \lim_{\substack{\tilde{\boldsymbol{x}} \to \boldsymbol{x} \\ \tilde{\boldsymbol{x}} \in \mathcal{R}_j}} \boldsymbol{f}_j(\tilde{\boldsymbol{x}}) : j = 1, 2, \ldots, \ell \right\}. \tag{10.8}$$

The velocity vector, $\dot{\boldsymbol{x}} = \boldsymbol{f}^0$, lies then on the intersection of the tangent plane to the surfaces of discontinuity and the convex set described by (10.8). This is illustrated in Figures 10.9 and 10.10.

10.3 A Simple Sliding Mode Controller

In this section, we discuss the problem of designing sliding mode controllers for single-input plants. The design of a variable structure sliding mode controller consists of two phases.

They are:

- sliding (switching) surface design so as to achieve the desired system behavior, like the stability to the origin, when restricted to the surface, and
- selecting feedback gains of the controller so that the closed-system is stable to the sliding surface.

There is a number of benefits when the system is in sliding. In particular,

- while in sliding the system is not affected by matched uncertainties, and
- while in sliding the system behavior is governed by a reduced set of differential equations. This is what we refer to as the *order reduction*. It is very useful in designing variable structure sliding mode controllers.

We discuss a sliding surface design phase for linear single-input plants. Without loss of generality, we assume that the plant model is already in its controller companion form,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u.$$

Let the sliding surface be

$$\begin{aligned} \sigma(\boldsymbol{x}) &= \boldsymbol{s}\boldsymbol{x} \\ &= \begin{bmatrix} s_1 & s_2 & \cdots & s_{n-1} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \\ &= s_1 x_1 + s_2 x_2 + \cdots + s_{n-1} x_{n-1} + x_n \\ &= 0. \end{aligned}$$

Note that, without loss of generality, we assumed that $s_n = 1$. If this was not the case, we could divide both sides of $\boldsymbol{s}\boldsymbol{x} = 0$ by $s_n \neq 0$ to ensure that $s_n = 1$. Using the above equation of sliding surface, we can write

$$x_n = -s_1 x_1 - s_2 x_2 - \cdots - s_{n-1} x_{n-1}.$$

We combine the equations of the plant model and that of the sliding surface. First note that

$$\dot{x}_{n-1} = x_n.$$

Hence, we can write

$$\left.\begin{array}{rcl}\dot{x}_1 & = & x_2\\ \dot{x}_2 & = & x_3\\ & \vdots & \\ \dot{x}_{n-1} & = & -s_1x_1 - s_2x_2 - \cdots - s_{n-1}x_{n-1},\end{array}\right\} \tag{10.9}$$

or equivalently in matrix form,

$$\dot{\boldsymbol{z}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & & & & \vdots\\ 0 & 0 & 0 & \cdots & 1\\ -s_1 & -s_2 & -s_3 & \cdots & -s_{n-1}\end{bmatrix}\boldsymbol{z}, \tag{10.10}$$

where $\boldsymbol{z} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{n-1}\end{bmatrix}^\top$. The above equation describes the system dynamics in sliding. Observe the order reduction of the system dynamics in sliding. We now have $n - m = n - 1$ first-order differential equations. The poles of the reduced-order system are the roots of the equation

$$\lambda^{n-1} + s_{n-1}\lambda^{n-2} + \cdots + s_2\lambda + s_1 = 0.$$

The response of the system in sliding is completely specified by an appropriate choice of the components $s_1, s_2, \ldots, s_{n-1}$ of the switching surface.

Consider again Example 10.2. The sliding surface for this example can be represented as

$$\sigma(\boldsymbol{x}) = \begin{bmatrix} s_1 & 1\end{bmatrix}\boldsymbol{x} = 0,$$

or equivalently

$$s_1x_1 + x_2 = 0.$$

The dynamics of the system in sliding are described by

$$\dot{z} = -s_1 z,$$

which is just a first-order system. Note that to ensure the reduced-system stability, we select $s_1 > 0$.

After designing a sliding surface, we construct a feedback controller. In this section, we discuss a controller design for linear single-input plants. The controller objective is to drive the plant state to the sliding surface and maintain it on the surface for all subsequent time. We use a generalized Lyapunov approach in constructing the controller. Specifically, we use a distance, $V = 0.5\sigma^2$, from the sliding surface σ as a Lyapunov function candidate. Then, we select the controllers' gains so that the time derivative of the chosen Lyapunov function candidate evaluated on the solution of the controlled system is negative definite with respect to the switching surface, thus ensuring the motion of the state trajectory to the surface as it is illustrated in Figure 10.7. Our goal is to find $u = u(\boldsymbol{x})$ so that

$$\frac{d}{dt}\left(\frac{1}{2}\sigma^2\right) = \sigma\dot{\sigma} < 0.$$

We consider the controller structure of the form:

$$u = k_1^{\pm} x_1 + k_2^{\pm} x_2 + \cdots + k_n^{\pm} x_n, \tag{10.11}$$

where $k_i^{\pm}$ are the feedback gains to be determined so that the condition $\sigma\dot{\sigma} < 0$ is satisfied. To determine the feedback gains, we substitute u, given by (10.11), into the expression $\sigma\dot{\sigma}$ and utilize (10.9) to get

$$\begin{aligned}
\sigma\dot{\sigma} &= \sigma \begin{bmatrix} s_1 & s_2 & \cdots & 1 \end{bmatrix} \dot{\boldsymbol{x}} \\
&= \sigma\left(s_1\dot{x}_1 + \cdots + s_{n-1}\dot{x}_{n-1} + \dot{x}_n\right) \\
&= \sigma\left(s_1 x_2 + \cdots + s_{n-1}x_n + \dot{x}_n\right) \\
&= \sigma(s_1 x_2 + \cdots + s_{n-1}x_n - a_0 x_1 - \cdots - a_{n-1}x_n + u) \\
&= \sigma(s_1 x_2 + \cdots + s_{n-1}x_n - a_0 x_1 - \cdots - a_{n-1}x_n + k_1^{\pm}x_1 + \cdots + k_n^{\pm}x_n).
\end{aligned}$$

Rearranging the terms yields

$$\sigma\dot{\sigma} = \sigma x_1\left(-a_0 + k_1^{\pm}\right) + \sigma x_2\left(-a_1 + s_1 + k_2^{\pm}\right) + \cdots + \sigma x_n\left(-a_{n-1} + s_{n-1} + k_n^{\pm}\right).$$

A possible way to ensure that $\sigma\dot{\sigma} < 0$ is to select the gains to force each term in $\sigma\dot{\sigma}$ to be negative, that is,

$$\left.\begin{aligned}
\sigma x_1\left(-a_0 + k_1^{\pm}\right) &< 0 \\
\sigma x_2\left(-a_1 + s_1 + k_2^{\pm}\right) &< 0 \\
&\vdots \\
\sigma x_n\left(-a_{n-1} + s_{n-1} + k_n^{\pm}\right) &< 0.
\end{aligned}\right\}$$

A possible choice of gains that leads to satisfaction of the attractivity to the surface condition, $\sigma\dot{\sigma} < 0$, is:

$$\left.\begin{array}{lll} k_1^+ & < & -|a_0| \quad \text{if} \quad \sigma x_1 > 0 \\ k_1^- & > & |a_0| \quad \text{if} \quad \sigma x_1 < 0 \\ & \vdots & \\ k_n^+ & < & -(|a_{n-1}| + |s_{n-1}|) \quad \text{if} \quad \sigma x_n > 0 \\ k_n^- & > & |a_{n-1}| + |s_{n-1}| \quad \text{if} \quad \sigma x_n < 0. \end{array}\right\} \tag{10.12}$$

We can also use the following controller gains:

$$\left.\begin{array}{lll} k_1^\pm & = & -(|a_0| + \varepsilon)\operatorname{sign}(x_1\sigma) \\ & \vdots & \\ k_n^\pm & = & -(|a_{n-1}| + |s_{n-1}| + \varepsilon)\operatorname{sign}(x_n\sigma), \end{array}\right\} \tag{10.13}$$

where $\varepsilon > 0$ is a design parameter. Note that the controllers' gains given by (10.13) satisfy conditions (10.12). The larger the value of ε the faster the trajectory converges to the sliding surface. In Figure 10.11, we summarize the above design method in the form of a flow chart.

10.4 Sliding in Multi-Input Systems

In this section, we analyze sliding mode behvior in linear multi-input plants. When the number of the plant inputs is m, we can use sliding surface of the form

$$\left\{\boldsymbol{x} : \boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{S} \in \mathbb{R}^{m\times n}\right\}.$$

We also use $\boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}$ to denote the above sliding surface. In many applications nonlinear sliding surfaces arise. Here, we use linear sliding surfaces only, although one can use nonlinear sliding surfaces as well. Let $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{S}\boldsymbol{x}$. Then,

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \begin{bmatrix} \sigma_1(\boldsymbol{x}) \\ \sigma_2(\boldsymbol{x}) \\ \vdots \\ \sigma_m(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_1\boldsymbol{x} \\ \boldsymbol{s}_2\boldsymbol{x} \\ \vdots \\ \boldsymbol{s}_m\boldsymbol{x} \end{bmatrix},$$

where $\boldsymbol{s}_i \in \mathbb{R}^{1\times n}$. Recall, that the system is in sliding mode if $\boldsymbol{\sigma}(\boldsymbol{x}(t)) = \boldsymbol{0}$ and $\dot{\boldsymbol{\sigma}}(\boldsymbol{x}(t)) = \boldsymbol{0}$ for $t \geq t_0$, where $\boldsymbol{x}$ is the state trajectory and t_0 is a specified time. It follows that in a sliding mode the velocity vector $\dot{\boldsymbol{x}}(t)$ is tangent to the switching surface. Equivalently,

$$(\nabla\sigma_i)^\top \dot{\boldsymbol{x}}(t) = \boldsymbol{s}_i\dot{\boldsymbol{x}}(t) = 0, \quad \text{for} \quad i = 1, 2, \ldots, m,$$

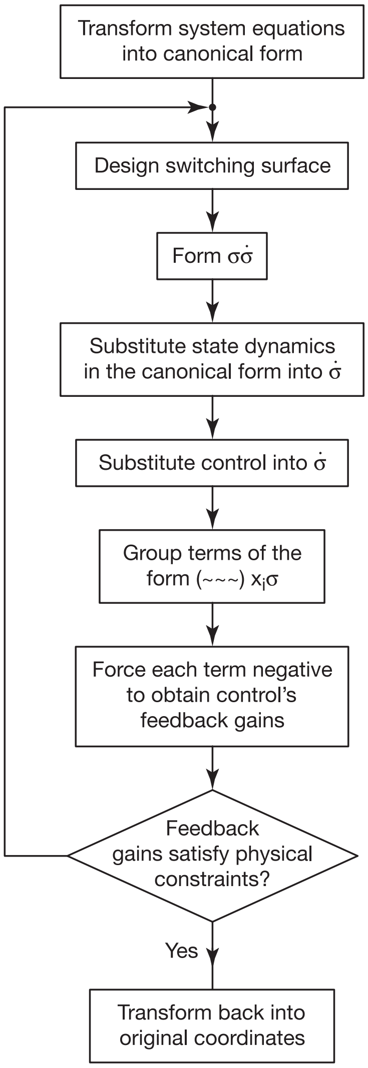


Figure 10.11: Summary of a single-input sliding mode controller design. Adapted from DeCarlo et al. [77].

that is,

$$\dot{\boldsymbol{\sigma}}(\boldsymbol{x}(t)) = \boldsymbol{S}\dot{\boldsymbol{x}}(t) = \boldsymbol{0}.$$

In summary, the system in sliding mode is described by the equations,

$$\left.\begin{array}{rcl} \boldsymbol{\sigma}(\boldsymbol{x}(t)) & = & \boldsymbol{0} \\ \dot{\boldsymbol{\sigma}}(\boldsymbol{x}(t)) & = & \boldsymbol{0}. \end{array}\right\} \tag{10.14}$$

Consider now the plant model $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$. Combining $\boldsymbol{S}\dot{\boldsymbol{x}}(t) = \boldsymbol{0}$ and the modeling equation of the plant, we get

$$\boldsymbol{S}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{S}\boldsymbol{B}\boldsymbol{u} = \boldsymbol{0}.$$

If the matrix $\boldsymbol{S}\boldsymbol{B}$ is nonsingular, then we can solve the above algebraic equation for $\boldsymbol{u}$. The result is called the *equivalent control* and it is denoted using the symbol $\boldsymbol{u}_{eq}$. We have

$$\boldsymbol{u}_{eq} = -\left(\boldsymbol{S}\boldsymbol{B}\right)^{-1}\boldsymbol{S}\boldsymbol{A}\boldsymbol{x}. \tag{10.15}$$

Substituting $\boldsymbol{u}_{eq}$ into the plant model equation yields

$$\dot{\boldsymbol{x}} = \left(\boldsymbol{I}_n - \boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\boldsymbol{S}\right)\boldsymbol{A}\boldsymbol{x}. \tag{10.16}$$

The above system is sometimes referred to as the *equivalent system.* Thus, we can describe the behavior of the system in sliding alternatively as

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} & = & \left(\boldsymbol{I}_n - \boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\boldsymbol{S}\right)\boldsymbol{A}\boldsymbol{x} \\ \boldsymbol{S}\boldsymbol{x} & = & \boldsymbol{0}. \end{array}\right\} \tag{10.17}$$

Note that in sliding, the system is governed by a reduced set of differential equations that are obtained by combining the above two sets of equations.

10.5 Sliding Surface Design

We give a sufficient condition for the existence of a sliding surface $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}$ such that the plant model $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$ restricted to it has arbitrarily prescribed poles. In our discussion, we use the following notation. For a full column rank matrix $\boldsymbol{X}$, its left inverse will be denoted $\boldsymbol{X}^g$. Thus, $\boldsymbol{X}^g\boldsymbol{X} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix whose size is equal to the number of columns of $\boldsymbol{X}$. The imaginary part of a complex number z is denoted $\Im(z)$. The set of eigenvalues of a square matrix $\boldsymbol{A}$, also called its spectrum, is denoted $\operatorname{eig}(\boldsymbol{A})$.

We now state and prove a theorem that can be found in [148]. A general method for designing a sliding surface is contained implicitly in the proof of this theorem.

Theorem 10.1 *Consider the plant model*

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \tag{10.18}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$, *rank* $(\boldsymbol{B}) = m$, *and the pair* $(\boldsymbol{A}, \boldsymbol{B})$ *is controllable. Let*

$$\{\lambda_1, \lambda_2, \ldots, \lambda_{n-m}\} \in \mathbb{C},$$

where $\mathbb{C}$ *denotes the set of complex numbers, be an arbitrary set of* $n - m$ *complex numbers such that any* λ_i *with* $\Im(\lambda_i) \neq 0$ *appears in this set in conjugate pair. Then, there exist full rank matrices* $\boldsymbol{W} \in \mathbb{R}^{n\times(n-m)}$ *and* $\boldsymbol{W}^g \in \mathbb{R}^{(n-m)\times n}$ *so that* $\boldsymbol{W}^g\boldsymbol{W} = \boldsymbol{I}_{n-m}$, $\boldsymbol{W}^g\boldsymbol{B} = \boldsymbol{O}$, *and the spectrum of the matrix* $\boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W}$ *is*

$$eig(\boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W}) = \{\lambda_1, \lambda_2, \ldots, \lambda_{n-m}\}.$$

Furthermore, there exists a matrix $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ *such that* $\boldsymbol{S}\boldsymbol{B} = \boldsymbol{I}_m$ *and the system (10.18) restricted to the surface* $\boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}$ *has the poles* $\lambda_1, \lambda_2, \ldots, \lambda_{n-m}$.

Proof We first prove the existence of the desired matrices $\boldsymbol{W}$ and $\boldsymbol{W}^g$. We can find $\boldsymbol{T} \in \mathbb{R}^{n\times(n-m)}$ so that $\begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix}$ is invertible because $\boldsymbol{B}$ has full rank. Let

$$\begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{T}^g \\ \boldsymbol{B}^g \end{bmatrix}.$$

We introduce a new coordinate $\boldsymbol{z}$ such that $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix} \boldsymbol{z}$. Then, in the new coordinates the system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$ has the form

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T} & \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B} \\ \boldsymbol{B}^g\boldsymbol{A}\boldsymbol{T} & \boldsymbol{B}^g\boldsymbol{A}\boldsymbol{B} \end{bmatrix} \boldsymbol{z} + \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I}_m \end{bmatrix} \boldsymbol{u}.$$

Because the pair $(\boldsymbol{A}, \boldsymbol{B})$ is controllable, we can use the PBH rank test, given on page 228, to conclude that

$$\text{rank} \begin{bmatrix} \lambda\boldsymbol{I}_{n-m} - \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T} & \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B} \end{bmatrix} = n - m \quad \text{for all} \quad \lambda \in \mathbb{C},$$

which means that the pair $(\boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T}, \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B})$ is controllable. The set $\{\lambda_1, \lambda_2, \ldots, \lambda_{n-m}\}$ is symmetric with respect to the real axis, therefore we can find a matrix $\boldsymbol{F} \in \mathbb{R}^{m\times(n-m)}$ such that

$$\text{eig}\,(\boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T} - \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B}\boldsymbol{F}) = \{\lambda_1, \lambda_2, \ldots, \lambda_{n-m}\}.$$

Let

$$\boldsymbol{W}^g = \boldsymbol{T}^g \tag{10.19}$$

and

$$\boldsymbol{W} = \boldsymbol{T} - \boldsymbol{B}\boldsymbol{F}. \tag{10.20}$$

Then, because by construction $\boldsymbol{T}^g\boldsymbol{T} = \boldsymbol{I}_{n-m}$ and $\boldsymbol{T}^g\boldsymbol{B} = \boldsymbol{O}$, we have $\boldsymbol{W}^g\boldsymbol{W} = \boldsymbol{I}_{n-m}$ and $\boldsymbol{W}^g\boldsymbol{B} = \boldsymbol{O}$. Note that

$$\boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W} = \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T} - \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B}\boldsymbol{F}.$$

Hence,

$$\operatorname{eig}(\boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W}) = \operatorname{eig}(\boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T} - \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B}\boldsymbol{F}) = \{\lambda_1, \lambda_2, \ldots, \lambda_{n-m}\}.$$

We next prove the existence of the sliding surface with the desired properties. We first note that the matrix $\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B} \end{bmatrix}$ is invertible because $\boldsymbol{W}^g\boldsymbol{W} = \boldsymbol{I}_{n-m}$, $\boldsymbol{W}^g\boldsymbol{B} = \boldsymbol{O}$, and $\boldsymbol{W}$, $\boldsymbol{B}$ are full rank. Hence, $\operatorname{Range}(\boldsymbol{W}) \cap \operatorname{Range}(\boldsymbol{B}) = \{\boldsymbol{0}\}$ and we write

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{W}^g \\ \boldsymbol{S} \end{bmatrix} \tag{10.21}$$

for some $\boldsymbol{S} \in \mathbb{R}^{m\times n}$. Note that $\boldsymbol{S}\boldsymbol{B} = \boldsymbol{I}_m$ and $\boldsymbol{S}\boldsymbol{W} = \boldsymbol{O}$. We introduce new coordinates

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}^g \\ \boldsymbol{S} \end{bmatrix} \boldsymbol{x}, \tag{10.22}$$

where $\boldsymbol{z}_2 = \boldsymbol{S}\boldsymbol{x}$. In the new coordinates, the equivalent system $\dot{\boldsymbol{x}} = (\boldsymbol{I}_n - \boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\boldsymbol{S})\,\boldsymbol{A}\boldsymbol{x}$ takes the form

$$\begin{aligned} \dot{\boldsymbol{z}} &= \begin{bmatrix} \boldsymbol{W}^g \\ \boldsymbol{S} \end{bmatrix} \left(\boldsymbol{I}_n - \boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\boldsymbol{S}\right) \boldsymbol{A} \begin{bmatrix} \boldsymbol{W} & \boldsymbol{B} \end{bmatrix} \boldsymbol{z} \\ &= \begin{bmatrix} \boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W} & \boldsymbol{W}^g\boldsymbol{A}\boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \boldsymbol{z}. \end{aligned}$$

The restriction to $\boldsymbol{z}_2 = \boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}$ is

$$\dot{\boldsymbol{z}}_1 = \boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W}\boldsymbol{z}_1,$$

and hence the system restricted to the sliding surface $\boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}$ has the same eigenvalues as the matrix $\boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W}$. The proof of the theorem is now complete.

□

Note that by (10.20),

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{n-m} & \boldsymbol{O} \\ -\boldsymbol{F} & \boldsymbol{I}_m \end{bmatrix}.$$

Therefore,

$$\begin{aligned} \begin{bmatrix} \boldsymbol{W} & \boldsymbol{B} \end{bmatrix}^{-1} &= \begin{bmatrix} \boldsymbol{I}_{n-m} & \boldsymbol{O} \\ -\boldsymbol{F} & \boldsymbol{I}_m \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \boldsymbol{I}_{n-m} & \boldsymbol{O} \\ \boldsymbol{F} & \boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{T}^g \\ \boldsymbol{B}^g \end{bmatrix}. \end{aligned} \tag{10.23}$$

Thus the inverse of the matrix $\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B} \end{bmatrix}$ has the form

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{W}^g \\ \boldsymbol{B}^g + \boldsymbol{F}\boldsymbol{T}^g \end{bmatrix},$$

where we invoked (10.19), that is, $\boldsymbol{W}^g = \boldsymbol{T}^g$. Taking into account (10.21), we obtain

$$\boxed{\boldsymbol{S} = \boldsymbol{B}^g + \boldsymbol{F}\boldsymbol{T}^g} \tag{10.24}$$

We combine the above results into a sliding surface design algorithm.

Algorithm 10.1 Switching Surface Construction

1. Find a matrix $\boldsymbol{T} \in \mathbb{R}^{n\times(n-m)}$ such that $\begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix}$ is invertible.

2. Compute
$$\begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{T}^g \\ \boldsymbol{B}^g \end{bmatrix}.$$

3. Find a matrix $\boldsymbol{F} \in \mathbb{R}^{m\times(n-m)}$ such that
$$\operatorname{eig}(\boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T} - \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B}\boldsymbol{F}) = \{\lambda_1, \lambda_2, \ldots, \lambda_{n-m}\}.$$

4. Construct the sliding surface matrix $\boldsymbol{S}$ using the relation,
$$\boldsymbol{S} = \boldsymbol{B}^g + \boldsymbol{F}\boldsymbol{T}^g.$$

We illustrate the above algorithm with a simple numerical example.

Example 10.4 Given a dynamical system model of the form (10.18), where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

We design a sliding surface

$$\boldsymbol{\sigma} = \boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}$$

so that the system restricted to it will behave like a linear system described by

$$\dot{z} = -z.$$

Following the steps of the of the design algorithm, we first find a matrix $\boldsymbol{T}$ such that the augmented matrix $\begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix}$ is invertible. For example, we can select

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top.$$

Note that

$$\begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix} = \boldsymbol{I}_3.$$

Hence, the inverse

$$\begin{bmatrix} \boldsymbol{T} & \boldsymbol{B} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{T}^g \\ \boldsymbol{B}^g \end{bmatrix} = \boldsymbol{I}_3.$$

Next, we find $\boldsymbol{F} \in \mathbb{R}^{m\times(n-m)}$ such that

$$\boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T} - \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B}\boldsymbol{F} = -1.$$

We have

$$\begin{aligned} \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{T} - \boldsymbol{T}^g\boldsymbol{A}\boldsymbol{B}\boldsymbol{F} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &\quad - \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \end{bmatrix} \\ &= -f_{11} \\ &= -1. \end{aligned}$$

We select

$$\boldsymbol{F} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

We then compute

$$\begin{aligned}\boldsymbol{S} &= \boldsymbol{B}^g + \boldsymbol{F}\boldsymbol{T}^g \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.\end{aligned}$$

Hence,

$$\boldsymbol{S} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

10.6 Sliding Mode and System Zeros

In this section, we show that the poles of the reduced-order system in sliding are the system zeros of the dynamical system described by

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{\sigma} = \boldsymbol{S}\boldsymbol{x}, \end{cases} \tag{10.25}$$

where $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ and we assume that $\det \boldsymbol{S}\boldsymbol{B} \neq 0$. Recall from Section 1.5 that the system zeros, or invariant zeros, are defined using the system matrix. The system matrix, $\boldsymbol{P}(s)$, corresponding to the plant (10.25) has the form

$$\boldsymbol{P}(s) = \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{S} & \boldsymbol{O} \end{bmatrix}. \tag{10.26}$$

The system matrix $\boldsymbol{P}(s)$ corresponding to the plant (10.25) is a square matrix of order $n+m$. The zeros of the determinant of the system matrix (10.26) are the system zeros of (10.25). The system zeros are invariant under the following set of transformations:

1. nonsingular state variable transformation;

2. nonsingular transformation of the inputs;

3. nonsingular transformation of the outputs;

4. state-feedback to the inputs;

5. output-feedback to the rates of change of the states.

To proceed, we evaluate the determinant of the system matrix (10.26). To facilitate this task, we premultiply the system matrix by the matrix

$$\begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{O} \\ -\boldsymbol{S}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1} & \boldsymbol{I}_m \end{bmatrix}.$$

The determinant of the above matrix is unity and so the determinant of the product is the same as the determinant of the system matrix,

$$\begin{aligned} \det \boldsymbol{P}(s) &= \det \left(\begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{O} \\ -\boldsymbol{S}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1} & \boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{S} & \boldsymbol{O} \end{bmatrix} \right) \\ &= \det \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{S}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B} \end{bmatrix} \\ &= \det (s\boldsymbol{I}_n - \boldsymbol{A}) \det \left(\boldsymbol{S}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B}\right). \end{aligned} \tag{10.27}$$

We next evaluate the characteristic polynomial of the equivalent system (10.16). While performing manipulations, we use the result of Exercise A.4 to obtain

$$\begin{aligned} &\det \left(s\boldsymbol{I}_n - \boldsymbol{A} + \boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\boldsymbol{S}\boldsymbol{A}\right) \\ &\quad = \det(s\boldsymbol{I}_n - \boldsymbol{A}) \det \left(\boldsymbol{I}_n + (s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\boldsymbol{S}\boldsymbol{A}\right) \\ &\quad = \det(s\boldsymbol{I}_n - \boldsymbol{A}) \det \left(\boldsymbol{I}_m + \boldsymbol{S}\boldsymbol{A}(s\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\right). \end{aligned} \tag{10.28}$$

In our further manipulations, we use the identity

$$\boldsymbol{A}\left(\operatorname{Adj}(s\boldsymbol{I}_n - \boldsymbol{A})\right) = s\operatorname{Adj}(s\boldsymbol{I}_n - \boldsymbol{A}) - \det(s\boldsymbol{I}_n - \boldsymbol{A})\boldsymbol{I}_n. \tag{10.29}$$

The above follows from the fact that

$$(s\boldsymbol{I}_n - \boldsymbol{A})(s\boldsymbol{I}_n - \boldsymbol{A})^{-1} = \boldsymbol{I}_n.$$

which is equivalent to

$$(s\boldsymbol{I}_n - \boldsymbol{A})\operatorname{Adj}(s\boldsymbol{I}_n - \boldsymbol{A}) = \det(s\boldsymbol{I}_n - \boldsymbol{A})\boldsymbol{I}_n. \tag{10.30}$$

Then we use (10.29) to get

$$
\begin{aligned}
&\det\left(\boldsymbol{I}_m+\boldsymbol{S}\boldsymbol{A}(s\boldsymbol{I}_n-\boldsymbol{A})^{-1}\boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\right)\\
&=\det\left(\frac{\det(s\boldsymbol{I}_n-\boldsymbol{A})\boldsymbol{I}_m+\boldsymbol{S}\left(s\operatorname{Adj}(s\boldsymbol{I}_n-\boldsymbol{A})-\det(s\boldsymbol{I}_n-\boldsymbol{A})\boldsymbol{I}_n\right)\boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}}{\det(s\boldsymbol{I}_n-\boldsymbol{A})}\right)\\
&=\det\left(\frac{\det(s\boldsymbol{I}_n-\boldsymbol{A})\boldsymbol{I}_m+s\boldsymbol{S}\operatorname{Adj}(s\boldsymbol{I}_n-\boldsymbol{A})-\det(s\boldsymbol{I}_n-\boldsymbol{A})\boldsymbol{I}_n(\boldsymbol{S}\boldsymbol{B})(\boldsymbol{S}\boldsymbol{B})^{-1}}{\det(s\boldsymbol{I}_n-\boldsymbol{A})}\right)\\
&=\det\left(\frac{s\boldsymbol{S}\operatorname{Adj}(s\boldsymbol{I}_n-\boldsymbol{A})\boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}}{\det(s\boldsymbol{I}_n-\boldsymbol{A})}\right)\\
&=s^m\det\left(\boldsymbol{S}(s\boldsymbol{I}_n-\boldsymbol{A})^{-1}\boldsymbol{B}\right)\det(\boldsymbol{S}\boldsymbol{B})^{-1}.
\end{aligned}
\tag{10.31}
$$

Using (10.31), we represent (10.28) as

$$
\begin{aligned}
&\det\left(s\boldsymbol{I}_n-\boldsymbol{A}+\boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\boldsymbol{S}\boldsymbol{A}\right)\\
&\quad=s^m\det(\boldsymbol{S}\boldsymbol{B})^{-1}\det(s\boldsymbol{I}_n-\boldsymbol{A})\det\left(\boldsymbol{S}(s\boldsymbol{I}_n-\boldsymbol{A})^{-1}\boldsymbol{B}\right).
\end{aligned}
$$

Taking into account (10.27) gives

$$
\boxed{\det\left(s\boldsymbol{I}_n-\boldsymbol{A}+\boldsymbol{B}(\boldsymbol{S}\boldsymbol{B})^{-1}\boldsymbol{S}\boldsymbol{A}\right)=\det(\boldsymbol{S}\boldsymbol{B})^{-1}s^m\det\boldsymbol{P}(s)}
\tag{10.32}
$$

Thus the poles of the equivalent system (10.16) contain m poles at the origin and the remaining $n-m$ poles are the system zeros of the system (10.25) characterized by the triple $(\boldsymbol{A},\boldsymbol{B},\boldsymbol{S})$.

We conclude that the dynamics, that is, the poles, of the system modeled by (10.25) in sliding along $\boldsymbol{\sigma}=\boldsymbol{S}\boldsymbol{x}$ are determined by its system zeros. This fact leads to the observation that the switching surface design can be viewed as choosing an output matrix $\boldsymbol{S}$ so that the system (10.25) has a desired set of system zeros locations, which in turn govern the dynamics of the system in sliding along $\boldsymbol{S}\boldsymbol{x}=\boldsymbol{0}$.

10.7 Regulator Design

In this Section, we present a regulator design algorithm for a class of nonlinear dynamical systems using an approach that is rooted in variable structure sliding mode control.

This design methodology consists of two steps: the manifold design and the control law design. We use Algorithm 10.1 on page 554 to construct an asymptotically stable invariant manifold so that all trajectories confined to this manifold converge to the origin. Once a manifold is constructed, the control strategy is obtained by solving the first-order matrix differential equation. The proposed design procedure is tested on the model of a two-link robotic manipulator from Section 2.4.6.

10.7.1 Outline of the Approach

Consider a nonlinear dynamical system modeled as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}, \tag{10.33}$$

where $\boldsymbol{x}$ is a vector of state variables of dimension n, $\boldsymbol{u}$ is the input vector of dimension m, $\boldsymbol{f}(\boldsymbol{x})$ is a state-dependent function of dimension n, $\boldsymbol{B}(\boldsymbol{x})$ is a state-dependent input function matrix of dimension $n \times m$ and it has full column rank everywhere.

Let $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{S}\boldsymbol{x} : \mathbb{R}^n \to \mathbb{R}^m$ be a vector function of dimension m.

The control objective is to devise a control policy that stabilizes the system (10.33) by steering its trajectories to a manifold,

$$\mathcal{M} = \{\boldsymbol{x} : \boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{0}\} \tag{10.34}$$

and forces the trajectories to stay on the manifold thereafter.

We further require that, (i) the system (10.33) confined to the manifold $\mathcal{M} = \{\boldsymbol{x} : \boldsymbol{\sigma} = \boldsymbol{0}\}$ is asymptotically stable to the origin, (ii) $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{S}\boldsymbol{x}$ is constructed so that the square $m \times m$ matrix

$$\boldsymbol{\sigma}_{\boldsymbol{x}}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{S}\boldsymbol{B}(\boldsymbol{x})$$

is invertible, where $\boldsymbol{\sigma}_{\boldsymbol{x}}$ is the Jacobian matrix of $\boldsymbol{\sigma}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$. We will show that when these two conditions are satisfied, the solution to the above control problem is obtained by solving the associated first-order matrix differential equation,

$$\boldsymbol{T}\dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} = \boldsymbol{0}, \tag{10.35}$$

where $\boldsymbol{T} \in \mathbb{R}^{m\times m}$ is a symmetric positive definite matrix, which is a design parameter. Substituting (10.33) into (10.35) yields

$$\begin{aligned} \boldsymbol{T}\dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} &= \boldsymbol{T}\boldsymbol{S}\dot{\boldsymbol{x}} + \boldsymbol{S}\boldsymbol{x} \\ &= \boldsymbol{T}\boldsymbol{S}\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{T}\boldsymbol{S}\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{S}\boldsymbol{x} \\ &= \boldsymbol{0}. \end{aligned}$$

Solving the above gives

$$\begin{aligned} \boldsymbol{u} &= -(\boldsymbol{T}\boldsymbol{S}\boldsymbol{B}(\boldsymbol{x}))^{-1}(\boldsymbol{T}\boldsymbol{S}\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{S}\boldsymbol{x}) \\ &= -(\boldsymbol{S}\boldsymbol{B}(\boldsymbol{x}))^{-1}\boldsymbol{S}\boldsymbol{f}(\boldsymbol{x}) - (\boldsymbol{S}\boldsymbol{B}(\boldsymbol{x}))^{-1}\boldsymbol{T}^{-1}\boldsymbol{S}\boldsymbol{x}. \end{aligned} \tag{10.36}$$

The use of the control law (10.36) in the nonlinear dynamical systems (10.33) makes the invariant manifold $\mathcal{M} = \{\boldsymbol{x} : \boldsymbol{\sigma} = \boldsymbol{0}\}$ attractive. The control law (10.36) drives closed-loop trajectories of (10.33) from any initial condition towards the invariant manifold $\mathcal{M}$. As stated above, $\boldsymbol{\sigma}$ is constructed in such a way that all trajectories of (10.33) confined on $\mathcal{M}$ converge to the origin. Therefore, the stability of the closed-loop trajectories is achieved in two modes: (i) closed-loop trajectories converging to the manifold $\mathcal{M} = \{\boldsymbol{x} : \boldsymbol{\sigma} = \boldsymbol{0}\}$, and (ii) closed-loop trajectories moving along the manifold $\mathcal{M} = \{\boldsymbol{x} : \boldsymbol{\sigma} = \boldsymbol{0}\}$ towards the origin.

The design method employs the following two-step design procedure:

1. Construct an asymptotically stable invariant manifold $\mathcal{M} = \{\boldsymbol{x} : \boldsymbol{\sigma} = \boldsymbol{0}\}$, (i.e. all trajectories of the system confined to $\mathcal{M}$ converge asymptotically to the origin).

2. Construct the controller $\boldsymbol{u}$ using (10.36).

This approach shares similarity with that of the variable structure sliding mode control (SMC). This relation was discussed in [294, 295, 254]. In the development of the proposed regulator, we adopt the integral tracking method used in the sliding mode tracking controller design described in [92]. The resulting controller drives the closed-loop dynamics so that the integral of the tracking error converges to zero.

10.7.2 System Description

The class of nonlinear systems that we consider is modeled by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \\ &= \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \hat{\boldsymbol{B}}\tilde{\boldsymbol{u}}, \end{aligned} \tag{10.37}$$

where we assume that the system (10.37) is transformable into the following "regular form" [333, 92],

$$\begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21}(\boldsymbol{x}) & \boldsymbol{A}_{22}(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{B}_u \end{bmatrix} \tilde{\boldsymbol{u}}, \tag{10.38}$$

where $\boldsymbol{x}_1$ is in $\mathbb{R}^{n-m}$, $\boldsymbol{x}_2$ is in $\mathbb{R}^m$, and the control vector $\boldsymbol{u}$ is in $\mathbb{R}^m$. The matrix $\boldsymbol{O} \in \mathbb{R}^{(n-m)\times m}$ is a matrix with zero entries. The matrix $\boldsymbol{B}_u \in \mathbb{R}^{m\times m}$ is invertible and the matrices $\boldsymbol{A}_{ij}$, $i = 1, 2$, $j = 1, 2$ have appropriate dimensions. Note that $\boldsymbol{A}_{1j}$, $j = 1, 2$, are constant matrices, while $\boldsymbol{A}_{2j}$, $j = 1, 2$, are state-dependent matrices. For a constructive algorithm that can be used to transform (10.37) into the regular form given by (10.38), we refer to [92, p. 66].

An example of physical systems that can be modeled using the above non-linear regular form are natural Lagrangian systems. A simple example of a natural Lagrangian system whose model can be represented in the regular form is a planar two-link robot presented in Section 2.4.6. Indeed, let $\boldsymbol{x}_1 = [\theta_1\ \theta_2]^\top$ be the vector of joint angles, $\boldsymbol{\tau} = [\tau_1\ \tau_2]^\top$ be the vector of torques applied to the joints. Then the dynamics of the two-link robotic manipulator can be represented as

$$\boldsymbol{H}(\boldsymbol{x}_1)\ddot{\boldsymbol{x}}_1 + \boldsymbol{C}(\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1)\dot{\boldsymbol{x}}_1 + \boldsymbol{g}(\boldsymbol{x}_1) = \boldsymbol{\tau}, \tag{10.39}$$

where $\boldsymbol{H}(\boldsymbol{x}_1)$ is the manipulator inertia matrix and $\boldsymbol{C}(\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1)\dot{\boldsymbol{x}}_1$ is the vector of centripetal and Coriolis torques, and $\boldsymbol{g}(\boldsymbol{x}_1) = \begin{bmatrix} g_1(\boldsymbol{x}_1) & g_2(\boldsymbol{x}_1) \end{bmatrix}^\top$ is the vector of gravitational torques. Because the inertia matrix $\boldsymbol{H}(\boldsymbol{x}_1)$ is positive definite over the whole workspace, we know that $\boldsymbol{H}^{-1}(\boldsymbol{x}_1)$ exists and is also positive definite. Let

$$\boldsymbol{G}(\boldsymbol{x}_1) = \begin{bmatrix} \frac{g_1(\boldsymbol{x}_1)}{x_1} & 0 \\ 0 & \frac{g_2(\boldsymbol{x}_1)}{x_2} \end{bmatrix}.$$

Using the above we represent the term $\boldsymbol{g}(\boldsymbol{x}_1)$ as

$$\boldsymbol{g}(\boldsymbol{x}_1) = \boldsymbol{G}(\boldsymbol{x}_1)\boldsymbol{x}_1. \tag{10.40}$$

Taking into account (10.40), we rewrite (10.39) as

$$\ddot{\boldsymbol{x}}_1 = -\boldsymbol{H}^{-1}(\boldsymbol{x}_1)\boldsymbol{C}(\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1)\dot{\boldsymbol{x}}_1 - \boldsymbol{H}^{-1}(\boldsymbol{x}_1)\boldsymbol{G}(\boldsymbol{x}_1)\boldsymbol{x}_1 + \boldsymbol{H}^{-1}(\boldsymbol{x}_1)\boldsymbol{\tau}. \tag{10.41}$$

Let $\boldsymbol{x}_2 = \dot{\boldsymbol{x}}_1$ and $\tilde{\boldsymbol{u}} = \boldsymbol{H}^{-1}(\boldsymbol{x}_1)\boldsymbol{\tau}$. Then, the dynamics of the planar two-link robot modeled by (10.41) can be represented in the form of (10.38), that is,

$$\begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_2 \\ -\boldsymbol{H}^{-1}(\boldsymbol{x}_1)\boldsymbol{G}(\boldsymbol{x}_1) & -\boldsymbol{H}^{-1}(\boldsymbol{x}_1)\boldsymbol{C}(\boldsymbol{x}_1, \boldsymbol{x}_2) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I}_2 \end{bmatrix} \tilde{\boldsymbol{u}}. \tag{10.42}$$

10.7.3 Regulator Construction

Suppose that the output $\boldsymbol{y} \in \mathbb{R}^p$ of system (10.38) has the form,

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix}. \tag{10.43}$$

Our objective is to design a regulator so that the output $\boldsymbol{y}$ tracks a constant reference vector $\boldsymbol{r} \in \mathbb{R}^p$. We introduce the vector variable $\boldsymbol{x}_r$ defined by the following differential equation,

$$\dot{\boldsymbol{x}}_r = \boldsymbol{r} - \boldsymbol{y}. \tag{10.44}$$

The controller should be designed in such a way so that

$$\lim_{t\to\infty} \dot{\boldsymbol{x}}_r(t) = \mathbf{0}.$$

To proceed, we define the augmented state vector,

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x} \end{bmatrix} = \left[\begin{array}{c} \boldsymbol{x}_r \\ \hline \underbrace{\begin{Bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{Bmatrix}}_{\boldsymbol{x}} \end{array}\right] \in \mathbb{R}^N, \tag{10.45}$$

where $N = n + p$. The dynamics of the overall augmented system are governed by the state equation,

$$\begin{bmatrix} \dot{\boldsymbol{x}}_r \\ \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{O} & -\boldsymbol{C}_1 & -\boldsymbol{C}_2 \\ \boldsymbol{O} & \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{21}(\boldsymbol{x}) & \boldsymbol{A}_{22}(\boldsymbol{x}) \end{bmatrix}}_{\tilde{\boldsymbol{A}}} \begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{I}_p \\ \boldsymbol{O} \\ \boldsymbol{O} \end{bmatrix}}_{\boldsymbol{B}_r} \boldsymbol{r} + \underbrace{\begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{O} \\ \boldsymbol{B}_u \end{bmatrix}}_{\tilde{\boldsymbol{B}}} \boldsymbol{u}. \tag{10.46}$$

Note that the matrices in the augmented state equation (10.46) are labeled as $\tilde{\boldsymbol{A}}$, $\boldsymbol{B}_r$, and $\tilde{\boldsymbol{B}}$, respectively. Hence, the augmented system model (10.46) can be represented as

$$\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} + \boldsymbol{B}_r\boldsymbol{r} + \tilde{\boldsymbol{B}}\boldsymbol{u}. \tag{10.47}$$

We make the following assumption:

Assumption 1: The pair

$$\left(\begin{bmatrix} \boldsymbol{O} & -\boldsymbol{C}_1 \\ \boldsymbol{O} & \boldsymbol{A}_{11} \end{bmatrix}, \quad \begin{bmatrix} -\boldsymbol{C}_2\boldsymbol{B}_u \\ \boldsymbol{A}_{12}\boldsymbol{B}_u \end{bmatrix} \right)$$

is controllable.

An essential element of the regulator design is the construction of an invariant manifold $\boldsymbol{\sigma} = \mathbf{0}$. Let the manifold function $\boldsymbol{\sigma}$ be defined as

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\tilde{\boldsymbol{x}}) = \boldsymbol{\sigma}(\boldsymbol{x}_r, \boldsymbol{x}_1, \boldsymbol{x}_2) = -\boldsymbol{S}_r\boldsymbol{x}_r + \boldsymbol{S}_1\boldsymbol{x}_1 + \boldsymbol{S}_2\boldsymbol{x}_2 = \begin{bmatrix} -\boldsymbol{S}_r & \boldsymbol{S}_1 & \boldsymbol{S}_2 \end{bmatrix} \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{S}}\tilde{\boldsymbol{x}}. \tag{10.48}$$

The negative sign in front of $\boldsymbol{S}_r$ is for the purpose of simplified notation. We construct $\boldsymbol{\sigma}$ so that the system modeled by (10.46) restricted to the manifold $\mathcal{M} = \{\tilde{\boldsymbol{x}} : \boldsymbol{\sigma}(\tilde{\boldsymbol{x}}) = \mathbf{0}\}$ is asymptotically stable, that is, the trajectories of the system restricted to the manifold $\mathcal{M}$

converge asymptotically towards the origin. We can use Algorithm 10.1 to construct such a manifold. We assume that

$$\det\left(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}}\right) = \det\left(\boldsymbol{S}_2\boldsymbol{B}_u\right) \neq 0. \tag{10.49}$$

Using (10.35), we perform manipulations to obtain,

$$\begin{aligned}
\boldsymbol{0} &= \boldsymbol{T}\dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} \\
&= \boldsymbol{T}\tilde{\boldsymbol{S}}\dot{\tilde{\boldsymbol{x}}} + \tilde{\boldsymbol{S}}\tilde{\boldsymbol{x}} \\
&= \boldsymbol{T}\tilde{\boldsymbol{S}}(\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} + \boldsymbol{B}_r\boldsymbol{r} + \tilde{\boldsymbol{B}}\tilde{\boldsymbol{u}}) + \tilde{\boldsymbol{S}}\tilde{\boldsymbol{x}} \\
&= \boldsymbol{T}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} + \boldsymbol{T}\tilde{\boldsymbol{S}}\boldsymbol{B}_r\boldsymbol{r} + \boldsymbol{T}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}}\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{S}}\tilde{\boldsymbol{x}}.
\end{aligned} \tag{10.50}$$

It follows from (10.48) and (10.46) that $\tilde{\boldsymbol{S}}\boldsymbol{B}_r = -\boldsymbol{S}_r$. Hence, we can re-write (10.50) as

$$\boldsymbol{0} = \boldsymbol{T}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \boldsymbol{T}\tilde{\boldsymbol{S}}_r\boldsymbol{r} + \boldsymbol{T}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}}\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{S}}\tilde{\boldsymbol{x}}. \tag{10.51}$$

Solving (10.51) for $\boldsymbol{u}$ yields

$$\boxed{\tilde{\boldsymbol{u}} = -(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} + \boldsymbol{T}^{-1}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{x}} - \boldsymbol{S}_r\boldsymbol{r})} \tag{10.52}$$

Substituting (10.52) into (10.46), we obtain the closed-loop system dynamics model,

$$\dot{\tilde{\boldsymbol{x}}} = \left(\boldsymbol{I} - \tilde{\boldsymbol{B}}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}\tilde{\boldsymbol{S}}\right)\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{B}}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}\boldsymbol{T}^{-1}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{B}}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}\boldsymbol{S}_r\boldsymbol{r} + \boldsymbol{B}_r\boldsymbol{r}. \tag{10.53}$$

In the following subsection, we show that the control law given by (10.52) forces the above closed-loop system to track the reference signal $\boldsymbol{r}$.

10.7.4 Analysis of the Closed-Loop System Behavior

To proceed with our analysis, we need the following lemma, which can be viewed as an extension of Theorem 10.1 to a class of non-liner systems.

Lemma 10.1 *Consider the system model given by* (10.46)*, where rank* $(\tilde{\boldsymbol{B}}) = m$*, Assumption 1 is satisfied, and a given set of* $N-m$ *complex numbers,* $\{\lambda_1, \ldots, \lambda_{N-m}\}$ *is such that any* λ_i *whose imaginary part is nonzero appears in the above set in a conjugate pair. Then, there exist full-rank matrices* $\boldsymbol{W} \in \mathbb{R}^{N\times(N-m)}$ *and* $\boldsymbol{W}^g \in \mathbb{R}^{(N-m)\times N}$ *such that*

$$\boldsymbol{W}^g\boldsymbol{W} = \boldsymbol{I}_{N-m}, \quad \boldsymbol{W}^g\tilde{\boldsymbol{B}} = \boldsymbol{O},$$

and the eigenvalues of $\boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W}$ *are*

$$eig(\boldsymbol{W}^g\tilde{\boldsymbol{A}}\boldsymbol{W}) = \{\lambda_1, \ldots, \lambda_{N-m}\}.$$

Proof: The proof of this lemma is based on the proof of Theorem 10.1 with an emphasis on the special structure of the non-linear plant. We begin our proof by defining an $N \times (N-m)$ real matrix of the form,

$$\boldsymbol{V} = \left[\begin{array}{c} \boldsymbol{I}_{N-m} \\ \hline \boldsymbol{O} \end{array}\right]. \tag{10.54}$$

Therefore,

$$\begin{bmatrix} \boldsymbol{V} & \tilde{\boldsymbol{B}} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{I}_{N-m} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{B}_u \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{I}_{N-m} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{B}_u^{-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}^g \\ \tilde{\boldsymbol{B}}^g \end{bmatrix}.$$

Hence,

$$\begin{bmatrix} \boldsymbol{V}^g \\ \tilde{\boldsymbol{B}}^g \end{bmatrix} \begin{bmatrix} \boldsymbol{V} & \tilde{\boldsymbol{B}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}^g\boldsymbol{V} & \boldsymbol{V}^g\tilde{\boldsymbol{B}} \\ \tilde{\boldsymbol{B}}^g\boldsymbol{V} & \tilde{\boldsymbol{B}}^g\tilde{\boldsymbol{B}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{N-m} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_m \end{bmatrix} = \boldsymbol{I}_N.$$

Note that

$$\boldsymbol{V}^g\boldsymbol{V} = \boldsymbol{I}_{N-m} \quad \text{and} \quad \boldsymbol{V}^g\tilde{\boldsymbol{B}} = \boldsymbol{O}. \tag{10.55}$$

By Assumptions 1, the pair

$$\left(\boldsymbol{V}^g\tilde{\boldsymbol{A}}\boldsymbol{V}, \boldsymbol{V}^g\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}\right) = \left(\begin{bmatrix} \boldsymbol{O} & -\boldsymbol{C}_1 \\ \boldsymbol{O} & \boldsymbol{A}_{11} \end{bmatrix}, \quad \begin{bmatrix} -\boldsymbol{C}_2\boldsymbol{B}_u \\ \boldsymbol{A}_{12}\boldsymbol{B}_u \end{bmatrix}\right)$$

is controllable. Therefore, for any set of complex numbers,

$$\{\lambda_1, \lambda_2, \ldots, \lambda_{N-m}\}$$

symmetric with respect to the real axis, we can always find a matrix $\boldsymbol{F}$ such that

$$\{\lambda_1, \lambda_2, \ldots, \lambda_{N-m}\} = \operatorname{eig}\left(\boldsymbol{V}^g\tilde{\boldsymbol{A}}\boldsymbol{V} - \boldsymbol{V}^g\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}\boldsymbol{F}\right).$$

The matrix $\boldsymbol{F}$ can be determined using any available method for pole placement. We proceed by performing the following manipulations,

$$\boldsymbol{V}^g\tilde{\boldsymbol{A}}\boldsymbol{V} - \boldsymbol{V}^g\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}\boldsymbol{F} = \boldsymbol{V}^g\tilde{\boldsymbol{A}}\left(\boldsymbol{V} - \tilde{\boldsymbol{B}}\boldsymbol{F}\right) = \boldsymbol{W}^g\tilde{\boldsymbol{A}}\boldsymbol{W}, \tag{10.56}$$

where $\boldsymbol{W}^g = \boldsymbol{V}^g$ and $\boldsymbol{W} = \boldsymbol{V} - \tilde{\boldsymbol{B}}\boldsymbol{F}$. Note that by (10.55),

$$\boldsymbol{W}^g\boldsymbol{W} = \boldsymbol{V}^g\left(\boldsymbol{V} - \tilde{\boldsymbol{B}}\boldsymbol{F}\right) = \boldsymbol{V}^g\boldsymbol{V} = \boldsymbol{I}_{N-m}. \tag{10.57}$$

We also have, $\boldsymbol{V}^g\tilde{\boldsymbol{B}} = \boldsymbol{W}^g\tilde{\boldsymbol{B}} = \boldsymbol{O}$, which completes the proof of the lemma.

$\square$

Theorem 10.2 *Suppose that the assumptions of Lemma 10.1 are satisfied. Then there exists an invariant manifold matrix* $\tilde{\boldsymbol{S}} \in \mathbb{R}^{m \times N}$ *such that* $\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}} = \boldsymbol{I}_m$ *and the closed-loop system* (10.53) *asymptotically tracks the reference signal* $\boldsymbol{r}$*, that is,* $\boldsymbol{y}(\infty) = \boldsymbol{r}$*.*

Proof: Consider the matrix, $\begin{bmatrix} \boldsymbol{W} & \tilde{\boldsymbol{B}} \end{bmatrix}$. Note that because $\boldsymbol{W} = \boldsymbol{V} - \tilde{\boldsymbol{B}}\boldsymbol{F}$, we have

$$\begin{bmatrix} \boldsymbol{W} & \tilde{\boldsymbol{B}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V} & \tilde{\boldsymbol{B}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{N-m} & \boldsymbol{O} \\ -\boldsymbol{F} & \boldsymbol{I}_m \end{bmatrix}.$$

Thus the matrix $\begin{bmatrix} \boldsymbol{W} & \tilde{\boldsymbol{B}} \end{bmatrix}$ is invertible and its inverse has the form

$$\begin{aligned} \begin{bmatrix} \boldsymbol{W} & \tilde{\boldsymbol{B}} \end{bmatrix}^{-1} &= \begin{bmatrix} \boldsymbol{I}_{N-m} & \boldsymbol{O} \\ -\boldsymbol{F} & \boldsymbol{I}_m \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{V} & \tilde{\boldsymbol{B}} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \boldsymbol{I}_{N-m} & \boldsymbol{O} \\ \boldsymbol{F} & \boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^g \\ \tilde{\boldsymbol{B}}^g \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{V}^g \\ \tilde{\boldsymbol{B}}^g + \boldsymbol{F}\boldsymbol{V}^g \end{bmatrix}. \end{aligned}$$

As in (10.24) in the proof of Theorem 10.1, we take

$$\begin{aligned} \tilde{\boldsymbol{S}} &= \tilde{\boldsymbol{B}}^g + \boldsymbol{F}\boldsymbol{V}^g \\ &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{B}_u \end{bmatrix} + \boldsymbol{F} \begin{bmatrix} \boldsymbol{I}_{N-m} & \boldsymbol{O} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{F} & \boldsymbol{B}_u^{-1} \end{bmatrix}. \end{aligned}$$

Note that by (10.55),

$$\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{B}_u^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{O} \\ \boldsymbol{B}_u \end{bmatrix} = \boldsymbol{I}_m.$$

Consider the following state-space transformation,

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}^g \\ \tilde{\boldsymbol{S}} \end{bmatrix} \tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{V}^g \\ \tilde{\boldsymbol{S}} \end{bmatrix} \tilde{\boldsymbol{x}}.$$

Note that $\boldsymbol{z}_2 = \tilde{\boldsymbol{S}}\tilde{\boldsymbol{x}} = \boldsymbol{\sigma}$. We also have

$$\begin{bmatrix} \boldsymbol{W}^g \\ \tilde{\boldsymbol{S}} \end{bmatrix} \begin{bmatrix} \boldsymbol{W} & \tilde{\boldsymbol{B}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}^g\boldsymbol{W} & \boldsymbol{W}^g\tilde{\boldsymbol{B}} \\ \tilde{\boldsymbol{S}}\boldsymbol{W} & \tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{N-m} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_m \end{bmatrix}, \tag{10.58}$$

where $\boldsymbol{I}_{N-m}$ is the identity matrix of dimension $(N-m)\times(N-m)$. Therefore, the matrices in (10.58) satisfy the following relations,

$$\left.\begin{array}{rclrcl} \boldsymbol{W}^g\boldsymbol{W} &=& \boldsymbol{I}_{N-m}, & \boldsymbol{W}^g\tilde{\boldsymbol{B}} &=& \boldsymbol{O}\in\mathbb{R}^{(N-m)\times m}, \\ \tilde{\boldsymbol{S}}\boldsymbol{W} &=& \boldsymbol{O}\in\mathbb{R}^{m\times(N-m)}, & \tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}} &=& \boldsymbol{I}_m. \end{array}\right\} \tag{10.59}$$

Using matrix relations shown in (10.59), we represent the closed-loop model (10.53) in the $\boldsymbol{z}$ coordinates,

$$\begin{aligned}
\begin{bmatrix} \dot{\boldsymbol{z}}_1 \\ \dot{\boldsymbol{z}}_2 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{W}^g \\ \tilde{\boldsymbol{S}} \end{bmatrix}(\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}\boldsymbol{T}^{-1}\tilde{\boldsymbol{S}})\begin{bmatrix} \boldsymbol{W} & \tilde{\boldsymbol{B}} \end{bmatrix}\begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} \\
&\quad -\begin{bmatrix} \boldsymbol{W}^g \\ \tilde{\boldsymbol{S}} \end{bmatrix}\tilde{\boldsymbol{B}}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}\tilde{\boldsymbol{S}}\boldsymbol{B}_r\boldsymbol{r}+\begin{bmatrix} \boldsymbol{W}^g \\ \tilde{\boldsymbol{S}} \end{bmatrix}\boldsymbol{B}_r\boldsymbol{r} \\
&= \begin{bmatrix} \boldsymbol{W}^g\tilde{\boldsymbol{A}}\boldsymbol{W} & \boldsymbol{W}^g\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}} \\ \boldsymbol{O} & -\boldsymbol{T}^{-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}+\begin{bmatrix} \boldsymbol{W}^g \\ \tilde{\boldsymbol{S}} \end{bmatrix}\left(-\tilde{\boldsymbol{B}}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}\tilde{\boldsymbol{S}}\boldsymbol{B}_r+\boldsymbol{B}_r\right)\boldsymbol{r} \\
&= \begin{bmatrix} \boldsymbol{W}^g\tilde{\boldsymbol{A}}\boldsymbol{W} & \boldsymbol{W}^g\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}} \\ \boldsymbol{O} & -\boldsymbol{T}^{-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}+\begin{bmatrix} \boldsymbol{O}+\boldsymbol{W}^g\boldsymbol{B}_r \\ -\tilde{\boldsymbol{S}}\boldsymbol{B}_r+\tilde{\boldsymbol{S}}\boldsymbol{B}_r \end{bmatrix}\boldsymbol{r} \\
&= \begin{bmatrix} \boldsymbol{W}^g\tilde{\boldsymbol{A}}\boldsymbol{W} & \boldsymbol{W}^g\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}} \\ \boldsymbol{O} & -\boldsymbol{T}^{-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}+\begin{bmatrix} \boldsymbol{W}^g\boldsymbol{B}_r \\ \boldsymbol{O} \end{bmatrix}\boldsymbol{r}.
\end{aligned} \tag{10.60}$$

Observe that $\boldsymbol{W}^g\tilde{\boldsymbol{B}}(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{B}})^{-1}\tilde{\boldsymbol{S}}\boldsymbol{B}_r=\boldsymbol{O}$ because $\boldsymbol{W}^g\tilde{\boldsymbol{B}}=\boldsymbol{O}$. Since $\boldsymbol{T}$ is positive definite, the eigenvalues of $-\boldsymbol{T}^{-1}$ are all in the open left-hand complex plane. This implies that the dynamics of $\boldsymbol{z}_2=\boldsymbol{\sigma}$ are asymptotically stable, that is, the invariant manifold, $\{\boldsymbol{\sigma}=\boldsymbol{0}\}$, is asymptotically attractive. By (10.56), the matrix $\boldsymbol{W}$ can be chosen so that the eigenvalues of the constant matrix $\boldsymbol{W}^g\tilde{\boldsymbol{A}}\boldsymbol{W}$ are in the open left-hand complex plane. In the steady-state the overall system trajectories converge to the point on the manifold, $\{\boldsymbol{z}_2=\boldsymbol{\sigma}=\boldsymbol{0}\}$, given by

$$\boldsymbol{0}=\boldsymbol{W}^g\tilde{\boldsymbol{A}}\boldsymbol{W}\boldsymbol{z}_1+\boldsymbol{W}^g\boldsymbol{B}_r\boldsymbol{r}. \tag{10.61}$$

We partition $\boldsymbol{F}$ as

$$\boldsymbol{F}=\begin{bmatrix} \boldsymbol{F}_1 & \boldsymbol{F}_2 \end{bmatrix}, \tag{10.62}$$

where $\boldsymbol{F}_1\in\mathbb{R}^{m\times p}$ and $\boldsymbol{F}_2\in\mathbb{R}^{m\times(n-m)}$. We have,

$$\boldsymbol{W}=\boldsymbol{V}-\tilde{\boldsymbol{B}}\boldsymbol{F}=\begin{bmatrix} \boldsymbol{I}_p & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n-m} \\ -\boldsymbol{B}_u\boldsymbol{F}_1 & -\boldsymbol{B}_u\boldsymbol{F}_2 \end{bmatrix} \quad\text{and}\quad \boldsymbol{W}^g\boldsymbol{B}_r=\begin{bmatrix} \boldsymbol{I}_p \\ \boldsymbol{O}_{(n-m)\times p} \end{bmatrix}. \tag{10.63}$$

Recall that $\boldsymbol{W}^g = \boldsymbol{V}^g$. Using the above and the fact that

$$\boldsymbol{z}_1 = \boldsymbol{W}^g \tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x}_1 \end{bmatrix},$$

we express (10.61) in the form,

$$\begin{aligned}
\mathbf{0} &= \boldsymbol{V}^g \begin{bmatrix} \boldsymbol{O} & -\boldsymbol{C}_1 & -\boldsymbol{C}_2 \\ \boldsymbol{O} & \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{21}(\boldsymbol{x}) & \boldsymbol{A}_{22}(\boldsymbol{x}) \end{bmatrix} \boldsymbol{W} \boldsymbol{z}_1 + \begin{bmatrix} \boldsymbol{I}_p \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{r} \\
&= \begin{bmatrix} \boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{F}_1 & -\boldsymbol{C}_1 + \boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{F}_2 \\ -\boldsymbol{A}_{12} \boldsymbol{B}_u \boldsymbol{F}_1 & \boldsymbol{A}_{11} - \boldsymbol{A}_{12} \boldsymbol{B}_u \boldsymbol{F}_2 \end{bmatrix} \boldsymbol{W}^g \tilde{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{I}_p \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{r} \\
&= \begin{bmatrix} \boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{F}_1 & -\boldsymbol{C}_1 + \boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{F}_2 \\ -\boldsymbol{A}_{12} \boldsymbol{B}_u \boldsymbol{F}_1 & \boldsymbol{A}_{11} - \boldsymbol{A}_{12} \boldsymbol{B}_u \boldsymbol{F}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x}_1 \end{bmatrix} + \begin{bmatrix} \boldsymbol{I}_p \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{r},
\end{aligned}$$

or, equivalently,

$$\mathbf{0} = \boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{F}_1 \boldsymbol{x}_r - (\boldsymbol{C}_1 - \boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{F}_2)\, \boldsymbol{x}_1 + \boldsymbol{r}, \tag{10.64}$$

$$\mathbf{0} = -\boldsymbol{A}_{12} \boldsymbol{B}_u \boldsymbol{F}_1 \boldsymbol{x}_r + (\boldsymbol{A}_{11} - \boldsymbol{A}_{12} \boldsymbol{B}_u \boldsymbol{F}_2)\, \boldsymbol{x}_1. \tag{10.65}$$

In the steady-state, $\boldsymbol{z}_2 = \mathbf{0}$, that is,

$$\begin{aligned}
\tilde{\boldsymbol{S}} \tilde{\boldsymbol{x}} &= \begin{bmatrix} -\boldsymbol{S}_r & \boldsymbol{S}_1 & \boldsymbol{S}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{F}_1 & \boldsymbol{F}_2 & \boldsymbol{B}_u^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \\
&= \mathbf{0}.
\end{aligned}$$

Hence,

$$\boldsymbol{S}_r \boldsymbol{x}_r = \boldsymbol{S}_1 \boldsymbol{x}_1 + \boldsymbol{S}_2 \boldsymbol{x}_2. \tag{10.66}$$

Substituting $\boldsymbol{S}_2 = \boldsymbol{B}_u^{-1}$ and (10.66) into (10.64), we obtain

$$\begin{aligned}
\mathbf{0} &= -\boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{S}_r \boldsymbol{x}_r - \boldsymbol{C}_1 \boldsymbol{x}_1 + \boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{S}_1 \boldsymbol{x}_1 + \boldsymbol{r} \\
&= \boldsymbol{C}_2 \boldsymbol{B}_u (-\boldsymbol{S}_1 \boldsymbol{x}_1 - \boldsymbol{S}_2 \boldsymbol{x}_2) - \boldsymbol{C}_1 \boldsymbol{x}_1 + \boldsymbol{C}_2 \boldsymbol{B}_u \boldsymbol{S}_1 \boldsymbol{x}_1 + \boldsymbol{r} \\
&= -\boldsymbol{C}_2 \boldsymbol{x}_2 - \boldsymbol{C}_1 \boldsymbol{x}_1 + \boldsymbol{r} \\
&= -\boldsymbol{y} + \boldsymbol{r},
\end{aligned}$$

that is, $\boldsymbol{y}(\infty) = \boldsymbol{r}$, which completes the proof of the theorem.

□

10.7.5 Example: Two-link robotic manipulator

We synthesize a regulator for the two-link robotic manipulator depicted in Figure 2.22 in Section 2.4.6. Recall that the Lagrangian has the form,

$$\begin{aligned} L &= \frac{1}{2}m_1 l_1^2\omega_1^2 + \frac{1}{2}m_2\left(l_1^2\omega_1^2 + l_2^2\omega_2^2 + 2l_1l_2\cos\left(\theta_2-\theta_1\right)\omega_1\omega_2\right) \\ &\quad + m_1 g l_1\cos(\theta_1) + m_2 g\left(l_1\cos(\theta_1)+l_2\cos(\theta_2)\right), \end{aligned}$$

where $\omega_i = \dot{\theta}_i$, $i = 1, 2$, are the angular velocities of the links, and m_i and l_i are the link masses and their lengths, respectively. We assume that the input torques, τ_1 and τ_2 are applied at each joint. The parameter values are given in Table 10.1. The Lagrange equations of motion

Table 10.1: Parameter numerical values of the two-link robotic manipulator.

$m_1 = m_2$	1 kg
$l_1 = l_2$	0.5 m
g	9.81 m/sec^2

are,

$$\begin{aligned} &\begin{bmatrix} (m_1+m_2)l_1^2 & m_2l_1l_2\cos(\theta_2-\theta_1) \\ m_2l_1l_2\cos(\theta_2-\theta_1) & m_2l_2^2 \end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} \\ &\quad = \begin{bmatrix} \tau_1-\tau_2+m_2l_1l_2\dot{\theta}_2^2\sin(\theta_2-\theta_1)-(m_1+m_2)gl_1\sin(\theta_1) \\ \tau_2-m_2l_1l_2\dot{\theta}_1^2\sin(\theta_2-\theta_1)-m_2gl_2\sin(\theta_2)\end{bmatrix}. \end{aligned} \tag{10.67}$$

Note that the manipulator inertia matrix, $\boldsymbol{H}$, on the left-hand side of the above equation is symmetric and positive definite, and its determinant is

$$\det\boldsymbol{H} = (m_1+m_2)m_2l_1^2l_2^2 - m_2^2l_1^2l_2^2\cos^2(\theta_2-\theta_1) > 0.$$

We represent the above model in state-space format, where the state variables are defined as, $x_1 = \theta_1$, $x_2 = \theta_2$, $x_3 = \dot{\theta}_1$, and $x_4 = \dot{\theta}_2$. The outputs are

$$y_1 = x_1 \quad \text{and} \quad y_2 = x_2.$$

Therefore, $\boldsymbol{C}_1 = \boldsymbol{I}_2$ and $\boldsymbol{C}_2 = \boldsymbol{O}$. Using the above notation, we represent (10.67) as

$$\boldsymbol{H}(\boldsymbol{x}_1)\ddot{\boldsymbol{x}}_1 = -\boldsymbol{C}(\boldsymbol{x}_1,\dot{\boldsymbol{x}}_1)\dot{\boldsymbol{x}}_1 - \boldsymbol{g}(\boldsymbol{x}_1) + \boldsymbol{B}_u\boldsymbol{\tau},$$

where

$$\boldsymbol{B}_u = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix},$$

and

$$\boldsymbol{g}(\boldsymbol{x}_1) = \boldsymbol{G}(\boldsymbol{x}_1)\boldsymbol{x}_1 = \begin{bmatrix} -(m_1+m_2)gl_1\frac{\sin(x_1)}{x_1} & 0 \\ 0 & -m_2 g l_2 \frac{\sin(x_2)}{x_2} \end{bmatrix}\boldsymbol{x}_1.$$

We represent the control $\tilde{\boldsymbol{u}}$ as in (10.42), where

$$\tilde{\boldsymbol{u}} = \boldsymbol{H}^{-1}\boldsymbol{B}_u\boldsymbol{\tau}.$$

We select

$$\boldsymbol{T} = 0.05\boldsymbol{I}_2,$$

and

$$\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} = \{-3, -3.5, -4, -4.5\}.$$

Applying the design algorithm, we obtain

$$\tilde{\boldsymbol{S}} = \begin{bmatrix} -18. & 0 & 8.5 & 0 & 1 & 0 \\ 0 & -10.5 & 0 & 6.5 & 0 & 1 \end{bmatrix}.$$

With the above data, we construct the regulator (10.52). In our simulations, we select the initial conditions to be,

$$\boldsymbol{x}_r(0) = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{x}(0) = \begin{bmatrix} -3/4\pi & \pi & 0 & 0 \end{bmatrix}^\top.$$

The components of the reference signal vector $\boldsymbol{r}$ are plotted in Figures 10.12 and 10.13, where we show plots of link angles θ_1 and θ_2 versus time. In Figures 10.14 and 10.15, we show plots of the time history of the x and y coordinates, respectively, of the end effector of the robotic manipulator. In Figure 10.16, we plot the path of the end effector in the (x, y) plane. As can be seen from the figures the reference signal is tracked as predicted by our analysis.

10.8 State Estimation of Uncertain Systems

So far in our discussion of variable structure sliding mode control system design we assumed that the entire state vector of the plant to be controlled is available to us through measurement. If the state vector cannot be measured, then we can try to determine its suitable

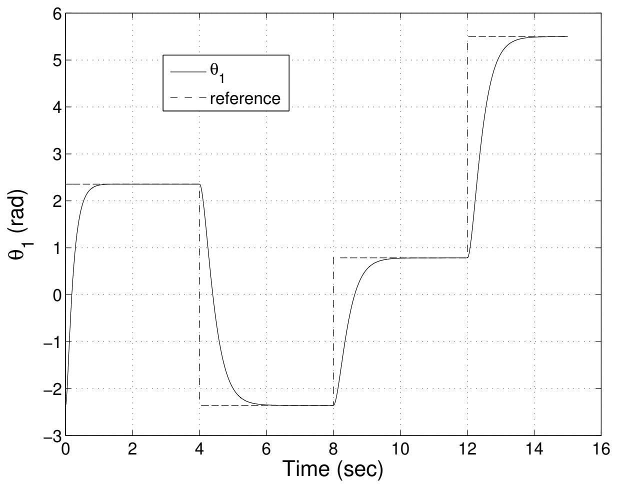


Figure 10.12: A plot of θ_1 and its reference versus time.

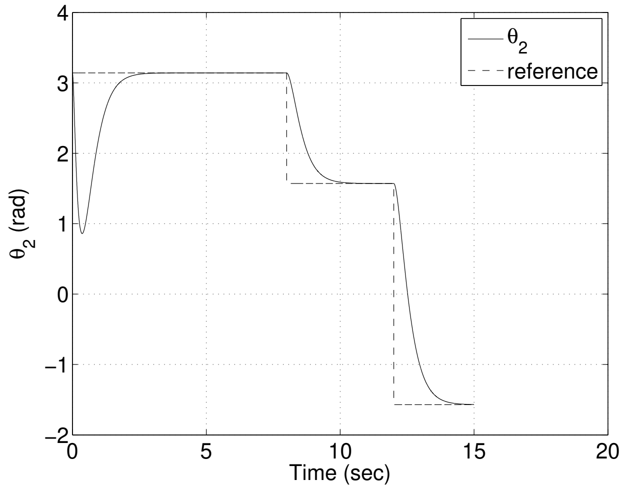


Figure 10.13: A plot of θ_2 and its reference versus time.

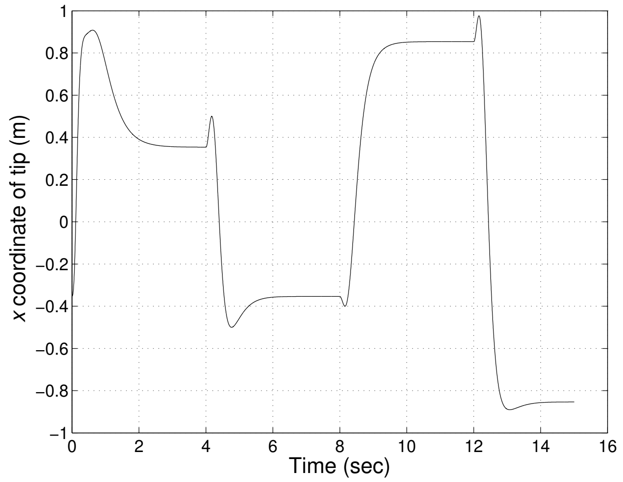


Figure 10.14: A plot of the x-coordinate of the end effector versus time.

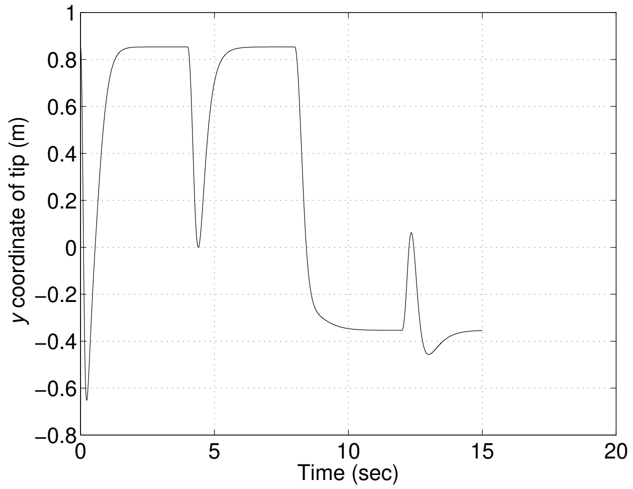


Figure 10.15: A plot of the y-coordinate of the end effector versus time.

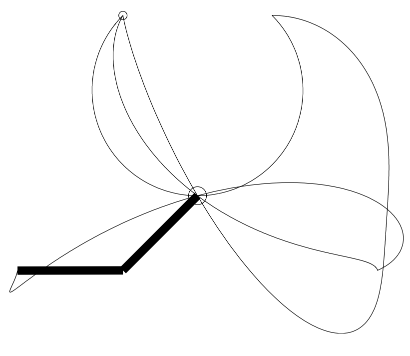


Figure 10.16: A plot of the path of the end effector in the (x, y)-plane.

estimate that can be used by the controller instead of the true state vector. If we use a state vector estimate for the inaccessible state, then we split a control design problem into three phases. The first phase is the design of a controller assuming the availability of the entire state vector of the plant. In the second phase, we construct an observer based on the plant input and output. Finally, in the third phase, we combine the controller and the observer into the combined controller-observer compensator, where the controller uses the plant state estimate instead of the actual state.

We discuss the problem of constructing a state estimator for a class of dynamical systems with matched uncertainties modeled by

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\left(\boldsymbol{u} + \boldsymbol{\xi}\right)\\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x},\end{aligned}$$

where $\boldsymbol{C} \in \mathbb{R}^{p\times n}$, and $p \geq m$, which means that the number of input channels does not exceed the number of output channels. The vector function $\boldsymbol{\xi}$ models the lumped uncertainties or nonlinearities in the plant. We assume that there is a nonnegative real-valued function $\rho = \rho\left(\boldsymbol{y}\right)$ such that

$$\|\boldsymbol{\xi}\| \leq \rho.$$

We assume that the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable. Our goal is to construct the plant state estimator. Let $\hat{\boldsymbol{x}}$ be an estimate of $\boldsymbol{x}$. Let $\boldsymbol{e}$ denote the estimation error, that is,

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t).$$

The observability of $(\boldsymbol{A}, \boldsymbol{C})$ implies the existence of a matrix $\boldsymbol{L} \in \mathbb{R}^{n\times p}$ such that $\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}$ has prescribed (symmetric with respect to the real axis) eigenvalues in the open left-half plane. Because $\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}$ is asymptotically stable, by the Lyapunov theorem, for any $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, there is a unique $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ such that

$$\left(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}\right)^\top \boldsymbol{P} + \boldsymbol{P}\left(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}\right) = -\boldsymbol{Q}. \tag{10.68}$$

We choose $\boldsymbol{Q}$, if possible, so that for some $\boldsymbol{F} \in \mathbb{R}^{m\times p}$,

$$\boldsymbol{F}\boldsymbol{C} = \boldsymbol{B}^\top \boldsymbol{P}. \tag{10.69}$$

We need this technical condition for the construction of the observer. This condition has an interesting system theoretic interpretation. Its satisfaction means that the transfer function matrix

$$\boldsymbol{G}(s) = \boldsymbol{F}\boldsymbol{C}\left(s\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{L}\boldsymbol{C}\right)^{-1}\boldsymbol{B}$$

is positive real, that is, the elements of $\boldsymbol{G}(s)$ are analytic in the open right-half plane and $\boldsymbol{G}(s) + \boldsymbol{G}^*(s) \geq 0$ in the open right-half plane. Here the superscript asterisk denotes the conjugate transpose.

An efficient method, using linear matrix inequalities (LMIs), for finding matrices $\boldsymbol{P}$, $\boldsymbol{L}$, and $\boldsymbol{F}$ that satisfy (10.68) and (10.69) is presented in Section 12.4.4.

To proceed, define the vector function

$$\boldsymbol{E}(\boldsymbol{e}, \eta) = \begin{cases} \dfrac{\boldsymbol{FCe}}{\|\boldsymbol{FCe}\|}\eta & \text{for} \quad \boldsymbol{FCe} \neq \boldsymbol{0} \\ \boldsymbol{0} & \text{for} \quad \boldsymbol{FCe} = \boldsymbol{0}, \end{cases}$$

where $\eta \geq 0$ is a design parameter. Note that if we have a single-input single-output plant, then we can write

$$\boldsymbol{E}(\boldsymbol{e}, \eta) = \eta \operatorname{sign}(\boldsymbol{FCe}).$$

We now introduce a sliding mode observer described in [339]. It has the form,

$$\boxed{\dot{\hat{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{LC})\hat{\boldsymbol{x}} + \boldsymbol{BE}(\boldsymbol{e}, \eta) + \boldsymbol{Ly} + \boldsymbol{Bu}} \tag{10.70}$$

If $\eta \geq \rho$, the function $\hat{\boldsymbol{x}}$ is an asymptotic estimate of the state $\boldsymbol{x}$, that is,

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \lim_{t\to\infty} (\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)) = \boldsymbol{0}.$$

To prove the above statement using Lyapunov's type of arguments, see Exercise 10.9, construct the differential equation satisfied by the estimation error $\boldsymbol{e}$,

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{LC})\boldsymbol{e} - \boldsymbol{BE}(\boldsymbol{e}, \eta) + \boldsymbol{B\xi}. \tag{10.71}$$

Then, show that

$$\frac{d}{dt}\left(\boldsymbol{e}^\top \boldsymbol{Pe}\right) = -\boldsymbol{e}^\top \boldsymbol{Qe} < 0,$$

which implies

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0}.$$

It follows from the above that the estimation error is insensitive to matched uncertainty.

A generalization of the sliding mode observer (10.71) is presented in Chapter 12.

10.9 Notes

The idea of controlling a dynamical system through varying its structure was originated in the 1950s. Early pioneers in the field of variable structure systems were S. V. Emelyanov and his co-workers in the Institute of Control Sciences in Moscow in the former Soviet Union. A large body of literature was produced by these researches, unfortunately mostly published in the Russian only. The first widely accessible books in English on variable structure sliding mode systems were books by Itkis [159] and Utkin [332]. A survey paper by Utkin [331] references many early contributions to the field; see also tutorials [77, 76] and a survey by Hung, Gao, and Hung [152]. At about the same time, A. F. Filippov was developing his theory of differential equations with discontinuous right-hand sides [102]. Filippov's approach is now accepted as the tool for the analysis of sliding mode systems. Sliding modes also found applications in the state estimation of dynamical uncertain systems as we show in Section 10.8. For further analysis of sliding mode observers, we refer the reader to Edwards and Spurgeon [90, 92]. For a stability analysis of uncertain closed-loop systems with the combined variable structure controller-observer compensators we refer to [147]. Output feedback sliding mode controllers are presented in [83, 149, 366, 367, 200, 91].

Impressive applications of sliding modes to solving optimization and computational problems are reported by Bhaya and Kaszkurewicz [33]. Ignaciuk and Bartoszewicz [156], on the other hand, apply discrete-time sliding-mode controllers to congestion control of communication networks.

10.10 Exercises

Exercise 10.1 Consider the following variable structure system that was analyzed in [152] (see also [331] for a similar system):

$$\left.\begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & -x_1 + 2x_2 + u, \end{array}\right\} \tag{10.72}$$

where

$$u = -4x_1 \operatorname{sign}\sigma(x_1, x_2), \quad \sigma(x_1, x_2) = x_1\left(0.5x_1 + x_2\right).$$

(i) Sketch phase plane portraits of two linear subsystems that constitute the given variable structure system (10.72).

(ii) Sketch a phase plane portrait of the variable structure system (10.72).

Exercise 10.2 (i) For the dynamical system model,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u,$$

construct the switching surface

$$\sigma(\boldsymbol{x}) = \boldsymbol{s}\boldsymbol{x} = 0$$

such that the system in sliding along the surface has its pole at -2.

(ii) Consider the following model of a dynamical system,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 2 & 0 & -1 \\ 0 & -2 & 2 \\ -1 & -2 & 3 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u.$$

Design a switching surface $\sigma(\boldsymbol{x}) = \boldsymbol{s}\boldsymbol{x} = 0$ such that the system restricted to it has poles at -4, -5.

(iii) Given the following dynamical system model,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 2 & 0 & -1 \\ 0 & -2 & 2 \\ -1 & -2 & 3 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u,$$

along with the switching surface

$$\sigma(\boldsymbol{x}) = \boldsymbol{s}\boldsymbol{x} = \begin{bmatrix} 8 & 8 & -15 \end{bmatrix} \boldsymbol{x} = 0.$$

Determine the reduced-order equations that describe the system behavior in a sliding mode.

Exercise 10.3 Consider a dynamical system modeled by the equations

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1(t) & a_2(t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u,$$

where

$$|a_1(t)| \leq 3, \quad |a_2(t)| \leq 4 \quad \text{for all} \quad t.$$

(i) Design a switching surface

$$\sigma(x_1, x_2) = s_1 x_1 + x_2 = 0$$

such that the system restricted to it has its pole at -2.

(ii) Design a sliding mode controller

$$u = k_1^{\pm} x_1 + k_2^{\pm} x_2,$$

so that the switching surface $\sigma = 0$ is globally attractive. The gains k_1 and k_2 are functions of $x_1\sigma$ and $x_2\sigma$, respectively.

Exercise 10.4 Consider the following model of an armature controlled DC motor controlling an inverted pendulum:

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} x_2 \\ 9.8 \sin x_1 + x_3 \\ -\frac{1}{L_a} x_2 - \frac{1}{L_a} x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} u \\
y &= x_1,
\end{aligned}$$

where L_a is the armature inductance of the motor.

(i) Derive the reduced-order model assuming L_a is negligible.

(ii) For the obtained reduced-order model from the first part, design the switching surface $\sigma = 0$ of the form

$$\sigma = \sigma(x_1, x_2) = \begin{bmatrix} s_1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

so that the system in sliding will have its pole at -3.

(iii) Use the above sliding surface to design a variable structure controller of the form

$$u = -k_1 x_1 \operatorname{sign}(x_1\sigma) - k_2 x_2 \operatorname{sign}(x_2\sigma).$$

The smaller the gains k_1, k_2, the better.

Exercise 10.5 Consider a dynamical system model,

$$\begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{A}_3 & A_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ b \end{bmatrix} u \tag{10.73}$$

$$y = \begin{bmatrix} \boldsymbol{C}_1 & C_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \end{bmatrix}, \tag{10.74}$$

where b and C_2 are nonzero scalars. Let

$$\sigma = \begin{bmatrix} \boldsymbol{S}_1 & S_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \end{bmatrix} = F \begin{bmatrix} \boldsymbol{C}_1 & C_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \end{bmatrix} = Fy, \tag{10.75}$$

where S_2 and F are nonzero scalars.

(i) Find the reduced-order system that models (10.73) when restricted to the surface $\sigma = 0$.

(ii) Find the transfer function $H(s) = \mathcal{L}(y(t))/\mathcal{L}(u(t)) = Y(s)/U(s)$. What is the relation between the zeros of $H(s)$ and the poles of the reduced-order system found in part (i)?

Exercise 10.6 (based on [200]) Prove the following

Lemma 10.2 *Let*

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u},$$

where the matrix $\boldsymbol{A}$ *has negative real distinct eigenvalues, that is,*

$$eig(\boldsymbol{A}) = \left\{ \; -\lambda_1, \quad -\lambda_2, \quad \ldots, \quad -\lambda_n \; \right\}, \qquad \lambda_i > 0, \qquad \lambda_i \neq \lambda_j \;\; \textit{for} \;\; i \neq j.$$

Let $\lambda_{\min} = \min \left\{ \; \lambda_1, \quad \lambda_2, \quad \ldots, \quad \lambda_n \; \right\}$ *and* K *be a positive constant such that*

$$\left\| e^{\boldsymbol{A}t} \right\| \leq K e^{-\lambda_{\min} t}$$

Then,

$$\|\boldsymbol{x}(t)\| \leq w(t),$$

where $w(t)$ *is a solution of*

$$\dot{w} = -\lambda_{\min} w + K \|\boldsymbol{B}\| \, \|\boldsymbol{u}\| \,, \quad \textit{subject to} \quad w(0) \geq K \|\boldsymbol{x}(0)\| \,.$$

Exercise 10.7 (based on [200]) Consider an n-th order model of an uncertain dynamical system,

$$\begin{aligned}
\begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{A}_3 & A_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ b \end{bmatrix} (u + \xi) \\
y &= \begin{bmatrix} \boldsymbol{C}_1 & C_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \end{bmatrix},
\end{aligned} \tag{10.76}$$

where b and C_2 are nonzero scalars and ξ models system's uncertainty, where

$$\xi = \boldsymbol{D}_1 \boldsymbol{x}_1 + D_2 x_2 + D_3 u + D_4(t).$$

In the above, $\boldsymbol{D}_1$ is a row vector whose size is equal to the dimension of $\boldsymbol{x}_1 \in \mathbb{R}^{n-1}$, D_2 and D_3 are scalars, where $|D_3| < 1$ and D_4 is an unknown bounded function of time whose bound is known to us, that is, $|D_4(t)| \le d_4$ where d_4 is known to us. Let $\sigma = Fy$ be as defined in Exercise 10.5. Assume that the switching surface,

$$\sigma = Fy = F \begin{bmatrix} \boldsymbol{C}_1 & C_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \end{bmatrix} = 0,$$

was chosen in such a way that the nominal system (10.76) restricted to it has the eigenvalues

$$\left\{ -\lambda_1, \quad -\lambda_2, \quad \ldots, \quad -\lambda_{n-1} \right\}, \quad \lambda_i > 0, \quad \lambda_i \neq \lambda_j \text{ for } i \neq j.$$

(i) Find $\dot{\sigma}$ evaluated on the trajectories of the uncertain system model (10.76).

(ii) For the controller

$$u = (S_2 b)^{-1} (1 - |D_3|)^{-1} (-Hw(t) - G|\sigma| - \delta) \operatorname{sign}(\sigma),$$

where $w(t)$ is a solution of a first-order differential equation such that $\|\boldsymbol{x}_1(t)\| \le w(t)$, find lower bounds for H, G and δ such that

$$\sigma\dot{\sigma} \le -\eta|\sigma|$$

for some $\eta > 0$. Use the lemma of Exercise 10.6 to find the form of the differential equation whose solution satisfies the condition $\|\boldsymbol{x}_1(t)\| \le w(t)$.

Exercise 10.8 For the dynamical system model

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} & = & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\,(u + f(y)) \\ y & = & \boldsymbol{c}\boldsymbol{x}, \end{array}\right\}$$

where

$$\boldsymbol{A} = \begin{bmatrix} -2 & 2 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad \text{and} \quad |f(y)| \le 10y,$$

design a sliding mode observer. The observer poles should be placed at -2 and -3. **Hint:** After constructing the required observer gain $\boldsymbol{l}$, find a symmetric positive definite matrix $\boldsymbol{Q}$ such that the solution, $\boldsymbol{P}$, to the Lyapunov matrix equation, $(\boldsymbol{A}-\boldsymbol{l}\boldsymbol{c})^\top\boldsymbol{P}+\boldsymbol{P}(\boldsymbol{A}-\boldsymbol{l}\boldsymbol{c}) = -\boldsymbol{Q}$, satisfies the condition, $\boldsymbol{b}^\top\boldsymbol{P} = \boldsymbol{c}$.

Exercise 10.9 Show that the estimation error $\boldsymbol{e}$ of the sliding mode observer (10.70) satisfies the equation

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\,\boldsymbol{e} - \boldsymbol{B}\boldsymbol{E}(\boldsymbol{e},\eta) + \boldsymbol{B}\boldsymbol{\xi}.$$

Then show that

$$\frac{d}{dt}\left(\boldsymbol{e}^\top\boldsymbol{P}\boldsymbol{e}\right) = -\boldsymbol{e}^\top\boldsymbol{Q}\boldsymbol{e} < 0,$$

which implies that

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0}.$$

Chapter 11

Vector Field Methods

> "I wrote this book without a book contract. I decided to just go write the book and then see if I could sell it. When you write a book without a book contract, you have to have a great deal of confidence in the topic and also a great deal of support at home." Johanna Neuman in [202, page 235]

In this chapter, we discuss methods that use vector fields in the controller construction for a class of nonlinear dynamical systems. The control design process, as in the case of linear systems, involves three steps. The first step is the design of a state-feedback control law. The second step involves the design of an observer. The third step is to combine the first two steps to obtain a combined controller-observer compensator. We use the so-called *feedback linearization* method to construct a state-feedback control strategy in the first stage of the above design algorithm. The basic idea of the feedback linearization approach is to use a control consisting of two components, one of which cancels out the plant's nonlinearities and the other controls the resulting linear system. The method is limited to a class of dynamical systems, plants, whose models are sufficiently smooth, that is, the plant models whose right-hand sides of the modeling differential equations are sufficiently many times differentiable. In the next section, we acquaint ourselves with some basic notions from differential geometry that we will be using in our further analysis. We then describe the class of plants for which the feedback linearization method is applicable.

11.1 Background Results From Differential Geometry

All functions that we encounter in this Chapter are assumed to have continuous partial derivatives to any order, that is, the functions that we work with are smooth.

Definition 11.1 *A vector-valued function of n variables,*

$$\boldsymbol{f} : \mathcal{D} \to \mathbb{R}^n, \quad \text{where} \quad \mathcal{D} \subseteq \mathbb{R}^n \quad \text{is a domain,}$$

is called a vector field on $\mathcal{D}$. A vector field is an n-dimensional column.

◇

Definition 11.2 *The transpose of a vector field is called a covector field. A covector field is an n-dimensional row.*

◇

Definition 11.3 *A differential of a real-valued function of n variables*

$$h : \mathcal{D} \to \mathbb{R}, \quad \text{where} \quad \mathcal{D} \subseteq \mathbb{R}^n \quad \text{is a domain,}$$

is a covector field, denoted dh, and defined as

$$dh = Dh = \nabla_{\boldsymbol{x}} h^\top = \begin{bmatrix} \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} & \cdots & \frac{\partial h}{\partial x_n} \end{bmatrix},$$

where $\nabla_{\boldsymbol{x}} h$ is the gradient of h, which is a vector field.

◇

In our controller construction for a class of nonlinear systems, we use three types of *Lie derivatives*. They are as follows:

1. Derivative of a vector field with respect to a vector field, also known as the *Lie bracket*.

   Given the vector valued functions $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$, where $\boldsymbol{f}$ and $\boldsymbol{g}$ are $\mathcal{C}^\infty$ vector fields. Their Lie bracket is defined as

   $$[\boldsymbol{f}, \boldsymbol{g}] = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \boldsymbol{g} - \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \boldsymbol{f}, \tag{11.1}$$

   where $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ and $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$ are the Jacobian matrices of $\boldsymbol{f}$ and $\boldsymbol{g}$, respectively. Sometimes in the literature the negative of the above definition is used. For convenience we also use an alternative notation of the form

   $$[\boldsymbol{f}, \boldsymbol{g}] = (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}). \tag{11.2}$$

   We define

   $$(\mathrm{ad}^k \boldsymbol{f}, \boldsymbol{g}) = [\boldsymbol{f}, (\mathrm{ad}^{k-1} \boldsymbol{f}, \boldsymbol{g})],$$

where

$$(\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) = \boldsymbol{g}.$$

For a single input linear system model, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{b}$. Hence,

$$\begin{aligned}
(\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) &= \boldsymbol{b} \\
(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) &= \boldsymbol{A}\boldsymbol{b} \\
&\vdots \\
(\mathrm{ad}^k \boldsymbol{f}, \boldsymbol{g}) &= \boldsymbol{A}^k\boldsymbol{b}.
\end{aligned}$$

Thus, the controllability matrix of a single-input linear system can be expressed in terms of Lie brackets as

$$\begin{bmatrix} \boldsymbol{b} & \boldsymbol{A}\boldsymbol{b} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b} \end{bmatrix} = \begin{bmatrix} (\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) & (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) & \cdots & (\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g}) \end{bmatrix}.$$

We call the $n \times n$ matrix

$$\begin{bmatrix} (\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) & (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) & \cdots & (\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g}) \end{bmatrix}, \tag{11.3}$$

the controllability matrix of the single-input nonlinear system modeled by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}. \tag{11.4}$$

2. Derivative of a function with respect to a vector field.

   Let $h : \mathbb{R}^n \to \mathbb{R}$ be a $\mathcal{C}^\infty$ function on $\mathbb{R}^n$. Let $Dh = \nabla h^\top$, where ∇h is the gradient (a column vector) of h with respect to $\boldsymbol{x}$. Then, the Lie derivative of the function h with respect to the vector field $\boldsymbol{f}$, denoted $L_{\boldsymbol{f}} h$ or $L_{\boldsymbol{f}}(h)$, is defined as

$$L_{\boldsymbol{f}} h = L_{\boldsymbol{f}}(h) = \langle \nabla h, \boldsymbol{f} \rangle = Dh\,\boldsymbol{f} = \frac{\partial h}{\partial x_1} f_1 + \frac{\partial h}{\partial x_2} f_2 + \cdots + \frac{\partial h}{\partial x_n} f_n.$$

   We use the following notation:

$$\begin{aligned}
L^0_{\boldsymbol{f}} h &= h \\
L^1_{\boldsymbol{f}} h &= L_{\boldsymbol{f}} h \\
L^2_{\boldsymbol{f}} h &= L_{\boldsymbol{f}} L_{\boldsymbol{f}} h \\
&\vdots \\
L^k_{\boldsymbol{f}} h &= L_{\boldsymbol{f}} L^{k-1}_{\boldsymbol{f}} h.
\end{aligned}$$

3. Derivative of Dh with respect to the vector field.

   The Lie derivative of Dh with respect to the vector field $\boldsymbol{f}$, denoted $L_{\boldsymbol{f}}(Dh)$, is defined as
$$L_{\boldsymbol{f}}(Dh) = \left(\frac{\partial \nabla h}{\partial x}\boldsymbol{f}\right)^{\top} + Dh\frac{\partial \boldsymbol{f}}{\partial x}. \tag{11.5}$$
   Note that
$$L_{\boldsymbol{f}}(Dh) = DL_{\boldsymbol{f}}h = \nabla L_{\boldsymbol{f}}^{\top}h. \tag{11.6}$$

The above three Lie derivatives satisfy the *Leibniz formula*,
$$\boxed{L_{[\boldsymbol{f},\boldsymbol{g}]}h = \langle \nabla h, [\boldsymbol{f},\boldsymbol{g}]\rangle = L_{\boldsymbol{g}}L_{\boldsymbol{f}}h - L_{\boldsymbol{f}}L_{\boldsymbol{g}}h} \tag{11.7}$$

11.2 Distributions

In this section, we present notions of a distribution and codistribution that we use in formulating sufficient conditions for a nonlinear model to be transformable into the controller form.

Definition 11.4 *Let $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_r$ be vector fields on $\mathcal{D} \subseteq \mathbb{R}^n$. Then, for any $\boldsymbol{x} \in \mathcal{D}$, $\boldsymbol{f}_1(\boldsymbol{x})$, $\boldsymbol{f}_2(\boldsymbol{x})$, ..., $\boldsymbol{f}_r(\boldsymbol{x})$ are vectors in $\mathbb{R}^n$ and*
$$\Delta(\boldsymbol{x}) = \mathit{span}\{\boldsymbol{f}_1(\boldsymbol{x}), \boldsymbol{f}_2(\boldsymbol{x}), \ldots, \boldsymbol{f}_r(\boldsymbol{x})\}$$
is a subspace of $\mathbb{R}^n$. To each point $\boldsymbol{x} \in \mathbb{R}^n$, there corresponds a subspace $\Delta(\boldsymbol{x})$. A distribution Δ is defined as
$$\Delta = \mathit{span}\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_r\}.$$
That is, Δ is the collection of all vector spaces $\Delta(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{D}$.

◇

The dimension of $\Delta(\boldsymbol{x})$ is
$$\dim(\Delta(\boldsymbol{x})) = \operatorname{rank}\begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{x}) & \boldsymbol{f}_2(\boldsymbol{x}) & \cdots & \boldsymbol{f}_r(\boldsymbol{x}) \end{bmatrix}.$$
This rank may vary with $\boldsymbol{x}$. If, however, the vectors $\boldsymbol{f}_1(\boldsymbol{x})$, $\boldsymbol{f}_2(\boldsymbol{x})$, ..., $\boldsymbol{f}_r(\boldsymbol{x})$ are linearly independent for all $\boldsymbol{x} \in \mathcal{D}$, then
$$\dim(\Delta(\boldsymbol{x})) = r \quad \text{for all} \quad \boldsymbol{x} \in \mathcal{D}. \tag{11.8}$$

Definition 11.5 *The distribution* $\Delta = span\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_r\}$ *is a nonsingular distribution on* $\mathcal{D}$ *if* (11.8) *is satisfied.*

◇

Example 11.1 The distribution,

$$\Delta = \text{span}\{\boldsymbol{f}_1, \boldsymbol{f}_2\},$$

where

$$\boldsymbol{f}_1(\boldsymbol{x}) = \begin{bmatrix} e^{x_2} \\ 1 \\ 0 \end{bmatrix} \text{ and } \boldsymbol{f}_2(\boldsymbol{x}) = \begin{bmatrix} 2 \\ 0 \\ x_2 \end{bmatrix}, \quad \boldsymbol{x} \in \mathbb{R}^3,$$

is nonsigular over $\mathcal{D} = \mathbb{R}^3$ because $\dim(\Delta(\boldsymbol{x})) = 2$ for all $\boldsymbol{x} \in \mathbb{R}^3$, that is,

$$\text{rank} \begin{bmatrix} e^{x_2} & 2 \\ 1 & 0 \\ 0 & x_2 \end{bmatrix} = 2 \quad \text{for all} \quad \boldsymbol{x} \in \mathbb{R}^3.$$

If Δ is a nonsingular distribution generated by $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_r$, then any vector field $\boldsymbol{g} \in \Delta$ can be expressed as

$$\boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{r} c_i(\boldsymbol{x}) \boldsymbol{f}_i(\boldsymbol{x}),$$

where $c_i(\cdot)$, $i = 1, \ldots, r$, are smooth functions defined on $\mathcal{D}$.

Definition 11.6 *A distribution* Δ *is involutive if*

$$\boldsymbol{g}_1 \in \Delta \quad \textit{and} \quad \boldsymbol{g}_2 \in \Delta \quad \textit{then} \quad [\boldsymbol{g}_1, \boldsymbol{g}_2] \in \Delta$$

◇

Theorem 11.1 *If* $\Delta = span\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_k\}$ *is a nonsingular distribution on* $\mathcal{D}$*, then it is involutive if and only if*

$$[\boldsymbol{f}_i, \boldsymbol{f}_j] \in \Delta \quad \textit{for all} \quad 1 \le i, j \le r$$

Example 11.2 (Kaczorek et al.) [166, p. 448]

Consider the nonsingular distribution of Example 11.1. This distribution is not involutive because

$$[\boldsymbol{f}_1, \boldsymbol{f}_2] \notin \operatorname{span}\{\boldsymbol{f}_1, \boldsymbol{f}_2\},$$

that is,

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 \end{bmatrix} \neq \operatorname{rank}\begin{bmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 & [\boldsymbol{f}_1, \boldsymbol{f}_2] \end{bmatrix}.$$

Indeed, the Lie bracket

$$\begin{aligned}
[\boldsymbol{f}_1, \boldsymbol{f}_2] &= \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}}\boldsymbol{f}_2 - \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{x}}\boldsymbol{f}_1 \\
&= \begin{bmatrix} 0 & e^{x_2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \\ x_2 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \\ x_2 \end{bmatrix}\begin{bmatrix} e^{x_2} \\ 1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \text{for all} \quad \boldsymbol{x} \in \mathbb{R}^3.
\end{aligned}$$

Therefore

$$\operatorname{rank}\begin{bmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 & [\boldsymbol{f}_1, \boldsymbol{f}_2] \end{bmatrix} = \operatorname{rank}\begin{bmatrix} e^{x_2} & 2 & 0 \\ 1 & 0 & 0 \\ 0 & x_2 & -1 \end{bmatrix} = 3$$

while $\operatorname{rank}\begin{bmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 \end{bmatrix} = 2.$

We now introduce the notion of a codistribution. These objects are dual of distributions and are defined using covector fields. The properties of codistributions are similar to that of distributions, except that now we work with rows rather than columns. One type of codistribution is of special interest to us; it is the annihilator of a distribution Δ denoted $\Delta^{\perp}$ and defined as

Definition 11.7 $\Delta^{\perp} = \{\boldsymbol{w} : \boldsymbol{w}\boldsymbol{v} = 0, \quad \textit{for all} \quad \boldsymbol{v} \in \Delta(\boldsymbol{x})\}$

◇

We also need the notion of complete integrability of a distribution.

Definition 11.8 *A nonsingular distribution* $\Delta = span\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_r\}$ *on* $\mathcal{D}$ *is completely integrable if for each* $\boldsymbol{x}_0 \in \mathcal{D}$ *there exists a neighborhood* $\mathcal{N}$ *of* $\boldsymbol{x}_0$ *and* $n-r$ *real-valued smooth functions* $h_1(\cdot), \ldots, h_{n-r}(\cdot)$ *that satisfy the conditions*

$$dh_j \begin{bmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 & \cdots & \boldsymbol{f}_r \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}, \quad 1 \le j \le n-r$$

and

$$\Delta^{\perp} = span\{dh_1, \ldots, dh_{n-r}\},$$

that is, the covector fields dh_j, $j = 1, \ldots, n-r$, *are linearly independent for all* $\boldsymbol{x} \in \mathcal{D}$.

◇

With the above background, we can state the theorem of Frobenius that gives a necessary and sufficient condition for a nonsingular distribution to be involutive.

Theorem 11.2 (Frobenius) *A nonsingular distribution is completely integrable if and only if it is involutive.*

11.3 A Nonlinear Plant Model

Dynamical systems, that we consider in this chapter, are modeled by the equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \tag{11.9}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}), \tag{11.10}$$

where $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\boldsymbol{G} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$, and the output map $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^p$. Note that $\boldsymbol{G}(\boldsymbol{x})$ may be written as $\boldsymbol{G}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{g}_1(\boldsymbol{x}) & \boldsymbol{g}_2(\boldsymbol{x}) & \cdots & \boldsymbol{g}_m(\boldsymbol{x}) \end{bmatrix}$, where $\boldsymbol{g}_i : \mathbb{R}^n \to \mathbb{R}^n$, $i = 1, 2, \ldots, m$. We write $\boldsymbol{f} \in \mathcal{C}^\infty$ if $\boldsymbol{f}$ is in all $\mathcal{C}^k$ for $k = 1, 2, \ldots$. The functions $\boldsymbol{f}$, $\boldsymbol{g}_i$, and $\boldsymbol{h}$ are assumed to be $\mathcal{C}^\infty$ vector fields on $\mathbb{R}^n$. We also assume that $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$ and $\boldsymbol{h}(\boldsymbol{0}) = \boldsymbol{0}$. We note that the linear system model $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$, $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$ is a special case of the above more general model. Indeed, we have $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, $\boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{B}$, and $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{x}$. A simple example of a nonlinear dynamical system whose model fits into the above class of systems is a model of one-link robot manipulator presented in Subsection 2.4.5.

11.4 Controller Form

One of the objectives of this chapter, is to devise a method for constructing state-feedback stabilizing controllers for a class of dynamical systems modeled by (11.9). To construct such controllers it is advantageous to work with an equivalent system model rather than with the original one. We first discuss a method of transforming a single-input nonlinear system model into an equivalent form that is a generalization of the controller form known from linear systems—see Section 5.4.1. We accomplish this using a transformation, described below, that is a generalization of the similarity transformation for single-input linear systems given by (5.27) on page 241.

A transformation $\boldsymbol{T} : \mathcal{D} \to \mathbb{R}^n$ is a *diffeomorphism* on $\mathcal{D}$ if it is invertible on $\mathcal{D}$, that is, there exists a function $\boldsymbol{T}^{-1}$ such that

$$\boldsymbol{T}^{-1}(\boldsymbol{T}(\boldsymbol{x})) = \boldsymbol{x} \quad \text{for all} \quad \boldsymbol{x} \in \mathcal{D},$$

and both $\boldsymbol{T}$ and $\boldsymbol{T}^{-1}$ are continuously differentiable

If the Jacobian matrix $\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}}$ is nonsingular at a point $\boldsymbol{x}_0 \in \mathcal{D}$, then it follows form the inverse function theorem, see Exercise 11.3, that there is a neighborhood $\mathcal{N}$ of $\boldsymbol{x}_0$ such that $\boldsymbol{T}$ restricted to $\mathcal{N}$ is a diffeomorphism on $\mathcal{N}$.

A transformation $\boldsymbol{T}$ is a *global diffeomorphism* if it is a diffeomorphism on $\mathbb{R}^n$ and

$$\boldsymbol{T}(\mathbb{R}^n) = \mathbb{R}^n.$$

We now present necessary and sufficient conditions for a transformation $\boldsymbol{T}$ to be a global diffeomorphism. This theorem comes from Khalil [178, p. 561].

Theorem 11.3 *A transformation $\boldsymbol{T}$ is a global diffeomorphism if and only if*

1. *the Jacobian matrix $\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}}$ is nonsingular for all $\boldsymbol{x} \in \mathbb{R}^n$;*

2. *$\boldsymbol{T}$ is proper, that is,*

$$\lim_{\|\boldsymbol{x}\| \to \infty} \|\boldsymbol{T}(\boldsymbol{x})\| = \infty.$$

Our goal is to construct a diffeomorphism such that

$$\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}), \quad \boldsymbol{T}(\boldsymbol{0}) = \boldsymbol{0},$$

with an inverse $\boldsymbol{x} = \boldsymbol{T}^{-1}(\boldsymbol{z})$, such that system model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u$ in the new coordinates has the form

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_{n-1} \\ \dot{z}_n \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ \vdots \\ z_n \\ \tilde{f}_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \tilde{g}_n \end{bmatrix} u, \tag{11.11}$$

where $\tilde{f}_n = \tilde{f}_n(z_1, z_2, \ldots, z_n)$, $\tilde{g}_n = \tilde{g}_n(z_1, z_2, \ldots, z_n)$, and $\tilde{g}_n \neq 0$. The above form is a generalization of the controller form for linear system from Section 5.4.1.

We devise a method for constructing the desired transformation. In our subsequent discussion, we will be using three types of Lie derivatives defined in Section 11.1. Suppose that such a transformation exists. Differentiating $\boldsymbol{z}(t)$ with respect to time t gives

$$\dot{\boldsymbol{z}}(t) = \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \left(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \right). \tag{11.12}$$

Comparing the right hand sides of (11.11) and (11.12), we obtain

$$\left. \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \right|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} = \begin{bmatrix} z_2 \\ z_3 \\ \vdots \\ z_n \\ \tilde{f}_n(z_1, z_2, \ldots, z_n) \end{bmatrix}, \tag{11.13}$$

and

$$\left. \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x}) \right|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \tilde{g}_n(z_1, z_2, \ldots, z_n) \end{bmatrix}. \tag{11.14}$$

We first consider (11.13), that is,

$$\begin{bmatrix} z_2 \\ z_3 \\ \vdots \\ z_n \\ \tilde{f}_n(z_1, z_2, \ldots, z_n) \end{bmatrix} = \begin{bmatrix} T_2 \\ T_3 \\ \vdots \\ T_n \\ \tilde{f}_n(z_1, z_2, \ldots, z_n) \end{bmatrix}$$

because $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$. Comparing the above with the left-hand side of (11.13) gives

$$\frac{\partial T_i}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = T_{i+1}, \quad i = 1, 2, \ldots, n-1,$$

or equivalently

$$T_{i+1} = L_{\boldsymbol{f}} T_i, \quad i = 1, 2, \ldots, n-1.$$

Therefore, the transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ has the form

$$\boldsymbol{T} = \begin{bmatrix} T_1 \\ L_{\boldsymbol{f}} T_1 \\ \vdots \\ L_{\boldsymbol{f}}^{n-2} T_1 \\ L_{\boldsymbol{f}}^{n-1} T_1 \end{bmatrix}. \tag{11.15}$$

The above means that the problem of constructing the desired transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ is reduced to finding its first component T_1. The remaining components of $\boldsymbol{T}$ can be successively computed using T_1. To specify T_1, we use (11.14), which we represent componentwise as

$$L_{\boldsymbol{g}} T_i = 0, \quad i = 1, 2, \ldots, n-1,$$

and

$$L_{\boldsymbol{g}} T_n = \tilde{g}_n.$$

Note that $L_{\boldsymbol{g}} T_1 = 0$ can be represented as

$$\frac{\partial T_1}{\partial \boldsymbol{x}} (\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) = 0.$$

Next, consider the relation $L_{\boldsymbol{g}} T_2 = 0$. We will show that

$$L_{\boldsymbol{g}} T_2 = \frac{\partial T_1}{\partial \boldsymbol{x}} (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) = 0. \tag{11.16}$$

By the Leibniz formula,

$$\begin{aligned} L_{\boldsymbol{g}} T_2 &= L_{\boldsymbol{g}} L_{\boldsymbol{f}} T_1 \\ &= L_{\boldsymbol{f}} L_{\boldsymbol{g}} T_1 + L_{[\boldsymbol{f}, \boldsymbol{g}]} T_1 \\ &= L_{[\boldsymbol{f}, \boldsymbol{g}]} T_1, \end{aligned} \tag{11.17}$$

because $L_{\boldsymbol{g}} T_1 = 0$, and hence $L_{\boldsymbol{f}} L_{\boldsymbol{g}} T_1 = 0$. Using (11.17), we obtain (11.16). Continuing in this manner, we get

$$\frac{\partial T_1}{\partial \boldsymbol{x}} (\mathrm{ad}^i \boldsymbol{f}, \boldsymbol{g}) = 0, \quad i = 0, 1, \ldots, n-2,$$

and

$$\frac{\partial T_1}{\partial \boldsymbol{x}}(\mathrm{ad}^{n-1}\boldsymbol{f}, \boldsymbol{g}) = \tilde{g}_n.$$

We represent the above n equations in matrix form as

$$\begin{aligned} & \frac{\partial T_1}{\partial \boldsymbol{x}} \left[\begin{array}{cccc} (\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) & (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) & \cdots & (\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g}) \end{array} \right] \\ = & \left[\begin{array}{cccc} L_{\boldsymbol{g}} T_1 & L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} T_1 & \cdots & L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} T_1 \end{array} \right] \\ = & \ \tilde{g}_n \left[\begin{array}{cccc} 0 & 0 & \cdots & 1 \end{array} \right]. \end{aligned}$$

Thus the derivative of T_1, that is, the row vector $\frac{\partial T_1}{\partial \boldsymbol{x}}$, is the last row of the inverse of the controllability matrix multiplied by $\tilde{g}_n$, provided the controllability matrix is invertible. Let $\boldsymbol{q}(\boldsymbol{x})$ denote the last row of the inverse of the controllability matrix multiplied by $\tilde{g}_n$. Then, the problem we have to solve is to find $T_1 : \mathbb{R}^n \to \mathbb{R}$ such that

$$\frac{\partial T_1(\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{q}(\boldsymbol{x}), \quad T_1(\mathbf{0}) = 0, \tag{11.18}$$

or equivalently

$$\nabla T_1 = \boldsymbol{q}(\boldsymbol{x})^\top, \quad T_1(\mathbf{0}) = 0. \tag{11.19}$$

The above problem can be viewed as finding a potential function T_1 for a given vector field $\boldsymbol{q}^\top$. A necessary and sufficient condition for a given vector field to have a potential function is given by Theorem A.21 on page 1026. The function $\tilde{g}_n$ is also referred to as an integrating factor.

We will show that if $\boldsymbol{q}^\top$ has a potential function, then the resulting state variable transformation is a diffeomorphism. Before we proceed with the proof, we first illustrate the results obtained so far with a numerical example.

Example 11.3 Consider a nonlinear model of one-link robot manipulator presented in Subsection 2.4.5. The controllability matrix of this system model is

$$\left[\begin{array}{ccc} (\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) & (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) & (\mathrm{ad}^2 \boldsymbol{f}, \boldsymbol{g}) \end{array} \right] = \left[\begin{array}{ccc} 0 & 0 & 10 \\ 0 & 10 & -100 \\ 10 & -100 & 900 \end{array} \right].$$

The above controllability matrix is of full rank on $\mathbb{R}^3$. The last row of its inverse is

$$\left[\begin{array}{ccc} 0.1 & 0 & 0 \end{array} \right].$$

Suppose that we take $\tilde{g}_n = 1$; then $\boldsymbol{q} = \begin{bmatrix} 0.1 & 0 & 0 \end{bmatrix}$, and hence we can take,

$$T_1 = 0.1\, x_1.$$

Having obtained T_1, we construct the desired transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$, where

$$\boldsymbol{T}(\boldsymbol{x}) = \begin{bmatrix} T_1 \\ L_{\boldsymbol{f}} T_1 \\ L_{\boldsymbol{f}}\left(L_{\boldsymbol{f}} T_1\right) \end{bmatrix} = \begin{bmatrix} 0.1\, x_1 \\ 0.1\, x_2 \\ 0.98 \sin x_1 + 0.1 x_3 \end{bmatrix}.$$

Note that the inverse transformation $\boldsymbol{x} = \boldsymbol{T}^{-1}(\boldsymbol{z})$ exists and has the form

$$\boldsymbol{T}^{-1}(\boldsymbol{z}) = \begin{bmatrix} 10\, z_1 \\ 10\, z_2 \\ 10 z_3 - 9.8 \sin(10\, z_1) \end{bmatrix}.$$

Furthermore,

$$\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0.98 \cos x_1 & 0 & 0.1 \end{bmatrix}.$$

Applying the transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ to the model of the one-link robot manipulator yields

$$\begin{aligned} \dot{\boldsymbol{z}} &= \left.\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})\right|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} + \left.\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x})\right|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} u \\ &= \begin{bmatrix} z_2 \\ z_3 \\ 9.8 \sin(10 z_1) + (9.8 \cos(10 z_1) - 10)\, z_2 - 10 z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u. \end{aligned} \tag{11.20}$$

Our objective now is to show that if there is a potential function for $\boldsymbol{q}(\boldsymbol{x})^\top$, then the resulting transformation is indeed a diffeomorphism. By construction, $\boldsymbol{T}$ is of $\mathcal{C}^\infty$ class, and $\boldsymbol{T}(\boldsymbol{0}) = \boldsymbol{0}$. We show that $\boldsymbol{T}$ is locally $\mathcal{C}^\infty$-invertible. By the inverse function theorem, stated in Exercise 11.3, if the Jacobian matrix of $\boldsymbol{T}$ evaluated at $\boldsymbol{0}$ is invertible, then $\boldsymbol{T}$ is locally $\mathcal{C}^\infty$-invertible. The Jacobian matrix of $\boldsymbol{T}$ is

$$D\boldsymbol{T} = \begin{bmatrix} DT_1 \\ DL_{\boldsymbol{f}} T_1 \\ \vdots \\ DL_{\boldsymbol{f}}^{n-2} T_1 \\ DL_{\boldsymbol{f}}^{n-1} T_1 \end{bmatrix}.$$

We compute the product of the above Jacobian matrix and the controllability matrix

$$\left[\begin{array}{cccc} (\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) & (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) & \cdots & (\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g}) \end{array}\right]$$

to get

$$\begin{bmatrix} DT_1 \\ DL_{\boldsymbol{f}} T_1 \\ \vdots \\ DL_{\boldsymbol{f}}^{n-2} T_1 \\ DL_{\boldsymbol{f}}^{n-1} T_1 \end{bmatrix} \left[\begin{array}{cccc} (\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) & (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) & \cdots & (\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g}) \end{array}\right].$$

Performing multiplications yields

$$\begin{aligned}
&D\boldsymbol{T} \left[\begin{array}{cccc} (\mathrm{ad}^0 \boldsymbol{f}, \boldsymbol{g}) & (\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}) & \cdots & (\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g}) \end{array}\right] \\
&= \begin{bmatrix} L_{\boldsymbol{g}} T_1 & L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} T_1 & \cdots & L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} T_1 \\ L_{\boldsymbol{g}} L_{\boldsymbol{f}} T_1 & L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} L_{\boldsymbol{f}} T_1 & \cdots & L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} L_{\boldsymbol{f}} T_1 \\ \vdots & \vdots & & \vdots \\ L_{\boldsymbol{g}} L_{\boldsymbol{f}}^{n-1} T_1 & L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} L_{\boldsymbol{f}}^{n-1} T_1 & \cdots & L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} L_{\boldsymbol{f}}^{n-1} T_1 \end{bmatrix} \\
&= \begin{bmatrix} L_{\boldsymbol{g}} T_1 & L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} T_1 & \cdots & L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} T_1 \\ L_{\boldsymbol{g}} T_2 & L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} T_2 & \cdots & L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} T_2 \\ \vdots & \vdots & & \vdots \\ L_{\boldsymbol{g}} T_n & L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} T_n & \cdots & L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} T_n \end{bmatrix} \\
&= \begin{bmatrix} L_{\boldsymbol{g}} T_1 & L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} T_1 & \cdots & L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} T_1 \\ L_{(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g})} T_1 & L_{(\mathrm{ad}^2 \boldsymbol{f}, \boldsymbol{g})} T_1 & \cdots & L_{(\mathrm{ad}^{n} \boldsymbol{f}, \boldsymbol{g})} T_1 \\ \vdots & \vdots & & \vdots \\ L_{(\mathrm{ad}^{n-1} \boldsymbol{f}, \boldsymbol{g})} T_1 & L_{(\mathrm{ad}^n \boldsymbol{f}, \boldsymbol{g})} T_1 & \cdots & L_{(\mathrm{ad}^{2n-2} \boldsymbol{f}, \boldsymbol{g})} T_1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & \cdots & 0 & \tilde{g}_n \\ 0 & 0 & \cdots & \tilde{g}_n & x \\ \vdots & \vdots & & \vdots & \vdots \\ \tilde{g}_n & x & \cdots & x & x \end{bmatrix},
\end{aligned}$$

where we use x to denote "don't care" elements and $\tilde{g}_n \neq 0$. It follows from the above that at $\boldsymbol{x} = \mathbf{0}$ the Jacobian matrix $D\boldsymbol{T}$ is invertible, that is,

$$\operatorname{rank} D\boldsymbol{T}(\mathbf{0}) = n.$$

We summarize the above analysis in the form of the following theorem,

Theorem 11.4 *There exists a diffeomorphic state variable transformation, in a neighborhood of $\boldsymbol{x} = \boldsymbol{0}$, if*

1. *in a neighborhood of $\boldsymbol{x} = \boldsymbol{0}$,*

$$\text{rank}\begin{bmatrix} (ad^0\boldsymbol{f},\boldsymbol{g}) & (ad^1\boldsymbol{f},\boldsymbol{g}) & \cdots & (ad^{n-1}\boldsymbol{f},\boldsymbol{g}) \end{bmatrix} = n,$$

2.

$$\boldsymbol{q}(\boldsymbol{x}) = \tilde{g}_n \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} (ad^0\boldsymbol{f},\boldsymbol{g}) & (ad^1\boldsymbol{f},\boldsymbol{g}) & \cdots & (ad^{n-1}\boldsymbol{f},\boldsymbol{g}) \end{bmatrix}^{-1}$$

has a potential function in a neighborhood of $\boldsymbol{x} = \boldsymbol{0}$ for some $\tilde{g}_n \neq 0$, that is, $\boldsymbol{q}(\boldsymbol{x})^\top$ is a conservative vector field in a neighborhood of $\boldsymbol{x} = \boldsymbol{0}$.

We reformulate the above conditions for the existence of the diffeomorphism transforming the given nonlinear system into the controller form using the notion of distributions that we introduced in Section 11.2.

Theorem 11.5 *For the single-input model,*

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}$$

there exists a diffeomorphism, $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$, transforming it into the controller form (11.11)*, in a neighborhood of $\boldsymbol{x} = \boldsymbol{0}$, if*

1. *in a neighborhood of $\boldsymbol{x} = \boldsymbol{0}$,*

$$\text{rank}\begin{bmatrix} (ad^0\boldsymbol{f},\boldsymbol{g}) & (ad^1\boldsymbol{f},\boldsymbol{g}) & \cdots & (ad^{n-1}\boldsymbol{f},\boldsymbol{g}) \end{bmatrix} = n;$$

2. *the vector fields*

$$(ad^0\boldsymbol{f},\boldsymbol{g}), (ad^1\boldsymbol{f},\boldsymbol{g}), \ldots, (ad^{n-2}\boldsymbol{f},\boldsymbol{g})$$

generate, in the neighborhood of $\boldsymbol{x} = \boldsymbol{0}$, a singular involutive distribution.

The above conditions are also necessary—see Khalil [178, pp. 567–568] for a proof of both, necessary and sufficient parts, using the Frobenius theorem.

The above method of transforming a class of single input nonlinear systems into the controller form can be generalized, with some effort, to multi-input systems. We illustrate the method with a numerical example.

Example 11.4 Consider the model of the permanent stepper motor given by (2.61). We first represent the model as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_1 u_1 + \boldsymbol{g}_2 u_2, \tag{11.21}$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x_2 \\ -K_1 x_2 - K_2 x_3 \sin(K_3 x_1) + K_2 x_4 \cos(K_3 x_1) \\ K_4 x_2 \sin(K_3 x_1) - K_5 x_3 \\ -K_4 x_2 \cos(K_3 x_1) - K_5 x_4 \end{bmatrix}, \ \boldsymbol{g}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \text{ and } \boldsymbol{g}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We form the controllability matrix for the above model,

$$\begin{bmatrix} \boldsymbol{g}_1 & \boldsymbol{g}_2 & \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_1\right) & \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_2\right) & \left(\mathrm{ad}^2 \boldsymbol{f}, \boldsymbol{g}_1\right) & \left(\mathrm{ad}^2 \boldsymbol{f}, \boldsymbol{g}_2\right) & \left(\mathrm{ad}^3 \boldsymbol{f}, \boldsymbol{g}_1\right) & \left(\mathrm{ad}^3 \boldsymbol{f}, \boldsymbol{g}_2\right) \end{bmatrix}, \tag{11.22}$$

where, for example,

$$\left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_1\right) = \begin{bmatrix} 0 \\ -K_2 \sin(K_3 x_1) \\ -K_5 \\ 0 \end{bmatrix}, \quad \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_2\right) = \begin{bmatrix} 0 \\ K_2 \cos(K_3 x_1) \\ 0 \\ -K_5 \end{bmatrix},$$

$$\begin{aligned} &\left(\mathrm{ad}^2 \boldsymbol{f}, \boldsymbol{g}_1\right) \\ &= \left[\boldsymbol{f}, \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_1\right)\right] = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_1\right) - \frac{\partial \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_1\right)}{\partial \boldsymbol{x}} \boldsymbol{f} \\ &= \begin{bmatrix} -K_2 \sin(K_3 x_1) \\ K_1 K_2 \sin(K_3 x_1) + K_2 K_5 \sin(K_3 x_1) + K_2 K_3 x_2 \cos(K_3 x_1) \\ -K_4 K_2 \sin^2(K_3 x_1) + {K_5}^2 \\ K_4 K_2 \cos(K_3 x_1) \sin(K_3 x_1) \end{bmatrix}, \end{aligned} \tag{11.23}$$

and

$$\begin{aligned} &\left(\mathrm{ad}^2 \boldsymbol{f}, \boldsymbol{g}_2\right) \\ &= \left[\boldsymbol{f}, \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_2\right)\right] = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_2\right) - \frac{\partial \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_2\right)}{\partial \boldsymbol{x}} \boldsymbol{f} \\ &= \begin{bmatrix} K_2 \cos(K_3 x_1) \\ -K_1 K_2 \cos(K_3 x_1) - K_2 K_5 \cos(K_3 x_1) + K_2 K_3 x_2 \sin(K_3 x_1) \\ K_4 K_2 \cos(K_3 x_1) \sin(K_3 x_1) \\ -K_4 K_2 \cos^2(K_3 x_1) + {K_5}^2 \end{bmatrix}. \end{aligned} \tag{11.24}$$

Observe that

$$\left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_2\right) = -K_5 \cot(K_3\, x_1)\, \boldsymbol{g}_1 - K_5\, \boldsymbol{g}_2 - \cot(K_3\, x_1) \left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_1\right), \tag{11.25}$$

that is, $\left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_2\right)$ is a linear combination of $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, and $\left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_1\right)$. We use this fact when selecting columns to construct a transformation into a controller form. Proceeding from left to right and using (11.25), we select linearly independent columns from the controllability matrix (11.22). The PM stepper motor system is controllable because we have found four linearly independent columns, $\boldsymbol{g}_1$, $\left(\mathrm{ad}^1 \boldsymbol{f}, \boldsymbol{g}_1\right)$, $\left(\mathrm{ad}^2 \boldsymbol{f}, \boldsymbol{g}_1\right)$, and $\boldsymbol{g}_2$ in its controllability matrix. Thus, the controllability indices of the PM stepper motor system are $d_1 = 3$ and $d_2 = 1$. Therefore, the controller form of the PM stepper motor model is

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ \tilde{f}_3(z_1, z_2, z_3, z_4) \\ \tilde{f}_4(z_1, z_2, z_3, z_4) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \tilde{g}_{31} & \tilde{g}_{32} \\ \tilde{g}_{41} & \tilde{g}_{42} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{11.26}$$

where the elements $\tilde{g}_{ij}$ may depend on z_k, $k = 1, 2, 3, 4$, and

$$\det \begin{bmatrix} \tilde{g}_{31} & \tilde{g}_{32} \\ \tilde{g}_{41} & \tilde{g}_{42} \end{bmatrix} \neq 0. \tag{11.27}$$

The transformation we are looking for has the form,

$$\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}) = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} T_1 \\ L_{\boldsymbol{f}} T_1 \\ L_{\boldsymbol{f}}^2 T_1 \\ T_4 \end{bmatrix}, \tag{11.28}$$

where T_1 and T_4 are functions that we have to find. There is no unique solution for T_1 and T_4. Differentiating $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ and using (11.21) yields

$$\begin{aligned} \dot{\boldsymbol{z}} &= \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} \\ &= \left.\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})\right|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} + u_1 \left.\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \boldsymbol{g}_1(\boldsymbol{x})\right|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} + u_2 \left.\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} \boldsymbol{g}_2(\boldsymbol{x})\right|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} \end{aligned} \tag{11.29}$$

where

$$\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}} = D\boldsymbol{T} = \begin{bmatrix} DT_1 \\ DT_2 \\ DT_3 \\ DT_4 \end{bmatrix}.$$

Comparing (11.26) and (11.29), we obtain

$$z_2 = T_2 = DT_1\boldsymbol{f} = L_{\boldsymbol{f}}T_1, \quad z_3 = T_3 = DT_2\boldsymbol{f} = L_{\boldsymbol{f}}T_2, \tag{11.30}$$

and

$$L_{\boldsymbol{g}_1}T_1 = 0, \quad L_{\boldsymbol{g}_1}T_2 = L_{\boldsymbol{g}_1}L_{\boldsymbol{f}}T_1 = 0, \tag{11.31}$$

$$L_{\boldsymbol{g}_2}T_1 = 0, \quad L_{\boldsymbol{g}_2}T_2 = L_{\boldsymbol{g}_2}L_{\boldsymbol{f}}T_1 = 0. \tag{11.32}$$

Applying the Leibniz formula (11.7) to the right hand sides of (11.31) and (11.32) gives

$$-L_{\boldsymbol{f}}L_{\boldsymbol{g}_1}T_1 = L_{[\boldsymbol{f},\boldsymbol{g}_1]}T_1 \quad \text{and} \quad -L_{\boldsymbol{f}}L_{\boldsymbol{g}_2}T_1 = L_{[\boldsymbol{f},\boldsymbol{g}_2]}T_1.$$

Because by (11.31) and (11.32), $L_{\boldsymbol{g}_1}T_1 = 0$ and $L_{\boldsymbol{g}_2}T_1 = 0$, we can write

$$DT_1 \begin{bmatrix} \boldsymbol{g}_1 & \left(\mathrm{ad}^1\boldsymbol{f},\boldsymbol{g}_1\right) & \boldsymbol{g}_2 & \left(\mathrm{ad}^1\boldsymbol{f},\boldsymbol{g}_2\right) \end{bmatrix} = \mathbf{0}, \tag{11.33}$$

that is,

$$\begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -K_2\sin(K_3x_1) & 0 & K_2\cos(K_3x_1) \\ 1 & -K_5 & 0 & 0 \\ 0 & 0 & 1 & -K_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}.$$

By inspection, a nontrivial solution to the above is

$$DT_1 = \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & 0 & 0 & 0 \end{bmatrix}, \tag{11.34}$$

where the function T_1 may only be a function of x_1 for DT_1 to be integrable. We specify T_1 later. To proceed, we note that by (11.28) and (11.29),

$$\begin{bmatrix} \tilde{g}_{31} & \tilde{g}_{32} \\ \tilde{g}_{41} & \tilde{g}_{42} \end{bmatrix} = \begin{bmatrix} (DT_3)\,\boldsymbol{g}_1 & (DT_3)\,\boldsymbol{g}_2 \\ (DT_4)\,\boldsymbol{g}_1 & (DT_4)\,\boldsymbol{g}_2 \end{bmatrix}, \tag{11.35}$$

Using the structure of the state transformation (11.28) and the Leibniz formula (11.7), we express the element $\tilde{g}_{31} = (DT_3)\,\boldsymbol{g}_1$ as

$$\begin{aligned} (DT_3)\,\boldsymbol{g}_1 &= L_{\boldsymbol{g}_1}L_{\boldsymbol{f}}T_2 \\ &= L_{[\boldsymbol{f},\boldsymbol{g}_1]}T_2 + L_{\boldsymbol{f}}L_{\boldsymbol{g}_1}T_2 \\ &= L_{[\boldsymbol{f},\boldsymbol{g}_1]}T_2, \end{aligned} \tag{11.36}$$

because $L_{\boldsymbol{g}_1}T_2 = 0$ and therefore $L_{\boldsymbol{f}}L_{\boldsymbol{g}_1}T_2 = 0$. Observing that $T_2 = L_{\boldsymbol{f}}T_1$, employing the Leibniz formula, and taking into account the fact that $L_{\boldsymbol{g}_1}T_1 = 0$, we get

$$\begin{aligned}
L_{[\boldsymbol{f},\boldsymbol{g}_1]}T_2 &= L_{[\boldsymbol{f},\boldsymbol{g}_1]}L_{\boldsymbol{f}}T_1 \\
&= L_{[\boldsymbol{f},[\boldsymbol{f},\boldsymbol{g}_1]]}T_1 + L_{\boldsymbol{f}}L_{[\boldsymbol{f},\boldsymbol{g}_1]}T_1 \\
&= L_{[\boldsymbol{f},[\boldsymbol{f},\boldsymbol{g}_1]]}T_1 + L_{\boldsymbol{f}}\left(L_{\boldsymbol{g}_1}L_{\boldsymbol{f}}T_1 - L_{\boldsymbol{f}}L_{\boldsymbol{g}_1}T_1\right) \\
&= L_{[\boldsymbol{f},[\boldsymbol{f},\boldsymbol{g}_1]]}T_1 \\
&= DT_1\left(\mathrm{ad}^2\boldsymbol{f},\boldsymbol{g}_1\right).
\end{aligned} \tag{11.37}$$

Hence,

$$\tilde{g}_{31} = DT_1\left(\mathrm{ad}^2\boldsymbol{f},\boldsymbol{g}_1\right). \tag{11.38}$$

Similarly, we find

$$\tilde{g}_{32} = DT_1\left(\mathrm{ad}^2\boldsymbol{f},\boldsymbol{g}_2\right). \tag{11.39}$$

Substituting (11.23), (11.24), and (11.34) into (11.38) and (11.39), we find

$$\tilde{g}_{31} = -\frac{\partial T_1}{\partial x_1}K_2\,\sin(K_3x_1), \tag{11.40}$$

and

$$\tilde{g}_{32} = \frac{\partial T_1}{\partial x_1}K_2\,\cos(K_3x_1). \tag{11.41}$$

Substituting the above expressions into (11.27) yields

$$\begin{aligned}
\tilde{g}_{31}\tilde{g}_{42} - \tilde{g}_{32}\tilde{g}_{41} &= \left(DT_1\left(\mathrm{ad}^2\boldsymbol{f},\boldsymbol{g}_1\right)\right)\left(\left(DT_4\right)\boldsymbol{g}_2\right) - \left(DT_1\left(\mathrm{ad}^2\boldsymbol{f},\boldsymbol{g}_2\right)\right)\left(\left(DT_4\right)\boldsymbol{g}_1\right) \\
&= -\frac{\partial T_1}{\partial x_1}K_2\,\sin(K_3x_1)\frac{\partial T_4}{\partial x_4} - \frac{\partial T_1}{\partial x_1}K_2\,\cos(K_3x_1)\frac{\partial T_4}{\partial x_3} \\
&\neq 0.
\end{aligned} \tag{11.42}$$

Suppose that we choose

$$\frac{\partial T_1}{\partial x_1} = \frac{1}{K_2}, \quad \frac{\partial T_4}{\partial x_3} = \cos(K_3x_1), \quad \text{and} \quad \frac{\partial T_4}{\partial x_4} = \sin(K_3x_1). \tag{11.43}$$

Substituting (11.43) into (11.42) gives

$$\det\begin{bmatrix} \tilde{g}_{31} & \tilde{g}_{32} \\ \tilde{g}_{41} & \tilde{g}_{42} \end{bmatrix} = -1.$$

Using (11.34), (11.43), and (11.28), we obtain

$$\left.\begin{array}{rcl} T_1 & = & x_1/K_2 \\ T_2 & = & L_{\boldsymbol{f}}T_1 = x_2/K_2 \\ T_3 & = & L_{\boldsymbol{f}}T_2 = -\frac{K_1}{K_2}x_2 - x_3\sin(K_3x_1) + x_4\cos(K_3x_1) \\ T_4 & = & x_3\cos(K_3x_1) + x_4\sin(K_3x_1) \end{array}\right\} \tag{11.44}$$

The Jacobian matrix of $\boldsymbol{T}$ is

$$D\boldsymbol{T} = \begin{bmatrix} 1/K_2 & 0 & 0 & 0 \\ 0 & 1/K_2 & 0 & 0 \\ -K_3x_3\cos(K_3x_1) - K_3x_4\sin(K_3x_1) & -K_1/K_2 & -\sin(K_3x_1) & \cos(K_3x_1) \\ -K_3x_3\sin(K_3x_1) + K_3x_4\cos(K_3x_1) & 0 & \cos(K_3x_1) & \sin(K_3x_1) \end{bmatrix} \tag{11.45}$$

We have

$$\boldsymbol{T}(\boldsymbol{0}) = \boldsymbol{0} \quad \text{and} \quad \det D\boldsymbol{T} = -1/K_2^2 \neq 0 \quad \text{for all} \quad \boldsymbol{x} \in \mathbb{R}^4, \tag{11.46}$$

which means that we were able to find a global transformation into the controller form. The inverse, $\boldsymbol{x} = \boldsymbol{T}^{-1}(\boldsymbol{z})$ is

$$\left.\begin{array}{rcl} x_1 & = & K_2z_1 \\ x_2 & = & K_2z_2 \\ x_3 & = & -K_1z_2\sin(K_3K_2z_1) - z_3\sin(K_3K_2z_1) + z_4\cos(K_3K_2z_1) \\ x_4 & = & K_1z_2\cos(K_3K_2z_1) + z_3\cos(K_3K_2z_1) + z_4\sin(K_3K_2z_1) \end{array}\right\} \tag{11.47}$$

We transform the PM stepper motor model into the new coordinates using (11.28), (11.44), and (11.44) to obtain

$$\begin{aligned} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} &= \begin{bmatrix} z_2 \\ z_3 \\ -K_2K_3z_2z_4 - K_1z_3 - K_2K_4z_2 - K_5z_3 - K_5K_1z_2 \\ K_2K_3z_2z_3 + K_1K_2K_3z_2^2 - K_5z_4 \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\sin(K_3K_2z_1) & \cos(K_3K_2z_1) \\ \cos(K_3K_2z_1) & \sin(K_3K_2z_1) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \end{aligned} \tag{11.48}$$

11.5 Linearizing State-Feedback Control

Once the plant model is transformed into the controller form, we can construct a state-feedback controller in the new coordinates and then using the inverse transformation, represent the controller in the original coordinates. While constructing the controller in the new coordinates, a part of the controller is used to cancel nonlinearities thus resulting in a linear system in the new coordinates. This procedure, as we mentioned at the beginning of this chapter, is called the feedback linearization. Then, we proceed to construct the other part of the controller. This part can be designed using linear control methods because the feedback linearized system is linear.

We illustrate the above method of state feedback design on a numerical example involving the one-link manipulator whose model was described in Subsection 2.4.5. The state space model of the manipulator, given by (2.83), was transformed into the controller form in Example 11.3. The controller form of the one-link manipulator model is given by (11.20). It is easy to design a stabilizing state-feedback controller in the new coordinates. It takes the form

$$u = -\tilde{f}_3\,(z_1, z_2, z_3) - (k_1 z_1 + k_2 z_2 + k_3 z_3), \tag{11.49}$$

where $\tilde{f}_3\,(z_1, z_2, z_3) = 9.8\sin(10z_1) + (9.8\cos(10z_1) - 10)\,z_2 - 10z_3$. Suppose that the desired closed-loop poles of the feedback linearized one-link manipulator are to be located at $\{-3.5, -4, -5\}$. Then, the linear feedback gains k_i, $i = 1, 2, 3$ that shift the poles to these desired locations are:

$$k_1 = 70, \quad k_2 = 51.5, \quad k_3 = 12.5.$$

Applying (11.49) with the above values of the linear feedback gains to the model (11.20) gives

$$\dot{\boldsymbol{z}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -70 & -51.5 & -12.5 \end{bmatrix} \boldsymbol{z}.$$

Next, applying the inverse transformation yields the controller in the original coordinates,

$$u = -7x_1 - 4.15x_2 - 0.25x_3 - 0.98x_2\cos(x_1) - 12.25\sin(x_1). \tag{11.50}$$

In Figure 11.1, three plots of the manipulator's link angle, $x_1 = \theta$, versus time are presented. The initial conditions have the form, $\boldsymbol{x}(0) = \begin{bmatrix} \theta(0) & 0 & 0 \end{bmatrix}^\top$. The control law applied to the manipulator is given by (11.50).

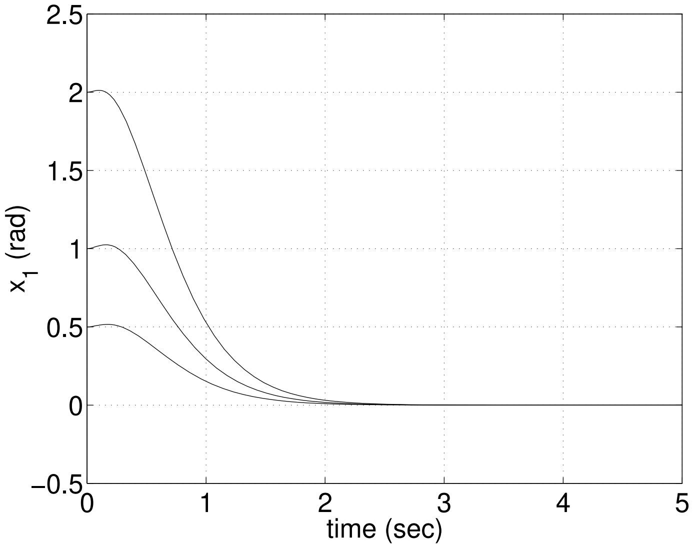


Figure 11.1: Plots of the one-link manipulator's link angle, $x_1 = \theta$, versus time for three different initial angles $\theta(0)$. The manipulator is driven by the state-feedback control law (11.50).

11.6 Observer Form

In this section, we discuss the problem of transforming a class of nonlinear system models into an observer form, which is then used to construct observers for these systems. We consider a single output nonlinear system model of the form

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} &=& \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \\ y &=& h(\boldsymbol{x}) \end{array}\right\} \tag{11.51}$$

First, we analyze the case of unforced system when $\boldsymbol{u} = \boldsymbol{0}$. In such a case (11.51) reduces to

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} &=& \boldsymbol{f}(\boldsymbol{x}) \\ y &=& h(\boldsymbol{x}) \end{array}\right\} \tag{11.52}$$

We desire to find a state variable transformation, written as $\boldsymbol{x} = \boldsymbol{T}(\boldsymbol{z})$ such that the model (11.52) in the $\boldsymbol{z}$-coordinates is

$$\left.\begin{array}{rcl} \dot{\boldsymbol{z}} &=& \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \boldsymbol{z} + \begin{bmatrix} \alpha_0(y) \\ \alpha_1(y) \\ \vdots \\ \alpha_{n-1}(y) \end{bmatrix} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{\alpha}(y) = \tilde{\boldsymbol{f}}(\boldsymbol{z}), \\ y &=& \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \boldsymbol{z} = \boldsymbol{c}\boldsymbol{z}. \end{array}\right\} \tag{11.53}$$

It is easy to verify that the state $\tilde{\boldsymbol{z}}$ of the dynamical system

$$\dot{\tilde{\boldsymbol{z}}} = (\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c})\,\tilde{\boldsymbol{z}} + \boldsymbol{\alpha}(y) + \boldsymbol{l}y, \tag{11.54}$$

asymptotically converges to the state $\boldsymbol{z}$ of (11.53) if the matrix $\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c}$ is asymptotically stable. This is because the dynamics of the error, $\boldsymbol{e} = \boldsymbol{z} - \tilde{\boldsymbol{z}}$, are described by

$$\dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c})\,\boldsymbol{e}, \quad \boldsymbol{e}(t_0) = \boldsymbol{z}(t_0) - \tilde{\boldsymbol{z}}(t_0) \tag{11.55}$$

We derive a sufficiency condition for the existence of a state variable transformation that brings (11.52) into the observer form (11.53). Taking the derivative of $\boldsymbol{x} = \boldsymbol{T}(\boldsymbol{z})$ with respect to time yields

$$\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\dot{\boldsymbol{z}} = \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\tilde{\boldsymbol{f}}(\boldsymbol{z}), \tag{11.56}$$

where we represent the Jacobian matrix of $\boldsymbol{T}(\boldsymbol{z})$, denoted $\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}$, as

$$\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}} = D\boldsymbol{T} = \begin{bmatrix} \frac{\partial \boldsymbol{T}}{\partial z_1} & \cdots & \frac{\partial \boldsymbol{T}}{\partial z_n} \end{bmatrix}. \tag{11.57}$$

Equating the right-hand sides of $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ and (11.56) yields

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{T}(\boldsymbol{z})) = \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\tilde{\boldsymbol{f}}(\boldsymbol{z}). \tag{11.58}$$

Taking the partial derivative of (11.58) with respect to z_k yields

$$\boxed{\frac{\partial \boldsymbol{f}(\boldsymbol{T}(\boldsymbol{z}))}{\partial z_k} = \frac{\partial\left(\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\right)}{\partial z_k}\tilde{\boldsymbol{f}}(\boldsymbol{z}) + \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\frac{\partial \tilde{\boldsymbol{f}}(\boldsymbol{z})}{\partial z_k}} \tag{11.59}$$

On the other hand, application of the chain rule gives

$$\frac{\partial \boldsymbol{f}(\boldsymbol{T}(\boldsymbol{z}))}{\partial z_k} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{T}(\boldsymbol{z})}{\partial z_k}. \tag{11.60}$$

Using (11.58) and the fact that $\boldsymbol{x} = \boldsymbol{T}(\boldsymbol{z})$, the first term on the right-hand side of (11.59) evaluates to

$$\frac{\partial\left(\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\right)}{\partial z_k}\tilde{\boldsymbol{f}}(\boldsymbol{z}) = \frac{\partial\left(\frac{\partial \boldsymbol{T}}{\partial z_k}\right)}{\partial \boldsymbol{z}}\tilde{\boldsymbol{f}}(\boldsymbol{z}) = \frac{\partial\left(\frac{\partial \boldsymbol{T}}{\partial z_k}\right)}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{x}}\tilde{\boldsymbol{f}}(\boldsymbol{z}) = \frac{\partial\left(\frac{\partial \boldsymbol{T}}{\partial z_k}\right)}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{T}(\boldsymbol{z})}{\partial \boldsymbol{z}}\tilde{\boldsymbol{f}}(\boldsymbol{z}) = \frac{\partial\left(\frac{\partial \boldsymbol{T}}{\partial z_k}\right)}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}). \tag{11.61}$$

Employing (11.53) and (11.57), we evaluate the second term on the right-hand side of (11.59) to get

$$\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\frac{\partial \tilde{\boldsymbol{f}}(\boldsymbol{z})}{\partial z_k} = \frac{\partial \boldsymbol{T}}{\partial z_{k+1}}, \quad k = 1, 2, \ldots, n-1. \tag{11.62}$$

Using the above and substituting (11.60) and (11.61) into (11.59) gives

$$\boxed{\frac{\partial \boldsymbol{T}}{\partial z_{k+1}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{T}(\boldsymbol{z})}{\partial z_k} - \frac{\partial \left(\frac{\partial \boldsymbol{T}}{\partial z_k}\right)}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}), \quad k = 1, 2, \ldots, n-1} \tag{11.63}$$

With the help of the Lie-derivative notation introduced in Section 11.4, we express (11.63) in an alternative way. Specifically, using (11.1) and (11.2), we represent (11.63) as

$$\frac{\partial \boldsymbol{T}}{\partial z_{k+1}} = \left[\boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_k}\right] = \left(\mathrm{ad}^1 \boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_k}\right), \quad k = 1, 2, \ldots, n-1. \tag{11.64}$$

Observe that

$$\begin{aligned}
\frac{\partial \boldsymbol{T}}{\partial z_2} &= \left(\mathrm{ad}^1 \boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_1}\right), \\
\frac{\partial \boldsymbol{T}}{\partial z_3} &= \left(\mathrm{ad}^1 \boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_2}\right) = \left(\mathrm{ad}^2 \boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_1}\right), \\
&\ \ \vdots \qquad\quad \vdots
\end{aligned}$$

and in general

$$\boxed{\frac{\partial \boldsymbol{T}}{\partial z_k} = \left(\mathrm{ad}^{k-1} \boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_1}\right), \quad k = 1, 2, \ldots, n} \tag{11.65}$$

With the aid of the above, we express all the columns of the Jacobian matrix (11.57) of $\boldsymbol{T}(\boldsymbol{z})$ in terms of the starting vector $\partial \boldsymbol{T}/\partial z_1$, that is,

$$\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}} = \left[\begin{array}{cccc} \left(\mathrm{ad}^0 \boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_1}\right) & \left(\mathrm{ad}^1 \boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_1}\right) & \cdots & \left(\mathrm{ad}^{n-1} \boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_1}\right) \end{array}\right] \tag{11.66}$$

To obtain an expression for the starting vector $\partial \boldsymbol{T}/\partial z_1$, we use the output equation

$$y = h(\boldsymbol{x}) = z_n. \tag{11.67}$$

We use the chain rule when taking the partial derivative of (11.67) with respect to $\boldsymbol{z}$ to get

$$\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}} = \left[\begin{array}{ccccc} 0 & 0 & \cdots & 0 & 1 \end{array}\right]. \tag{11.68}$$

We utilize (11.5) to express the first component on the left-hand side of the above as

$$\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{T}}{\partial z_1} = L_{\boldsymbol{f}}^0 (Dh) \frac{\partial \boldsymbol{T}}{\partial z_1} = 0. \tag{11.69}$$

Using the Leibniz formula, we represent the second component of the left-hand side of (11.68) as

$$\begin{aligned} 0 &= \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{T}}{\partial z_2} \\ &= \left\langle \nabla h, \left[\boldsymbol{f}, \frac{\partial \boldsymbol{T}}{\partial z_1} \right] \right\rangle \\ &= L_{\frac{\partial \boldsymbol{T}}{\partial z_1}} L_{\boldsymbol{f}} h - L_{\boldsymbol{f}} L_{\frac{\partial \boldsymbol{T}}{\partial z_1}} h \\ &= L_{\frac{\partial \boldsymbol{T}}{\partial z_1}} L_{\boldsymbol{f}} h, \end{aligned} \tag{11.70}$$

because $\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{T}}{\partial z_1} = 0$ by (11.69). We use (11.6) to represent (11.70) as

$$\begin{aligned} 0 &= L_{\frac{\partial \boldsymbol{T}}{\partial z_1}} L_{\boldsymbol{f}} h \\ &= D L_{\boldsymbol{f}} h \frac{\partial \boldsymbol{T}}{\partial z_1} \\ &= L_{\boldsymbol{f}}^1 (Dh) \frac{\partial \boldsymbol{T}}{\partial z_1}. \end{aligned} \tag{11.71}$$

By repeated application of the Leibniz formula and (11.66), we express (11.68) equivalently as

$$\begin{bmatrix} L_{\boldsymbol{f}}^0 (Dh) \\ L_{\boldsymbol{f}}^1 (Dh) \\ \vdots \\ L_{\boldsymbol{f}}^{n-2} (Dh) \\ L_{\boldsymbol{f}}^{n-1} (Dh) \end{bmatrix} \frac{\partial \boldsymbol{T}}{\partial z_1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{11.72}$$

The matrix

$$\begin{bmatrix} L_{\boldsymbol{f}}^0 (Dh) \\ L_{\boldsymbol{f}}^1 (Dh) \\ \vdots \\ L_{\boldsymbol{f}}^{n-2} (Dh) \\ L_{\boldsymbol{f}}^{n-1} (Dh) \end{bmatrix} \tag{11.73}$$

is called the observability matrix of the nonlinear system model (11.52). It follows from (11.72) that the starting vector $\partial \boldsymbol{T} / \partial z_1$ is equal to the last column of the inverse of the observability matrix given by (11.73).

We next discuss the case of transforming a nonlinear system model with inputs, described by (11.51), into its observer form. We proceed as recommended by Krener and Respondek [193, Section 6]. First transform the unforced system into the observer form (11.53). Then, check if the vector field $\boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$ is transformed into the form

$$\left(\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\right)^{-1} \boldsymbol{G}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{T}(\boldsymbol{z})}\, \boldsymbol{u} = \boldsymbol{\beta}(y, \boldsymbol{u}). \tag{11.74}$$

If this is the case, then set

$$\boldsymbol{\gamma}(y, \boldsymbol{u}) = \boldsymbol{\alpha}(y) + \boldsymbol{\beta}(y, \boldsymbol{u}).$$

The above means that the system with inputs is transformed into the observer form,

$$\left.\begin{array}{rcl} \dot{\boldsymbol{z}} & = & \boldsymbol{A}\boldsymbol{z} + \boldsymbol{\gamma}(y, \boldsymbol{u}) \\ y & = & \boldsymbol{c}\boldsymbol{z} \end{array}\right\} \tag{11.75}$$

where the matrices $\boldsymbol{A}$ and $\boldsymbol{c}$ are as in (11.53). If the unforced system cannot be transformed into the observer form or $\boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$ does not transform into (11.74), then the original system (11.51) cannot be transformed into the observer form (11.75).

Example 11.5 Consider the one-link robot manipulator model that we analyzed in Subsection 2.4.5. The observability matrix for the system is

$$\begin{bmatrix} L^0_{\boldsymbol{f}}(Dh) \\ L^1_{\boldsymbol{f}}(Dh) \\ L^2_{\boldsymbol{f}}(Dh) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 9.8\cos(x_1) & 0 & 1 \end{bmatrix}. \tag{11.76}$$

The starting vector $\partial \boldsymbol{T}/\partial z_1$ is the last column of the inverse of the above observability matrix,

$$\partial \boldsymbol{T}/\partial z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{11.77}$$

We find the Jacobian matrix of the desired transformation:

$$\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}} = \left[\ \left(\mathrm{ad}^0 \boldsymbol{f}, \tfrac{\partial \boldsymbol{T}}{\partial z_1}\right) \ \left(\mathrm{ad}^1 \boldsymbol{f}, \tfrac{\partial \boldsymbol{T}}{\partial z_1}\right) \ \left(\mathrm{ad}^2 \boldsymbol{f}, \tfrac{\partial \boldsymbol{T}}{\partial z_1}\right)\ \right] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -10 \\ 1 & -10 & 90 \end{bmatrix}. \tag{11.78}$$

Therefore, we can take

$$\boldsymbol{x} = \boldsymbol{T}(\boldsymbol{z}) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -10 \\ 1 & -10 & 90 \end{bmatrix} \boldsymbol{z}. \tag{11.79}$$

Note that $\boldsymbol{T}(\mathbf{0}) = \mathbf{0}$. The inverse transformation is

$$\boldsymbol{z} = \boldsymbol{T}^{-1}(\boldsymbol{x}) = \boldsymbol{T}^{-1}\boldsymbol{x} = \begin{bmatrix} 10 & 10 & 1 \\ 10 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}. \tag{11.80}$$

The system in the new coordinates has the form

$$\begin{aligned} \dot{\boldsymbol{z}} &= \left(\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\right)^{-1} \boldsymbol{f}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\boldsymbol{T}(\boldsymbol{z})} + \left(\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{z}}\right)^{-1} \boldsymbol{g}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\boldsymbol{T}(\boldsymbol{z})} u = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{\gamma}(y, u) \\ y &= \boldsymbol{c}\boldsymbol{z}. \end{aligned}$$

Performing simple manipulations yields

$$\dot{\boldsymbol{z}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{z} + \begin{bmatrix} 98\sin(y) + 10u \\ 9.8\sin(y) - 10y \\ -10y \end{bmatrix} \tag{11.81}$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \boldsymbol{z} = z_3. \tag{11.82}$$

11.7 Asymptotic Observer

Once the nonlinear system model (11.51) is in the observer form, we proceed with the construction of an asymptotic observer using the technique from the linear systems. The observer dynamics in the new coordinates are

$$\dot{\tilde{\boldsymbol{z}}} = (\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c})\,\tilde{\boldsymbol{z}} + \boldsymbol{l}y + \boldsymbol{\gamma}(y, \boldsymbol{u}), \tag{11.83}$$

where the observer gain vector $\boldsymbol{l}$ is chosen so that the matrix $(\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c})$ has its eigenvalues in the desired locations in the open left-hand complex plane. It is easy to verify that the state estimation error, $\boldsymbol{z} - \tilde{\boldsymbol{z}}$, satisfies the linear differential equation,

$$\frac{d}{dt}(\boldsymbol{z} - \tilde{\boldsymbol{z}}) = (\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c})\,(\boldsymbol{z} - \tilde{\boldsymbol{z}})\,.$$

Applying the inverse transformation to (11.83), we obtain the state estimator dynamics in the original coordinates.

We illustrate the above method of constructing an observer using the model of the one-link manipulator from Subsection 2.4.5. We first construct an observer for the one-link manipulator model in the observer form given by (11.81) and (11.82). Suppose that we wish the eigenvalues of the observer matrix, $\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c}$, to be located at $\{-9, -10, -11\}$. Then, the observer gain vector is $\boldsymbol{l} = \begin{bmatrix} 990 & 299 & 30 \end{bmatrix}^\top$. With the above choice of the observer design parameters, its dynamics in the new coordinates become

$$
\begin{aligned}
\dot{\tilde{\boldsymbol{z}}} &= \begin{bmatrix} 0 & 0 & -990 \\ 1 & 0 & -299 \\ 0 & 1 & -30 \end{bmatrix} \tilde{\boldsymbol{z}} + \begin{bmatrix} 990 \\ 299 \\ 30 \end{bmatrix} y + \begin{bmatrix} 98\sin(y) + 10u \\ 9.8\sin(y) - 10y \\ -10y \end{bmatrix} && (11.84) \\
y &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \boldsymbol{z} = z_3. && (11.85)
\end{aligned}
$$

Using transformations (11.79) and (11.80), we represent the observer dynamics in the original coordinates,

$$
\begin{aligned}
\dot{\tilde{\boldsymbol{x}}} &= \begin{bmatrix} -20 & 1 & 0 \\ -89 & 0 & 1 \\ 100 & -10 & -10 \end{bmatrix} \tilde{\boldsymbol{x}} + \begin{bmatrix} 30 \\ -1 \\ 700 \end{bmatrix} y + \begin{bmatrix} -10y \\ 9.8\sin(y) + 90y \\ -800y + 10u \end{bmatrix} && (11.86) \\
y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{x} = x_1. && (11.87)
\end{aligned}
$$

Performing simple manipulations, we represent the above observer as:

$$
\begin{aligned}
\dot{\tilde{\boldsymbol{x}}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -10 & -10 \end{bmatrix} \tilde{\boldsymbol{x}} + \begin{bmatrix} 0 \\ 9.8\sin(y) \\ 0 \end{bmatrix} - \begin{bmatrix} 20 \\ 89 \\ -100 \end{bmatrix} (\tilde{y} - y) + \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} u && (11.88) \\
y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{x} = x_1 && (11.89) \\
\tilde{y} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tilde{\boldsymbol{x}} = \tilde{x}_1 && (11.90)
\end{aligned}
$$

11.8 Combined Controller-Observer Compensator

Having constructed an observer, we use the state estimates, rather than the states themselves, in the state-feedback control law implementation. The question now is how the incorporation of the observer affects the stability of the closed-loop system with the combined controller-observer compensator in the loop. Recall that in the case of linear systems, we were able to show that the closed-loop system was asymptotically stable. In the present case, one has to be careful when analyzing the stability of the closed-loop system with the combined controller-observer compensator in the loop. The plant's nonlinearities may cause

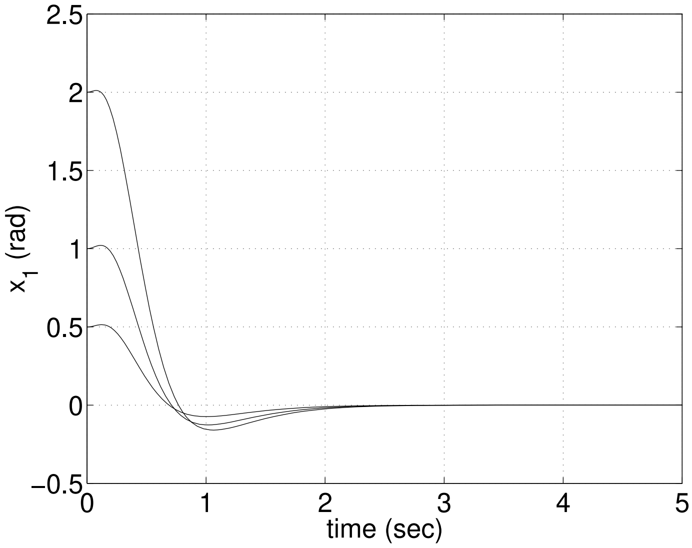


Figure 11.2: Plots of the one-link manipulator's link angle, $x_1 = \theta$, versus time for three different initial angles $\theta(0)$. The manipulator is driven by the combined controller-estimator compensator.

difficulties in stabilizing the closed-loop system using the combined controller-observer compensator. However, for certain "nice" nonlinearities, for example the ones for which the Lipschitz condition is satisfied, one should be able to find stability conditions in terms of the nonlinearity's Lipschitz constant and the location of the observer's poles rather easily.

We illustrate the implementation of the combined controller-observer compensator on the one-link manipulator model. In this example, the controller is

$$u = -7\tilde{x}_1 - 4.15\tilde{x}_2 - 0.25\tilde{x}_3 - 0.98\tilde{x}_2\cos(\tilde{x}_1) - 12.25\sin(\tilde{x}_1). \tag{11.91}$$

In Figure 11.2, plots of the one-link manipulator output for three different initial conditions are shown. The manipulator is driven by the combined controller-observer compensator consisting of the observer (11.88), (11.89), and (11.90) and the control law (11.91). The initial conditions of the observer were set to zero while the initial conditions of the plant were the same as when we implemented the state-feedback control alone. Comparing plots of Figures 11.1 and 11.2, we can see that the presence of the observer alters the system output in a noticeable way.

11.9 Notes

Friedland [108, page 123] has the following remarks to offer on the method of feedback linearization: "The idea underlying feedback linearization is fairly obvious. The method probably was used for many years in practical control system design without anyone having reported it. Concurrently, a number of mathematical investigations on the use of differential geometry in dynamical systems were conducted without much notice by control system designers."

Our exposition of feedback linearization is based on Sommer [309] and [368]. The research in applying differential geometric methods to the control design intensified after the papers by Su [311], and by Hunt, Su, and Meyer [153] appeared. Bakker and Nijmeijer [19] showed the equivalence of the Sommer's approach and that of Hunt, Su, and Meyer. Our treatment of observers follows the approach of Bestle and Zeitz [31], which in turn was presented, using the differential geometry notation, in [338]. Zeitz [372] offers a generalization of the method proposed by Walcott et al. [338]. A short and readable introduction to feedback linearization is [158]. For further study of differential geometric techniques in nonlinear feedback design, we recommend books by Isidori [157], and Marino and Tomei [230].

11.10 Exercises

Exercise 11.1 Verify the Leibniz formula (11.7).

Exercise 11.2 Consider the one-link robot manipulator of Subsection 2.4.5 whose model is given on page 97 by (2.83) and (2.84). Treat this model as a simulation model. In order to obtain a design model, eliminate the armature inductance from the design model. In other words, neglect high-frequency parasitics represented by L_a in the simulation model. For this set $L_a = 0$, compute x_3 from the equation obtained by setting $L_a = 0$ and substitute the result into the second of the state equations. The result will be a second-order design model. Represent the design model in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}(u + \xi),$$

where $\xi = 9.8\sin(x_1)$ represents the model nonlinearity. Use the design model to construct:

(a) the sliding surface, $\sigma = \boldsymbol{s}\boldsymbol{x} = 0$, so that the design model restricted to this surface will have a pole at -3;

(b) construct a control law of the form

$$u = -k_1 x_1 \text{sign}(x_1\sigma) - k_2 x_2 \text{sign}(x_2\sigma)$$

so that $\sigma\dot{\sigma} < 0$ for all $\boldsymbol{x} \in \mathbb{R}^2$ for which $\sigma(\boldsymbol{x}) \neq 0$. While constructing the controller u use the fact that

$$-1 \leq \frac{\sin(x_1)}{x_1} \leq 1.$$

(c) Apply the above control law to the simulation model in which $L_a = 10$ mH. Plot x_1, x_2 and x_3 versus time and compare these plots with the plots of x_1 and x_2 versus time when the above controller was applied to the design model. Investigate a destabilizing effect of increasing L_a on the behavior of the closed-loop system composed of the simulation model and the control law from part (b).

Exercise 11.3 Let $\mathcal{S}$ be an open set in $\mathbb{R}^n$ and let $\boldsymbol{f} : \mathcal{S} \to \mathbb{R}^n$, where

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) & f_2(\boldsymbol{x}) & \cdots & f_n(\boldsymbol{x}) \end{bmatrix}^\top.$$

Recall that if all partial derivatives of all functions f_i exist and are continuous in $\mathcal{S}$, then we say that $\boldsymbol{f}$ is a $\mathcal{C}^1$ function, map, in $\mathcal{S}$. We say that $\boldsymbol{f}$ is a $\mathcal{C}^p$ function, map, in $\mathcal{S}$ if each component f_i is continuous and has continuous partial derivatives in $\mathcal{S}$ for all $k = 1, 2, \ldots, p$.

Let $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a $\mathcal{C}^p$-map, $p > 0$, on an open subset $\mathcal{S}$ of $\mathbb{R}^n$, and let $\boldsymbol{a}$ be a point of $\mathcal{S}$. We say that $\boldsymbol{f}$ is locally $\mathcal{C}^p$-invertible at $\boldsymbol{a}$ if there is an open set $\mathcal{U} \subset \mathcal{S}$ containing $\boldsymbol{a}$ and an open set $\mathcal{V}$ containing $\boldsymbol{f}(\boldsymbol{a})$ such that $\boldsymbol{f} : \mathcal{U} \to \mathcal{V}$ has a $\mathcal{C}^p$ inverse $\boldsymbol{h} : \mathcal{V} \to \mathcal{U}$ so that for all $\boldsymbol{x} \in \mathcal{U}$,

$$\boldsymbol{h}(\boldsymbol{f}(\boldsymbol{x})) = \boldsymbol{x}.$$

Figure 11.3 illustrates the sets $\mathcal{U}$ and $\mathcal{V}$ for one variable case. The Jacobian matrix of $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, denoted $\boldsymbol{J}(\boldsymbol{x})$, is defined as

$$\boldsymbol{J}(\boldsymbol{x}) = D\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \nabla f_1(\boldsymbol{x})^\top \\ \nabla f_2(\boldsymbol{x})^\top \\ \vdots \\ \nabla f_n(\boldsymbol{x})^\top \end{bmatrix} = \begin{bmatrix} Df_1(\boldsymbol{x}) \\ Df_2(\boldsymbol{x}) \\ \vdots \\ Df_n(\boldsymbol{x}) \end{bmatrix}.$$

Prove the inverse function theorem:

Theorem 11.6 *Let $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a $\mathcal{C}^p$-map, $p > 0$, on an open subset $\mathcal{S}$ of $\mathbb{R}^n$, and let $\boldsymbol{a}$ be a point of $\mathcal{S}$ such that the Jacobian matrix $\boldsymbol{J}(\boldsymbol{a})$ is invertible, and $\boldsymbol{b} = \boldsymbol{f}(\boldsymbol{a})$. Then, there exist open sets $\mathcal{U}$ and $\mathcal{V}$ in $\mathbb{R}^n$ such that $\boldsymbol{a} \in \mathcal{U}$, $\boldsymbol{b} \in \mathcal{V}$, $\boldsymbol{f}$ is one-to-one on $\mathcal{U}$, $\boldsymbol{f}(\mathcal{U}) = \mathcal{V}$, and $\boldsymbol{f}$ is locally $\mathcal{C}^p$-invertible at $\boldsymbol{a}$.*

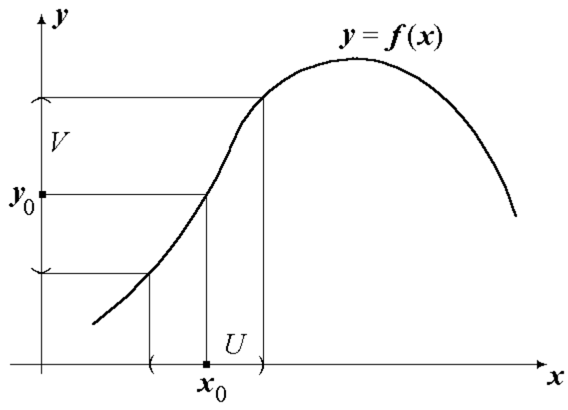


Figure 11.3: An illustration of the sets $\mathcal{U}$ and $\mathcal{V}$ that appear in the inverse function theorem.

Exercise 11.4 For the dynamical system model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1 x_2 + x_2 \\ x_1 + x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u,$$

(i) Construct a smooth state-space transformation, $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$, that transforms the system into the controller form;

(ii) Transform the system into the controller form;

(iii) In the $\boldsymbol{z}$-coordinates, design a linearizing state-feedback controller so that the linear closed-loop system has poles at $-2 \pm 2j$;

(iv) Represent the control law in the $\boldsymbol{x}$-coordinates.

Exercise 11.5 Consider the following model of a dynamical system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1^2 + x_2 \\ x_1^3 + x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u.$$

(i) Transform the system into the controller form;

(ii) Design the linearizing state-feedback controller such that the closed-loop system in the new coordinates has its poles at -1 and -2;

(iii) Linearize the closed-loop system about $\boldsymbol{x} = \mathbf{0}$ and find the poles of the linearized model;

(iv) Perform numerical simulations of the dynamical behavior of the closed-loop system for a few initial conditions. Plot x_1 versus time as well as x_2 versus time.

Exercise 11.6 Consider the following model of a dynamical system,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_2 + 2x_2^2 \\ 2x_2 \end{bmatrix} + \begin{bmatrix} x_2 \\ 1 \end{bmatrix} u.$$

(i) Construct a smooth state-space transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ that transforms the system into the controller form, then transform the system into the controller form;

(ii) In the $\boldsymbol{z}$-coordinates, design a linearizing state-feedback controller so that the linear closed-loop system has poles at $-2 \pm 2j$, then represent the control law in the $\boldsymbol{x}$-coordinates;

(iii) Perform numerical simulations of the dynamical behavior of the closed-loop system for a few initial conditions. Plot x_1 versus time as well as x_2 versus time.

Exercise 11.7 Consider a dynamical system modeled by the equations,

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -x_1 \\ x_1 - x_2 - x_2^3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \\ y &= x_2. \end{aligned}$$

(i) Find a smooth state-space transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ that brings the above system model into the controller form;

(ii) Find the system's equations in the new coordinates;

(iii) Find a smooth state-space transformation $\boldsymbol{x} = \boldsymbol{T}(\boldsymbol{z})$ that brings the above system model into the observer form;

(iv) Find the system's observer form.

Exercise 11.8 Consider a dynamical system modeled by the following equations:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} x_1^2 + x_2 \\ x_1^3 + x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \end{aligned}$$

Note that this is the same plant as in Exercise 11.5 plus the output map.

(i) Transform the above model into the observer form;

(ii) Design an observer so that the poles of the linearized error dynamics are located at -3 and -4;

(iii) Use the control law developed in Exercise 11.5 and the observer from the previous part of this exercise to construct a combined controller-observer compensator. Simulate the behavior of the closed-loop system for a few different initial conditions. Note that because the state variables of the observer are available to you, you can always set zero initial conditions on the observer if you wish to do so. Plot the output of the plant versus time.

Chapter 12

Unknown Input Observer (UIO) Design

> The best piece of advice I got before I wrote this book was from a colleague of mine, David Ignatius, who is editor of the "Outlook" section at *The Washington Post*. He had written a novel about the Middle East [SIRO], a very good novel. He said, "When you're writing this book always imagine whether the reader is going to want to turn the page. Whether what you're writing, how you're writing, and the anecdotes you're telling will compel that reader to want to turn the page." That's what I tried to keep in mind. Hence the book is a really a combination of analysis of history, of autobiography, of anecdote, put into my own kind of Cuisinart." Thomas Friedman in [202, p. 288]

In this Chapter, we present design procedures for full- and reduced-order observers for a class of dynamical systems modeled by a triple $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ where some or all of the inputs are unknown. The unknown input can be a combination of unmeasurable or unmeasured disturbances, unknown control action, and unmodeled system dynamics. The first design method uses a projection operator approach to the state estimation where the state of the system, whose state is to be estimated, is decomposed into known and unknown components. The unknown component is, in general, a skew projection, that is, not necessarily orthogonal, of the whole state along the subspace in which the available state component resides. We then construct a dynamical system to estimate the unknown component. Finally, we combine the output of the dynamical system, which estimates the unknown state component, with the available state information to obtain the observer that estimates the whole state. In the second design method we employ a sliding mode approach combined with the second method of Lyapunov. We include design algorithms and illustrate the results with numerical examples. Portions of this Chapter are based on our paper [151].

12.1 Introduction

Since the early developments of observers for systems without uncertainties, observers for plants with both known and unknown inputs have been developed resulting in the so-called unknown input observer (UIO) architectures, such as, for example, in [141, 346, 313, 32, 197, 198, 360, 142, 148, 71, 57, 68, 56, 143]. Barmish and Galimidi [22] showed that robust stabilization of uncertain systems via the Luenberger observer is possible provided the uncertain parameters do not exceed a certain computable threshold. More recently, observer architectures utilizing the concept of sliding modes were proposed for uncertain systems, see, for example, [339, 338, 340, 146, 369, 367, 370, 92, 330, 365, 123, 186]. Other methods of the observer design for linear systems developed prior 1983 are reported by O'Reilly in [258].

Observers for uncertain systems play an essential role in robust model-based fault detection [92, 56, 93, 291, 162]. The basic idea behind the use of the observers for fault detection is to form residuals from the difference between the actual system outputs and the estimated outputs using an observer. Once a fault occurs, the residuals are expected to react by becoming greater than a pre-specified threshold. When the system under consideration is subject to unknown disturbances or unknown inputs, their effect has to be decoupled from the residuals to avoid false alarms.

12.2 Uncertain System Model

The class of uncertain dynamical systems that we consider is modeled by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} && (12.1)\\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}, && (12.2) \end{aligned}$$

where the input matrix $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ and the output matrix $\boldsymbol{C} \in \mathbb{R}^{p\times n}$. We assume that the model parameters $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ are known. We further assume that some or all of the inputs are unknown and that the last m_2 inputs are unknown and the remaining $m_1 = m - m_2$ inputs are known. We partition the input matrix $\boldsymbol{B}$ corresponding to the known and unknown inputs as

$$\boldsymbol{B} = \left[\begin{array}{cc} \boldsymbol{B}_1 & \boldsymbol{B}_2 \end{array}\right],$$

where $\boldsymbol{B}_1 \in \mathbb{R}^{n\times m_1}$ and $\boldsymbol{B}_2 \in \mathbb{R}^{n\times m_2}$. Let

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix}.$$

Then, the system model (12.1) can be represented as

$$\boxed{\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{B}_2\boldsymbol{u}_2} \tag{12.3}$$

The vector function $\boldsymbol{u}_2$ may also model lumped uncertainties or nonlinearities in the plant. The output matrix $\boldsymbol{C} \in \mathbb{R}^{p\times n}$. We assume that the pair $(\boldsymbol{A}, \boldsymbol{C})$ is detectable.

12.3 A Projection Operator Approach

12.3.1 Background Results

In this section we employ a projection operator approach to the problem of state observation of uncertain systems. In our further discussion, we assume that the matrix $\boldsymbol{B}_2$ has full column rank. We begin our presentation by noticing that because the system output $\boldsymbol{y}$ is known, it would seem reasonable to decompose the state $\boldsymbol{x}$ as

$$\begin{aligned} \boldsymbol{x} &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{x} + \boldsymbol{M}\boldsymbol{C}\boldsymbol{x} \\ &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{x} + \boldsymbol{M}\boldsymbol{y}, \end{aligned} \tag{12.4}$$

where $\boldsymbol{M}$ is an $n\times p$ real matrix, and the unknown part of the decomposition is $(\boldsymbol{I}-\boldsymbol{M}\boldsymbol{C})\boldsymbol{x}$. Let $\boldsymbol{q} = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{x}$, then $\boldsymbol{x} = \boldsymbol{q} + \boldsymbol{M}\boldsymbol{y}$, and we have

$$\begin{aligned} \dot{\boldsymbol{q}} &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\dot{\boldsymbol{x}} \\ &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{B}_2\boldsymbol{u}_2) \\ &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_1\boldsymbol{u}_1) + (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{B}_2\boldsymbol{u}_2 \\ &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1) + (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{B}_2\boldsymbol{u}_2. \end{aligned}$$

If $\boldsymbol{M}$ is chosen so that $(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{B}_2 = \boldsymbol{O}$, then the dynamics of $\boldsymbol{q}$ depend only on the known quantities $\boldsymbol{u}_1$ and $\boldsymbol{y}$:

$$\dot{\boldsymbol{q}} = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1). \tag{12.5}$$

Note that if we start the above dynamical system with the initial condition $\boldsymbol{q}(0) = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{x}(0)$, then $\boldsymbol{x} = \boldsymbol{q} + \boldsymbol{M}\boldsymbol{C}\boldsymbol{x} = \boldsymbol{q} + \boldsymbol{M}\boldsymbol{y}$ for all $t \geq 0$. But since $\boldsymbol{x}(0)$ is assumed to be unknown, then $\tilde{\boldsymbol{x}} = \boldsymbol{q} + \boldsymbol{M}\boldsymbol{y}$ is only an approximation of $\boldsymbol{x}$. To improve convergence rate or to ensure convergence, we add an extra term to the right-hand side of (12.5) to obtain

$$\begin{aligned}\dot{\boldsymbol{q}} &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{q} - \boldsymbol{C}\boldsymbol{M}\boldsymbol{y}))\\ &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}\boldsymbol{C}(\boldsymbol{x} - \boldsymbol{q} - \boldsymbol{M}\boldsymbol{y})).\end{aligned}$$

Let $\boldsymbol{e} = \boldsymbol{x} - \tilde{\boldsymbol{x}}$. We will show that

$$\dot{\boldsymbol{e}} = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{e}$$

and $\boldsymbol{e}(t) \to \boldsymbol{0}$ as $t \to \infty$ under mild conditions.

Since

$$\text{rank }(\boldsymbol{M}\boldsymbol{C}\boldsymbol{B}_2) \le \text{rank }(\boldsymbol{C}\boldsymbol{B}_2) \le \text{rank }(\boldsymbol{B}_2),$$

the equality $(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{B}_2 = \boldsymbol{O}$ makes it necessary that rank $(\boldsymbol{C}\boldsymbol{B}_2) = \text{rank}(\boldsymbol{B}_2)$, which we assume throughout the Chapter. This rank condition also implies that there must be at least as many independent outputs as unknown inputs for the method to work.

For the error dynamics matrix $(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})$ to be asymptotically stable, it is not sufficient for $(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})$ to be asymptotically stable. It is possible for a product of a projection matrix and an asymptotically stable matrix to be unstable as the following simple example shows.

Example 12.1 Let

$$\boldsymbol{\Pi} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \text{ and } \boldsymbol{A} = \begin{bmatrix} 1 & -3 \\ 3 & -2 \end{bmatrix}.$$

It is easy to check that $\boldsymbol{A}$ is asymptotically stable while $\boldsymbol{\Pi}\boldsymbol{A}$ is unstable. Furthermore, the system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ restricted to the range of $\boldsymbol{\Pi}$ is governed by $\dot{z} = z$, which is also unstable in this case.

We analyze convergence properties of the proposed full-order observer and then we use the results of our analysis to propose a new type of reduced-order observer for uncertain systems. Consider the following dynamical system model,

$$\dot{\boldsymbol{q}} = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{q} - \boldsymbol{C}\boldsymbol{M}\boldsymbol{y})) \tag{12.6}$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{q} + \boldsymbol{M}\boldsymbol{y}. \tag{12.7}$$

We will now show that $\tilde{\boldsymbol{x}} \to \boldsymbol{x}$ as $t \to \infty$. To this end let

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \tilde{\boldsymbol{x}}(t)$$

denote the estimation error. Then, using $(\boldsymbol{I} - \boldsymbol{MC})\boldsymbol{B}_2 = \boldsymbol{O}$ and $\boldsymbol{y} = \boldsymbol{Cx}$, we obtain

$$\begin{aligned}
\frac{d\boldsymbol{e}}{dt} &= \frac{d}{dt}(\boldsymbol{x} - \tilde{\boldsymbol{x}}) \\
&= \frac{d}{dt}(\boldsymbol{x} - \boldsymbol{q} - \boldsymbol{MCx}) \\
&= \frac{d}{dt}((\boldsymbol{I} - \boldsymbol{MC})\boldsymbol{x} - \boldsymbol{q}) \\
&= (\boldsymbol{I} - \boldsymbol{MC})(\boldsymbol{Ax} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{B}_2\boldsymbol{u}_2) \\
&\quad -(\boldsymbol{I} - \boldsymbol{MC})(\boldsymbol{Aq} + \boldsymbol{AMy} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{Cq} - \boldsymbol{CMy})) \\
&= (\boldsymbol{I} - \boldsymbol{MC})(\boldsymbol{Ax} + \boldsymbol{B}_1\boldsymbol{u}_1) + (\boldsymbol{I} - \boldsymbol{MC})\boldsymbol{B}_2\boldsymbol{u}_2 \\
&\quad -(\boldsymbol{I} - \boldsymbol{MC})(\boldsymbol{Aq} + \boldsymbol{AMCx} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}(\boldsymbol{Cx} - \boldsymbol{Cq} - \boldsymbol{CMCx})) \\
&= (\boldsymbol{I} - \boldsymbol{MC})(\boldsymbol{A} - \boldsymbol{LC})(\boldsymbol{x} - \boldsymbol{q} - \boldsymbol{MCx}) \\
&= (\boldsymbol{I} - \boldsymbol{MC})(\boldsymbol{A} - \boldsymbol{LC})\,\boldsymbol{e}. \qquad (12.8)
\end{aligned}$$

Our objective is to specify $\boldsymbol{M}$ and $\boldsymbol{L}$ and a set of initial conditions so that $\boldsymbol{e}(t) \to \boldsymbol{0}$ as $t \to \infty$. A particular class of solutions to $(\boldsymbol{I} - \boldsymbol{MC})\boldsymbol{B}_2 = \boldsymbol{O}$ is given by

$$\boldsymbol{M} = \boldsymbol{B}_2\left((\boldsymbol{CB}_2)^\dagger + \boldsymbol{H}_0\left(\boldsymbol{I}_p - (\boldsymbol{CB}_2)(\boldsymbol{CB}_2)^\dagger\right)\right),$$

where the superscript † denotes the Moore-Penrose pseudo-inverse operation and $\boldsymbol{H}_0 \in \mathbb{R}^{m_2 \times p}$ is a design parameter matrix. (See, for example, Kaczorek [165, Section 1.5] or [62, Section 12.5] for more information on the subject of pseudo-inverse matrices.) Because, by assumption, rank $(\boldsymbol{CB}_2) =$ rank $\boldsymbol{B}_2$ and $\boldsymbol{B}_2$ has full rank, we have $(\boldsymbol{CB}_2)^\dagger(\boldsymbol{CB}_2) = \boldsymbol{I}_{m_2}$. If $\boldsymbol{CB}_2$ is a square matrix, then $\boldsymbol{CB}_2$ is invertible by assumption and the above $\boldsymbol{M}$ reduces to $\boldsymbol{B}_2(\boldsymbol{CB}_2)^{-1}$. Furthermore, it is easy to check that for the above class of $\boldsymbol{M}$, the product $\boldsymbol{MC}$ is a projection (not necessarily orthogonal):

$$(\boldsymbol{MC})^2 = \boldsymbol{MC}.$$

It follows that

$$\tilde{\boldsymbol{\Pi}} = \boldsymbol{I} - \boldsymbol{MC}$$

is also a projection.

To proceed further, we need the following lemma.

Lemma 12.1 *Let* $\tilde{\boldsymbol{\Pi}} : \mathbb{R}^n \to \mathbb{R}^n$ *be a projection, that is,* $\tilde{\boldsymbol{\Pi}}^2 = \tilde{\boldsymbol{\Pi}}$, *and let* $\operatorname{rank} \tilde{\boldsymbol{\Pi}} = n - m_2$. *Then* $\tilde{\boldsymbol{\Pi}}$ *has* $(n - m_2)$ *eigenvalues equal to* 1 *while the remaining* m_2 *eigenvalues are located at* 0 *and there is a basis of* $\mathbb{R}^n$ *in which the matrix* $\tilde{\boldsymbol{\Pi}}$ *relative to this basis has the form*

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix},$$

that is, there is an invertible matrix $\boldsymbol{T}$ *whose columns are eigenvectors of* $\tilde{\boldsymbol{\Pi}}$ *such that*

$$\boldsymbol{T}^{-1}\tilde{\boldsymbol{\Pi}}\boldsymbol{T} = \boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}. \tag{12.9}$$

Proof See Smith [304, pp. 156–158 and pp. 194–195].

□

In the next Section, we provide a numerically efficient method for constructing $\boldsymbol{T}$ such that (12.9) holds; see (12.19).

12.3.2 Constructing Full-Order UIO

We begin this section by introducing the following coordinate transformation,

$$\tilde{\boldsymbol{e}} = \boldsymbol{T}^{-1}\boldsymbol{e}, \tag{12.10}$$

where the transformation matrix $\boldsymbol{T}$ is obtained, using Lemma 12.1, from the representation of the projection operator $\tilde{\boldsymbol{\Pi}}$ in the form

$$\tilde{\boldsymbol{\Pi}} = \boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{T}^{-1}. \tag{12.11}$$

Applying the coordinate transformation (12.10) to the error equation (12.8) gives

$$\begin{aligned} \dot{\tilde{\boldsymbol{e}}} &= \boldsymbol{\Pi}\boldsymbol{T}^{-1}(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{T}\tilde{\boldsymbol{e}} \\ &= \boldsymbol{\Pi}\left(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} - \left(\boldsymbol{T}^{-1}\boldsymbol{L}\right)(\boldsymbol{C}\boldsymbol{T})\right)\tilde{\boldsymbol{e}}. \end{aligned} \tag{12.12}$$

Let

$$\tilde{\boldsymbol{A}} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} = \begin{bmatrix} \tilde{\boldsymbol{A}}_{11} & \tilde{\boldsymbol{A}}_{12} \\ \tilde{\boldsymbol{A}}_{21} & \tilde{\boldsymbol{A}}_{21} \end{bmatrix}, \quad \tilde{\boldsymbol{L}} = \boldsymbol{T}^{-1}\boldsymbol{L} = \begin{bmatrix} \tilde{\boldsymbol{L}}_1 \\ \tilde{\boldsymbol{L}}_2 \end{bmatrix}, \quad \tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{T} = \begin{bmatrix} \tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{C}}_2 \end{bmatrix}, \tag{12.13}$$

where $\tilde{\boldsymbol{A}}_{11} \in \mathbb{R}^{(n-m_2)\times(n-m_2)}$, $\tilde{\boldsymbol{L}}_1 \in \mathbb{R}^{(n-m_2)\times p}$, $\tilde{\boldsymbol{C}}_1 \in \mathbb{R}^{p\times(n-m_2)}$, and the remaining block submatrices are of appropriate dimensions. Using the above notation, we represent (12.12) in the form,

$$\begin{aligned}
\dot{\tilde{\boldsymbol{e}}} &= \boldsymbol{\Pi}\left(\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{L}}\tilde{\boldsymbol{C}}\right)\tilde{\boldsymbol{e}} \\
&= \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}\left(\begin{bmatrix} \tilde{\boldsymbol{A}}_{11} & \tilde{\boldsymbol{A}}_{12} \\ \tilde{\boldsymbol{A}}_{21} & \tilde{\boldsymbol{A}}_{21} \end{bmatrix} - \begin{bmatrix} \tilde{\boldsymbol{L}}_1 \\ \tilde{\boldsymbol{L}}_2 \end{bmatrix}\begin{bmatrix} \tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{C}}_2 \end{bmatrix}\right)\tilde{\boldsymbol{e}} \\
&= \begin{bmatrix} \tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{A}}_{12} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_2 \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}\tilde{\boldsymbol{e}} \qquad (12.14)
\end{aligned}$$

Let

$$\tilde{\boldsymbol{e}} = \begin{bmatrix} \tilde{\boldsymbol{e}}_1 \\ \tilde{\boldsymbol{e}}_2 \end{bmatrix}, \qquad (12.15)$$

where $\tilde{\boldsymbol{e}}_1 \in \mathbb{R}^{n-m_2}$. Note that $\dot{\tilde{\boldsymbol{e}}}_2 = \boldsymbol{0}$. Hence if $\tilde{\boldsymbol{e}}_2(0) = \boldsymbol{0}$, then $\tilde{\boldsymbol{e}}_2 = \boldsymbol{0}$ for all $t \geq 0$. Thus if $\tilde{\boldsymbol{e}}_2 = \boldsymbol{0}$ then

$$\dot{\tilde{\boldsymbol{e}}}_1 = \left(\tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1\right)\tilde{\boldsymbol{e}}_1,$$

and so if $\tilde{\boldsymbol{e}}_2 = \boldsymbol{0}$ and $\tilde{\boldsymbol{e}}_1 \to \boldsymbol{0}$ then $\tilde{\boldsymbol{e}} \to \boldsymbol{0}$. Obviously $\tilde{\boldsymbol{e}}_1 \to \boldsymbol{0}$ for arbitrary $\tilde{\boldsymbol{e}}_1(0)$ if and only if the matrix $\left(\tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1\right)$ is asymptotically stable.

We give a condition on $\boldsymbol{q}(0)$ that guarantees that $\tilde{\boldsymbol{e}}_2 = \boldsymbol{0}$. We have,

$$\begin{aligned}
\boldsymbol{MC} &= \boldsymbol{I} - \tilde{\boldsymbol{\Pi}} \\
&= \boldsymbol{T}(\boldsymbol{I} - \boldsymbol{\Pi})\boldsymbol{T}^{-1}.
\end{aligned}$$

Hence

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{e}}_2 \end{bmatrix} &= (\boldsymbol{I}_n - \boldsymbol{\Pi})\boldsymbol{T}^{-1}\boldsymbol{e} \\
&= \boldsymbol{T}^{-1}\boldsymbol{MCe} \\
&= \boldsymbol{T}^{-1}\boldsymbol{MC}(\boldsymbol{x} - \boldsymbol{q} - \boldsymbol{MCx}) \\
&= \boldsymbol{T}^{-1}(\boldsymbol{MCx} - \boldsymbol{MCq} - (\boldsymbol{MC})^2\boldsymbol{x}) \\
&= -\boldsymbol{T}^{-1}\boldsymbol{MCq}. \qquad (12.16)
\end{aligned}$$

Therefore $\tilde{\boldsymbol{e}}_2(0) = \boldsymbol{0}$ if and only if $\boldsymbol{MCq}(0) = \boldsymbol{0}$, which is equivalent to

$$\boldsymbol{q}(0) = (\boldsymbol{I} - \boldsymbol{MC})\boldsymbol{v}$$

for arbitrary $\boldsymbol{v} \in \mathbb{R}^n$. In particular $\boldsymbol{q}(0) = \boldsymbol{0}$ satisfies the above condition.

In summary, we proved the following theorem,

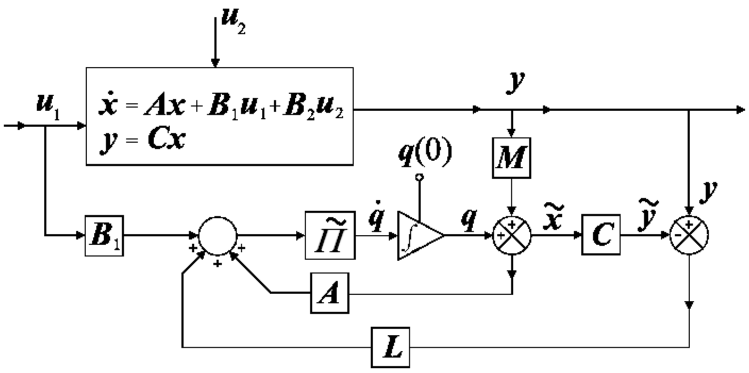


Figure 12.1: Full-order UIO described in Theorem 12.1.

Theorem 12.1 *If the following conditions are satisfied:*

1. *rank* $(\boldsymbol{C}\boldsymbol{B}_2) =$ *rank* $\boldsymbol{B}_2$*;*
2. *the pair* $(\tilde{\boldsymbol{A}}_{11}, \tilde{\boldsymbol{C}}_1)$ *defined in (12.13) is detectable;*
3. $\boldsymbol{q}(0) = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\boldsymbol{v}$ *for arbitrary* $\boldsymbol{v} \in \mathbb{R}^n$*,*

then there exists a gain matrix $\boldsymbol{L}$ *such that the estimation error,* $\boldsymbol{e} = \boldsymbol{x} - \tilde{\boldsymbol{x}}$*, of the full-order observer given by*

$$\begin{aligned} \dot{\boldsymbol{q}} &= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{q} - \boldsymbol{C}\boldsymbol{M}\boldsymbol{y})) \\ \tilde{\boldsymbol{x}} &= \boldsymbol{q} + \boldsymbol{M}\boldsymbol{y} \end{aligned}$$

converges to $\mathbf{0}$ *as* $t \to \infty$.

In Figure 12.1, we show a block diagram of the full-order UIO of Theorem 12.1.

Theorem 12.2 *The second condition of Theorem 12.1 that states that the pair* $(\tilde{\boldsymbol{A}}_{11}, \tilde{\boldsymbol{C}}_1)$ *defined in (12.13) is detectable, is equivalent to*

$$\text{rank} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}_2 \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} = n + m_2 \text{ for all } s \text{ such that } Re(s) \geq 0.$$

Proof We begin the proof by considering the projection matrix, $\tilde{\boldsymbol{\Pi}} = \boldsymbol{I} - \boldsymbol{M}\boldsymbol{C}$, where

$$\boldsymbol{M} = \boldsymbol{B}_2\left((\boldsymbol{C}\boldsymbol{B}_2)^{\dagger} + \boldsymbol{H}_0\left(\boldsymbol{I}_p - (\boldsymbol{C}\boldsymbol{B}_2)(\boldsymbol{C}\boldsymbol{B}_2)^{\dagger}\right)\right).$$

To simplify further analysis, we let $\boldsymbol{F} = (\boldsymbol{C}\boldsymbol{B}_2)^{\dagger} + \boldsymbol{H}_0\left(\boldsymbol{I}_p - (\boldsymbol{C}\boldsymbol{B}_2)(\boldsymbol{C}\boldsymbol{B}_2)^{\dagger}\right)$ and $\boldsymbol{S} = \boldsymbol{F}\boldsymbol{C}$. Note that

$$\boldsymbol{S}\boldsymbol{B}_2 = \boldsymbol{I}_{m_2}. \tag{12.17}$$

Thus, rank $\boldsymbol{S} = m_2$ and so we can find a full rank $n \times (n - m_2)$ matrix $\boldsymbol{W}$ such that

$$\boldsymbol{S}\boldsymbol{W} = \boldsymbol{O}. \tag{12.18}$$

Combining (12.17) and (12.18) we conclude that $[\boldsymbol{W}\ \boldsymbol{B}_2]$ is invertible. Let

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B}_2 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{W}^g \\ \boldsymbol{N} \end{bmatrix},$$

where $\boldsymbol{W}^g$ is $(n - m_2) \times n$ and $\boldsymbol{N}$ is $m_2 \times n$. Then

$$\boldsymbol{N}\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_{m_2} \end{bmatrix} = \boldsymbol{S}\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B}_2 \end{bmatrix}.$$

Since $[\boldsymbol{W}\ \boldsymbol{B}_2]$ is full rank, we conclude that $\boldsymbol{N} = \boldsymbol{S}$. Since

$$\begin{bmatrix} \boldsymbol{W}^g \\ \boldsymbol{S} \end{bmatrix}\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B}_2 \end{bmatrix} = \boldsymbol{I}_n,$$

we have

$$\boldsymbol{W}^g\boldsymbol{W} = \boldsymbol{I}_{n-m_2} \quad \text{and} \quad \boldsymbol{W}^g\boldsymbol{B}_2 = \boldsymbol{O}.$$

Let

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{B}_2 \end{bmatrix}. \tag{12.19}$$

Then

$$\begin{aligned}
\boldsymbol{T}^{-1}\tilde{\boldsymbol{\Pi}}\boldsymbol{T} &= \begin{bmatrix} \boldsymbol{W}^g \\ \boldsymbol{S} \end{bmatrix}(\boldsymbol{I}_n - \boldsymbol{B}\boldsymbol{S})\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B}_2 \end{bmatrix} \\
&= \boldsymbol{I}_n - \begin{bmatrix} \boldsymbol{W}^g\boldsymbol{B}_2\boldsymbol{S} \\ \boldsymbol{S}\boldsymbol{B}_2\boldsymbol{S} \end{bmatrix}\begin{bmatrix} \boldsymbol{W} & \boldsymbol{B}_2 \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{m_2} \end{bmatrix} - \begin{bmatrix} \boldsymbol{W}^g\boldsymbol{B}_2\boldsymbol{S}\boldsymbol{W} & \boldsymbol{W}^g\boldsymbol{B}_2\boldsymbol{S}\boldsymbol{B}_2 \\ \boldsymbol{S}\boldsymbol{B}_2\boldsymbol{S}\boldsymbol{W} & \boldsymbol{S}\boldsymbol{B}_2\boldsymbol{S}\boldsymbol{B}_2 \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{m_2} \end{bmatrix} - \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{m_2} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}.
\end{aligned} \tag{12.20}$$

We apply the following coordinate transformation to the system modeled by (12.1) and (12.2):

$$\begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{\sigma} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}^g \\ \boldsymbol{S} \end{bmatrix} \boldsymbol{x} = \boldsymbol{T}^{-1}\boldsymbol{x}.$$

Then

$$\begin{aligned}
\begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\boldsymbol{\sigma}} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{W}^g\boldsymbol{A}\boldsymbol{W} & \boldsymbol{W}^g\boldsymbol{A}\boldsymbol{B}_2 \\ \boldsymbol{S}\boldsymbol{A}\boldsymbol{W} & \boldsymbol{S}\boldsymbol{A}\boldsymbol{B}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{\sigma} \end{bmatrix} + \boldsymbol{T}^{-1}\boldsymbol{B}_1\boldsymbol{u}_1 + \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I}_{m_2} \end{bmatrix} \boldsymbol{u}_2 \\
&= \begin{bmatrix} \tilde{\boldsymbol{A}}_{11} & \tilde{\boldsymbol{A}}_{12} \\ \tilde{\boldsymbol{A}}_{21} & \tilde{\boldsymbol{A}}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{\sigma} \end{bmatrix} + \boldsymbol{T}^{-1}\boldsymbol{B}_1\boldsymbol{u}_1 + \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I}_{m_2} \end{bmatrix} \boldsymbol{u}_2 \\
\boldsymbol{y} &= \begin{bmatrix} \boldsymbol{C}\boldsymbol{W} & \boldsymbol{C}\boldsymbol{B}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{\sigma} \end{bmatrix} \\
&= \begin{bmatrix} \tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{C}}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{\sigma} \end{bmatrix}.
\end{aligned}$$

If the trajectory of the system described by the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{F}\boldsymbol{C})$ resides in the null space of $\boldsymbol{S}$, then such a motion is described by

$$\dot{\boldsymbol{z}} = \tilde{\boldsymbol{A}}_{11}\boldsymbol{z}.$$

It follows from 1.5 that the poles of the above system are the zeros of the system described by the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{F}\boldsymbol{C})$, which are the complex numbers s for which the system matrix

$$\begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}_2 \\ \boldsymbol{F}\boldsymbol{C} & \boldsymbol{O} \end{bmatrix}$$

loses its normal-rank—see Definition 1.5 on page 45. On the other hand, the zeros of the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$ are also the zeros of the squared-down system $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{F}\boldsymbol{C})$, that is, the zeros of $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$ form a subset of the set of the eigenvalues of $\tilde{\boldsymbol{A}}_{11}$. It is well known that zeros are invariant with respect to similarity transformations. Therefore,

$$\operatorname{rank} \begin{bmatrix} s\boldsymbol{I}_{n-m_2} - \tilde{\boldsymbol{A}}_{11} & -\tilde{\boldsymbol{A}}_{12} & \boldsymbol{O} \\ -\tilde{\boldsymbol{A}}_{21} & s\boldsymbol{I}_{m_2} - \tilde{\boldsymbol{A}}_{22} & \boldsymbol{I}_{m_2} \\ \tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{C}}_2 & \boldsymbol{O} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} s\boldsymbol{I}_{n-m_2} - \tilde{\boldsymbol{A}}_{11} \\ \tilde{\boldsymbol{C}}_1 \end{bmatrix} + 2m_2 \quad \text{for} \quad s \in \mathbb{C},$$

which means that zeros of the system $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$ are in the open left-half plane if and only if the pair $(\tilde{\boldsymbol{A}}_{11}, \tilde{\boldsymbol{C}}_1)$ is detectable.

□

12.3.3 Reduced-Order UIO

The error dynamics of the full-order observer that we analyzed above are given by equation (12.14):

$$\dot{\tilde{\boldsymbol{e}}} = \begin{bmatrix} \tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{A}}_{12} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_2 \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \tilde{\boldsymbol{e}}$$

The reader may have noticed that since we choose the initial condition for $\boldsymbol{q}$ to force $\tilde{\boldsymbol{e}}_2(t) = \boldsymbol{0}$ for $t \geq 0$, the dynamics of the error are completely determined by the dynamics of $\tilde{\boldsymbol{e}}_1$, which are given by

$$\dot{\tilde{\boldsymbol{e}}}_1 = \left(\tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1\right)\tilde{\boldsymbol{e}}_1, \tag{12.21}$$

an $(n - m_2)$-dimensional system. This motivates us to apply the transformation from $\boldsymbol{e}$ to $\tilde{\boldsymbol{e}}$ to $\boldsymbol{q}$:

$$\tilde{\boldsymbol{q}} = \boldsymbol{T}^{-1}\boldsymbol{q}.$$

From equation (12.7),

$$\dot{\boldsymbol{q}} = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{q} - \boldsymbol{C}\boldsymbol{M}\boldsymbol{y}))$$

and $\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C} = \boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{T}^{-1}$, we obtain

$$\begin{aligned} \dot{\tilde{\boldsymbol{q}}} &= \boldsymbol{\Pi}(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\tilde{\boldsymbol{q}} + \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{T}^{-1}\boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{T}^{-1}\boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{T}\tilde{\boldsymbol{q}} - \boldsymbol{C}\boldsymbol{M}\boldsymbol{y})) \\ &= \boldsymbol{\Pi}\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\tilde{\boldsymbol{q}} + \boldsymbol{\Pi}\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{\Pi}\boldsymbol{T}^{-1}\boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{\Pi}\boldsymbol{T}^{-1}\boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{T}\tilde{\boldsymbol{q}} - \boldsymbol{C}\boldsymbol{M}\boldsymbol{y}) \\ &= \boldsymbol{\Pi}\left(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} - \boldsymbol{T}^{-1}\boldsymbol{L}\boldsymbol{C}\boldsymbol{T}\right)\tilde{\boldsymbol{q}} + \boldsymbol{\Pi}\left(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{M} + \boldsymbol{T}^{-1}\boldsymbol{L} - \boldsymbol{T}^{-1}\boldsymbol{L}\boldsymbol{C}\boldsymbol{M}\right)\boldsymbol{y} \\ &\quad + \boldsymbol{\Pi}\boldsymbol{T}^{-1}\boldsymbol{B}_1\boldsymbol{u}_1. \end{aligned} \tag{12.22}$$

Using the notation defined in (12.13), we have

$$\dot{\tilde{\boldsymbol{q}}} = \boldsymbol{\Pi}\left(\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{L}}\tilde{\boldsymbol{C}}\right)\tilde{\boldsymbol{q}} + \boldsymbol{\Pi}\boldsymbol{T}^{-1}\left[\left(\boldsymbol{A}\boldsymbol{M} + \boldsymbol{T}\tilde{\boldsymbol{L}}(\boldsymbol{I}_p - \boldsymbol{C}\boldsymbol{M})\right)\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1\right].$$

Let

$$\tilde{\boldsymbol{q}} = \begin{bmatrix} \tilde{\boldsymbol{q}}_1 \\ \tilde{\boldsymbol{q}}_2 \end{bmatrix},$$

where $\tilde{\boldsymbol{q}}_1 \in \mathbb{R}^{n-m_2}$ and $\tilde{\boldsymbol{q}}_2 \in \mathbb{R}^{m_2}$. Since

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix},$$

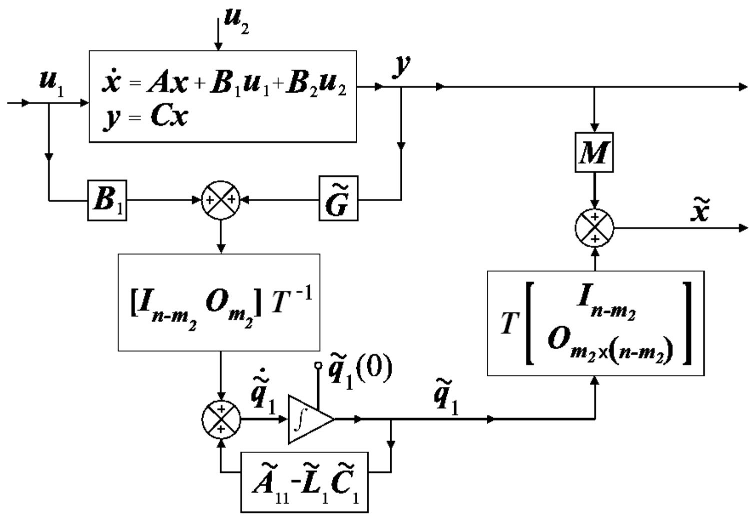


Figure 12.2: Reduced-order UIO.

we have $\dot{\tilde{\boldsymbol{q}}}_2(t) = \mathbf{0}$. Therefore, setting $\tilde{\boldsymbol{q}}_2(0) = \mathbf{0}$ ensures that $\tilde{\boldsymbol{q}}_2(t) = \mathbf{0}$ for $t \geq 0$. We thus can remove m_2 observer states from the observer dynamics. Let $\tilde{\boldsymbol{G}} = \boldsymbol{AM} + \boldsymbol{T}\tilde{\boldsymbol{L}}(\boldsymbol{I}_p - \boldsymbol{CM})$. Then the resulting reduced-order observer takes the form,

$$\begin{aligned} \dot{\tilde{\boldsymbol{q}}}_1 &= (\tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1)\tilde{\boldsymbol{q}}_1 + \left[\begin{array}{cc} \boldsymbol{I}_{n-m_2} & \boldsymbol{O}_{m_2} \end{array}\right]\boldsymbol{T}^{-1}\left(\tilde{\boldsymbol{G}}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1\right), \quad \tilde{\boldsymbol{q}}_1(0) = \mathbf{0} \\ \tilde{\boldsymbol{x}} &= \boldsymbol{T}\left[\begin{array}{c} \boldsymbol{I}_{n-m_2} \\ \boldsymbol{O}_{m_2\times(n-m_2)} \end{array}\right]\tilde{\boldsymbol{q}}_1 + \boldsymbol{My}, \end{aligned}$$

where the vector $\tilde{\boldsymbol{x}}$ is the estimate of the plant state $\boldsymbol{x}$. In Figure 12.2, we show a block diagram of the above reduced-order UIO.

We now summarize the above considerations in the form of the following design algorithm.

Algorithm 12.1 Reduced-Order Unknown Input Observer Design

For a given a quadruple of matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ modeling the plant,

1. Check that rank $(\boldsymbol{CB}_2) = \operatorname{rank} \boldsymbol{B}_2$
   If rank $(\boldsymbol{CB}_2) < \operatorname{rank} \boldsymbol{B}_2$, STOP. The observer does not exist.

2. Compute
$$\boldsymbol{M} = \boldsymbol{B}_2\left((\boldsymbol{CB}_2)^\dagger + \boldsymbol{H}_0\left(\boldsymbol{I}_p - (\boldsymbol{CB}_2)(\boldsymbol{CB}_2)^\dagger\right)\right),$$
   where the superscript $\dagger$ denotes the Moore-Penrose pseudo-inverse operation and $\boldsymbol{H}_0 \in \mathbb{R}^{m_2\times p}$ is a design parameter matrix.

3. Compute the projector
$$\tilde{\boldsymbol{\Pi}} = \boldsymbol{I}_n - \boldsymbol{M}\boldsymbol{C}.$$

4. Compute $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{B}_2 \end{bmatrix}$—see12.19. Note that
$$\tilde{\boldsymbol{\Pi}} = \boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{T}^{-1},$$
where
$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}.$$

5. Compute
$$\tilde{\boldsymbol{A}} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} = \begin{bmatrix} \tilde{\boldsymbol{A}}_{11} & \tilde{\boldsymbol{A}}_{12} \\ \tilde{\boldsymbol{A}}_{21} & \tilde{\boldsymbol{A}}_{22} \end{bmatrix} \quad \text{and} \quad \tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{T} = \begin{bmatrix} \tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{C}}_2 \end{bmatrix},$$
where $\tilde{\boldsymbol{A}}_{11} \in \mathbb{R}^{(n-m_2)\times(n-m_2)}$ and $\tilde{\boldsymbol{C}}_1 \in \mathbb{R}^{p\times(n-m_2)}$

6. Check detectability of the pair $(\tilde{\boldsymbol{A}}_{11}, \tilde{\boldsymbol{C}}_1)$
If the pair $(\tilde{\boldsymbol{A}}_{11}, \tilde{\boldsymbol{C}}_1)$ is not detectable, STOP. The observer does not exist.
Note that if the matrix $\tilde{\boldsymbol{A}}_{11}$ is asymptotically stable, then the pair $(\tilde{\boldsymbol{A}}_{11}, \tilde{\boldsymbol{C}}_1)$ is detectable for an arbitrary matrix $\tilde{\boldsymbol{C}}_1$.

7. If there are eigenvalues of $\tilde{\boldsymbol{A}}_{11}$ that are not asymptotically stable, construct $\tilde{\boldsymbol{L}}_1$ so that the matrix $(\tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1)$ has its eigenvalues in locations as close to the desired eigenvalues as possible.

8. Form
$$\tilde{\boldsymbol{L}} = \begin{bmatrix} \tilde{\boldsymbol{L}}_1 \\ \boldsymbol{O}_{m_2\times p} \end{bmatrix},$$
where $\boldsymbol{O}_{m_2\times p}$ is an $m_2 \times p$ zero matrix.

9. Compute matrix
$$\tilde{\boldsymbol{G}} = \boldsymbol{A}\boldsymbol{M} + \boldsymbol{T}\tilde{\boldsymbol{L}}(\boldsymbol{I}_p - \boldsymbol{C}\boldsymbol{M}).$$

10. Construct the observer,
$$\begin{aligned} \dot{\tilde{\boldsymbol{q}}}_1 &= (\tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1)\tilde{\boldsymbol{q}}_1 + \begin{bmatrix} \boldsymbol{I}_{n-m_2} & \boldsymbol{O}_{m_2} \end{bmatrix}\boldsymbol{T}^{-1}\left(\tilde{\boldsymbol{G}}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1\right), \quad \tilde{\boldsymbol{q}}_1(0) = \boldsymbol{0} \\ \tilde{\boldsymbol{x}} &= \boldsymbol{T}\begin{bmatrix} \boldsymbol{I}_{n-m_2} \\ \boldsymbol{O}_{m_2\times(n-m_2)} \end{bmatrix}\tilde{\boldsymbol{q}}_1 + \boldsymbol{M}\boldsymbol{y}. \end{aligned}$$
The vector $\tilde{\boldsymbol{x}}$ is the estimate of the state $\boldsymbol{x}$.

Example 12.2 We consider the fifth-order lateral axis model of an L-1011 fixed-wing aircraft, with the actuator dynamics neglected, at cruise flight conditions. This model can be found in the book by Edwards and Spurgeon [92, pages 122, 123 and pages 179, 180]. We assume that the inputs to the system are unknown and there are no known inputs. We have,

$$\boldsymbol{A} = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.1540 & -0.0042 & 1.5400 & 0.0000 \\ 0.0000 & 0.2490 & -1.0000 & -5.2000 & 0.0000 \\ 0.0386 & -0.9960 & -0.0003 & -0.1170 & 0.0000 \\ 0.0000 & 0.5000 & 0.0000 & 0.0000 & -0.5000 \end{bmatrix}$$

and

$$\boldsymbol{B}_2 = \begin{bmatrix} 0.0000 & 0.0000 \\ -0.7440 & -0.0320 \\ 0.3370 & -1.1200 \\ 0.0200 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where $\boldsymbol{u}_2 = \begin{bmatrix} \cos(t) & \sin(t) \end{bmatrix}^\top$. We first check that rank $(\boldsymbol{C}\boldsymbol{B}_2) = \operatorname{rank} \boldsymbol{B}_2$. We then compute the matrix $\boldsymbol{M}$, where in this example we set $\boldsymbol{H}_0 = \boldsymbol{O}$,

$$\boldsymbol{M} = \boldsymbol{B}_2(\boldsymbol{C}\boldsymbol{B}_2)^\dagger = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.9993 & 0.0000 & -0.0265 & 0.0000 \\ 0.0000 & 1.0000 & 0.0008 & 0.0000 \\ -0.0265 & 0.0008 & 0.0007 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}.$$

Then, the projector $\tilde{\boldsymbol{\Pi}}$ is

$$\tilde{\boldsymbol{\Pi}} = \boldsymbol{I}_5 - \boldsymbol{M}\boldsymbol{C} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0007 & -0.0000 & 0.0265 & 0.9993 \\ 0.0000 & -0.0000 & 0.0000 & -0.0008 & 0.0000 \\ 0.0000 & 0.0265 & -0.0008 & 0.9993 & -0.0265 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}.$$

We next use (12.19) to compute $\boldsymbol{T}$ such that

$$\tilde{\boldsymbol{\Pi}} = \boldsymbol{T}\boldsymbol{\Pi}\boldsymbol{T}^{-1} = \boldsymbol{T} \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_{3\times 2} \\ \boldsymbol{O}_{3\times 2}^\top & \boldsymbol{O}_{2\times 2} \end{bmatrix} \boldsymbol{T}^{-1}.$$

We have

$$\boldsymbol{T} = \begin{bmatrix} -0.7169 & -0.0190 & -0.6969 & 0 & 0 \\ -0.4930 & 0.0131 & 0.5069 & -0.7440 & -0.0320 \\ 0.0000 & -0.0008 & 0.0000 & 0.3370 & -1.1200 \\ -0.0138 & 0.9996 & -0.0131 & 0.0200 & 0 \\ -0.4927 & -0.0135 & 0.5072 & 0 & 0 \end{bmatrix}.$$

Hence,

$$\tilde{\boldsymbol{A}}_{11} = \begin{bmatrix} -0.0124 & -0.0102 & 0.0145 \\ 0.4662 & -0.0863 & -0.5326 \\ -0.0000 & 0.0139 & -0.0004 \end{bmatrix} \text{ and } \tilde{\boldsymbol{C}}_1 = \begin{bmatrix} -0.0004 & 0.0265 & -0.0003 \\ 0.0000 & -0.0008 & 0.0000 \\ -0.0138 & 0.9996 & -0.0131 \\ -0.7169 & -0.0190 & -0.6969 \end{bmatrix}.$$

Note that the pair $(\tilde{\boldsymbol{A}}_{11}, \tilde{\boldsymbol{C}}_1)$ is detectable and that the matrix $\tilde{\boldsymbol{A}}_{11}$ is unstable with its eigenvalues at $\{0.0000, -0.0495 + 0.1041j, -0.0495 - 0.1041j\}$. We proceed to stabilize $\tilde{\boldsymbol{A}}_{11}$. We select the desired eigenvalues to be located at $-3, -4, -5$. The gain matrix $\tilde{\boldsymbol{L}}_1$ such that $\text{eig}(\tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1) = \{-3, -4, -5\}$ is

$$\tilde{\boldsymbol{L}}_1 = \begin{bmatrix} 0.3866 & -0.0110 & 14.5695 & -3.5473 \\ 0.2097 & -0.0060 & 7.9007 & -0.2190 \\ -0.4028 & 0.0115 & -15.1794 & -2.0853 \end{bmatrix}.$$

We obtained the above gain matrix using MATLAB's command `place`. We next form the matrix $\tilde{\boldsymbol{L}}$ by adding two zero rows to $\tilde{\boldsymbol{L}}_1$ and compute

$$\begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_{3\times 2} \end{bmatrix} \boldsymbol{T}^{-1}\tilde{\boldsymbol{G}} = \begin{bmatrix} -0.0787 & -0.7280 & 14.5813 & -3.5473 \\ -0.8007 & -0.0248 & 7.9275 & -0.2190 \\ 0.1035 & -0.6854 & -15.1934 & -2.0853 \end{bmatrix}.$$

The reduced-order UIO has the form

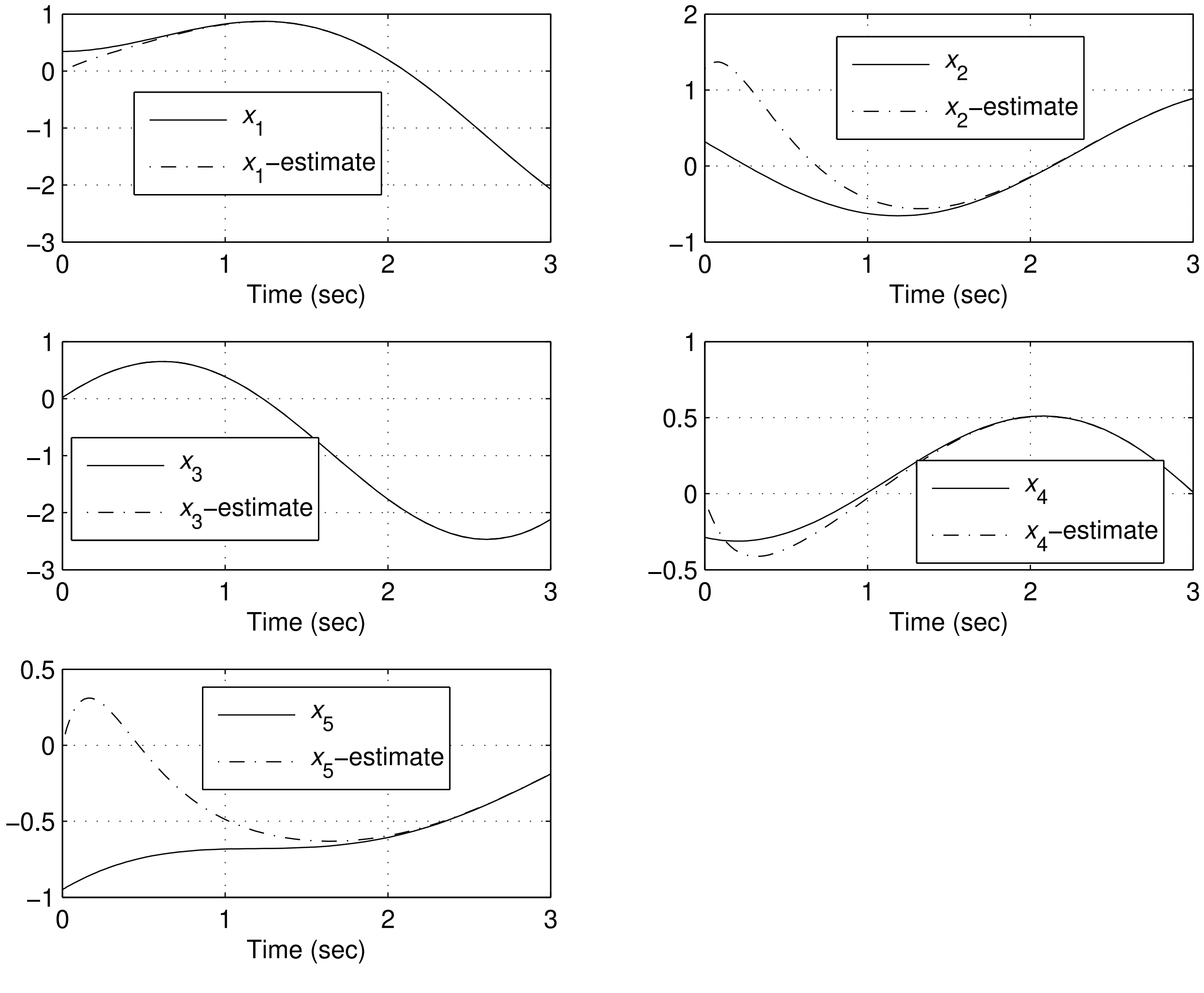


Figure 12.3: Plots of x_is and their estimates versus time for Example 12.2.

$$
\begin{aligned}
\dot{\tilde{\boldsymbol{q}}}_1 &= (\tilde{\boldsymbol{A}}_{11} - \tilde{\boldsymbol{L}}_1\tilde{\boldsymbol{C}}_1)\tilde{\boldsymbol{q}}_1 + \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0} \end{bmatrix} \boldsymbol{T}^{-1}\tilde{\boldsymbol{G}}\boldsymbol{y} \\
&= \begin{bmatrix} -2.3538 & -14.6523 & -2.2668 \\ 0.4186 & -7.9939 & -0.5818 \\ -1.7052 & 15.1589 & -1.6523 \end{bmatrix} \tilde{\boldsymbol{q}}_1 + \begin{bmatrix} -0.0787 & -0.7280 & 14.5813 & -3.5473 \\ -0.8007 & -0.0248 & 7.9275 & -0.2190 \\ 0.1035 & -0.6854 & -15.1934 & -2.0853 \end{bmatrix} \boldsymbol{y} \\
\tilde{\boldsymbol{x}} &= \boldsymbol{T} \begin{bmatrix} \boldsymbol{I}_3 \\ \boldsymbol{O}_{2\times 3} \end{bmatrix} \tilde{\boldsymbol{q}}_1 + \boldsymbol{M}\boldsymbol{y} \\
&= \begin{bmatrix} -0.7169 & -0.0190 & -0.6969 \\ -0.4930 & 0.0131 & 0.5069 \\ 0.0000 & -0.0008 & 0.0000 \\ -0.0138 & 0.9996 & -0.0131 \\ -0.4927 & -0.0135 & 0.5072 \end{bmatrix} \tilde{\boldsymbol{q}}_1 + \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.9993 & 0.0000 & -0.0265 & 0.0000 \\ 0.0000 & 1.0000 & 0.0008 & 0.0000 \\ -0.0265 & 0.0008 & 0.0007 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \boldsymbol{y}.
\end{aligned}
$$

In Figure 12.3, we show plots of the system state variables and their estimates versus time. The initial conditions of the plant were selected randomly to be equal to

$$
\boldsymbol{x}(0) = \begin{bmatrix} 0.0342 & 0.3200 & 0.0178 & -0.0287 & -0.2497 \end{bmatrix}^\top .
$$

The initial conditions of the observer were set to zero.

12.3.4 Relations With Other UIO Architectures

Our analysis, from the previous sections, can be extended to cover the case

$$
\dot{\boldsymbol{q}} = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{A}\boldsymbol{M}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1) + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{q} - \boldsymbol{C}\boldsymbol{M}\boldsymbol{y}), \tag{12.23}
$$

where the term $\boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{q} - \boldsymbol{C}\boldsymbol{M}\boldsymbol{y})$ is not premultiplied by $(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})$. This case leads to the observer analyzed in [57, 56] even though the approach there is quite different. Indeed, we can equivalently represent the dynamics of the proposed full-order observer as follows:

$$
\begin{aligned}
\dot{\boldsymbol{q}} &= ((\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\,\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\,\boldsymbol{q} + ([(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\,\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}]\,\boldsymbol{M} + \boldsymbol{L})\,\boldsymbol{y} + (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{C})\,\boldsymbol{B}_1\boldsymbol{u}_1 \\
&= (\boldsymbol{T}\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{q} + \boldsymbol{K}\boldsymbol{y} + \boldsymbol{T}\boldsymbol{B}_1\boldsymbol{u}_1, \\
\tilde{\boldsymbol{x}} &= \boldsymbol{q} + \boldsymbol{M}\boldsymbol{y},
\end{aligned}
$$

where, using the notation similar to that in [57, 56],

$$
\begin{aligned}
\boldsymbol{T} &= \boldsymbol{I} - \boldsymbol{M}\boldsymbol{C}, \quad \boldsymbol{K}_1 = \boldsymbol{L}, \\
\boldsymbol{K}_2 &= [\boldsymbol{T}\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}]\,\boldsymbol{M}, \quad \boldsymbol{K} = \boldsymbol{K}_1 + \boldsymbol{K}_2.
\end{aligned}
$$

In addition, the conditions for the existence of the full-order observer presented in [57, 56] and our observers are equivalent.

The observer given by (12.23) is also the same as the one proposed by Yang and Wilde [360] and further analyzed by Darouach, Zasadzinski, and Xu [71]. The connections are as follows: (i) $\boldsymbol{M}$ is called $-\boldsymbol{E}$ in [360, 71], (ii) $(\boldsymbol{I}-\boldsymbol{MC})$ corresponds to $\boldsymbol{P}$ there, (iii) $\boldsymbol{B}_1$ is $\boldsymbol{B}$ and $\boldsymbol{B}_2$ is $\boldsymbol{D}$ in [360, 71], (iv) $(\boldsymbol{I}-\boldsymbol{MC})(\boldsymbol{A}-\boldsymbol{LC})$ corresponds to $\boldsymbol{N}$.

A different approach is proposed by Hou and Müller [142] who first transform the system (12.3) into the form

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}_1\boldsymbol{u}_1+\begin{bmatrix}\boldsymbol{O}\\ \boldsymbol{I}_{m_2}\end{bmatrix}\boldsymbol{u}_2\\ &= \begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}+\begin{bmatrix}\boldsymbol{B}_{11}\\ \boldsymbol{B}_{12}\end{bmatrix}\boldsymbol{u}_1\\ &\quad+\begin{bmatrix}\boldsymbol{O}\\ \boldsymbol{I}_{m_2}\end{bmatrix}\boldsymbol{u}_2.\end{aligned}$$

Note that $\boldsymbol{x}_1$ in the new coordinates is independent of $\boldsymbol{u}_2$ and we have

$$\dot{\boldsymbol{x}}_1=\boldsymbol{A}_{11}\boldsymbol{x}_1+\boldsymbol{A}_{12}\boldsymbol{x}_2+\boldsymbol{B}_{11}\boldsymbol{u}_1. \tag{12.24}$$

Let

$$\boldsymbol{y}=\boldsymbol{C}\boldsymbol{x}=\begin{bmatrix}\boldsymbol{C}_1 & \boldsymbol{C}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix},$$

where $\boldsymbol{C}_2\in\mathbb{R}^{p\times m_2}$. Because by assumption, $\operatorname{rank}(\boldsymbol{C}\boldsymbol{B}_2)=\operatorname{rank}\boldsymbol{B}_2=m$, the submatrix $\boldsymbol{C}_2$ has a left inverse, $\boldsymbol{C}_2^\dagger$. Hence we can compute

$$\boldsymbol{x}_2=-\boldsymbol{C}_2^\dagger\boldsymbol{C}_1\boldsymbol{x}_1+\boldsymbol{C}_2^\dagger\boldsymbol{y}. \tag{12.25}$$

Substituting the above into (12.24) gives

$$\dot{\boldsymbol{x}}_1=\left(\boldsymbol{A}_{11}-\boldsymbol{A}_{12}\boldsymbol{C}_2^\dagger\boldsymbol{C}_1\right)\boldsymbol{x}_1+\boldsymbol{A}_{12}\boldsymbol{C}_2^\dagger\boldsymbol{y}+\boldsymbol{B}_{11}\boldsymbol{u}_1.$$

Hou and Müller [142] propose now to construct an observer for $\boldsymbol{x}_1$ using only known signals and then substitute the estimate of $\boldsymbol{x}_1$ into (12.25) to obtain an estimate of $\boldsymbol{x}_2$.

12.4 Sliding Mode UIO

12.4.1 Sliding Mode UIO Architecture

We present a sliding mode UIO following the exposition in our paper [151]. We assume that $\boldsymbol{u}_2$ is bounded, that is, there exists a nonnegative real number, ρ, such that

$$\|\boldsymbol{u}_2\| \leq \rho.$$

Let $\hat{\boldsymbol{x}}$ be an estimate of $\boldsymbol{x}$. Let $\boldsymbol{e}$ denote the estimation error, that is,

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t).$$

The observability of $(\boldsymbol{A}, \boldsymbol{C})$ implies the existence of a matrix $\boldsymbol{L} \in \mathbb{R}^{n\times p}$ such that the matrix $(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})$ has prescribed (symmetric with respect to the real axis) eigenvalues in the open left-half plane. Because $(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})$ is asymptotically stable, for any $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, there is a unique $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ such that

$$(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})^\top \boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}) = -\boldsymbol{Q}. \tag{12.26}$$

We choose $\boldsymbol{Q}$, if possible, so that for some $\boldsymbol{F} \in \mathbb{R}^{m_2\times p}$,

$$\boldsymbol{F}\boldsymbol{C} = \boldsymbol{B}_2^\top \boldsymbol{P}. \tag{12.27}$$

We need this technical condition to ensure the realizablity of the observer.

To proceed, we define the vector function

$$\boldsymbol{E}(\boldsymbol{e}, \eta) = \begin{cases} \eta \dfrac{\boldsymbol{F}\boldsymbol{C}\boldsymbol{e}}{\|\boldsymbol{F}\boldsymbol{C}\boldsymbol{e}\|_2} & \text{for} \quad \boldsymbol{F}\boldsymbol{C}\boldsymbol{e} \neq \boldsymbol{0} \\ \boldsymbol{0} & \text{for} \quad \boldsymbol{F}\boldsymbol{C}\boldsymbol{e} = \boldsymbol{0}, \end{cases}$$

where $\eta \geq \rho$ is a design parameter. In the case of single-input single-output plant, we can write

$$\boldsymbol{E}(\boldsymbol{e}, \eta) = \eta \operatorname{sign}(\boldsymbol{F}\boldsymbol{C}\boldsymbol{e}).$$

We note that

$$\boldsymbol{C}\boldsymbol{e} = \boldsymbol{C}(\boldsymbol{x} - \hat{\boldsymbol{x}}) = \boldsymbol{y} - \hat{\boldsymbol{y}}.$$

The vector function $\boldsymbol{E}(\boldsymbol{e}, \eta)$ is an essential ingredient of the sliding mode observer that we present next. When implementing the function $\boldsymbol{E}$, we use the output measurements $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, that is, instead of using $E(\boldsymbol{e}, \eta)$, we utilize

$$\boldsymbol{E}(\boldsymbol{y}, \hat{\boldsymbol{y}}, \eta) = \begin{cases} \eta \dfrac{\boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}})}{\|\boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}})\|_2} & \text{for} \quad \boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}}) \neq \boldsymbol{0} \\ \boldsymbol{0} & \text{for} \quad \boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}}) = \boldsymbol{0}. \end{cases}$$

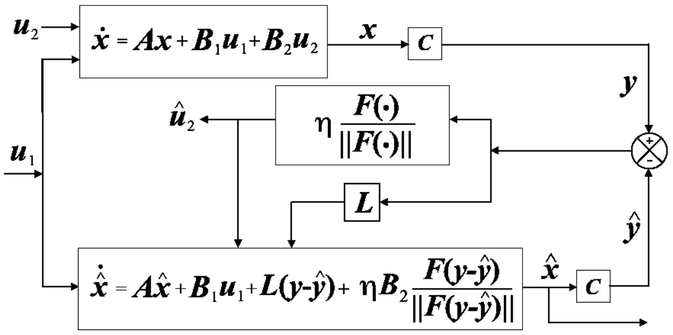


Figure 12.4: Sliding mode observer (12.28) can also be used as an unknown input estimator—see (12.33).

Hence for the case of single-input single-output plant, we have

$$\boldsymbol{E}(\hat{y}, \boldsymbol{e}, y) = \eta \operatorname{sign}\left(F(y - \hat{y})\right).$$

Using arguments similar to those found in [339], we can show that the state $\hat{\boldsymbol{x}}$ of the dynamical system

$$\boxed{\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}(\boldsymbol{y} - \hat{\boldsymbol{y}}) + \boldsymbol{B}_2\boldsymbol{E}(\boldsymbol{y}, \hat{\boldsymbol{y}}, \eta)} \tag{12.28}$$

for $\eta \geq \rho$, is an asymptotic estimate of the state $\boldsymbol{x}$ of the system described by (12.3) and (12.2), that is,

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \lim_{t\to\infty} \left(\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\right) = \boldsymbol{0}.$$

In Figure 12.4, we show a block diagram of the sliding mode observer given by (12.28). To prove the above statement using Lyapunov's type of arguments, first represent (12.28) as

$$\dot{\hat{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\,\hat{\boldsymbol{x}} + \boldsymbol{L}\boldsymbol{y} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{B}_2\boldsymbol{E}(\boldsymbol{e}, \eta).$$

Then construct the differential equation describing the dynamics of the estimation error $\boldsymbol{e}$,

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\,\boldsymbol{e} + \boldsymbol{B}_2\boldsymbol{u}_2 - \boldsymbol{B}_2\boldsymbol{E}(\boldsymbol{e}, \eta). \tag{12.29}$$

Then, show that

$$\frac{d}{dt}\left(\boldsymbol{e}^\top \boldsymbol{P}\boldsymbol{e}\right) = -\boldsymbol{e}^\top \boldsymbol{Q}\boldsymbol{e} < 0,$$

which implies

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0}.$$

It follows from the above that the estimation error is insensitive to the uncertainty modeled by the term $\boldsymbol{B}_2\boldsymbol{u}_2$. In summary, the design of the sliding mode UIO for a system modeled by the quadruple $(\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{C})$ can be thought of as finding a triple of matrices $(\boldsymbol{L}, \boldsymbol{F}, \boldsymbol{P})$ satisfying (12.26) and (12.27) for some $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$. Edwards and Spurgeon [92] (see also Saif and Xiong [291]) present necessary and sufficient conditions for the existence of the above observer, which are:

(i) $\operatorname{rank} \boldsymbol{B}_2 = \operatorname{rank} \boldsymbol{C}\boldsymbol{B}_2 = r$;

(ii) the system zeros of the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$ are in the open left-hand complex plane, that is,

$$\operatorname{rank} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}_2 \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} = n + r \ \text{ for all } \ s \ \text{ such that } \ \operatorname{Re}(s) \geq 0.$$

It is interesting to note that the above conditions are also necessary and sufficient for the existence of the observers with unknown inputs analyzed by us in the previous sections as well for the Edwards and Spurgeon UIO discussed later in Section 12.5.

The rank condition,

$$\operatorname{rank} \boldsymbol{B}_2 = \operatorname{rank} \boldsymbol{C}\boldsymbol{B}_2$$

is also referred to as the observer matching condition. This condition can be restrictive in some applications. To construct a sliding mode observer for systems that do not satisfy the observer matching condition, Kalsi et al. [171] propose a method where auxiliary outputs are generated using high-gain approximate differentiators and then employ the auxiliary outputs in the design of a sliding mode observer.

12.4.2 Reconstructing the Unknown Input

We show here how the UIO given by (12.28) can be used to reconstruct the unknown input $\boldsymbol{u}_2$. Consider the manifold,

$$\{\boldsymbol{e} : \boldsymbol{\sigma}(\boldsymbol{e}) = \boldsymbol{F}\boldsymbol{C}\boldsymbol{e} = \boldsymbol{0}\}. \tag{12.30}$$

We can always choose η such that the above manifold is attractive and it will be reached in a finite time. Thus, after the estimation error trajectory, $\boldsymbol{e}$, enters a sliding mode along the sliding surface, we have $\boldsymbol{\sigma}(\boldsymbol{e}) = \boldsymbol{0}$ and

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{F}\boldsymbol{C}\left(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}\right)\boldsymbol{e} + \boldsymbol{F}\boldsymbol{C}\boldsymbol{B}_2\boldsymbol{u}_2 - \boldsymbol{F}\boldsymbol{C}\boldsymbol{B}_2\boldsymbol{E}(\boldsymbol{e}, \eta) = \boldsymbol{0}. \tag{12.31}$$

Because $\boldsymbol{e} \to \boldsymbol{0}$ as $t \to \infty$, we obtain

$$\boldsymbol{F}\boldsymbol{C}\boldsymbol{B}_2\boldsymbol{u}_2 = \boldsymbol{F}\boldsymbol{C}\boldsymbol{B}_2\boldsymbol{E}(\boldsymbol{e}, \eta). \tag{12.32}$$

Recall that $\boldsymbol{F}\boldsymbol{C} = \boldsymbol{B}_2^\top \boldsymbol{P}$. Assuming that $\boldsymbol{B}_2$ is of full column rank implies that the square matrix $\boldsymbol{B}_2^\top \boldsymbol{P}\boldsymbol{B}_2$ is invertible. Hence, the steady-state condition (12.32) can be re-written as

$$\boldsymbol{B}_2^\top \boldsymbol{P}\boldsymbol{B}_2 \boldsymbol{u}_2 = \boldsymbol{B}_2^\top \boldsymbol{P}\boldsymbol{B}_2 \boldsymbol{E}(\boldsymbol{e}, \rho).$$

Therefore, the unknown input estimate is the output of the nonlinear term, $\boldsymbol{E}(\boldsymbol{e}, \rho)$, that is,

$$\boxed{\hat{\boldsymbol{u}}_2 = \boldsymbol{E}(\boldsymbol{e}, \rho)} \tag{12.33}$$

See Figure 12.4 for an illustration of (12.33). For a rigorous proof of the above result, see [171].

12.4.3 Sliding Mode UIO Analysis and Construction

We first present a lemma that will serve us as a design platform of the sliding mode observer for uncertain systems. The lemma is a minor modification of Lemma 1 of Corless and Tu [68] who proved it constructively using a singular value decomposition approach. We offer a different constructive proof using the Q-R decomposition.

Lemma 12.2 *For a triple* $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C}) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m_2} \times \mathbb{R}^{p\times n}$,

$$\mathit{rank}\, \boldsymbol{B}_2 = \mathit{rank}\, (\boldsymbol{C}\boldsymbol{B}_2) = r, \tag{12.34}$$

if and only if there exist nonsingular matrices $\boldsymbol{T}$ *and* $\boldsymbol{S}$ *such that*

$$\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}, \quad \boldsymbol{T}\boldsymbol{B}_2 = \begin{bmatrix} \boldsymbol{B}_{2_1} \\ \boldsymbol{O} \end{bmatrix}, \quad \boldsymbol{S}\boldsymbol{C}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{C}_{22} \end{bmatrix}, \tag{12.35}$$

where $\boldsymbol{A}_{11} \in \mathbb{R}^{r\times r}$, $\boldsymbol{A}_{22} \in \mathbb{R}^{(n-r)\times(n-r)}$, $\boldsymbol{B}_{2_1} \in \mathbb{R}^{r\times m_2}$, $\mathit{rank}\, \boldsymbol{B}_{2_1} = r$, *and* $\boldsymbol{C}_{22} \in \mathbb{R}^{(p-r)\times(n-r)}$.

Proof ($\Longrightarrow$) The proof is constructive. Using the Q-R decomposition applied to $\boldsymbol{B}_2$, we obtain

$$\boldsymbol{B}_2 = \boldsymbol{Q}_{B_2} \boldsymbol{R}_{B_2},$$

where $\boldsymbol{Q}_{B_2} \in \mathbb{R}^{n\times n}$ is an unitary matrix and the matrix $\boldsymbol{R}_{B_2} \in \mathbb{R}^{n\times m_2}$ is upper triangular where

$$\text{rank}\, \boldsymbol{R}_{B_2} = r.$$

Let $\boldsymbol{T}_1 = \boldsymbol{Q}_{B_2}^{-1}$. Then, we obtain

$$\boldsymbol{T}_1 \boldsymbol{B}_2 = \begin{bmatrix} \tilde{\boldsymbol{B}}_{2_1} \\ \boldsymbol{O} \end{bmatrix},$$

where $\tilde{\boldsymbol{G}}_1 \in \mathbb{R}^{r\times m_2}$. We next partition the matrix $\boldsymbol{C}\boldsymbol{T}_1^{-1}$ as follows,

$$\boldsymbol{C}\boldsymbol{T}_1^{-1} = \begin{bmatrix} \tilde{\boldsymbol{C}}_1 & \tilde{\boldsymbol{C}}_2 \end{bmatrix},$$

where $\tilde{\boldsymbol{C}}_1 \in \mathbb{R}^{p\times r}$. Note that

$$\boldsymbol{C}\boldsymbol{B}_2 = \left(\boldsymbol{C}\boldsymbol{T}_1^{-1}\right)(\boldsymbol{T}_1\boldsymbol{B}_2) = \tilde{\boldsymbol{C}}_1\tilde{\boldsymbol{B}}_{2_1}.$$

By the lemma hypothesis, $\operatorname{rank}\boldsymbol{B}_2 = \operatorname{rank}(\boldsymbol{C}\boldsymbol{B}_2) = r$. Hence

$$\operatorname{rank}\tilde{\boldsymbol{C}}_1 = r.$$

Applying the Q-R decomposition to $\tilde{\boldsymbol{C}}_1$ yields

$$\tilde{\boldsymbol{C}}_1 = \boldsymbol{Q}_{\tilde{C}_1}\boldsymbol{R}_{\tilde{C}_1},$$

where

$$\boldsymbol{R}_{\tilde{C}_1} = \begin{bmatrix} \boldsymbol{C}_{11} \\ \boldsymbol{O} \end{bmatrix} \quad \text{and} \quad \det\boldsymbol{C}_{11} \neq 0.$$

Note that $\boldsymbol{C}_{11} \in \mathbb{R}^{r\times r}$. Let $\boldsymbol{S} = \boldsymbol{Q}_{\tilde{C}_1}^{-1}$, then,

$$\boldsymbol{S}\boldsymbol{C}\boldsymbol{T}_1^{-1} = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{O} & \boldsymbol{C}_{22} \end{bmatrix}.$$

Postmultiplying $\boldsymbol{S}\boldsymbol{C}\boldsymbol{T}_1^{-1}$ by

$$\boldsymbol{T}_2^{-1} = \begin{bmatrix} \boldsymbol{C}_{11}^{-1} & -\boldsymbol{C}_{11}^{-1}\boldsymbol{C}_{12} \\ \boldsymbol{O} & \boldsymbol{I}_{n-r} \end{bmatrix}$$

gives

$$\boldsymbol{S}\boldsymbol{C}\boldsymbol{T}_1^{-1}\boldsymbol{T}_2^{-1} = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{O} & \boldsymbol{C}_{22} \end{bmatrix}\begin{bmatrix} \boldsymbol{C}_{11}^{-1} & -\boldsymbol{C}_{11}^{-1}\boldsymbol{C}_{12} \\ \boldsymbol{O} & \boldsymbol{I}_{n-r} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{C}_{22} \end{bmatrix}.$$

We then have $\boldsymbol{T} = \boldsymbol{T}_2\boldsymbol{T}_1$.

($\Longleftarrow$) By inspection.

$\square$

We next show that the systems

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} &=& \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_2\boldsymbol{u}_2 \\ \boldsymbol{y} &=& \boldsymbol{C}\boldsymbol{x} \end{array}\right\}$$

and

$$\left.\begin{array}{rcl}\dot{\tilde{\boldsymbol{x}}} & = & \boldsymbol{TAT}^{-1}\tilde{\boldsymbol{x}}+\boldsymbol{TB}_2\boldsymbol{u}_2\\ \tilde{\boldsymbol{y}} & = & \boldsymbol{SCT}^{-1}\boldsymbol{x}\end{array}\right\}$$

have the same system zeros, that is, their system matrices have the same rank for all $s \in \mathbb{C}$, where $\mathbb{C}$ is the set of complex numbers. To prove the above statement, we note the following relation between the system matrices of the above models:

$$\begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{TAT}^{-1} & \boldsymbol{TB}_2 \\ \boldsymbol{SCT}^{-1} & \boldsymbol{O} \end{bmatrix} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}_2 \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{T}^{-1} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_q \end{bmatrix}. \tag{12.36}$$

Suppose now that we have two matrices, $\boldsymbol{X} \in \mathbb{R}^{r\times s}$ and $\boldsymbol{Y} \in \mathbb{R}^{s\times t}$. Then, we have the following bounds on the rank of the product of these matrices:

$$\operatorname{rank}\boldsymbol{X} + \operatorname{rank}\boldsymbol{Y} - s \le \operatorname{rank}(\boldsymbol{XY}) \le \min(\operatorname{rank}\boldsymbol{X}, \operatorname{rank}\boldsymbol{Y}).$$

The above are referred to as the Sylvester's inequalities—see, for example, Gantmacher [110, pages 65, 66] for their proof or Section A.10 and Exercise A.15 in the Appendix. Applying the Sylvester's inequalities to (12.36), we conclude that

$$\operatorname{rank}\begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{TAT}^{-1} & \boldsymbol{TB}_2 \\ \boldsymbol{SCT}^{-1} & \boldsymbol{O} \end{bmatrix} = \operatorname{rank}\begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}_2 \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} \quad \text{for all} \quad s \in \mathbb{C}. \tag{12.37}$$

Lemma 12.3 *Assume that rank* $\boldsymbol{B}_2 =$ *rank* $(\boldsymbol{CB}_2) = r$. *Then, the pair* $(\boldsymbol{A}_{22}, \boldsymbol{C}_{22})$ *is detectable if and only if*

$$\mathit{rank}\begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}_2 \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} = n + r \quad \textit{for all} \quad s \quad \textit{such that} \quad \mathit{Re}(s) \ge 0. \tag{12.38}$$

Proof By assumption, rank $\boldsymbol{B}_2 = \operatorname{rank}(\boldsymbol{CB}_2) = r$. By Lemma 12.2, the above condition is equivalent to the existence of nonsingular matrices $\boldsymbol{T}$ and $\boldsymbol{S}$ such that

$$\boldsymbol{TAT}^{-1} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}, \quad \boldsymbol{TB}_2 = \begin{bmatrix} \boldsymbol{B}_{2_1} \\ \boldsymbol{O} \end{bmatrix}, \quad \boldsymbol{SCT}^{-1} = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{C}_{22} \end{bmatrix},$$

where $\boldsymbol{B}_{2_1} \in \mathbb{R}^{r\times m_2}$ and rank $\boldsymbol{B}_{2_1} = r$. Then, by (12.37) for any $s \in \mathbb{C}$,

$$\begin{aligned}
\operatorname{rank}\begin{bmatrix} s\boldsymbol{I}_r - \boldsymbol{A}_{11} & -\boldsymbol{A}_{12} & \boldsymbol{B}_{2_1} \\ -\boldsymbol{A}_{21} & s\boldsymbol{I}_{n-r} - \boldsymbol{A}_{22} & \boldsymbol{O} \\ \boldsymbol{I}_r & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{C}_{22} & \boldsymbol{O} \end{bmatrix} &= \operatorname{rank}\begin{bmatrix} -\boldsymbol{A}_{12} & \boldsymbol{B}_{2_1} \\ s\boldsymbol{I}_{n-r} - \boldsymbol{A}_{22} & \boldsymbol{O} \\ \boldsymbol{C}_{22} & \boldsymbol{O} \end{bmatrix} + r \\
&= \operatorname{rank}\begin{bmatrix} s\boldsymbol{I}_{n-r} - \boldsymbol{A}_{22} \\ \boldsymbol{C}_{22} \end{bmatrix} + 2r.
\end{aligned}$$

It follows from the above that the pair $(\boldsymbol{A}_{22}, \boldsymbol{C}_{22})$ is detectable if and only if the rank condition (12.38) holds.

□

Note that if $m_2 = p = r$, then rank $\begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}_2 \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} = n+r$ for all s such that $\mathrm{Re}(s) \geq 0$ if and only if the matrix $\boldsymbol{A}_{22}$ is asymptotically stable.

To proceed, we need the following technical lemma.

Lemma 12.4 *For a given matrix $\boldsymbol{V} \in \mathbb{R}^{n\times m}$,*

$$\ker(\boldsymbol{V}) = \ker(\boldsymbol{V}^\top \boldsymbol{V}).$$

Proof See Exercise 12.5.

□

The following theorem appears in Corless and Tu [68, Lemma 3, page 760]. A related result was obtained by Edwards and Spurgeon [92, Proposition 6.2, page 138]. An algorithm for constructing matrices $\boldsymbol{L}$, $\boldsymbol{F}$, and $\boldsymbol{P}$ that are essential ingredients of the sliding mode observer for the uncertain systems is contained in the proof of the theorem.

Theorem 12.3 *There exists a triple of matrices $(\boldsymbol{L}, \boldsymbol{F}, \boldsymbol{P}) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{m_2\times p} \times \mathbb{R}^{n\times n}$ such that*

$$(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})^\top \boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}) < 0 \tag{12.39}$$

and

$$\boldsymbol{F}\boldsymbol{C} = \boldsymbol{B}_2^\top \boldsymbol{P} \tag{12.40}$$

if and only if

(i) *rank $\boldsymbol{B}_2 =$ rank $(\boldsymbol{C}\boldsymbol{B}_2) = r$;*

(ii) *the system zeros of the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$ are in the open left-hand complex plane, that is,*

$$\text{rank} \begin{bmatrix} s\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}_2 \\ \boldsymbol{C} & \boldsymbol{O} \end{bmatrix} = n + r \ \text{ for all } \ s \ \text{ such that } \ \mathrm{Re}(s) \geq 0.$$

Proof ($\Longleftarrow$) We follow the arguments of Corless and Tu [68]. By Lemma 12.2, the condition, rank $\boldsymbol{B}_2 =$ rank $(\boldsymbol{C}\boldsymbol{B}_2) = r$, is equivalent to the existence of nonsingular matrices $\boldsymbol{T}$ and $\boldsymbol{S}$ such that

$$\hat{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}, \hat{\boldsymbol{B}}_2 = \boldsymbol{T}\boldsymbol{B}_2 = \begin{bmatrix} \boldsymbol{B}_{21} \\ \boldsymbol{O} \end{bmatrix}, \quad \hat{\boldsymbol{C}} = \boldsymbol{S}\boldsymbol{C}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{C}_{22} \end{bmatrix},$$

where $\boldsymbol{B}_{2_1} \in \mathbb{R}^{r\times m_2}$ and $\operatorname{rank} \boldsymbol{B}_{2_1} = r$. Let

$$\hat{\boldsymbol{P}} = \boldsymbol{T}^{-\top}\boldsymbol{P}\boldsymbol{T}^{-1}, \quad \hat{\boldsymbol{L}} = \boldsymbol{T}\boldsymbol{L}\boldsymbol{S}^{-1}, \quad \text{and} \quad \hat{\boldsymbol{F}} = \boldsymbol{F}\boldsymbol{S}^{-1}. \tag{12.41}$$

To proceed, note that condition **(ii)** is equivalent to the existence of a matrix $\boldsymbol{L}_{22}$ such that the eigenvalues of $(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})$ are all in the open left-half complex plane. Then, for any symmetric positive positive definite $\boldsymbol{Q}_{22}$, the symmetric solution $\boldsymbol{P}_{22}$ to the Lyapunov matrix equation,

$$(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{22} + \boldsymbol{P}_{22}(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22}) = -\boldsymbol{Q}_{22},$$

is also positive definite. Let

$$\hat{\boldsymbol{L}} = \begin{bmatrix} \kappa \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{L}_{22} \end{bmatrix},$$

where $\kappa > 0$ is a design parameter whose lower bound is determined in the following considerations. We have

$$\begin{aligned} \hat{\boldsymbol{A}} - \hat{\boldsymbol{L}}\hat{\boldsymbol{C}} &= \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} - \begin{bmatrix} \kappa \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{L}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{C}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_{11} - \kappa \boldsymbol{I}_r & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22} \end{bmatrix}. \end{aligned}$$

Let

$$\hat{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{P}_{22} \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{12}^\top & \boldsymbol{Q}_{22} \end{bmatrix}.$$

Using the above, we obtain

$$\begin{aligned} -\hat{\boldsymbol{Q}} &= (\hat{\boldsymbol{A}} - \hat{\boldsymbol{L}}\hat{\boldsymbol{C}})^\top \hat{\boldsymbol{P}} + \hat{\boldsymbol{P}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{L}}\hat{\boldsymbol{C}}) \\ &= \begin{bmatrix} \boldsymbol{A}_{11}^\top - \kappa \boldsymbol{I}_r & \boldsymbol{A}_{21}^\top \\ \boldsymbol{A}_{12}^\top & (\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{P}_{22} \end{bmatrix} \\ &\quad + \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{P}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{11} - \kappa \boldsymbol{I}_r & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_{11}^\top - \kappa \boldsymbol{I}_r & \boldsymbol{A}_{21}^\top \boldsymbol{P}_{22} \\ \boldsymbol{A}_{12}^\top & (\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{22} \end{bmatrix} + \begin{bmatrix} \boldsymbol{A}_{11} - \kappa \boldsymbol{I}_r & \boldsymbol{A}_{12} \\ \boldsymbol{P}_{22}\boldsymbol{A}_{21} & \boldsymbol{P}_{22}(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22}) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_{11}^\top + \boldsymbol{A}_{11} - 2\kappa \boldsymbol{I}_r & \boldsymbol{A}_{21}^\top \boldsymbol{P}_{22} + \boldsymbol{A}_{12} \\ \boldsymbol{A}_{12}^\top + \boldsymbol{P}_{22}\boldsymbol{A}_{21} & (\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{22} + \boldsymbol{P}_{22}(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22}) \end{bmatrix} \\ &= \begin{bmatrix} -\boldsymbol{Q}_{11} & -\boldsymbol{Q}_{12} \\ -\boldsymbol{Q}_{12}^\top & -\boldsymbol{Q}_{22} \end{bmatrix}, \end{aligned} \tag{12.42}$$

where $\boldsymbol{Q}_{22} = -\left((\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{22} + \boldsymbol{P}_{22}(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})\right)$ is positive definite by construction. Our goal is to obtain a lower bound on the parameter κ that would yield a positive definite $\hat{\boldsymbol{Q}}$. Using the Schur complement of the positive definite $\boldsymbol{Q}_{22}$, we have that $\hat{\boldsymbol{Q}}$ is positive definite if and only if

$$\boldsymbol{Q}_{11} > \boldsymbol{Q}_{12}\boldsymbol{Q}_{22}^{-1}\boldsymbol{Q}_{12}^\top.$$

Employing the above in (12.42), we obtain

$$2\kappa \boldsymbol{I}_r - \left(\boldsymbol{A}_{11}^\top + \boldsymbol{A}_{11}\right) > \left(\boldsymbol{A}_{21}^\top \boldsymbol{P}_{22} + \boldsymbol{A}_{12}\right) \boldsymbol{Q}_{22}^{-1} \left(\boldsymbol{A}_{12}^\top + \boldsymbol{P}_{22}\boldsymbol{A}_{21}\right).$$

Hence, $\hat{\boldsymbol{Q}}$ is positive definite if

$$\kappa > \frac{1}{2}\lambda_{\max}\left(\boldsymbol{A}_{11}^\top + \boldsymbol{A}_{11} + \left(\boldsymbol{A}_{21}^\top \boldsymbol{P}_{22} + \boldsymbol{A}_{12}\right) \boldsymbol{Q}_{22}^{-1} \left(\boldsymbol{A}_{12}^\top + \boldsymbol{P}_{22}\boldsymbol{A}_{21}\right)\right).$$

Let

$$\hat{\boldsymbol{F}} = \left[\begin{array}{cc} \boldsymbol{B}_{2_1}^\top & \boldsymbol{O} \end{array}\right].$$

Then, it is easy to see that

$$(\hat{\boldsymbol{A}} - \hat{\boldsymbol{L}}\hat{\boldsymbol{C}})^\top \hat{\boldsymbol{P}} + \hat{\boldsymbol{P}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{L}}\hat{\boldsymbol{C}}) < 0, \tag{12.43}$$

and

$$\hat{\boldsymbol{F}}\hat{\boldsymbol{C}} = \hat{\boldsymbol{B}}_2^\top \hat{\boldsymbol{P}}, \tag{12.44}$$

which are conditions (12.39) and (12.40) in the new basis. Hence, the proof of the sufficiency conditions for the existence of the desired triple of matrices $(\boldsymbol{L}, \boldsymbol{F}, \boldsymbol{P})$ is complete.
($\Longrightarrow$) We assume that there exists a triple of matrices $(\boldsymbol{L}, \boldsymbol{F}, \boldsymbol{P})$ such that the conditions

$$(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})^\top \boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}) < 0 \quad \text{and} \quad \boldsymbol{F}\boldsymbol{C} = \boldsymbol{B}_2^\top \boldsymbol{P}$$

are satisfied. Because $\boldsymbol{P}$ is symmetric positive definite, it can be represented as

$$\boldsymbol{P} = \boldsymbol{P}^{\frac{1}{2}}\boldsymbol{P}^{\frac{1}{2}},$$

where $\boldsymbol{P}^{\frac{1}{2}} \in \mathbb{R}^{n\times n}$ is also symmetric positive definite and hence nonsingular. We refer to $\boldsymbol{P}^{\frac{1}{2}}$ as a square root of $\boldsymbol{P}$—see, for example, Luenberger [224, page 459] or Section A.4 in the Appendix. We then have

$$\boldsymbol{B}_2^\top \boldsymbol{P}\boldsymbol{B}_2 = \boldsymbol{B}_2^\top \boldsymbol{P}^{\frac{1}{2}}\boldsymbol{P}^{\frac{1}{2}}\boldsymbol{B}_2 = \left(\boldsymbol{P}^{\frac{1}{2}}\boldsymbol{B}_2\right)^\top \left(\boldsymbol{P}^{\frac{1}{2}}\boldsymbol{B}_2\right).$$

By Lemma 12.4 and the fact that $\boldsymbol{P}^{\frac{1}{2}}$ is nonsingular,

$$\ker\left(\boldsymbol{B}_2^\top \boldsymbol{P}\boldsymbol{B}_2\right) = \ker\left(\boldsymbol{P}^{\frac{1}{2}}\boldsymbol{B}_2\right) = \ker(\boldsymbol{B}_2).$$

Therefore,

$$\operatorname{rank}\left(\boldsymbol{B}_2^\top \boldsymbol{P}\boldsymbol{B}_2\right) = \operatorname{rank}\boldsymbol{B}_2.$$

Using the above, the condition, $\boldsymbol{B}_2^\top\boldsymbol{P} = \boldsymbol{F}\boldsymbol{C}$, and the Sylvester's inequality, we obtain

$$\operatorname{rank}\boldsymbol{B}_2 = \operatorname{rank}\left(\boldsymbol{B}_2^\top \boldsymbol{P}\boldsymbol{B}_2\right) = \operatorname{rank}\left(\boldsymbol{F}\boldsymbol{C}\boldsymbol{B}_2\right) \le \operatorname{rank}\left(\boldsymbol{C}\boldsymbol{B}_2\right) \le \operatorname{rank}\boldsymbol{B}_2.$$

Hence,

$$\operatorname{rank}\left(\boldsymbol{C}\boldsymbol{B}_2\right) = \operatorname{rank}\boldsymbol{B}_2.$$

Our goal now is to show that conditions (12.39) and (12.40) imply that the system zeros of the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$ are in the open left-hand complex plane. Because $\operatorname{rank}(\boldsymbol{C}\boldsymbol{B}_2) = \operatorname{rank}\boldsymbol{B}_2$, by Lemma 12.2, we can construct transformation matrices $\boldsymbol{T}$ and $\boldsymbol{S}$ so that the transformed matrices $\boldsymbol{A}$, $\boldsymbol{B}_2$, and $\boldsymbol{C}$ take the form given by (12.35). By Lemma 12.3, the system zeros of the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$ are in the open left-hand complex plane if and only the pair $(\boldsymbol{A}_{22}, \boldsymbol{C}_{22})$ is detectable. Let matrices $\hat{\boldsymbol{P}}$, $\hat{\boldsymbol{L}}$, and $\hat{\boldsymbol{F}}$ be defined by (12.41), where

$$\hat{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{P}_{11} & \boldsymbol{P}_{12} \\ \boldsymbol{P}_{21} & \boldsymbol{P}_{22} \end{bmatrix}, \quad \hat{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{L}_{12} \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} \end{bmatrix}, \quad \text{and} \quad \hat{\boldsymbol{F}} = \begin{bmatrix} \boldsymbol{F}_1 & \boldsymbol{F}_2 \end{bmatrix}.$$

Because $\hat{\boldsymbol{P}}$ is symmetric positive definite, the submatrix $\boldsymbol{P}_{22} \in \mathbb{R}^{(n-r)\times(n-r)}$ is symmetric positive definite and hence invertible. We write down the corresponding 2-by-2 block equation of inequality (12.43),

$$(\boldsymbol{A}_{12} - \boldsymbol{L}_{12}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{12} + (\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{22} + \boldsymbol{P}_{21}(\boldsymbol{A}_{12} - \boldsymbol{L}_{12}\boldsymbol{C}_{22}) + \boldsymbol{P}_{22}(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22}) < 0. \tag{12.45}$$

Equation (12.44) yields

$$\boldsymbol{F}_2\boldsymbol{C}_{22} = \boldsymbol{B}_{2_1}^\top \boldsymbol{P}_{12},$$

where $\boldsymbol{B}_{2_1}$ has a full row rank. Therefore, $\boldsymbol{B}_{2_1}\boldsymbol{B}_{2_1}^\top \in \mathbb{R}^{r\times r}$ is invertible and we have

$$\boldsymbol{P}_{12} = \left(\boldsymbol{B}_{2_1}\boldsymbol{B}_{2_1}^\top\right)^{-1} \boldsymbol{B}_{2_1}\boldsymbol{F}_2\boldsymbol{C}_{22}. \tag{12.46}$$

Note that

$$\boldsymbol{P}_{21} = \boldsymbol{P}_{12}^\top = \boldsymbol{C}_{22}^\top \boldsymbol{F}_2^\top \boldsymbol{B}_{2_1}^\top \left(\boldsymbol{B}_{2_1}\boldsymbol{B}_{2_1}^\top\right)^{-1}.$$

Substituting the above into (12.45), performing manipulations and denoting

$$\tilde{\boldsymbol{L}}_{22} = \boldsymbol{L}_{22} - \boldsymbol{P}_{22}^{-1}(\boldsymbol{A}_{12} - \boldsymbol{L}_{12}\boldsymbol{C}_{22})^\top \left(\boldsymbol{B}_{2_1}\boldsymbol{B}_{2_1}^\top\right)^{-1} \boldsymbol{B}_{2_1}\boldsymbol{F}_2,$$

we obtain

$$(\boldsymbol{A}_{22} - \tilde{\boldsymbol{L}}_{22}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{22} + \boldsymbol{P}_{22}(\boldsymbol{A}_{22} - \tilde{\boldsymbol{L}}_{22}\boldsymbol{C}_{22}) < 0.$$

Because $\boldsymbol{P}_{22}$ is symmetric and positive definite, the above implies that $(\boldsymbol{A}_{22} - \tilde{\boldsymbol{L}}_{22}\boldsymbol{C}_{22})$ is asymptotically stable, which in turn implies that the pair $(\boldsymbol{A}_{22}, \boldsymbol{C}_{22})$ is detectable. The proof of the theorem is now complete.

□

We now summarize the above considerations in the form of the following design algorithm.

Algorithm 12.2 Sliding-Mode UIO Design

Given a quadruple of matrices $(\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{C})$ modeling the plant.

1. Check that the rank condition, rank $(\boldsymbol{C}\boldsymbol{B}_2) = \operatorname{rank} \boldsymbol{B}_2$, is satisfied.
   If rank $(\boldsymbol{C}\boldsymbol{B}_2) \neq \operatorname{rank} \boldsymbol{B}_2$, the sliding-mode observer cannot be constructed, STOP.

2. Transform the triple $(\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{B}_2)$.
   Use the method of the proof of Lemma 12.2 to construct nonsingular matrices $\boldsymbol{T}$ and $\boldsymbol{S}$ and compute

$$\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}, \quad \boldsymbol{T}\boldsymbol{B}_2 = \begin{bmatrix} \boldsymbol{B}_{2_1} \\ \boldsymbol{O} \end{bmatrix}, \quad \boldsymbol{S}\boldsymbol{C}\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{C}_{22} \end{bmatrix},$$

   where $\boldsymbol{B}_{2_1} \in \mathbb{R}^{r \times m_2}$ and rank $\boldsymbol{B}_{2_1} = r$.

3. Check detectability of $(\boldsymbol{A}_{22}, \boldsymbol{C}_{22})$.
   If the pair $(\boldsymbol{A}_{22}, \boldsymbol{C}_{22})$ is not detectable, the sliding-mode observer cannot be constructed, STOP.

4. Construct a matrix $\boldsymbol{L}_{22}$ so that the eigenvalues of $(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})$ are in the open left-half plane.

5. Choose a positive definite $\boldsymbol{Q}_{22} \in \mathbb{R}^{(n-r)\times(n-r)}$ and solve for positive definite $\boldsymbol{P}_{22}$ the Lyapunov matrix equation,

$$(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{22} + \boldsymbol{P}_{22}(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22}) = -\boldsymbol{Q}_{22}.$$

6. Choose κ that satisfies the condition,

$$\kappa > \frac{1}{2}\lambda_{\max}\left(\boldsymbol{A}_{11}^\top + \boldsymbol{A}_{11} + \left(\boldsymbol{A}_{21}^\top \boldsymbol{P}_{22} + \boldsymbol{A}_{12}\right)\boldsymbol{Q}_{22}^{-1}\left(\boldsymbol{A}_{12}^\top + \boldsymbol{P}_{22}\boldsymbol{A}_{21}\right)\right).$$

7. Construct,

$$\hat{\boldsymbol{L}} = \begin{bmatrix} \kappa \boldsymbol{I}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{L}_{22} \end{bmatrix}, \quad \hat{\boldsymbol{F}} = \begin{bmatrix} \boldsymbol{B}_{2_1}^\top & \boldsymbol{O} \end{bmatrix}.$$

8. Compute

$$\boldsymbol{L} = \boldsymbol{T}^{-1}\hat{\boldsymbol{L}}\boldsymbol{S}, \quad \boldsymbol{F} = \hat{\boldsymbol{F}}\boldsymbol{S}.$$

9. Construct the observer,

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{L}(\boldsymbol{y} - \hat{\boldsymbol{y}}) + \boldsymbol{B}_2\boldsymbol{E}(\boldsymbol{y}, \hat{\boldsymbol{y}}, \eta),$$

where

$$\boldsymbol{E}(\boldsymbol{y}, \hat{\boldsymbol{y}}, \eta) = \begin{cases} \eta \frac{\boldsymbol{F}(\boldsymbol{y}-\hat{\boldsymbol{y}})}{\|\boldsymbol{F}(\boldsymbol{y}-\hat{\boldsymbol{y}})\|_2} & \text{for} \quad \boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}}) \neq \boldsymbol{0} \\ \boldsymbol{0} & \text{for} \quad \boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}}) = \boldsymbol{0}. \end{cases}$$

Note that $\boldsymbol{C}\boldsymbol{e} = \boldsymbol{C}(\boldsymbol{x} - \hat{\boldsymbol{x}}) = \boldsymbol{y} - \hat{\boldsymbol{y}}$.

Example 12.3 We consider the same fifth-order lateral axis model of an L-1011 fixed-wing aircraft that we considered in Example 12.2. We have rank $(\boldsymbol{C}\boldsymbol{B}_2) = \operatorname{rank} \boldsymbol{B}_2$. We then use Lemma 12.2 to compute the transformation matrices T and S,

$$\boldsymbol{T} = \begin{bmatrix} -0.0000 & -0.9106 & 0.4125 & 0.0245 & 0.9106 \\ 0.0000 & -0.4124 & -0.9110 & 0.0103 & 0.4124 \\ -0.9998 & -0.0005 & 0.0000 & -0.0179 & 0 \\ -0.0179 & 0.0265 & -0.0008 & 0.9995 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

and

$$\boldsymbol{S} = \begin{bmatrix} -0.9106 & 0.4125 & 0.0245 & -0.0000 \\ -0.4124 & -0.9110 & 0.0103 & 0.0000 \\ 0.0265 & -0.0008 & 0.9996 & -0.0000 \\ 0.0000 & 0.0000 & -0.0000 & 1.0000 \end{bmatrix}.$$

We next transform the given model into the new coordinates and find the pair$(\boldsymbol{A}_{22}, \boldsymbol{C}_{22})$, where

$$\boldsymbol{A}_{22} = \begin{bmatrix} 0.0006 & 0.0025 & 0.0178 \\ -0.0368 & -0.0993 & -0.9971 \\ -0.0002 & 0.0133 & -0.0004 \end{bmatrix} \quad \text{and} \quad \boldsymbol{C}_{22} = \begin{bmatrix} -0.0179 & 0.9998 & -0.0265 \\ -0.9998 & -0.0179 & -0.0000 \end{bmatrix}.$$

The pair $(\boldsymbol{A}_{22}, \boldsymbol{C}_{22})$ is detectable and we use MATLAB's `place` function to construct the gain matrix $\boldsymbol{L}_{22}$ so that the eigenvalues of the matrix $(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})$ are located at $-3, -4, -5$, where

$$\boldsymbol{L}_{22} = \begin{bmatrix} -0.1441 & -4.0022 \\ 7.5018 & 0.0744 \\ -14.9883 & -0.4586 \end{bmatrix}.$$

We then solve the Lyapunov matrix equation,

$$(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})^\top \boldsymbol{P}_{22} + \boldsymbol{P}_{22}(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22}) = -\boldsymbol{I}_3,$$

to obtain

$$\boldsymbol{P}_{22} = \begin{bmatrix} 0.1273 & -0.0445 & -0.0236 \\ -0.0445 & 0.9995 & 0.4735 \\ -0.0236 & 0.4735 & 0.3059 \end{bmatrix}.$$

After that we compute

$$\frac{1}{2}\lambda_{\max}\left(\boldsymbol{A}_{11}^\top + \boldsymbol{A}_{11} + \left(\boldsymbol{A}_{21}^\top \boldsymbol{P}_{22} + \boldsymbol{A}_{12}\right)\boldsymbol{Q}_{22}^{-1}\left(\boldsymbol{A}_{12}^\top + \boldsymbol{P}_{22}\boldsymbol{A}_{21}\right)\right) = 12.6790$$

and select

$$\kappa = 13.6790.$$

Finally we construct

$$\boldsymbol{L} = \boldsymbol{T}^{-1}\hat{\boldsymbol{L}}\boldsymbol{S} = \begin{bmatrix} 0.0003 & -0.0000 & 0.0100 & 4.0002 \\ 13.2773 & 0.0115 & -15.1363 & -0.4544 \\ 0.0001 & 13.6790 & 0.0044 & -0.0001 \\ -0.1532 & 0.0044 & 7.9048 & 0.1580 \\ -0.3976 & 0.0114 & -14.9831 & -0.4586 \end{bmatrix}$$

and

$$\boldsymbol{F} = \hat{\boldsymbol{F}}\boldsymbol{S} = \begin{bmatrix} -0.7440 & 0.3370 & 0.0200 & -0.0000 \\ -0.0320 & -1.1200 & -0.0000 & 0.0000 \end{bmatrix}.$$

Let $\eta = 7$. We then construct the UIO,

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{L}(\boldsymbol{y} - \hat{\boldsymbol{y}}) + \boldsymbol{B}_2\tilde{\boldsymbol{E}}(\boldsymbol{y}, \hat{\boldsymbol{y}}, \eta),$$

where

$$\tilde{\boldsymbol{E}}(\boldsymbol{y}, \hat{\boldsymbol{y}}, \eta) = \begin{cases} \eta\frac{\boldsymbol{F}(\boldsymbol{y}-\hat{\boldsymbol{y}})}{\|\boldsymbol{F}(\boldsymbol{y}-\hat{\boldsymbol{y}})\|+\mu} & \text{for} \quad \boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}}) \neq \boldsymbol{0} \\ \boldsymbol{0} & \text{for} \quad \boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}}) = \boldsymbol{0}, \end{cases}$$

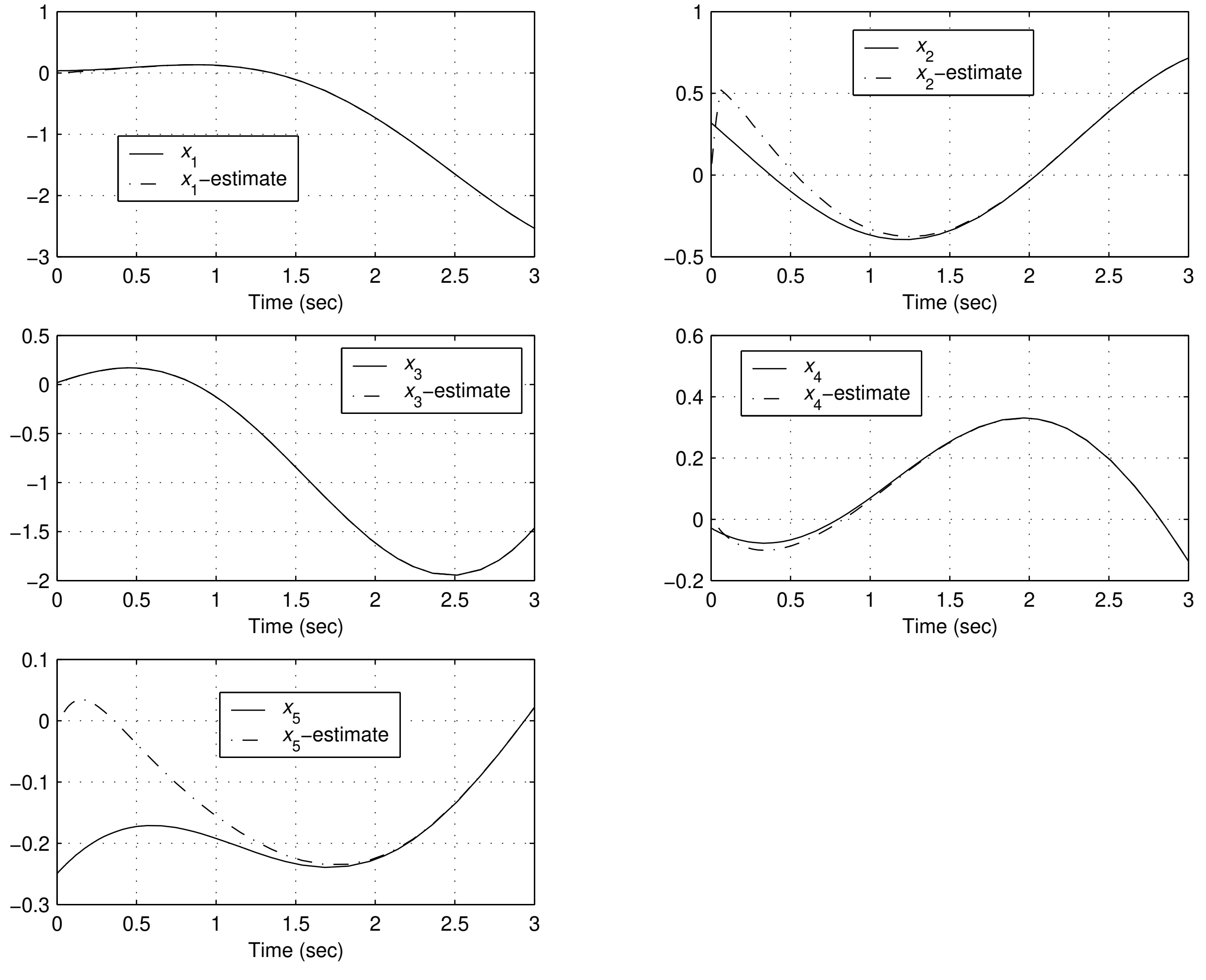


Figure 12.5: Plots of x_is and their estimates versus time for Example 12.3.

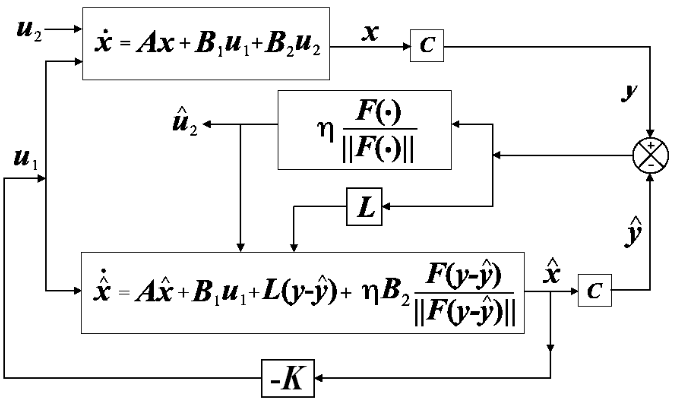


Figure 12.6: Sliding mode observer (12.28) as a part of the combined controller-observer compensator and an unknown input estimator.

where $\mu = 0.0005$. The parameter μ was introduced to smooth out the discontinuity to facilitate simulations. In Figure 12.5, we show plots of the system state variables and their estimates versus time. The initial conditions of the plant were selected randomly to be equal to

$$\boldsymbol{x}(0) = \begin{bmatrix} 0.0342 & 0.3200 & 0.0178 & -0.0287 & -0.2497 \end{bmatrix}^\top .$$

The initial conditions of the observer were set to zero.

In the next example, we illustrate the use of the sliding mode UIO as a part of the combined controller-observer compensator and an unknown input estimator as shown in Figure 12.6.

Example 12.4 We consider a plant with unknown input modeled by (12.3), where

$$\boldsymbol{A} = \begin{bmatrix} 0.6733 & -0.9068 & -0.3089 & 1.6703 \\ 1.2736 & -0.4747 & -0.8369 & -1.5417 \\ -1.0398 & -0.6661 & -0.6283 & -0.2720 \\ 0.4880 & -0.5020 & 0.5009 & 0.3416 \end{bmatrix}, \quad \boldsymbol{B}_1 = \begin{bmatrix} 0.4844 & 0.6678 \\ 0.9527 & 0.8982 \\ 0.1314 & -0.5493 \\ -1.7419 & -0.3175 \end{bmatrix},$$

$$\boldsymbol{B}_2 = \begin{bmatrix} -0.5827 \\ -0.1642 \\ -0.9351 \\ 1.1881 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} -1.2128 & 1.3227 & -0.3618 & 0.3871 \\ -0.3812 & 0.3853 & -0.4264 & 0.9028 \end{bmatrix}.$$

Let the unknown input be

$$u_2 = 3\cos(5t).$$

We use 12.2 to design the a sliding-mode UIO to reconstruct the unknown input u_2. Let the eigenvalues of $(\boldsymbol{A}_{22} - \boldsymbol{L}_{22}\boldsymbol{C}_{22})$ be placed at $\{-3, -4, -5\}$. Using MATLAB's function `place` we obtain

$$\boldsymbol{L}_{22} = \begin{bmatrix} -37.5312 \\ 31.9351 \\ -13.5910 \end{bmatrix}.$$

We select $\kappa = 3.7417$.

The desired closed-loop poles of the system are to be located at $\{-4, -5, -6, -7\}$. Using MATLAB's function `place` we obtain

$$\boldsymbol{K} = \begin{bmatrix} -3.6021 & 8.0249 & 8.1459 & -2.3902 \\ -9.3810 & -9.7238 & -38.9690 & -13.7649 \end{bmatrix}$$

such that $\text{eig}(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{B}_1) = \{-4, -5, -6, -7\}$.

In our simulations, the vector of the initial conditions of the plant is

$$\boldsymbol{x}(0) = \begin{bmatrix} 0.6470 & -3.4975 & -3.4439 & 0.3146 \end{bmatrix}^\top$$

and we set zero initial conditions for the observer. In Figure 12.7, we show plots of the plant's states and their estimates. In Figure 12.8, we show an estimate of the unknown input u_2, which is the output of the nonlinear element E.

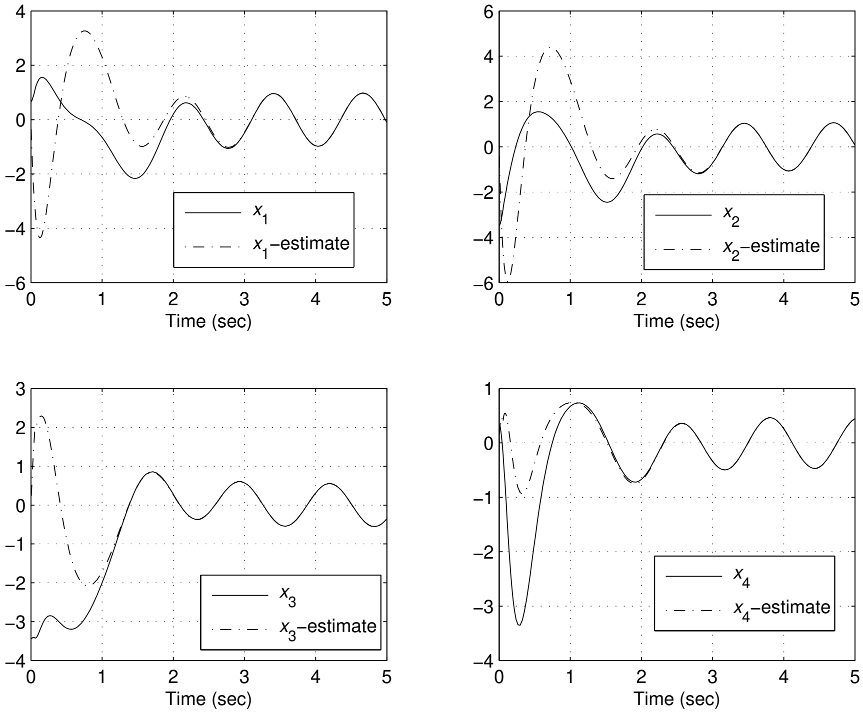


Figure 12.7: Plots of the states and their estimates.

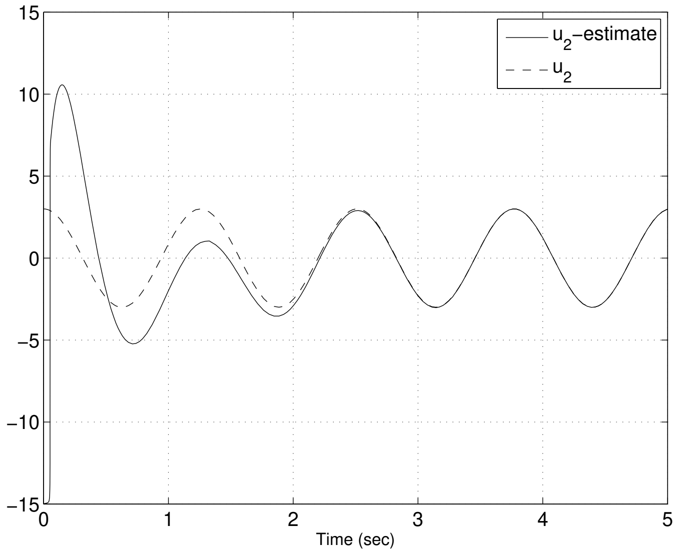


Figure 12.8: Plots of the unknown input and its estimate.

12.4.4 Sliding Mode UIO Design Using LMIs

We can use LMIs to design a sliding mode UIO. Xiang, Su, and Chu [356] represent the problem of finding a triple $(\boldsymbol{L}, \boldsymbol{F}, \boldsymbol{P})$ subject to (12.26) and (12.27) in the form of an LMI feasibility problem.

To proceed, let $\boldsymbol{G}$ be the orthogonal complement of $\boldsymbol{B}_2$, which means that $\boldsymbol{B}_2\boldsymbol{G} = \boldsymbol{O}$, and we write

$$\boldsymbol{G} = \boldsymbol{B}_2^{\perp},$$

where we assume that the matrix rank $\boldsymbol{B}_2 = m_2$, that is, the matrix $\boldsymbol{B}_2$ has full column rank. Then the problem of finding a triple $(\boldsymbol{L}, \boldsymbol{F}, \boldsymbol{P})$, $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ such that

$$(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})^\top \boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}) < 0 \quad \text{and} \quad \boldsymbol{B}_2^\top \boldsymbol{P} = \boldsymbol{F}\boldsymbol{C} \tag{12.47}$$

is equivalent to finding matrices

$$\begin{aligned} \boldsymbol{W}_1 &= \boldsymbol{W}_1^\top \in \mathbb{R}^{(n-m_2)\times(n-m_2)}, \\ \boldsymbol{W}_2 &= \boldsymbol{W}_2^\top \in \mathbb{R}^{p\times p}, \\ \boldsymbol{Y} &\in \mathbb{R}^{n\times p}, \end{aligned}$$

such that

$$\boldsymbol{G}\boldsymbol{W}_1\boldsymbol{G}^\top + \boldsymbol{C}^\top\boldsymbol{W}_2\boldsymbol{C} > 0 \tag{12.48}$$

and

$$\boldsymbol{G}\boldsymbol{W}_1\boldsymbol{G}^\top\boldsymbol{A} + \boldsymbol{C}^\top\boldsymbol{W}_2\boldsymbol{C}\boldsymbol{A} - \boldsymbol{Y}\boldsymbol{C} + \left(\boldsymbol{G}\boldsymbol{W}_1\boldsymbol{G}^\top\boldsymbol{A} + \boldsymbol{C}^\top\boldsymbol{W}_2\boldsymbol{C}\boldsymbol{A} - \boldsymbol{Y}\boldsymbol{C}\right)^\top < 0. \tag{12.49}$$

If a feasible solution to (12.48) and (12.49) exists, then

$$\boldsymbol{P} = \boldsymbol{G}\boldsymbol{W}_1\boldsymbol{G}^\top + \boldsymbol{C}^\top\boldsymbol{W}_2\boldsymbol{C}, \tag{12.50}$$

$$\boldsymbol{L} = \boldsymbol{P}^{-1}\boldsymbol{Y}, \tag{12.51}$$

$$\boldsymbol{F} = \boldsymbol{G}^\top\boldsymbol{C}^\top\boldsymbol{W}_2. \tag{12.52}$$

Example 12.5 We use LMIs to find a triple $(\boldsymbol{L}, \boldsymbol{F}, \boldsymbol{P})$ satisfying (12.47) for the same system as in Example 12.4. Recall that

$$\boldsymbol{A} = \begin{bmatrix} 0.6733 & -0.9068 & -0.3089 & 1.6703 \\ 1.2736 & -0.4747 & -0.8369 & -1.5417 \\ -1.0398 & -0.6661 & -0.6283 & -0.2720 \\ 0.4880 & -0.5020 & 0.5009 & 0.3416 \end{bmatrix}, \quad \boldsymbol{B}_2 = \begin{bmatrix} -0.5827 \\ -0.1642 \\ -0.9351 \\ 1.1881 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} -1.2128 & 1.3227 & -0.3618 & 0.3871 \\ -0.3812 & 0.3853 & -0.4264 & 0.9028 \end{bmatrix}.$$

We mention that the above matrices were generated using MATLAB's function `randn`.

We use MATLAB's LMI toolbox to solve the LMIs given by (12.49). Then, using (12.50), (12.51), and (12.52), we compute the matrices

$$\boldsymbol{P} = \begin{bmatrix} 178.2434 & -54.7774 & 11.9292 & 66.0388 \\ -54.7774 & 67.3161 & 16.2845 & 16.3716 \\ 11.9292 & 16.2845 & 61.8596 & 6.8944 \\ 66.0388 & 16.3716 & 6.8944 & 154.8361 \end{bmatrix}, \quad \boldsymbol{L} = \begin{bmatrix} -80.7624 & -30.7668 \\ 79.1421 & -578.7866 \\ -194.5307 & 302.2489 \\ 174.7920 & 2.1652 \end{bmatrix},$$

$$\boldsymbol{F} = \begin{bmatrix} -28.5969 & 163.2860 \end{bmatrix}.$$

12.5 The Edwards and Spurgeon Sliding Mode UIO

In this Section, we present an alternative approach to the sliding mode UIO design proposed by Edwards and Spurgeon [92]. The construction of this observer involves a number of state variable transformations that bring the plant model into a special canonical form that is then used to construct the observer. The inverse transformations are performed to obtain the final form of the observer. The necessary and sufficient conditions for the existence of the Edwards and Spurgeon observer are the same as for the previously discussed UIO architectures.

12.5.1 The Canonical Form—The Design Platform

We begin by assuming that

$$\operatorname{rank} \boldsymbol{B}_2 = m_2 \quad \text{and} \quad \operatorname{rank} \boldsymbol{C} = p.$$

We then construct

$$\boldsymbol{T}_c = \begin{bmatrix} \boldsymbol{N}_c^\top \\ \boldsymbol{C} \end{bmatrix},$$

where $\boldsymbol{N}_c \in \mathbb{R}^{n\times(n-p)}$ is the matrix whose columns span the null space of $\boldsymbol{C}$. The matrix $\boldsymbol{T}_c$ is invertible.

We use $\boldsymbol{T}_c$ to transform the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$ into the form,

$$\boldsymbol{A}_c = \boldsymbol{T}_c\boldsymbol{A}\boldsymbol{T}_c^{-1}, \quad \boldsymbol{B}_{2_c} = \boldsymbol{T}_c\boldsymbol{B}_2 = \begin{bmatrix} \boldsymbol{B}_{2_{c_1}} \\ \boldsymbol{B}_{2_{c_2}} \end{bmatrix},$$

$$\boldsymbol{C}_c = \boldsymbol{C}\boldsymbol{T}_c^{-1} = \begin{bmatrix} \boldsymbol{O}_{p\times(n-p)} & \boldsymbol{I}_p \end{bmatrix},$$

where $\boldsymbol{B}_{2_{c_1}} \in \mathbb{R}^{(n-p)\times m_2}$ and $\boldsymbol{B}_{2_{c_2}} \in \mathbb{R}^{p\times m_2}$.

The next state coordinate transformation, $\boldsymbol{T}_b$, contains two submatrices that require their careful description. First note that by assumption

$$\operatorname{rank} \boldsymbol{B}_2 = \operatorname{rank}(\boldsymbol{C}\boldsymbol{B}_2) = m_2$$

and

$$\boldsymbol{C}\boldsymbol{B}_2 = \boldsymbol{C}_c\boldsymbol{B}_{2_c} = \boldsymbol{B}_{2_{c_2}}.$$

Hence

$$\operatorname{rank} \boldsymbol{B}_{2_{c_2}} = m_2,$$

where recall $\boldsymbol{B}_{2_{c_2}} \in \mathbb{R}^{p\times m_2}$ and $p \geq m_2$. Therefore

$$\boldsymbol{B}_{2_{c_2}}^{\dagger} = \left(\boldsymbol{B}_{2_{c_2}}^{\top}\boldsymbol{B}_{2_{c_2}}\right)^{-1}\boldsymbol{B}_{2_{c_2}}^{\top}$$

is well defined. Next, there exists an orthogonal matrix $\boldsymbol{T}_{b_2} \in \mathbb{R}^{p\times p}$ such that

$$\boldsymbol{T}_{b_2}\boldsymbol{B}_{2_{c_2}} = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{B}_{2_{b_2}} \end{bmatrix},$$

where $\boldsymbol{B}_{2b} \in \mathbb{R}^{m_2\times m_2}$ and $\operatorname{rank}\boldsymbol{B}_{2b} = m_2$. We use $\boldsymbol{B}_{2_{c_2}}^{\dagger}$ and $\boldsymbol{T}_{b_2}$ to construct the coordinate transformation

$$\boldsymbol{T}_b = \begin{bmatrix} \boldsymbol{I}_{n-p} & -\boldsymbol{B}_{2_{c_1}}\boldsymbol{B}_{2_{c_2}}^{\dagger} \\ \boldsymbol{O}_{p\times(n-p)} & \boldsymbol{T}_{b_2} \end{bmatrix}.$$

Applying the transformation $\boldsymbol{T}_b$ to the triple $(\boldsymbol{A}_c, \boldsymbol{B}_{2c}, \boldsymbol{C}_c)$ gives

$$\boldsymbol{B}_{2_b} = \boldsymbol{T}_b\boldsymbol{B}_{2c}, \quad \boldsymbol{C}_b = \boldsymbol{C}_c\boldsymbol{T}_b^{-1}$$

and

$$\boldsymbol{A}_b = \boldsymbol{T}_b\boldsymbol{A}_c\boldsymbol{T}_b^{-1} = \begin{bmatrix} \boldsymbol{A}_{b11} & \boldsymbol{A}_{b12} \\ \boldsymbol{A}_{b21} & \boldsymbol{A}_{b22} \end{bmatrix},$$

where $\boldsymbol{A}_{b11} \in \mathbb{R}^{(n-m_2)\times(n-m_2)}$ and it has the form

$$\boldsymbol{A}_{b11} = \begin{bmatrix} \boldsymbol{A}_{1111} & \boldsymbol{A}_{1112} \\ \boldsymbol{A}_{1121} & \boldsymbol{A}_{1122} \end{bmatrix},$$

where $\boldsymbol{A}_{1111} \in \mathbb{R}^{(n-p)\times(n-p)}$ and $\boldsymbol{A}_{1121} \in \mathbb{R}^{(p-m_2)\times(n-p)}$. If the pair $(\boldsymbol{A}_{1111}, \boldsymbol{A}_{1121})$ is not observable, we construct a transformation $\boldsymbol{T}_{obs} \in \mathbb{R}^{(n-p)\times(n-p)}$ such that

$$\boldsymbol{T}_{obs}\boldsymbol{A}_{1111}\boldsymbol{T}_{obs}^{-1} = \begin{bmatrix} \boldsymbol{A}_{11}^{o} & \boldsymbol{A}_{12}^{o} \\ \boldsymbol{O} & \boldsymbol{A}_{22}^{o} \end{bmatrix} \quad \text{and} \quad \boldsymbol{A}_{1121}\boldsymbol{T}_{obs}^{-1} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{A}_{21}^{o} \end{bmatrix},$$

where $\boldsymbol{A}_{11}^o \in \mathbb{R}^{r\times r}$, $\boldsymbol{A}_{21}^o \in \mathbb{R}^{(p-m_2)\times(n-p-r)}$, $\boldsymbol{A}_{22}^o \in \mathbb{R}^{(n-p-r)\times(n-p-r)}$, the pair $(\boldsymbol{A}_{22}^o, \boldsymbol{A}_{21}^o)$ is observable, and r is number of unobservable states of the pair $(\boldsymbol{A}_{1111}, \boldsymbol{A}_{1121})$. We embed the transformation $\boldsymbol{T}_{obs}$ into

$$\boldsymbol{T}_a = \begin{bmatrix} \boldsymbol{T}_{obs} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_p \end{bmatrix}.$$

Applying $\boldsymbol{T}_a$ to the triple $(\boldsymbol{A}_b, \boldsymbol{B}_{2_b}, \boldsymbol{C}_b)$ yields

$$\boldsymbol{A}_a = \boldsymbol{T}_a \boldsymbol{A}_b \boldsymbol{T}_a^{-1} = \begin{bmatrix} \boldsymbol{A}_{a11} & \boldsymbol{A}_{a12} \\ \boldsymbol{A}_{a21} & \boldsymbol{A}_{a22} \end{bmatrix},$$

where

$$\boldsymbol{A}_{a11} = \left[\begin{array}{cc|c} \boldsymbol{A}_{11}^o & \boldsymbol{A}_{12}^o & \\ & & \boldsymbol{A}_{12}^m \\ \boldsymbol{O} & \boldsymbol{A}_{22}^o & \\ \hline \boldsymbol{O} & \boldsymbol{A}_{21}^o & \boldsymbol{A}_{22}^m \end{array}\right] \in \mathbb{R}^{(n-m_2)\times(n-m_2)},$$

$$\boldsymbol{B}_{2_a} = \boldsymbol{T}_a \boldsymbol{B}_{2_b} = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{B}_{2_{b_2}} \end{bmatrix}, \quad \boldsymbol{B}_{2_{b_2}} \in \mathbb{R}^{m_2\times m_2},$$

and

$$\boldsymbol{C}_a = \boldsymbol{C}_b \boldsymbol{T}_a^{-1} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{T}_{b_2}^\top \end{bmatrix}, \quad \boldsymbol{T}_{b_2} \in \mathbb{R}^{p\times p}.$$

Remark 12.1 *The eigenvalues of $\boldsymbol{A}_{11}^o$ are the system zeros of the system represented by the triple $(\boldsymbol{A}_a, \boldsymbol{B}_{2_a}, \boldsymbol{C}_a)$. Because system zeros are invariant under similarity transformations, the eigenvalues of $\boldsymbol{A}_{11}^o$ are the system zeros of the system represented by the triple $(\boldsymbol{A}, \boldsymbol{B}_2, \boldsymbol{C})$.*

◇

By assumption, the system zeros are in the open left-half complex plane. The pair $(\boldsymbol{A}_{22}^o, \boldsymbol{A}_{21}^o)$ is observable; therefore we can construct a gain matrix $\boldsymbol{L}_a \in \mathbb{R}^{(n-p-r)\times(p-m_2)}$ such that the matrix

$$\boldsymbol{A}_{22}^o - \boldsymbol{L}_a \boldsymbol{A}_{21}^o$$

has its eigenvalues in pre-specified locations in the left-half plane. Let

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{O}_{r\times(p-m_2)} \\ \boldsymbol{L}_a \end{bmatrix}$$

Then because the eigenvalues of the matrix $\boldsymbol{A}_{11}^o$ are asymptotically stable, the matrix

$$\mathcal{A}_{11} = \begin{bmatrix} \boldsymbol{A}_{11}^o & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{22}^o \end{bmatrix} - \begin{bmatrix} \boldsymbol{O}_{r\times(p-m_2)} \\ \boldsymbol{L}_a \end{bmatrix} \begin{bmatrix} \boldsymbol{O}_{(p-m_2)\times r} & \boldsymbol{A}_{21}^o \end{bmatrix}$$

has its all eigenvalues in the left-half plane, where $\mathcal{A}_{11} \in \mathbb{R}^{(n-p)\times(n-p)}$.

Let

$$\bar{\boldsymbol{L}} = \left[\begin{array}{cc} \boldsymbol{L} & \boldsymbol{O}_{(n-p)\times m_2} \end{array} \right].$$

We construct the transformation

$$\boldsymbol{T}_L = \left[\begin{array}{cc} \boldsymbol{I}_{n-p} & -\bar{\boldsymbol{L}} \\ \boldsymbol{O}_{p\times(n-p)} & \boldsymbol{T}_{b2} \end{array} \right]$$

and apply it to the triple $(\boldsymbol{A}_a, \boldsymbol{B}_{2_a}, \boldsymbol{C}_a)$ to obtain

$$\mathcal{B}_2 = \boldsymbol{T}_L \boldsymbol{B}_{2_a} = \left[\begin{array}{c} \boldsymbol{O} \\ \mathcal{B}_{2_2} \end{array} \right], \quad \mathcal{C} = \boldsymbol{C}_a \boldsymbol{T}_L^{-1} = \left[\begin{array}{cc} \boldsymbol{O} & \boldsymbol{I}_p \end{array} \right],$$

and

$$\mathcal{A} = \boldsymbol{T}_L \boldsymbol{A}_a \boldsymbol{T}_L^{-1} = \left[\begin{array}{cc} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & \mathcal{A}_{22} \end{array} \right],$$

where $\mathcal{A}_{11} \in \mathbb{R}^{(n-p)\times(n-p)}$, $\mathcal{A}_{12} \in \mathbb{R}^{(n-p)\times p}$, $\mathcal{A}_{21} \in \mathbb{R}^{p\times(n-p)}$ and $\mathcal{A}_{22} \in \mathbb{R}^{p\times p}$. The matrix $\mathcal{A}_{11}$ has asymptotically stable eigenvalues.

The plant model represented by the quadruple $(\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{C})$ takes the following form in the new coordinates

$$\left. \begin{array}{rcl} \dot{\boldsymbol{x}}_1 & = & \mathcal{A}_{11}\boldsymbol{x}_1 + \mathcal{A}_{12}\boldsymbol{y} + \mathcal{B}_{1_1}\boldsymbol{u}_1 \\ \dot{\boldsymbol{y}} & = & \mathcal{A}_{21}\boldsymbol{x}_1 + \mathcal{A}_{22}\boldsymbol{y} + \mathcal{B}_{1_2}\boldsymbol{u}_1 + \mathcal{B}_{2_2}\boldsymbol{u}_2 \end{array} \right\} \tag{12.53}$$

We refer to the above representation as the *Edwards-Spurgeon canonical form.*

12.5.2 Observer Architecture

We are now ready to present the Edwards-Spurgeon UIO. Let $\hat{\boldsymbol{x}}_1$ denote an estimate of $\boldsymbol{x}_1$, and let $\hat{\boldsymbol{y}}$ denote an estimate of $\boldsymbol{y}$. We also let

$$\boldsymbol{e}_y = \boldsymbol{y} - \hat{\boldsymbol{y}}$$

to denote the estimation error of $\boldsymbol{y}$. Edwards and Spurgeon [92, p. 134] propose their UIO for the plant model in the canonical form (12.53),

$$\left. \begin{array}{rcl} \dot{\hat{\boldsymbol{x}}}_1 & = & \mathcal{A}_{11}\hat{\boldsymbol{x}}_1 + \mathcal{A}_{12}\hat{\boldsymbol{y}} + \mathcal{B}_{1_1}\boldsymbol{u}_1 + \mathcal{A}_{12}\boldsymbol{e}_y \\ \dot{\hat{\boldsymbol{y}}} & = & \mathcal{A}_{21}\hat{\boldsymbol{x}}_1 + \mathcal{A}_{22}\hat{\boldsymbol{y}} + \mathcal{B}_{1_2}\boldsymbol{u}_1 + \left(\mathcal{A}_{22} - \mathcal{A}_{22}^s\right)\boldsymbol{e}_y + \boldsymbol{\nu}, \end{array} \right\} \tag{12.54}$$

where $\mathcal{A}_{22}^s$ is a stable design matrix. The discontinuous injection term $\boldsymbol{\nu}$ is

$$\boldsymbol{\nu} = \begin{cases} \rho\|\mathcal{B}_{2_2}\|\dfrac{\boldsymbol{P}_2\boldsymbol{e}_y}{\|\boldsymbol{P}_2\boldsymbol{e}_y\|} & \text{if } \boldsymbol{e}_y \neq \boldsymbol{0} \\ \boldsymbol{0} & \text{otherwise} \end{cases}$$

where $\boldsymbol{P}_2 = \boldsymbol{P}_2^\top > 0$ is the solution to the matrix Lyapunov equation

$$\mathcal{A}_{22}^{s\top}\boldsymbol{P}_2 + \boldsymbol{P}_2\mathcal{A}_{22}^s = -\boldsymbol{Q}_2 \tag{12.55}$$

for some symmetric positive definite $\boldsymbol{Q}_2$, and the gain $\rho \geq \|\boldsymbol{u}_2\|$.

Let $\boldsymbol{e}_1 = \boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1$. The dynamics of the observation error are

$$\begin{aligned} \dot{\boldsymbol{e}}_1 &= \mathcal{A}_{11}\hat{\boldsymbol{e}}_1 \\ \dot{\boldsymbol{e}}_y &= \mathcal{A}_{21}\hat{\boldsymbol{e}}_1 + \mathcal{A}_{22}^s\hat{\boldsymbol{e}}_y + \mathcal{B}_{2_2}\boldsymbol{u}_2 - \boldsymbol{\nu}. \end{aligned}$$

One can show using the Lyapunov second method that the above error system is asymptotically stable.

Let the linear term coefficient $\boldsymbol{G}_l$ be defined as

$$\boldsymbol{G}_l = \boldsymbol{T}_c^{-1}\boldsymbol{T}_b^{-1}\boldsymbol{T}_a^{-1}\boldsymbol{T}_L^{-1}\begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} - \mathcal{A}_{22}^s \end{bmatrix},$$

and the nonlinear term coefficient $\mathcal{G}_n$ be defined as

$$\boldsymbol{G}_n = \boldsymbol{T}_c^{-1}\boldsymbol{T}_b^{-1}\boldsymbol{T}_a^{-1}\boldsymbol{T}_L^{-1}\begin{bmatrix} \boldsymbol{O}_{(n-p)\times p} \\ \boldsymbol{I}_p \end{bmatrix}.$$

Recall that

$$\boldsymbol{e}_y = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{C}\boldsymbol{x} - \boldsymbol{C}\hat{\boldsymbol{x}} = \boldsymbol{C}\boldsymbol{e}.$$

Then the Edwards-Spurgeon UIO in the original coordinates takes the form

$$\boxed{\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}_1\boldsymbol{u}_1 + \boldsymbol{G}_l\boldsymbol{e}_y + \boldsymbol{G}_n\boldsymbol{\nu}} \tag{12.56}$$

12.5.3 Reconstructing the Unknown Input

We show here how the Edwards-Spurgeon observer (12.56) can be used to reconstruct the unknown input $\boldsymbol{u}_2$. Consider the manifold,

$$\{\boldsymbol{e} : \boldsymbol{\sigma}(\boldsymbol{e}) = \boldsymbol{C}\boldsymbol{e} = \boldsymbol{0}\}. \tag{12.57}$$

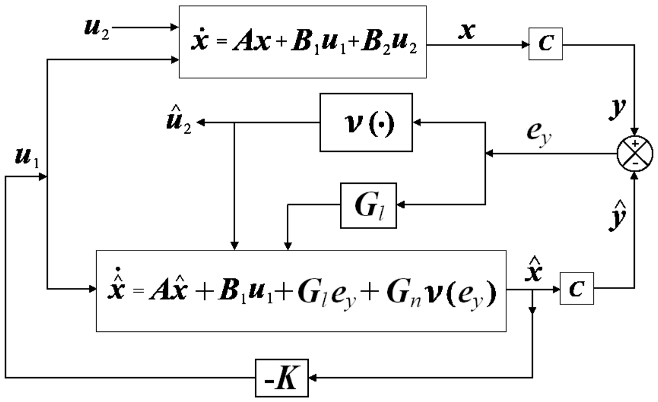


Figure 12.9: The Edwards-Spurgeon sliding mode UIO given by (12.56) as a part of the combined observer-controller compensator and an unknown input estimator.

We can always choose ρ such that the above manifold is attractive and it will be reached in a finite time. Thus, after the estimation error trajectory, $\boldsymbol{e}$, enters a sliding mode along the sliding surface, we have $\boldsymbol{\sigma}(\boldsymbol{e}) = \mathbf{0}$, and

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}\left(\boldsymbol{A} - \boldsymbol{G}_l\boldsymbol{C}\right)\boldsymbol{e} + \boldsymbol{C}\boldsymbol{B}_2\boldsymbol{u}_2 - \boldsymbol{C}\boldsymbol{G}_n\boldsymbol{\nu}(\boldsymbol{C}\boldsymbol{e}) = \mathbf{0}. \tag{12.58}$$

Because $\boldsymbol{e} \to \mathbf{0}$ as $t \to \infty$, we obtain

$$\boldsymbol{C}\boldsymbol{B}_2\boldsymbol{u}_2 = \boldsymbol{C}\boldsymbol{G}_n\boldsymbol{\nu}.$$

From the above, we obtain a least-squares estimate of the unknown input of the form

$$\boxed{\hat{\boldsymbol{u}}_2 = (\boldsymbol{C}\boldsymbol{B}_2)^{\dagger}\boldsymbol{C}\boldsymbol{G}_n\boldsymbol{\nu}} \tag{12.59}$$

As in the previous case, the Edwards-Spurgeon sliding mode UIO can also be used in the combined controller-observer compensator architecture and as unknown input estimator as depicted in Figure 12.9.

Example 12.6 We construct the Edwards-Spurgeon sliding UIO given by (12.56) for the plant in Example 12.4 and use it in the architecture depicted in Figure 12.9.

We have $m_2 = 1$. We start with computing $\boldsymbol{T}_c$, where we employ the command, `Nc=null(C)`, to find $\boldsymbol{N}_c$. We use

$$\rho\|\mathcal{B}_{2_2}\| = 10\| \begin{bmatrix} 1.2877 & 1.6302 \end{bmatrix}^\top \|.$$

The gain matrix $\boldsymbol{L}$ was selected using the `place` command so that the eigenvalues of $\mathcal{A}_{11}$ are placed at $\{-5, -6\}$. The design parameter matrix, $\boldsymbol{A}_{22}^s$, was selected as

$$\boldsymbol{A}_{22}^s = \begin{bmatrix} -6 & 0 \\ 0 & -7 \end{bmatrix}.$$

For $\boldsymbol{Q}_2 = 2\boldsymbol{I}_2$, the solution to (12.55) is

$$\boldsymbol{P}_2 = \begin{bmatrix} 0.1667 & 0 \\ 0 & 0.1429 \end{bmatrix}.$$

The nonlinear discontinuous term $\boldsymbol{\nu}$ was approximated with

$$\tilde{\boldsymbol{\nu}} = \rho \|\mathcal{B}_{2_2}\| \frac{\boldsymbol{P}_2 \boldsymbol{e}_y}{\|\boldsymbol{P}_2 \boldsymbol{e}_y\| + \mu},$$

where $\mu = 0.01$.

The feedback matrix $\boldsymbol{K}$ was computed using the `place` function so that the eigenvalues of $(\boldsymbol{A} - \boldsymbol{B}_1\boldsymbol{K})$ are located at $\{-4, -5, -6, -7\}$.

In Figure 12.10, we show plots of the plant's states and their estimates. In Figure 12.11, we show plots of the unknown input, $u_2 = 3\cos(5t)$, and its estimate.

12.6 Notes

In his practical guide for the selection and installation of observers in control systems, Ellis [94, page 3] writes: "Observers add complexity to the system and require computational resources. They may be less robust than physical sensors, especially when plant parameters change substantially during operation. Still, an observer applied with skill can bring substantial performance benefits and do so, in many cases, while reducing cost or increasing reliability." Examples of impressive applications of nonlinear observers to the control of electric machinery can be found in [72, 330, 308]. Röbenack and Lynch [282] presented a method to the observer design for a class of nonlinear plants that yields almost linear observation error dynamics. An application of the sliding mode observer presented in Subsection 12.4 to decentralized dynamic output feedback control of nonlinear interconnected systems is reported in [172].

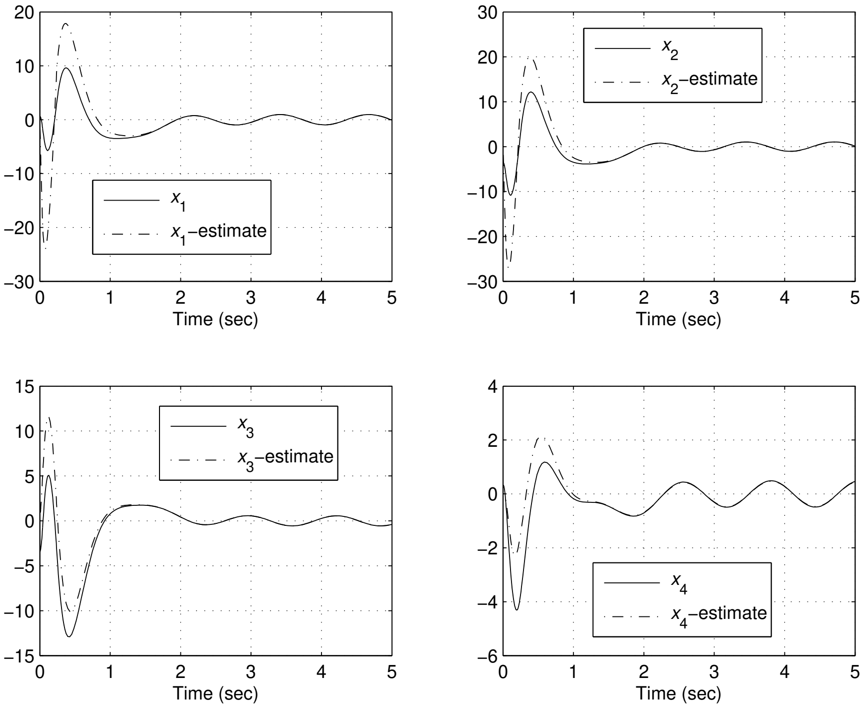


Figure 12.10: Plots of states and their estimates using the Edwards-Spurgeon UIO.

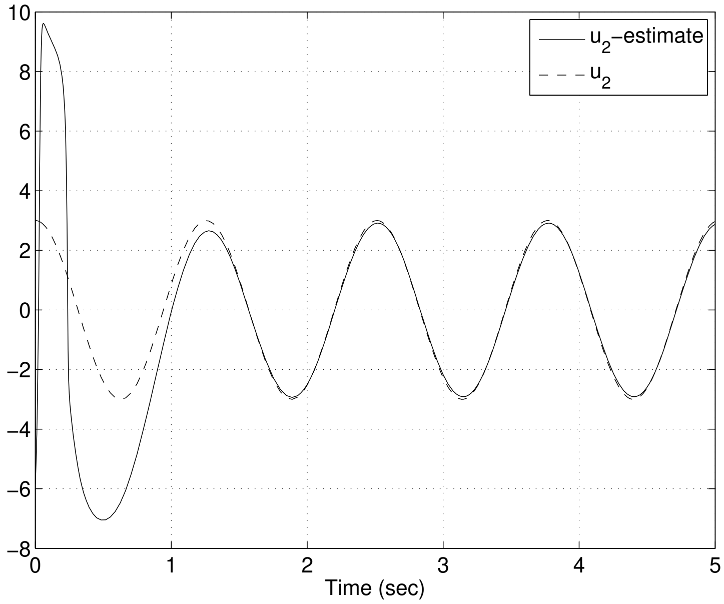


Figure 12.11: Plots of the unknown input and its estimate using the Edwards-Spurgeon UIO.

Sliding-mode observers for dynamical systems with unknown inputs have found applications in robust detection and reconstruction of actuator and sensor faults [319, 357, 318, 359, 358, 8, 9, 320] and fault detection and isolation (FDI) [291]. In particular, the purpose of the FDI scheme is to generate an alarm when a fault, such as a component malfunction, develops in the process being monitored and to identify the location of the fault. The fault signal can often be modeled as an unknown input to the plant. Thus, unknown input observers can be used for FDI purposes. For example, Edwards and Spurgeon (see [92, Section 6.5], Edwards, Spurgeon, and Patton [93]), and Saif and Xiong [291] use sliding mode observers to develop FDI schemes for reconstructing the actuator and sensor fault signals.

12.7 Exercises

Exercise 12.1 Design and test the full-order UIO described in Theorem 12.1 for the system model of Example 12.2.

Exercise 12.2 Design and test the UIO of Hou and Müller, described in Section 12.3.4, for the system model of Example 12.2.

Exercise 12.3 Consider two system models that appear in [360] and [71]. The first system, using our notation, is modeled as

$$\begin{aligned}\boldsymbol{A} &= \begin{bmatrix} -2 & -2 & 0 \\ 0 & 0 & 1 \\ 0 & -3 & -4 \end{bmatrix}, \quad \boldsymbol{B}_1 = \boldsymbol{O}, \quad \boldsymbol{B}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\ \boldsymbol{C} &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.\end{aligned}$$

The second system is modeled by

$$\begin{aligned}\boldsymbol{A} &= \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & -1 \end{bmatrix}, \quad \boldsymbol{B}_1 = \boldsymbol{O}, \quad \boldsymbol{B}_2 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \\ \boldsymbol{C} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.\end{aligned}$$

Construct UIOs developed in Section 12.3.2, 12.3.3, and 12.4.3 for the above systems.

Exercise 12.4 Construct the Hou and Müller UIO, described in Section 12.3.4, for each system of Exercise 12.3.

Exercise 12.5 Prove Lemma 12.4.

Exercise 12.6 Show that the conditions in (12.47) are equivalent to the LMIs given by (12.48) and (12.49).

Exercise 12.7 Use an LMI toolbox to find matrices $\boldsymbol{L}$, $\boldsymbol{F}$, and $\boldsymbol{P}$ of Theorem 12.3 for the system of Example 12.3.

Chapter 13

Model-based Predictive Control (MPC)

> A system—organism, corporate firm, computer program—spends energy feeding the past back into the present because this is an economical way for the system to deal with the future. To see into the future one must see into the past. A constant pulse of the past along feedback loops informs and controls the future. [175, p. 439]

The model-based predictive control (MPC) methodology is also referred to as the moving horizon control or the receding horizon control. The idea behind this approach can be explained using an example of driving a car. The driver looks at the road ahead of him and taking into account the present state and the previous action predicts his action up to some distance ahead, which we refer to as the prediction horizon. Based on the prediction, the driver adjusts the driving direction. The MPC main idea is illustrated in Figure 13.1.

The MPC is constructed using control and optimization tools. The objective of this chapter is to introduce the reader to the linear MPC which refers to the family of MPC schemes in which linear models of the controlled objects are used in the control law synthesis.

In the MPC approach, the current control action is computed on-line rather than using a pre-computed, off-line, control law.

A model predictive controller uses, at each sampling instant, the plant's current input and output measurements, the plant's current state, and the plant's model to

- calculate, over a finite horizon, a future control sequence that optimizes a given performance index and satisfies constraints on the control action;

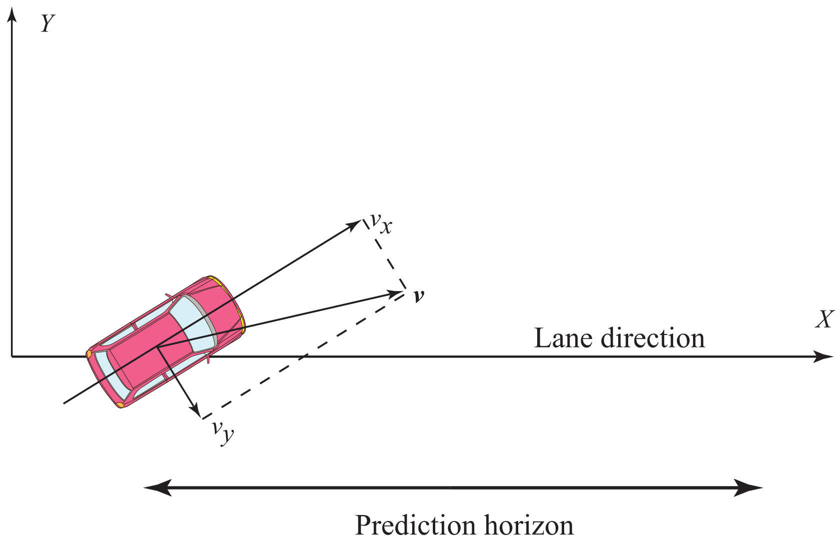


Figure 13.1: The driver predicts future travel direction based on the current state of the car and the current position of the steering wheel.

- use the first control in the sequence as the plant's input.

The MPC strategy is illustrated in Figure 13.2, where N_p is the prediction horizon, $u(t+k|t)$ is the predicted control action at $t+k$ given $u(t)$. Similarly, $y(t+k|t)$ is the predicted output at $t+k$ given $y(t)$.

In Figure 13.3, we show basic structure of an MPC-controlled plant, where we assume that the plant's state is available to us.

13.1 Simple Discrete-Time MPC

The development a discrete MPC in this section follows that of [342, Chapter 1]. We consider a discretized model of a dynamic system of the form,

$$\boldsymbol{x}[k+1] = \boldsymbol{\Phi}\boldsymbol{x}[k] + \boldsymbol{\Gamma}\boldsymbol{u}[k] \tag{13.1}$$

$$\boldsymbol{y}[k] = \boldsymbol{C}\boldsymbol{x}[k], \tag{13.2}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{n\times n}$, $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times m}$, and $\boldsymbol{C} \in \mathbb{R}^{p\times n}$.

Applying the backward difference operator, $\Delta\boldsymbol{x}[k+1] = \boldsymbol{x}[k+1] - \boldsymbol{x}[k]$, to (13.1) gives

$$\Delta\boldsymbol{x}[k+1] = \boldsymbol{\Phi}\Delta\boldsymbol{x}[k] + \boldsymbol{\Gamma}\Delta\boldsymbol{u}[k], \tag{13.3}$$

where $\Delta\boldsymbol{u}[k+1] = \boldsymbol{u}[k+1] - \boldsymbol{u}[k]$.

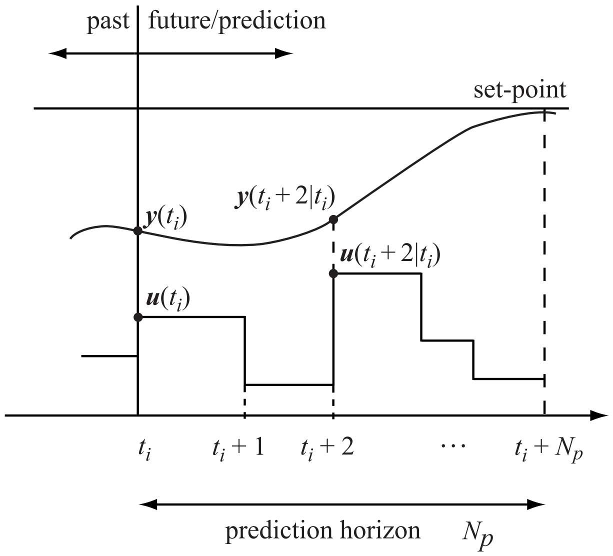


Figure 13.2: Controller action construction using model-based predictive control (MPC) approach.

We now apply the backward difference operator to (13.2) to obtain

$$\begin{aligned}\Delta\boldsymbol{y}[k+1] &= \boldsymbol{y}[k+1]-\boldsymbol{y}[k]\\ &= \boldsymbol{C}\boldsymbol{x}[k+1]-\boldsymbol{C}\boldsymbol{x}[k]\\ &= \boldsymbol{C}\Delta\boldsymbol{x}[k+1].\end{aligned}$$

Substituting into the above (13.3) yields

$$\Delta\boldsymbol{y}[k+1]=\boldsymbol{C}\boldsymbol{\Phi}\Delta\boldsymbol{x}[k]+\boldsymbol{C}\boldsymbol{\Gamma}\Delta\boldsymbol{u}[k].$$

Hence,

$$\boldsymbol{y}[k+1]=\boldsymbol{y}[k]+\boldsymbol{C}\boldsymbol{\Phi}\Delta\boldsymbol{x}[k]+\boldsymbol{C}\boldsymbol{\Gamma}\Delta\boldsymbol{u}[k]. \tag{13.4}$$

We combine (13.3) and (13.4) into one equation to obtain

$$\begin{bmatrix}\Delta\boldsymbol{x}[k+1]\\ \boldsymbol{y}[k+1]\end{bmatrix}=\begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{O}\\ \boldsymbol{C}\boldsymbol{\Phi} & \boldsymbol{I}_p\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{x}[k]\\ \boldsymbol{y}[k]\end{bmatrix}+\begin{bmatrix}\boldsymbol{\Gamma}\\ \boldsymbol{C}\boldsymbol{\Gamma}\end{bmatrix}\Delta\boldsymbol{u}[k]. \tag{13.5}$$

We represent (13.2) as

$$\boldsymbol{y}[k]=\begin{bmatrix}\boldsymbol{O} & \boldsymbol{I}_p\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{x}[k]\\ \boldsymbol{y}[k]\end{bmatrix}. \tag{13.6}$$

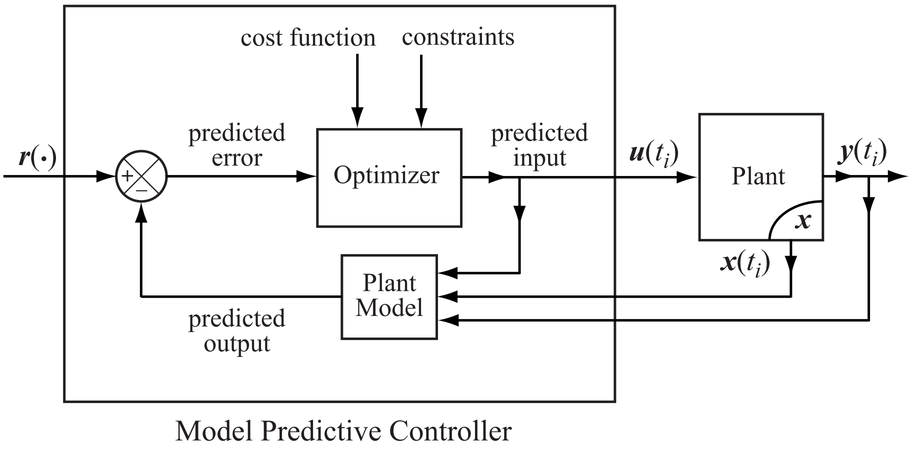


Figure 13.3: State feedback model predictive controller.

We define the augmented state vector,

$$\boldsymbol{x}_a[k] = \begin{bmatrix} \Delta\boldsymbol{x}[k] \\ \boldsymbol{y}[k] \end{bmatrix}. \tag{13.7}$$

Let

$$\boldsymbol{\Phi}_a = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{O} \\ \boldsymbol{C\Phi} & \boldsymbol{I}_p \end{bmatrix}, \quad \boldsymbol{\Gamma}_a = \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{C\Gamma} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{C}_a = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_p \end{bmatrix}. \tag{13.8}$$

Using the above notation, we represent (13.5) and (13.6) in a compact format as

$$\boldsymbol{x}_a[k+1] = \boldsymbol{\Phi}_a\boldsymbol{x}_a[k] + \boldsymbol{\Gamma}_a\Delta\boldsymbol{u}[k] \tag{13.9}$$

$$\boldsymbol{y}[k] = \boldsymbol{C}_a\boldsymbol{x}_a[k], \tag{13.10}$$

where $\boldsymbol{\Phi}_a \in \mathbb{R}^{(n+p)\times(n+p)}$, $\boldsymbol{\Gamma}_a \in \mathbb{R}^{(n+p)\times m}$, and $\boldsymbol{C}_a \in \mathbb{R}^{p\times(n+p)}$.

Suppose now that the state vector $\boldsymbol{x}_a$ at each sampling time, k, is available to us. Our control objective is to construct a control sequence,

$$\Delta\boldsymbol{u}[k], \Delta\boldsymbol{u}[k+1], \ldots, \Delta\boldsymbol{u}[k+N_p-1], \tag{13.11}$$

where N_p is the prediction horizon, such that a given cost function and constraints are satisfied. The above control sequence will result in a predicted sequence of the state vectors,

$$\boldsymbol{x}_a(k+1|k), \boldsymbol{x}_a(k+2|k), \ldots, \boldsymbol{x}_a(k+N_p|k),$$

which can then be used to compute predicted sequence of the plant's outputs,

$$\boldsymbol{y}(k+1|k), \boldsymbol{y}(k+2|k), \ldots, \boldsymbol{y}(k+N_p|k). \tag{13.12}$$

Using the above information, we can compute the control sequence (13.11) and then apply $\boldsymbol{u}[k]$ to the plant to generate $\boldsymbol{x}[k+1]$. We repeat the process again, using $\boldsymbol{x}[k+1]$ as an initial condition to compute $\boldsymbol{u}[k+1]$, until we reach the boundary of the control horizon, N_c.

We now present an approach to construct $\boldsymbol{u}[k]$ given $\boldsymbol{x}[k]$. Using the plant model parameters and the measurement of $\boldsymbol{x}_a[k]$ we evaluate the augmented states over the prediction horizon successively applying the recursion formula (13.9) to obtain,

$$\begin{aligned}
\boldsymbol{x}_a(k+1|k) &= \boldsymbol{\Phi}_a\boldsymbol{x}_a[k] + \boldsymbol{\Gamma}_a\Delta\boldsymbol{u}[k] \\
\boldsymbol{x}_a(k+2|k) &= \boldsymbol{\Phi}_a\boldsymbol{x}_a(k+1|k) + \boldsymbol{\Gamma}_a\Delta\boldsymbol{u}[k+1] \\
&\quad = \boldsymbol{\Phi}_a^2\boldsymbol{x}_a[k] + \boldsymbol{\Phi}_a\boldsymbol{\Gamma}_a\Delta\boldsymbol{u}[k] + \boldsymbol{\Gamma}_a\Delta\boldsymbol{u}[k+1] \\
&\vdots \\
\boldsymbol{x}_a(k+N_p|k) &= \boldsymbol{\Phi}_a^{N_p}\boldsymbol{x}_a[k] + \boldsymbol{\Phi}_a^{N_p-1}\boldsymbol{\Gamma}_a\Delta\boldsymbol{u}[k] + \cdots + \boldsymbol{\Gamma}_a\Delta\boldsymbol{u}[k+N_p-1]
\end{aligned}$$

We represent the above set of equations in the form,

$$\begin{bmatrix} \boldsymbol{x}_a(k+1|k) \\ \boldsymbol{x}_a(k+2|k) \\ \vdots \\ \boldsymbol{x}_a(k+N_p|k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_a \\ \boldsymbol{\Phi}_a^2 \\ \vdots \\ \boldsymbol{\Phi}_a^{N_p} \end{bmatrix} \boldsymbol{x}_a[k] + \begin{bmatrix} \boldsymbol{\Gamma}_a & & & \\ \boldsymbol{\Phi}_a\boldsymbol{\Gamma}_a & \boldsymbol{\Gamma}_a & & \\ \vdots & & \ddots & \\ \boldsymbol{\Phi}_a^{N_p-1}\boldsymbol{\Gamma}_a & & \cdots & \boldsymbol{\Gamma}_a \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}[k] \\ \Delta\boldsymbol{u}[k+1] \\ \vdots \\ \Delta\boldsymbol{u}[k+N_p-1] \end{bmatrix}. \tag{13.13}$$

We wish to design a controller that would force the plant output, $\boldsymbol{y}$, to track a given reference signal, $\boldsymbol{r}$. Using (13.10) and (13.13), we compute the sequence of predicted outputs (13.12),

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{y}(k+1|k) \\ \boldsymbol{y}(k+2|k) \\ \vdots \\ \boldsymbol{y}(k+N_p|k) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{C}_a\boldsymbol{x}_a(k+1|k) \\ \boldsymbol{C}_a\boldsymbol{x}_a(k+2|k) \\ \vdots \\ \boldsymbol{C}_a\boldsymbol{x}_a(k+N_p|k) \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{C}_a\boldsymbol{\Phi}_a \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^2 \\ \vdots \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^{N_p} \end{bmatrix} \boldsymbol{x}_a[k] \qquad (13.14) \\
&+ \begin{bmatrix} \boldsymbol{C}_a\boldsymbol{\Gamma}_a & & & \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a\boldsymbol{\Gamma}_a & \boldsymbol{C}_a\boldsymbol{\Gamma}_a & & \\ \vdots & & \ddots & \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^{N_p-1}\boldsymbol{\Gamma}_a & & \cdots & \boldsymbol{C}_a\boldsymbol{\Gamma}_a \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}[k] \\ \Delta\boldsymbol{u}[k+1] \\ \vdots \\ \Delta\boldsymbol{u}[k+N_p-1] \end{bmatrix}. \qquad (13.15)
\end{aligned}$$

We write the above compactly as

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{x}_a[k] + \boldsymbol{Z}\Delta\boldsymbol{U}, \tag{13.16}$$

where

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}(k+1|k) \\ \boldsymbol{y}(k+2|k) \\ \vdots \\ \boldsymbol{y}(k+N_p|k) \end{bmatrix}, \quad \Delta\boldsymbol{U} = \begin{bmatrix} \Delta\boldsymbol{u}[k] \\ \Delta\boldsymbol{u}[k+1] \\ \vdots \\ \Delta\boldsymbol{u}[k+N_p-1] \end{bmatrix},$$

and

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{C}_a\boldsymbol{\Phi}_a \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^2 \\ \vdots \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^{N_p} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{C}_a\boldsymbol{\Gamma}_a & & \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a\boldsymbol{\Gamma}_a & \boldsymbol{C}_a\boldsymbol{\Gamma}_a & \\ \vdots & & \ddots & \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^{N_p-1}\boldsymbol{\Gamma}_a & & \cdots & \boldsymbol{C}_a\boldsymbol{\Gamma}_a \end{bmatrix}.$$

Suppose now that we wish to construct a control sequence, $\Delta\boldsymbol{u}[k], \ldots, \Delta\boldsymbol{u}[k+N_p-1]$, that would minimize the cost function

$$J(\Delta\boldsymbol{U}) = \frac{1}{2}\left(\boldsymbol{r}_p - \boldsymbol{Y}\right)^\top \boldsymbol{Q}\left(\boldsymbol{r}_p - \boldsymbol{Y}\right) + \frac{1}{2}\Delta\boldsymbol{U}^\top\boldsymbol{R}\Delta\boldsymbol{U}, \tag{13.17}$$

where $\boldsymbol{Q} = \boldsymbol{Q}^\top \geq 0$ and $\boldsymbol{R} = \boldsymbol{R}^\top > 0$ are real symmetric positive semi-definite and positive-definite weight matrices, respectively. The multiplying scalar, $1/2$, is just to make subsequent manipulations cleaner. Finally, the vector $\boldsymbol{r}_p$ consists of the values of the command signal at sampling times, $k, k+1, \ldots, k+N_p-1$. The selection of the weight matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$ reflects our control objective to keep the tracking error $\|\boldsymbol{r}_p - \boldsymbol{Y}\|$ "small" using the control actions that are "not too large."

We first apply the first-order necessary condition (FONC) test to $J(\Delta\boldsymbol{U})$,

$$\frac{\partial J}{\partial \Delta\boldsymbol{U}} = \boldsymbol{0}^\top.$$

Then, we solve the above equation for $\Delta\boldsymbol{U} = \Delta\boldsymbol{U}^*$, where

$$\begin{aligned} \frac{\partial J}{\partial \Delta\boldsymbol{U}} &= -\left(\boldsymbol{r}_p - \boldsymbol{W}\boldsymbol{x}_a - \boldsymbol{Z}\Delta\boldsymbol{U}\right)^\top \boldsymbol{Q}\boldsymbol{Z} + \Delta\boldsymbol{U}^\top\boldsymbol{R} \\ &= \boldsymbol{0}^\top. \end{aligned}$$

Performing simple manipulations yields

$$-\boldsymbol{r}_p^\top\boldsymbol{Q}\boldsymbol{Z} + \boldsymbol{x}_a^\top\boldsymbol{W}^\top\boldsymbol{Q}\boldsymbol{Z} + \Delta\boldsymbol{U}^\top\boldsymbol{Z}^\top\boldsymbol{Q}\boldsymbol{Z} + \Delta\boldsymbol{U}^\top\boldsymbol{R} = \boldsymbol{0}^\top.$$

Applying the transposition operation to both sides of the above equation and rearranging terms, we obtain

$$\left(\boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\right)\Delta\boldsymbol{U}=\boldsymbol{Z}^{\top}\boldsymbol{Q}\left(\boldsymbol{r}_p-\boldsymbol{W}\boldsymbol{x}_a\right).$$

Note that the matrix $\left(\boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\right)$ is invertible, and in fact, positive definite because $\boldsymbol{R}=\boldsymbol{R}^{\top}>0$ and $\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}$ is also symmetric and at least positive semi-definite. Hence, $\Delta\boldsymbol{U}$ that satisfies the FONC is

$$\Delta\boldsymbol{U}^{*}=\left(\boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}^{\top}\boldsymbol{Q}\left(\boldsymbol{r}_p-\boldsymbol{W}\boldsymbol{x}_a\right). \tag{13.18}$$

Now, applying the second derivative test to $J(\Delta\boldsymbol{U})$, which we refer to as the second-order sufficiency condition (SONC), we obtain

$$\begin{aligned}\frac{\partial^2 J}{\partial\Delta\boldsymbol{U}^2} &= \boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\\ &> 0,\end{aligned}$$

which implies that $\Delta\boldsymbol{U}^{*}$ is a strict minimizer of J.

Using (13.18), we compute $\Delta\boldsymbol{u}[k]$,

$$\begin{aligned}\Delta\boldsymbol{u}[k] &= \overbrace{\left[\begin{array}{cccc}\boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O}\end{array}\right]}^{N_p \text{ block matrices}}\left(\boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}^{\top}\boldsymbol{Q}\left(\boldsymbol{r}_p-\boldsymbol{W}\boldsymbol{x}_a\right)\\ &= \boldsymbol{K}_{\boldsymbol{r}}\boldsymbol{r}_p-\boldsymbol{K}_{\boldsymbol{x}}\Delta\boldsymbol{x}[k]-\boldsymbol{K}_{\boldsymbol{y}}\boldsymbol{y}[k],\end{aligned} \tag{13.19}$$

where

$$\boldsymbol{K}_{\boldsymbol{r}}=\left[\begin{array}{cccc}\boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O}\end{array}\right]\left(\boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}^{\top}\boldsymbol{Q},\quad \boldsymbol{K}_{\boldsymbol{x}}=\boldsymbol{K}_{\boldsymbol{r}}\boldsymbol{W}\left[\begin{array}{c}\boldsymbol{I}_n\\ \boldsymbol{O}\end{array}\right],$$

and

$$\boldsymbol{K}_{\boldsymbol{y}}=\boldsymbol{K}_{\boldsymbol{r}}\boldsymbol{W}\left[\begin{array}{c}\boldsymbol{O}\\ \boldsymbol{I}_p\end{array}\right].$$

An implementation of the above controller, using a discrete-time integrator, is shown in Figure 13.4.

13.2 MPC With Constraints

An attractive feature of the model-based predictive control approach is that a control engineer can incorporate different types of constraints on the control action. We consider three types of such constraints.

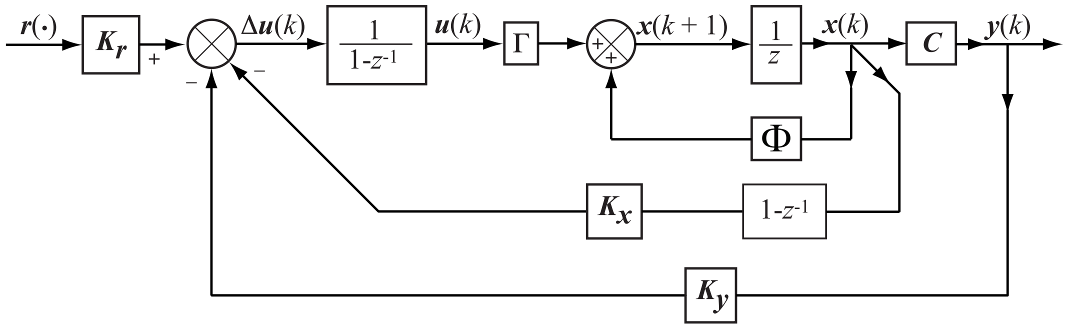


Figure 13.4: Discrete-time MPC for linear time-invariant systems.

13.2.1 Constraints on the Rate of Change of the Control Action

Hard constraints on the rate of change of the control signal can be expressed as

$$\Delta u_i^{\min} \leq \Delta u_i[k] \leq \Delta u_i^{\max}, \quad i = 1, 2, \ldots, m \tag{13.20}$$

Let

$$\Delta \boldsymbol{u}^{\min} = \begin{bmatrix} \Delta u_1^{\min} & \cdots & \Delta u_m^{\min} \end{bmatrix}^\top \quad \text{and} \quad \Delta \boldsymbol{u}^{\max} = \begin{bmatrix} \Delta u_1^{\max} & \cdots & \Delta u_m^{\max} \end{bmatrix}^\top.$$

Then, we can express (13.20) as

$$\Delta \boldsymbol{u}^{\min} \leq \Delta \boldsymbol{u}[k] \leq \Delta \boldsymbol{u}^{\max}. \tag{13.21}$$

The above can be equivalently represented as

$$\begin{bmatrix} -\boldsymbol{I}_m \\ \boldsymbol{I}_m \end{bmatrix} \Delta \boldsymbol{u}[k] \leq \begin{bmatrix} -\Delta \boldsymbol{u}^{\min} \\ \Delta \boldsymbol{u}^{\max} \end{bmatrix}. \tag{13.22}$$

Using the above approach we can represent constraints on the rate of change of the control action over the whole prediction horizon, N_p, in terms of $\Delta \boldsymbol{U}[k]$, by augmenting the above inequality to incorporate constraints for the remaining sampling times. That is, if the above constraints are imposed on the rate of change of the control action for all sampling times within the prediction horizon, then this can be expressed in terms of $\Delta \boldsymbol{U}[k]$ as

$$\begin{bmatrix} -\boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{I}_m & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_m & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \vdots & & & & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & -\boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & -\boldsymbol{I}_m \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}[k] \\ \Delta\boldsymbol{u}[k+1] \\ \vdots \\ \Delta\boldsymbol{u}[k+N_p-2] \\ \Delta\boldsymbol{u}[k+N_p-1] \end{bmatrix} \leq \begin{bmatrix} -\Delta\boldsymbol{u}^{\min} \\ \Delta\boldsymbol{u}^{\max} \\ -\Delta\boldsymbol{u}^{\min} \\ \Delta\boldsymbol{u}^{\max} \\ \vdots \\ -\Delta\boldsymbol{u}^{\min} \\ \Delta\boldsymbol{u}^{\max} \\ -\Delta\boldsymbol{u}^{\min} \\ \Delta\boldsymbol{u}^{\max} \end{bmatrix}. \tag{13.23}$$

On the other hand, if the constraints on the rate of change of the control are imposed only on the first component of $\Delta \boldsymbol{U}[k]$, then we express this in terms of $\Delta \boldsymbol{U}[k]$ as

$$\begin{bmatrix} -\boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \Delta\boldsymbol{U}[k] \leq \begin{bmatrix} -\Delta\boldsymbol{u}^{\min} \\ \Delta\boldsymbol{u}^{\max} \end{bmatrix}.$$

13.2.2 Constraints on the Control Action Magnitude

Hard constraints on the control action magnitude at the time sampling k have the form

$$u_i^{\min} \leq u_i[k] \leq u_i^{\max}, \quad i = 1, 2 \ldots, m \tag{13.24}$$

We now express constraints on the control action magnitude over the whole prediction horizon in term of $\Delta \boldsymbol{U}[k]$. To accomplish our goal, we first note that

$$\begin{aligned} \boldsymbol{u}[k] &= \boldsymbol{u}[k-1] + \Delta\boldsymbol{u}[k] \\ &= \boldsymbol{u}[k-1] + \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} \end{bmatrix} \Delta\boldsymbol{U}[k], \end{aligned}$$

where, recall that $\Delta \boldsymbol{U}[k] = \begin{bmatrix} \Delta\boldsymbol{u}[k] & \Delta\boldsymbol{u}[k+1] & \cdots & \Delta\boldsymbol{u}[k+N_p-1] \end{bmatrix}^\top$.

Similarly,

$$\begin{aligned} \boldsymbol{u}[k+1] &= \boldsymbol{u}[k] + \Delta\boldsymbol{u}[k+1] \\ &= \boldsymbol{u}[k-1] + \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} \end{bmatrix} \Delta\boldsymbol{U}[k] + \Delta\boldsymbol{u}[k+1] \\ &= \boldsymbol{u}[k-1] + \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{I}_m & \cdots & \boldsymbol{O} \end{bmatrix} \Delta\boldsymbol{U}[k]. \end{aligned}$$

Continuing in this manner, we obtain

$$\begin{bmatrix} \boldsymbol{u}[k] \\ \boldsymbol{u}[k+1] \\ \vdots \\ \boldsymbol{u}[k+N_p-1] \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{I}_m \\ \vdots \\ \boldsymbol{I}_m \end{bmatrix} \boldsymbol{u}[k-1] + \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{I}_m & \boldsymbol{I}_m & \cdots & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{I}_m & \boldsymbol{I}_m & \cdots & \boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}[k] \\ \Delta\boldsymbol{u}[k+1] \\ \vdots \\ \Delta\boldsymbol{u}[k+N_p-1] \end{bmatrix}. \tag{13.25}$$

Let

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{I}_m \\ \vdots \\ \boldsymbol{I}_m \end{bmatrix} \quad \text{and} \quad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{I}_m & \boldsymbol{I}_m & \cdots & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{I}_m & \boldsymbol{I}_m & \cdots & \boldsymbol{I}_m \end{bmatrix}.$$

Then, we can represent (13.25) as

$$\boldsymbol{U}[k] = \boldsymbol{E}\boldsymbol{u}[k-1] + \boldsymbol{H}\Delta\boldsymbol{U}[k]. \tag{13.26}$$

Suppose now that we are faced with constructing a control action subject to the following constraints,

$$\boldsymbol{U}^{\min} \le \boldsymbol{U}[k] \le \boldsymbol{U}^{\max}.$$

The above constraints can be equivalently represented as

$$\begin{bmatrix} -\boldsymbol{U}[k] \\ \boldsymbol{U}[k] \end{bmatrix} \le \begin{bmatrix} -\boldsymbol{U}^{\min} \\ \boldsymbol{U}^{\max} \end{bmatrix},$$

Taking into account (13.26), we write the above as

$$\begin{bmatrix} -\left(\boldsymbol{E}\boldsymbol{u}[k-1] + \boldsymbol{H}\Delta\boldsymbol{U}[k]\right) \\ \boldsymbol{E}\boldsymbol{u}[k-1] + \boldsymbol{H}\Delta\boldsymbol{U}[k] \end{bmatrix} \le \begin{bmatrix} -\boldsymbol{U}^{\min} \\ \boldsymbol{U}^{\max} \end{bmatrix}. \tag{13.27}$$

The above, in turn, can be represented as

$$\begin{bmatrix} -\boldsymbol{H} \\ \boldsymbol{H} \end{bmatrix} \Delta\boldsymbol{U}[k] \le \begin{bmatrix} -\boldsymbol{U}^{\min} + \boldsymbol{E}\boldsymbol{u}[k-1] \\ \boldsymbol{U}^{\max} - \boldsymbol{E}\boldsymbol{u}[k-1] \end{bmatrix}. \tag{13.28}$$

In a special case, when a control designer elects to impose constraints only on the first component of $\Delta\boldsymbol{U}[k]$, that is, on $\Delta\boldsymbol{u}[k]$ only, then this scenario is expressed in terms of $\Delta\boldsymbol{U}[k]$ as

$$\begin{bmatrix} -\boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{I}_m & \boldsymbol{O} & \cdots & \boldsymbol{O} \end{bmatrix} \Delta\boldsymbol{U}[k] \le \begin{bmatrix} -\boldsymbol{U}^{\min} + \boldsymbol{E}\boldsymbol{u}[k-1] \\ \boldsymbol{U}^{\max} - \boldsymbol{E}\boldsymbol{u}[k-1] \end{bmatrix}. \tag{13.29}$$

13.2.3 Constraints on the Plant Output

Recall from (13.16) that the predicted plant output is, $\boldsymbol{Y}[k] = \boldsymbol{W}\boldsymbol{x}_a[k] + \boldsymbol{Z}\Delta\boldsymbol{U}[k]$. Suppose now that the following constraints are imposed on the predicted plant's output,

$$\boldsymbol{Y}^{\min} \leq \boldsymbol{Y}[k] \leq \boldsymbol{Y}^{\max}.$$

We represent the above as

$$\begin{bmatrix} -\boldsymbol{Y}[k] \\ \boldsymbol{Y}[k] \end{bmatrix} \leq \begin{bmatrix} -\boldsymbol{Y}^{\min} \\ \boldsymbol{Y}^{\max} \end{bmatrix}.$$

Taking into the account the expression for $\boldsymbol{Y}$ given by (13.16), we represent the above as

$$\begin{bmatrix} -\boldsymbol{Z} \\ \boldsymbol{Z} \end{bmatrix} \Delta\boldsymbol{U}[k] \leq \begin{bmatrix} -\boldsymbol{Y}^{\min} + \boldsymbol{W}\boldsymbol{x}_a[k] \\ \boldsymbol{Y}^{\max} - \boldsymbol{W}\boldsymbol{x}_a[k] \end{bmatrix}. \tag{13.30}$$

The above discussion clearly demonstrates that we need an effective method of minimizing a function of many variables, $J(\Delta\boldsymbol{U})$, subject to inequality constraints such as given by (13.22), (13.27), and (13.30), or their combination. In the following, we present a powerful method for solving such optimization problems. This is the method that we shall use to implement our MPCs.

Example 13.1 We formulate the optimization problem that the model predictive controller is to solve at each sampling period for a discrete dynamic system model,

$$\begin{aligned} \boldsymbol{x}[k+1] &= \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \boldsymbol{x}[k] + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u[k] \\ y[k] &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}[k], \end{aligned}$$

The system output is to track the unit step. The prediction horizon is $N_p = 2$. The constraints on the control signal are to be imposed only on the first component of $\boldsymbol{U}[k]$ and $\Delta\boldsymbol{U}[k]$, respectively, and they have the form:

$$-3 \leq u[k] \leq 5 \quad \text{and} \quad -2 \leq \Delta u[k] \leq 1.$$

Because $N_p = 2$,

$$\Delta\boldsymbol{U}[k] = \begin{bmatrix} \Delta u[k] \\ \Delta u[k+1] \end{bmatrix}.$$

The constraints on the control magnitude on the first component of $\boldsymbol{U}[k]$ can be expressed as

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \Delta\boldsymbol{U}[k] \leq \begin{bmatrix} 3 + u[k-1] \\ 5 - u[k-1] \end{bmatrix}$$

The constraints on the first component of $\Delta \boldsymbol{U}[k]$ have the form

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \Delta \boldsymbol{U}[k] \leq \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

Let

$$\boldsymbol{Y} = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \end{bmatrix} = \boldsymbol{W}\boldsymbol{x}_a[k] + \boldsymbol{Z}\Delta\boldsymbol{U} \quad \text{and} \quad \boldsymbol{r}_p = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

where $\boldsymbol{Y}$ is the vector containing predicted plant outputs over the prediction horizon and $\boldsymbol{r}_p$ is the vector of the reference signals over the prediction horizon. Then, the optimization problem has the form

$$\min \tfrac{1}{2} \left(\boldsymbol{r}_p - \boldsymbol{Y}\right)^\top \boldsymbol{Q} \left(\boldsymbol{r}_p - \boldsymbol{Y}\right) + \tfrac{1}{2}\Delta\boldsymbol{U}^\top \boldsymbol{R}\Delta\boldsymbol{U}$$

subject to

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ -1 & 0 \\ 1 & 0 \end{bmatrix} \Delta \boldsymbol{U}[k] \leq \begin{bmatrix} 3+u[k-1] \\ 5-u[k-1] \\ 2 \\ 1 \end{bmatrix},$$

where $\boldsymbol{Q} = \boldsymbol{Q}^\top \geq 0$ and $\boldsymbol{R} = \boldsymbol{R}^\top > 0$. For example, $\boldsymbol{Q} = \boldsymbol{R} = \boldsymbol{I}_2$.

13.3 Solving Constrained Optimization Problems

If follows from the discussion in the previous section that at each sampling time the MPC calls for a solution to a constrained optimization problem of the form,

$$\begin{array}{ll} \text{minimize} & J(\Delta\boldsymbol{U}) \\ \text{subject to} & \boldsymbol{g}(\Delta\boldsymbol{U}) \leq \boldsymbol{0}, \end{array}$$

where $\boldsymbol{g}(\Delta\boldsymbol{U}) \leq \boldsymbol{0}$ contains inequality constrains given by (13.22), (13.27), and (13.30), or their combination. In this section, we present an iterative methods for solving the above optimization problems. To proceed, we first present the descent gradient method, followed by the Newton's method, for solving unconstrained optimization problems of the form,

$$\text{minimize } J(\Delta\boldsymbol{U})$$

13.3.1 Gradient Descent Method

For the sake of simplicity, we denote the argument of a function of many variables as $\boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{R}^N$, and the function that we will be minimizing will be denoted as f, where $f : \mathbb{R}^N \to \mathbb{R}$.

The method of the gradient descent is based on the following property of the gradient of a differentiable function, f, on $\mathbb{R}^N$:

Theorem 13.1 *At a given point $\boldsymbol{x}^{(0)}$, the vector*

$$\boldsymbol{v} = -\nabla f\left(\boldsymbol{x}^{(0)}\right)$$

points in the direction of most rapid decrease of f and the rate of increase of f at $\boldsymbol{x}^{(0)}$ in the direction $\boldsymbol{v}$ is $-\left\|\nabla f\left(\boldsymbol{x}^{(0)}\right)\right\|$, equivalently, the rate of decrease of f at $\boldsymbol{x}^{(0)}$ in the direction $\boldsymbol{v}$ is $\left\|\nabla f\left(\boldsymbol{x}^{(0)}\right)\right\|$.

Thus, if we wish to minimize a differentiable function, f, then moving in the direction of the negative gradient is a good direction. The gradient descent algorithm rests on the above observation and has the form,

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha \nabla f\left(\boldsymbol{x}^{(k)}\right), \tag{13.31}$$

where $\alpha > 0$ is a step size.

13.3.2 Second-Order Necessary Conditions for a Minimum

Suppose we are given a function f of one variable x. Recall a well-known theorem referred to as the second-order Taylor's formula or the extended law of mean (see [272, p. 1] or [62, Section 5.6]):

Theorem 13.2 *Suppose that $f(x)$, $f'(x)$, $f''(x)$ exist on the closed interval $[a, b] = \{x \in \mathbb{R} : a \le x \le b\}$. If x^*, x are any two different points of $[a, b]$, then there exists a point z strictly between x^* and x such that*

$$f(x) = f\left(x^*\right) + f'\left(x^*\right)\left(x - x^*\right) + \frac{f''(z)}{2}\left(x - x^*\right)^2 .$$

Using the above formula we observe that if

- $f'\left(x^*\right) = 0$, and
- $f''\left(x^*\right) > 0$,

then

$$f(x) = f\left(x^*\right) + \text{ a positive number}$$

for all x "close" to x^*. Indeed, if $f''(x)$ is continuous at x^* and $f''\left(x^*\right) > 0$, then $f''\left(x\right) > 0$ for all x in some neighborhood of x^*. Therefore,

$$f(x) > f\left(x^*\right) \qquad \text{for all } x \text{ close to } x^*,$$

which means that x^* is a strict local minimizer of f.

We now extend the above result for functions of many variables. To proceed, we need the second-order Taylor's formula for such functions that can be found in [272, p. 11].

Theorem 13.3 *Suppose that $\boldsymbol{x}^*$, $\boldsymbol{x}$ are points in $\mathbb{R}^N$ and that f is a function of N variables with continuous first and second partial derivatives on some open set containing the line segment*

$$[\boldsymbol{x}^*, \boldsymbol{x}] = \left\{\boldsymbol{w} \in \mathbb{R}^N : \boldsymbol{w} = \boldsymbol{x}^* + t\left(\boldsymbol{x} - \boldsymbol{x}^*\right); 0 \le t \le 1\right\}$$

joining $\boldsymbol{x}^*$ and $\boldsymbol{x}$. Then there exists a $\boldsymbol{z} \in [\boldsymbol{x}^*, \boldsymbol{x}]$ such that

$$f(\boldsymbol{x}) = f\left(\boldsymbol{x}^*\right) + \nabla f\left(\boldsymbol{x}^*\right)^\top \left(\boldsymbol{x} - \boldsymbol{x}^*\right) + \frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{x}^*\right)^\top \boldsymbol{F}\left(\boldsymbol{z}\right)\left(\boldsymbol{x} - \boldsymbol{x}^*\right), \tag{13.32}$$

where $\boldsymbol{F}(\cdot)$ is the Hessian of f, that is, the second derivative of f.

Hence if

- $\boldsymbol{x}^*$ is a critical point, that is, $\nabla f\left(\boldsymbol{x}^*\right) = \mathbf{0}$, and
- $\boldsymbol{F}\left(\boldsymbol{x}^*\right) > 0$,

then using (13.32), we conclude that

$$f(\boldsymbol{x}) = f\left(\boldsymbol{x}^*\right) + 0 + \text{a positive number}$$

for all $\boldsymbol{x}$ in a neighborhood of $\boldsymbol{x}^*$. Therefore for all $\boldsymbol{x} \neq \boldsymbol{x}^*$ in some neighborhood of $\boldsymbol{x}^*$, we have

$$f(\boldsymbol{x}) > f\left(\boldsymbol{x}^*\right),$$

which implies that $\boldsymbol{x}^*$ is a strict local minimizer.

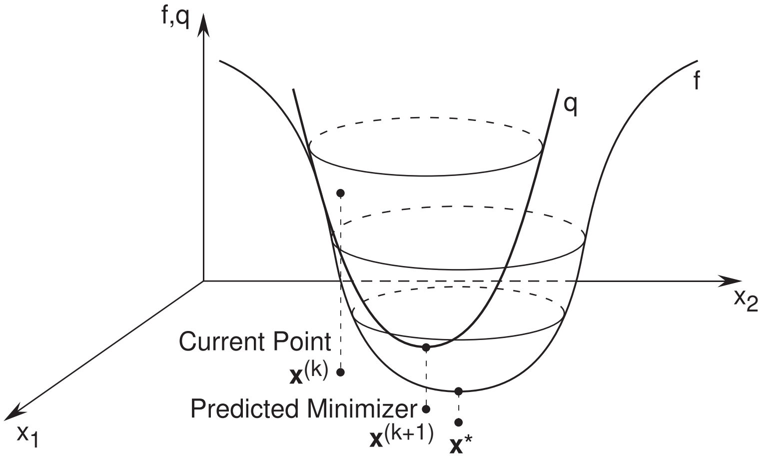


Figure 13.5: Newton's method minimizes the quadratic approximation of the objective function that utilizes first and second derivatives of the optimized function.

13.3.3 Newton's Method

The idea behind Newton's method for function minimization is to minimize the quadratic approximation rather than the function itself as illustrated in Figure 13.5. Newton's method seeks a critical point, $\boldsymbol{x}^*$, of a given function. If at this critical point we have $\boldsymbol{F}(\boldsymbol{x}^*) > 0$, then $\boldsymbol{x}^*$ is a strict local minimizer of f.

We obtain a quadratic approximation q of f at $\boldsymbol{x}^*$ from the second-order Taylor series expansion of f about $\boldsymbol{x}^*$,

$$q(\boldsymbol{x}) = f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^\top \boldsymbol{F}(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*).$$

Note that

$$q(\boldsymbol{x}^*) = f(\boldsymbol{x}^*), \quad \nabla q(\boldsymbol{x}^*) = \nabla f(\boldsymbol{x}^*),$$

as well as their Hessians, that is, their second derivatives evaluated at $\boldsymbol{x}^*$ are equal.

A critical point of q can be obtained by solving the algebraic equation,

$$\nabla q(\boldsymbol{x}) = \boldsymbol{0},$$

that is, by solving the equation

$$\nabla q(\boldsymbol{x}) = \nabla f(\boldsymbol{x}^*) + \boldsymbol{F}(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*) = \boldsymbol{0}.$$

Suppose now that we have a quadratic approximation of f at a point $\boldsymbol{x}^{(k)}$, that is,

$$q(\boldsymbol{x}) = f\left(\boldsymbol{x}^{(k)}\right) + \nabla f\left(\boldsymbol{x}^{(k)}\right)^\top \left(\boldsymbol{x} - \boldsymbol{x}^{(k)}\right) + \frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{x}^{(k)}\right)^\top \boldsymbol{F}\left(\boldsymbol{x}^{(k)}\right)\left(\boldsymbol{x} - \boldsymbol{x}^{(k)}\right).$$

We assume that $\det \boldsymbol{F}\left(\boldsymbol{x}^{(k)}\right) \neq 0$. Denoting the solution to the above equation as $\boldsymbol{x}^{(k+1)}$, we obtain

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{F}\left(\boldsymbol{x}^{(k)}\right)^{-1} \nabla f\left(\boldsymbol{x}^{(k)}\right)$$

The above is known as the Newton's method for minimizing a function of many variables f. Note that $\boldsymbol{x}^{(k+1)}$ is a critical point of the quadratic function q that approximates f at $\boldsymbol{x}^{(k)}$.

A computationally efficient representation of Newton's algorithm has the form,

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \Delta \boldsymbol{x}^{(k)},$$

where $\Delta \boldsymbol{x}^{(k)}$ is obtained by solving the equation,

$$\boldsymbol{F}\left(\boldsymbol{x}^{(k)}\right) \Delta \boldsymbol{x}^{(k)} = \nabla f\left(\boldsymbol{x}^{(k)}\right).$$

13.4 Minimization Subject to Equality Constraints

We now discuss the problem of finding a point $\boldsymbol{x} \in \mathbb{R}^N$ that minimizes $f(\boldsymbol{x})$ subject to equality constraints,

$$\left.\begin{array}{rcl} h_1(\boldsymbol{x}) & = & 0 \\ h_2(\boldsymbol{x}) & = & 0 \\ & \vdots & \\ h_M(\boldsymbol{x}) & = & 0 \end{array}\right\}$$

where $M \leq N$. We write the above equality constraints in a compact form as

$$\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0},$$

where $\boldsymbol{h} : \mathbb{R}^N \to \mathbb{R}^M$. We refer to the set of points satisfying the above constraints as the surface.

We now introduce the notion of the tangent plane to the surface $S = \{\boldsymbol{x} : \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}\}$ at a point $\boldsymbol{x}^* \in S$. First, we define a curve on the surface S as a family of points $\boldsymbol{x}(t) \in S$ continuously parameterized by t for $t \in [a, b]$. The curve is differentiable if $\dot{\boldsymbol{x}}(t) = d\boldsymbol{x}(t)/dt$ exists. A curve is said to pass through the point $\boldsymbol{x}^* \in S$ if $\boldsymbol{x}^* = \boldsymbol{x}(t^*)$ for some $t^* \in [a, b]$. The tangent plane to the surface $S = \{\boldsymbol{x} : \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}\}$ at $\boldsymbol{x}^*$ is the collection of the derivatives at $\boldsymbol{x}^*$ of all differentiable curves on S that pass through $\boldsymbol{x}^*$.

To proceed, we need the notion of a regular point of the constraints. We say that $\boldsymbol{x}^*$ satisfying the constraints, that is, $\boldsymbol{h}(\boldsymbol{x}^*) = \mathbf{0}$, is a regular point of the constraints if the gradient vectors,

$$\nabla h_1(\boldsymbol{x}^*), \ldots, \nabla h_M(\boldsymbol{x}^*)$$

are linearly independent.

One can show, see, for example [224, p. 298], that the tangent space at a regular point $\boldsymbol{x}^*$, denoted $T(\boldsymbol{x}^*)$, to the surface $\{\boldsymbol{x} : \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}\}$ at the regular point $\boldsymbol{x}^*$ is

$$T(\boldsymbol{x}^*) = \left\{ \boldsymbol{y} : \begin{bmatrix} \nabla h_1(\boldsymbol{x}^*)^\top \\ \vdots \\ \nabla h_M(\boldsymbol{x}^*)^\top \end{bmatrix} \boldsymbol{y} = \mathbf{0} \right\}. \tag{13.33}$$

With the above notions in place, we are ready to state and prove the following result which is known as the first-order necessary condition (FONC) for function minimization subject to equality constraints.

Theorem 13.4 *Let $\boldsymbol{x}^*$ be a local minimizer (or maximizer) of f subject to the constraints $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ and let $\boldsymbol{x}^*$ be a regular point of the constraints. Then there exists a vector $\boldsymbol{\lambda}^*$ such that*

$$\nabla f(\boldsymbol{x}^*) + \begin{bmatrix} \nabla h_1(\boldsymbol{x}^*) & \cdots & \nabla h_M(\boldsymbol{x}^*) \end{bmatrix} \boldsymbol{\lambda}^* = \mathbf{0}.$$

Proof Let $\boldsymbol{x}(t)$ be a differentiable curve passing through $\boldsymbol{x}^*$ on the surface $S = \{\boldsymbol{x} : \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}\}$ such that $\dot{\boldsymbol{x}}(t^*) = \boldsymbol{y}$ where $t^* \in [a, b]$. Note that the tangent space is given by (13.33). Because $\boldsymbol{x}^*$ is a local minimizer of f on S, we have

$$\left. \frac{d}{dt} f(\boldsymbol{x}(t)) \right|_{t=t^*} = 0.$$

Applying the chain rule to the above gives

$$\nabla f(\boldsymbol{x}^*)^\top \boldsymbol{y} = 0.$$

Thus $\nabla f(\boldsymbol{x}^*)$ is orthogonal to the tangent space $T(\boldsymbol{x}^*)$. That is, $\nabla f(\boldsymbol{x}^*)$ is a linear combination of the gradients $\nabla h_1(\boldsymbol{x}^*), \ldots, \nabla h_M(\boldsymbol{x}^*)$. This fact can be expressed as

$$\nabla f(\boldsymbol{x}^*) + \begin{bmatrix} \nabla h_1(\boldsymbol{x}^*) & \cdots & \nabla h_M(\boldsymbol{x}^*) \end{bmatrix} \boldsymbol{\lambda}^* = \mathbf{0}$$

for some constant vector $\boldsymbol{\lambda}^* \in \mathbb{R}^M$.

□

The vector $\boldsymbol{\lambda}^*$ is called the vector of Lagrange multipliers.

We now introduce the Lagrangian associated with the constrained optimization problem,

$$l(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{h}(\boldsymbol{x}).$$

Then the FONC can be expressed as

$$\begin{aligned}\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{\lambda}) &= \mathbf{0}\\ \nabla_{\boldsymbol{\lambda}} l(\boldsymbol{x},\boldsymbol{\lambda}) &= \mathbf{0}.\end{aligned}$$

Note that the second of the above condition is equivalent to $\boldsymbol{h}(\boldsymbol{x})=\mathbf{0}$, that is,

$$\nabla_{\boldsymbol{\lambda}} l(\boldsymbol{x},\boldsymbol{\lambda}) = \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}.$$

Equivalently the FONC can be written as

$$\begin{aligned}\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{\lambda}) &= \mathbf{0}\\ \boldsymbol{h}(\boldsymbol{x}) &= \mathbf{0}.\end{aligned}$$

We now apply Newton's method to solve the above system of equations iteratively,

$$\begin{bmatrix}\boldsymbol{x}^{(k+1)}\\ \boldsymbol{\lambda}^{(k+1)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}^{(k)}\\ \boldsymbol{\lambda}^{(k)}\end{bmatrix} + \begin{bmatrix}\boldsymbol{d}^{(k)}\\ \boldsymbol{y}^{(k)}\end{bmatrix},$$

where $\boldsymbol{d}^{(k)}$ and $\boldsymbol{y}^{(k)}$ are obtained by solving the matrix equation,

$$\begin{bmatrix}\boldsymbol{L}\left(\boldsymbol{x}^{(k)},\boldsymbol{\lambda}^{(k)}\right) & D\boldsymbol{h}\left(\boldsymbol{x}^{(k)}\right)^{\top}\\ D\boldsymbol{h}\left(\boldsymbol{x}^{(k)}\right) & \boldsymbol{O}\end{bmatrix}\begin{bmatrix}\boldsymbol{d}^{(k)}\\ \boldsymbol{y}^{(k)}\end{bmatrix} = \begin{bmatrix}-\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{\lambda})\\ -\boldsymbol{h}\left(\boldsymbol{x}^{(k)}\right)\end{bmatrix},$$

where $\boldsymbol{L}(\boldsymbol{x},\boldsymbol{\lambda})$ is the Hessian of $l(\boldsymbol{x},\boldsymbol{\lambda})$ with respect to $\boldsymbol{x}$, and $D\boldsymbol{h}(\boldsymbol{x})$ is the Jacobian matrix of $\boldsymbol{h}(\boldsymbol{x})$, that is,

$$D\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix}\nabla h_1(\boldsymbol{x})^{\top}\\ \vdots\\ \nabla h_M(\boldsymbol{x})^{\top}\end{bmatrix}.$$

The above algorithm is also referred to as in the literature as sequential quadratic programming (SQP); see, for example [252, Section 15.5].

13.5 Lagrangian Algorithm for Equality Constraints

The first-order Lagrangian algorithm for the optimization problem involving minimizing f subject to the equality constraints, $\boldsymbol{h}(\boldsymbol{x})=\mathbf{0}$, has the form,

$$\begin{aligned}\boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} - \alpha_k\left(\nabla f\left(\boldsymbol{x}^{(k)}\right) + D\boldsymbol{h}\left(\boldsymbol{x}^{(k)}\right)^{\top}\boldsymbol{\lambda}^{(k)}\right)\\ \boldsymbol{\lambda}^{(k+1)} &= \boldsymbol{\lambda}^{(k)} + \beta_k\boldsymbol{h}\left(\boldsymbol{x}^{(k)}\right),\end{aligned}$$

where α_k and β_k are positive constants. Note that the update for $\boldsymbol{x}^{(k)}$ is a descent gradient for minimizing the Lagrangian with respect to $\boldsymbol{x}$, while the update for $\boldsymbol{\lambda}^{(k)}$ is a gradient ascent for maximizing the Lagrangian with respect to $\boldsymbol{\lambda}$. For an analysis of the above algorithm, we recommend [62, pp. 522–524]

13.6 Minimization Subject to Inequality Constraints

We discuss the problem of finding a point $\boldsymbol{x} \in \mathbb{R}^N$ that minimizes $f(\boldsymbol{x})$ subject to inequality constraints,

$$\left.\begin{array}{rcl} g_1(\boldsymbol{x}) & \leq & 0 \\ g_2(\boldsymbol{x}) & \leq & 0 \\ & \vdots & \\ g_P(\boldsymbol{x}) & \leq & 0 \end{array}\right\}$$

We write the above inequality constraints in a compact form as

$$\boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0},$$

where $\boldsymbol{h} : \mathbb{R}^N \to \mathbb{R}^P$.

Let $\boldsymbol{x}^*$ be a point satisfying the constraints, that is, $\boldsymbol{g}(\boldsymbol{x}^*) \leq \mathbf{0}$, and let J be the set of indices j for which $g_j(\boldsymbol{x}^*) = 0$. Then $\boldsymbol{x}^*$ is said to be a regular point of the constraints $\boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}$ if the gradient vectors,

$$\nabla g_j(\boldsymbol{x}^*), \quad j \in J,$$

are linearly independent.

We say that a constraint $g_j(\boldsymbol{x}) \leq 0$ is active at $\boldsymbol{x}^*$ if $g_j(\boldsymbol{x}^*) = 0$. Thus the index set J, defined above, contains indices of active constraints.

Theorem 13.5 *Let $\boldsymbol{x}^*$ be a regular point and a local minimizer of f subject to $\boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}$. Then there exists a vector $\boldsymbol{\mu}^* \in \mathbb{R}^P$ such that*

1. $\boldsymbol{\mu}^* \geq 0$,
2. $\nabla f(\boldsymbol{x}^*) + \begin{bmatrix} \nabla g_1(\boldsymbol{x}^*) & \cdots & \nabla g_P(\boldsymbol{x}^*) \end{bmatrix} \boldsymbol{\mu}^* = \mathbf{0}$,
3. $\boldsymbol{\mu}^{*\top} \boldsymbol{g}(\boldsymbol{x}^*) = 0$.

Proof First note that because $\boldsymbol{x}^*$ is a relative minimizer over the constraint set $\{\boldsymbol{x} : \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}\}$, it is also a minimizer over a subset of the constraint set obtained by setting the active constraints to zero. Therefore, for the resulting equality constrained problem, by Theorem 13.4, we have

$$\nabla f\left(\boldsymbol{x}^*\right) + \left[\begin{array}{ccc} \nabla g_1\left(\boldsymbol{x}^*\right) & \cdots & \nabla g_P\left(\boldsymbol{x}^*\right) \end{array}\right] \boldsymbol{\mu}^* = \mathbf{0}, \tag{13.34}$$

where $\mu_j^* = 0$ if $g_j\left(\boldsymbol{x}^*\right) < 0$. This means that $\boldsymbol{\mu}^{*\top} \boldsymbol{g}\left(\boldsymbol{x}^*\right) = 0$. Thus μ_i may be non-zero only if the corresponding constraint is active, that is, $g_i\left(\boldsymbol{x}^*\right) = 0$.

It thus remains to show that $\mu^* \geq \mathbf{0}$. We prove this by contraposition. We suppose that for some $k \in J$, we have $\mu_k^* < 0$. Consider next a surface formed by all other active constraints, that is, the surface

$$\{\boldsymbol{x} : g_j(\boldsymbol{x}) = 0, j \in J, j \neq k\}. \tag{13.35}$$

Also consider a tangent space to the above surface at $\boldsymbol{x}^*$. By assumption $\boldsymbol{x}^*$ is a regular point of the active constraints. Therefore, there exists a vector $\boldsymbol{y}$ such that

$$\nabla g_k\left(\boldsymbol{y}\right)^\top \boldsymbol{y} < 0.$$

Let now $\boldsymbol{x}(t)$, $t \in [-a, a]$, $a > 0$, be a curve on the surface (13.35) such that

$$\dot{\boldsymbol{x}}(0) = \boldsymbol{y}.$$

Note that for small t, the curve $\boldsymbol{x}(t)$ is feasible. We apply the transposition operator to the both sides of (13.34) to obtain

$$\nabla f\left(\boldsymbol{x}^*\right)^\top + \sum_{i=1}^{P} \mu_i^* \nabla g_i\left(\boldsymbol{x}^*\right)^\top = \mathbf{0}^\top.$$

Post-multiplying the above by $\boldsymbol{y}$ an taking into account the fact that $\boldsymbol{y}$ belongs to the tangent space to the surface (13.35) gives

$$\begin{aligned} \nabla f\left(\boldsymbol{x}^*\right)^\top \boldsymbol{y} &= -\mu_k^* \nabla g_k\left(\boldsymbol{x}^*\right)^\top \boldsymbol{y} \\ &< 0. \end{aligned}$$

Suppose, without loss of generality, that $\|\boldsymbol{y}\|_2 = 1$. Then, we have that

$$\begin{aligned} \frac{df(\boldsymbol{x}(t))}{dt} &= \nabla f\left(\boldsymbol{x}^*\right)^\top \boldsymbol{y} \\ &< 0, \end{aligned}$$

that is, the rate of increase of f at $\boldsymbol{x}^*$ in the direction $\boldsymbol{y}$ is negative. This would mean that we could decrease the value of f moving just slightly away from $\boldsymbol{x}^*$ along $\boldsymbol{y}$ while, at the same time, preserving feasibility. But this contradicts the minimality of $f(\boldsymbol{x}^*)$. In summary, if $\boldsymbol{x}^*$ is a relative minimizer then we also have the components of $\boldsymbol{\mu}^*$ all non-negative.

□

The vector $\boldsymbol{\mu}^*$ is called the vector of the Karush-Kuhn-Tucker (KKT) multipliers.

13.7 Lagrangian Algorithm for Inequality Constraints

We present a first-order Lagrangian algorithm for the optimization problem involving inequality constraints,

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}, \end{array}$$

where $\boldsymbol{g} : \mathbb{R}^N \to \boldsymbol{g} : \mathbb{R}^P$. The Lagrangian function is

$$l(\boldsymbol{x}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\mu}^\top \boldsymbol{g}(\boldsymbol{x}).$$

The first-order Lagrangian algorithm for the above optimization problem involving minimizing f subject to the inequality constraints, $\boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}$, has the form,

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} - \alpha_k \left(\nabla f\left(\boldsymbol{x}^{(k)}\right) + D\boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right)^\top \boldsymbol{\mu}^{(k)} \right) \\ \boldsymbol{\mu}^{(k+1)} &= \left[\boldsymbol{\mu}^{(k)} + \beta_k \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right)\right]_+, \end{aligned}$$

where the operation $[\cdot]_+ = \max(\cdot, 0)$ is applied component-wise.

For an analysis of the above algorithm, we recommend [62, pp. 524–528]

Example 13.2 We illustrate the design of the discrete MPC subject to the constraints on the control input on a plant modeled by the following continuous state-space model,

$$\left.\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} -0.1 & -3.0 \\ 1 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & 10 \end{bmatrix} \boldsymbol{x}. \end{aligned}\right\} \tag{13.36}$$

The system with the transfer function as the above system was used by Wang [342, p. 70] to test her discrete MPC designs. We use different optimizer than Wang. The reader can compare the results of two different MPC synthesis methods. The MPC controller's objective is to force the plant's output to track the unit step.

We proceed to generate the discrete state-space model employing the MATLAB's command `c2dm`. We use the sampling time $h = 0.1$. The command input is the unit step. The weight matrices in the cost function (13.17) are $\boldsymbol{R} = 0.01\boldsymbol{I}_3$ and $\boldsymbol{Q} = \boldsymbol{I}_3$. The prediction horizon is $N_p = 3$. Our objective is to design a discrete-MPC subject to the constraints on the control of the form,

$$-0.3 \le u[k] \le 0.5$$

First, we consider the case when the constraints are imposed only on the first component of $\boldsymbol{U}(k)$. The constraints take the form of (13.29). Thus, in our example, the constraints can be written as,

$$\begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u[k] \\ \Delta u[k+1] \\ \Delta u[k+2] \end{bmatrix} \le \begin{bmatrix} 0.3 + u[k-1] \\ 0.5 - u[k-1] \end{bmatrix}.$$

We employed the first-order Lagrangian algorithm for inequality constraints presented in Section 13.7, where $\alpha_k = \beta_k = 0.005$. We set zero initial conditions. Plots of the control effort and the plant's output are shown in Figure 13.6. We wish to emphasize that the simulations were performed for the discretized plant. Thus, the control and output were evaluated at the sampling instances and marked on the plots using circles. For the purpose of visualization the control action was presented in the form of a staircase function while the plant output values were connected by straight lines.

Next we tested the discrete MPC where the constraints, $-0.3 \le u[k] \le 0.5$, were imposed on all the elements of $\boldsymbol{U}[k]$. We use (13.28) to express the constraints, where

$$\begin{bmatrix} -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & -1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta u[k] \\ \Delta u[k+1] \\ \Delta u[k+2] \end{bmatrix} \le \begin{bmatrix} 0.3 + u[k-1] \\ 0.3 + u[k-1] \\ 0.3 + u[k-1] \\ 0.5 - u[k-1] \\ 0.5 - u[k-1] \\ 0.5 - u[k-1] \end{bmatrix}.$$

Example 13.3 In this example, we apply the discrete MPC controller from Example 13.2 to the continuous plant modeled by (13.36). The MPC controller's objective is to force the plant's output to track the unit step. The zero-order hold is applied to the discrete MPC controller's output sequence resulting in a piece-wise constant input to the continuous plant

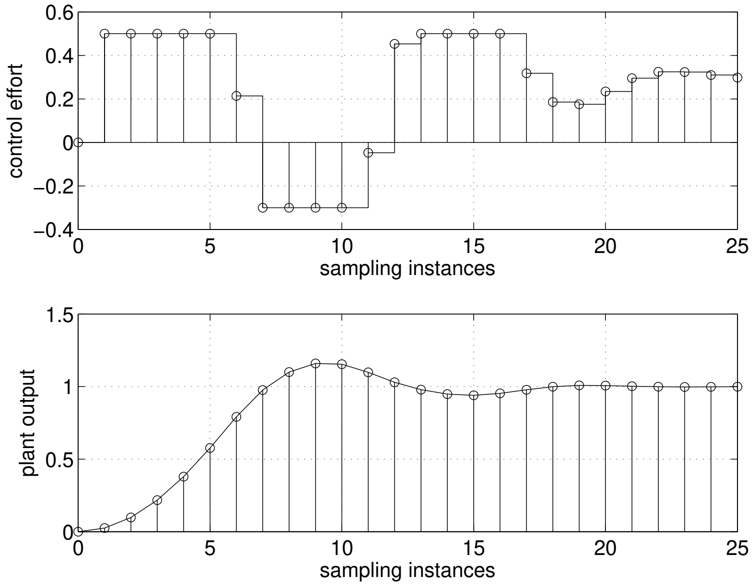


Figure 13.6: Plots of control effort and the plant's output versus time of the closed-loop system of Example 13.2 for the case when the constraint was imposed only on the first component of $\boldsymbol{U}[k]$.

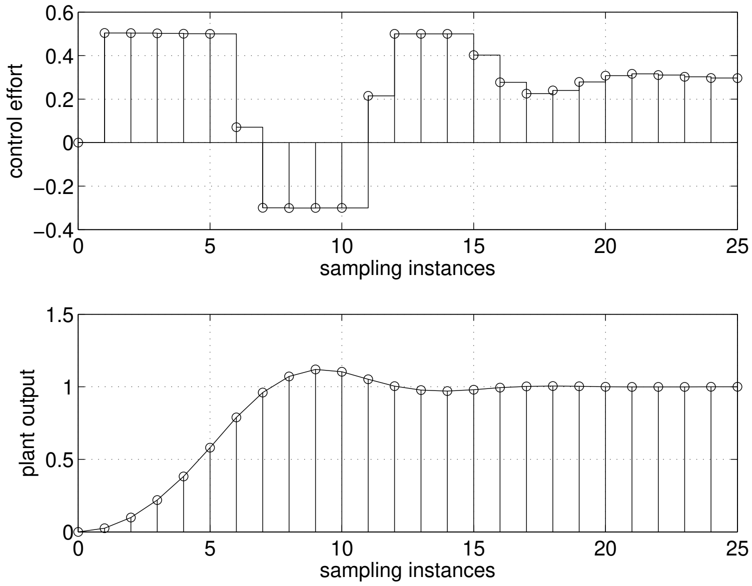


Figure 13.7: Plots of control effort and the plant's output versus time of the closed-loop system of Example 13.2 for the case when the constraints were imposed on all the components of $\boldsymbol{U}[k]$.

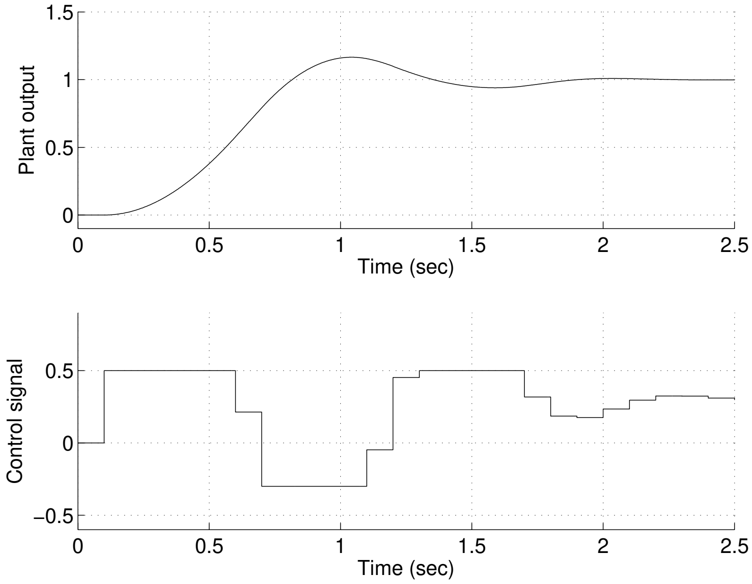


Figure 13.8: Plots of control effort and the plant's output versus time of the closed-loop system of Example 13.3 for the case when the constraints were imposed only on the first component of $\boldsymbol{U}(k)$.

model. In Figure 13.8, we show plots of the plant output as well as the control effort versus time. As in the previous example, the prediction horizon is $N_p = 3$ and the constraints on the control are

$$-0.3 \leq u[k] \leq 0.5$$

13.8 Observer-Based MPC

If the plant state is not accessible, we need to use an observer in the controller implementation as shown in Figure 13.9.

13.9 Notes

The key reason for huge popularity of the MPC approach is its ability to systematically take into account constraints thus allowing processes to operate at the limits of achievable performance [188]. For some impressive industrial applications of MPCs see [188, 49]. Nonlinear model predictive controllers are presented in [233, 240].

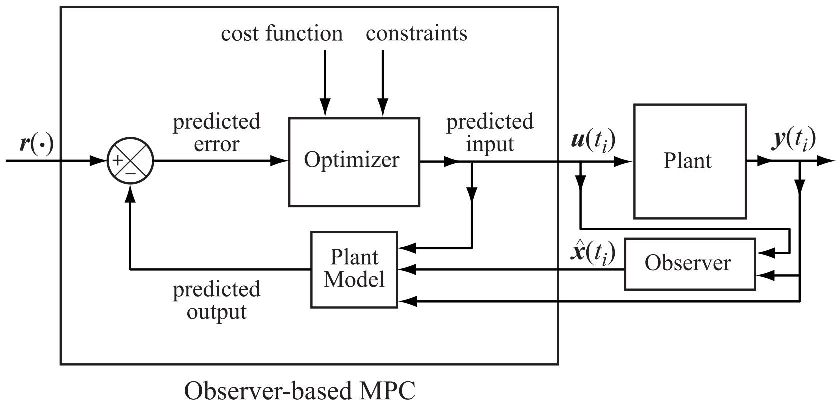


Figure 13.9: MPC implementation when the plant's state is not available.

An insightful description of model predictive control is offered by Mayne et al. [234, pp. 789–790], where they write, "Model predictive control (MPC) or receding horizon control (RHC) is a form of control in which the current control action is obtained by solving *on-line*, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. This is its main difference from conventional control which uses a pre-computed control law. With the clarity gained by hindsight, it can be recognized that the *raison d'être* for model predictive control is its ability to handle control problems where off-line computation of a control law is difficult or impossible although other features, such as its capability for controlling multivariable plants, were initially deemed more important. Nearly every application imposes constraints; actuators are naturally limited in the force (or equivalent) they can apply, safety limits states such as temperature, pressure and velocity and efficiency often dictates steady-state operation close to the boundary of the set of permissible states. The prevalence of hard constraints is accompanied by a dearth of control methods for handling them, despite a continuous demand from industry that has had, in their absence, to resort often to ad hoc methods. Model predictive control is one of few suitable methods, and this fact makes it an important tool for the control engineer, particularly in the process industries where plants being controlled are sufficiently 'slow' to permit its implementation."

13.10 Exercises

Exercise 13.1 Derive a closed-form solution of a general quadratic minimization problem subject to a linear equality constraint,

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^n \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \end{array}$$

where $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $m < n$, and rank $\boldsymbol{A} = m$.

Exercise 13.2 Find all point that satisfy the KKT conditions for the following quadratic minimization problem subject to a linear equality constraint,

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^n \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \end{array}$$

where $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $m < n$, rank $\boldsymbol{A} = m$, and $\boldsymbol{b} \geq 0$.

Exercise 13.3 Consider the following optimization problem,

$$\begin{array}{ll} \text{optimize} & (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{subject to} & x_2 - x_1^2 \geq 0 \\ & 2 - x_1 - x_2 \geq 0 \\ & x_1 \geq 0. \end{array}$$

The point $\boldsymbol{x}^* = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ satisfies the KKT conditions.

(i) Does $\boldsymbol{x}^*$ satisfy the FONC for minimum or maximum? What are the KKT multipliers?

(ii) Does $\boldsymbol{x}^*$ satisfy the SOSC? Carefully justify your answer.

Exercise 13.4 **(i)** Use the KKT conditions to solve the quadratic minimization problem subject to a linear inequality constraint,

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^n \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \end{array}$$

$$\boldsymbol{Q} = \boldsymbol{I}_3, \quad \boldsymbol{c} = \begin{bmatrix} -2 \\ -3 \\ -1 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 \\ 3 & -2 & -3 \\ 1 & -3 & 2 \end{bmatrix}, \text{ and } \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

(ii) Implement in MATLAB the Lagrangian algorithm for inequality constraints of Section 13.7 to solve the above problem. You may wish to compare your solution with that of Wang [342, p. 60], where an active set method was used to solve this numerical problem.

Exercise 13.5 **(i)** Use the KKT conditions to solve the quadratic minimization problem subject to a linear inequality constraint,

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^n \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}, \end{array}$$

$$\boldsymbol{Q} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 3 & 2 \end{bmatrix}, \text{ and } \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}.$$

(ii) Implement in MATLAB the Lagrangian algorithm for inequality constraints of Section 13.7 to solve the above problem. You may wish to compare your solution with that of Wang [342, p. 64], where the Hildreth's quadratic programming algorithm was used to solve this numerical problem.

Exercise 13.6 **(i)** Use the KKT conditions to solve the quadratic minimization problem subject to a linear inequality constraint,

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^n \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}, \end{array}$$

$$\boldsymbol{Q} = \boldsymbol{I}_2, \quad \boldsymbol{c} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \text{ and } \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

(ii) Implement in MATLAB the Lagrangian algorithm for inequality constraints of Section 13.7 to solve the above problem. You may wish to compare your solution with that of Wang [342, p. 67], where the Hildreth's quadratic programming algorithm was used to solve this numerical problem.

Exercise 13.7 Show that if the matrix $\boldsymbol{A}$ in the model (3.27) is invertible, then the matrix $\boldsymbol{\Gamma}$ in the discretized model (3.31) can also be computed as

$$\boldsymbol{\Gamma} = (\boldsymbol{\Phi} - \boldsymbol{I}_n)\boldsymbol{A}^{-1}\boldsymbol{B}.$$

Exercise 13.8 Design an MPC for the linearized model of the nonlinear model of one-link robot manipulator presented in Subsection 2.4.5. Proceed as follows:

(a) Linearize the nonlinear model about the equilibrium pair

$$\boldsymbol{x}_e = \begin{bmatrix} 30\pi/180 & 0 & -4.9 \end{bmatrix}^\top, \quad u_e = -4.9.$$

(b) Discretize the linearized model and use it as the basis for your design. You can help yourself using MATLAB's `c2dm`. You as a designer select the sampling interval h, for example, $h = 0.1$.

(c) First design an MPC without any constraints and implement the controller on the nonlinear continuous model. Write a script that animates the behavior of the closed-loop system for different initial conditions.

(d) Impose constraints on the input and re-design your MPC. Implement the controller on the nonlinear continuous model. Write a script that animates the behavior of the closed-loop system for different initial conditions.

(e) Implement the combined MPC controller-observer compensator on the nonlinear continuous model. Write a script that animates the behavior of the closed-loop system for different initial conditions.

Exercise 13.9 Design an MPC for the linearized model of Exercise 4.5 of the nonlinear model of a double-link robot manipulator of Subsection 2.4.6. Proceed as follows:

(a) Discretize the linearized model and use it as the basis for your design. You can help yourself using MATLAB's `c2dm`. You as a designer select the sampling interval h, for example, $h = 0.1$.

(b) First design an MPC without any constraints and implement the controller on the nonlinear continuous model. Write a script that animates the behavior of the closed-loop system for different initial conditions.

(c) Impose constraints on the input and re-design your MPC. Implement the controller on the nonlinear continuous model. Write a script that animates the behavior of the closed-loop system for different initial conditions.

(d) Implement the combined MPC controller-observer compensator on the nonlinear continuous model. Write a script that animates the behavior of the closed-loop system for different initial conditions.

Chapter 14

Proportional-Integral-Derivative (PID) Control

> Thomas Stevic, a controls engineer at DMAX, said that when designing a solution to a problem, he finds that he typically has two masters to serve. The first master is the person who is responsible for the output of his design. This is often "the person who pays the bills," Stevic said. "These people usually do not care how a machine or process works. Their concern is: Does the machine make what it is supposed to make? While exotic solutions may increase quality, decrease production time, make better use of resources and, from an engineering viewpoint, be very cool to use and inch closer to perfection, their use is not often a benefit to the first master."
> The second master is described by Stevic as the person or people who will "maintain the system after I am gone. Ultimately, my designs have to address the least capable people who may be required to maintain or modify my design."
> Given the requirements of the second master, Stevic says that, given a choice between using adaptive neural network control or a thermostat, if the thermostat will work, that is what he will choose. "If neural nets or fuzzy logic are required to produce the desired result, I will use them, but only after examining the simpler solution," he said. [121, page 57]

In the previous chapters, we analyzed control algorithms that require a mathematical model of the dynamical system to be controlled. The mathematical model is then used to construct a controller. In many practical situations, however, it is not feasible to obtain a quality mathematical model of the controlled system. In this chapter, we present proportional-integral-derivative (PID) controllers that are most useful when a mathematical model of the process to be controlled is not available to the control engineer.

The proportional-integral-derivative (PID) controllers are without a doubt the most widely used controllers in industry today. Åström and Hägglund [15, p. 198] regard the PID controller as the "bread and butter" of control engineering. Knospe [183] estimates that over 90% of control loops employ PID control.

To better appreciate the effectiveness of the PID control strategies, we first discuss a general control design principle referred to as the internal model principle (IMP).

14.1 The Internal Model Principle (IMP)

Consider a block diagram of a closed-loop system shown in Figure 14.1. Using basic rules of

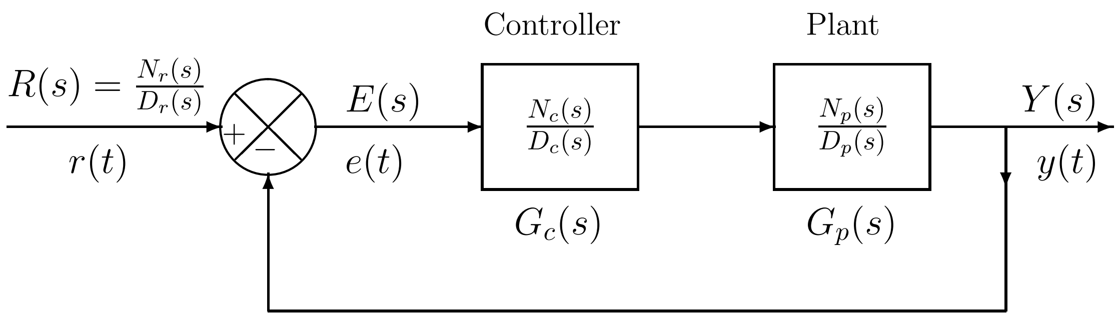


Figure 14.1: Block diagram of a closed-loop control system.

the block diagram algebra yields

$$\begin{aligned} E(s) &= R(s) - G_p(s)G_c(s)E(s) \\ &= \frac{N_r(s)}{D_r(s)} - \frac{N_p(s)N_c(s)}{D_p(s)D_c(s)}E(s). \end{aligned}$$

Performing simple manipulations, we obtain

$$\boxed{E(s) = \frac{D_p(s)D_c(s)}{D_p(s)D_c(s) + N_p(s)N_c(s)} \frac{N_r(s)}{D_r(s)}} \tag{14.1}$$

Our objective is to design a controller $G_c(s) = \frac{N_c(s)}{D_c(s)}$ so that

$$\lim_{t\to\infty} e(t) = \lim_{t\to\infty} (r(t) - y(t)) = 0,$$

where $e(t) = \mathcal{L}^{-1}(E(s))$ is the inverse Laplace transform of $E(s)$.

Suppose that the poles of the Laplace transform of the reference signal $r(t)$ are in the closed right half-plane, that is, they belong to the set $\{s : \Re(s) \geq 0\}$, where $\Re(s)$ denotes the real part of the complex variable s. The polynomial

$$P_c(s) = D_p(s)D_c(s) + N_p(s)N_c(s)$$

is the closed-loop characteristic polynomial (CLCP) of the closed-loop system depicted in Figure 14.1 and the closed-loop characteristic polynomial zeros are the closed-loop poles. Using (14.1), we can easily prove the following result that we refer to as the Internal Model Principle (IMP).

Theorem 14.1 (Internal Model Principle) *In the configuration depicted in Figure 14.1, where the poles of $R(s)$ are in the closed right half-plane,*

$$\lim_{t\to\infty} e(t) = 0$$

if and only if

1. *the closed-loop poles are in the open left-half plane;*
2. *$D_r(s)$ is a factor of the open-loop characteristic polynomial $D_p(s)D_c(s)$, that is, there is a polynomial, say $Q(s)$, such that $D_p(s)D_c(s) = Q(s)D_r(s)$.*

The second condition of the IMP an be interpreted as follows: the controller, $G_c(s)$, must be chosen in such a way that the open-loop transfer function, $G_p(s)G_c(s)$, contains a model of the reference signal to be tracked. If none of the poles of $R(s)$ is a pole of the plant's transfer function, $G_p(s)$, then we can restate the IMP as follows:

Any good tracking controller must stabilize the closed-loop system and must contain a model of the reference signal.

For an alternative approach to the IMP, we recommend Wolovich [353, pp. 254–255].

We illustrate the IMP with two numerical examples. We use the final value theorem that we state next.

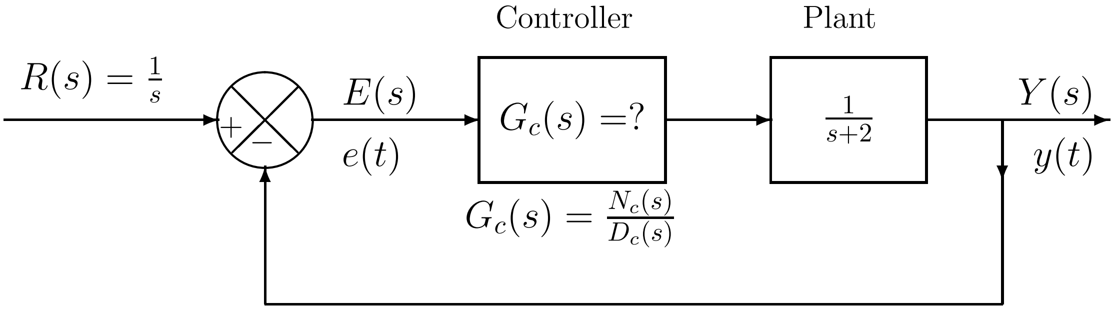


Figure 14.2: A closed-loop control system of Example 14.1.

Theorem 14.2 *Suppose that $F(s) = \mathcal{L}(f(t))$ has poles only in the open left half-plane with possible exception of a single-order pole at $s = 0$. Then*

$$\lim_{s\to 0} sF(s) = \lim_{t\to\infty} f(t).$$

Example 14.1 For the closed-loop system shown in Figure 14.2, our objective is to construct a transfer function $G_c(s)$ such that $\lim_{t\to\infty} e(t) = 0$. The Laplace transform of the tracking error, $e(t)$, is

$$E(s) = \frac{1}{1 + \frac{1}{s+2}\frac{N_c(s)}{D_c(s)}}\frac{1}{s} = \frac{(s+2)D_c(s)}{(s+2)D_c(s) + N_c(s)}\frac{1}{s}.$$

Let $N_c(s) = 1$ and $D_c(s) = s$, that is, let

$$G_c(s) = \frac{1}{s},$$

Then,

$$E(s) = \frac{(s+2)s}{s^2+2s+1}\frac{1}{s} = \frac{s+2}{s^2+2s+1}.$$

It is easy to check using, for example, the final value theorem that $\lim_{t\to\infty} e(t) = 0$. Thus, in this case, a simple integral (I) controller does the job.

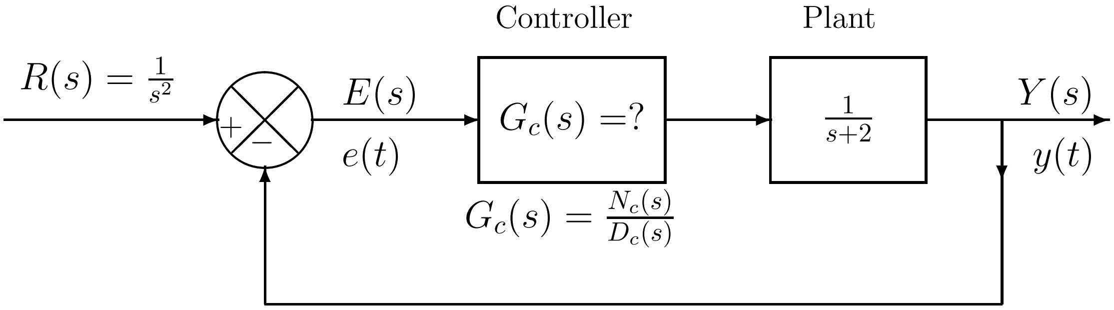


Figure 14.3: A closed-loop control system of Example 14.2.

Example 14.2 Consider the closed-loop system shown in Figure 14.3. We wish to construct a controller $G_c(s) = \frac{N_c(s)}{D_c(s)}$ so that $\lim_{t\to\infty} e(t) = 0$, where the reference signal is the unit ramp function. In Example 14.1, the simple integrator was all that we needed for the unit step as the reference input. Let us try the same controller here. Then, the Laplace transform of the error signal is

$$\begin{aligned} E(s) &= \frac{1}{1 + \frac{1}{s+2}\frac{N_c(s)}{D_c(s)}}\frac{1}{s^2} \\ &= \frac{(s+2)D_c(s)}{(s+2)D_c(s) + N_c(s)}\frac{1}{s^2} \\ &= \frac{s+2}{s^2+2s+1}\frac{1}{s}. \end{aligned}$$

Note that now $e(\infty) = 2$. Thus the simple integrator is not enough now to force the steady-state error to zero. Note that in this case the polynomial $D_r(s)$ was not a factor of $D_c(s)$.

Let us then try a controller that would satisfy the second condition of the IMP, the divisibility condition. If we take the double integrator then the divisibility condition will be satisfied and we obtain

$$E(s) = \frac{s+2}{s^3+2s^2+1}.$$

However, $e(t)$ is divergent because the poles of $E(s)$ are not stable. The three poles of $E(s)$ are located at -2.2056, and $0.1028 \pm j0.6655$. This time the first condition of the Internal Model Principle is not satisfied.

Finally, let us try a controller of the form

$$G_c(s) = \frac{N_c(s)}{D_c(s)} = \frac{s+a}{s^2}, \tag{14.2}$$

where a is a design parameter that should be chosen to satisfy the both conditions of the IMP. Note that the above controller contains the model of the reference signal. Thus, we only need to select the design parameter a so that the CLCP, $P_c(s)$, has its zeros in the open left half-plane, where

$$\begin{aligned} P_c(s) &= D_p(s)D_c(s) + N_p(s)N_c(s) \\ &= (s+2)s^2 + s + a \\ &= s^3 + 2s^2 + s + a \end{aligned}$$

Selecting $a = 1$ yields $P_c(s)$ with its zeros at

$$-1.7549, \ -0.1226 \pm j0.7449$$

and

$$E(s) = \frac{s+2}{s^3 + 2s^2 + s + 1}$$

Applying the final value theorem to $E(s)$, we obtain, $\lim_{t\to\infty} e(t) = 0$.

The selection of $a = 1$ results in the desired steady-state behavior of the closed-loop system. However, the above controller may, or may not, guarantee the desired transient performance of the closed-loop system. Thus further tuning of the parameter a may be necessary.

In industrial applications, control engineers usually specify the performance of the controlled system based on the system step response. In such a case $r(t)$ is a step function, which sometimes is also referred to as the constant *set-point*, which can be represented as

$$r(t) = \begin{cases} 0 & \text{for} \quad t < 0 \\ A & \text{for} \quad t \geq 0, \end{cases}$$

where A is the constant set-point value. Recall that the Laplace transform of the above signal is

$$R(s) = \mathcal{L}(r(t)) = \frac{A}{s}.$$

Taking the above into account it is easy to see that the integral (I) controller contains the model of the unit step function as well as any scalar multiple of it. If, in addition, the zeros of the CLCP are all in the open left-half plane, then $e(\infty) = 0$ for any step input. For this reason, the I controller is often referred to as as *reset controller* because it automatically ensures $e(\infty) = 0$ without an operator manually altering or "resetting" the set-point value to achieve a desired output value.

14.2 PID Controller Architecture

We begin our discussion with a basic PID controller architecture, also known as the type A PID controller; it consists of three-terms. Later we discuss two modifications, B and C, of the type A PID control architecture. The three terms of a PID controller attempt to fulfill the transient and steady-state specifications.

To proceed, we need some notation. Let $u(t)$ denote the output of the controller, which is the input signal to the plant. Recall that $e(t)$ is the tracking error, that is, the difference between the desired plant output, $r(t)$, and the actual plant output, $y(t)$. Thus

$$e(t) = r(t) - y(t). \tag{14.3}$$

The three-term type A PID control law can be described as

$$u(t) = u_P(t) + u_I(t) + u_D(t), \tag{14.4}$$

where u_P is the proportional term, u_I is the integral term, and u_D is the derivative term. We now analyze each term one by one.

- The proportional term, u_P, has the form

  $$u_P(t) = K_P e(t), \tag{14.5}$$

  where K_P is the proportional gain.

  The proportional term responds immediately to the current tracking error. Typically, however, it cannot achieve the desired tracking accuracy without excessively large gain.

- The integral term, u_I, is given by

  $$u_I(t) = \frac{K_P}{T_I}\int_0^t e(\tau)d\tau, \tag{14.6}$$

where T_I is the integral time constant.

The integral term yields a zero steady-state error in tracking a step function (a constant set-point). This term is slow in response to the current tracking error.

- The derivative term, u_D, is

$$u_D(t) = K_P T_D \frac{de(t)}{dt}, \tag{14.7}$$

where T_D is the derivative time constant.

The derivative term is especially effective for plants with significant dead-time resulting in a poor representation of previous control actions in the current tracking error. This, in turn, as pointed out by Knospe [183], may lead to large transient error when PI control is used. The derivative action combats this problem by generating its action based on a prediction of future tracking error. But there is a cost to be paid—the derivative term amplifies higher frequencies sensor noise. To alleviate the problem, a filtering of the differentiated signal is employed thus resulting in a PID controller with a derivative filter referred to as a PIDF controller. We present a PIDF controller in a section on implementing PID controllers.

Combining the above terms together, we obtain the type A PID control action, in the time domain,

$$\begin{aligned} u(t) &= u_P(t) + u_I(t) + u_D(t) \\ &= K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right). \end{aligned} \tag{14.8}$$

It is advantageous to work in the Laplace domain when analyzing the PID-controlled plant. The Laplace transfer function of the type A PID controller, denoted $G_{PID}(s)$, is

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_P \left(1 + \frac{1}{T_I s} + T_D s \right), \tag{14.9}$$

where $U(s) = \mathcal{L}(u(t))$ is the Laplace transform of $u(t)$ and $E(s) = \mathcal{L}(e(t))$ is the Laplace transform of the tracking error $e(t)$. In Figure 14.4, we show a PID-controlled system with the type A PID controller in the forward path.

14.3 PID Controller Tuning

The control engineer's task is to select the PID controller's parameters to meet given performance specifications. This process is referred to as the controller tuning. The most widely

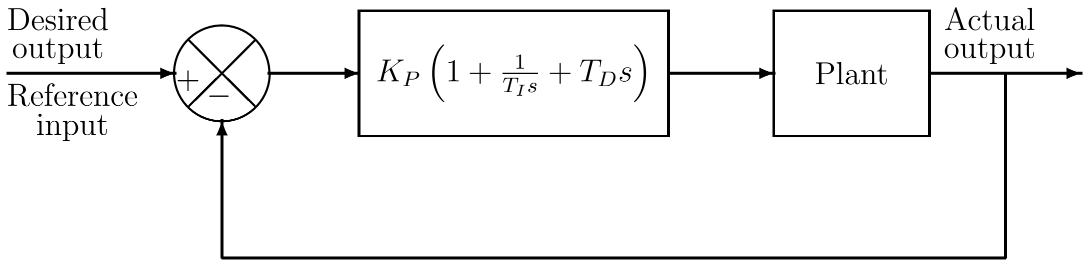


Figure 14.4: A PID-controlled system.

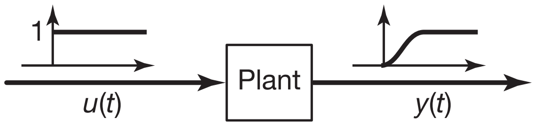


Figure 14.5: Open-loop test on the plant—measuring the step response of the plant.

known PID tuning rules were proposed by Ziegler and Nichols [374] in 1942. They proposed two sets of tuning rules of the type A PID controllers. In both cases the controller's parameters are selected based on the experimental measurements of the controlled process.

14.3.1 The Open-Loop Method—Tuning Based on Open-Loop Tests on the Plant

The first method of choosing control settings for the PID controller, proposed by Ziegler and Nichols, is based on open-loop tests on the plant as shown in Figure 14.5. There is a restriction on the applicability of this method. The method can only be applied if the step response is S-shaped as shown in Figure 14.6. The S-shaped step response can be characterized using three parameters: the delay time (dead-time) L, the time constant T, and the steady-state gain (process gain) K.

We add that a plant with the S-shaped step response can be approximately modeled by a first-order transfer function with transportation delay,

$$\frac{Y(s)}{U(s)} = G_d(s) = \frac{Ke^{-Ls}}{Ts+1}. \tag{14.10}$$

We illustrate the procedure of approximating a system with an S-shaped step response using

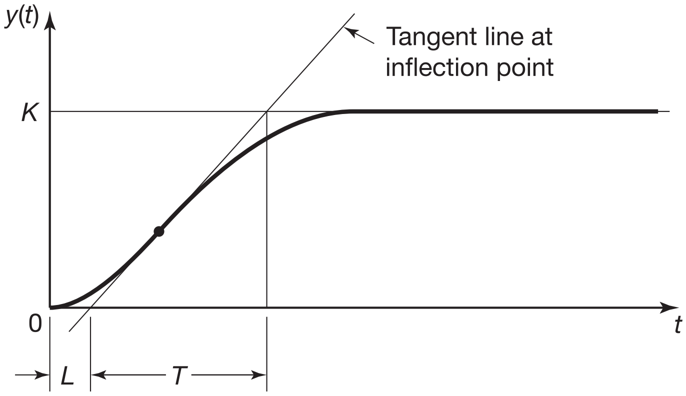


Figure 14.6: An S-shaped step response and its characterization using three parameters.

a first-order transfer function with transportation delay given by (14.10) with the following example.

Example 14.3 We consider, as in Ogata [255, p. 444], a plant modeled by the second-order transfer function

$$G_p(s) = \frac{10}{(s+1)(s+5)} = \frac{10}{s^2+6s+5}. \tag{14.11}$$

Using the procedure illustrated in Figure 14.6, we approximate the second-order transfer function (14.11) using the first-order transfer function with transportation delay,

$$G_d(s) = \frac{Ke^{-Ls}}{Ts+1} = \frac{2e^{-0.053s}}{0.798s+1}. \tag{14.12}$$

In Figure 14.7, we compare the step responses of the above two transfer functions. One can see that approximating the S-shaped step step response of a dynamical system using a first-order transfer function with transportation delay is a challenging problem. Optimization methods, such as genetic algorithms and particle swarm optimizers are the tools that can be used here effectively to determine parameters L and T that result in $G_d(s)$ that "best" approximates a given $G_p(s)$.

Ziegler and Nichols proposed a set of tuning rules, formulas, using the parameters that characterize the step response of the plant modeled by the transfer function (14.10). These tuning formulas are shown in Table 14.1.

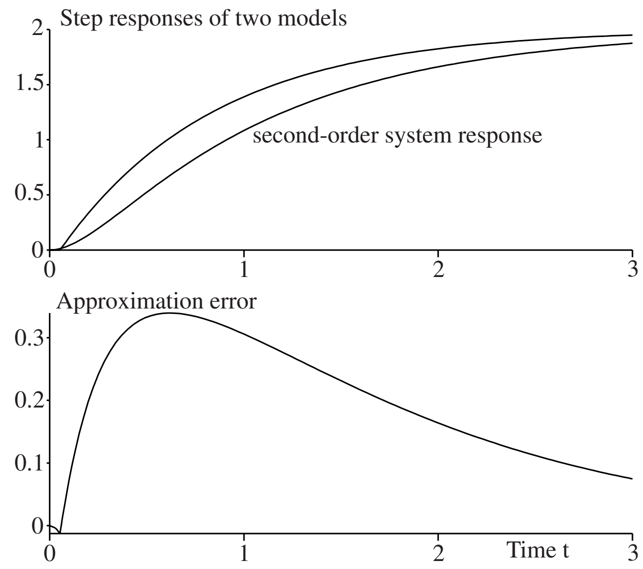


Figure 14.7: Comparison of step responses of the second-order system and its approximation using a first-order system with transportation delay in Example 14.3.

Table 14.1: Ziegler-Nichols tuning formulas based on the step response of the plant (open-loop method).

Controller	K_P	T_I	T_D
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$

Substituting the tuning parameters given in Table 14.1 into the transfer function of the PID controller gives

$$\begin{aligned} G_{PID}(s) &= K_P\left(1 + \frac{1}{T_I}s + T_D s\right) \\ &= 1.2\frac{T}{L}\left(1 + \frac{1}{2Ls} + 0.5Ls\right) \\ &= 0.6T\frac{\left(s + \frac{1}{L}\right)^2}{s}. \end{aligned} \tag{14.13}$$

Thus the type A PID controllers tuned using the formulas of the open-loop method can be parameterized by two parameters, T and L.

The ACT controller manufacturer, as well as Li, Ang, and Chong [211], present slightly

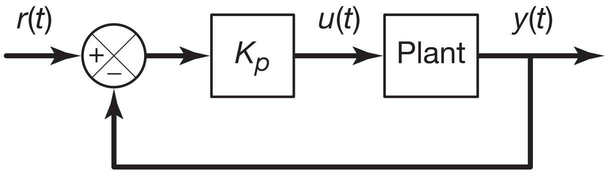


Figure 14.8: Experimental setup to determine the critical gain, K_{cr} and critical period of oscillation, P_{cr}.

modified Ziegler-Nichols open-loop tuning formulas shown in Table 14.2. The only difference between the formulas shown in Table 14.1 and those in Table 14.2 is in the first column where all the column elements of the first column of Table 14.2 are obtained by dividing by K the corresponding elements of Table 14.1.

Table 14.2: Modified Ziegler-Nichols tuning formulas based on the step response of the plant (open-loop method).

Controller	K_P	T_I	T_D
P	$\frac{T}{LK}$	∞	0
PI	$0.9\frac{T}{LK}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{LK}$	$2L$	$0.5L$

14.3.2 The Closed-Loop Method—Tuning Based on Closed-Loop Tests on the Plant

The second method of Ziegler and Nichols of tuning PID controllers is also known as the the closed-loop method. This method requires an experiment on the closed-loop system as shown in Figure 14.8.

In this experiment, we have a pure P-controller in the *feed-forward path*. A step is applied as the input and the response is observed. The test is repeated with increased or decreased controller gain until a sustained oscillation is achieved, as shown in Figure 14.9. This gain is called the "critical gain" and denoted K_{cr}. The period of the sustained oscillation is called the critical period and denoted P_{cr}. Using the parameters K_{cr} and P_{cr}, Ziegler and Nichols proposed tuning formulas depicted in Table 14.3. Substituting the tuning parameters from

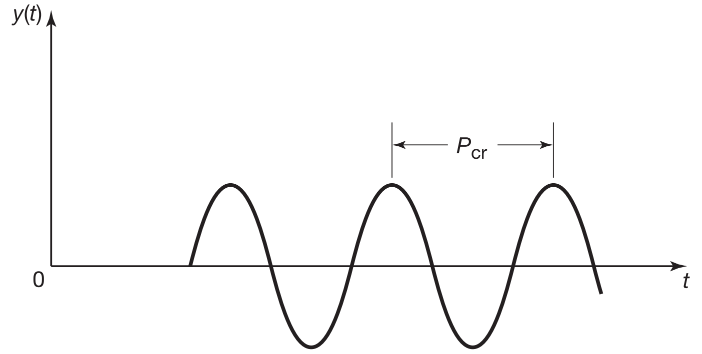


Figure 14.9: Measuring the period of the sustained oscillation, P_{cr}.

Table 14.3: Ziegler-Nichols tuning formulas based on critical gain K_{cr} and critical period P_{cr} (closed-loop method).

Controller	K_P	T_I	T_D
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{P_{cr}}{1.2}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Table 14.3 into the transfer function of the type A PID controller, we obtain

$$\begin{aligned} G_{PID}(s) &= K_P\left(1+\frac{1}{T_I}s+T_Ds\right) \\ &= 0.6K_{cr}\left(1+\frac{1}{0.5P_{cr}s}+0.125P_{cr}s\right) \\ &= 0.075K_{cr}P_{cr}\frac{\left(s+\frac{4}{P_{cr}}\right)^2}{s}. \end{aligned} \tag{14.14}$$

As in the open-loop method, the type A PID controllers tuned using the formulas of the closed-loop method can be parameterized by two parameters; this time by K_{cr} and P_{cr}.

In summary, both the open-loop and closed-loop Ziegler-Nichols tuning rules allow us to parameterize the type A PID controllers using just two parameters. That is, we can represent a family of type A PID controllers as

$$G_{A-PID} = K_c\frac{(s+a)^2}{s}, \tag{14.15}$$

where K_c and a are design parameters. The above representation can be used to computationally obtain parameters' values that satisfy design specifications—see Exercise 14.2.

14.4 Implementing PID Controllers

If the reference input is a step, then because of the presence of the derivative term, the controller output will involve an impulse function. In addition, the derivative term amplifies high frequency sensor noise. To alleviate the above problems, the pure derivative term is replaced with a derivative filter, which results in the PIDF controller.

14.4.1 PIDF Controller

In the PIDF controller, the pure differentiator is approximated with the pure differentiator cascaded with a first-order low-pass filter. Thus the pure derivative term is replaced by the approximate differentiator and so the derivative term contribution would take the form,

$$U_D(s) = \frac{K_P T_D s}{\gamma T_D s + 1} E(s), \tag{14.16}$$

where γ is a small parameter, for example, $\gamma = 0.1$. The transfer function of the type A PIDF controller has the form

$$G_{PIDF} = K_P \left(1 + \frac{1}{T_I} s + \frac{T_D s}{\gamma T_D s + 1} \right). \tag{14.17}$$

For a step reference, the PIDF controller output will involve a sharp pulse rather than an impulse function. This is referred to as the *set-point kick phenomenon*.

Example 14.4 In this example, we use the open-loop Ziegler-Nichols tuning formulas given in Table 14.1 to tune the PIDF controller for plant models in Example 14.3. We first apply the PIDF controller to the second-order transfer function given by (14.11). We add that the proportional gain was tuned as

$$K_P = 1.2 \frac{T}{L} = 18.068$$

because for the modified gain, $K_P = 1.2 \frac{T}{LK}$, the closed-loop system becomes unstable. The remaining parameters are:

$$T_I = 2L = 0.106, \quad T_D = 0.5L = 0.0265, \quad \text{and} \quad \gamma = 0.1.$$

The plant output and the control signal plots versus time are shown in Figure 14.10. The system output is very oscillatory and takes long time to settle down. We can observe the set-point kick phenomenon in the plot of the control effort versus time.

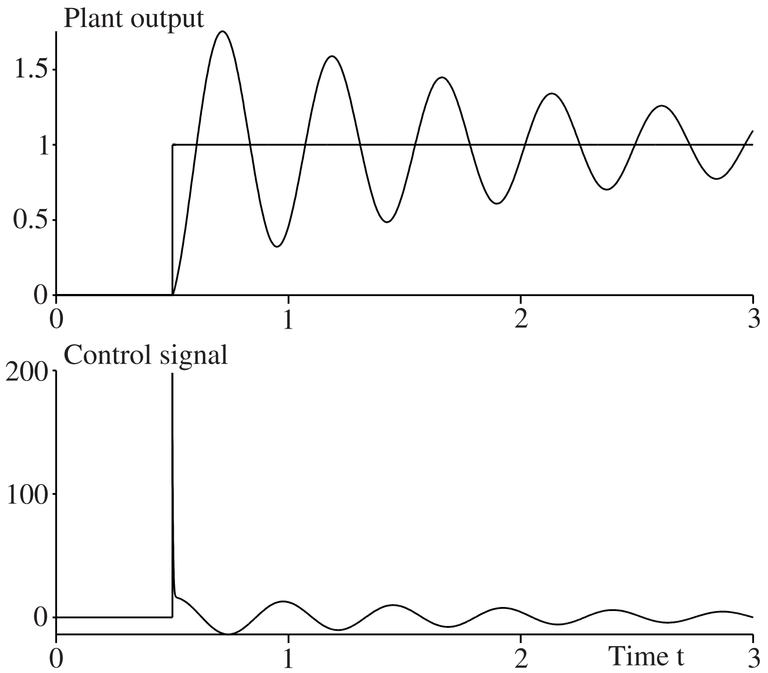


Figure 14.10: PIDF-controlled second-order system's step response of Example 14.4, where the reference input is a step function delayed by 0.5 s in order to better observe the set-point kick phenomenon.

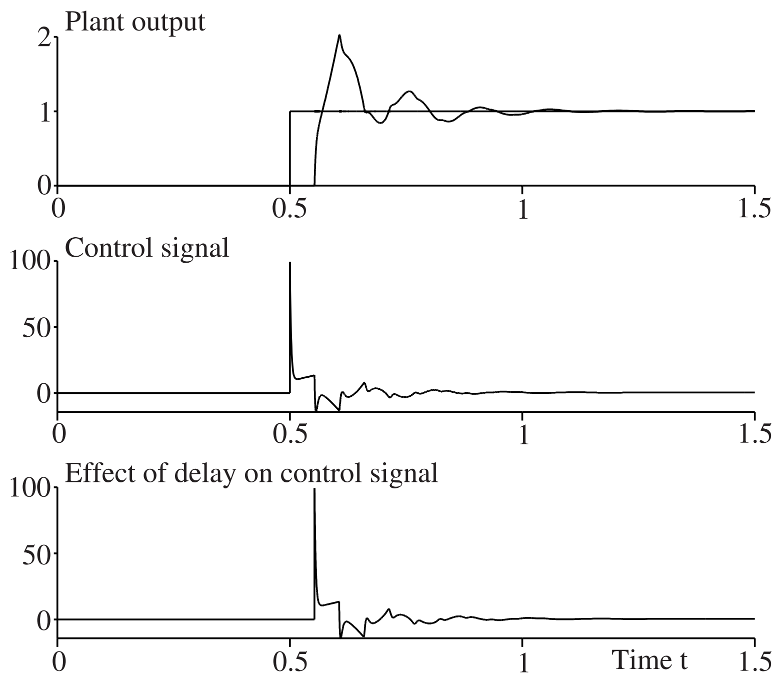


Figure 14.11: Step response of the PIDF-controlled first-order system with transportation delay in Example 14.4.

We then applied the type A PIDF controller to the transfer function given by (14.12). Here, on the other hand, we used $K_P = 1.2\frac{T}{LK}$ because for $K_P = 1.2\frac{T}{L}$ the closed-loop system becomes unstable. The plant output and the control signal plots versus time are shown in Figure 14.11. Here, as in the previous case, we can observe a set-point kick phenomenon. The closed-loop system is much faster here but the overshoot is at unacceptable level.

We conclude that the open-loop Ziegler-Nichols tuning rules do not necessarily yield excellent performance of the PID-controlled system. The controller requires careful additional tuning. To illustrate this point, we doubled the values of K_P, T_I, and T_D. The re-tuned PIDF controller performance applied to the second-order plant is shown in Figure 14.12. This re-tuning was suggested by Ogata [256, p. 165]. However the re-tuned controller does not work for the first-order plant with transportation delay. The closed-loop system driven by this controller is in fact unstable.

As we observed before, in addition to the transient performance, the designer has to deal with the set-point kick phenomenon. We address this issue next.

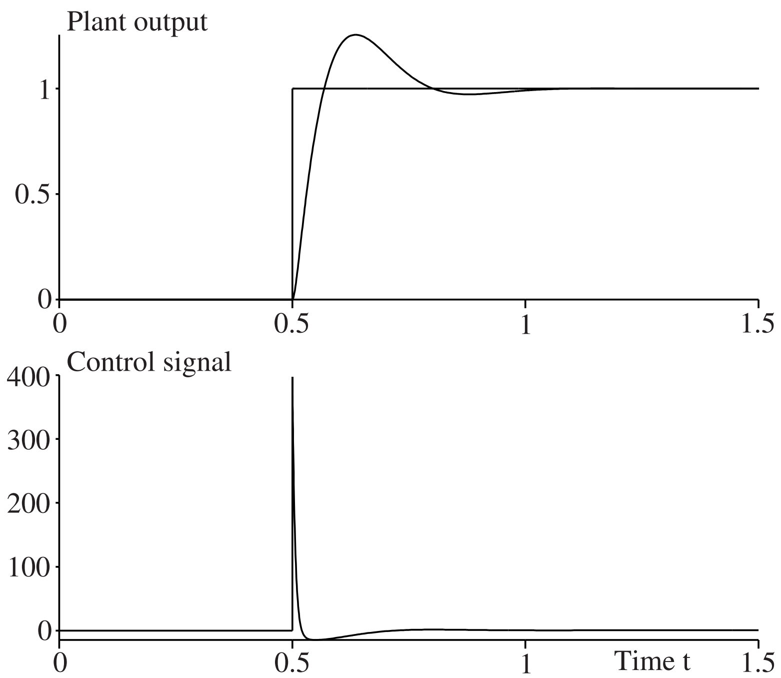


Figure 14.12: Step response of the closed-loop system with re-tuned PIDF controller and second-order plant in Example 14.4.

14.4.2 Integrator Windup

When implementing a PID controller, we need to take into account physical constraints of the controlled plant. In particular, a control engineer has to remember that all actuators have physical limitations; a motor has a finite limited speed while a valve can only be opened to its limits. Thus a control engineer is faced with the problem of actuator saturation. To illustrate the issue, consider Example 14.4. Suppose that the actuator saturation limits are ± 10, that is,

$$|u(t)| \leq 10.$$

These constraints are violated in the above example because of the set-point kick phenomenon. Let us now impose the constraints on the control action and see how this affects the closed-loop system behavior.

Example 14.5 First, we consider the second-order plant (14.11) simulated in the previous example. We impose constraints on the actuator, $|u(t)| \leq 10$. The PIDF controller parameters are the same as the ones we used in the simulation shown in Figure 14.12. The output of the closed-loop system is shown in Figure 14.13.

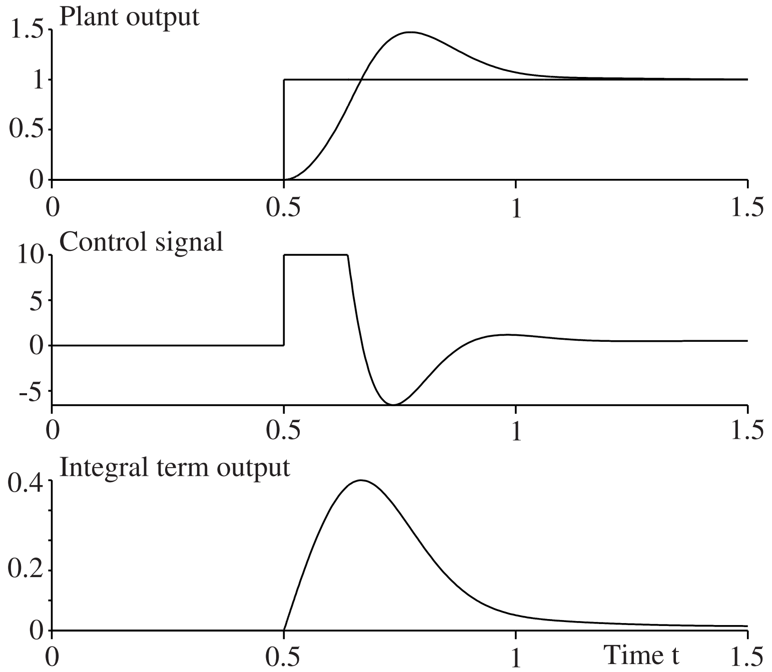


Figure 14.13: A PIDF-controlled second-order plant with with constrained actuator.

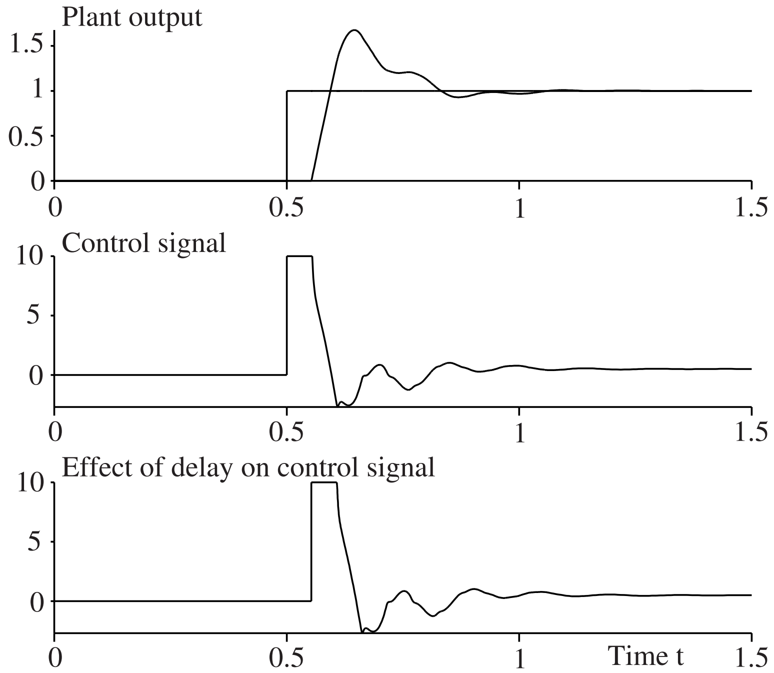


Figure 14.14: Plant output of the PIDF-controlled first-order plant with with transportation delay and constraints on input.

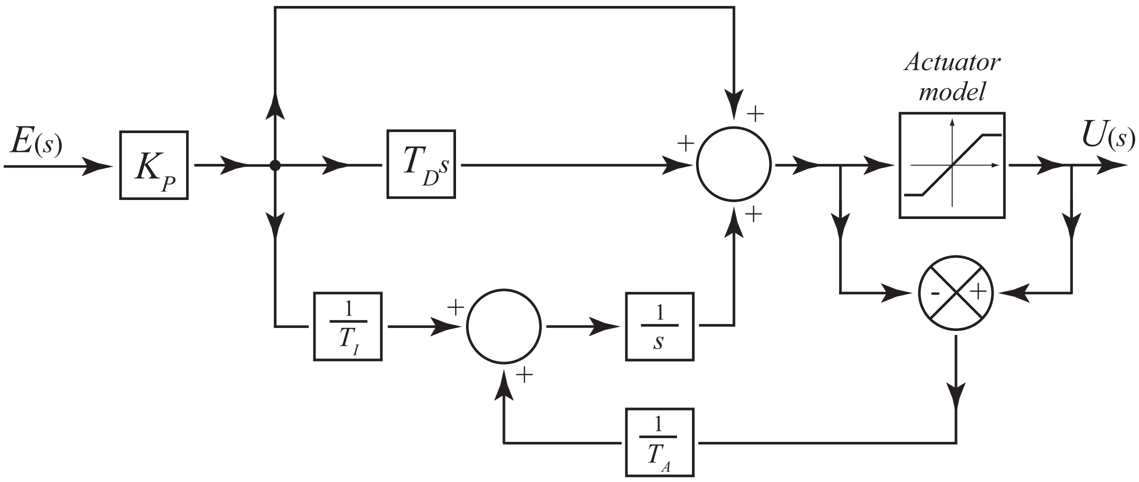


Figure 14.15: A type A PID controller with the integrator anti-windup.

As can be seen in Figure 14.13 and 14.14, in both cases the control signal saturates in the moment the reference input is applied. The input to the plant remains saturated in the sense that the feedback loop is broken because the controller's output is not responding to its input. The output of the integral term continues to increase because the tracking error is positive. The output of the integral term starts decreasing when the plant's output becomes larger than the reference input. However, the controller stays saturated for some time because of the excessively large output of the integral term. As a result of the integral term excessive output, the plant's output has a large overshoot. This phenomenon of excessive integrator term output resulting in the large overshoot of the plant's output is called the *integrator windup*.

We can avoid or reduce the integrator windup by reducing the integral term output when the actuator saturates. A possible *anti-windup* scheme is shown in Figure 14.15. This anti-windup mechanism is referred to as tracking or back calculation. In Figure 14.16, we show the output of the PIDF-controlled second-order plant with the anti-wind, where $T_A = T_I/0.075$. In Figure 14.17, we show the output of the first-order plant with transportation delay controlled with a PIDF controller with the integrator windup. The PIDF controller parameters are the same as in the simulation of Figure 14.11, and $T_A = 0.21$.

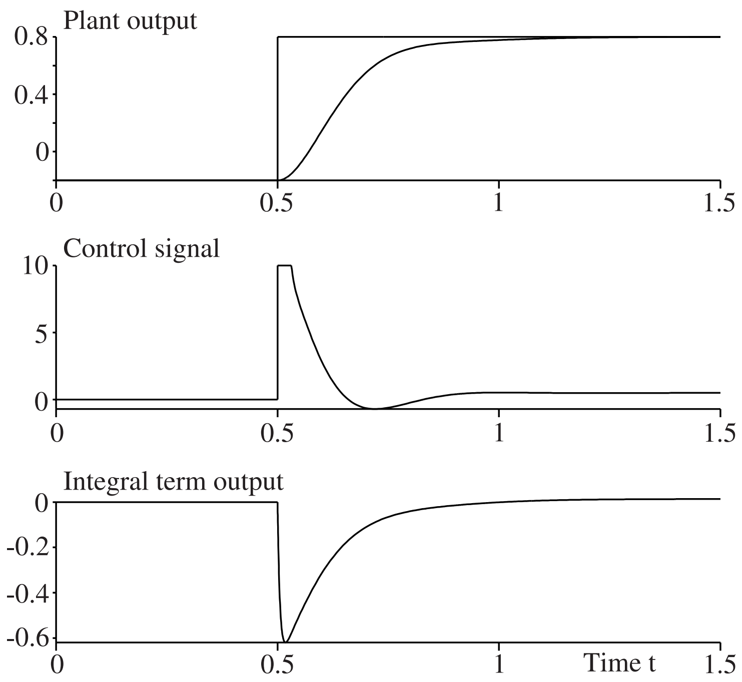


Figure 14.16: Plant output of the PIDF-controlled second-order plant with the integrator ani-windup circuit.

14.5 Modified PID Control Schemes

We now discuss two alternative schemes to eliminate the set-point kick phenomenon. We first take another look at the type A PID control architecture as shown in Figure 14.18. In order to avoid differentiating a step reference input, it is proposed to operate the derivative action in the feedback as shown in Figure 14.19; hence its name, the PI-D controller also referred to as the type B PID controller. A practical implementation of this controller that we refer to as the type B PIDF controller action can be described as

$$U(s) = K_P\left(1+\frac{1}{T_I s}\right)E(s) - \frac{K_P T_D s}{\gamma T_D s + 1}Y(s).$$

We can go even further and move the proportional and derivative actions into feedback giving us the I-PD control or the type C PID controller shown in Figure 14.20. A practical implementation of this controller that we refer to as the type C PIDF controller action can be described as

$$U(s) = \frac{K_P}{T_I s}E(s) - K_P\left(1+\frac{T_D s}{\gamma T_D s + 1}\right)Y(s).$$

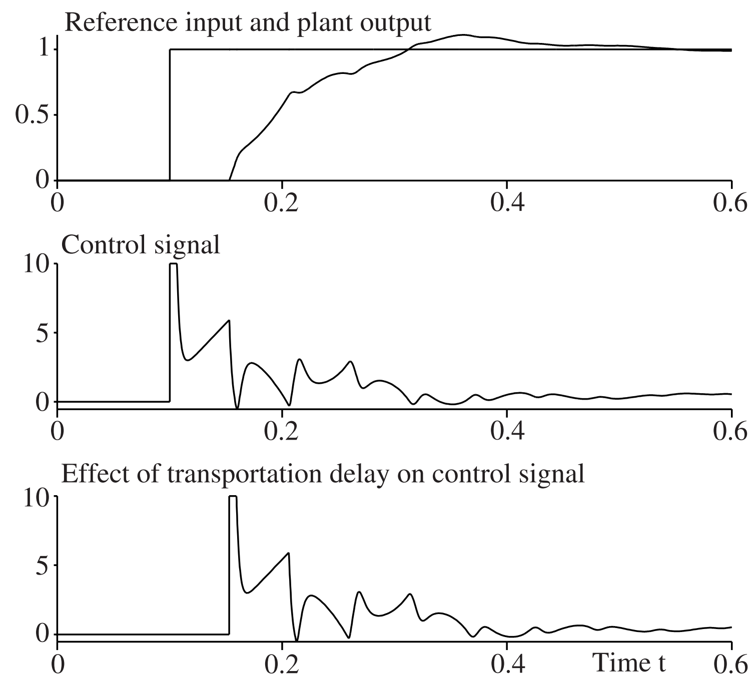


Figure 14.17: Plant output of the PIDF-controlled first-order plant with transportation delay and the integrator ani-windup.

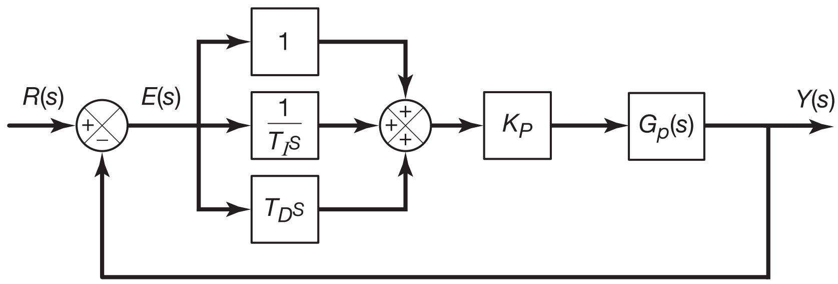


Figure 14.18: More detailed view of the type A PID controller.

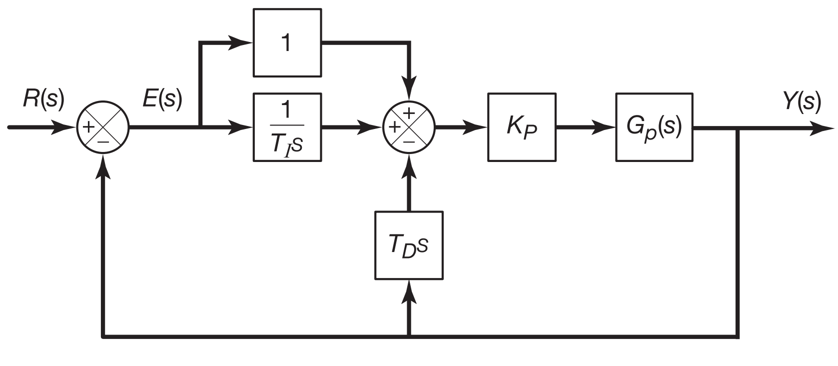


Figure 14.19: Type B PID controller.

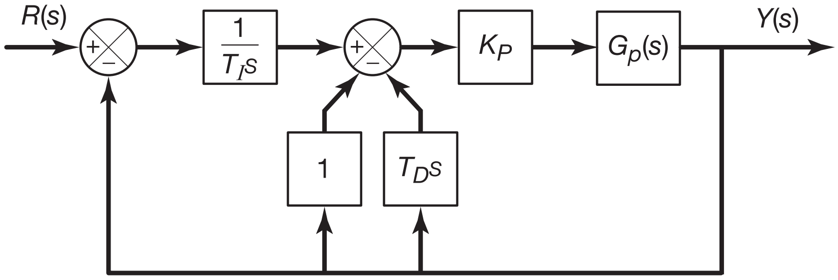


Figure 14.20: Type C PID controller.

14.6 Digital Implementation

In this section we present a method of implementing PID controllers using a computer. To obtain a computer implementation, we need to discretize the continuous-time differential equation describing a controller at hand.

We discuss discretization of different terms of the type A PIDF controller. The types B and C PIDF controllers can be easily obtained by modifying the A PIDF architecture. In our discussion, we assume that the sampling instants are equally spaced. We denote the sampling period as h, and sampling instants as t_k, where $k = 0, 1, 2, \ldots$.

- The discrete-time version of the proportional term, $u_P(t_k)$, has the form

$$u_P(t_k) = K_P e(t_k), \tag{14.18}$$

  where K_P is the proportional gain.

- Recall that the continuous-time integral term, $u_I(t)$, is given by

$$u_I(t) = \frac{K_P}{T_I} \int_0^t e(\tau) d\tau,$$

  Differentiating both sides of the above integral equation yields

$$\frac{du_I(t)}{dt} = \frac{K_P}{T_I} e(t).$$

  We can use several methods to discretize the above differential equation. Using the forward difference discretization gives

$$u_I(t_{k+1}) = u_I(t_k) + \frac{K_P h}{T_I} e(t_k). \tag{14.19}$$

  For other discretization methods, the reader may wish to consult Section 3.7 on numerical techniques.

- Performing cross-multiplication in (14.16) and representing the resulting polynomial equation, in the Laplace variable s, into the time domain gives the continuous-time differential equation describing the filtered derivative term,

$$\gamma T_D \frac{du_D(t)}{dt} + u_D(t) = K_P T_D \frac{de(t)}{dt}. \tag{14.20}$$

The forward difference discretization of the above differential equation yields

$$u_D(t_{k+1}) = \left(1 - \frac{h}{\gamma T_D}\right) u_D(t_k) + \frac{K_P}{\gamma} \left(e(t_{k+1}) - e(t_k)\right). \tag{14.21}$$

On the other hand, the backward difference discretization of the differential equation (14.20) gives,

$$u_D(t_k) = \frac{\gamma T_D}{\gamma T_D + h} u_D(t_{k-1}) + \frac{T_D K_P}{\gamma T_D + h} \left(e(t_k) - e(t_{k-1})\right). \tag{14.22}$$

Combining (14.18), (14.19), and (14.22), we obtain a discrete-time approximation of the type A PIDF controller,

$$\left.\begin{array}{rcl} u_P(t_k) &=& K_P e(t_k) \\ u_D(t_k) &=& \frac{\gamma T_D}{\gamma T_D + h} u_D(t_{k-1}) + \frac{T_D K_P}{\gamma T_D + h} \left(e(t_k) - e(t_{k-1})\right) \\ u(t_k) &=& u_P(t_k) + u_I(t_k) + u_D(t_k) \\ u_I(t_{k+1}) &=& u_I(t_k) + \frac{K_P h}{T_I} e(t_k) \end{array}\right\} \tag{14.23}$$

To implement the above controller, one can use the zero-order hold—see Exercises 14.5, 14.6, and 14.7.

14.7 Notes

The first PID controllers for industrial applications were introduced in 1939 by the Taylor Instrument Company and the Foxboro Instrument Company. For an account of the PID controller development along with an overview of early process control devises, we recommend Bennett [30]. For an account of recent advances in PID control, see an article by Li, Ang, and Chong [211]. See also Hägglund and Åström for a discussion of automatic tuning of PID controllers [126].

Many engineers are familiar with PID controllers and prefer to enhance the performance of a well-known solution through additional capabilities rather than switch to an untested solution. Knospe [183] observers that even alternative digital control methods have not supplanted PID's primacy in industrial control with a possible exception of model predictive control, where the control action is calculated on the fly. Knospe [183] adds that PID control still has a role even in this case by handling lower level loops.

Welander states in [348] that about 90% of loops that employ PID control are probably PI control. This is because derivative action is the least understood. "*Derivative* acts as a brake or dampener on the control effort. The more the controller tries to change the value, the more it counteracts the effort." Welander concludes his analysis of the D part in PID architecture as follows: "Ultimately, the use of derivative action can be beneficial, but it has to be used in the right situations and in the right amounts. Your starting assumption should be to see if the process will run satisfactorily on PI control alone." He continues to say that, adding derivative should be done cautiously and with appropriate filtration. "Like cooking with hot sauce, in the right context and in skilled hands, the result can be remarkable. But if it's used inappropriately, it's a recipe for disaster" [348].

14.8 Exercises

Exercise 14.1 Propose a simple controller for the system in Example 14.1 that yields $e(\infty) = 0$ when the reference signal is

$$r(t) = \frac{1}{2}t^2 1(t),$$

where $1(t)$ is the unit step function. Recall that

$$R(s) = \mathcal{L}(r(t)) = \frac{1}{s^3}.$$

Exercise 14.2 (Based on Ogata [257, p. 692]) For the PID-controlled system shown in Figure 14.21, write a MATLAB program that finds a combination of the parameters K_c and a such that the maximum overshoot of the system output $y(t)$ is in the range $(5\%, 10\%)$. Assume the following search region for K_c and a,

$$5 \le K_c \le 20 \quad \text{and} \quad 5 \le a \le 10.$$

Plot the unit-step response for the first found set of the parameters that satisfy the overshoot specifications.

Exercise 14.3 Compare the performance of the type B PID controller on the plants of Example 14.3 with constraints on the plant input of the form, $|u(t)| \le 10$. Plot $u(t)$ versus time.

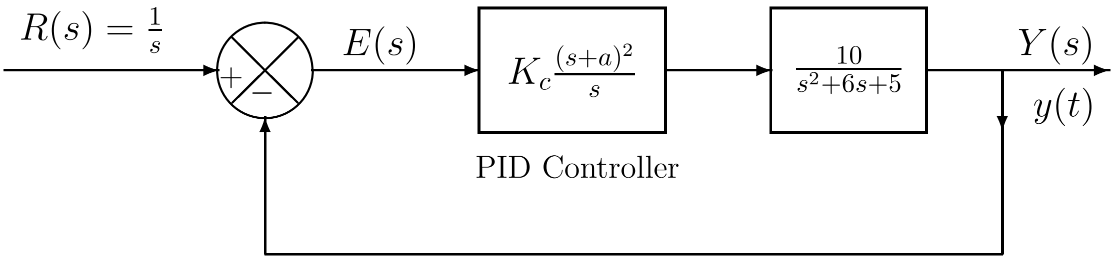


Figure 14.21: A PID controlled system of Exercise 14.2.

Exercise 14.4 Compare the performance of the type C PID controller on the plants of Example 14.3 with constraints on the plant input of the form, $|u(t)| \leq 10$. Plot $u(t)$ versus time.

Exercise 14.5 Compare the performance of the digital PI controller, $u(t_k) = u_P(t_k)+u_I(t_k)$ given by (14.23), on the plants of Example 14.3 with constraints on the plant input of the form, $|u(t)| \leq 10$. Use the zero-order hold. Analyze the controller performance with and without the integrator anti-windup. Plot $u(t)$ versus time.

Exercise 14.6 Compare the performance of the digital type B PIDF controller on the plants of Example 14.3 with constraints on the plant input of the form, $|u(t)| \leq 10$. Equations describing this controller can be obtained by modifying the derivative term of the architecture (14.23). Use the zero-order hold in the controller implementation. Plot $u(t)$ versus time.

Exercise 14.7 Compare the performance of the digital type C PIDF controller on the plants of Example 14.3 with constraints on the plant input of the form, $|u(t)| \leq 10$. Equations describing this controller can be obtained by modifying the proportional and derivative terms of the architecture (14.23). Use the zero-order hold in the controller implementation. Plot $u(t)$ versus time.

Exercise 14.8 For the one-link manipulator from Exercise 5.22, use the linearized model from Part **(b)** of that exercise to design

(a) digital, type A PIDF controller. Implement and perform animation of the closed-loop system comprised of the PID controller driving the nonlinear model of the robot;

(b) assume that the actuator saturation limits are ± 10. Design, implement, and perform animation of the digital, type A PIDF controller with the integrator anti-windup on the nonlinear model of the robot.

Exercise 14.9 For the nonlinear model of a double-link robot manipulator in Subsection 2.4.6, use the linearized model of this robot obtained in Exercise 4.5 to design

(a) digital, type A PIDF controller. Implement and perform animation of the closed-loop system comprised of the PID controller driving the nonlinear model of the robot;

(b) assume that the actuator saturation limits are ± 10. Design, implement, and perform animation of the digital, type A PIDF controller with the integrator anti-windup on the nonlinear model of the robot.

Chapter 15

Fuzzy Systems

> It is in the response to mistakes, of technique or of judgment, that one sees most clearly the human side of science's practitioners: a mix of pride, stubbornness, humility, honesty, dismay, and regret that one could expect to encounter in any section of the population. There are the inevitable diehards, whose pride forces them to concoct ever-more-elaborate explanations for the facts in order to avoid confession of error. There are the crusaders, determined at all costs to expose the foolishness. And there are the "told-you-so's," generally members of the scientific elite, who make little contribution while the debate is raging but are ready to heap scorn on the perpetrators afterward for their "obvious" mistakes. All this, of course, is the antithesis of objective science, but follows as inevitably as night follows day from the fact that science is practiced by people and not by automatons. [20, page 291]

15.1 Motivation and Basic Definitions

In Chapter 14, we present a controller design approach when a mathematical model of the process to be controlled is not available to the control engineer. In this Chapter, we discuss an alternative approach to the controller design when a mathematical model of the controlled system is not available to us. *Fuzzy logic control* offers a way of dealing with modeling problems by implementing linguistic, non-formally expressed control laws derived from expert knowledge. This approach is especially successful in the case of processes such as washing machines, camera focusing, or drying processes that are being controlled well by

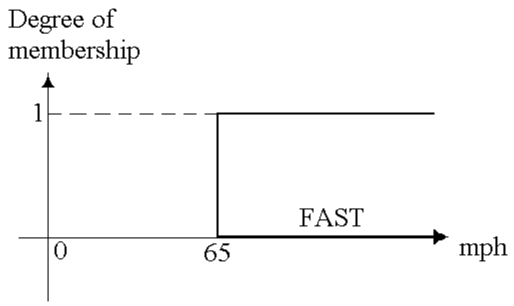


Figure 15.1: An example of a crisp set—fast cars.

human operators. The control of such processes can be captured in the form of rules

IF (process state) THEN (control action).

Thus, by a rule we will understand an IF-THEN statement that defines the set of facts that must be true (the IF part) before a set of actions (the THEN part) can be executed. The basic notion of fuzzy logic is *fuzzy set*. An ordinary set, or simply a set, is a collection of objects. Ordinary sets have two membership states: inclusion or exclusion. An object is either in the set or it is not. Thus the degree to which an object is a member of a set is either 0 or 1. For example, a car is fast if it is driven faster than 65 mph, and slow if it is driven slower than 65 mph—see Figure 15.1. This is the reason we sometimes refer to ordinary sets as *crisp* sets. Fuzzy sets, on the other hand, allow for partial membership in a set. The degree to which an object is a member of a fuzzy set can vary between 0 and 1. With a fuzzy set, we have a gradual transition from membership to nonmembership. For example, a car moving with velocity of 65 mph is a member of the fuzzy set FAST to degree 0.5. A car being driven 57.5 mph is FAST to degree 0.25—see Figure 15.2.

We next illustrate the crisp and fuzzy versions of the sets SLOW and FAST. Observe the sudden change of membership in the crisp SLOW and FAST sets at a speed of 65 mph in Figure 15.3. The fuzzy SLOW and FAST sets allow for a gradual change from SLOW to FAST for a car moving between 50 and 80 mph as it is illustrated in Figure 15.4. Notice that a car moving at a speed between 50 and 80 mph has partial membership in SLOW and

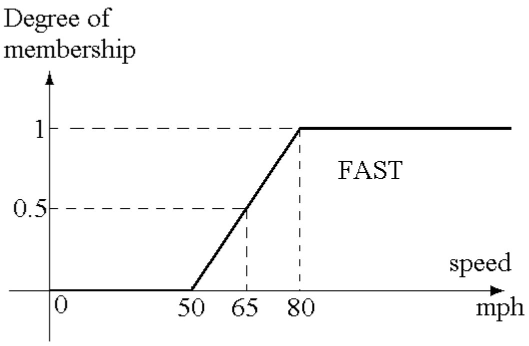


Figure 15.2: A fuzzy set FAST.

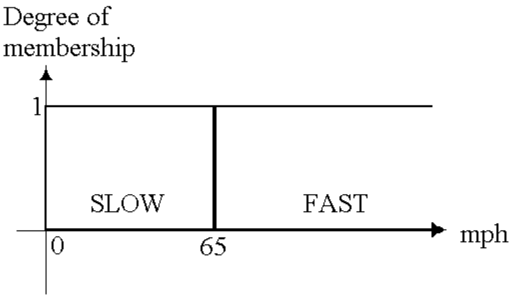


Figure 15.3: Crisp sets for SLOW and FAST.

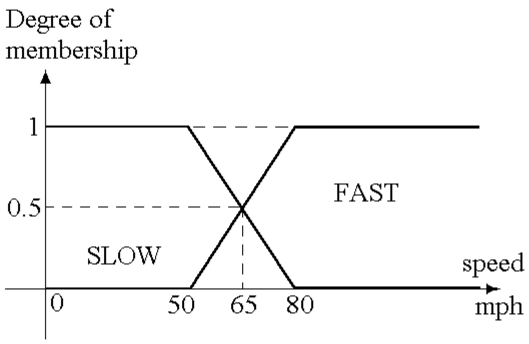


Figure 15.4: Fuzzy sets for SLOW and FAST.

partial membership in FAST at the same time. In the above example, the variable SPEED can have two possible linguistic values: SLOW and FAST. For this reason, we refer to the variable SPEED as the *linguistic variable* because its possible values are the linguistic values: in this example they are SLOW and FAST.

With the above examples in mind, we formally define a fuzzy set. For this, we need to introduce some notation. Let U be a set. We use $u \in U$ to denote that u is an element of U. We call U the *universe of discourse*, or just the *universe*. We first recall the definition of a function. A function A from a set X to a set Y is a set of ordered pairs in $X \times Y$ with the property that if (x, y) and (x, z) are elements of A, then $y = z$.

Definition 15.1 (Fuzzy set) *A fuzzy set A on U is a function that maps U into the interval $[0, 1]$. Equivalently,*

$$A = \{(u, \mu_A(u)) \,:\, u \in U\},$$

where μ_A is commonly called the membership function of the fuzzy set A.

◇

Note that the concept of a fuzzy set generalizes that of a crisp set in the following way. Let $V \subset U$. Define $\mu_V : U \to [0, 1]$ by $\mu_V(u) = 1$ if $u \in V$ and $\mu_V(u) = 0$ otherwise. Then V can be identified with the fuzzy set associated with μ_V. In other words, the subsets of U are identified with the functions from U to the two point set $\{0, 1\}$.

Example 15.1 Consider the following universe $U = \{4, 5, 6, 7, 8\}$. Let

$$A = \{(4, 0.0), (5, 0.5), (6, 1.0), (7, 0.5), (8, 0.0)\}.$$

Then A is a fuzzy set. One interpretation of A is closeness to 6.

Let A and B be fuzzy sets on U. We define the *union*, *intersection*, and *complement* using the following membership functions:

union: $\mu_{A\cup B}(u) = \max\{\mu_A(u)\, ,\, \mu_B(u)\}$ for all $u \in U$;

intersection: $\mu_{A\cap B}(u) = \min\{\mu_A(u)\, ,\, \mu_B(u)\}$ for all $u \in U$;

complement: $\mu_{\bar{A}}(u) = 1 - \mu_A(u)$ for all $u \in U$.

Example 15.2 Consider another fuzzy set B, "number close to 4", given by

$$B = \{(4, 1.0), (5, 0.8), (6, 0.5), (7, 0.2), (8, 0.0)\}.$$

Then,

$$A \cup B = \{(4, 1.0), (5, 0.8), (6, 1.0), (7, 0.5), (8, 0.0)\}.$$

The intersection of the two fuzzy sets A and B is

$$A \cap B = \{(4, 0.0), (5, 0.5), (6, 0.5), (7, 0.2), (8, 0.0)\}.$$

The complement of the fuzzy set A is

$$\bar{A} = \{(4, 1.0), (5, 0.5), (6, 0.0), (7, 0.5), (8, 1.0)\}.$$

Definition 15.2 (Fuzzy number) *A fuzzy number is a fuzzy set F on a universe $U \subset \mathbb{R}$ such that*

$$\max_{u\in U} \mu_F(u) = 1,$$

and

$$\mu_F(\lambda u_1 + (1-\lambda)u_2) \geq \min(\mu_F(u_1), \mu_F(u_2)),$$

for all u_1, u_2 $\in U$ and all $\lambda \in [0, 1]$.

◇

A fuzzy set that satisfies the conditions $\max_{u\in U} \mu_F(u) = 1$ and $\mu_F(\lambda u_1 + (1-\lambda)u_2) \geq \min(\mu_F(u_1), \mu_F(u_2))$ is called *normal* and *convex*, respectively. Thus, a fuzzy number is a fuzzy set that is both normal and convex. Examples of convex and non-convex fuzzy sets are given in Figure 15.5.

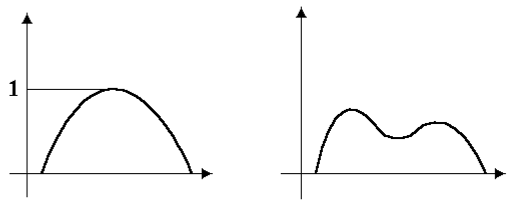


Figure 15.5: The first fuzzy set is a fuzzy number, while the second one is not.

15.2 Standard Additive Model

Let X and Y be nonempty sets and let I and J be nonempty index sets. Suppose we are given collections of fuzzy sets $\{A_\alpha : \alpha \in I\}$ and $\{B_\beta : \beta \in J\}$ on X and Y, respectively, and rules of the form

$$\text{IF } A_\alpha \text{ THEN } B_\beta.$$

The IF-THEN rules along with the collections of fuzzy sets form a function R from $\{A_\alpha : \alpha \in I\}$ to $\{B_\beta : \beta \in J\}$, which we call the *fuzzy system* associated with the rules.

The *support* of a function into $\mathbb{R}$ is the subset of the domain where the function is nonzero. The support of the fuzzy sets A_α and B_β and the IF-THEN rules describe subsets, or patches, in the Cartesian product space $X \times Y$, as illustrated in Figure 15.6. Suppose the support of each A_α and each B_β is a one point set, say $x_\alpha \in X$ and $y_\beta \in Y$, respectively. If $X = \bigcup\{x_\alpha : \alpha \in I\}$, then R can be identified with a function $\tilde{R}$ from X to Y by

$$\tilde{R}(x_\alpha) = y_\alpha$$

if

$$R(A_\alpha) = B_\alpha.$$

One can think of this case as having complete knowledge and the patches are just points.

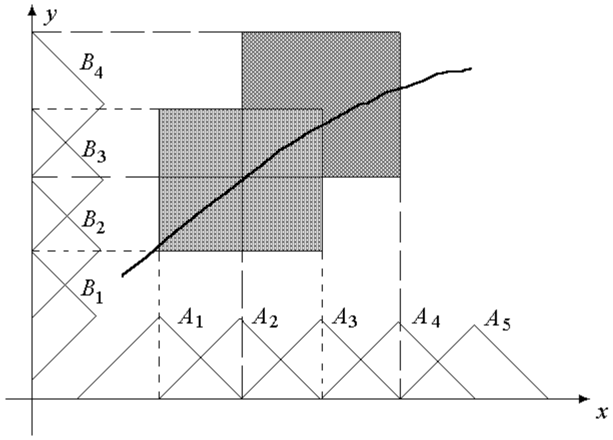


Figure 15.6: Function approximation using IF-THEN rules.

A *fuzzy system* can be thought of as a "fuzzified" function from X to Y with fuzzy sets replacing points in the domain. The size of a patch is usually inversely proportional to the knowledge or confidence of the situation. A point $x \in X$ is usually in the support of more than one A_α with confidence represented by the values of the membership functions μ_α. The fuzzy set $R(A_\alpha)$ represents the desired action corresponding to A_α. This is illustrated in Figure 15.7. Suppose we are given the following two rules:

$$\text{IF } A_3 \text{ THEN } B_3,$$
$$\text{IF } A_4 \text{ THEN } B_2.$$

The point x_0 is contained in the support of A_3 and A_4. Thus, according to the given rules, both B_3 and B_2 need to be taken into account. The process by which we combine these actions is called *defuzzification*. The process of defuzzification is also known as fuzzy reasoning in the literature. The outcome of defuzzification is a numerical value $y \in Y$. An example of a defuzzification algorithm is a centroidal defuzzifier of the form

$$y = \frac{\sum_\alpha \mu_{A_\alpha}(x)\xi(x, A_\alpha, R(A_\alpha))}{\sum_\alpha \mu_{A_\alpha}(x)}, \tag{15.1}$$

where $\xi(x, A_\alpha, R(A_\alpha))$ is a function into the real numbers. We now give an example of ξ.

Let $\xi(x, A_\alpha, R(A_\alpha))$ be the centroid of the support of $R(A_\alpha)$. For example, the centroids of the support of $R(A_3) = B_3$ is c_3 and that of $R(A_4) = B_2$ is c_2. A fuzzy system with the centroidal defuzzifier is called the *standard additive model* (SAM). The term standard additive model was coined by Kosko [187]. A schematic representation of a SAM is shown

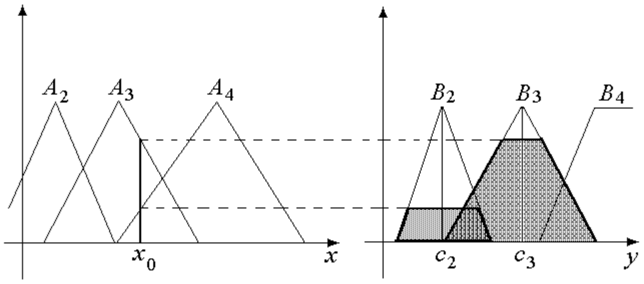


Figure 15.7: Generating output fuzzy sets from THEN-part sets of IF-THEN rules.

in Figure 15.8. In the above description of the SAM, we were concerned with a single-input single-output case. The above construction can be extended to a many-inputs case. Suppose we are given IF-THEN rules of the form

$$\text{IF } A_i \text{ AND } B_i \text{ THEN } C_i,$$

The SAM corresponding to this case is illustrated in Figure 15.9. We could use a defuzzification algorithm of the form

$$z = \frac{\sum_{l=1}^{M} \mu_{A_l}(x)\mu_{B_l}(y)c_l}{\sum_{l=1}^{M} \mu_{A_l}(x)\mu_{B_l}(y)}, \tag{15.2}$$

where M is the number of IF-THEN rules, and c_l is the centroid of the fuzzy set C_l, $l = 1, 2, \ldots, M$. This type of defuzzifier is called the *center average defuzzifier* or the *algebraic product defuzzifier*.

Another type of defuzzifier that we could use is

$$z = \frac{\sum_{l=1}^{M} \min\left\{\mu_{A_l}(x), \mu_{B_l}(y)\right\} c_l}{\sum_{l=1}^{M} \min\left\{\mu_{A_l}(x), \mu_{B_l}(y)\right\}}. \tag{15.3}$$

The above type of defuzzifier is called the *center-of-gravity defuzzifier*.

An illustration of fuzzy reasoning using defuzzification scheme (15.3) is given in Figure 15.10. In general, for q inputs, the center average defuzzifier takes the form

$$z = \frac{\sum_{l=1}^{M} \prod_{j=1}^{q} \mu_{A_j^l}(x_j)c_l}{\sum_{l=1}^{M} \prod_{j=1}^{q} \mu_{A_j^l}(x_j)}. \tag{15.4}$$

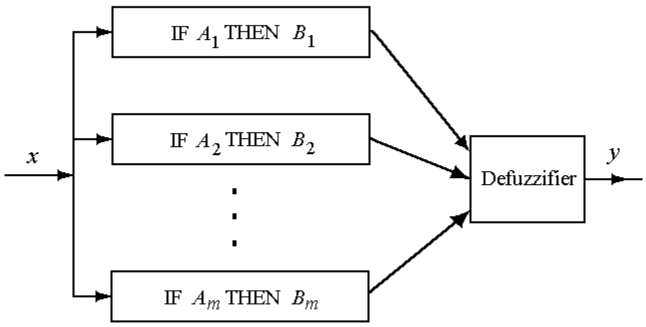


Figure 15.8: Single-input single-output standard additive model schematic representation.

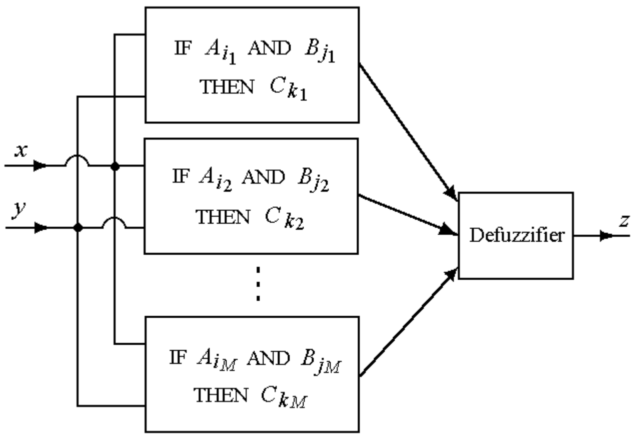


Figure 15.9: Two-input single-output standard additive model representation.

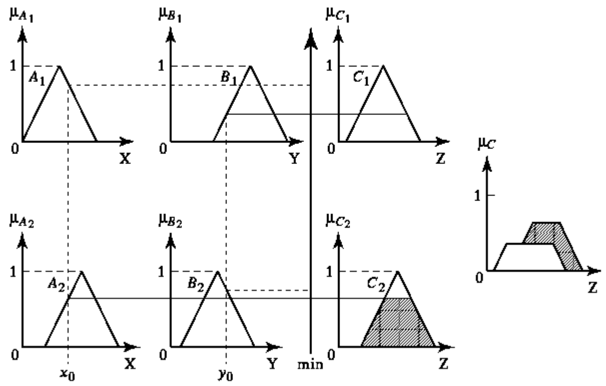


Figure 15.10: An illustration of fuzzy reasoning in a two-input fuzzy system.

15.3 Fuzzy Logic Control

A block diagram of a closed-loop system with a fuzzy logic controller (FLC) is shown in Figure 15.11. The controller uses the control error e and the rate of change in the control error, $\dot{e}$ or Δe, as its inputs. A typical fuzzy logic control architecture has the structure of a SAM. Designing a fuzzy logic controller can be summarized in the following six-step algorithm:

- Identify ranges of the controller inputs;
- Identify ranges of the controller outputs;
- Create fuzzy sets for each input and output;
- Translate the interaction of the inputs and outputs into IF-THEN rules, and then form the rules matrix;
- Decide on the defuzzifier, then apply it to crisp inputs to produce crisp outputs;
- Implement the controller, test it, and modify it if necessary.

As an example, suppose we use five membership functions for the controller's inputs and its output. We then generate the rules. A possible set of rules is shown in Table 15.1, where LN denotes large negative, N means negative, ZE is zero, P is positive, and LP refers to large

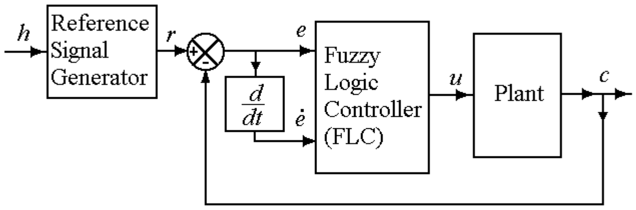


Figure 15.11: A diagram of a closed-loop system with a fuzzy logic controller.

Table 15.1: The rules matrix.

change-in-error / **error**	LN	N	ZE	P	LP
LN	LN	LN	LN	N	ZE
N	LN	LN	N	ZE	P
ZE	N	N	ZE	P	P
P	N	ZE	P	LP	LP
LP	ZE	P	LP	LP	LP

positive. The next step involves choosing a defuzzifier. We use the modified center average defuzzifier to obtain

$$u = \frac{\sum_{k=1}^{25} m_k^o p_k / s_k^o}{\sum_{k=1}^{25} p_k / s_k^o},$$

where for $k = 1, 2, \ldots, 25$, following the rules shown in Table 15.1, we compute

$$p_k = \mu_{i_k}^e \mu_{j_k}^{ce}, \quad i, j = 1, 2, \ldots, 5.$$

The functions μ_i^e, μ_j^{ec} are the membership functions corresponding to the linguistic variables ERROR and CHANGE-IN-ERROR, respectively. The parameters m_k^o and s_k^o are the m and s parameters of the controller output membership functions—see figures below for their definition. The modified center average defuzzifier takes into account the shape of the membership functions, and in particular their "spread". As it is pointed out by Wang [344], the sharper the shape of μ, the stronger is our belief that the value should be closer to its m. This observation is reflected in the formula for the defuzzifier we use, where we divide the weights p_k by s_k^o—the spread of the corresponding controller output membership function. We can use triangular membership functions whose equations are given next. The expression for the function shown in Figure 15.12 is

$$\mu(x, m, s) = \max\left(0, 1 - \frac{|x - m|}{s}\right).$$

The expression for the function shown in Figure 15.13 is

$$\mu(x, m, s) = \min\left(1, \max\left(0, 1 + \frac{x - m}{s}\right)\right).$$

Finally, the expression for the function shown in Figure 15.14 is

$$\mu(x, m, s) = \min\left(1, \max\left(0, 1 - \frac{x - m}{s}\right)\right).$$

Example 15.3 Consider the vehicle model given in Subsection 2.4.2. Let the input signal be $u = \delta_f$—the front wheel steer angle, and $\delta_r = 0$. We design a simple fuzzy logic controller for the above vehicle model. The range for the error is $[-0.5, 0.5]$. The range for the change-in-error is $[-0.06, 0.06]$. The range for the controller output is $[-3\,\text{deg}, 3\,\text{deg}]$. The parameters of the membership functions are given in Table 15.2.

Plots of the error, the change-in-error, and controller's output membership functions are given in Figures 15.15, 15.16, and 15.17, respectively. We created a nonlinear mapping from

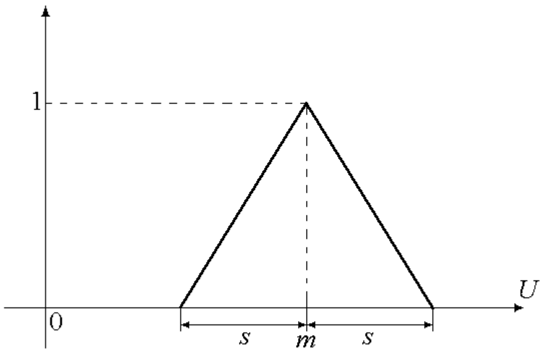


Figure 15.12: A triangular shaped fuzzy set.

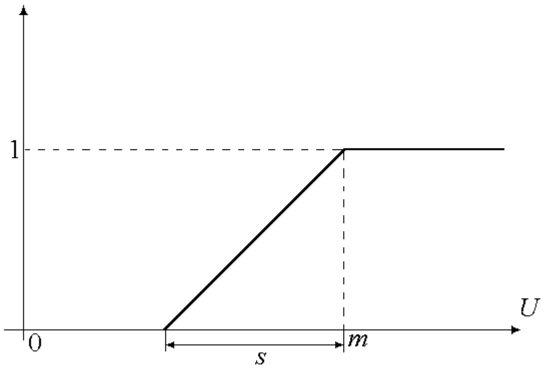


Figure 15.13: A fuzzy set at the right end of a universe of discourse.

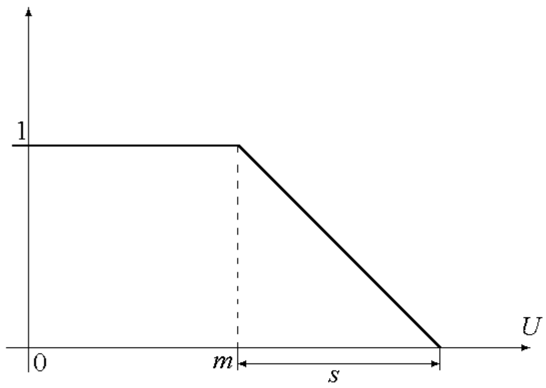


Figure 15.14: A fuzzy set at the left end of a universe of discourse.

Table 15.2: The m and s-parameters of the membership functions.

error		**change-in-error**		**output**	
m	s	m	s	m	s
$m_1 = -0.4$	$s_1 = 0.1$	$m_1 = -0.05$	$s_1 = 0.025$	$m_1 = -2.5$	$s_1 = 1$
$m_2 = -0.2$	$s_2 = 0.2$	$m_2 = -0.025$	$s_2 = 0.025$	$m_2 = -1$	$s_2 = 1$
$m_3 = 0$	$s_3 = 0.2$	$m_3 = 0$	$s_3 = 0.025$	$m_3 = 0$	$s_3 = 1$
$m_4 = 0.2$	$s_4 = 0.2$	$m_4 = 0.025$	$s_4 = 0.025$	$m_4 = 1$	$s_4 = 1$
$m_5 = 0.4$	$s_5 = 0.1$	$m_5 = 0.05$	$s_5 = 0.025$	$m_5 = 2.5$	$s_5 = 1$

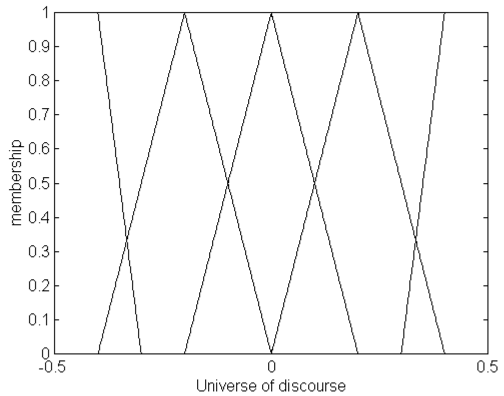


Figure 15.15: Fuzzy numbers of the linguistic variable ERROR in Example 15.3.

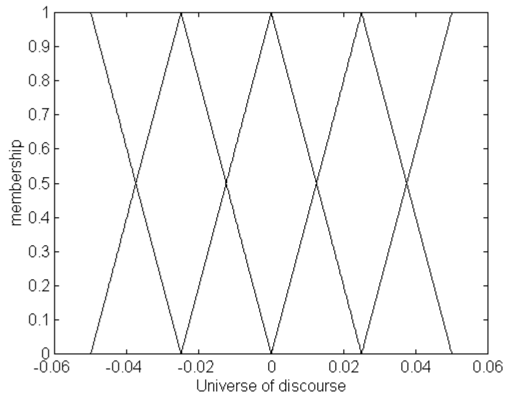


Figure 15.16: Fuzzy numbers of the linguistic variable CHANGE-IN-ERROR in Example 15.3.

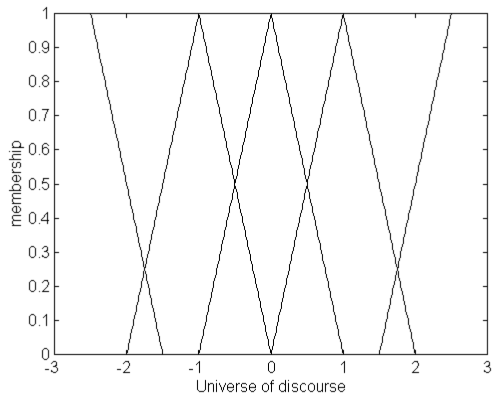


Figure 15.17: Fuzzy numbers of the linguistic variable CONTROL in Example 15.3.

the controller's input space into its output space and formed a look-up table that implements the obtained nonlinear mapping representing fuzzy logic controller. We used SIMULINK's 2-D Look-Up Table block to implement the controller. The look-up table size was 22×22. The output of the look-up table was scaled by the factor $-2\pi/180$ before using it as the vehicle's steering input. The negative sign is to take into account the opposite directions of the lateral axes of the vehicle coordinate system and the fixed system. The factor $\pi/180$ converts degrees into radians. We multiplied the output of the look-up table by 2 to increase the control authority. This allows us to use the same controller to track step inputs with large amplitudes. A plot of the control surface is given in Figure 15.18. Next, a reference signal generator was constructed. We selected the following transfer function of the signal generator:

$$\frac{R(s)}{H(s)} = \frac{1}{s^3 + 1.75s^2 + 2.15s + 1},$$

where $R(s) = \mathcal{L}(r(t))$, $H(s) = \mathcal{L}(61(t))$, and $1(t)$ is the unit step function. The closed-loop system was simulated using SIMULINK. Plots of Y and r versus X are shown in Figure 15.19.

Example 15.4 This example is based on Example 6.1 found in Driankov, Hellendoorn, and Reinfrank [86, pages 266–272]. We consider a nonlinear dynamical system modeled by the

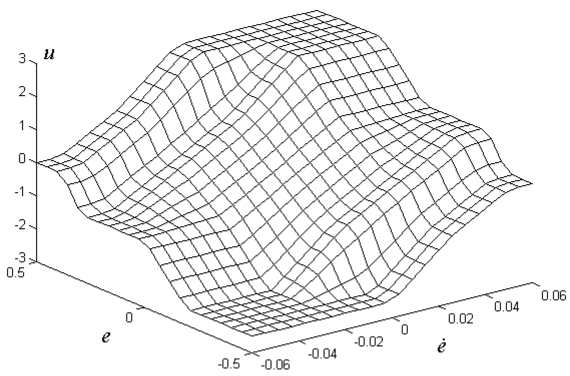


Figure 15.18: A plot of the control surface $u = f(\dot{e}, e)$ in Example 15.3.

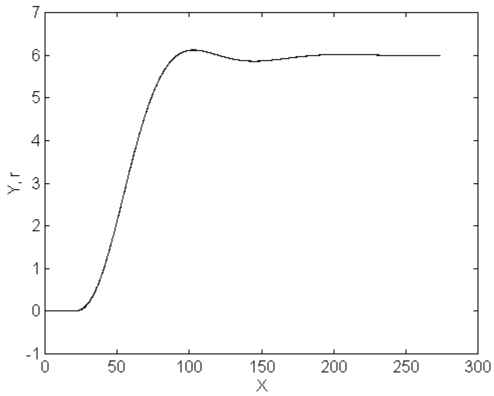


Figure 15.19: Plots of Y and r versus X in Example 15.3.

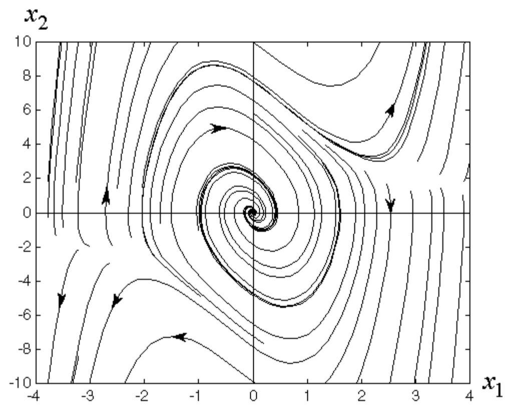


Figure 15.20: A phase-plane portrait of the uncontrolled system in Example 15.4.

differential equation,

$$\ddot{x} + a_1(1 - x^2)\dot{x} + a_2 x = bu, \tag{15.5}$$

where $|u| \leq 15$, $a_1 = 2.2165$, $a_2 = b = 12.7388$. Let $x_1 = x$ and $x_2 = \dot{x}_1$. Then, the above system model in state space format has the form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ -a_2 x_1 - a_1\left(1 - x_1^2\right) x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} u.$$

A phase-plane portrait of the uncontrolled system is shown in Figure 15.20. Plots of x_1 and x_2 versus time for the initial conditions $x_1(0) = -1$ and $x_2(0) = 0$ are shown in Figure 15.21. The relevant membership functions are shown in Figure 15.22, 15.23, 15.24, and 15.25. The following MATLAB function can be used to generate all the relevant membership functions.

```
function y = trap_a(x,a1,a2,c,s)
%TRAP_A is an asymmetric trapezoid type membership function,
% also called a ``pi'' function,
% where x is the function domain, a1 is the length of the
% left trapezoid's side
% at the base, a2 is the length of the right side at the base,
% c is the trapezoid's center, and s is a half of the
% trapezoid upper side. Thus the length of the trapezoid upper
```

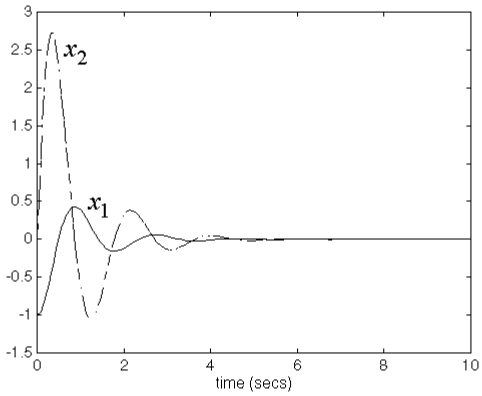


Figure 15.21: Plots of x_1 and x_2 versus time of the uncontrolled system of Example 15.4.

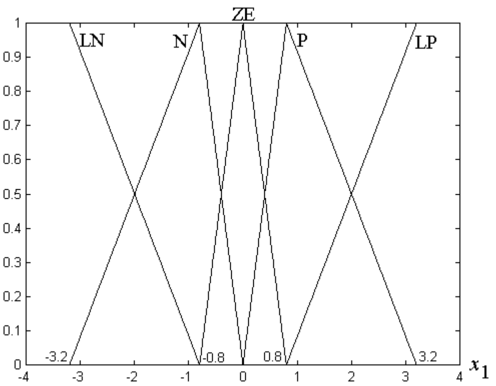


Figure 15.22: Membership functions of x_1 of Example 15.4.

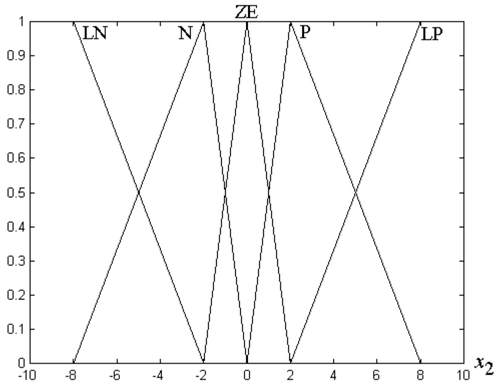


Figure 15.23: Membership functions of x_2 of Example 15.4.

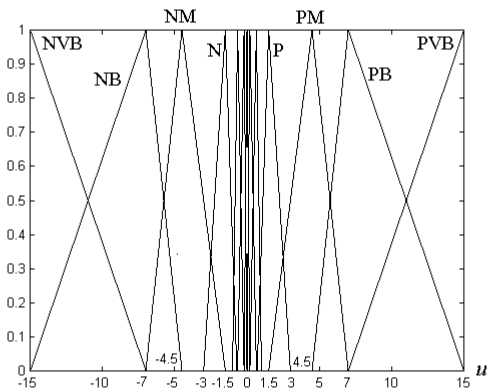


Figure 15.24: Membership functions of u of Example 15.4.

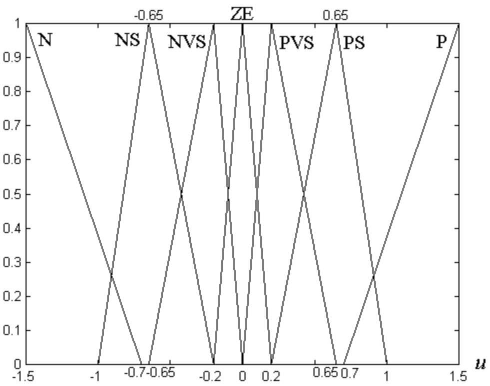


Figure 15.25: Magnification of the membership functions of u on the interval $[-1.5, 1.5]$ of Example 15.4.

```
% side is 2*s. The length of the trapezoid base is then 2*s+a1+a2.
% Set s=0 to get a triangular membership function.

y1=min(0.5,(max((-0.5),(0.5 +((x-c+s)./a1)))));
y2=min(0.5,(max((-0.5),(0.5 -((x-c-s)./a2)))));
y=y1+y2;
```

We used the rules matrix given in Table 15.3 to produce a nonlinear mapping $u = \Phi(x_1, x_2)$ and to construct a look-up table that implements the obtained mapping representing a fuzzy logic controller. The 2-D Look-Up Table block to implement the controller was used. The look-up table size was 22×22. A plot of $u = \Phi(x_1, x_2)$ is shown in Figure 15.26. A SIMULINK diagram of the closed-loop system is given in Figure 15.27. A phase-portrait of the closed-loop system is depicted in Figure 15.28. Plots x_1 and x_2 versus time for $x_1(0) = -1$ and $x_2(0) = 0$ for the closed-loop system are shown in Figure 15.29.

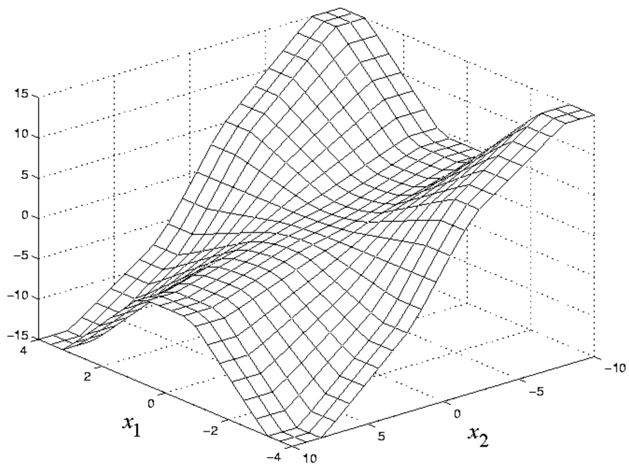


Figure 15.26: A plot of $u = \Phi(x_1, x_2)$ of Example 15.4.

Table 15.3: The rules matrix of Example 15.4.

x_1 / x_2	LN	N	ZE	P	LP
LN	PVB	PM	PM	PM	PVB
N	PB	P	PS	PS	PM
ZE	P	PVS	ZE	NVS	N
P	NM	NS	NS	N	NB
LP	NVB	NM	NM	NM	NVB

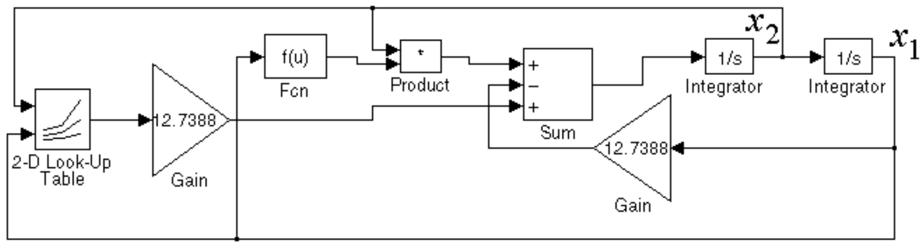


Figure 15.27: A SIMULINK diagram of the closed-loop system of Example 15.4.

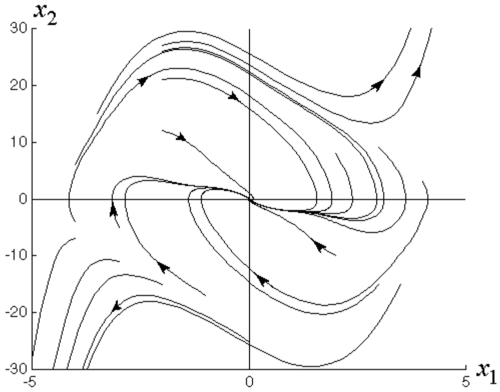


Figure 15.28: A phase-plane portrait of the closed-loop system of Example 15.4.

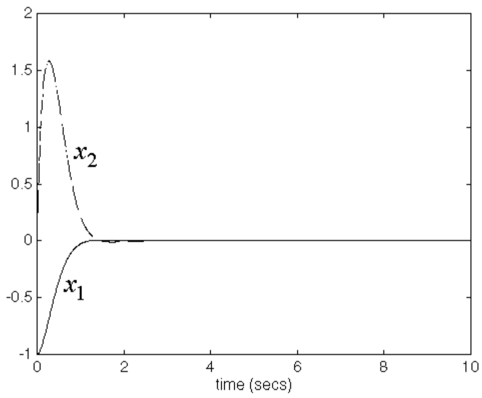


Figure 15.29: Plots x_1 and x_2 versus time for $x_1(0) = -1$ and $x_2(0) = 0$ for the closed-loop system in Example 15.4.

15.4 Stabilization Using Fuzzy Models

In this section, a systematic Lyapunov based stabilizing control design method for uncertain nonlinear dynamical systems using fuzzy models is presented. The controller is constructed using a design model of the dynamical process to be controlled. Recall that the dynamical process to be controlled is also called the plant. We obtain the design model from the truth model using a fuzzy modeling approach.

15.4.1 Fuzzy Modeling

In many situations there may be human experts that can provide a linguistic description, in terms of IF-THEN rules, of the plant dynamical behavior. Combining available mathematical description of the plant with its linguistic description results in a fuzzy system model. Such an approach to modeling of continuous-time dynamical processes was proposed by Takagi and Sugeno [317] and further developed by Sugeno and Kang [312]. This type of model is called the Takagi-Sugeno (T-S) or Takagi-Sugeno-Kang (TSK) fuzzy model. We mention here that the discrete-time version of the TSK model was analyzed by Tanaka and Sugeno [322], and Tanaka [321], while Wang, Tanaka, and Griffin [341], Cao, Rees, and Feng [51], and Feng et al. [101] used TSK fuzzy models to design stabilizing controllers for nonlinear systems.

One can construct a TSK model if local description of the plant is available in terms of local linear models,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i \boldsymbol{x}(t) + \boldsymbol{B}_i \boldsymbol{u}(t), \quad i = 1, 2, \ldots, r,$$

where the state vector $\boldsymbol{x}(t) \in \mathbb{R}^n$, the control input $\boldsymbol{u}(t) \in \mathbb{R}^m$, and the matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are of appropriate dimensions. The above information is then fused with the available IF-THEN rules, where the i-th rule can have the form:

$$\text{Rule i} \;:\; \text{IF } x_1(t) \text{ is } F_1^i \text{ AND} \cdots \text{ AND } x_n(t) \text{ is } F_n^i,$$

$$\text{THEN } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i \boldsymbol{x}(t) + \boldsymbol{B}_i \boldsymbol{u}(t),$$

where F_j^i, $j = 1, 2, \ldots, n$, is the j-th fuzzy set of the i-th rule. Let $\mu_j^i(x_j)$ be the membership function of the fuzzy set F_j^i, and let

$$w^i = w^i(\boldsymbol{x}) = \prod_{j=1}^{n} \mu_j^i(x_j).$$

Then, the resulting fuzzy system model is constructed as the weighted average of the local models and has the form,

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \frac{\sum_{i=1}^r w^i \left(\boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{B}_i \boldsymbol{u}\right)}{\sum_{i=1}^r w^i} \\
&= \sum_{i=1}^{r} \alpha_i \left(\boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{B}_i \boldsymbol{u}\right) \\
&= \left(\sum_{i=1}^{r} \alpha_i \boldsymbol{A}_i\right) \boldsymbol{x} + \left(\sum_{i=1}^{r} \alpha_i \boldsymbol{B}_i\right) \boldsymbol{u} \\
&= \boldsymbol{A}(\boldsymbol{\alpha}) \boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\alpha}) \boldsymbol{u},
\end{aligned} \tag{15.6}$$

where for $i = 1, 2, \ldots, r$,

$$\alpha_i = \frac{w^i}{\sum_{i=1}^r w^i}. \tag{15.7}$$

Note that, for $i = 1, 2, \ldots, r$,

$$\alpha_i \geq 0, \quad \sum_{i=1}^{r} \alpha_i = 1 \quad \text{and} \quad \boldsymbol{\alpha} = \left[\begin{array}{cccc} \alpha_1 & \alpha_2 & \cdots & \alpha_r \end{array}\right]^\top \in [0, 1]^r. \tag{15.8}$$

A system model given by (15.6) and (15.8) is also referred to as in the literature as the *polytopic system*—see, for example, Takaba [316].

15.4.2 Constructing a Fuzzy Design Model Using a Nonlinear Model

In the stabilization problem of a nonlinear plant, we are concerned with constructing a controller so that starting from an arbitrary point, in some neighborhood of the operating point, the controller forces the closed-loop system trajectory to converge to this operating point. On the other hand, if the starting point coincides with the operating point, the closed-loop system trajectory is expected to stay at this point for all subsequent time. When designing a controller, we use a TSK fuzzy design model obtained by combining a mathematical description of the plant in terms of its truth model and a linguistic description of the plant. The essential components of the TSK model are linear local models, which describe the plant dynamical behavior at its different operating points. We show that using common linearization approach to construct local models may result in a fuzzy design model different from what the designer had in his or her mind. For example, suppose that the truth model of a plant has the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}. \tag{15.9}$$

For simplicity of notation, let

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}.$$

Then, we can represent (15.9) as

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}). \tag{15.10}$$

Expanding $\boldsymbol{F}$ into a Taylor series around $(\boldsymbol{x}_0, \boldsymbol{u}_0)$ yields

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}_0, \boldsymbol{u}_0) + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} (\boldsymbol{x} - \boldsymbol{x}_0) + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} (\boldsymbol{u} - \boldsymbol{u}_0) + \text{higher order terms}, \tag{15.11}$$

where

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{k=1}^{m} u_k \boldsymbol{g}_k(\boldsymbol{x}), \tag{15.12}$$

where $\boldsymbol{g}_k$ is the k-th column of $\boldsymbol{G}$. To write expressions for the second and the third terms of (15.11) as functions of $\boldsymbol{f}$ and $\boldsymbol{G}$, let g_{ij} be the (i, j)-th element of the matrix $\boldsymbol{G}$. Then,

$$\left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0} + \sum_{k=1}^{m} u_k \left.\frac{\partial \boldsymbol{g}_k}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0} + \boldsymbol{H}(\boldsymbol{x}_0, \boldsymbol{u}_0), \tag{15.13}$$

where the (i,j)-th element of the $n \times n$ matrix $\boldsymbol{H}$ has the form

$$\sum_{k=1}^{m} u_k \frac{\partial g_{ik}(\boldsymbol{x})}{\partial x_j}\Bigg|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}}.$$

Finally,

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\Bigg|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} = \boldsymbol{G}(\boldsymbol{x}_0). \tag{15.14}$$

A point $(\boldsymbol{x}_0^\top, \boldsymbol{u}_0^\top)^\top \in \mathbb{R}^{n+m}$ is an equilibrium point of (15.10) if $\boldsymbol{F}(\boldsymbol{x}_0, \boldsymbol{u}_0) = \boldsymbol{0}$, that is, if at $(\boldsymbol{x}_0, \boldsymbol{u}_0)$ we have $\dot{\boldsymbol{x}} = \boldsymbol{0}$. Let $\Delta \boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_0$ and $\Delta \boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}_0$. Note that

$$\frac{d\boldsymbol{x}_0}{dt} = \boldsymbol{0}.$$

Then, the linearized model about the equilibrium $(\boldsymbol{x}_0, \boldsymbol{u}_0)$ is obtained by neglecting higher order terms and observing that at the equilibrium point $\boldsymbol{F}(\boldsymbol{x}_0, \boldsymbol{u}_0) = \boldsymbol{0}$. The linearized model has the form

$$\frac{d}{dt}\Delta \boldsymbol{x} = \boldsymbol{A}\Delta \boldsymbol{x} + \boldsymbol{B}\Delta \boldsymbol{u}, \tag{15.15}$$

where

$$\boldsymbol{A} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\Bigg|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} \quad \text{and} \quad \boldsymbol{B} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\Bigg|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}}.$$

Suppose now that we wish to construct a stabilizing controller for the truth model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$, where $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$, so that the pair $(\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{u} = \boldsymbol{0})$ constitutes an asymptotically stable equilibrium point of the closed-loop system. Our first step is to construct a fuzzy design model. We start with generating linear local models describing plant's behavior at selected operating points. It is natural to first construct a local linear model that describes plant's behavior about the equilibrium state $\boldsymbol{x} = \boldsymbol{0}$. We can obtain this first local linear model using the above described linearization technique. The resulting model is given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{B}_1\boldsymbol{u}.$$

Next, we construct local linear models describing plant's behavior at the remaining operating points. Suppose that $\boldsymbol{x} = \boldsymbol{x}_j$ is the next operating state of interest. The result of Taylor's linearization of a nonlinear model about an operating non-equilibrium point is, in general, an affine, rather than a linear model. Even if the operating point is an equilibrium point, Taylor series linearization, in general, will not yield, in general, a local model that is linear. Indeed, suppose that the operating point $(\boldsymbol{x}_j, \boldsymbol{u}_j)$ is an equilibrium point, that is,

$$\boldsymbol{f}(\boldsymbol{x}_j) + \boldsymbol{G}(\boldsymbol{x}_j)\boldsymbol{u}_j = \boldsymbol{0}. \tag{15.16}$$

By (15.11), the resulting linearized model is

$$\begin{aligned}\frac{d}{dt}(\boldsymbol{x}-\boldsymbol{x}_j) &= \boldsymbol{f}(\boldsymbol{x}_j)+\boldsymbol{G}(\boldsymbol{x}_j)\boldsymbol{u}_j+\boldsymbol{A}_j(\boldsymbol{x}-\boldsymbol{x}_j)+\boldsymbol{B}_j(\boldsymbol{u}-\boldsymbol{u}_j) &(15.17)\\ &= \boldsymbol{A}_j(\boldsymbol{x}-\boldsymbol{x}_j)+\boldsymbol{B}_j(\boldsymbol{u}-\boldsymbol{u}_j). &(15.18)\end{aligned}$$

We can represent model (15.18) in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_j\boldsymbol{x}+\boldsymbol{B}_j\boldsymbol{u}-(\boldsymbol{A}_j\boldsymbol{x}_j+\boldsymbol{B}_j\boldsymbol{u}_j)\,. \tag{15.19}$$

The term $(\boldsymbol{A}_j\boldsymbol{x}_j+\boldsymbol{B}_j\boldsymbol{u}_j)$ does not have to be equal to zero, and hence (15.19) is not a linear model, but rather an affine model. We illustrate, on a numerical example, that the term $(\boldsymbol{A}_j\boldsymbol{x}_j+\boldsymbol{B}_j\boldsymbol{u}_j)$ is indeed, in general, not equal to zero.

Example 15.5 Consider a subsystem of model (2.78) of the form

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g\sin(x_1)-mlax_2^2\sin(2x_1)/2}{4l/3-mla\cos^2(x_1)}\end{bmatrix} + \begin{bmatrix}0\\ -\frac{a\cos(x_1)}{4l/3-mla\cos^2(x_1)}\end{bmatrix}u, \tag{15.20}$$

where $g = 9.8$ m/sec^2, $m = 2$ kg, $M = 8$ kg, $a = 1/(m+M)$, $l = 0.5$ m. We linearize (15.20) about $x_1 = \pi/4$. For $x_1 = \pi/4$ to be a component of an equilibrium state, we have to have $x_2 = 0$. Hence, the equilibrium state is

$$\boldsymbol{x}_e = \begin{bmatrix}\pi/4 & 0\end{bmatrix}^\top.$$

We next compute $u = u_e$ so that equation (15.16) holds. Performing simple calculations, we get $u_e = 98$. Then, using (15.13) we obtain

$$\boldsymbol{A} = \begin{bmatrix}0 & 1\\ 22.4745 & 0\end{bmatrix}. \tag{15.21}$$

Next, using (15.14) gives

$$\boldsymbol{B} = \begin{bmatrix}0\\ -0.1147\end{bmatrix}. \tag{15.22}$$

Note that

$$\boldsymbol{A}\boldsymbol{x}_e+\boldsymbol{B}u_e = \begin{bmatrix}0\\ 6.4108\end{bmatrix} \neq \boldsymbol{0}.$$

The above example illustrates the fact that the system being linear in $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_0$ and $\Delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}_0$ is, in general, only affine, and not linear in $\boldsymbol{x}$ and $\boldsymbol{u}$ because of the presence of the offset term $-(\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{B}\boldsymbol{u}_0)$. The designer must than be careful when constructing local models using Taylor series linearization because this approach, in general, will not yield local models linear in $\boldsymbol{x}$ and $\boldsymbol{u}$. We note that Taylor's linearization yields a linear system in $\boldsymbol{x}$ and $\boldsymbol{u}$ if the equilibrium point is $(\boldsymbol{x}_0^\top, \boldsymbol{u}_0^\top) = (\boldsymbol{0}^\top, \boldsymbol{0}^\top)$.

We now describe our method of constructing local models using the truth model of the given plant. We assume that the truth model has the form of (15.9). Suppose that we are given an operating state $\boldsymbol{x}_0$ that does not have to be an equilibrium state of (15.9). Our goal is to construct a linear model in $\boldsymbol{x}$ and $\boldsymbol{u}$ that approximates the behavior of (15.9) in the vicinity of the operating state $\boldsymbol{x}_0$, that is, we wish to find constant matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ such that in a neighborhood of $\boldsymbol{x}_0$,

$$\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \approx \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \tag{15.23}$$

and

$$\boldsymbol{f}(\boldsymbol{x}_0) + \boldsymbol{G}(\boldsymbol{x}_0)\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{B}\boldsymbol{u} \quad \text{for all} \quad \boldsymbol{u}. \tag{15.24}$$

Because $\boldsymbol{u}$ is arbitrary, we have to have

$$\boldsymbol{G}(\boldsymbol{x}_0) = \boldsymbol{B}. \tag{15.25}$$

Thus, we are left with finding a constant matrix $\boldsymbol{A}$ such that in a neighborhood of $\boldsymbol{x}_0$,

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{A}\boldsymbol{x}, \tag{15.26}$$

and

$$\boldsymbol{f}(\boldsymbol{x}_0) = \boldsymbol{A}\boldsymbol{x}_0. \tag{15.27}$$

Let $\boldsymbol{a}_i^\top$ denote the i-th row of the matrix $\boldsymbol{A}$. Then, for the purpose of further analysis, we represent condition (15.26) as

$$f_i(\boldsymbol{x}) \approx \boldsymbol{a}_i^\top \boldsymbol{x}, \quad i = 1, 2, \ldots, n, \tag{15.28}$$

and we represent (15.27) as

$$f_i(\boldsymbol{x}_0) = \boldsymbol{a}_i^\top \boldsymbol{x}_0, \quad i = 1, 2, \ldots, n, \tag{15.29}$$

where the i-th component of $\boldsymbol{f}$ is $f_i : \mathbb{R}^n \to \mathbb{R}$. Expanding the left hand side of (15.28) about $\boldsymbol{x}_0$ and neglecting second and higher order terms yields

$$f_i(\boldsymbol{x}_0) + \nabla f_i(\boldsymbol{x}_0)^\top (\boldsymbol{x} - \boldsymbol{x}_0) \approx \boldsymbol{a}_i^\top \boldsymbol{x}, \tag{15.30}$$

where $\nabla f_i(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^n$ is the gradient (a column vector) of f_i at $\boldsymbol{x}$. Now, using (15.29), we represent (15.30) as

$$\nabla f_i(\boldsymbol{x}_0)^\top (\boldsymbol{x} - \boldsymbol{x}_0) \approx \boldsymbol{a}_i^\top (\boldsymbol{x} - \boldsymbol{x}_0), \tag{15.31}$$

where $\boldsymbol{x}$ is arbitrary but "close" to $\boldsymbol{x}_0$. Our task now is to determine a constant vector $\boldsymbol{a}_i$ that is as "close as possible" to $\nabla f_i(\boldsymbol{x}_0)$ and satisfies the constraint $\boldsymbol{a}_i^\top \boldsymbol{x}_0 = f_i(\boldsymbol{x}_0)$. Let

$$E = \frac{1}{2} \|\nabla f_i(\boldsymbol{x}_0) - \boldsymbol{a}_i\|_2^2 .$$

Then, we can formulate our task as a constrained optimization problem of the form

$$\left.\begin{array}{l} \underset{\boldsymbol{a}_i}{\text{minimize}}\ E \\ \text{subject to } \boldsymbol{a}_i^\top \boldsymbol{x}_0 = f_i(\boldsymbol{x}_0). \end{array}\right\} \tag{15.32}$$

We note that (15.32) is a convex constrained optimization problem. This means that the first-order necessary condition for a minimum of E is also sufficient, see, for example, Luenberger [224, Section 6.5] or Chong and Żak [62, Chapter 21]. The first-order conditions for the optimization problem given by (15.32) are

$$\nabla_{\boldsymbol{a}_i} E + \lambda \nabla_{\boldsymbol{a}_i} \left(\boldsymbol{a}_i^\top \boldsymbol{x}_0 - f_i(\boldsymbol{x}_0) \right) = \boldsymbol{0}, \tag{15.33}$$

$$\boldsymbol{a}_i^\top \boldsymbol{x}_0 = f_i(\boldsymbol{x}_0), \tag{15.34}$$

where λ is the Lagrange multiplier and the subscript $\boldsymbol{a}_i$ in $\nabla_{\boldsymbol{a}_i}$ indicates that the gradient, ∇ is computed with respect to $\boldsymbol{a}_i$. Performing the required differentiation in (15.33) yields

$$\boldsymbol{a}_i - \nabla f_i(\boldsymbol{x}_0) + \lambda \boldsymbol{x}_0 = \boldsymbol{0}, \tag{15.35}$$

$$\boldsymbol{a}_i^\top \boldsymbol{x}_0 = f_i(\boldsymbol{x}_0). \tag{15.36}$$

Recall that we consider now the case when $\boldsymbol{x}_0 \neq \boldsymbol{0}$. Premultiplying (15.35) by $\boldsymbol{x}_0^\top$ and substituting (15.36) into the resulting equation yields

$$\lambda = \frac{\boldsymbol{x}_0^\top \nabla f_i(\boldsymbol{x}_0) - f_i(\boldsymbol{x}_0)}{\|\boldsymbol{x}_0\|^2}. \tag{15.37}$$

Substituting λ given by (15.37) into (15.35), we obtain

$$\boxed{\boldsymbol{a}_i = \nabla f_i(\boldsymbol{x}_0) + \frac{f_i(\boldsymbol{x}_0) - \boldsymbol{x}_0^\top \nabla f_i(\boldsymbol{x}_0)}{\|\boldsymbol{x}_0\|^2} \boldsymbol{x}_0, \quad \boldsymbol{x}_0 \neq \boldsymbol{0}} \tag{15.38}$$

We now summarize the above development. Let

$$\boldsymbol{J}(\boldsymbol{x}_0) = \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \right|_{\boldsymbol{x} = \boldsymbol{x}_0}$$

be the Jacobian matrix of $\boldsymbol{f}$ evaluated at the operating state $\boldsymbol{x} = \boldsymbol{x}_0$. Then a linear model in $\boldsymbol{x}$ and $\boldsymbol{u}$ that approximates the behavior of (15.9) in the vicinity of the operating state $\boldsymbol{x}_0$ has the form

$$\boxed{\dot{\boldsymbol{x}} = \left[\boldsymbol{J}(\boldsymbol{x}_0) + \frac{(\boldsymbol{f}(\boldsymbol{x}_0) - \boldsymbol{J}(\boldsymbol{x}_0)\boldsymbol{x}_0)\,\boldsymbol{x}_0^\top}{\|\boldsymbol{x}_0\|^2}\right]\boldsymbol{x} + \boldsymbol{G}(\boldsymbol{x}_0)\boldsymbol{u}, \quad \boldsymbol{x}_0 \neq \boldsymbol{0}} \tag{15.39}$$

On the other hand, if the operating pair $(\boldsymbol{x}_0, \boldsymbol{u}_0)$ is an equilibrium pair, then Taylor's linearization of (15.9) would yield the following linearized model;

$$\frac{d}{dt}(\boldsymbol{x} - \boldsymbol{x}_0) = \left[\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0} + \sum_{k=1}^{m} u_k \left.\frac{\partial \boldsymbol{g}_k}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}}\right](\boldsymbol{x} - \boldsymbol{x}_0) + \boldsymbol{G}(\boldsymbol{x}_0)(\boldsymbol{u} - \boldsymbol{u}_0),$$

where the matrix

$$\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0} + \sum_{k=1}^{m} u_k \left.\frac{\partial \boldsymbol{g}_k}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}}$$

is computed in (15.13).

Example 15.6 We apply (15.38) to construct local linear models of the nonlinear model (15.20) that we analyzed in Example 15.5. This nonlinear model was also considered by Wang et al. [341, page 20]. Suppose that, as in Wang et al. [341], we wish to construct two local models; one describing the plant operation about $x_1 = 0$ and the other local model when x_1 is about $\pm\pi/2$. In both models, we assume that $x_2 = 0$ and u is arbitrary. Note that the nonlinear model (15.20) is in the format given by (15.9). We first construct a local model about the operating state $x_1 = x_2 = 0$. The local model for this case can be obtained using Taylor's linearization about the equilibrium point $(\boldsymbol{x}_0, \boldsymbol{u}_0) = (\boldsymbol{0}, \boldsymbol{0})$.

We next construct the second local model that corresponds to $x_1 = 88°\pi/180°$ and $x_2 = 0$. Let $\beta = \cos(88°)$. Then, the input matrix $\boldsymbol{B}$ can be obtained from (15.20) using (15.25),

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ -\frac{a\beta}{4l/3 - mla\beta^2} \end{bmatrix}.$$

We then compute the matrix $\boldsymbol{A}$ of the second local model using (15.38). We first compute the first row $\boldsymbol{A}$. We note that $f_1(\boldsymbol{x}) = x_2$, and hence

$$\nabla f_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top.$$

The operating state is $\boldsymbol{x}_0 = \begin{bmatrix} 88°\pi/180° & 0 \end{bmatrix}^\top$. We compute $\boldsymbol{a}_1^\top$, the first row of $\boldsymbol{A}$, using (15.38) to get

$$\boldsymbol{a}_1^\top = \nabla^\top f_1(\boldsymbol{x}_0) + \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

We again use (15.38) to compute $\boldsymbol{a}_2^\top$, the second row of the matrix $\boldsymbol{A}$, of the second local model,

$$\boldsymbol{a}_2^\top = \begin{bmatrix} f_2(\boldsymbol{x}_0)/(88°\pi/180°) & 0 \end{bmatrix} \approx \begin{bmatrix} \frac{g}{4l/3 - mla\beta^2}\left(\frac{2}{\pi}\right) & 0 \end{bmatrix}.$$

The above obtained local models are the same as the ones that were proposed by Wang et al. [341] using a heuristic approach.

15.4.3 Stabilizability of Fuzzy Models

In this subsection, we first present sufficient conditions for asymptotic stability in the large of unforced continuous-time fuzzy model

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{r} \alpha_i \boldsymbol{A}_i \boldsymbol{x}, \tag{15.40}$$

where $\alpha_i = \alpha_i(\boldsymbol{x}(t)) \geq 0$ for $i = 1, 2, \ldots, r$, and $\sum_{i=1}^{r} \alpha_i = 1$. The above model can be viewed as an example of unforced continuous-time polytopic model. We next discuss the stabilizability problem of forced fuzzy models using fuzzy state-feedback control strategies.

It is straightforward to show that a sufficient condition for asymptotic stability in the large of the equilibrium state $\boldsymbol{x} = \boldsymbol{0}$ of system (15.40) is that there exists a symmetric positive definite matrix $\boldsymbol{P}$ such that for $i = 1, 2, \ldots, r$,

$$\boldsymbol{A}_i^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i < 0. \tag{15.41}$$

It is obvious that a necessary condition for the existence of a common symmetric positive definite $\boldsymbol{P}$ satisfying (15.41) is that each $\boldsymbol{A}_i$ be asymptotically stable, that is, the eigenvalues of each $\boldsymbol{A}_i$ be in the open left-hand complex plane. We now give another necessary condition for the existence of a common $\boldsymbol{P}$ that satisfies (15.41).

Theorem 15.1 *Suppose that each $\boldsymbol{A}_i$, $i = 1, 2, \ldots, r$, in the model (15.40) is asymptotically stable and there exists a symmetric positive definite $\boldsymbol{P}$ such that conditions (15.41) hold. Then, the matrices*

$$\sum_{k=1}^{s} \boldsymbol{A}_{i_k},$$

where $i_k \in \{1, 2, \ldots, r\}$ and $s = 1, 2, 3, \ldots, r$ are asymptotically stable.

Proof Suppose that conditions (15.41) hold. Let

$$\boldsymbol{A}_{i_k}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_{i_k} = -\boldsymbol{Q}_{i_k}, \tag{15.42}$$

where $\boldsymbol{Q}_{i_k} = \boldsymbol{Q}_{i_k}^\top > 0$ for $i_k \in \{1, 2, \ldots, r\}$. Summing both sides of (15.42) yields

$$\left(\sum_{k=1}^{s} \boldsymbol{A}_{i_k}^\top\right)\boldsymbol{P} + \boldsymbol{P}\left(\sum_{k=1}^{s} \boldsymbol{A}_{i_k}\right) = -\sum_{k=1}^{s} \boldsymbol{Q}_{i_k}. \tag{15.43}$$

By assumption $\boldsymbol{P} = \boldsymbol{P}^\top > 0$, and because $\boldsymbol{Q}_{i_k} = \boldsymbol{Q}_{i_k}^\top > 0$ for $i_k \in \{1, 2, \ldots, r\}$, the symmetric matrices

$$\sum_{k=1}^{s} \boldsymbol{Q}_{i_k}, \quad s = 2, 3, \ldots, r,$$

are also positive definite. Hence, by the Lyapunov theorem, the matrices $\sum_{k=1}^{s} \boldsymbol{A}_{i_k}$, $s = 2, 3, \ldots, r$, are asymptotically stable.

□

The following corollary can be viewed as a continuous-time counterpart of Theorem 4.3 of Tanaka and Sugeno [322, pages 146–147].

Corollary 15.1 *Suppose that each $\boldsymbol{A}_i$, $i = 1, 2, \ldots, r$ in fuzzy model (15.40) is asymptotically stable and there exists a symmetric positive definite $\boldsymbol{P}$ such that conditions (15.41) hold. Then, the matrices*

$$\boldsymbol{A}_i + \boldsymbol{A}_j, \quad i, j = 1, 2, \ldots, r,$$

are asymptotically stable.

We mention that Tanaka and Sugeno's Theorem 4.3 in [322] can be generalized to correspond to Theorem 15.1 rather than to the above corollary.

The above corollary can be restated in an equivalent way as follows. If there exist i and j such that a matrix $\boldsymbol{A}_i + \boldsymbol{A}_j$ is not asymptotically stable, then there is no positive definite matrix $\boldsymbol{P}$ such that $\boldsymbol{A}_i^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i < 0$ for $i = 1, 2, \ldots, r$. The following example illustrates this fact.

Example 15.7 Given the unforced fuzzy system model described by (15.40), where $r = 2$ and

$$\boldsymbol{A}_1 = \begin{bmatrix} -1 & 4 \\ 0 & -2 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} -1 & 0 \\ 4 & -2 \end{bmatrix}.$$

Obviously each $\boldsymbol{A}_i$, $i = 1, 2$, is asymptotically stable. However, there is no common positive definite $\boldsymbol{P}$ such that $\boldsymbol{A}_i^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i < 0$ for $i = 1, 2$ because the matrix

$$\boldsymbol{A}_1 + \boldsymbol{A}_2 = \begin{bmatrix} -1 & 4 \\ 0 & -2 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 4 & -2 \end{bmatrix} = \begin{bmatrix} -2 & 4 \\ 4 & -4 \end{bmatrix}$$

is unstable since its eigenvalues are

$$\{1.1231, -7.1231\}.$$

Theorem 15.1 can be used to check if there does not exist a positive definite matrix $\boldsymbol{P}$ satisfying conditions (15.41) because this theorem gives us a sufficient condition for non-existence of a common $\boldsymbol{P}$ that satisfies (15.41).

We now present a sufficiency condition for asymptotic stabilizability in the large of the model given by (15.6) and (15.8) using fuzzy state feedback control law

$$\boldsymbol{u} = -\sum_{j=1}^{r} \alpha_j \boldsymbol{K}_j \boldsymbol{x}. \tag{15.44}$$

The closed-loop system has the form

$$\boxed{\dot{\boldsymbol{x}} = \sum_{i=1}^{r}\sum_{j=1}^{r} \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j\right) \boldsymbol{x}} \tag{15.45}$$

To obtain the above model we used the fact that $\sum_{j=1}^{r} \alpha_j = 1$. Suppose that the gain matrices, $\boldsymbol{K}_i$, were chosen so that the matrices

$$\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_i, \quad i = 1, 2, \ldots, r,$$

have their eigenvalues in the open left-hand complex plane, that is, they are asymptotically stable. Let

$$\boldsymbol{G}_{ij} = (\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j) + (\boldsymbol{A}_j - \boldsymbol{B}_j \boldsymbol{K}_i) \quad \text{for} \quad i < j \le r.$$

We present sufficient conditions of Wang et al. [341] for the closed-loop system modeled by (15.45) to be asymptotically stable in the large.

Theorem 15.2 *If for some symmetric positive definite $\boldsymbol{P}$ the following conditions are satisfied:*

$$(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_i)^\top \boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_i) < 0, \quad i = 1, 2, \ldots, r \tag{15.46}$$

and

$$\boldsymbol{G}_{ij}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{G}_{ij} < 0, \quad i < j \le r, \tag{15.47}$$

then the closed-loop system modeled by (15.45) is asymptotically stable in the large.

Proof Let $\boldsymbol{P}$ be such that (15.46) and (15.47) are satisfied. We evaluate the time derivative of $\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)$ on the trajectories of (15.45) to get

$$\begin{aligned}
\frac{d}{dt}\left(\boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t)\right) &= 2\boldsymbol{x}^\top \boldsymbol{P}\dot{\boldsymbol{x}} \\
&= 2\boldsymbol{x}^\top \boldsymbol{P}\left(\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j)\boldsymbol{x}\right) \\
&= \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\boldsymbol{x}^\top\Big((\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j)^\top \boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j)\Big)\boldsymbol{x} \\
&= \sum_{i=1}^{r}\alpha_i^2\boldsymbol{x}^\top\Big((\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_i)^\top \boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_i)\Big)\boldsymbol{x} \\
&\qquad + \sum_{i=1}^{r}\sum_{j>i}^{r}\alpha_i\alpha_j\boldsymbol{x}^\top\left(\boldsymbol{G}_{ij}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{G}_{ij}\right)\boldsymbol{x}.
\end{aligned} \tag{15.48}$$

Hence, when (15.46) and (15.47) are satisfied, the positive definite quadratic function $\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{x}$ becomes a Lyapunov function for the system (15.45), which implies asymptotic stability in the large of (15.45). The proof of the theorem is complete.

□

We give a sufficient condition for the stability of system (15.45) for the case when conditions (15.47) are not satisfied. To proceed, we need the following notation. Suppose that there is a positive definite $\boldsymbol{P}$ such that conditions (15.46) are satisfied. Represent conditions (15.46) as

$$(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_i)^\top \boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_i) = -\boldsymbol{Q}_i, \quad i = 1, 2, \ldots, r, \tag{15.49}$$

where each $\boldsymbol{Q}_i$ is symmetric positive definite. Let λ_i denote the minimum eigenvalue of $\boldsymbol{Q}_i$. We have

$$\lambda_i > 0, \quad i = 1, 2, \ldots, r, \tag{15.50}$$

because $\boldsymbol{Q}_i = \boldsymbol{Q}_i^\top > 0$. Let

$$\boldsymbol{G}_{ij}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{G}_{ij} = -\boldsymbol{Q}_{ij}, \quad i < j \le r, \tag{15.51}$$

where now we do not require $\boldsymbol{Q}_{ij}$ to be positive definite. Note that $\boldsymbol{Q}_{ij} = \boldsymbol{Q}_{ij}^\top$, and hence the eigenvalues of $\boldsymbol{Q}_{ij}$ are real. Let λ_{ij} denote the minimum eigenvalue of $\boldsymbol{Q}_{ij}$. We are ready to state and prove the following theorem.

Theorem 15.3 *Suppose that each $(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_i)$, $i = 1, 2, \ldots, r$, in (15.45) is asymptotically stable and there exists a symmetric positive definite $\boldsymbol{P}$ such that conditions (15.46) hold. Then, the closed-loop fuzzy model (15.45) is asymptotically stable in the large if the matrix*

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & \lambda_{12}/2 & \cdots & \lambda_{1r}/2 \\ \lambda_{12}/2 & \lambda_2 & & \lambda_{2r}/2 \\ \vdots & & \ddots & \vdots \\ \lambda_{1r}/2 & \lambda_{2r}/2 & \cdots & \lambda_r \end{bmatrix} \tag{15.52}$$

is positive definite.

Proof Let

$$W = W(\boldsymbol{x}(t)) = \boldsymbol{x}(t)^\top \boldsymbol{P}\boldsymbol{x}(t).$$

be the Lyapunov function candidate for the closed-loop system modeled by (15.45). We then evaluate the time derivative of W on the trajectories of (15.45) to get

$$\begin{aligned} \dot{W} &= 2\boldsymbol{x}^\top \boldsymbol{P}\dot{\boldsymbol{x}} \\ &= 2\boldsymbol{x}^\top \boldsymbol{P}\left(\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\left(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j\right)\boldsymbol{x}\right) \\ &= \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\boldsymbol{x}^\top\Big(\left(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j\right)^\top\boldsymbol{P} + \boldsymbol{P}\left(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j\right)\Big)\boldsymbol{x} \\ &= -\sum_{i=1}^{r}\alpha_i^2\boldsymbol{x}^\top\boldsymbol{Q}_i\boldsymbol{x} - \sum_{i=1}^{r}\sum_{j>i}\alpha_i\alpha_j\boldsymbol{x}^\top\boldsymbol{Q}_{ij}\boldsymbol{x}. \end{aligned} \tag{15.53}$$

Recall that for any symmetric matrix $\boldsymbol{M} = \boldsymbol{M}^\top$,

$$\lambda_{\min}(\boldsymbol{M})\|\boldsymbol{x}\|^2 \le \boldsymbol{x}^\top\boldsymbol{M}\boldsymbol{x},$$

where $\lambda_{\min}(\boldsymbol{M})$ is the smallest eigenvalue of $\boldsymbol{M}$. Using this fact in (15.53) yields

$$\begin{aligned}
\dot{W} &\le -\left(\sum_{i=1}^{r}\alpha_i^2\lambda_i + \sum_{i=1}^{r}\sum_{j>i}^{r}\alpha_i\alpha_j\lambda_{ij}\right)\|\boldsymbol{x}\|^2 \\
&= -\left(\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_r\end{bmatrix}\begin{bmatrix}\lambda_1 & \lambda_{12}/2 & \cdots & \lambda_{1r}/2 \\ \lambda_{12}/2 & \lambda_2 & & \lambda_{2r}/2 \\ \vdots & & \ddots & \vdots \\ \lambda_{1r}/2 & \lambda_{2r}/2 & \cdots & \lambda_r\end{bmatrix}\begin{bmatrix}\alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_r\end{bmatrix}\right)\|\boldsymbol{x}\|^2 \\
&= -\left(\boldsymbol{\alpha}^\top\boldsymbol{\Lambda}\boldsymbol{\alpha}\right)\|\boldsymbol{x}\|^2. \qquad (15.54)
\end{aligned}$$

The Lyapunov derivative is negative definite if the matrix $\boldsymbol{\Lambda}$ is positive definite, which completes the proof.

□

15.4.4 A Lyapunov Based Stabilizer

In this subsection, we apply a Lyapunov approach to the fuzzy control design for uncertain dynamical systems of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{\alpha})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\alpha})(\boldsymbol{u} + \boldsymbol{h}(t,\boldsymbol{x})), \qquad (15.55)$$

where $\boldsymbol{\alpha}$ satisfies (15.8) and the vector-valued function $\boldsymbol{h}$ represents modeling uncertainties or disturbances. The only information about uncertainties available to the designer are their bounds. Specifically, we assume that the uncertainty modeling function is bounded by a non-negative function $\eta = \eta(t,\boldsymbol{x})$, that is,

$$\|\boldsymbol{h}(t,\boldsymbol{x})\| \le \eta(t,\boldsymbol{x}),$$

where $\|\cdot\|$ denotes the vector p-norm. Observe that $\boldsymbol{h}$ affects the system dynamics via the input matrix $\boldsymbol{B}(\boldsymbol{\alpha})$ in a similar fashion as the control input $\boldsymbol{u}$ does. For this reason, we say that the uncertainty $\boldsymbol{h}$ satisfies the matching condition.

Some of the Lyapunov based control strategies, using non-fuzzy controllers, for uncertain systems are documented in Zinober [375]. The method of constructing stabilizing controllers, presented in this section, requires that each of the matrices $\boldsymbol{A}_i$, $i = 1, 2, \ldots, r$ is already asymptotically stable, and that there exists a common symmetric positive definite $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ such that, for $i = 1, 2, \ldots, r$,

$$\boldsymbol{A}_i^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i < 0. \qquad (15.56)$$

If this is not the case, we break the design of stabilizing control law, $\boldsymbol{u}$, into two parts $\boldsymbol{u} = \boldsymbol{u}_1 + \boldsymbol{u}_2$. We first design $\boldsymbol{u}_1$ of the form

$$\boldsymbol{u}_1 = -\sum_{j=1}^{r} \alpha_j \boldsymbol{K}_j \boldsymbol{x} \tag{15.57}$$

so that the system

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{r} \sum_{j=1}^{r} \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j \right) \boldsymbol{x}$$

is asymptotically stable in the large and there exists a common $\boldsymbol{P}$ such that conditions (15.46) and (15.47) or conditions of Theorem 15.3 are satisfied. Note that the system modeled by (15.55) and driven by (15.57) has the form

$$\begin{aligned} \dot{\boldsymbol{x}} &= \sum_{i=1}^{r} \alpha_i \boldsymbol{A}_i \boldsymbol{x} - \sum_{i=1}^{r} \alpha_i \boldsymbol{B}_i \left(\sum_{j=1}^{r} \alpha_j \boldsymbol{K}_j \boldsymbol{x} \right) + \sum_{i=1}^{r} \alpha_i \boldsymbol{B}_i \left(\boldsymbol{u}_2 + \boldsymbol{h} \right) \\ &= \sum_{i=1}^{r} \sum_{j=1}^{r} \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j \right) \boldsymbol{x} + \sum_{i=1}^{r} \alpha_i \boldsymbol{B}_i \left(\boldsymbol{u}_2 + \boldsymbol{h} \right). \end{aligned} \tag{15.58}$$

We next proceed with the design of $\boldsymbol{u}_2$. The role of $\boldsymbol{u}_2$ is to reject the uncertainty $\boldsymbol{h}$. Then, $\boldsymbol{u} = \boldsymbol{u}_1 + \boldsymbol{u}_2$ will asymptotically stabilize in the large the uncertain system (15.55). We propose a family of fuzzy controllers $\boldsymbol{u}_2$ that reject the uncertainty $\boldsymbol{h}$. These controllers are obtained by fuzzifying the controllers proposed in [149]. To proceed, we need the following notation. Let

$$\boldsymbol{\mu} = \left[\begin{array}{cccc} \mu_1 & \mu_2 & \dots & \mu_m \end{array} \right]^\top = \sum_{i=1}^{r} \alpha_i \boldsymbol{B}_i^\top \boldsymbol{P} \boldsymbol{x}. \tag{15.59}$$

Let $\|\cdot\|_p$ denote the p-norm of a vector, that is, if a vector $\boldsymbol{v} \in \mathbb{R}^m$, then its p-norm is defined as

$$\|\boldsymbol{v}\|_p = \left(|v_1|^p + |v_2|^p + \cdots + |v_m|^p \right)^{1/p}, \quad p \geq 1.$$

A well-known result concerning p-norms is the Hölder's inequality,

$$\left| \boldsymbol{v}^\top \boldsymbol{w} \right| \leq \|\boldsymbol{v}\|_p \|\boldsymbol{w}\|_q, \quad \frac{1}{p} + \frac{1}{q} = 1.$$

All norms on $\mathbb{R}^m$ are equivalent in the sense that if $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are norms on $\mathbb{R}^m$, then there exist positive constants C_1 and C_2 such that

$$C_1 \|\boldsymbol{v}\|_\alpha \leq \|\boldsymbol{v}\|_\beta \leq C_2 \|\boldsymbol{v}\|_\alpha \quad \text{for all} \quad \boldsymbol{v} \in \mathbb{R}^m. \tag{15.60}$$

Finally, the gradient of $\|\boldsymbol{\mu}\|_p$, where it is defined, has the form

$$\nabla\|\boldsymbol{\mu}\|_p = \begin{bmatrix} \frac{\partial\|\boldsymbol{\mu}\|_p}{\partial\mu_1} & \frac{\partial\|\boldsymbol{\mu}\|_p}{\partial\mu_2} & \cdots & \frac{\partial\|\boldsymbol{\mu}\|_p}{\partial\mu_m} \end{bmatrix}^\top, \quad p \geq 1. \tag{15.61}$$

Applying the chain rule, we compute

$$\nabla\|\boldsymbol{\mu}\|_p = \frac{1}{\|\boldsymbol{\mu}\|_p^{p-1}} \begin{bmatrix} |\mu_1|^{p-1}\operatorname{sign}(\mu_1) \\ |\mu_2|^{p-1}\operatorname{sign}(\mu_2) \\ \vdots \\ |\mu_m|^{p-1}\operatorname{sign}(\mu_m) \end{bmatrix}. \tag{15.62}$$

Let $\|\boldsymbol{h}(t,\boldsymbol{x})\|_q \leq \eta_q$, where $\eta_q \geq 0$ is a constant. We then show that when

$$\boldsymbol{u}_2 = -\eta_q\nabla\|\boldsymbol{\mu}\|_p, \quad p \geq 1, \quad \text{and} \quad \frac{1}{p}+\frac{1}{q} = 1, \tag{15.63}$$

then the closed-loop system driven by $\boldsymbol{u} = \boldsymbol{u}_1 + \boldsymbol{u}_2$ is asymptotically stable in the large for any uncertainty $\boldsymbol{h}$ such that $\|\boldsymbol{h}\|_q \leq \eta_q$.

Theorem 15.4 *Suppose that $\boldsymbol{u}_1$, given by (15.57), is such that there exists a common $\boldsymbol{P}$ satisfying conditions (15.46) and (15.47) or conditions of Theorem 15.3. Then, the system (15.55) driven by $\boldsymbol{u} = \boldsymbol{u}_1 + \boldsymbol{u}_2$, where $\boldsymbol{u}_2$ is given by (15.63), is asymptotically stable in the large for any uncertainty $\boldsymbol{h}$ such that $\|\boldsymbol{h}\|_q \leq \eta_q$.*

Proof Suppose that $\boldsymbol{u}_1$, given by (15.57), was constructed such that (15.46) and (15.47), or conditions of Theorem 15.3 hold. Then, (15.55) driven by $\boldsymbol{u}_1$ takes the form (15.58), that is,

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\left(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j\right)\boldsymbol{x} + \sum_{i=1}^{r}\alpha_i\boldsymbol{B}_i\left(\boldsymbol{u}_2+\boldsymbol{h}\right).$$

We now show that

$$W = \boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{x}, \tag{15.64}$$

is the Lyapunov function for the closed-loop system, where $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ satisfies (15.46) and (15.47) or conditions of Theorem 15.3. For this, we evaluate the Lyapunov derivative of W, that is, we evaluate $\frac{d}{dt}W$ on the trajectories of (15.58), and show that the Lyapunov derivative of W is negative definite. Before we proceed with evaluating $\frac{d}{dt}W$, observe that using the fact that

$$\operatorname{sign}(\mu_j) = \frac{|\mu_j|}{\mu_j},$$

and (15.62), we obtain

$$\boldsymbol{\mu}^\top \nabla \|\boldsymbol{\mu}\|_p = \|\boldsymbol{\mu}\|_p.$$

Hence,

$$\begin{aligned}
\frac{d}{dt}W &= 2\boldsymbol{x}^\top \boldsymbol{P}\left(\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\left(\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}_j\right)\boldsymbol{x} + \sum_{i=1}^{r}\alpha_i\boldsymbol{B}_i\left(\boldsymbol{u}_2 + \boldsymbol{h}\right)\right)\\
&< -2\eta_q\boldsymbol{\mu}^\top\nabla\|\boldsymbol{\mu}\|_p + 2\boldsymbol{\mu}^\top\boldsymbol{h}\\
&\le -2\eta_q\|\boldsymbol{\mu}\|_p + 2\left|\boldsymbol{\mu}^\top\boldsymbol{h}\right|. \qquad (15.65)
\end{aligned}$$

Applying Hölder's inequality to the above yields

$$\begin{aligned}
\frac{d}{dt}W &< -2\eta_q\|\boldsymbol{\mu}\|_p + 2\|\boldsymbol{\mu}\|_p\|\boldsymbol{h}\|_q\\
&\le -2\eta_q\|\boldsymbol{\mu}\|_p + 2\|\boldsymbol{\mu}\|_p\,\eta_q. \qquad (15.66)
\end{aligned}$$

Therefore, $\frac{d}{dt}W < 0$ and by Lyapunov's theorem, the closed loop-system is asymptotically stable in the large.

□

Example 15.8 Using system model (2.79), we propose the following two rules describing the plant dynamics:

$$\text{Rule 1 : IF } x_1(t) \text{ is about } 0$$

$$\text{THEN } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_1\boldsymbol{x}(t) + \boldsymbol{B}_1(u(t) + h),$$

and

$$\text{Rule 2 : IF } x_1(t) \text{ is about } \pm\pi/4$$

$$\text{THEN } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_2\boldsymbol{x}(t) + \boldsymbol{B}_2(u(t) + h),$$

where $h = -f_c + mlx_2^2\sin(x_1)$. Local linear models are constructed using the procedure described in the previous section, and in particular relations (15.25) and (15.38). We have

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0\\ \frac{g}{4l/3-aml} & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ -\frac{amg}{4/3-am} & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_1 = \begin{bmatrix} 0\\ -\frac{a}{4l/3-aml}\\ 0\\ \frac{4a/3}{4/3-ma} \end{bmatrix}.$$

The matrices $\boldsymbol{A}_2$ and $\boldsymbol{B}_2$ of the local model corresponding to the second rule are:

$$\boldsymbol{A}_2 = \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ \frac{g\sin(x_1)}{4l/3 - mla\cos^2(x_1)}\left(\frac{1}{x_1}\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-mag\sin(2x_1)/2}{4/3 - ma\cos^2(x_1)}\left(\frac{1}{x_1}\right) & 0 & 0 & 0 \end{array}\right]_{|x_1=\pm\pi/4} \quad \text{and} \quad \boldsymbol{B}_2 = \left[\begin{array}{c} 0 \\ \frac{-a\cos(x_1)}{4l/3 - mla\cos^2(x_1)} \\ 0 \\ \frac{4a/3}{4/3 - ma\cos^2(x_1)} \end{array}\right]_{|x_1=\pm\pi/4}.$$

The parameter values are the same as in Example 15.5. We use weighting functions of the form:

$$w_1(x_1) = \frac{1 - 1/\left(1 + \exp(-14(x_1 - \pi/8))\right)}{1 + \exp(-14(x_1 + \pi/8))} \quad \text{and} \quad w_2(x_1) = 1 - w_1(x_1).$$

The corresponding fuzzy system model is

$$\dot{\boldsymbol{x}} = (\alpha_1 \boldsymbol{A}_1 + \alpha_2 \boldsymbol{A}_2)\,\boldsymbol{x} + (\alpha_1 \boldsymbol{B}_1 + \alpha_2 \boldsymbol{B}_2)\,(u - \mu_c \text{sign}(x_4) + mlx_2^2 \sin(x_1)), \tag{15.67}$$

where $\alpha_i = w_i$, $i = 1, 2$. Note that $w_1 + w_2 = 1$ for all t. The friction coefficient $\mu_c = 0.005$. We then proceed to construct a stabilizing controller. We first construct

$$u_1 = -\left(\alpha_1 \boldsymbol{K}_1 + \alpha_2 \boldsymbol{K}_2\right)\boldsymbol{x}$$

such that the system

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{2}\sum_{j=1}^{2} \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j\right)\boldsymbol{x} \tag{15.68}$$

is asymptotically stable in the large. To test our stabilization algorithm, we randomly chose the set of desired poles for $(\boldsymbol{A}_1 - \boldsymbol{B}_1\boldsymbol{K}_1)$ to be in the interval $[-4, -1]$. The set of desired poles is

$$\{-1.0970, -2.1263, -2.5553, -3.9090\}.$$

The desired poles for $(\boldsymbol{A}_2 - \boldsymbol{B}_2\boldsymbol{K}_2)$ were also chosen in random fashion to be in the interval $[-4, -1]$. In this case, we generated the set of poles of the form,

$$\{-1.5794, -1.6857, -2.7908, -3.6895\}.$$

For this choice of the closed-loop poles of the local models, we obtain the following feedback gain matrices:

$$\boldsymbol{K}_1 = \left[\begin{array}{cccc} -294.8766 & -73.1208 & -13.4726 & -27.3362 \end{array}\right],$$

and

$$\boldsymbol{K}_2 = \left[\begin{array}{cccc} -440.3915 & -118.5144 & -19.1611 & -35.5575 \end{array}\right].$$

Let

$$\boldsymbol{G}_{12} = \boldsymbol{A}_1 - \boldsymbol{B}_1\boldsymbol{K}_2 + \boldsymbol{A}_2 - \boldsymbol{B}_2\boldsymbol{K}_1,$$

and

$$\boldsymbol{P} = \begin{bmatrix} 54.9580 & 15.6219 & 6.9389 & 12.1165 \\ 15.6219 & 4.5429 & 2.1011 & 3.5488 \\ 6.9389 & 2.1011 & 1.3972 & 1.7978 \\ 12.1165 & 3.5488 & 1.7978 & 2.9375 \end{bmatrix}.$$

Then, it is easy to check that sufficient conditions, given by (15.46) and (15.47), for asymptotic stability in the large of system (15.68) are satisfied. This completes the design of u_1.

We now construct u_2. We first use (15.59) to construct

$$\mu = \left(\alpha_1\boldsymbol{B}_1^\top\boldsymbol{P} + \alpha_2\boldsymbol{B}_2^\top\boldsymbol{P}\right)\boldsymbol{x}.$$

In this example, the number of inputs is $m = 1$, and hence (15.63) reduces to

$$u_2 = -\eta_q \operatorname{sign}(\mu),$$

where we can take $\eta_q = |h|$. We combine u_1 and u_2 to obtain the following stabilizing controller,

$$u = u_1 + u_2 = -\left(\alpha_1\boldsymbol{K}_1 + \alpha_2\boldsymbol{K}_2\right)\boldsymbol{x} - \eta_q \operatorname{sign}(\mu). \tag{15.69}$$

The above controller globally stabilizes the design model. In our simulations we connected the controller to the truth model given by (2.79). Typical responses are shown in Figure 15.30 and Figure 15.31. The controller locally stabilizes the truth model. We observed that when $|x_1(0)| > 1.3$ rads, and $x_2(0) = x_3(0) = x_4(0) = 0$, then the equilibrium state $\boldsymbol{x} = \boldsymbol{0}$ of the closed-loop system consisting of the truth model driven by (15.69) becomes unstable.

We summarize this section as follows. A common approach to the control system design is to use a design model to construct a controller. The design model is a simplified version of the truth model and usually neglects high frequency dynamics of the plant. Thus, the designer has to cope with an incomplete mathematical model of the process to be controlled. To account for modeling uncertainties, we used a two-term controller. The first term of the controller stabilized the nominal plant, that is, the plant model that does not include modeling uncertainties. The role of the second term of the controller was to reject modeling uncertainties. In both cases fuzzy state-feedback control strategies were used. In order to be able to apply the proposed controllers in real-life systems, we have to account for limited authority of actuators, unmatched uncertainties, as well as inaccessibility of state variables.

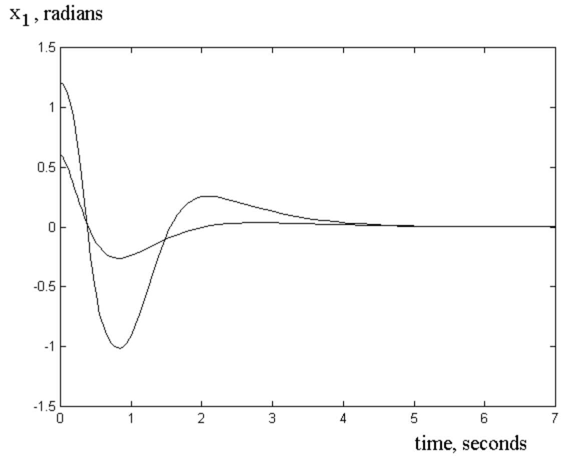


Figure 15.30: Plots of x_1 versus time for two different initial values of x_1: $x_1(0) = 1.2$ and $x_1(0) = 0.6$. In both cases $x_2(0) = x_3(0) = x_4(0) = 0$.

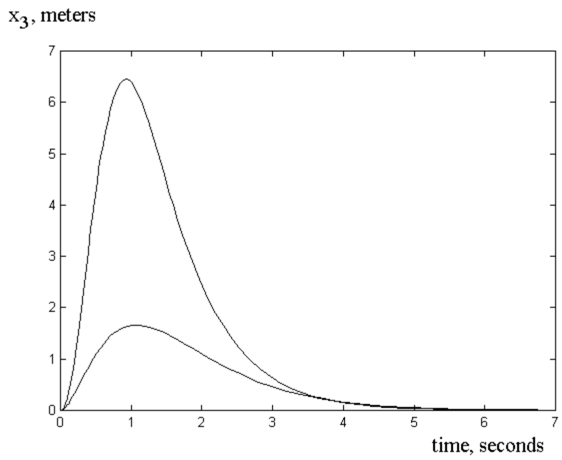


Figure 15.31: Plots of x_3 versus time for two different initial values of x_1: $x_1(0) = 1.2$ and $x_1(0) = 0.6$. In both cases $x_2(0) = x_3(0) = x_4(0) = 0$.

15.5 Stability of Discrete Fuzzy Models

We analyze the model of the form

$$\boldsymbol{x}[k+1]=\sum_{i=1}^{r} w_i(\boldsymbol{x}[k])\boldsymbol{A}_i\boldsymbol{x}[k], \tag{15.70}$$

where

$$w_i(\boldsymbol{x}[k])\geq 0 \quad \text{and} \quad \sum_{i=1}^{r} w_i(\boldsymbol{x}[k])=1.$$

To proceed, we need the following notation. Given two $n\times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Then, $\boldsymbol{A}<\boldsymbol{B}$ means that

$$\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}<\boldsymbol{x}^\top\boldsymbol{B}\boldsymbol{x} \quad \text{for all} \quad \boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{x}\neq\boldsymbol{0},$$

that is, the quadratic form $\boldsymbol{x}^\top(\boldsymbol{A}-\boldsymbol{B})\boldsymbol{x}$ is negative definite.

We prove the following matrix inequality lemma that is useful in the stability analysis of discrete fuzzy models.

Lemma 15.1 *Let* $\boldsymbol{A},\boldsymbol{B},\boldsymbol{P}\in\mathbb{R}^{n\times n}$. *If* $\boldsymbol{P}=\boldsymbol{P}^\top>0$ *such that*

$$\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{A}-\boldsymbol{P}<0 \quad \textit{and} \quad \boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{B}-\boldsymbol{P}<0,$$

then

$$\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{B}+\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{A}-2\boldsymbol{P}<0.$$

Proof We first represent $\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{B}+\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{A}-2\boldsymbol{P}$ as

$$\begin{aligned}\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{B}+\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{A}-2\boldsymbol{P} &= -(\boldsymbol{A}-\boldsymbol{B})^\top\boldsymbol{P}(\boldsymbol{A}-\boldsymbol{B})+\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{A}+\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{B}-2\boldsymbol{P}\\ &= -(\boldsymbol{A}-\boldsymbol{B})^\top\boldsymbol{P}(\boldsymbol{A}-\boldsymbol{B})+(\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{A}-\boldsymbol{P})\\ &\quad +(\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{B}-\boldsymbol{P}).\end{aligned}$$

The first term, $-(\boldsymbol{A}-\boldsymbol{B})^\top\boldsymbol{P}(\boldsymbol{A}-\boldsymbol{B})$, is non-positive definite because by assumption $\boldsymbol{P}=\boldsymbol{P}^\top>0$, while the second and the third terms are negative definite by assumption. Hence,

$$\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{B}+\boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{A}-2\boldsymbol{P}<0,$$

and the lemma is proven.

□

Using Lemma 15.1, we now prove a sufficient condition for global asymptotic stability of model (15.70).

Theorem 15.5 *The equilibrium state, $\boldsymbol{x}_e = \mathbf{0}$, of model (15.70) is globally asymptotically stable if there exists a positive definite matrix $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ such that for $i = 1, 2, \ldots, r$,*

$$\boldsymbol{A}_i^\top \boldsymbol{P} \boldsymbol{A}_i - \boldsymbol{P} < 0. \tag{15.71}$$

Proof Suppose that there exists a symmetric positive definite matrix $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ such that for $i = 1, 2, \ldots, r$ inequalities given by (15.71) are satisfied. Consider then the Lyapunov function candidate

$$V(\boldsymbol{x}[k]) = \boldsymbol{x}[k]^\top \boldsymbol{P} \boldsymbol{x}[k]. \tag{15.72}$$

To prove the theorem, it is enough to show that

$$\Delta V(\boldsymbol{x}[k]) < 0.$$

We have

$$\begin{aligned} \Delta V(\boldsymbol{x}[k]) &= V(\boldsymbol{x}[k+1]) - V(\boldsymbol{x}[k]) \\ &= \boldsymbol{x}[k+1]^\top \boldsymbol{P} \boldsymbol{x}[k+1] - \boldsymbol{x}[k]^\top \boldsymbol{P} \boldsymbol{x}[k] \\ &= \left(\sum_{i=1}^{r} w_i \boldsymbol{x}^\top \boldsymbol{A}_i^\top \right) \boldsymbol{P} \left(\sum_{i=1}^{r} w_i \boldsymbol{A}_i \boldsymbol{x} \right) - \boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{x} \\ &= \boldsymbol{x}^\top \left(\left(\sum_{i=1}^{r} w_i \boldsymbol{A}_i^\top \right) \boldsymbol{P} \left(\sum_{i=1}^{r} w_i \boldsymbol{A}_i \right) - \boldsymbol{P} \right) \boldsymbol{x}. \end{aligned} \tag{15.73}$$

To proceed, we represent (15.73) in an equivalent form as

$$\Delta V(\boldsymbol{x}) = \boldsymbol{x}^\top \left(\left(\sum_{i=1}^{r} w_i \boldsymbol{A}_i^\top \right) \boldsymbol{P} \left(\sum_{j=1}^{r} w_j \boldsymbol{A}_j \right) - \boldsymbol{P} \right) \boldsymbol{x}. \tag{15.74}$$

By assumption

$$\sum_{i=1}^{r} w_i = 1,$$

and hence

$$\sum_{i=1}^{r} w_i \sum_{j=1}^{r} w_j = \sum_{i=1}^{r} \sum_{j=1}^{r} w_i w_j = 1. \tag{15.75}$$

Substituting (15.75) into (15.74) yields

$$\begin{aligned} \Delta V(\boldsymbol{x}) &= \boldsymbol{x}^\top \left(\left(\sum_{i=1}^{r} w_i \boldsymbol{A}_i^\top \right) \boldsymbol{P} \left(\sum_{j=1}^{r} w_j \boldsymbol{A}_j \right) - \sum_{i=1}^{r} \sum_{j=1}^{r} w_i w_j \boldsymbol{P} \right) \boldsymbol{x} \\ &= \boldsymbol{x}^\top \left(\sum_{i=1}^{r} \sum_{j=1}^{r} w_i w_j \left(\boldsymbol{A}_i^\top \boldsymbol{P} \boldsymbol{A}_j - \boldsymbol{P} \right) \right) \boldsymbol{x}. \end{aligned} \tag{15.76}$$

We rearrange (15.76) to get

$$\begin{aligned}\Delta V(\boldsymbol{x}[k]) &= \sum_{i=1}^{r} w_i^2 \boldsymbol{x}^\top \left(\boldsymbol{A}_i^\top \boldsymbol{P}\boldsymbol{A}_i - \boldsymbol{P}\right)\boldsymbol{x} + \sum_{i=1}^{r}\sum_{\substack{j=1\\ j\neq i}}^{r} w_i w_j \boldsymbol{x}^\top \left(\boldsymbol{A}_i^\top \boldsymbol{P}\boldsymbol{A}_j - \boldsymbol{P}\right)\boldsymbol{x}\\ &= \sum_{i=1}^{r} w_i^2 \boldsymbol{x}^\top \left(\boldsymbol{A}_i^\top \boldsymbol{P}\boldsymbol{A}_i - \boldsymbol{P}\right)\boldsymbol{x}\\ &\quad + \sum_{i=1}^{r}\sum_{j>i}^{r} w_i w_j \boldsymbol{x}^\top \left(\boldsymbol{A}_i^\top \boldsymbol{P}\boldsymbol{A}_j + \boldsymbol{A}_j^\top \boldsymbol{P}\boldsymbol{A}_i - 2\boldsymbol{P}\right)\boldsymbol{x}. \end{aligned} \tag{15.77}$$

The first term in (15.77) is negative definite because of the assumption and the fact that at least one of the weights, w_i, is positive because $w_i \geq 0$ and $\sum_{i=1}^{r} w_i = 1$. The second term in (15.77) is nonpositive definite by virtue of Lemma 15.1. Hence, $V = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}$ is a Lyapunov function for the fuzzy model (15.70), and by the Lyapunov theorem the equilibrium state $\boldsymbol{x}_e = \boldsymbol{0}$ of the model is globally asymptotically stable.

□

The above theorem gives no method for finding a common positive definite matrix $\boldsymbol{P}$ that satisfies r inequalities (15.71). All of the matrices $\boldsymbol{A}_i$, $i = 1, 2, \ldots, r$, are asymptotically stable if there exists a common positive definite $\boldsymbol{P}$ that satisfies (15.71). However, the asymptotic stability of all the matrices $\boldsymbol{A}_i$ is only necessary but not sufficient for the existence of a common symmetric positive definite $\boldsymbol{P}$ that satisfies (15.71). The following theorem provides a necessary, but not sufficient, condition for the existence of positive definite $\boldsymbol{P}$ satisfying (15.71).

Theorem 15.6 *Given r square $n \times n$ matrices $\boldsymbol{A}_i$ that are asymptotically stable and nonsingular. If there exists a symmetric positive definite matrix $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ such that for $i = 1, 2, \ldots, r$,*

$$\boldsymbol{A}_i^\top \boldsymbol{P}\boldsymbol{A}_i - \boldsymbol{P} < 0, \tag{15.78}$$

then for $i, j = 1, 2, \ldots, r$, the matrices $\boldsymbol{A}_i\boldsymbol{A}_j$ are asymptotically stable.

Proof By assumption, there exists a symmetric positive definite matrix $\boldsymbol{P}$ such that for $j = 1, 2, \ldots, r$,

$$\boldsymbol{A}_j^\top \boldsymbol{P}\boldsymbol{A}_j - \boldsymbol{P} < 0. \tag{15.79}$$

Premultiplying the above by $\left(\boldsymbol{A}_j^\top\right)^{-1}$ and postmultiplying by $\boldsymbol{A}_j^{-1}$ yields

$$\boldsymbol{P} - \left(\boldsymbol{A}_j^\top\right)^{-1} \boldsymbol{P}\boldsymbol{A}_j^{-1} < 0. \tag{15.80}$$

Hence,

$$P < \left(A_j^\top\right)^{-1} P A_j^{-1}. \tag{15.81}$$

On the other hand, using (15.78) we get

$$A_i^\top P A_i < P. \tag{15.82}$$

Combining (15.81) and (15.82), we obtain

$$A_i^\top P A_i < \left(A_j^\top\right)^{-1} P A_j^{-1}. \tag{15.83}$$

Premultiplying both sides of (15.83) by $A_j^\top$ and postmultiplying by A_j, we get, for $i, j = 1, 2, \ldots, r$,

$$\left(A_i A_j\right)^\top P \left(A_i A_j\right) - P < 0. \tag{15.84}$$

Therefore, for $i, j = 1, 2, \ldots, r$, the matrices $A_i A_j$ are asymptotically stable, which completes the proof.

□

The above theorem can be stated in an equivalent way as follows. Assume that for $i = 1, 2, \ldots, r$, the matrices A_i are asymptotically stable and nonsingular. If for some $i, j = 1, 2, \ldots, r$, the matrix $A_i A_j$ is unstable, then there does not exist a common positive definite P such that

$$A_i^\top P A_i - P < 0 \quad \text{for} \quad i = 1, 2, \ldots, r.$$

Example 15.9 Let

$$A_1 = \begin{bmatrix} 1 & -0.5 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} -1 & -0.5 \\ 1 & 0 \end{bmatrix}.$$

Both matrices, A_1 and A_2, are asymptotically stable. Indeed, the eigenvalues of A_1 are located at $\{0.5+\jmath 0.5, 0.5-\jmath 0.5\}$, while the eigenvalues of A_2 are located at $\{-0.5+\jmath 0.5, -0.5-\jmath 0.5\}$. However, the matrix $A_1 A_2$ has its eigenvalues located at $\{-1.866, -0.134\}$, and hence is unstable. Thus, it follows from Theorem 15.6, that there is no positive definite P such that

$$A_i^\top P A_i - P < 0 \quad \text{for} \quad i = 1, 2. \tag{15.85}$$

15.6 Fuzzy Observer

In this section, we are concerned with the problem of designing an observer using a TSK model of the plant. We first discuss the issue of constructing a TSK model using input-output data. To construct a TSK model, we need local description of the plant in terms of linear models,

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}_i\boldsymbol{x}(t) + \boldsymbol{B}_i\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}_i\boldsymbol{x}(t), \end{cases}$$

where $i = 1, 2, \ldots, r$, state vector $\boldsymbol{x}(t) \in \mathbb{R}^n$, the control input $\boldsymbol{u}(t) \in \mathbb{R}^m$, the output $\boldsymbol{y}(t) \in \mathbb{R}^p$, and the matrices $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, and $\boldsymbol{C}_i$ are of appropriate dimensions. The above information is then fused with the available IF-THEN rules, where the i-th rule can have the form:

$$\text{Rule i}: \text{ IF } z_1(t) \text{ is } F_1^i \text{ AND } \cdots \text{ AND } z_q(t) \text{ is } F_q^i,$$

$$\text{THEN} \begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}_i\boldsymbol{x}(t) + \boldsymbol{B}_i\boldsymbol{u}(t) \\ \boldsymbol{y} &= \boldsymbol{C}_i\boldsymbol{x}(t), \end{cases}$$

where F_j^i, $j = 1, 2, \ldots, q$, is the j-th fuzzy set of the i-th rule, and $z_1(t), \ldots, z_q(t)$ are measurable system variables. Let $\mu_j^i(z_j)$ be the membership function of the fuzzy set F_j^i, and let

$$w^i = w^i(\boldsymbol{z}) = \prod_{j=1}^{n} \mu_j^i(z_j),$$

where r is the number of fuzzy rules. The resulting fuzzy system model is the weighted average of the local models and has the form of (15.6), that is,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \frac{\sum_{i=1}^r w^i\left(\boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{B}_i\boldsymbol{u}\right)}{\sum_{i=1}^r w^i} \\ &= \sum_{i=1}^r \alpha_i\left(\boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{B}_i\boldsymbol{u}\right) \\ &= \left(\sum_{i=1}^r \alpha_i\boldsymbol{A}_i\right)\boldsymbol{x} + \left(\sum_{i=1}^r \alpha_i\boldsymbol{B}_i\right)\boldsymbol{u} \\ &= \boldsymbol{A}(\boldsymbol{\alpha})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\alpha})\boldsymbol{u}, \end{aligned}$$

where for $i = 1, 2, \ldots, r$,

$$\alpha_i = \frac{w^i}{\sum_{i=1}^r w^i}.$$

Similarly,

$$\begin{aligned}\boldsymbol{y} &= \frac{\sum_{i=1}^{r} w^i \boldsymbol{C}_i \boldsymbol{x}}{\sum_{i=1}^{r} w^i} \\ &= \left(\sum_{i=1}^{r} \alpha_i \boldsymbol{C}_i \right) \boldsymbol{x} \\ &= \boldsymbol{C}(\boldsymbol{\alpha}) \boldsymbol{x}. \end{aligned} \tag{15.86}$$

The coefficients α_is satisfy (15.8), that is,

$$\alpha_i \geq 0, \quad \sum_{i=1}^{r} \alpha_i = 1 \quad \text{and} \quad \boldsymbol{\alpha} = \left[\begin{array}{cccc} \alpha_1 & \alpha_2 & \cdots & \alpha_r \end{array} \right]^\top \in [0,1]^r.$$

A fuzzy observer for the system modeled by (15.6) and (15.86) can have the form

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}(\boldsymbol{\alpha}) \tilde{\boldsymbol{x}} + \boldsymbol{B}(\boldsymbol{\alpha}) \boldsymbol{u} + \boldsymbol{L}(\boldsymbol{\alpha}) \left(\boldsymbol{y} - \tilde{\boldsymbol{y}} \right), \tag{15.87}$$

where $\tilde{\boldsymbol{x}}$ is the estimate of the plant state $\boldsymbol{x}$ and

$$\boldsymbol{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{r} \alpha_i \boldsymbol{L}_i.$$

The observer (15.87) can be represented equivalently as

$$\boxed{\dot{\tilde{\boldsymbol{x}}} = \sum_{i=1}^{r} \alpha_i \boldsymbol{A}_i \tilde{\boldsymbol{x}} + \sum_{i=1}^{r} \alpha_i \boldsymbol{B}_i \boldsymbol{u} + \sum_{i=1}^{r} \alpha_i \boldsymbol{L}_i \left(\boldsymbol{y} - \tilde{\boldsymbol{y}} \right)} \tag{15.88}$$

We note that the observer (15.88) can be viewed as a result of implementing the following r fuzzy rules:

$$\text{Rule i}: \text{ IF } z_1(t) \text{ is } F_1^i \text{ AND } \cdots \text{ AND } z_q(t) \text{ is } F_q^i,$$

$$\text{THEN} \left\{ \begin{array}{rcl} \dot{\tilde{\boldsymbol{x}}} &=& \boldsymbol{A}_i \tilde{\boldsymbol{x}} + \boldsymbol{B}_i \boldsymbol{u} + \boldsymbol{L}_i \left(\boldsymbol{y} - \tilde{\boldsymbol{y}} \right) \\ \tilde{\boldsymbol{y}} &=& \boldsymbol{C}_i \tilde{\boldsymbol{x}}, \end{array} \right.$$

where, as before, $z_1(t), \ldots, z_q(t)$ are measurable system variables. We use the state estimate in the controller implementation to get

$$\boldsymbol{u} = -\sum_{j=1}^{r} \alpha_j \boldsymbol{K}_j \tilde{\boldsymbol{x}} + \boldsymbol{v}. \tag{15.89}$$

Substituting the above into the plant model yields

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{r} \alpha_i \boldsymbol{A}_i \boldsymbol{x} - \sum_{i=1}^{r} \sum_{j=1}^{r} \alpha_i \alpha_j \boldsymbol{B}_i \boldsymbol{K}_j \tilde{\boldsymbol{x}} + \sum_{i=1}^{r} \alpha_i \boldsymbol{B}_i \boldsymbol{v}. \tag{15.90}$$

Substituting (15.86) and (15.89) into the observer equation (15.88), we obtain

$$\begin{aligned}\dot{\tilde{\boldsymbol{x}}} &= \sum_{i=1}^{r}\alpha_i\boldsymbol{A}_i\tilde{\boldsymbol{x}} - \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\boldsymbol{B}_i\boldsymbol{K}_j\tilde{\boldsymbol{x}} + \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\boldsymbol{L}_i\boldsymbol{C}_j\left(\boldsymbol{x}-\tilde{\boldsymbol{x}}\right) + \sum_{i=1}^{r}\alpha_i\boldsymbol{B}_i\boldsymbol{v}\\ &= \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\left(\boldsymbol{A}_i-\boldsymbol{B}_i\boldsymbol{K}_j\right)\tilde{\boldsymbol{x}} + \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\boldsymbol{L}_i\boldsymbol{C}_j\left(\boldsymbol{x}-\tilde{\boldsymbol{x}}\right) + \sum_{i=1}^{r}\alpha_i\boldsymbol{B}_i\boldsymbol{v}. \end{aligned} \tag{15.91}$$

Equations (15.90) and (15.91) constitute the closed-loop system driven by the combined fuzzy controller-observer compensator. We represent the above two equations modeling the closed-loop system in matrix form as

$$\begin{aligned}\begin{bmatrix}\dot{\boldsymbol{x}}\\ \dot{\tilde{\boldsymbol{x}}}\end{bmatrix} &= \begin{bmatrix}\sum_{i=1}^{r}\alpha_i\boldsymbol{A}_i & -\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\boldsymbol{B}_i\boldsymbol{K}_j\\ \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\boldsymbol{L}_i\boldsymbol{C}_j & \sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\left(\boldsymbol{A}_i-\boldsymbol{B}_i\boldsymbol{K}_j-\boldsymbol{L}_i\boldsymbol{C}_j\right)\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ \tilde{\boldsymbol{x}}\end{bmatrix}\\ &\quad + \begin{bmatrix}\sum_{i=1}^{r}\alpha_i\boldsymbol{B}_i\\ \sum_{i=1}^{r}\alpha_i\boldsymbol{B}_i\end{bmatrix}\boldsymbol{v} \end{aligned} \tag{15.92}$$

$$\boldsymbol{y} = \begin{bmatrix}\sum_{i=1}^{r}\alpha_i\boldsymbol{C}_i & \boldsymbol{O}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ \tilde{\boldsymbol{x}}\end{bmatrix}. \tag{15.93}$$

15.6.1 The Comparison Method for Linear Systems

We present the comparison method for linear systems. that is used by us to investigate stability properties of the closed-loop system, modeled by (15.92) and (15.93), consisting of a fuzzy model of a given plant and a fuzzy controller-estimator compensator.

Let $\boldsymbol{A}$ be a constant $n \times n$ matrix and $e^{\boldsymbol{A}t}$ denote the matrix exponential. We give a necessary and sufficient condition for the matrix exponential to have all its elements non-negative for $t \geq 0$. This result appears in [27, page 171].

Lemma 15.2 *A necessary and sufficient condition that all elements of* $e^{\boldsymbol{A}t}$ *be non-negative for* $t \geq 0$ *is that*

$$a_{ij} \geq 0 \quad i \neq j. \tag{15.94}$$

Proof We follow the arguments of Bellman [27, page 172], see also [26]. We first prove the necessary condition. Because

$$e^{\boldsymbol{A}t} = \boldsymbol{I}_n + \boldsymbol{A}t + \frac{1}{2}\boldsymbol{A}^2t^2 + \cdots$$

it is clear that (15.94) is necessary for the result to hold for small t, which establishes the necessity condition

To prove the sufficiency part, let c_1 be a scalar so that all the elements of $\boldsymbol{A} + c_1\boldsymbol{I}_n$ are non-negative. Then, obviously, all the elements of $e^{(\boldsymbol{A}+c_1\boldsymbol{I}_n)t}$ are non-negative. Also, the elements of $e^{-c_1\boldsymbol{I}_n t}$ are non-negative because exponentials are always non-negative. The matrices $\boldsymbol{A} + c_1\boldsymbol{I}_n$ and $-c_1\boldsymbol{I}_n$ commute. Hence,

$$\begin{aligned} e^{\boldsymbol{A}t} &= e^{(\boldsymbol{A}+c_1\boldsymbol{I}_n)t - c_1\boldsymbol{I}_n t} \\ &= e^{(\boldsymbol{A}+c_1\boldsymbol{I}_n)t} e^{-c_1\boldsymbol{I}_n t}, \end{aligned}$$

which establishes the desired non-negativity condition and the sufficiency condition.

$\square$

Using the above lemma, we prove the following lemma due to Bellman [28]. We use the following notation: $\boldsymbol{x} \le \boldsymbol{y}$ implies component-by-component majorization, that is, $x_i \le y_i$ for $i = 1, 2, \ldots, n$, where x_i and y_i are the i-th components of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

Lemma 15.3 *If $a_{ij} \ge 0$, $i \ne j$, then for $t \ge 0$,*

$$\frac{d\boldsymbol{x}}{dt} \le \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{x}(0) = \boldsymbol{c}, \tag{15.95}$$

implies $\boldsymbol{x} \le \boldsymbol{y}$, where

$$\frac{d\boldsymbol{y}}{dt} = \boldsymbol{A}\boldsymbol{y}, \quad \boldsymbol{y}(0) = \boldsymbol{c}. \tag{15.96}$$

Proof We can represent (15.95) as

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{A}\boldsymbol{x}(t) - \boldsymbol{f}(t), \quad \boldsymbol{x}(0) = \boldsymbol{c}, \tag{15.97}$$

where $\boldsymbol{f}(t)$ is a vector-valued function whose components are non-negative for all $t \ge 0$. The solution of (15.97) is represented as

$$\begin{aligned} \boldsymbol{x}(t) &= e^{\boldsymbol{A}t}\boldsymbol{c} - \int_0^t e^{\boldsymbol{A}(t-s)}\boldsymbol{f}(s)ds \\ &= \boldsymbol{y}(t) - \int_0^t e^{\boldsymbol{A}(t-s)}\boldsymbol{f}(s)ds. \end{aligned} \tag{15.98}$$

In the above, s is the integration variable and $t \ge 0$. Hence, $t \ge s \ge 0$. Using Lemma 15.2, we conclude that the components of the integrand of the second term in (15.98) are all non-negative for $t \ge s \ge 0$. Therefore,

$$\int_0^t e^{\boldsymbol{A}(t-s)}\boldsymbol{f}(s)ds \ge \boldsymbol{0}, \quad \text{for} \quad t \ge 0,$$

which implies

$$\boldsymbol{x} \leq \boldsymbol{y}.$$

The proof is complete.

□

Equation (15.96) is referred to as the *equation of comparison* for (15.95).

15.6.2 Stability Analysis of the Closed-Loop System

We now perform a stability analysis of the closed-loop system given by (15.92) and (15.93). We first apply a state variable transformation to bring the system into a form convenient for further analysis. The transformation has the form,

$$\begin{bmatrix} \boldsymbol{x} \\ \tilde{\boldsymbol{x}} - \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{O} \\ -\boldsymbol{I}_n & \boldsymbol{I}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \tilde{\boldsymbol{x}} \end{bmatrix}. \tag{15.99}$$

The closed-loop system model in the new coordinates is

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\tilde{\boldsymbol{x}}} - \dot{\boldsymbol{x}} \end{bmatrix} &= \begin{bmatrix} \sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j\right) & -\sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j \boldsymbol{B}_i \boldsymbol{K}_j \\ \boldsymbol{O} & \sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{L}_i \boldsymbol{C}_j\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \tilde{\boldsymbol{x}} - \boldsymbol{x} \end{bmatrix} \\ &\quad + \begin{bmatrix} \sum_{i=1}^r \alpha_i \boldsymbol{B}_i \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{v} \qquad (15.100) \\ \boldsymbol{y} &= \begin{bmatrix} \sum_{i=1}^r \alpha_i \boldsymbol{C}_i & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \tilde{\boldsymbol{x}} - \boldsymbol{x} \end{bmatrix}. \qquad (15.101) \end{aligned}$$

Let $\boldsymbol{e} = \tilde{\boldsymbol{x}} - \boldsymbol{x}$. We now present a sufficient condition for the unforced closed-loop system to be globally uniformly asymptotically stable, that is, we will give a condition for uniform asymptotic stability in the large of the system

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{e}} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j\right) & -\sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j \boldsymbol{B}_i \boldsymbol{K}_j \\ \boldsymbol{O} & \sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{L}_i \boldsymbol{C}_j\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{e} \end{bmatrix}. \tag{15.102}$$

This condition was proposed by Ma, Sun, and He [225].

Theorem 15.7 *If there exist two scalar functions $V(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ and $\tilde{V}(\boldsymbol{e}) : \mathbb{R}^n \to \mathbb{R}$ and positive real numbers γ_i, $\tilde{\gamma}_i$, $i = 1, \ldots, 4$ such that*

$$\gamma_1 \|\boldsymbol{x}\|^2 \leq V(\boldsymbol{x}) \leq \gamma_2 \|\boldsymbol{x}\|^2, \quad \tilde{\gamma}_1 \|\boldsymbol{e}\|^2 \leq \tilde{V}(\boldsymbol{e}) \leq \tilde{\gamma}_2 \|\boldsymbol{e}\|^2, \tag{15.103}$$

$$\left.\begin{array}{l} \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \sum_{i=1}^{r} \sum_{j=1}^{r} \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j\right) \boldsymbol{x} \leq -\gamma_3 \|\boldsymbol{x}\|^2 \\ \frac{\partial \tilde{V}(\boldsymbol{e})}{\partial \boldsymbol{e}} \sum_{i=1}^{r} \sum_{j=1}^{r} \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{L}_i \boldsymbol{C}_j\right) \boldsymbol{e} \leq -\tilde{\gamma}_3 \|\boldsymbol{e}\|^2, \end{array}\right\} \tag{15.104}$$

and

$$\left\| \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\| \leq \gamma_4 \|\boldsymbol{x}\|, \qquad \left\| \frac{\partial \tilde{V}(\boldsymbol{e})}{\partial \boldsymbol{e}} \right\| \leq \tilde{\gamma}_4 \|\boldsymbol{e}\|, \tag{15.105}$$

then the system (15.102) is globally uniformly asymptotically stable.

Proof We evaluate the time derivative of V on the trajectories of the system (15.102),

$$\begin{aligned} \dot{V}(\boldsymbol{x}) &= \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} \\ &= \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \sum_{i=1}^{r} \sum_{j=1}^{r} \alpha_i \alpha_j \left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_j\right) \boldsymbol{x} - \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \sum_{i=1}^{r} \sum_{j=1}^{r} \alpha_i \alpha_j \boldsymbol{B}_i \boldsymbol{K}_j \boldsymbol{e} \\ &\leq -\gamma_3 \|\boldsymbol{x}\|^2 + \left\| \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\| \sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| \, \|\boldsymbol{e}\| \\ &\leq -\frac{\gamma_3}{2} \|\boldsymbol{x}\|^2 - \frac{\gamma_3}{2} \|\boldsymbol{x}\|^2 + \gamma_4 \sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| \, \|\boldsymbol{e}\| \|\boldsymbol{x}\|. \end{aligned} \tag{15.106}$$

To proceed, we observe that

$$\frac{\gamma_3}{2} \|\boldsymbol{x}\|^2 - \gamma_4 \sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| \, \|\boldsymbol{e}\| \|\boldsymbol{x}\| + \frac{\gamma_4^2}{2\gamma_3} \left(\sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| \right)^2 \|\boldsymbol{e}\|^2 \geq 0. \tag{15.107}$$

Indeed, we can represent the left-hand side of the above as

$$\begin{bmatrix} \|\boldsymbol{x}\| & \|\boldsymbol{e}\| \end{bmatrix} \begin{bmatrix} \frac{\gamma_3}{2} & -\frac{\gamma_4}{2} \sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| \\ -\frac{\gamma_4}{2} \sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| & \frac{\gamma_4^2}{2\gamma_3} \left(\sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| \right)^2 \end{bmatrix} \begin{bmatrix} \|\boldsymbol{x}\| \\ \|\boldsymbol{e}\| \end{bmatrix}. \tag{15.108}$$

It is easy to check that the above quadratic form is positive semi-definite—all three principle minors are non-negative. We use this fact and (15.103) to further evaluate (15.106) as

$$\begin{aligned} \dot{V}(\boldsymbol{x}) &\leq -\frac{\gamma_3}{2} \|\boldsymbol{x}\|^2 + \frac{\gamma_4^2}{2\gamma_3} \left(\sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| \right)^2 \|\boldsymbol{e}\|^2 \\ &\leq -\frac{\gamma_3}{2\gamma_2} V(\boldsymbol{x}) + \frac{\gamma_4^2}{2\gamma_3 \tilde{\gamma}_1} \left(\sum_{i=1}^{r} \sum_{j=1}^{r} \|\boldsymbol{B}_i \boldsymbol{K}_j\| \right)^2 \tilde{V}(\boldsymbol{e}). \end{aligned} \tag{15.109}$$

Let $a = \left(\sum_{i=1}^{r}\sum_{j=1}^{r}\|\boldsymbol{B}_i\boldsymbol{K}_j\|\right)^2$. Then, (15.109) can be written as

$$\dot{V}(\boldsymbol{x}) \leq -\frac{\gamma_3}{2\gamma_2}V(\boldsymbol{x}) + \frac{a\gamma_4^2}{2\gamma_3\tilde{\gamma}_1}\tilde{V}(\boldsymbol{e}). \tag{15.110}$$

Evaluating the time derivative of $\tilde{V}(\boldsymbol{e})$ on the trajectories of (15.102) gives

$$\begin{aligned}\dot{\tilde{V}}(\boldsymbol{e}) &= \frac{\partial\tilde{V}(\boldsymbol{e})}{\partial\boldsymbol{e}}\dot{\boldsymbol{e}} \\ &\leq \frac{\partial\tilde{V}(\boldsymbol{e})}{\partial\boldsymbol{e}}\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j\left(\boldsymbol{A}_i - \boldsymbol{L}_i\boldsymbol{C}_j\right)\boldsymbol{e} \\ &\leq -\tilde{\gamma}_3\|\boldsymbol{e}\|^2 \\ &\leq -\frac{\tilde{\gamma}_3}{\tilde{\gamma}_2}\tilde{V}(\boldsymbol{e}).\end{aligned} \tag{15.111}$$

Combining (15.110) and (15.111) into a single matrix differential inequality of the form yields

$$\begin{aligned}\begin{bmatrix}\dot{V}(\boldsymbol{x}) \\ \dot{\tilde{V}}(\boldsymbol{e})\end{bmatrix} &\leq \begin{bmatrix}-\frac{\gamma_3}{2\gamma_2} & \frac{a\gamma_4^2}{2\gamma_3\tilde{\gamma}_1} \\ 0 & -\frac{\tilde{\gamma}_3}{\tilde{\gamma}_2}\end{bmatrix}\begin{bmatrix}V(\boldsymbol{x}) \\ \tilde{V}(\boldsymbol{e})\end{bmatrix} \\ &= \boldsymbol{A}\begin{bmatrix}V(\boldsymbol{x}) \\ \tilde{V}(\boldsymbol{e})\end{bmatrix}.\end{aligned} \tag{15.112}$$

We then refer to Lemma 15.3 to conclude that the system (15.102) is globally uniformly asymptotically stable. Indeed, because the equation of comparison for (15.112) is globally uniformly asymptotically stable, we have to have

$$V(\boldsymbol{x}) \to 0 \quad \text{and} \quad \tilde{V}(\boldsymbol{e}) \to 0 \quad \text{as} \quad t \to \infty.$$

Because V and $\tilde{V}$ are both positive definite decrescent and radially unbounded functions, the above can hold provided $\boldsymbol{x} \to \boldsymbol{0}$ and $\boldsymbol{e} \to \boldsymbol{0}$ uniformly with respect to t and irrespective of the initial conditions $\boldsymbol{x}(0)$ and $\boldsymbol{e}(0)$. Therefore, the closed-loop system is globally uniformly asymptotically stable.

□

15.7 Adaptive Fuzzy Control

In this Section, we analyze adaptive fuzzy robust tracking controllers for a class of uncertain dynamical systems. The controllers' construction and their analysis involves sliding modes.

15.7.1 Plant Model and Control Objective

We consider a class of plants modeled by the differential equation

$$x^{(n)} = f(\boldsymbol{x}) + g(\boldsymbol{x})\,u + \eta(t, \boldsymbol{x}), \tag{15.113}$$

where $\boldsymbol{x} = \begin{bmatrix} x & \dot{x} & \cdots & x^{(n-1)} \end{bmatrix}^\top$, and the functions $f = f(\boldsymbol{x})$ and $g = g(\boldsymbol{x})$ are unknown to us. The function $\eta = \eta(t, \boldsymbol{x})$ models the plant's disturbances. We assume that

$$|\eta(t, \boldsymbol{x})| \leq d, \tag{15.114}$$

where $d > 0$ is known to us. We further assume that the function $g(\boldsymbol{x})$ has a positive lower bound, that is, there exists a positive constant $\underline{g}$ such that $g(\boldsymbol{x}) \geq \underline{g} > 0$. Let

$$\boldsymbol{x}_d = \boldsymbol{x}_d(t) = \begin{bmatrix} x_d(t) & \dot{x}_d(t) & \cdots & x_d^{(n-1)}(t) \end{bmatrix}^\top$$

denote the desired state trajectory and let

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{x} - \boldsymbol{x}_d \\ &= \begin{bmatrix} x - x_d & \dot{x} - \dot{x}_d & \cdots & x^{(n-1)} - x_d^{(n-1)} \end{bmatrix}^\top \\ &= \begin{bmatrix} e & \dot{e} & \cdots & e^{(n-1)} \end{bmatrix}^\top. \end{aligned}$$

We wish to construct a controller u such that

$$\lim_{t \to \infty} \boldsymbol{e}(t) = \boldsymbol{0}.$$

We analyze two different fuzzy adaptive tracking controllers and show that the plant (15.113) driven by these controllers is stable and the adaptation parameters are bounded.

In the next subsection, we present background results used in the adaptive fuzzy controllers' construction.

15.7.2 Background Results

Fuzzy Logic System

Recall, from Section 15.2, that a fuzzy logic controller with q inputs, $v_1, v_2, \ldots, v_q$, and the center average defuzzifier can be represented as

$$u = \frac{\sum_{l=1}^{M} \prod_{j=1}^{q} \mu_{F_j^l}(v_j)\theta_l}{\sum_{l=1}^{M} \prod_{j=1}^{q} \mu_{F_j^l}(v_j)}, \tag{15.115}$$

where M is the number of fuzzy rules used to construct the controller, F_j^l is the j-th fuzzy set corresponding to the l-th fuzzy rule, and c_l can be taken as a centroid of the l-th fuzzy set corresponding to the controller's output, u. To proceed, we represent (15.115) as

$$u = \sum_{l=1}^{M} \left(\frac{\prod_{j=1}^{q} \mu_{F_j^l}(v_j)}{\sum_{l=1}^{M} \prod_{j=1}^{q} \mu_{F_j^l}(v_j)} \right) \theta_l. \tag{15.116}$$

Let

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_M \end{bmatrix}^\top$$

and

$$\boldsymbol{\xi} = \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_M \end{bmatrix}^\top = \begin{bmatrix} \frac{\prod_{j=1}^{q} \mu_{F_j^1}(v_j)}{\sum_{l=1}^{M} \prod_{j=1}^{q} \mu_{F_j^l}(v_j)} & \cdots & \frac{\prod_{j=1}^{q} \mu_{F_j^M}(v_j)}{\sum_{l=1}^{M} \prod_{j=1}^{q} \mu_{F_j^l}(v_j)} \end{bmatrix}^\top.$$

Then, we can represent (15.116) as

$$u = \boldsymbol{\theta}^\top \boldsymbol{\xi}, \tag{15.117}$$

where the vector $\boldsymbol{\xi}$ is referred to as the *regressor* vector in the control literature.

Example 15.10 Consider an uncertain dynamical system model,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (u + \zeta), \quad |\zeta| \le 5.$$

Our objective is to design a controller that contains an **adaptive fuzzy term** so that the state of the above system will track a reference signal generated by the reference model,

$$\begin{aligned} \dot{\boldsymbol{z}} &= \begin{bmatrix} 0 & 1 \\ -1 & -1.4 \end{bmatrix} \boldsymbol{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \\ &= \boldsymbol{A}_d \boldsymbol{z} + \boldsymbol{b}_d r, \end{aligned}$$

where r is a reference input such that $|r| \le 2$. We choose the controller settings and show that the tracking error of the closed-loop system driven by your controller asymptotically decays to zero. The solution to the Lyapunov matrix equation $\boldsymbol{A}_d^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_d = -\boldsymbol{I}_2$ is

$$\boldsymbol{P} = \begin{bmatrix} 1.4143 & 0.5000 \\ 0.5000 & 0.7143 \end{bmatrix}.$$

To proceed, we define the tracking error $\boldsymbol{e}_x(t) = \boldsymbol{x}(t) - \boldsymbol{z}(t)$. Note that $\boldsymbol{b} = \boldsymbol{b}_d$ and $\boldsymbol{A}_d = \boldsymbol{A} - \boldsymbol{b}\boldsymbol{k}$, where $\boldsymbol{k} = \begin{bmatrix} -1 & -0.4 \end{bmatrix}$. Then, the differential equation that describes the error dynamics is

$$\begin{aligned}\dot{\boldsymbol{e}}_x &= \dot{\boldsymbol{x}} - \dot{\boldsymbol{z}}\\ &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}(u+\zeta) - \boldsymbol{A}_d\boldsymbol{z} - \boldsymbol{b}r\\ &= (\boldsymbol{A}_d + \boldsymbol{b}\boldsymbol{k})\boldsymbol{x} + \boldsymbol{b}(u+\zeta) - \boldsymbol{A}_d\boldsymbol{z} - \boldsymbol{b}r\\ &= \boldsymbol{A}_d\boldsymbol{e}_x + \boldsymbol{b}(u+\zeta - r + \boldsymbol{k}\boldsymbol{x}).\end{aligned}$$

We propose the following control strategy,

$$u = -u_{fuzzy} - \eta\frac{\boldsymbol{b}^\top\boldsymbol{P}\boldsymbol{e}_x}{|\boldsymbol{b}^\top\boldsymbol{P}\boldsymbol{e}_x|} = -u_{fuzzy} - \eta\,\mathrm{sign}(0.5e_1 + 0.7143e_2),$$

where η will be determined later, and

$$u_{fuzzy} = \boldsymbol{\theta}^\top\boldsymbol{\xi},$$

where the vector $\boldsymbol{\xi}$ is bounded. The role of the fuzzy term is to minimize the effect of uncertainties modeled by the bounded function ζ. Let $\boldsymbol{\theta}^*$ denote the "optimal" weight vector such that

$$\zeta = \boldsymbol{\theta}^{*\top}\boldsymbol{\xi} + e, \tag{15.118}$$

where e is the approximation error. We assume that the approximation error is bounded by ρ, that is,

$$|e| \le \rho. \tag{15.119}$$

Consider the following Lyapunov function candidate,

$$V = \frac{1}{2}\boldsymbol{e}_x^\top\boldsymbol{P}\boldsymbol{e}_x + \frac{1}{2\alpha}\boldsymbol{\phi}^\top\boldsymbol{\phi},$$

where

$$\boldsymbol{\phi} = \boldsymbol{\theta} - \boldsymbol{\theta}^*. \tag{15.120}$$

We note that we are now in the augmented state space $(\boldsymbol{e}_x, \boldsymbol{\phi})$. The Lyapunov derivative is

$$\begin{aligned}\dot{V} &= \boldsymbol{e}_x^\top\boldsymbol{P}\dot{\boldsymbol{e}}_x + \frac{1}{\alpha}\boldsymbol{\phi}^\top\dot{\boldsymbol{\phi}}\\ &= \boldsymbol{e}_x^\top\boldsymbol{P}\left(\boldsymbol{A}_d\boldsymbol{e}_x + \boldsymbol{b}(u+\zeta - r + \boldsymbol{k}\boldsymbol{x})\right) + \frac{1}{\alpha}\boldsymbol{\phi}^\top\dot{\boldsymbol{\phi}}\\ &= -\frac{1}{2}\boldsymbol{e}_x^\top\boldsymbol{e}_x + \boldsymbol{e}_x^\top\boldsymbol{P}\boldsymbol{b}u + \boldsymbol{e}_x^\top\boldsymbol{P}\boldsymbol{b}\left(\zeta - r + \boldsymbol{k}\boldsymbol{x}\right) + \frac{1}{\alpha}\boldsymbol{\phi}^\top\dot{\boldsymbol{\phi}}.\end{aligned}$$

Substituting (15.118) and the control strategy into the above gives

$$\dot{V} = -\frac{1}{2}\|\boldsymbol{e}_x\|^2 - \eta\boldsymbol{e}_x^\top\boldsymbol{P}\boldsymbol{b}\frac{\boldsymbol{b}^\top\boldsymbol{P}\boldsymbol{e}_x}{|\boldsymbol{b}^\top\boldsymbol{P}\boldsymbol{e}_x|} - \boldsymbol{e}_x^\top\boldsymbol{P}\boldsymbol{b}\boldsymbol{\theta}^\top\boldsymbol{\xi} + \boldsymbol{e}_x^\top\boldsymbol{P}\boldsymbol{b}\left(\boldsymbol{\theta}^{*\top}\boldsymbol{\xi} + e - r + \boldsymbol{k}\boldsymbol{x}\right) + \frac{1}{\alpha}\boldsymbol{\phi}^\top\dot{\boldsymbol{\phi}}.$$

Substituting into the above (15.119) and (15.120) and performing manipulations yields

$$\begin{aligned}
\dot{V} &= -\frac{1}{2}\|\boldsymbol{e}_x\|^2 - \eta|\boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}| - \boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}\left(\boldsymbol{\theta}-\boldsymbol{\theta}^*\right)^\top \boldsymbol{\xi} + \boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}\left(e - r + \boldsymbol{k}\boldsymbol{x}\right) + \frac{1}{\alpha}\boldsymbol{\phi}^\top\dot{\boldsymbol{\phi}} \\
&\le -\frac{1}{2}\|\boldsymbol{e}_x\|^2 - \eta|\boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}| - \boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}\boldsymbol{\phi}^\top\boldsymbol{\xi} + \rho|\boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}| + |\boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}|\left(|r| + |\boldsymbol{k}\boldsymbol{x}|\right) + \frac{1}{\alpha}\boldsymbol{\phi}^\top\dot{\boldsymbol{\phi}} \\
&= -\frac{1}{2}\|\boldsymbol{e}_x\|^2 - |\boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}|\left(\eta - \rho - |r| - |\boldsymbol{k}\boldsymbol{x}|\right) - \boldsymbol{\phi}^\top\left(\left(\boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}\right)\boldsymbol{\xi} - \frac{1}{\alpha}\dot{\boldsymbol{\phi}}\right).
\end{aligned}$$

We select

$$\eta \ge \rho + |r| + |\boldsymbol{k}\boldsymbol{x}| \quad\text{and}\quad \dot{\boldsymbol{\phi}} = \dot{\boldsymbol{\theta}} = \alpha\left(\boldsymbol{e}_x^\top \boldsymbol{P}\boldsymbol{b}\right)\boldsymbol{\xi}$$

to obtain

$$\dot{V} \le -\frac{1}{2}\|\boldsymbol{e}_x\|^2. \tag{15.121}$$

Thus $\dot{V}$ is negative semi-definite in the augmented $(\boldsymbol{e}_x, \boldsymbol{\phi})$-space and therefore $\boldsymbol{e}_x$ as well as $\boldsymbol{\phi}$, and hence $\boldsymbol{\theta}$ are bounded. Hence the proposed controller is also bounded, which implies that $\dot{\boldsymbol{e}}_x$ is bounded. This in turn implies that

$$\frac{d}{dt}\|\boldsymbol{e}_x\|^2 = 2\boldsymbol{e}_x^\top \dot{\boldsymbol{e}}_x$$

is bounded and hence $\|\boldsymbol{e}_x\|^2$ is uniformly continuous. Integrating both sides of (15.121) gives

$$\int_0^t \dot{V}(\tau)d\tau = V(t) - V(0) \le \int_0^t \left(-\frac{1}{2}\|\boldsymbol{e}_x\|^2\right)d\tau$$

and hence

$$\int_0^t \|\boldsymbol{e}_x\|^2 d\tau \le 2(V(0) - V(t)) \le 2V(0)$$

because $\dot{V} \le 0$ and $V > 0$, which means that

$$\lim_{t\to\infty}\int_0^t \|\boldsymbol{e}_x\|^2 d\tau$$

exists and is finite. By Barbalat's lemma, stated on page 395, $\|\boldsymbol{e}_x\|^2 \to 0$ as $t \to \infty$, which implies that

$$\lim_{t\to\infty}\boldsymbol{e}_x = \boldsymbol{0}.$$

Projection Operator

Consider a vector-valued function of time,

$$\boldsymbol{\theta}(t) = \begin{bmatrix} \theta_1(t) & \cdots & \theta_m(t) \end{bmatrix}^\top \in \mathbb{R}^m,$$

where, $\dot{\theta}_i(t) = \alpha_i(t)$, $i = 1, \cdots, m$. We wish the components $\theta_i(t)$s to lie between $\underline{\theta}_i$ and $\bar{\theta}_i$, that is, $\underline{\theta}_i \le \theta_i(t) \le \bar{\theta}_i$. One way to achieve this goal is by setting $\dot{\theta}_i(t) = 0$ when $\theta_i(t)$ reaches either of the bounds and tends to go beyond the bound, that is,

$$\dot{\theta}_i(t) = \begin{cases} 0 & \text{if } \theta_i = \underline{\theta}_i \text{ and } \alpha_i(t) < 0 \\ 0 & \text{if } \theta_i = \bar{\theta}_i \text{ and } \alpha_i(t) > 0 \\ \alpha_i(t) & \text{otherwise.} \end{cases} \tag{15.122}$$

For compactness, we use the notation *Proj* for the right hand side of (15.122). Then, we write

$$\dot{\theta}_i(t) = Proj_{\theta_i}(\alpha_i),\ i = 1, \ldots, m, \tag{15.123}$$

or in vector notation,

$$\dot{\boldsymbol{\theta}}(t) = Proj_{\boldsymbol{\theta}}(\boldsymbol{\alpha}). \tag{15.124}$$

Sliding Mode Tracking Controller

Let

$$\boldsymbol{s} = \begin{bmatrix} s_1 & s_2 & \cdots & s_{n-1} & 1 \end{bmatrix} \in \mathbb{R}^{1\times n}$$

and $\sigma(\boldsymbol{e}) = \boldsymbol{s}\boldsymbol{e}$. Then, $\{\boldsymbol{e} : \sigma(\boldsymbol{e}) = 0\}$ represents a sliding surface in the tracking error, $\boldsymbol{e}$, space. The coefficients s_i of the sliding surface are chosen so that the system (15.113) restricted to the sliding surface is asymptotically stable—see Section 10.5 for a sliding surface design algorithm. Our objective is to construct a fuzzy tracking controller that forces the tracking error to converge to the sliding surface, $\{\sigma(\boldsymbol{e}) = 0\}$, and then stay in some neighborhood of the sliding surface for all subsequent time. If f and g were known exactly to us and $\eta = 0$, then the controller

$$u^* = \frac{1}{g}\left(-f - \mu\sigma + x_d^{(n)} - \boldsymbol{k}^\top\boldsymbol{e}\right), \tag{15.125}$$

where $\boldsymbol{k} = \begin{bmatrix} 0 & s_1 & \cdots & s_{n-1} \end{bmatrix}^\top$ and $\mu > 0$ is a design parameter, would force the tracking error, $\boldsymbol{e}$, to behave as desired. We can verify this using standard arguments from sliding mode control practice—see Section 10.2. We consider the function $V = 0.5\sigma^2$. Note that V

is positive definite with respect to the sliding surface $\{\sigma = 0\}$. The function V can be viewed as a distance measure to the sliding surface $\{\sigma = 0\}$. The time derivative of V evaluated on the trajectories of the closed-loop system (15.113), (15.125) is

$$\begin{aligned} \dot{V} &= \sigma\dot{\sigma} \\ &= \sigma \boldsymbol{s}\dot{\boldsymbol{e}} \\ &= \sigma\left(e^{(n)} + s_{n-1}e^{(n-1)} + \cdots + s_1\dot{e}\right) \\ &= \sigma\left(x^{(n)} - x_d^{(n)} + \boldsymbol{k}^\top \boldsymbol{e}\right) \\ &= \sigma\left(f + gu - x_d^{(n)} + \boldsymbol{k}^\top \boldsymbol{e}\right) \end{aligned} \tag{15.126}$$

Substituting into the equation above the control law given by (15.125) yields

$$\dot{V} = -\mu\sigma^2. \tag{15.127}$$

Thus, the sliding surface, $\{\sigma = 0\}$, is asymptotically attractive and the system restricted to the sliding surface can be made asymptotically stable with respect to the origin by an appropriate choice of the parameters, s_i, of the sliding surface.

The control law (15.125) requires perfect knowledge of the plant's model. However, the plant model components f and g are unknown to us. In the following subsection, we describe how to use adaptive fuzzy logic controllers to approximate, as closely as possible, the performance of (15.125). We consider two cases. In the first case, the function $f = f(\boldsymbol{x})$ is unknown to us while $g = g(\boldsymbol{x}) > 0$ is known to us. In the second case, both f and g are unknown to us. In both cases, we consider the presence of a bounded disturbance modeled by η.

15.7.3 Controllers

Unknown $f(\boldsymbol{x})$

When in our plant model (15.113) only $g(\boldsymbol{x})$ is known, $\eta \neq 0$, and $f(\boldsymbol{x})$ is unknown, we propose to approximate $f(\boldsymbol{x})$ with fuzzy logic system of the form $\boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f(\boldsymbol{x})$, where the functions $\boldsymbol{\theta}_f$ and $\boldsymbol{\xi}_f$ are described after defining relevant symbols. We use $\boldsymbol{\theta}_f^*$ to denote an "optimal" vector such that

$$\boldsymbol{\theta}_f^* = \operatorname{argmin}_{\boldsymbol{\theta}_f} \sup_{\boldsymbol{x}\in\Omega} \left| f(\boldsymbol{x}) - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f(\boldsymbol{x}) \right|, \tag{15.128}$$

where $\Omega \subseteq \mathbb{R}^n$ is a region to which the state $\boldsymbol{x}$ is constrained to reside.

The notation argmin (or $\arg\min$) stands for the argument, x^*, that yields the minimal value of the minimized function, $f(x)$. This is well-defined only if the minimal value of $f(x)$

is reached at a single point. Thus, $x^* = \arg\min_x f(x)$ holds if and only if x^* is the unique minimizer of $f(x)$.

We assume that

$$\left| f(\boldsymbol{x}) - \boldsymbol{\theta}_f^{*\top} \boldsymbol{\xi}_f(\boldsymbol{x}) \right| \leq d_f \quad \text{for all} \quad \boldsymbol{x} \in \Omega, \tag{15.129}$$

where $d_f > 0$ and each element of $\boldsymbol{\theta}_f^*$ is constant and bounded below and above as follows

$$\underline{\theta}_{fi} \leq \theta_{fi}^* \leq \bar{\theta}_{fi}, \quad \text{for all} \quad i = 1, \ldots, m, \tag{15.130}$$

or in vector notation,

$$\underline{\boldsymbol{\theta}}_f \leq \boldsymbol{\theta}_f^* \leq \bar{\boldsymbol{\theta}}_f. \tag{15.131}$$

We define the adaptation parameter error as

$$\boldsymbol{\phi}_f = \boldsymbol{\theta}_f - \boldsymbol{\theta}_f^*. \tag{15.132}$$

We use the adaptation law

$$\dot{\boldsymbol{\phi}}_f(t) = \dot{\boldsymbol{\theta}}_f(t) = Proj_{\boldsymbol{\theta}_f}(\gamma_f \sigma \boldsymbol{\xi}_f(t)), \tag{15.133}$$

where $\gamma_f > 0$ is a design parameter. It is easy to verify, using the definition of $Proj$ and the fact that $\underline{\theta}_{fi} \leq \theta_{fi}^* \leq \bar{\theta}_{fi}$ for each i, that

$$\boldsymbol{\phi}_f^\top \left(\frac{1}{\gamma_f} Proj_{\boldsymbol{\theta}_f} \left(\gamma_f \sigma \boldsymbol{\xi}_f \right) - \sigma \boldsymbol{\xi}_f \right) \leq 0. \tag{15.134}$$

We analyze the following fuzzy adaptive control law,

$$\boxed{u = \frac{1}{g} \left(-\boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f - \mu\sigma + x_d^{(n)} - \boldsymbol{k}^\top \boldsymbol{e} + u_s \right)} \tag{15.135}$$

with u_s satisfying

$$\sigma \left(f - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + u_s + \eta \right) \leq \epsilon, \tag{15.136}$$

$$\sigma u_s \leq 0, \tag{15.137}$$

where $\epsilon > 0$ is a design parameter. We state and prove a theorem concerning the dynamical behavior of the closed-loop system driven by the control law (15.135).

Theorem 15.8 *For the closed-loop system,*

$$\left. \begin{array}{rcl} x^{(n)} &=& f + gu + \eta \\ u &=& \frac{1}{g} \left(-\boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f - \mu\sigma + x_d^{(n)} - \boldsymbol{k}^\top \boldsymbol{e} + u_s \right) \\ \dot{\boldsymbol{\theta}}_f &=& Proj_{\boldsymbol{\theta}_f}(\gamma_f \sigma \boldsymbol{\xi}_f) \end{array} \right\} \tag{15.138}$$

where u_s satisfies (15.136) and (15.137), we have:

(i)

$$\sigma^2(t) \leq e^{-2\mu t}\sigma^2(0) + \frac{\epsilon}{\mu}; \tag{15.139}$$

(ii) if $\eta = 0$ and there exists $\boldsymbol{\theta}_f^*$ such that $f(\boldsymbol{x}) = \boldsymbol{\theta}_f^{*\top}\boldsymbol{\xi}_f(\boldsymbol{x})$, then the origin of the augmented $(\sigma,\ \boldsymbol{\phi}_f)$-space is stable, and hence $\sigma(t)$ and $\boldsymbol{\phi}_f(t)$ are bounded for all $t \geq 0$ and $e(t) \to 0$ as $t \to \infty$.

Proof To prove *(i)*, consider $V = 0.5\sigma^2$. The time derivative of V evaluated on the solutions of the closed-loop system (15.113), (15.135) is

$$\begin{aligned} \dot{V} &= \sigma\dot{\sigma} \\ &= -\mu\sigma^2 + \sigma\left(f - \boldsymbol{\theta}_f^\top \xi_f + u_s + \eta\right). \end{aligned}$$

Taking into account (15.136) yields

$$\dot{V} \leq -\mu\sigma^2 + \epsilon = -2\mu V + \epsilon.$$

Applying to the above the comparison theorem, presented in Subsection A.14.2, we obtain

$$\begin{aligned} V(t) &\leq e^{-2\mu t}V(0) + \frac{\epsilon}{2\mu}\left(1 - e^{-2\mu t}\right) \\ &\leq e^{-2\mu t}V(0) + \frac{\epsilon}{2\mu}. \end{aligned}$$

Noting in the above that $V = 0.5\sigma^2$, we obtain (15.139), which proves the first part of the theorem.

To prove *(ii)*, consider the following Lyapunov function candidate,

$$V = \frac{1}{2}\left(\sigma^2 + \frac{1}{\gamma_f}\boldsymbol{\phi}_f^\top\boldsymbol{\phi}_f\right). \tag{15.140}$$

We evaluate the time derivative of the above Lyapunov function candidate on the solutions of the closed-loop system (15.138), where $\eta = 0$. We obtain,

$$\dot{V} = \sigma\left(-\mu\sigma + f - \boldsymbol{\theta}_f^\top\boldsymbol{\xi}_f + u_s\right) + \frac{1}{\gamma_f}\boldsymbol{\phi}_f^\top\dot{\boldsymbol{\phi}}_f.$$

Taking into account the assumption that $f = \boldsymbol{\theta}_f^{*\top}\boldsymbol{\xi}_f$ and using (15.132), (15.133), (15.134) and (15.137) gives

$$\begin{aligned} \dot{V} &= -\mu\sigma^2 + \sigma u_s - \left(\boldsymbol{\theta}_f^\top\boldsymbol{\xi}_f - \boldsymbol{\theta}_f^{*T}\boldsymbol{\xi}_f\right)\sigma + \frac{1}{\gamma_f}\boldsymbol{\phi}_f^\top\dot{\boldsymbol{\phi}}_f \\ &= -\mu\sigma^2 + \sigma u_s - \boldsymbol{\phi}_f^\top\boldsymbol{\xi}_f\sigma + \boldsymbol{\phi}_f^\top\frac{1}{\gamma_f}Proj_{\boldsymbol{\theta}_f}(\gamma_f\sigma\boldsymbol{\xi}_f) \\ &\leq -\mu\sigma^2 + \sigma u_s \\ &\leq -\mu\sigma^2. \end{aligned} \tag{15.141}$$

Thus, $\dot{V} \leq 0$ in the $\begin{bmatrix} \sigma & \boldsymbol{\phi}_f \end{bmatrix}^\top$-space, which implies that the zero solution of the closed-loop system (15.138) is stable, and therefore, $\sigma(t)$ and $\boldsymbol{\phi}_f(t)$ are bounded for $t \geq 0$.

To show that $\sigma(t) \to 0$ as $t \to \infty$, we integrate (15.141) from 0 to t to obtain

$$\int_0^t \dot{V}(\tau) d\tau = V(t) - V(0) \leq \int_0^t \left(-\mu\sigma^2\right) d\tau, \tag{15.142}$$

or equivalently,

$$\int_0^t \mu\sigma^2 d\tau \leq V(0) - V(t) \leq V(0). \tag{15.143}$$

Thus, $\lim_{t\to\infty} \int_0^t \sigma^2(\tau) d\tau$ exists and is finite. It follows from the first part of the theorem that σ^2 is bounded and because

$$\frac{d}{dt}\sigma^2 = 2\sigma\dot{\sigma} \leq -\mu\sigma^2 + \epsilon, \tag{15.144}$$

we conclude that $\frac{d}{dt}\sigma^2$ is also bounded. Therefore, σ^2 is uniformly continuous. By Barbalat's lemma, stated on page 395, $\sigma^2 \to 0$ as $t \to \infty$. This implies that the tracking error, $e(t) \to 0$ as $t \to \infty$, which concludes the proof of the theorem.

□

Selecting u_s in control law (15.135)

We discuss a method that can be used to select the component u_s while implementing the control law (15.135)). Let h be a constant such that

$$h \geq \left\| \boldsymbol{\xi}_f \right\| \left\| \bar{\boldsymbol{\theta}}_f - \underline{\boldsymbol{\theta}}_f \right\| .$$

Given the design parameter ϵ, choose positive numbers ϵ_1, ϵ_2, and ϵ_3 such that $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3$. Then, u_s can be chosen as

$$u_s = -k_s\sigma, \tag{15.145}$$

where

$$k_s \geq \frac{d_f^2}{4\epsilon_1} + \frac{h^2}{4\epsilon_2} + \frac{d^2}{4\epsilon_3}. \tag{15.146}$$

We verify that the above choice of u_s guarantees that (15.145) satisfies (15.136) and (15.137). We first consider the left-hand side of (15.136). Adding and subtracting ${\boldsymbol{\theta}_f^*}^\top \boldsymbol{\xi}_f$ and using (15.132) yields

$$\begin{aligned} \sigma(f - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + u_s + \eta) &= \sigma(f - {\boldsymbol{\theta}_f^*}^\top \boldsymbol{\xi}_f + {\boldsymbol{\theta}_f^*}^\top \boldsymbol{\xi}_f - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + u_s + \eta) \\ &= \sigma(f - {\boldsymbol{\theta}_f^*}^\top \boldsymbol{\xi}_f - {\boldsymbol{\phi}_f}^\top \boldsymbol{\xi}_f - k_s\sigma + \eta). \end{aligned} \tag{15.147}$$

We use (15.146) and group the terms to obtain

$$\begin{aligned}\sigma(f-\boldsymbol{\theta}_f^\top\boldsymbol{\xi}_f+u_s+\eta) &\le \sigma\left(f-\boldsymbol{\theta}_f^{*\top}\boldsymbol{\xi}_f-\frac{d_f^2}{4\epsilon_1}\sigma\right)\\ &\quad -\sigma\left(\boldsymbol{\phi}_f{}^\top\boldsymbol{\xi}_f+\frac{h^2}{4\epsilon_2}\sigma\right)+\sigma\left(\eta-\frac{d^2}{4\epsilon_3}\sigma\right). \end{aligned} \tag{15.148}$$

Next, we consider each term on the right hand side of the above inequality separately. We start with the first term. Completing the squares, we obtain

$$\sigma\left(f-\boldsymbol{\theta}_f^{*\top}\boldsymbol{\xi}_f-\frac{d_f^2}{4\epsilon_1}\sigma\right)=-\left(\frac{d_f}{2\sqrt{\epsilon_1}}\sigma-\frac{f-\boldsymbol{\theta}_f^{*\top}\boldsymbol{\xi}_f}{d_f/\sqrt{\epsilon_1}}\right)^2+\left(\frac{f-\boldsymbol{\theta}_f^{*\top}\boldsymbol{\xi}_f}{d_f/\sqrt{\epsilon_1}}\right)^2.$$

Neglecting the first term in the above and using the assumption (15.129), we get

$$\sigma\left(f-\boldsymbol{\theta}_f^{*\top}\boldsymbol{\xi}_f-\frac{d_f^2}{4\epsilon_1}\sigma\right)\le\frac{d_f^2}{d_f^2/\epsilon_1}\le\epsilon_1. \tag{15.149}$$

Completing the squares in the second term on the right-hand side of (15.148) gives

$$\begin{aligned}-\sigma\left(\boldsymbol{\phi}_f{}^\top\boldsymbol{\xi}_f+\frac{h^2}{4\epsilon_2}\sigma\right) &= -\left(\frac{h}{2\sqrt{\epsilon_2}}\sigma+\frac{\boldsymbol{\phi}_f^\top\boldsymbol{\xi}_f}{h/\sqrt{\epsilon_2}}\right)^2+\left(\frac{\boldsymbol{\phi}_f^\top\boldsymbol{\xi}_f}{h/\sqrt{\epsilon_2}}\right)^2\\ &\le \left(\frac{\boldsymbol{\phi}_f^\top\boldsymbol{\xi}_f}{h/\sqrt{\epsilon_2}}\right)^2\\ &= \left(\frac{(\boldsymbol{\theta}_f-\boldsymbol{\theta}_f^*)^\top\boldsymbol{\xi}}{h}\right)^2\epsilon_2\\ &\le \frac{\|\boldsymbol{\theta}_f-\boldsymbol{\theta}_f^*\|^2\,\|\boldsymbol{\xi}_f\|^2}{h^2}\epsilon_2\\ &\le \frac{\|\boldsymbol{\theta}_f-\boldsymbol{\theta}_f^*\|^2\,\|\boldsymbol{\xi}_f\|^2}{\|\boldsymbol{\xi}_f\|^2\,\|\bar{\boldsymbol{\theta}}_f-\underline{\boldsymbol{\theta}}_f\|^2}\epsilon_2 \qquad (15.150)\\ &\le \epsilon_2. \qquad (15.151)\end{aligned}$$

Completing the squares in the third term on the right-hand side of (15.148) yields

$$\begin{aligned}\sigma\left(\eta(t)-\frac{d^2}{4\epsilon_3}\sigma\right) &= -\left(\frac{d}{2\sqrt{\epsilon_3}}\sigma-\frac{\eta}{d/\sqrt{\epsilon_3}}\right)^2+\frac{\eta^2}{d^2}\epsilon_3\\ &\le \epsilon_3.\end{aligned}$$

Thus,

$$\sigma\left(f-\boldsymbol{\theta}_f^\top\boldsymbol{\xi}_f+u_s+\eta\right)\le\epsilon_1+\epsilon_2+\epsilon_3=\epsilon.$$

and hence (15.136) is satisfied.

Also, $u_s=-k_s\sigma$, $k_s>0$, satisfies (15.137).

Unknown $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$

We assume that $g(\boldsymbol{x})$ and $f(\boldsymbol{x})$ are unknown to us and $\eta \neq 0$. We approximate $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ with fuzzy logic systems, $\boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f(\boldsymbol{x})$ and $\boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g(\boldsymbol{x})$, respectively. The conditions on $\boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f(\boldsymbol{x})$ are the same as in the previous case, and the conditions on $\boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g(\boldsymbol{x})$ are similar, that is, the optimal parameter vector, $\boldsymbol{\theta}_g^*$, satisfies

$$\boldsymbol{\theta}_g^* = \operatorname{argmin}_{\boldsymbol{\theta}_g} \sup_{\boldsymbol{x}\in\Omega} \left| g(\boldsymbol{x}) - {\boldsymbol{\theta}_g}^\top \boldsymbol{\xi}_g(\boldsymbol{x}) \right|. \tag{15.152}$$

We assume that

$$\left| g(\boldsymbol{x}) - {\boldsymbol{\theta}_g^*}^\top \boldsymbol{\xi}(\boldsymbol{x}) \right| \leq d_g, \quad \text{for all } \boldsymbol{x} \in \Omega, \tag{15.153}$$

where

$$\underline{\theta}_{gi} \leq \theta_{gi}^* \leq \bar{\theta}_{gi}, \quad \text{for all } i = 1, \cdots, m. \tag{15.154}$$

The parameter error, $\boldsymbol{\phi}_g = \boldsymbol{\theta}_g - \boldsymbol{\theta}_g^*$, is adapted according to the law,

$$\dot{\boldsymbol{\phi}}_g = \dot{\boldsymbol{\theta}}_g = Proj_{\boldsymbol{\theta}_g}(\gamma_g \sigma \boldsymbol{\xi}_g u_a),$$

where

$$u_a = \frac{1}{\boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g} \left(-\boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + x_d^{(n)} - \boldsymbol{k}^\top \boldsymbol{e} \right). \tag{15.155}$$

The initial values of the components of $\boldsymbol{\theta}_g$ are chosen to satisfy (15.154), where for each $i = 1, \ldots, m$, we have $\theta_{gi} > 0$. This ensures that $\boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g > 0$ because we choose fuzzy sets so that at any t at least one rule fires and so at least one component of $\boldsymbol{\xi}$ is nonzero and in fact positive. Using the definition of $Proj$, one can verify that

$$\boldsymbol{\phi}_g^\top \left(\frac{1}{\gamma_g} Proj_{\boldsymbol{\theta}_g} \left(\gamma_g \sigma \boldsymbol{\xi}_g u_a \right) - \sigma \boldsymbol{\xi}_g u_a \right) \leq 0. \tag{15.156}$$

We analyze the following fuzzy adaptive control law,

$$\boxed{u = \frac{1}{\boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g} \left(-\boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + x_d^{(n)} - \boldsymbol{k}^\top \boldsymbol{e} \right) - \frac{1}{\underline{g}} \mu \sigma + u_s} \tag{15.157}$$

with u_s satisfying the following conditions,

$$\sigma \left(f - x_d^{(n)} + \boldsymbol{k}^\top \boldsymbol{e} + g u_a + g u_s + \eta \right) \leq \epsilon, \tag{15.158}$$

$$\sigma u_s \leq 0, \tag{15.159}$$

where $\epsilon > 0$ is a design parameter. Using the definition of u_a, given by (15.155), we represent (15.157) as

$$u = u_a - \frac{1}{\underline{g}} \mu \sigma + u_s. \tag{15.160}$$

Theorem 15.9 *For the closed-loop system,*

$$\left.\begin{array}{rcl} x^{(n)} & = & f + gu + \eta \\ u & = & \frac{1}{\boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g}\left(-\boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + x_d^{(n)} - \boldsymbol{k}^\top \boldsymbol{e}\right) - \frac{1}{\underline{g}}\mu\sigma + u_s \\ \dot{\boldsymbol{\theta}}_f & = & Proj_{\boldsymbol{\theta}_f}(\gamma_f \sigma \boldsymbol{\xi}_f) \\ \dot{\boldsymbol{\theta}}_g & = & Proj_{\boldsymbol{\theta}_g}(\gamma_g \sigma \boldsymbol{\xi}_g u_a) \end{array}\right\} \tag{15.161}$$

where u_s satisfies (15.158) and (15.159), we have:

(i)

$$\sigma^2(t) \le e^{-2\mu t}\sigma^2(0) + \frac{\epsilon}{\mu}; \tag{15.162}$$

(ii) if $\eta = 0$ and there exist $\boldsymbol{\theta}_f^*$ and $\boldsymbol{\theta}_g^*$ such that $f(\boldsymbol{x}) = {\boldsymbol{\theta}_f^*}^\top \boldsymbol{\xi}_f(\boldsymbol{x})$ and $g(\boldsymbol{x}) = {\boldsymbol{\theta}_g^*}^\top \boldsymbol{\xi}_g(\boldsymbol{x})$, then the origin of the $\left[\begin{array}{ccc} \sigma & \boldsymbol{\phi}_f & \boldsymbol{\phi}_g \end{array}\right]^\top$-space is stable and hence $\sigma(t)$, $\boldsymbol{\phi}_f(t)$, and $\boldsymbol{\phi}_g(t)$ are bounded and $e(t) \to 0$ as $t \to \infty$.

Proof To prove *(i)*, consider $V = 0.5\sigma^2$. The time derivative of V evaluated on the solutions of the closed-loop system (15.113), (15.157) is

$$\begin{aligned} \dot{V} &= \sigma\dot{\sigma} \\ &= \sigma\left(f + gu + \eta - x_d^{(n)} + \boldsymbol{k}^\top \boldsymbol{e}\right). \end{aligned}$$

Substituting into the above the alternative expression for the control law given by (15.160) and applying (15.158) gives

$$\begin{aligned} \dot{V} &= \sigma\left(f + gu_a - \frac{g}{\underline{g}}\mu\sigma + gu_s + \eta - x_d^{(n)} + \boldsymbol{k}^\top \boldsymbol{e}\right) \\ &\le -\frac{g}{\underline{g}}\mu\sigma^2 + \epsilon \\ &= -2\mu V + \epsilon. \end{aligned}$$

Invoking the comparison theorem, we obtain (15.162).

To prove *(ii)*, consider the following Lyapunov function candidate,

$$V = \frac{1}{2}\left(\sigma^2 + \frac{1}{\gamma_f}\boldsymbol{\phi}_f^\top \boldsymbol{\phi}_f + \frac{1}{\gamma_g}\boldsymbol{\phi}_g^\top \boldsymbol{\phi}_g\right). \tag{15.163}$$

By assumption, $f = {\boldsymbol{\theta}_f^*}^\top \boldsymbol{\xi}_f$, $g = {\boldsymbol{\theta}_g^*}^\top \boldsymbol{\xi}_g$, $\eta = 0$ as well as conditions (15.134), (15.156), and (15.159) hold. In addition, it follows from (15.155) that

$$x_d^{(n)} - \boldsymbol{k}^\top \boldsymbol{e} = \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + \boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g u_a.$$

Taking into account all of the above in evaluating the time derivative of V on the trajectories of the closed-loop system (15.161), we obtain

$$
\begin{aligned}
\dot{V} &= \sigma\left(-\frac{g}{\underline{g}}\mu\sigma + f - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + \left(g - \boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g\right) u_a + g u_s\right) + \frac{1}{\gamma_f}\boldsymbol{\phi}_f^\top \dot{\boldsymbol{\phi}}_f + \frac{1}{\gamma_g}\boldsymbol{\phi}_g^\top \dot{\boldsymbol{\phi}}_g \\
&= \sigma\left(f - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f + \left(g - \boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g\right) u_a + g u_s\right) - \frac{g}{\underline{g}}\mu\sigma^2 \\
&\quad + \boldsymbol{\phi}_f^\top \frac{1}{\gamma_f} Proj_{\boldsymbol{\theta}_f}(\gamma_f \sigma \boldsymbol{\xi}_f) + \boldsymbol{\phi}_g^\top \frac{1}{\gamma_g} Proj_{\boldsymbol{\theta}_g}(\gamma_g \sigma \boldsymbol{\xi}_g u_a) \\
&\le -\frac{g}{\underline{g}}\mu\sigma^2 + \sigma g u_s - \boldsymbol{\phi}_f^\top \boldsymbol{\xi}_f \sigma + \boldsymbol{\phi}_f^\top \frac{1}{\gamma_f} Proj_{\boldsymbol{\theta}_f}(\gamma_f \sigma \boldsymbol{\xi}_f) \\
&\quad - \sigma \boldsymbol{\phi}_g^\top \boldsymbol{\xi}_g u_a + \boldsymbol{\phi}_g^\top \frac{1}{\gamma_g} Proj_{\boldsymbol{\theta}_g}(\gamma_g \sigma \boldsymbol{\xi}_g u_a) \\
&\le -\mu\sigma^2 + g\sigma u_s \\
&\le -\mu\sigma^2 .
\end{aligned}
$$

It follows from the above that the closed-loop system (15.161) is stable, and therefore, $\sigma(t)$, $\boldsymbol{\phi}_f(t)$ and $\boldsymbol{\phi}_g(t)$ are bounded for $t \ge 0$.

Applying Barbalat's lemma, similarly the way we did in the proof of Theorem 15.8, we conclude that $\sigma(t) \to 0$ as $t \to \infty$, which completes the proof of the theorem.

□

Selecting u_s in control law (15.157)

The component u_s appearing in the control law (15.157) can be chosen similarly as in the previous case. We now give a lower bound on the gain k_s. Let

$$h_f \ge \left\|\boldsymbol{\xi}_f\right\| \left\|\bar{\boldsymbol{\theta}}_f - \underline{\boldsymbol{\theta}}_f\right\| \text{ and } h_g \ge \left\|\boldsymbol{\xi}_g\right\| \left\|\bar{\boldsymbol{\theta}}_g - \underline{\boldsymbol{\theta}}_g\right\| |u_a| .$$

Given the design parameter ϵ, select positive numbers ϵ_1, ϵ_2, ϵ_3, and ϵ_4 such that $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4$. Then, k_s should satisfy the following condition,

$$k_s \ge \frac{1}{\underline{g}}\left(\frac{d_f^2 + |u_a|^2 d_g^2}{2\epsilon_1} + \frac{h_f^2}{4\epsilon_2} + \frac{h_g^2}{4\epsilon_3} + \frac{d^2}{4\epsilon_4}\right). \tag{15.164}$$

When the gain k_s satisfies the above condition, $u_s = -k_s\sigma$ guarantees that (15.158) and (15.159) are satisfied. To show that this is indeed the case, consider the left-hand side of (15.158). Use $\boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g u_a + \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f = x_d^{(n)} - \boldsymbol{k}^\top \boldsymbol{e}$ from (15.155), and then add and subtract ${\boldsymbol{\theta}_f^*}^\top \boldsymbol{\xi}_f$ and

$\boldsymbol{\theta}_g^{*\top}\boldsymbol{\xi}_g u_a$ to obtain

$$\begin{aligned}
\sigma\left(f - x_d^{(n)} + \boldsymbol{k}^\top \boldsymbol{e} + g u_a + g u_s + \eta\right) &= \sigma\left(f - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f - \boldsymbol{\theta}_g^\top \boldsymbol{\xi}_g u_a + g u_a + g u_s + \eta\right) \\
&= \sigma\Big(f - \boldsymbol{\theta}_f^{*\top}\boldsymbol{\xi}_f - \boldsymbol{\phi}_f^\top \boldsymbol{\xi}_f + (g - \boldsymbol{\theta}_g^{*\top}\boldsymbol{\xi}_g) u_a \\
&\qquad -\boldsymbol{\phi}_g^\top \boldsymbol{\xi}_g u_a + g u_s + \eta\Big).
\end{aligned}$$

Taking into account (15.164) in the above and rearranging terms gives

$$\begin{aligned}
\sigma\left(f - x_d^{(n)} + \boldsymbol{k}^\top e + g u_a + g u_s + \eta\right) &\le \sigma\left(f - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f - \frac{d_f^2}{2\epsilon_1}\sigma\right) \\
&\quad +\sigma\left(\left(g - \boldsymbol{\theta}_g^{*\top}\boldsymbol{\xi}_g\right) u_a - \frac{|u_a|^2 d_g^2}{2\epsilon_1}\sigma\right) \\
&\quad -\sigma\left(\boldsymbol{\phi}_f^\top \boldsymbol{\xi}_f + \frac{h_f^2}{4\epsilon_2}\sigma\right) - \sigma\left(\boldsymbol{\phi}_g^\top \boldsymbol{\xi}_g u_a + \frac{h_g^2}{4\epsilon_3}\sigma\right) \\
&\quad +\sigma\left(\eta - \frac{d^2}{4\epsilon_4}\sigma\right).
\end{aligned}$$

Completing the squares, similarly as in the previous case, yields

$$\begin{aligned}
\sigma\left(f - x_d^{(n)} + \boldsymbol{k}^\top e + g u_a + g u_s + \eta\right) &\le \left(\frac{f - \boldsymbol{\theta}_f^\top \boldsymbol{\xi}_f}{\sqrt{2} d_f/\sqrt{\epsilon_1}}\right)^2 + \left(\frac{(g - \boldsymbol{\theta}_g^{*\top}\boldsymbol{\xi}_g) u_a}{\sqrt{2}\,|u_a|\, d_g/\sqrt{\epsilon_1}}\right)^2 \\
&\quad + \left(\frac{\boldsymbol{\phi}_f^\top \boldsymbol{\xi}_f}{h_f^2/\sqrt{\epsilon_2}}\right)^2 + \left(\frac{\boldsymbol{\phi}_g^\top \boldsymbol{\xi}_g u_a}{h_g^2/\sqrt{\epsilon_3}}\right)^2 + \left(\frac{\eta}{d/\sqrt{\epsilon_4}}\right)^2 \\
&\le \frac{1}{2}\epsilon_1 + \frac{1}{2}\epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 \\
&= \epsilon.
\end{aligned}$$

Hence, (15.158) is satisfied, and so is (15.159) by the fact that $u_s = -k_s\sigma$, where $k_s > 0$.

15.7.4 Examples

In this Subsection, we present simulation results illustrating the performance of the control strategies discussed above. The controllers are tested on a simplified model of an electric drive system used by Fischle and Schröder [104] to test their fuzzy adaptive controllers. The plant is modeled by the differential equation,

$$\ddot{x} = -\frac{5}{J}\arctan(5\dot{x}) + \frac{c_T}{J}u = f(x, \dot{x}) + g u, \tag{15.165}$$

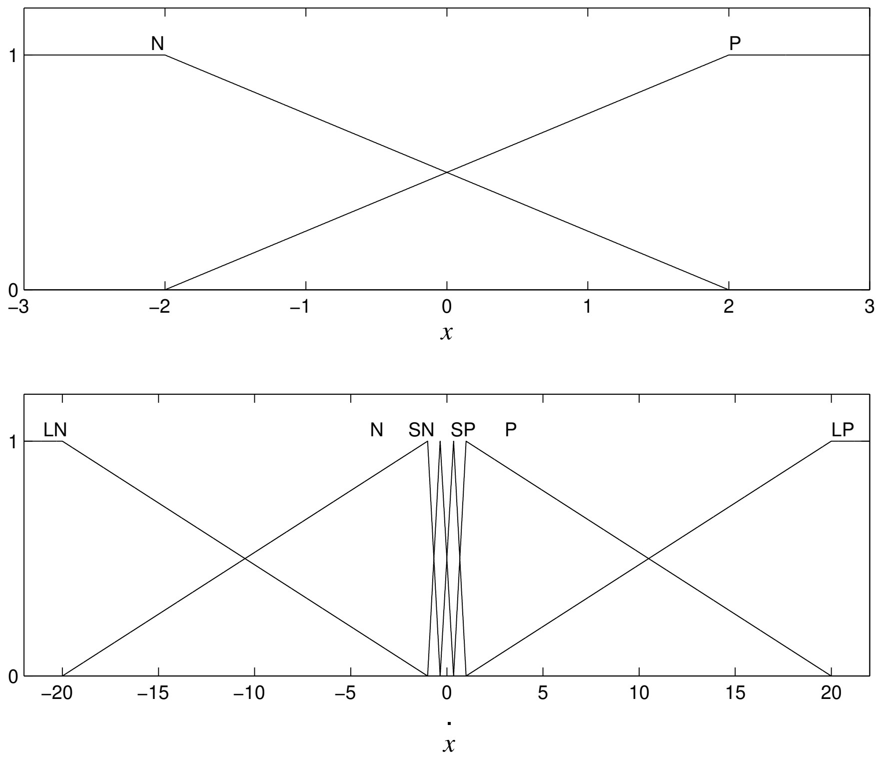


Figure 15.32: Fuzzy sets for x and $\dot{x}$.

where x is the angular position, c_T =10 Nm/A, and $J = 0.1$ kg·m^2. We generate a reference signal x_d using the reference system whose transfer function is given by $\frac{400}{s^2+40s+400}$. The input signal, $w(t)$, to the reference system is changing its value randomly between -1.5 and 1.5 every 0.5 seconds.

Example 15.11 In this example, we assume that we know g, where $g = 100$, and that there are no disturbances affecting the system, that is, $\eta = 0$. We use the control law given by (15.135), where $f(\boldsymbol{x})$ is approximated by a fuzzy logic system. We use fuzzy sets for x and $\dot{x}$ as in Fischle and Schröder [104]. They are shown in Figure 15.32. There are two fuzzy sets for x and six fuzzy sets for $\dot{x}$. Thus, we can have twelve fuzzy rules possible. They have the form:

$$\text{Rule 1: IF } x \text{ is N AND } \dot{x} \text{ is LN THEN } y = \theta_1$$
$$\vdots$$
$$\text{Rule 12: IF } x \text{ is P AND } \dot{x} \text{ is LP THEN } y = \theta_{12},$$

where θ_{fi}, $i = 1, \ldots, 12$ are the adaptation parameters. The bounds on the components of the adaptation parameter vector, $\boldsymbol{\theta}_f$, are chosen to be $\bar{\theta}_{fi} = 200$ and $\underline{\theta}_{fi} = -200$. All the

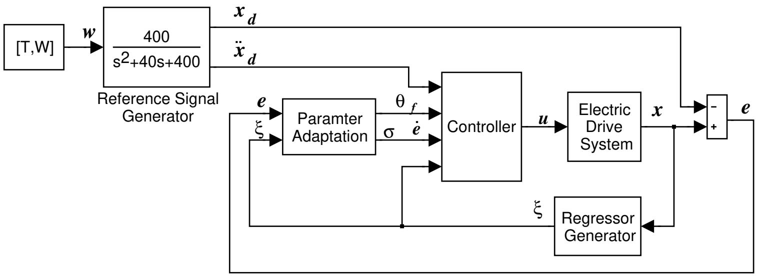


Figure 15.33: SIMULINK block diagram of fuzzy adaptive robust control system of Example 15.11.

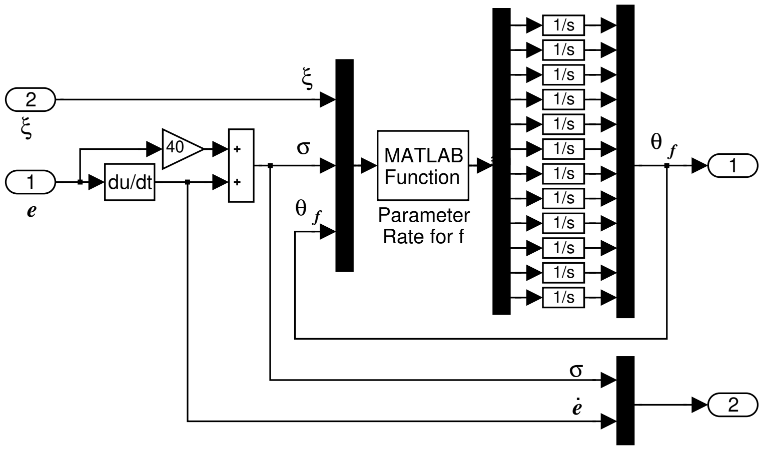


Figure 15.34: Parameter adaptation block of Example 15.11.

initial values of the components of $\boldsymbol{\theta}_f$ are set to zero. We chose $d_f = 50$, $d_g = 10$, and $\gamma_f = 5000$. The values of the remaining design parameters are: $\mu = 250$, $\epsilon_1 = 50$, $\epsilon_2 = 50$. As $\eta(t) = 0$, we do not need to worry about ϵ_3.

SIMULINK is used to simulate the closed-loop system. Its block diagram is depicted in Figure 15.33. The regressor generator block produces $\boldsymbol{\xi}$ using the input x and the parameter adaptation block updates $\boldsymbol{\theta}_f$. It is made up of sets of integrators as shown in Figure 15.34. Adaptation rates of the parameters θ_{fi} are computed with the `MATLAB function` block, depicted in Figure 15.34, using the m-file shown in Figure 15.11. The vector $\boldsymbol{s}$ that defines the sliding surface, $\{\boldsymbol{e} : \boldsymbol{s}\boldsymbol{e} = 0\}$, was chosen as $\boldsymbol{s} = \begin{bmatrix} 40 & 1 \end{bmatrix}$. The simulation results are shown in Figures 15.36 and 15.37. The output $x(t)$, the reference signal $x_d(t)$, and the tracking error $e(t)$ are shown in Figure 15.36. The tracking error, $e(t)$, is so small that one cannot distinguish between the actual state $x(t)$ and the desired state $x_d(t)$. The plots of the control signal u and the components of $\boldsymbol{\theta}_f$ versus time are depicted in Figure 15.37. As

```
%FARC1f_UPDATE.M calculates adapation rates
% (needs 'farc1.mdl' 'farc1_fuzzifier.m', 'af_input_gen.m')

function rate = farc1f_update(x)
% x(1)--x(12) are components of xi,
% x(13) is sigma,  x(14)--x(25) are components of theta
rate = zeros(12,1);
r = 5000;              %adaptation gain(learning rate)
theta_lb = -200;
theta_ub =  200;

for i = 1:12;
   alpha = r*x(13)*x(i);
   rate(i) = alpha*(((x(i+13)>theta_lb)|(alpha>0))&((x(i+13)<...
             theta_ub)|(alpha<0)));
end;
```

Figure 15.35: An m-file used in the block diagram of Figure 15.34 to compute adaptation rates.

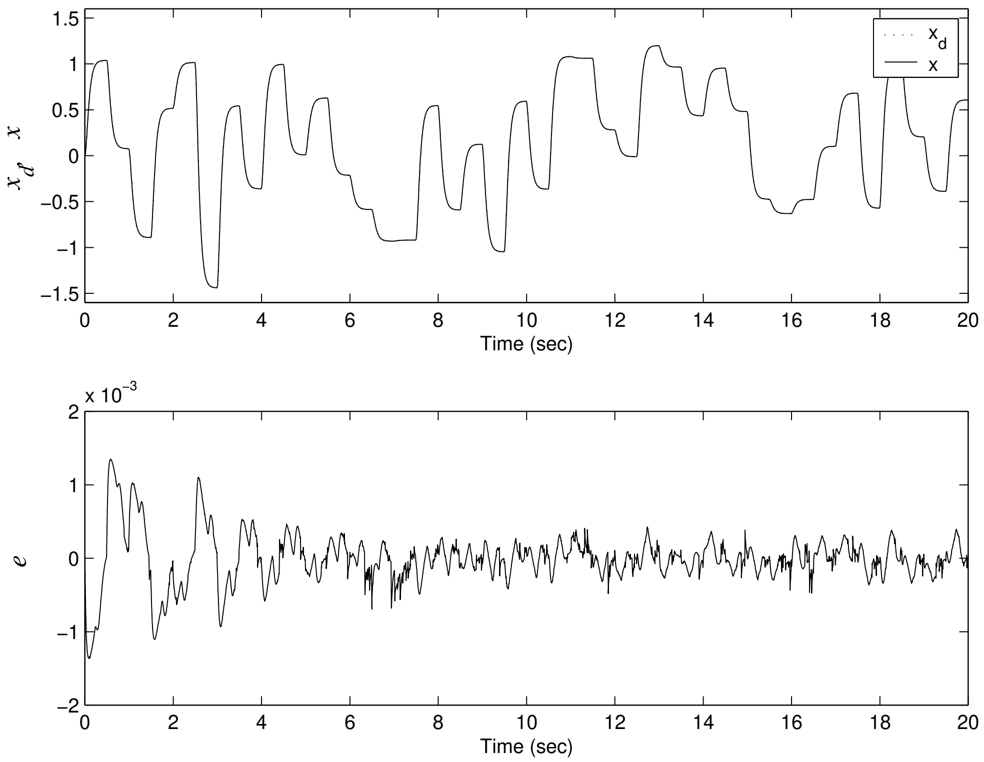


Figure 15.36: Plots of $x(t)$ and error $e(t)$ in Example 15.11.

the fuzzy logic system adapts the parameter $\boldsymbol{\theta}_f$, the error, $e(t)$, gets smaller and smaller. If we used smaller ϵ, we would achieve even smaller tracking error, at the expense of higher control effort.

Example 15.12 In this example, we assume that $f(\boldsymbol{x})$ and g are unknown to us and we employ the control law (15.157). Unknown $f(\boldsymbol{x})$ and g are approximated by two separate fuzzy logic systems using fuzzy sets shown in Figure 15.32 for both fuzzy logic systems.

The bounds on the adaptation parameters are chosen as: $\bar{\theta}_{fi} = 200$, $\underline{\theta}_{fi} = -200$, $\bar{\theta}_{gi} = 150$, and $\underline{\theta}_{gi} = 50$. All the initial values of the components of the adaptation parameter vector $\boldsymbol{\theta}_f$ are set to zero. The initial values of $\boldsymbol{\theta}_g$ are set to 150 to avoid excessively large control signal in the early stages of the simulation run. The selected gains are $\gamma_f = 5000$ and $\gamma_g = 1000$. The other design parameters are: $\epsilon_1 = 50$, $\epsilon_2 = 100$, $\epsilon_3 = 50$, and $\epsilon_4 = 50$. The remaining parameters are the same as in the previous example.

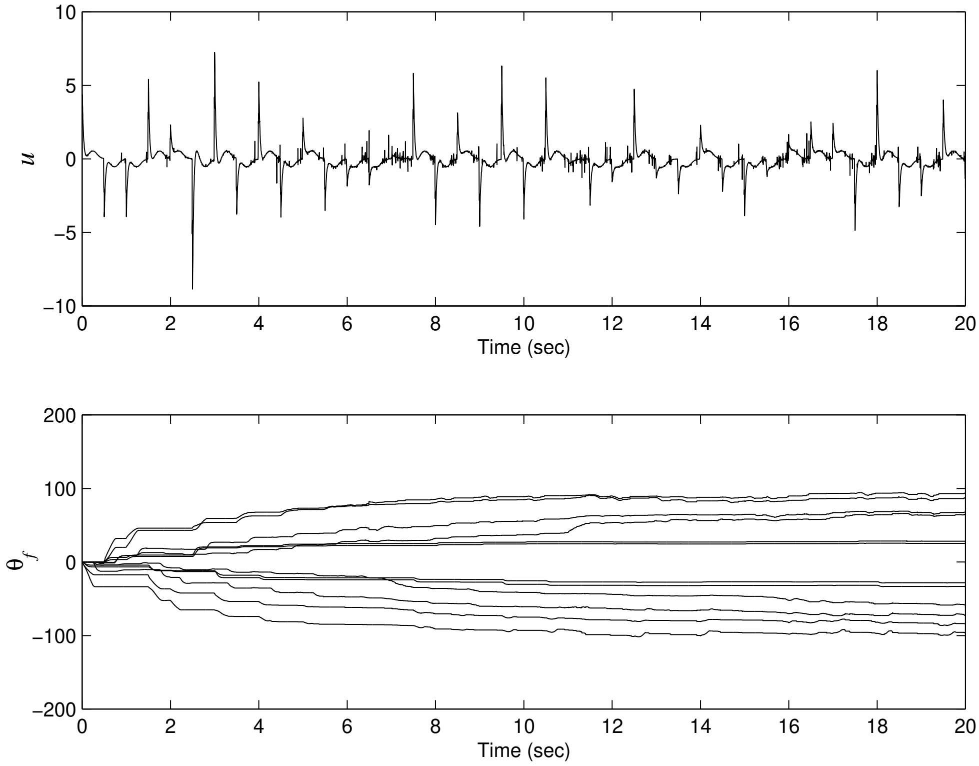


Figure 15.37: Plots of control effort $u(t)$ and the components of the parameter vector $\boldsymbol{\theta}_f$ versus time in Example 15.11.

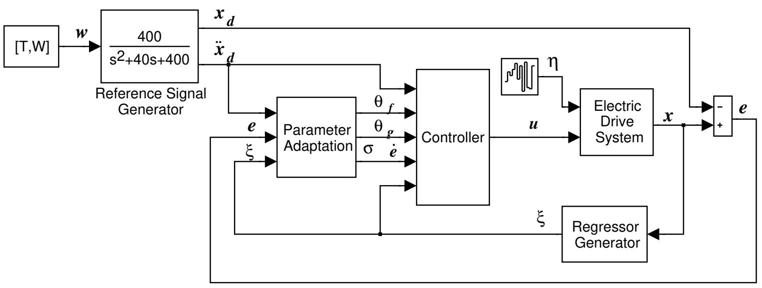


Figure 15.38: SIMULINK block diagram of the fuzzy adaptive robust control system of Example 15.12.

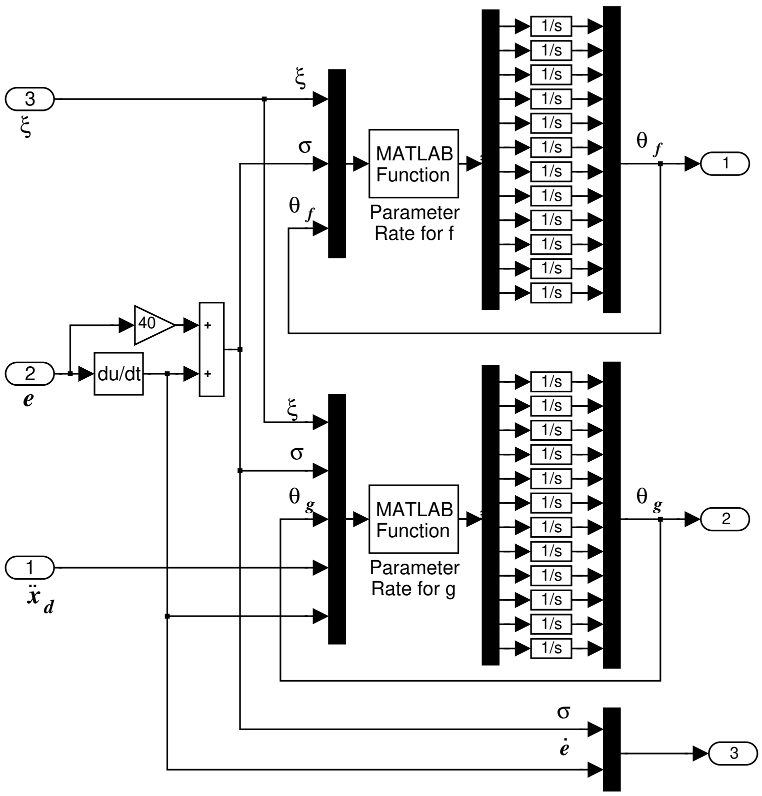


Figure 15.39: Parameter adaptation block of Example 15.12.

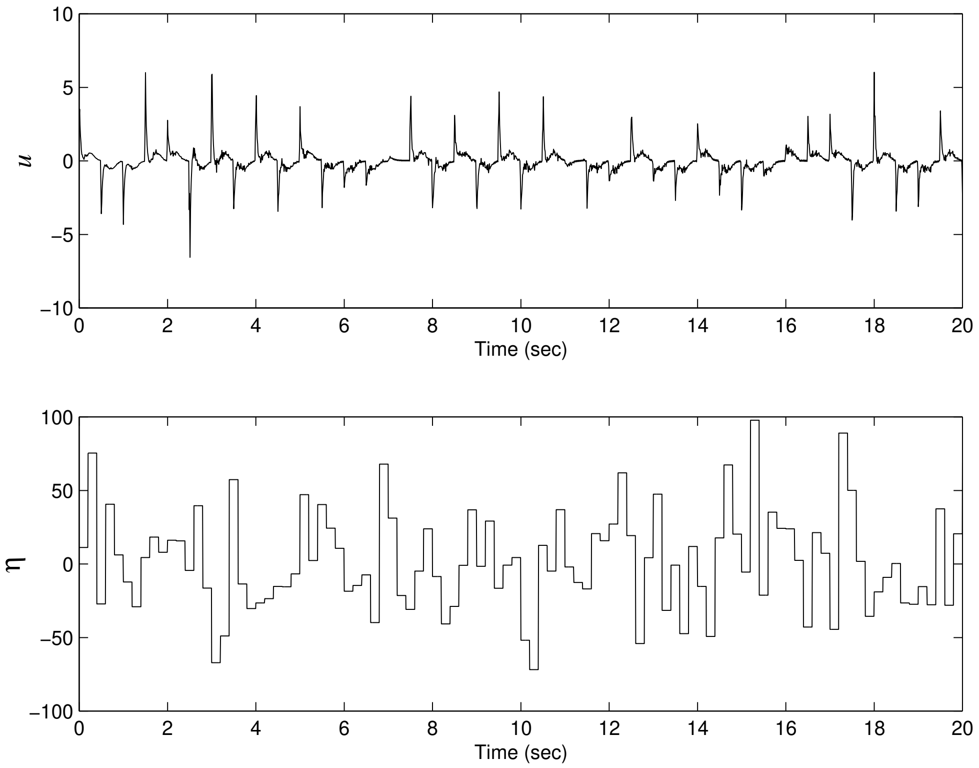


Figure 15.40: Plots of the control effort $u(t)$ and disturbance $\eta(t)$ versus time in Example 15.12.

SIMULINK block diagrams of the closed-loop system and the parameter adaptation law are depicted in Figure 15.38 and Figure 15.39, respectively. The disturbance, η, is a random signal, whose plot versus time is shown in Figure 15.40. We used $d = 100$ so that the condition, $|\eta(t)| \leq d$, is satisfied. We chose, as before, $\boldsymbol{s} = \begin{bmatrix} 40 & 1 \end{bmatrix}$. The simulation results are shown in Figures 15.41, 15.40, and 15.42. As can be seen in Figure 15.41, the tracking error, $e(t)$, remains very small even in the presence of the disturbance, η. Plots of the time history of the adaptation parameters are shown in Figure 15.42.

15.8 Notes

The reader is encouraged to peruse a text on classical logic to become familiar with basic concepts of logic before undertaking his or her further study of fuzzy logic. An excellent

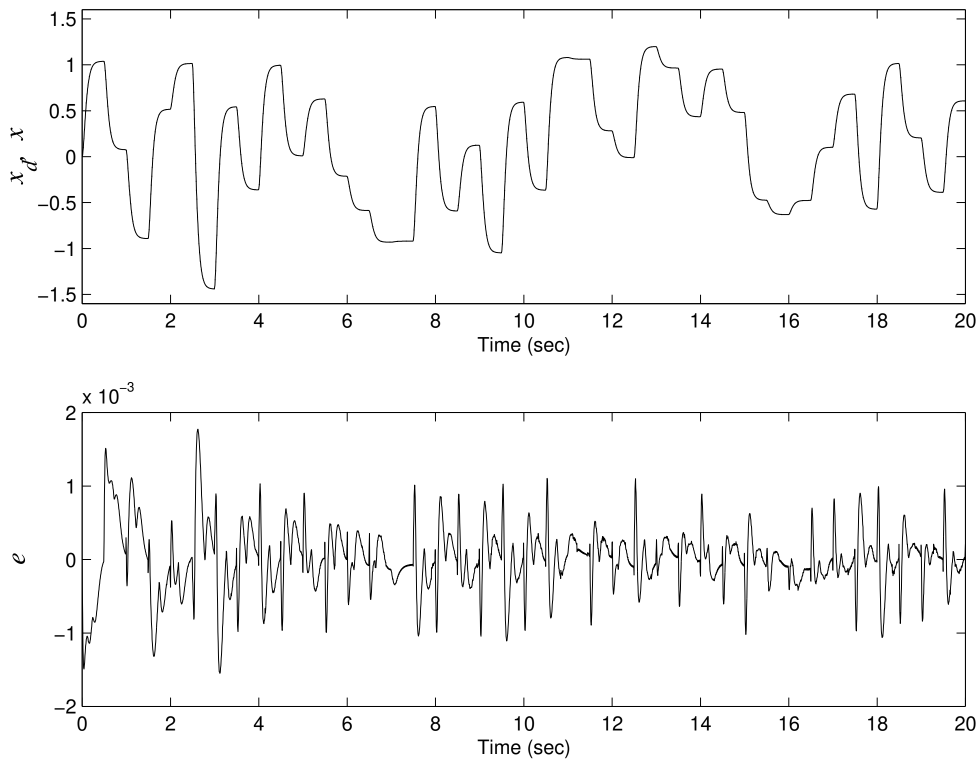


Figure 15.41: Plots of $x(t)$ and error $e(t)$ versus time in Example 15.12.

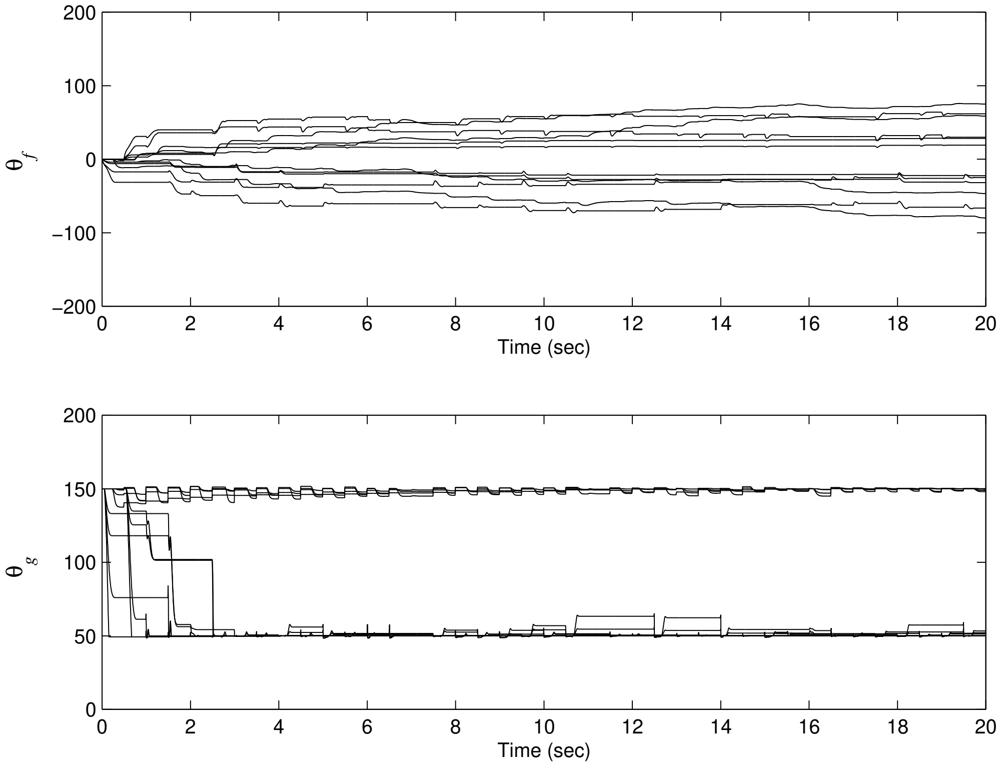


Figure 15.42: Plots of the components of the parameter vectors $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_g$ versus time in Example 15.12

introduction to logic is Copi [66]. For an introduction to fuzzy set theory the reader is referred to Klir, St.Clair, and Yuan [182], while for an introduction to fuzzy control, see Driankov, Hellendoorn, and Reinfrank [86]. Kosko [187] is the first book on fuzzy engineering. Graduate texts on fuzzy systems and control are Pedrycz [268] and Wang [345]. Paradigms and practice of fuzzy modeling are the topics of the edited book by Pedrycz [269]. Lee and Lin [215], and Jang, Sun, and Mizutani [161] cover applications of fuzzy systems as well as issues of merging fuzzy logic and neural networks. Mendel [239], on the other hand, presents extensions of the original fuzzy logic to deal with uncertainties.

The first chapter of Kosko's book [187] contains a thrilling account of the beginnings of fuzzy logic and fuzzy engineering. Kosko [187, page 5] observes that "Fuzziness began as vagueness in the late nineteenth century." A concept is vague when it has blurred boundaries. For example, the concept *mountain* is vague because we do not know where a mountain ends and a hill begins. It was Bertrand Russell who in 1923 published a paper in which he described the concept of vagueness in terms of symbolic logic. In the 1920s, Polish mathematician Jan Łukasiewicz invented the multivalued logic. Max Black applied multivalued logic to sets of objects in the 1930s. In the 1950s, a number of logicians worked out a vague set algebra. In his 1965 paper, Zadeh [364] used the term *fuzzy* to mean *vague*. According to Kosko [187, page 8], "Zadeh's 1965 paper applied Łukasiewicz's logic to each object in a set to work out a complete fuzzy set algebra and to extend the convex separation theorem to pattern recognition." The beginning of fuzzy engineering, on the other hand, can be marked by the work of Mamdani and Assilian [228] who first used fuzzy calculus to construct a fuzzy logic controller.

The method of constructing fuzzy design models described in Section 15.4 as well as the stabilizability conditions of fuzzy models along with a Lyapunov based stabilizer were first reported in [326]. Teixeira, Assunção, and Avellar [325] present Lyapunov-based design methods of fuzzy regulators and observers using LMIs. The controllers analyzed in Section 15.7 come from Lee et al. [206]. Other fuzzy adaptive control strategies were developed by Wang [343] and by Fischle and Schröder [104]. Fischle and Schröder [104, page 38] observe that a major limitation of all existing stable adaptive fuzzy controllers, including the ones analyzed by us, is that they were developed for systems with unlimited actuators' authority, that is, for systems with no constraints on the effort level of the actuators. If we are serious about real-life applications of fuzzy adaptive controllers, then the case of the actuators with limited authority has to be rigorously analyzed. Another issue of practical importance is the convergence of the adaptation parameters to their "correct" values. Friedland [108, page 345] states: "One may argue that the failure of a parameter estimate to converge to its correct value is a subsidiary issue as long as the adaptive control works (even with the incorrect parameter estimate). This reasoning might be valid if you can ascertain that the algorithm works for every input to which the system might be subjected. But in most applications the performance of the algorithm is tested only with a limited set of inputs. Satisfactory performance for this set of inputs is not a reliable indicator that its performance will be satisfactory for inputs outside the test set." In the current state-of-the-art adaptive control algorithms, the convergence of the adaptation parameters is usually ensured through the so called *persistently exciting* inputs to the adaptation algorithms. However, the injection of the persistently exciting signals into the adaptation algorithms may not be feasible in real life applications. Thus, an alternative approach to the issues of the convergence of the adaptation parameters is needed.

15.9 Exercises

Exercise 15.1 Consider a process whose input u is a function of two variables e and $\dot{e}$. The membership functions for e, $\dot{e}$ and u are given in Table 15.4. The control rules are:

Table 15.4: Membership matrix table for e, $\dot{e}$ and u for Exercise 15.1.

Fuzzy Numbers	Universe of discourse						
	-3	-2	-1	0	1	2	3
LN	1	0.5	0.25	0	0	0	0
SN	0.5	1	0.5	0.25	0	0	0
ZE	0	0.25	0.5	1	0.5	0.25	0
SP	0	0	0	0.25	0.5	1	0.5
LP	0	0	0	0	0.25	0.5	1

1. If e is ZE and $\dot{e}$ is SP then u is SN
2. If e is ZE and $\dot{e}$ is ZE then u is ZE
3. If e is SN and $\dot{e}$ is SN then u is LP
4. If e is SP and $\dot{e}$ is ZE then u is SN.

Determine the value of u, for $e = 1$ and $\dot{e} = -2$, using

Table 15.5: The rules matrix for Exercise 15.3.

change-in-error / **error**	LN	N	ZE	P	LP
LN	LP	LP	LP	LP	LP
N	P	P	P	LP	LP
ZE	ZE	ZE	ZE	ZE	ZE
P	LN	LN	N	N	N
LP	LN	LN	LN	LN	LN

(a) the algebraic product defuzzifier;

(b) the center-of-gravity defuzzifier.

Exercise 15.2 Use Lemma 15.3 to prove the following result due to Bailey [18]:
Let $\boldsymbol{B}$ be a matrix with negative diagonal elements and non-negative off-diagonal elements. If $\boldsymbol{x}(t;\boldsymbol{x}_0,t_0)$ and $\boldsymbol{y}(t;\boldsymbol{y}_0,t_0)$ are solutions of $\dot{\boldsymbol{x}} \le \boldsymbol{B}\boldsymbol{x}$ and $\dot{\boldsymbol{y}} = \boldsymbol{B}\boldsymbol{y}$, respectively, and $\boldsymbol{x}_0 = \boldsymbol{y}_0$, then $\boldsymbol{x}(t;\boldsymbol{x}_0,t_0) \le \boldsymbol{y}(t;\boldsymbol{y}_0,t_0)$ for all $t \ge t_0$.

Exercise 15.3 Consider the one-link robot manipulator of Exercise 11.2, whose model is given by (2.83) and (2.84). Let

$$\boldsymbol{P} = \boldsymbol{s}^\top \left(\boldsymbol{s}\boldsymbol{s}^\top\right)^{-1} \boldsymbol{s},$$

where $\boldsymbol{s} \in \mathbb{R}^{1\times 2}$ is the row vector that was found in Exercise 11.2. Then, $(\boldsymbol{I}_n - \boldsymbol{P})$ is the orthogonal projector of $\boldsymbol{x}$ onto the sliding surface $\{\boldsymbol{x} : \boldsymbol{s}\boldsymbol{x} = 0\}$. The Euclidean distance of the point $\boldsymbol{x}$ to the surface $\{\boldsymbol{x} : \boldsymbol{s}\boldsymbol{x} = 0\}$ is

$$\|\boldsymbol{P}\boldsymbol{x}\|_2.$$

Design a fuzzy sliding mode controller whose inputs are:

$$\|\boldsymbol{P}\boldsymbol{x}(t)\|_2 \operatorname{sign}(\sigma(\boldsymbol{x}(t))) \quad \text{and} \quad \Delta\|\boldsymbol{P}\boldsymbol{x}(t)\|_2 = \|\boldsymbol{P}\boldsymbol{x}(t)\|_2 - \|\boldsymbol{P}\boldsymbol{x}(t-T_d)\|_2,$$

where T_d is the time delay that is a design parameter. Use the rules matrix given in Table 15.5.

Exercise 15.4 Consider a dynamical system modeled by (15.6)–(15.8). Let

$$\alpha_j(t) = 1 - \sum_{\substack{i=1\\ i\neq j}}^{r} \alpha_i(t). \tag{15.166}$$

Represent (15.6) as

$$\begin{aligned}\dot{\boldsymbol{x}} &= \left(\alpha_j \boldsymbol{A}_j + \sum_{\substack{i=1\\ i\neq j}}^{r} \alpha_i \boldsymbol{A}_i\right)\boldsymbol{x} + \left(\alpha_j \boldsymbol{B}_j + \sum_{\substack{i=1\\ i\neq j}}^{r} \alpha_i \boldsymbol{B}_i\right)\boldsymbol{u} \\ &= \left(\boldsymbol{A}_j + \sum_{\substack{i=1\\ i\neq j}}^{r} \alpha_i\left(\boldsymbol{A}_i - \boldsymbol{A}_j\right)\right)\boldsymbol{x} + \left(\boldsymbol{B}_j + \sum_{\substack{i=1\\ i\neq j}}^{r} \alpha_i\left(\boldsymbol{B}_i - \boldsymbol{B}_j\right)\right)\boldsymbol{u}. \qquad (15.167)\end{aligned}$$

Let

$$\boldsymbol{\alpha}^j(t) = \boldsymbol{\alpha}^j = \begin{bmatrix} \alpha_1 & \cdots & \alpha_{j-1} & \alpha_{j+1} & \cdots & \alpha_r \end{bmatrix}^\top .$$

Observe that $\boldsymbol{\alpha}^j \in [0,1]^{r-1}$, that is, the values of the vector-valued function $\boldsymbol{\alpha}^j(\cdot)$ are in the $(r-1)$-dimensional unit hypercube. Next, let

$$\Delta\boldsymbol{A}(\boldsymbol{\alpha}^j) = \sum_{\substack{i=1\\ i\neq j}}^{r} \alpha_i\left(\boldsymbol{A}_i - \boldsymbol{A}_j\right) \quad \text{and} \quad \Delta\boldsymbol{B}(\boldsymbol{\alpha}^j) = \sum_{\substack{i=1\\ i\neq j}}^{r} \alpha_i\left(\boldsymbol{B}_i - \boldsymbol{B}_j\right).$$

Using the above notation, represent model (15.167), in an equivalent way, as

$$\begin{aligned}\dot{\boldsymbol{x}} &= \left(\boldsymbol{A}_j + \Delta\boldsymbol{A}(\boldsymbol{\alpha}^j)\right)\boldsymbol{x} + \left(\boldsymbol{B}_j + \Delta\boldsymbol{B}(\boldsymbol{\alpha}^j)\right)\boldsymbol{u} \\ &= \boldsymbol{A}_j\boldsymbol{x} + \boldsymbol{B}_j\boldsymbol{u} + \boldsymbol{F}(\boldsymbol{\alpha}^j, \boldsymbol{x}, \boldsymbol{u}), \qquad (15.168)\end{aligned}$$

where $\boldsymbol{F}(\boldsymbol{\alpha}^j, \boldsymbol{x}, \boldsymbol{u}) = \Delta\boldsymbol{A}(\boldsymbol{\alpha}^j)\boldsymbol{x} + \Delta\boldsymbol{B}(\boldsymbol{\alpha}^j)\boldsymbol{u}$. Let

$$\boldsymbol{F}(\cdot) = \boldsymbol{f}(\cdot) + \boldsymbol{B}_j\boldsymbol{h}(\cdot). \qquad (15.169)$$

We refer to $\boldsymbol{f}$ as the *unmatched uncertainty*, while $\boldsymbol{h}$ is referred to as the *matched uncertainty*. Hence, (15.168) takes the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_j\boldsymbol{x} + \boldsymbol{B}_j\boldsymbol{u} + \boldsymbol{f} + \boldsymbol{B}_j\boldsymbol{h}. \qquad (15.170)$$

Assume that

$$\begin{cases} \|\boldsymbol{f}\| = \|\boldsymbol{f}(\boldsymbol{\alpha}^j, \boldsymbol{x})\| \le \alpha_f\|\boldsymbol{x}\| \\ \|\boldsymbol{h}\| = \|\boldsymbol{h}(\boldsymbol{\alpha}^j, \boldsymbol{x}, \boldsymbol{u})\| \le \beta_x\|\boldsymbol{x}\| + \beta_u\|\boldsymbol{u}\|, \end{cases} \qquad (15.171)$$

Prove the following theorem:

Theorem 15.10 *Suppose that $\boldsymbol{A}_j$ is asymptotically stable, and $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ is the solution to the Lyapunov matrix equation $\boldsymbol{A}_j^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_j = -2\boldsymbol{Q}$ for some $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$. Suppose also that*

$$\alpha_f < \frac{\lambda_{\min}(\boldsymbol{Q})}{\lambda_{\max}(\boldsymbol{P})} \quad \textit{and} \quad \beta_u < 1.$$

Then, the state feedback controller $\boldsymbol{u} = -\gamma \boldsymbol{B}_j^\top \boldsymbol{P}\boldsymbol{x}$, *where*

$$\gamma > \frac{\beta_x^2}{4\left(\lambda_{\min}(\boldsymbol{Q}) - \alpha_f \lambda_{\max}(\boldsymbol{P})\right)(1-\beta_u)}$$

globally asymptotically stabilizes the uncertain system for arbitrary $\boldsymbol{f}$ *and* $\boldsymbol{h}$ *that satisfy the norm bounds (15.171).*

Exercise 15.5 In this Exercise, the following definition is used:

Definition 15.3 *We say that system (15.168) satisfies the matching conditions if there exist continuous matrix functions* $\boldsymbol{D}(\cdot) : \mathbb{R}^{r-1} \to \mathbb{R}^{m\times n}$ *and* $\boldsymbol{E}(\cdot) : \mathbb{R}^{r-1} \to \mathbb{R}^{m\times m}$ *such that*

$$\begin{cases} \Delta \boldsymbol{A}(\boldsymbol{\alpha}^j) &= \boldsymbol{B}_j \boldsymbol{D}(\boldsymbol{\alpha}^j) \\ \Delta \boldsymbol{B}(\boldsymbol{\alpha}^j) &= \boldsymbol{B}_j \boldsymbol{E}(\boldsymbol{\alpha}^j) \end{cases}$$

for all $\boldsymbol{\alpha}^j \in [0,1]^{r-1}$.

$\diamond$

Consider the following algorithm for linear controller design for fuzzy system models of the form (15.167) that is the subject of Exercise 15.4. The algorithm is a consequence of Theorem 15.10 of Exercise 15.4.

Algorithm 15.1 Linear Controller Design for Fuzzy Systems

Given fuzzy model (15.167)

1. Set $j = 1$;

2. Check matching conditions (15.172). If the matching conditions are satisfied, calculate matrices $\boldsymbol{D}$ and $\boldsymbol{E}$, and check if $\alpha_u < 1$. If the matching conditions are not satisfied, express the uncertain element as $\boldsymbol{F} = \boldsymbol{f} + \boldsymbol{B}_j\boldsymbol{h}$, calculate β_u and check if $\beta_u < 1$. If $\beta_u > 1$, set $j = j + 1$ and repeat the above computations.

3. Stabilize $\boldsymbol{A}_j$, if necessary. Calculate $\boldsymbol{u}_1 = -\boldsymbol{K}_1\boldsymbol{x}$ so that the matrix $\boldsymbol{A}_j - \boldsymbol{B}_j\boldsymbol{K}_1$ has its eigenvalues in the desired locations. Update the matched uncertainty involving the state vector, that is, replace $\boldsymbol{D}$ with $\boldsymbol{D} - \boldsymbol{E}\boldsymbol{K}_1$;

4. Solve the Lyapunov equation $\boldsymbol{A}_j^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_j = -2\boldsymbol{Q}$ for some $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$. Choosing $\boldsymbol{Q} = \boldsymbol{I}_n$ maximizes the ratio $\lambda_{\min}(\boldsymbol{Q})/\lambda_{\max}(\boldsymbol{P})$. Check if $\alpha_f < \lambda_{\min}(\boldsymbol{Q})/\lambda_{\max}(\boldsymbol{P})$. If α_f does not satisfy the condition, set $j = j + 1$ and go to step 2;

5. Construct $\boldsymbol{u}_2 = -\gamma \boldsymbol{B}_j^\top \boldsymbol{P}\boldsymbol{x}$;

6. Construct the resulting state feedback linear control law $\boldsymbol{u} = \boldsymbol{u}_1 + \boldsymbol{u}_2$.

Illustrate the above algorithm on a reduced-order model of an inverted pendulum mounted on a cart. The equations of motion for the pendulum are

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g\sin(x_1) - amlx_2^2 \sin(2x_1)/2 - a\cos(x_1)u}{4l/3 - aml\cos^2(x_1)}, \end{cases} \tag{15.172}$$

where x_1 is the angle of the pendulum from the vertical line, x_2 is the angular velocity of the pendulum, u is the control force applied to the cart, and $g = 9.8$ m/sec^2 is the magnitude of the acceleration due to gravity. Numerical values of the parameters are given in Table 15.6. Consider modeling equations (15.172) as a truth model. Use the truth model

Table 15.6: Parameter numerical values of the model of Exercise 15.5.

m=mass of pendulum	2.0 kg
M=mass of cart	8.0 kg
$2l$=length of pendulum	1.0 m
$a = 1/(m+M)$	

in your simulations to evaluate the performance of the design. Design a linear stabilizing controller using the above algorithm utilizing a design model. Assume that, by using insight and experience, you obtained the following two rules describing the plant dynamics:

$$\text{Rule 1}\ :\ \text{IF } x_1(t) \text{ is about } 0$$

$$\text{THEN } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_1\boldsymbol{x}(t) + \boldsymbol{b}_1 u(t),$$

and

$$\text{Rule 2}\ :\ \text{IF } x_1(t) \text{ is about } \pm\pi/2$$

$$\text{THEN } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_2\boldsymbol{x}(t) + \boldsymbol{b}_2 u(t),$$

where

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 \\ \frac{g}{4l/3 - aml} & 0 \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} 0 \\ -\frac{a}{4l/3 - aml} \end{bmatrix},$$

and

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 \\ \frac{2g}{\pi(4l/3-amlb^2)} & 0 \end{bmatrix}, \quad \boldsymbol{b}_2 = \begin{bmatrix} 0 \\ -\frac{ab}{4l/3-amlb^2} \end{bmatrix},$$

where $b = \cos(88°)$. Use the following membership functions:

$$\mu_1(x_1) = \frac{1 - 1/\left(1+\exp(-7(x_1-\pi/4))\right)}{1+\exp(-7(x_1+\pi/4))} \quad \text{and} \quad \mu_2(x_1) = 1 - \mu_1(x_1).$$

The above membership functions are the same as the ones employed by Cao et al. [51]. The corresponding fuzzy system model is

$$\dot{\boldsymbol{x}} = (\alpha_1 \boldsymbol{A}_1 + \alpha_2 \boldsymbol{A}_2)\, \boldsymbol{x} + (\alpha_1 \boldsymbol{b}_1 + \alpha_2 \boldsymbol{b}_2)\, u, \tag{15.173}$$

where $\alpha_i = \mu_i$, $i = 1, 2$, because $\mu_1 + \mu_2 = 1$ for all t. Proceed to construct a stabilizing linear controller according to the above algorithm. First, represent model (15.173) as

$$\dot{\boldsymbol{x}} = (\boldsymbol{A}_1 + \alpha_2 (\boldsymbol{A}_2 - \boldsymbol{A}_1))\, \boldsymbol{x} + (\boldsymbol{b}_1 + \alpha_2 (\boldsymbol{b}_2 - \boldsymbol{b}_1))\, u. \tag{15.174}$$

Check that the uncertain elements of system model (15.174) satisfy the matching conditions. Suppose that the desired eigenvalues of $\boldsymbol{A}_1 - \boldsymbol{b}_1 \boldsymbol{k}_1$ are $\{-2, -2\}$. Construct the controller, and test its performance on the truth model. Generate plots of x_1 versus time for different initial angular displacements $x_1(0)$. You can set $x_2(0) = 0$.

Exercise 15.6 The objective of this exercise is to understand better the dynamical behavior of the nonlinear projector used by us in adaptive fuzzy control and defined by (15.122). For example, take $\alpha(t) = \sin(0.5\pi t)$, $\underline{\theta} = -0.3$, and $\bar{\theta} = 0.5$. Compare solutions of $\dot{\theta} = Proj_\theta(\alpha)$ and $\dot{\theta} = \alpha$ for a number of initial conditions. Below is an example of MATLAB's function that implements the differential equation, $\dot{\theta} = Proj_\theta(\alpha)$, where `low=` $\underline{\theta}$ `and upper=` $\bar{\theta}$.

```
function theta_dot = Proj(t,theta)
low=-0.3;
upper=0.5;
alpha=sin(0.5*pi*t);
if (theta <= low)&(alpha<0)
   theta_dot=0;
elseif (theta >= upper)&(alpha>0)
   theta_dot=0;
else
   theta_dot=alpha;
   end
```

Exercise 15.7 For the fuzzy model the stick balancer given in Example 15.8 by equation (15.67), where

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix},$$

construct a fuzzy observer. (Note that we have $\boldsymbol{C}_1 = \boldsymbol{C}_2 = \boldsymbol{C}(\boldsymbol{\alpha})$.) Implement the combined controller-observer compensator using first the fuzzy model and then the nonlinear model of the stick balancer. Write MATLAB scripts that animate the dynamical behavior of the closed-loop systems along with animation of the angle θ and the input u data.

Chapter 16

Neural Networks

> We tend to regard forgetting as a failure of memory, the result of an inadequacy in the machinery for remembering, but a moment's reflection shows that that cannot always be right. If we were to remember every telephone number we have ever used, every scene we have ever looked at, every conversation we have ever heard, every sentence we have ever read, we should be overwhelmed with memories. In fact we transfer to our long-term memory stores only a tiny fraction of the contents of our short-term stores. We retain only what is likely to be useful. [116, page 331]

The motivation behind studying artificial neural networks, commonly referred to as neural networks, or just neural nets, is to build computers whose construction would mimic the organization of the brain. The brain is the most complex natural information processing system. It is capable of organizing its building blocks, called neurons, to perform computations in a very efficient way. A drawing of a typical biological neuron is given in Figure 16.1. The neuron collects signals from other neurons through its dendrites. The signals have the form of electrical pulses. These pulses travel to the neuron's cell body where they are being processed. The surface of the cell body of the neuron is dotted with numerous receptor sites that can receive messages from other neurons in a fashion similar to the dendrites. The neuron may produce its own pulses in response to excitation from other neurons. The neuron's own pulses travel through a long, thin strand called an axon, which splits into many end branches. The neuron's message to the other neurons is delivered via the end branches. The key to the transmission of the neuron's message is the synapse, shown in Figure 16.2. The end branches of the neuron's axon terminate in synaptic knobs. The synaptic knob of the first neuron is separated from the receptor site of another neuron by a microscopic distance.

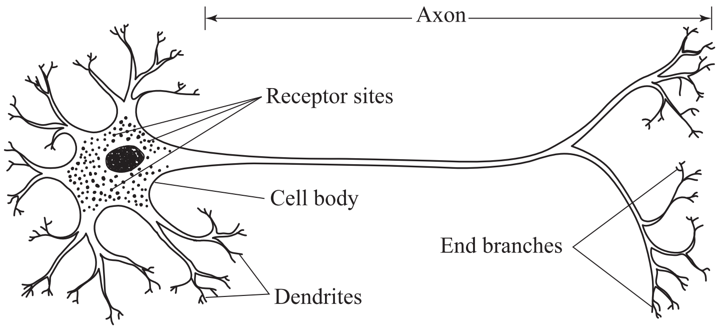


Figure 16.1: A typical biological neuron.

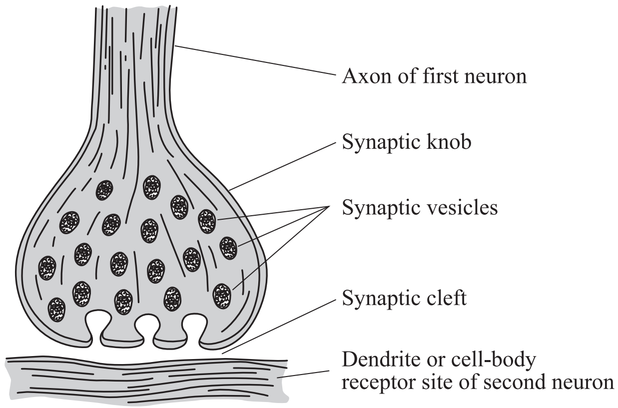


Figure 16.2: The synapse is a junction point where neuron meets neuron.

The cell body of the first neuron produces chemical substances called neurotransmitters, which are delivered down to the synaptic vesicles. The neurotransmitters are stored in the synaptic vesicles until the neuron fires and a burst of neurotransmitters is released by the vesicles. The neurotransmitters then flow across the synaptic cleft and act on the second neuron. The neurotransmitters of the first neuron may stimulate or inhibit the second neuron activity. The first neuron has many other synaptic knobs elsewhere, while the second neuron has many other receptor sites that receive messages from many other neurons. Each neuron is receiving messages from hundreds or thousands of other neurons. The neuron sets off the nervous pulse, that is, it fires off its own nervous pulse only when it is properly stimulated. It was observed that, in general, the neuron will not fire in response to a single message. To fire, the neuron needs many messages arriving simultaneously or in quick succession from other neurons. In addition, the stimulating messages to fire must outweigh the messages that inhibit the neuron from firing.

In the following sections, we discuss a few mathematical models of the biological neuron. These models are referred to as the artificial neurons. Then, we analyze neural networks that are composed of the artificial neurons, and finally we discuss applications of the artificial neural networks in the controller design of dynamical systems.

16.1 Threshold Logic Unit

A simple model of a biological neuron is the so called the *threshold logic unit* (TLU), shown in Figure 16.3. The TLU is also called the *perceptron*. A TLU is parameterized by a set of weights and a threshold (bias). The summation operation is followed by the sign activation function, which models the cell body of a biological neuron. The dendrites are modeled by the inputs and their strengths are modeled by the weights. The axon corresponds to the output y. Suppose that we are given a TLU with n weights, $w_1, w_2, \ldots, w_n$, and a threshold t. Given an input vector $\boldsymbol{x}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & \cdots & x_n^{(i)} \end{bmatrix}^\top \in \mathbb{R}^n$, then the corresponding output of the TLU is

$$y_i = \operatorname{sign}\left(\sum_{j=1}^{n} w_j x_j^{(i)} - t\right), \tag{16.1}$$

where

$$\operatorname{sign}(z) = \begin{cases} -1 & \text{if} \quad z < 0 \\ 1 & \text{if} \quad z \geq 0. \end{cases}$$

We note that sometimes in the literature the sign function is undefined at 0, or it is defined as a point to a set mapping, that is, $\operatorname{sign}(0) \in [-1, 1]$. Here, we define, $\operatorname{sign}(0) = 1$. Using

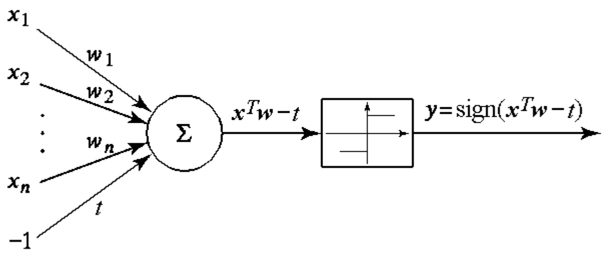


Figure 16.3: A threshold logic unit (TLU), also called the perceptron, as a model of a biological neuron.

the scalar product notation, we represent (16.1) as

$$y_i = \operatorname{sign}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - t\right).$$

Note that $y_i \in \{-1, 1\}$. Specifically,

$$y_i = \begin{cases} -1 & \text{if} \quad \langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - t < 0 \\ 1 & \text{if} \quad \langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - t \geq 0 \end{cases}. \tag{16.2}$$

The goal of the TLU (perceptron) is to correctly classify the set of given input vectors $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(p)}$ into one of the two classes, $\mathcal{C}_1$ or $\mathcal{C}_2$. In other words, if a given vector, say $\boldsymbol{x}^{(i)}$, belongs to class $\mathcal{C}_1$, then the corresponding output y_i should be $+1$. On the other hand, if $\boldsymbol{x}^{(i)}$ belongs to class $\mathcal{C}_2$ then the TLU's output should be -1. Thus, we are faced with a two-class pattern classification problem. The TLU's role is to find a hyperplane that separates two decision regions. The separating hyperplane (decision boundary) has the form

$$\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle - t^* = \sum_{i=1}^{n} w_i^* x_i - t^* = 0. \tag{16.3}$$

In the case of two variables x_1 and x_2 the decision boundary is a straight line. An illustration of such a case is given in Figure 16.4. In this case a point that lies above the boundary line is assigned to class $\mathcal{C}_1$, while a point below the boundary line is assigned to class $\mathcal{C}_2$. We present an iterative algorithm for the decision boundary weights and threshold provided that a decision boundary exists. We assume that we are given p training pairs

$$\left(\boldsymbol{x}^{(1)}, d_1\right), \left(\boldsymbol{x}^{(2)}, d_2\right), \ldots, \left(\boldsymbol{x}^{(p)}, d_p\right),$$

where for $i = 1, 2, \ldots, p$, the input vectors $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$ and $d_i \in \{-1, 1\}$ represent the desired responses. We would like to determine a hyperplane (16.3) such that each input vector lies on a pre-assigned side of the hyperplane. If such a hyperplane exists for the given training set, then the set of input vectors is referred to as *linearly separable*, and the hyperplane is referred to as a *separating hyperplane*. If no such hyperplane exists, then the training set is referred to as *linearly nonseparable*. Finding a separating hyperplane is equivalent to finding a corresponding weight vector and a threshold. If we find such a weight vector and a threshold, then we say that the TLU correctly classifies the training set. Thus, designing a TLU that correctly classifies the given training set amounts to finding a weight vector $\boldsymbol{w}^*$ and a threshold t^* such that for $i = 1, 2, \ldots, p$, we have

$$\begin{aligned} \langle \boldsymbol{w}^*, \boldsymbol{x}^{(i)} \rangle - t^* < 0 \quad &\text{if} \quad d_i = -1 \\ \langle \boldsymbol{w}^*, \boldsymbol{x}^{(i)} \rangle - t^* \geq 0 \quad &\text{if} \quad d_i = 1. \end{aligned}$$

In our discussion, we assume the threshold value, t, to be equal zero. We can achieve this by increasing the dimension of every input vector, $\boldsymbol{x}^{(i)}$, by augmenting it with an entry equal to -1, and by increasing the dimension of the weight vector to $n+1$ by augmenting it with t. Indeed,

$$y_i = \operatorname{sign}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - t\right) = \operatorname{sign}\left(\langle \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{x}}^{(i)} \rangle\right),$$

where $\tilde{\boldsymbol{w}}^\top = \begin{bmatrix} w_1 & w_2 & \cdots & w_n & t \end{bmatrix}$ and $\tilde{\boldsymbol{x}}^{(i)\top} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & \cdots & x_n^{(i)} & -1 \end{bmatrix}$. In our further discussion, we assume that the given input vectors have been already augmented. In view of the above, we can formulate the training problem of a TLU as follows:

TLU Training Problem

Given p training pairs $\left(\boldsymbol{x}^{(1)}, d_1\right), \ldots, \left(\boldsymbol{x}^{(p)}, d_p\right)$, where $\boldsymbol{x}^{(i)} \in \mathbb{R}^{n+1}$ and $d_i \in \{-1, 1\}$. Determine a weight vector $\boldsymbol{w} \in \mathbb{R}^{n+1}$, if there exists one, such that for $i = 1, \ldots, p$,

$$\operatorname{sign}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle\right) = d_i.$$

Assume that the given vectors $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(p)}$ are linearly separable, that is, there exists a weight vector $\boldsymbol{w}^*$ such that

$$\left\langle \boldsymbol{w}^*, \boldsymbol{x}^{(i)} \right\rangle \geq 0 \quad \text{for every input vector } \boldsymbol{x}^{(i)} \text{ from class} \quad \mathcal{C}_1$$

and

$$\left\langle \boldsymbol{w}^*, \boldsymbol{x}^{(i)} \right\rangle < 0 \quad \text{for every input vector } \boldsymbol{x}^{(i)} \text{ from class} \quad \mathcal{C}_2.$$

We can now formulate the algorithm for adapting the weight vector resulting in a separating hyperplane of a given set of linearly separable points. The idea is very simple and we proceed as follows. If the given vector $\boldsymbol{x}^{(i)}$ is classified correctly, do not update the weights, that is,

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k].$$

Otherwise,

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] - \boldsymbol{x}^{(i)} \quad \text{if} \quad \boldsymbol{w}[k]^\top \boldsymbol{x}^{(i)} \geq 0 \quad \text{and} \quad \boldsymbol{x}^{(i)} \in \mathcal{C}_2 \tag{16.4}$$

and

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + \boldsymbol{x}^{(i)} \quad \text{if} \quad \boldsymbol{w}(k)^\top \boldsymbol{x}^{(i)} < 0 \quad \text{and} \quad \boldsymbol{x}^{(i)} \in \mathcal{C}_1. \tag{16.5}$$

Note that if $\boldsymbol{w}[k]^\top \boldsymbol{x}^{(i)} \geq 0$, that is, $y(k) = 1$ and $\boldsymbol{x}^{(i)} \in \mathcal{C}_2$, that is, $d_i = -1$, then the error between the actual TLU output and the desired TLU output is $d_i - y[k] = -2$. Therefore, we can represent the weight update in (16.4) as

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + \frac{1}{2}(d_i - y[k])\boldsymbol{x}^{(i)}. \tag{16.6}$$

Analogously, if $\boldsymbol{w}[k]^\top \boldsymbol{x}^{(i)} < 0$, that is, $y[k] = -1$, and $\boldsymbol{x}^{(i)} \in \mathcal{C}_1$, then $d_i - y[k] = 2$. Here, as in the previous case, we can represent the weight update in (16.5) as (16.6). We now summarize the above considerations in the form of the following algorithm.

Algorithm 16.1 TLU (Perceptron) Learning

Given a set of training pairs $\left\{(\boldsymbol{x}^{(1)}, d_1), \ldots, (\boldsymbol{x}^{(i)}, d_i), \ldots, (\boldsymbol{x}^{(p)}, d_p)\right\}$.

1. **Initialize Weight Vector and Threshold**
   Set the values of the components of the initial weight vector $\boldsymbol{w}[0]$ and threshold $t[0]$ to zero or to small random values.

2. **Present New Input Vector and Desired Output**
   Present the new input vector, $\boldsymbol{x}^{(i)}$ and the corresponding desired output d_i.

3. **Calculate Actual Output**

$$y[k] = \operatorname{sign}\left(\left\langle \boldsymbol{w}[k], \boldsymbol{x}^{(i)} \right\rangle - t[k]\right).$$

4. **Adapt Weight Vector and Threshold**

$$\begin{aligned} \boldsymbol{w}[k+1] &= \boldsymbol{w}[k] + \frac{1}{2}(d_i - y[k])\boldsymbol{x}^{(i)}, \\ t[k+1] &= t[k] - \frac{1}{2}(d_i - y[k]). \end{aligned}$$

Note that the weight vector and threshold are unchanged if the correct decision is made by the TLU.

5. **Repeat by Going to Step 2**

Example 16.1 We used the above algorithm to generate a separating line for eight training pairs. For $i = 1, \ldots, 4$ and $d_i = 1$,

$$\begin{bmatrix} \boldsymbol{x}^{(1)} & \boldsymbol{x}^{(2)} & \boldsymbol{x}^{(3)} & \boldsymbol{x}^{(4)} \end{bmatrix} = \begin{bmatrix} -5 & -4 & 3 & 6 \\ -4 & 6 & 4 & 7 \end{bmatrix}.$$

The above points are labeled "x" in Figure 16.4. The other four training pairs corresponding to $d_i = -1$, $i = 5, \ldots, 8$, labeled "o" are

$$\begin{bmatrix} \boldsymbol{x}^{(5)} & \boldsymbol{x}^{(6)} & \boldsymbol{x}^{(7)} & \boldsymbol{x}^{(8)} \end{bmatrix} = \begin{bmatrix} -6 & 0 & 3 & 8 \\ -8 & -1 & -4 & 3 \end{bmatrix}$$

The weight vector and threshold were initialized randomly, where

$$\boldsymbol{w}[0] = \begin{bmatrix} -0.4860 & 0.8913 \end{bmatrix}^\top \quad \text{and} \quad t[0] = 0.7621$$

The TLU learning algorithm generated a separating line, shown in Figure 16.4, corresponding to the weight vector $\boldsymbol{w}^* = \begin{bmatrix} -5.5140 & 6.8913 \end{bmatrix}^\top$ and threshold $t^* = -2.2379$.

Our goal is to prove that the TLU learning algorithm can always find a separating hyperplane for linearly separable input vectors. In our proof, we use some technical results that we present next.

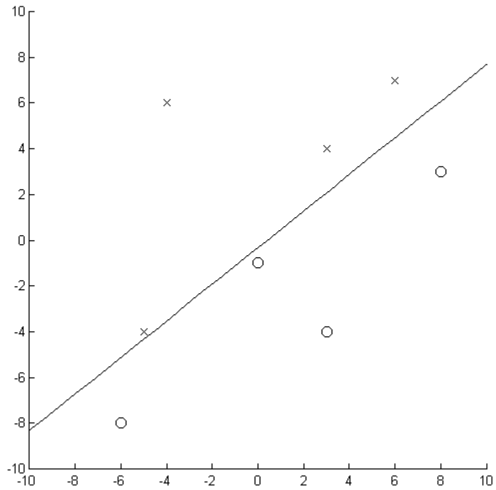


Figure 16.4: An example of separable points in two-dimensional space and a separating line.

Lemma 16.1 *Let $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(p)}$, $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$, be a set of p input vectors, $\boldsymbol{w} \in \mathbb{R}^n$ be a weight vector of an n-input TLU, and $t \in \mathbb{R}$ its threshold. Then, there exists a threshold t' such that, for $i = 1, 2, \ldots, p$,*

$$\left\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \right\rangle - t' \neq 0,$$

and

$$\operatorname{sign}\left(\left\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \right\rangle - t\right) = \operatorname{sign}\left(\left\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \right\rangle - t'\right).$$

Proof Assume, without loss of generality, that the input vectors are sorted into two sets $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(l)}\right\}$ and $\left\{\boldsymbol{x}^{(l+1)}, \ldots, \boldsymbol{x}^{(p)}\right\}$ such that the desired response of the vectors from the first set is $d_i = -1$, while the desired response of the vectors from the second set is $d_i = 1$. Let

$$\delta = \min\left\{\left|\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - t\right| : 1 \leq i \leq l\right\}.$$

It follows from (16.2) that $\delta > 0$. Let $t' = t - \delta/2$. Then,

$$\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - t' = \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - t + \delta/2 \neq 0 \quad \text{for} \quad i = 1, \ldots, p,$$

and the output of the TLU with the modified threshold, t', is the same as the output of the TLU with the old threshold, t.

□

By the above lemma, we can always assume that $\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - t' \neq 0$, for $i = 1, \ldots, p$ and that the output of a TLU can be defined as

$$y_i = \begin{cases} -1 & \text{if} \quad \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - t < 0 \\ 1 & \text{if} \quad \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - t > 0. \end{cases}$$

We further simplify the formulation of the TLU learning problem by preprocessing training pairs. Specifically, we replace every training pair, $\left(\boldsymbol{x}^{(i)}, d_i\right)$ with $d_i = -1$ with their negative, $\left(-\boldsymbol{x}^{(i)}, -d_i\right)$. After performing the above preprocessing, the TLU learning problem can be reformulated in a simple form that we use while analyzing the TLU learning algorithm.

Simplified TLU Learning Problem

Given a set of input vectors $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(p)}\right\}$. Determine a weight vector $\boldsymbol{w}$, if there exists one, such that for $i = 1, \ldots, p$,

$$\left\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \right\rangle > 0.$$

In view of the above, for a linearly separable set of vectors, $\left\{\boldsymbol{x}^{(i)}\right\}$, after preprocessing there exists a hyperplane passing through the origin in $\mathbb{R}^{n+1}$ such that all the vectors lie on one side of the hyperplane. The normal to the hyperplane is the weight vector $\boldsymbol{w}$. This is illustrated in Figure 16.5. Taking into account the simplified TLU learning problem, we can state the TLU learning algorithm equivalently as follows:

Simplified TLU Learning Algorithm

1. **Initialize Weight Vector**
   Set the values of the components of the initial weight vector $\boldsymbol{w}[0]$ to zero.

2. **Test**
   If $\left\langle \boldsymbol{w}[k], \boldsymbol{x}^{(i)} \right\rangle \leq 0$, go to **Add**, else **Test** next input vector.

3. **Add**

   $$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + \boldsymbol{x}^{(i)}, \quad k = k + 1.$$

   Go to **Test**.

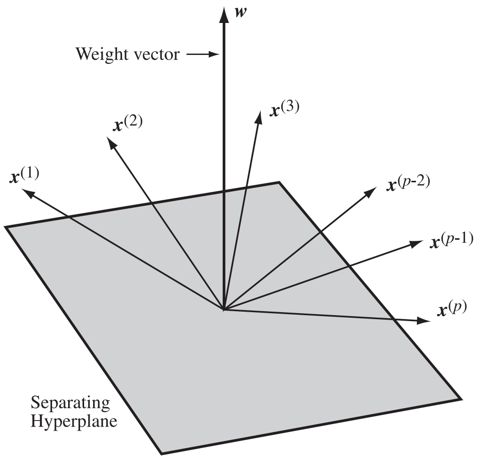


Figure 16.5: Linearly separable vectors after their preprocessing.

The algorithm is said to *converge* if for some $k < \infty$,

$$\left\langle \boldsymbol{w}[k], \boldsymbol{x}^{(i)} \right\rangle > 0, \quad \text{for} \quad i = 1, \ldots, p.$$

The following theorem states that if the set of input vectors is linearly separable, then the TLU learning algorithm converges.

Theorem 16.1 (The TLU Convergence Theorem) *If the set of vectors* $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(p)}\right\}$ *is linearly separable, then the simplified TLU learning algorithm determines a weight vector* $\boldsymbol{w}$ *in a finite number of steps, such that* $\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle > 0$ *for* $i = 1, 2, \ldots, p$

Proof The proof is adapted from Siu, Roychowdhury, and Kailath [299]. We begin by observing that because the set of input vectors is linearly separable, there exists a weight vector $\boldsymbol{w}^*$ and a constant $\delta > 0$ such that for $i = 1, 2, \ldots, p$, we have

$$\langle \boldsymbol{w}^*, \boldsymbol{x}^{(i)} \rangle > \delta.$$

To simplify further manipulations, we assume that $\|\boldsymbol{w}^*\| = 1$. We also assume that $\left\| \boldsymbol{x}^{(i)} \right\| \leq L$, for $i = 1, 2, \ldots, p$, that is, the lengths of all the input vectors are bounded above by L. Let k denote the number of times the weight vector $\boldsymbol{w}$, in the simplified TLU learning algorithm, has been updated, and let $\boldsymbol{w}[k]$ be the value of the weight vector, $\boldsymbol{w}$, after the k-th update. Our goal is to show that k is bounded above. To proceed, we note that if an input vector $\boldsymbol{x}^{(i)}$ is used for the k-th update in the algorithm, then

$$\boldsymbol{w}[k] = \boldsymbol{w}[k-1] + \boldsymbol{x}^{(i)},$$

where $\langle \boldsymbol{w}[k-1], \boldsymbol{x}^{(i)} \rangle = \boldsymbol{w}[k-1]^\top \boldsymbol{x}^{(i)} \leq 0$. We first derive an upper bound on $\|\boldsymbol{w}[k]\|$. We have,

$$\begin{aligned} \|\boldsymbol{w}[k]\|^2 &= \|\boldsymbol{w}[k-1] + \boldsymbol{x}^{(i)}\|^2 \\ &= \left(\boldsymbol{w}[k-1] + \boldsymbol{x}^{(i)}\right)^\top \left(\boldsymbol{w}[k-1] + \boldsymbol{x}^{(i)}\right) \\ &= \boldsymbol{w}[k-1]^\top \boldsymbol{w}[k-1] + \boldsymbol{x}^{(i)\top} \boldsymbol{x}^{(i)} + 2\boldsymbol{w}[k-1]^\top \boldsymbol{x}^{(i)} \\ &\leq \|\boldsymbol{w}[k-1]\|^2 + L^2, \end{aligned}$$

because $\boldsymbol{w}[k-1]^\top \boldsymbol{x}^{(i)} \leq 0$ and $\|\boldsymbol{x}^{(i)}\|^2 \leq L^2$. Suppose that for the $(k-1)$-st update, the input vector $\boldsymbol{x}^{(j)}$ was used, that is,

$$\boldsymbol{w}[k-1] = \boldsymbol{w}[k-2] + \boldsymbol{x}^{(j)}.$$

Hence,

$$\begin{aligned} \|\boldsymbol{w}[k]\|^2 &\leq \|\boldsymbol{w}[k-1]\|^2 + L^2 \\ &= \|\boldsymbol{w}[k-2] + \boldsymbol{x}^{(j)}\|^2 + L^2 \\ &\leq \|\boldsymbol{w}[k-2]\|^2 + 2L^2. \end{aligned}$$

Continuing in this manner, we obtain,

$$\|\boldsymbol{w}[k]\|^2 \leq kL^2, \tag{16.7}$$

because by assumption, $\boldsymbol{w}(0) = \mathbf{0}$.

We next derive a lower bound on $\|\boldsymbol{w}[k]\|$. We have,

$$\begin{aligned} \langle \boldsymbol{w}^*, \boldsymbol{w}[k] \rangle &= \langle \boldsymbol{w}^*, \boldsymbol{w}[k-1] \rangle + \left\langle \boldsymbol{w}^*, \boldsymbol{x}^{(i)} \right\rangle \\ &\geq \langle \boldsymbol{w}^*, \boldsymbol{w}[k-1] \rangle + \delta, \end{aligned}$$

because we assumed that $\langle \boldsymbol{w}^*, \boldsymbol{x}^{(i)} \rangle > \delta$. Observing that $\boldsymbol{w}[k-1] = \boldsymbol{w}[k-2] + \boldsymbol{x}^{(j)}$, we get

$$\begin{aligned} \langle \boldsymbol{w}^*, \boldsymbol{w}[k] \rangle &\geq \langle \boldsymbol{w}^*, \boldsymbol{w}[k-1] \rangle + \delta \\ &\geq \langle \boldsymbol{w}^*, \boldsymbol{w}[k-2] \rangle + 2\delta. \end{aligned}$$

Continuing in this manner, we obtain

$$\langle \boldsymbol{w}^*, \boldsymbol{w}[k] \rangle \geq k\delta. \tag{16.8}$$

Applying to the above the Cauchy-Schwarz inequality yields

$$\|\boldsymbol{w}^*\| \|\boldsymbol{w}[k]\| \geq \langle \boldsymbol{w}^*, \boldsymbol{w}[k] \rangle \geq k\delta.$$

By assumption $\|\boldsymbol{w}^*\| = 1$. Hence,

$$\|\boldsymbol{w}[k]\| \geq k\delta.$$

Combining the obtained upper and lower bounds on $\|\boldsymbol{w}[k]\|^2$, given by (16.7) and (16.8), we have

$$k^2\delta^2 \leq \|\boldsymbol{w}[k]\|^2 \leq kL^2.$$

Therefore,

$$k \leq \frac{L^2}{\delta^2}, \tag{16.9}$$

which implies that the TLU learning algorithm converges. The proof of the TLU convergence theorem is complete.

□

We now present a sufficient condition for linear nonseparability.

Theorem 16.2 *If there exists a positive linear combination (PLC) that equals* $\mathbf{0}$ *of the given set of input vectors, then the given set of input vectors is linearly nonseparable.*

Proof We prove the theorem by contradiction. Assume that there exists a PLC of the given input vectors that is equal to $\mathbf{0}$, that is, there exist nonnegative constants, q_i, such that for some j, $1 \leq j \leq p$, we have $q_j > 0$, and

$$\sum_{i=1}^{p} q_i \boldsymbol{x}^{(i)} = \mathbf{0}.$$

Suppose now that the given input vectors are linearly separable, that is, there is a weight vector, $\boldsymbol{w}$, such that $\left\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \right\rangle > 0$ for $i = 1, \ldots, p$. We premultiply $\sum_{i=1}^{p} q_i \boldsymbol{x}^{(i)} = \mathbf{0}$ by $\boldsymbol{w}^\top$ to get

$$\boldsymbol{w}^\top \sum_{i=1}^{p} q_i \boldsymbol{x}^{(i)} = \sum_{i=1}^{p} q_i \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle > 0$$

since $\left\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \right\rangle > 0$ and $q_j > 0$. However, we arrived at a contradiction because by assumption $\sum_{i=1}^{p} q_i \boldsymbol{x}^{(i)} = \mathbf{0}$, which completes the proof.

□

Example 16.2 Suppose we are given the following training set:

$$\begin{aligned}
\boldsymbol{x}^{(1)} &= \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \; d_1 = -1; \quad \boldsymbol{x}^{(2)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; d_2 = -1; \\
\boldsymbol{x}^{(3)} &= \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \; d_3 = 1; \quad \boldsymbol{x}^{(4)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \; d_4 = 1.
\end{aligned}$$

The preceding training set represents the 2-input Exclusive-OR (XOR) function. Let us now preprocess the training set so we can apply Theorem 16.2. By preprocessing we mean eliminating the threshold by augmenting the input vectors with -1 and then negating the input vectors for which the desired output is -1. The preprocessed input vectors are:

$$\tilde{\boldsymbol{x}}^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \tilde{\boldsymbol{x}}^{(2)} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}, \quad \tilde{\boldsymbol{x}}^{(3)} = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}, \quad \tilde{\boldsymbol{x}}^{(4)} = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}.$$

Note that $\sum_{i=1}^{4} \tilde{\boldsymbol{x}}^{(i)} = \boldsymbol{0}$. It follows from Theorem 16.2 that the given input vectors are not linearly separable.

16.2 Backpropagation

In Section 16.1, we discussed an iterative learning algorithm for a threshold logic unit. This computational unit keeps on iterating until it reaches a solution, if a solution exists. In this section, we present iterative learning algorithms for computational units with differentiable activation functions. This family of algorithms are called the backpropagation algorithms because they back propagate the error between the desired and actual output of the neural network for the purpose of adjusting the network's weights in order to reduce the learning error. Specifically, a typical backpropagation algorithm consists of two passes: the forward pass during which the input vectors are presented to the network and the actual outputs of the network are calculated. Then, the error between the actual and desired outputs are calculated. In the reverse pass, the error gradient with respect to the network's weights is calculated by propagating the error backwards through the network. Once the error gradient is calculated, the weights are adjusted using, for example, a descent gradient method given by (13.31) in Subsection 13.3.1.

We first analyze a single computational unit shown in Figure 16.6. It is called the *perceptron*. We consider a simple case, where we train the perceptron to learn the training pair

$$(\boldsymbol{x}, d),$$

where $\boldsymbol{x} \in \mathbb{R}^{n+1}$, $x_{n+1} = (\boldsymbol{x})_{n+1} = -1$, and $d \in [-1, 1]$. Define the error function

$$\begin{aligned} E(\boldsymbol{w}) &= \frac{1}{2}(d-y)^2 \\ &= \frac{1}{2}(d - f(net))^2, \end{aligned} \tag{16.10}$$

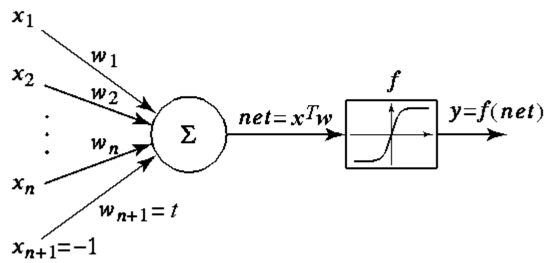


Figure 16.6: A single perceptron with a differentiable sigmoidal activation function.

where $net = \boldsymbol{x}^\top \boldsymbol{w}$. Note that $w_{n+1} = t$, where t is a threshold, bias. The gradient of E, with respect to $\boldsymbol{w}$, is

$$\nabla E(\boldsymbol{w}) = -(d - f(net))f^{'}(net)\boldsymbol{x}. \tag{16.11}$$

Using the descent gradient algorithm with a fixed step size to minimize E, we obtain

$$\begin{aligned} \boldsymbol{w}[k+1] &= \boldsymbol{w}[k] - \alpha \nabla E\left(\boldsymbol{w}[k]\right) \\ &= \boldsymbol{w}[k] + \alpha \left(d_k - f\left(net_k\right)\right) f^{'}\left(net_k\right) \boldsymbol{x}[k], \end{aligned} \tag{16.12}$$

where $net_k = \boldsymbol{x}[k]^\top \boldsymbol{w}[k]$. Let

$$\delta^k = \left(d_k - f\left(net_k\right)\right) f^{'}\left(net_k\right). \tag{16.13}$$

Then, we represent (16.12) as

$$\boxed{\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + \alpha \delta^k \boldsymbol{x}[k]} \tag{16.14}$$

We generalize the above formula for a three-layer feedforward neural network shown in Figure 16.7. We wish to train the network to learn the training pairs

$$\left(\boldsymbol{x}^{(1)}, \boldsymbol{d}^{(1)}\right), \left(\boldsymbol{x}^{(2)}, \boldsymbol{d}^{(2)}\right), \ldots, \left(\boldsymbol{x}^{(p)}, \boldsymbol{d}^{(p)}\right),$$

where $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$ and $\boldsymbol{d}^{(i)} \in \mathbb{R}^m$. The network has n inputs, which is the dimension of the training vectors, and m outputs. The network's inputs are connected to a layer of hidden

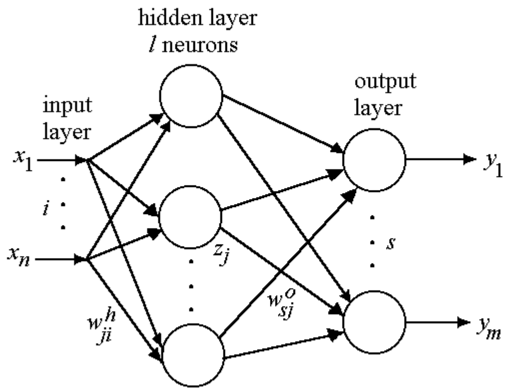


Figure 16.7: A three-layer feedforward neural network.

units, which in turn is connected to a layer of output units. To construct a network to perform some task, we must set the weights of the connections appropriately. The weights specify the strength of the influence of one unit upon another. We derive rules for adjusting the weights w_{ji} and w_{kj} such that the error function

$$\begin{aligned} E(\boldsymbol{w}) &= \frac{1}{2}\sum_{s=1}^{m}(d_s - y_s)^2 \\ &= \frac{1}{2}\sum_{s=1}^{m}(d_s - f_o(net_s))^2 \end{aligned}$$

is minimized, where

$$net_s = \sum_{j=1}^{l} w^o_{sj} z_j. \tag{16.15}$$

The minimization is performed over the space of weights

$$\left\{w^h_{ji}, w^o_{sj} : i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, l,\ s = 1, 2, \ldots, m\right\}.$$

We first give formulas for updating the weights connecting the hidden and output layers.

Using the descent gradient algorithm in a similar way as in (16.14), we obtain

$$\begin{aligned} w^o_{sj}[k+1] &= w^o_{sj}[k] - \alpha_o \frac{\partial E\left(\boldsymbol{w}[k]\right)}{\partial w^o_{sj}[k]} \\ &= w^o_{sj}[k] + \alpha_o \left(d_s - y_s\right) f'_o(net_s) z_j[k], \end{aligned} \tag{16.16}$$

where f_o' is the derivative of $f_o(net_s)$ with respect to net_s, and the values of z_j are computed by feeding into the network the input vector $\boldsymbol{x}$ and then propagating it forward through the network. We have

$$z_j = f_h\left(\sum_{i=1}^{n} w_{ji}^h x_i\right) = f_h(net_j), \quad j = 1, 2, \ldots, l.$$

To derive the update formulas for the weights in the hidden layer, we use the chain rule to compute $\partial E/\partial w_{ji}^h$, and then apply the gradient descent algorithm to obtain

$$\frac{\partial E}{\partial w_{ji}^h} = \frac{\partial E}{\partial z_j}\frac{\partial z_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}^h}, \tag{16.17}$$

where

$$\frac{\partial net_j}{\partial w_{ji}^h} = x_i, \tag{16.18}$$

$$\frac{\partial z_j}{\partial net_j} = f_h'(net_j). \tag{16.19}$$

Finally, using (16.15) gives

$$\begin{aligned}
\frac{\partial E}{\partial z_j} &= \frac{\partial}{\partial z_j}\left(\frac{1}{2}\sum_{s=1}^{m}\left(d_s - f_o\left(net_s\right)\right)^2\right) \\
&= -\sum_{s=1}^{m}\left(d_s - f_o\left(net_s\right)\right)\frac{\partial f_o\left(net_s\right)}{\partial z_j} \\
&= -\sum_{s=1}^{m}\left(d_s - y_s\right) f_o'\left(net_s\right) w_{sj}^o. \qquad (16.20)
\end{aligned}$$

Combining (16.17)–(16.20), we obtain rules for updating the hidden weights,

$$\begin{aligned}
w_{ji}^h[k+1] &= w_{ji}^h[k] - \alpha_h \frac{\partial E\left(\boldsymbol{w}[k]\right)}{\partial w_{ji}^h[k]} \\
&= w_{ji}^h[k] + \alpha_h\left(\sum_{s=1}^{m}\left(d_s - y_s\right) f_o'\left(net_s\right) w_{sj}^o[k]\right) f_h'(net_j)x_i. \qquad (16.21)
\end{aligned}$$

A commonly used activation function is the sigmoid,

$$f(v) = \frac{1}{1+e^{-v}}. \tag{16.22}$$

It is easy to check that the activation function (16.22) has the useful property,

$$\frac{df(v)}{dv} = f(v)(1 - f(v)). \tag{16.23}$$

We summarize the above derivations in the following algorithm.

Algorithm 16.2 Backpropagation Training

Given a set of training pairs $\left\{\left(\boldsymbol{x}^{(1)}, \boldsymbol{d}^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(i)}, \boldsymbol{d}^{(i)}\right), \ldots, \left(\boldsymbol{x}^{(p)}, \boldsymbol{d}^{(p)}\right)\right\}$.

1. **Initialize Weights and Thresholds**
   Initialize the weights and thresholds to small random values.

2. **Present New Input Vector and Desired Output**
   Present the new input vector, $\boldsymbol{x}^{(i)}$ and the corresponding desired output $\boldsymbol{d}^{(i)}$. Calculate actual outputs.

3. **Calculate Error Gradients**

4. **Adapt Weight Vectors and Thresholds**

$$\begin{aligned}
\boldsymbol{w}[k+1] &= \boldsymbol{w}[k] - \alpha \nabla_{\boldsymbol{w}} E\left(\boldsymbol{w}[k], \boldsymbol{t}[k]\right), \\
\boldsymbol{t}[k+1] &= \boldsymbol{t}[k] - \alpha \nabla_{\boldsymbol{t}} E\left(\boldsymbol{w}[k], \boldsymbol{t}[k]\right).
\end{aligned}$$

5. **Repeat by Returning to Step 2**

Example 16.3 We consider the two-input XOR problem. We showed, in Example 16.2, that this logic function cannot be implemented using a single TLU. Here, we use a two layer neural network with two nodes in the hidden layer and one node in the output layer as shown in Figure 16.8.

In this example, we use the unipolar representation of the two-input XOR function, that is, the elements of the training pairs are from the set $\{0, 1\}$. Therefore, the input vectors and the corresponding desired outputs have the form

$$\begin{aligned}
\boldsymbol{x}^{(1)} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ d_1 = 0; \quad \boldsymbol{x}^{(2)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ d_2 = 0; \\
\boldsymbol{x}^{(3)} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ d_3 = 1; \quad \boldsymbol{x}^{(4)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ d_4 = 1.
\end{aligned}$$

The error function for this example takes the form

$$E = \frac{1}{2}(d_i - y_i)^2.$$

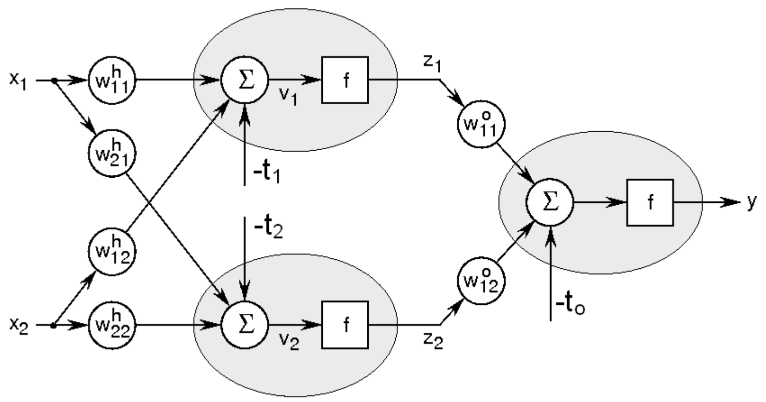


Figure 16.8: A two-layer feedforward neural network for solving the XOR problem.

The activation function for all three neurons is the sigmoid given by (16.22). The outputs of the hidden neurons are

$$z_1 = f(w_{11}^h x_1^{(i)} + w_{12}^h x_2^{(i)} - t_1) \tag{16.24}$$
$$z_2 = f(w_{21}^h x_1^{(i)} + w_{22}^h x_2^{(i)} - t_2) \tag{16.25}$$

The network output is

$$y = f(w_{11}^o z_1 + w_{12}^o z_2 - t_o) \tag{16.26}$$

The threshold parameters are treated as learning weights. Thus, the above simple network requires updating of nine weights, where, in particular,

$$\begin{aligned}
\frac{\partial E}{\partial t_1} &= (d_i - y)y' w_{11}^o z_1' = (d_i - y)y(1-y)w_{11}^o z_1(1-z_1) \\
\frac{\partial E}{\partial t_2} &= (d_i - y)y' w_{12}^o z_2' = (d_i - y)y(1-y)w_{12}^o z_2(1-z_2) \\
\frac{\partial E}{\partial t_o} &= (d_i - y)y' = (d_i - y)y(1-y).
\end{aligned}$$

In calculating the above, we used (16.23). In our simulation the weights of the network were initialized to small random values in the interval $[-0.1, 0.1]$. The four patterns were presented in a mixed order. We used a fixed learning rate of 0.75 for all nine weights. After 50,000 iterations we obtained the following weight values:

$$\boldsymbol{W}^h = \begin{bmatrix} -4.9288 & -4.9334 \\ -6.7240 & -6.7526 \end{bmatrix}, \quad \boldsymbol{W}^o = \begin{bmatrix} 10.2447 & -10.7258 \end{bmatrix},$$

Table 16.1: The desired and actual responses of the trained network in Example 16.3.

x_1	x_2	d_i	y_i
0	0	0	0.0087
1	1	0	0.0162
0	1	1	0.9870
1	0	1	0.9870

and $t_1 = -7.3398$, $t_2 = -2.7853$, and $t_o = 4.8701$. In Table 16.1 we show the desired and actual responses of the trained network. In Figure 16.9 we show a plot of the function that the trained network implements.

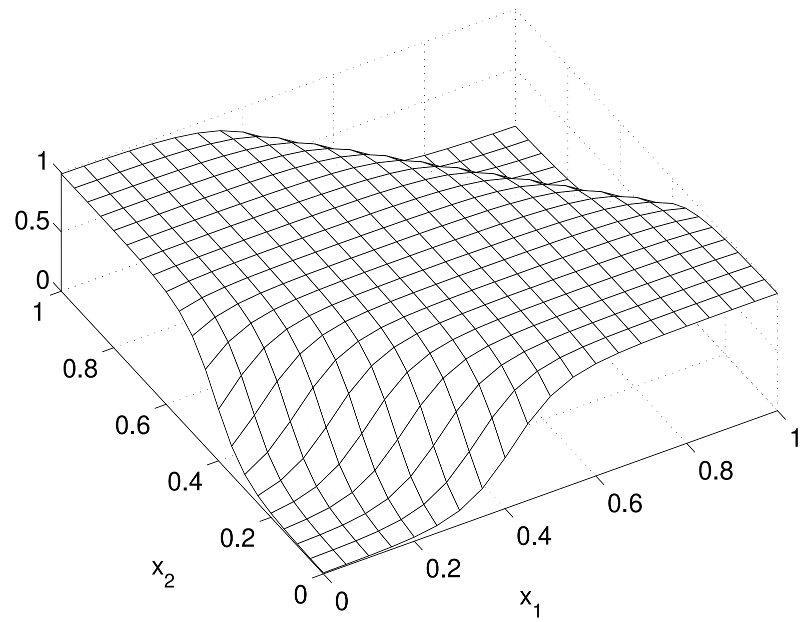


Figure 16.9: The function implemented by the trained neural network to solve the two-input XOR problem in Example 16.3.

16.3 Neural Fuzzy Identifier

In this section, we describe a method that can be used to construct identifiers of dynamical systems that in turn could be employed to construct neural-network-based fuzzy logic control strategies. The idea behind the method is to apply the backpropagation algorithm to a fuzzy logic system.

It follows from our discussion in Section 15.2 that a typical fuzzy logic system can be thought of as a nonlinear system described by

$$f(x_1, \ldots, x_n) = f(\boldsymbol{x}) = \frac{\sum_{l=1}^{M} v_l \left(\prod_{i=1}^{n} \mu_{il}(x_i)\right)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{il}(x_i)}, \tag{16.27}$$

where, for example,

$$\mu_{il} = \exp\left\{-\frac{(x_i - m_{il})^2}{2\sigma_{il}^2}\right\},$$

M is the number of IF-THEN fuzzy rules, $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top$, and m_{il} and σ_{il} are the parameters of the Gaussian membership functions, while v_l is a weight parameter. These parameters are determined using the backpropagation algorithm discussed in the previous section. To facilitate our discussion, we represent (16.27), following Wang [345, page 169], as a feedforward network shown in Figure 16.10. The variables that characterize the feedforward network depicted in Figure 16.10 satisfy the following relations:

$$\left.\begin{array}{rcl} z_l & = & \prod_{i=1}^{n} \exp\left\{-\frac{(x_i - m_{il})^2}{2\sigma_{il}^2}\right\}, \quad l = 1, 2 \ldots, M \\ b & = & \sum_{l=1}^{M} z_l \\ a & = & \sum_{l=1}^{M} v_l z_l \\ f & = & a/b \end{array}\right\} \tag{16.28}$$

We then use this feedforward network as a component of an identifier of dynamical systems. The identification scheme involving (16.27) is depicted in Figure 16.11. The adaptation algorithm for the identifier parameters is derived using the backpropagation algorithm. The cost function to be minimized is

$$E = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(y - f)^2,$$

where y is the output of the identified plant and f is the output of the identifier. We use a gradient algorithm to minimize E over the space of parameters

$$\boldsymbol{w} = \{v_l,\ \sigma_{il},\ m_{il}\}.$$

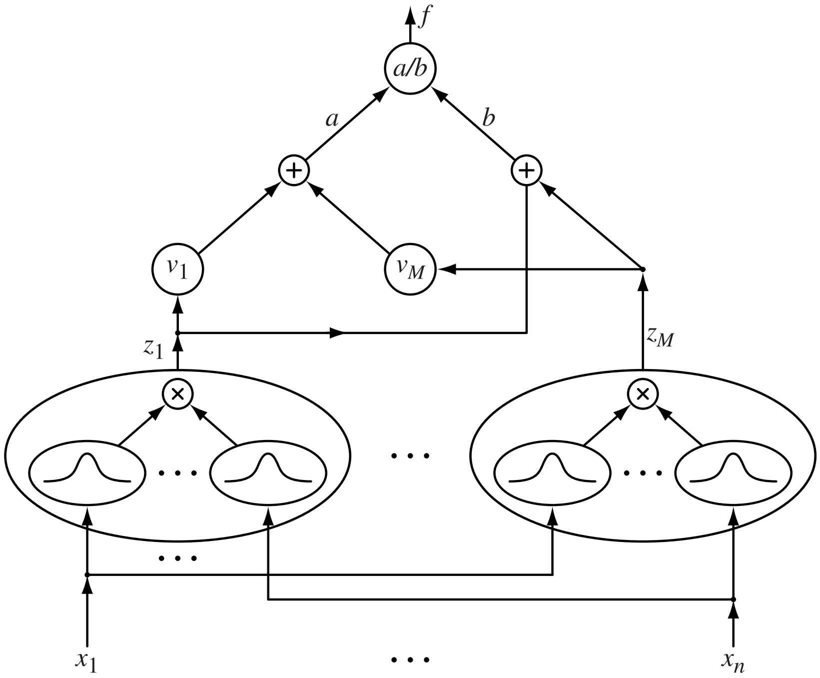


Figure 16.10: Fuzzy logic system (16.27) represented as a feedforward network.

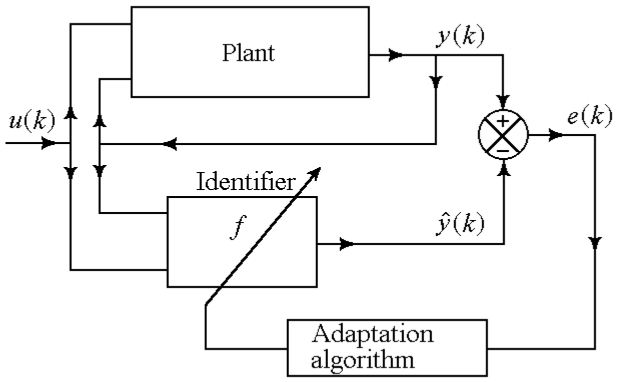


Figure 16.11: Neural fuzzy identification scheme of a nonlinear dynamical system.

The gradient algorithm that we use has the form

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] - \alpha \nabla E(\boldsymbol{w}[k]), \tag{16.29}$$

where α is a fixed step size. Note that (16.29) is just a simple descent gradient algorithm that we used in our discussion of the backpropagation algorithm.

In order to be able to use the above algorithm, we need to compute partial derivatives of E with respect to each component of $\boldsymbol{w}$. For this we first fix indices i, and l. We next compute the partial derivatives of E with respect to v_j. Then, the parameters v_j will be adapted accordingly as

$$v_j[k+1] = v_j[k] - \alpha \frac{\partial E(\boldsymbol{w}[k])}{\partial v_j}.$$

We note that E depends upon v_j only through a, where $f = a/b$. Hence, using the chain rule, we obtain

$$\begin{aligned}
\frac{\partial E}{\partial v_j}(\boldsymbol{w}) &= \frac{\partial}{\partial v_j}\left(\frac{1}{2}(y-f)^2\right) \\
&= \frac{\partial}{\partial v_j}\left(\frac{1}{2}(y-a/b)^2\right) \\
&= -\frac{(y-f)}{b}\frac{\partial a}{\partial v_j} \\
&= \frac{f-y}{b} z_j.
\end{aligned}$$

We next compute the partial derivative of E with respect to m_{st}. Observe that E depends upon m_{st} only through z_t. Using the chain rule again gives

$$\begin{aligned}
\frac{\partial E}{\partial m_{st}}(\boldsymbol{w}) &= -(y-f)\frac{\partial f}{\partial z_t}\frac{\partial z_t}{\partial m_{st}} \\
&= -(y-f)\left(\frac{v_t b - a}{b^2}\right) z_t \frac{(x_s - m_{st})}{\sigma_{st}^2} \\
&= (f-y)\left(\frac{v_t - f}{b}\right) z_t \frac{(x_s - m_{st})}{\sigma_{st}^2}.
\end{aligned}$$

Hence, the adaptation formula for m_{st} takes the form

$$m_{st}[k+1] = m_{st}[k] - \alpha(f-y)\left(\frac{v_t - f}{b}\right) z_t \frac{(x_s - m_{st}[k])}{\sigma_{st}^2[k]}.$$

To find the partial derivative of E with respect to σ_{st}, we note that E depends upon σ_{st} only through z_t. Applying the chain rule gives

$$\begin{aligned}
\frac{\partial E}{\partial \sigma_{st}}(\boldsymbol{w}) &= -(y-f)\frac{\partial f}{\partial z_t}\frac{\partial z_t}{\partial \sigma_{st}} \\
&= (f-y)\left(\frac{v_t-f}{b}\right) z_t \frac{(x_i-m_{st})^2}{\sigma_{st}^3}.
\end{aligned}$$

We are now ready to formulate an algorithm for updating on-line the weights of the neural fuzzy identifier.

Algorithm 16.3 On-line Identification

1. **Initialize the identifier parameters**
   Set the values of v_l, m_{il}, and σ_{il} to small random values.

2. **Calculate the output of the identifier at the time instant k**
   Compute

$$\begin{aligned}
z_l &= \prod_{i=1}^{n} \exp\left(-\frac{(x_i-m_{il})^2}{2\sigma_{il}^2}\right), \\
b &= \sum_{l=1}^{M} z_l, \\
a &= \sum_{l=1}^{M} v_l z_l, \\
f &= a/b;
\end{aligned}$$

3. **Calculate the identification error**

$$e[k] = y[k] - f[k];$$

4. **Adapt the identifier parameters**

$$\begin{aligned}
v_l[k+1] &= v_l[k] + \alpha\frac{e[k]}{b[k]} z_l[k], \\
m_{il}[k+1] &= m_{il}[k] + \alpha\frac{e[k]}{b[k]}\left(v_l[k]-f[k]\right) z_l[k]\frac{(x_i[k]-m_{il}[k])}{\sigma_{il}^2[k]}, \\
\sigma_{il}[k+1] &= \sigma_{il}[k] + \alpha\frac{e[k]}{b[k]}\left(v_l[k]-f[k]\right) z_l[k]\frac{(x_i[k]-m_{il}[k])^2}{\sigma_{il}^3[k]}, \\
k &= k+1;
\end{aligned}$$

5. **Repeat by Returning to Step 2**

Example 16.4 The plant to be identified is the same as that considered by Wang [345, page 175], and is described by

$$y[k+1] = 0.3y[k] + 0.6y[k-1] + g(u[k]),$$

where the "unknown" function $g(u)$ has the form

$$g(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u),$$

and $u[k] = \sin(2\pi k/200)$. The identifier has the structure shown in Figure 16.11 and is described by

$$\hat{y}[k+1] = 0.3y[k] + 0.6y[k-1] + f(u[k]),$$

where f has the form of (16.27) with $M = 10$. We chose $\alpha = 0.5$, $y[0] = y[1] = 0$, $f[1] = f[2] = -3$, and we started with $u[1] = 0$. The identifier's parameters were initialized in a random fashion using MATLAB's command `rand`. After the identifier was trained, we tested its prediction power using the same input signal as before and the initial conditions $y[0] = y[1] = 0$, $f[1] = f[2] = -3$, and $u[1] = 0$. The results of this experiment are shown in Figure 16.13.

16.4 Radial-Basis Function (RBF) Networks

In this section, we present radial-basis function (RBF) neural networks and their applications to interpolation and then to identification of unknown nonlinear dynamical systems using only the systems' inputs and outputs data. We first show how an RBF network can be used for interpolation purposes. Then, we concern ourselves with the problem of identification of dynamical systems. Our exposition of using RBF neural networks to identify dynamical systems is based on the paper by Liu, Kadirkamanathan, and Billings [218]. The RBF neural identifier is an adaptive system for which the learning algorithm is developed using Lyapunov techniques. The algorithm can add new nodes to the neural identifier to obtain the desired identification accuracy.

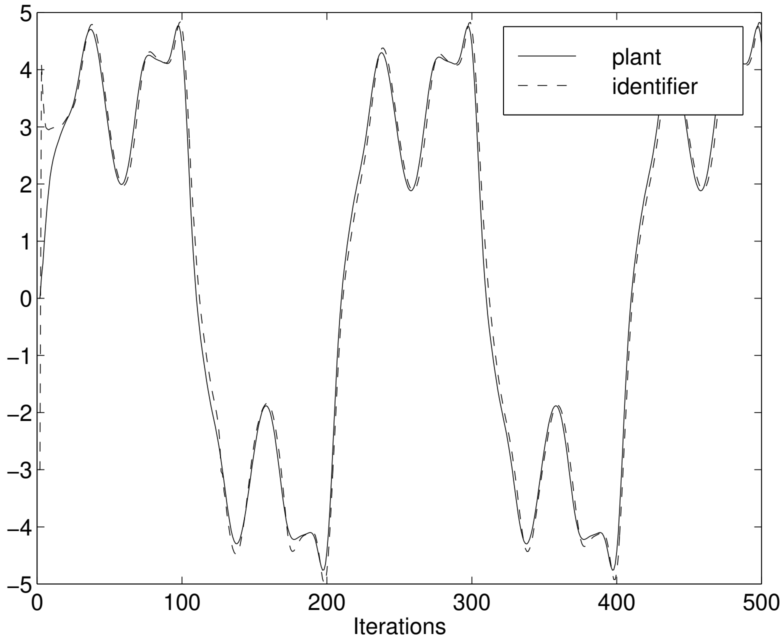


Figure 16.12: Plots of the outputs of the plant and its identifier during the learning phase in Example 16.4.

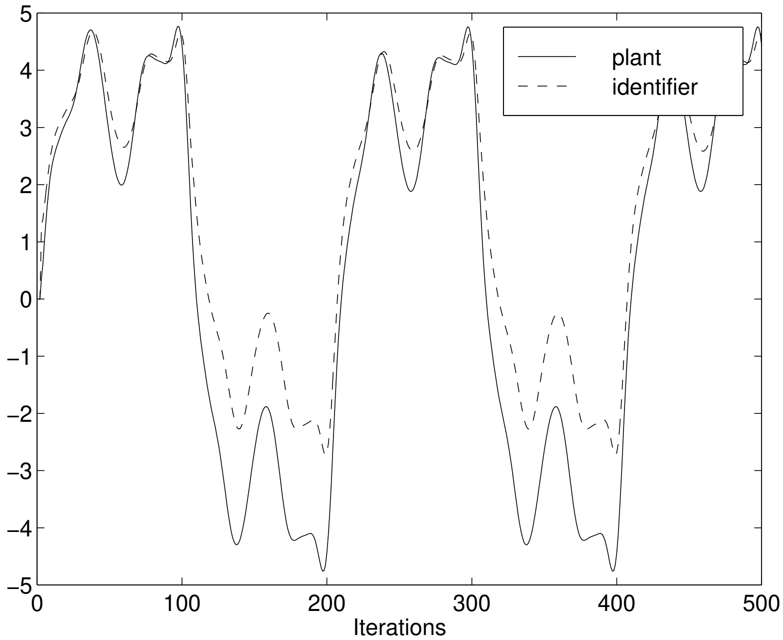


Figure 16.13: Plots of the outputs of the plant and its identifier during the testing phase in Example 16.4.

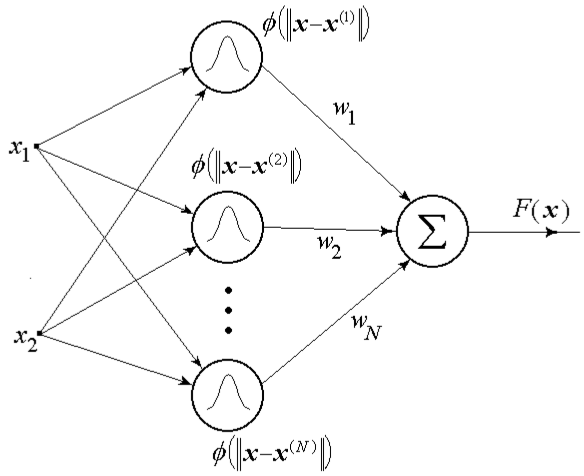


Figure 16.14: A two-input radial-basis function (RBF) network.

16.4.1 Interpolation Using RBF Networks

A diagram of a radial-basis function (RBF) network is shown in Figure 16.14. The functions $\phi\left(\left\|\boldsymbol{x}-\boldsymbol{x}^{(i)}\right\|\right)$, $i=1,2,\ldots,N$, are called the radial-basis functions, where $\|\cdot\|$ denotes a p-norm that is usually the Euclidean 2-norm. The real parameters w_i, $i=1,2,\ldots,N$, are the weights. A common choice for ϕ is

$$\phi\left(\left\|\boldsymbol{x}-\boldsymbol{x}^{(i)}\right\|\right)=\exp\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}^{(i)}\right\|^2}{2\sigma_i^2}\right), \tag{16.30}$$

where $\boldsymbol{x}^{(i)}$ is the center of the basis function and σ_i is sometimes called the radius of the radial basis function. The function given by (16.30) is called the Gaussian radial basis function.

The RBF networks are used to perform interpolation. The interpolation problem can be formulated as: For a given set of N different points,

$$\left\{\boldsymbol{x}^{(i)}:\boldsymbol{x}^{(i)}\in\mathbb{R}^n,\ i=1,2,\ldots,N\right\}$$

and a corresponding set of N real numbers

$$\{d_i: d_i\in\mathbb{R},\ i=1,2,\ldots,N\}$$

find a function

$$F:\mathbb{R}^n\to\mathbb{R}$$

such that

$$F\left(\boldsymbol{x}^{(i)}\right) = d_i,\ i = 1, 2, \ldots, N. \tag{16.31}$$

We attempt to solve the interpolation problem using the RBF network. The function that the RBF network implements is

$$F(\boldsymbol{x}) = \sum_{j=1}^{N} w_j \phi\left(\left\|\boldsymbol{x} - \boldsymbol{x}^{(j)}\right\|\right). \tag{16.32}$$

Substituting the interpolation conditions given by (16.31) into (16.32) gives

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N1} & \phi_{N2} & \cdots & \phi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}, \tag{16.33}$$

where $\phi_{ij} = \phi\left(\left\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right\|\right)$. Let $\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}$, $\boldsymbol{d} = \begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix}$, and

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N1} & \phi_{N2} & \cdots & \phi_{NN} \end{bmatrix}.$$

Using the above notation, we represent (16.33) as

$$\boldsymbol{\Phi w} = \boldsymbol{d}. \tag{16.34}$$

The matrix $\boldsymbol{\Phi}$ is called the *interpolation matrix*. If the interpolation matrix is nonsingular, then we can solve (16.34) for $\boldsymbol{w}$ to obtain

$$\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\boldsymbol{d}. \tag{16.35}$$

Example 16.5 We construct an RBF network that interpolates data given in Table 16.2. We have, $\phi_{ij} = \exp\left(-\frac{(x^{(i)}-x^{(j)})^2}{2\sigma_i^2}\right)$. We set $\sigma_i = 1$, for $i = 1, \ldots, 5$, and evaluate (16.35) to obtain

$$\boldsymbol{w} = \begin{bmatrix} 0.8000 & 0.1459 & -0.0390 & 0.5737 & 0.2227 \end{bmatrix}^\top.$$

The interpolation result is shown in Figure 16.15. In Figure (16.15)(a), we used a solid line to show the result of interpolating using the Gaussian radial-radial basis functions, while a broken line indicates the result of the straight line interpolation. The plots of the basis functions that contribute to the interpolating curve are shown in Figure (16.15)(b).

Table 16.2: Interpolation data for Example 16.5.

$x^{(i)}$	d_i
0	0.8
6	0.2
7	0.4
8	0.6
10	0.3

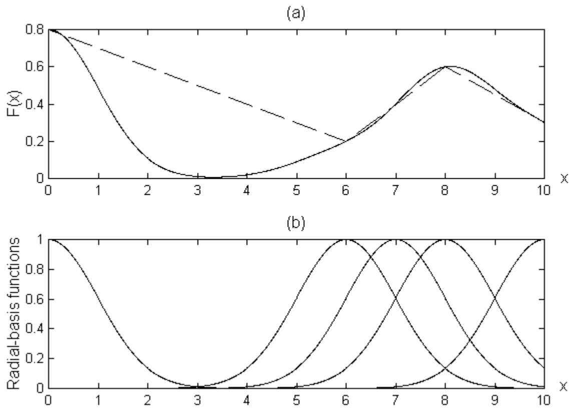


Figure 16.15: Results of interpolating data from Table 16.2 using radial-basis functions.

If the number of radial basis functions was fixed and the rows of the matrix $\mathbf{\Phi}$ and the elements of the vector $\boldsymbol{d}$ were obtained sequentially, we could use the recursive least squares algorithm, described in Section A.11, to compute the least squares estimate of $\boldsymbol{w}$.

16.4.2 Identification of a Single-Input Single-State System

Here, we discuss the identification of a single-input single-state dynamical system using neural network. We consider a dynamical system modeled by

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \tag{16.36}$$

where $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is an unknown, possibly nonlinear, function of x and u. We will construct a neural network to identify the function f. Adding and subtracting $ax(t)$, where $a > 0$ is a design parameter, to the right-hand side of (16.36) gives

$$\begin{aligned} \dot{x}(t) &= -ax(t) + ax(t) + f(x(t), u(t)) \\ &= -ax(t) + g(x(t), u(t)), \end{aligned} \tag{16.37}$$

where $g(x(t), u(t)) = ax(t) + f(x(t), u(t))$. The reason for the above manipulations will become apparent when we discuss learning rules of the neural network identifier of the dynamical system (16.36). This identifier is described by

$$\dot{\hat{x}}(t) = -a\hat{x}(t) + \hat{g}(x(t), u(t), \boldsymbol{m}_k, \sigma_k), \quad \hat{x}(0) = \hat{x}_0, \tag{16.38}$$

where $\hat{x}$ is the state of the identification network, $\boldsymbol{m}_k$ and σ_k are adjustable parameters of the identifier. The function $\hat{g}$ has the form,

$$\hat{g}(x(t), u(t), \boldsymbol{m}_k, \sigma_k) = \sum_{k=1}^{K} w_k \phi(x, u, \boldsymbol{m}_k, \sigma_k), \tag{16.39}$$

where for $k = 1, 2, \ldots, K$, we have $\boldsymbol{m}_k = \begin{bmatrix} m_{xk} & m_{uk} \end{bmatrix}$, and

$$\phi(x(t), u(t), \boldsymbol{m}_k, \sigma_k) = \exp\left(-\frac{1}{2\sigma_k^2} \left((x(t) - m_{xk})^2 + (u(t) - m_{uk})^2 \right) \right).$$

Thus, we postulate that the function $\hat{g}$ be generated using an RBF network.

Suppose that the basis functions are given to us. The problem of choosing basis functions is discussed later. Suppose also that we know the "optimal" weights w_k^*, $k = 1, 2, \ldots, K$,

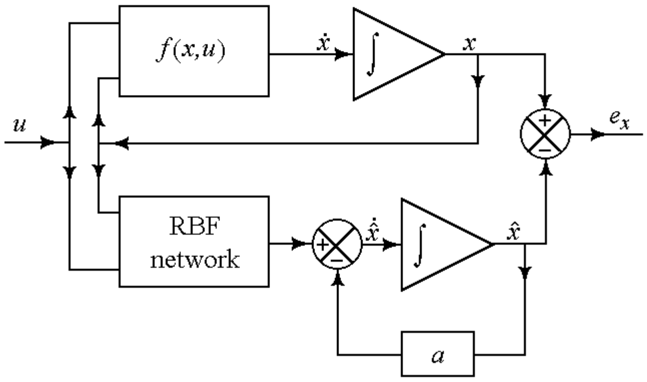


Figure 16.16: A configuration for the radial-basis function neural identifier.

that result in the least identification error between g in (16.37) and $\hat{g}$ in (16.38), where the identification error is defined as

$$e(t) = g(x(t), u(t)) - \sum_{k=1}^{K} w_k^* \phi(x(t), u(t), \boldsymbol{m}_k, \sigma_k). \tag{16.40}$$

Using the above notation, we represent the dynamical system model (16.37) in the form

$$\dot{x}(t) = -ax(t) + \sum_{k=1}^{K} w_k^* \phi(x(t), u(t), \boldsymbol{m}_k, \sigma_k) + e(t). \tag{16.41}$$

The neural identifier, on the other hand, can be represented as

$$\dot{\hat{x}}(t) = -a\hat{x}(t) + \sum_{k=1}^{K} w_k \phi(x(t), u(t), \boldsymbol{m}_k, \sigma_k). \tag{16.42}$$

A schematic of the above identification architecture is depicted in Figure 16.16.

16.4.3 Learning Algorithm for the RBF Identifier

We assume that both the state x and the input u of the dynamical system model (16.36), and hence that of (16.37), are bounded. We define the state estimation error to be

$$e_x = x - \hat{x},$$

while the weights' estimation errors are

$$\xi_k = w_k^* - w_k, \quad k = 1, \ldots, K.$$

We assume that the modeling error, e, defined in (16.40), is bounded by a finite constant e_K, that is, $|e| \leq e_K$. However, e_K is unknown to us. Next, we use (16.41) and (16.42) to form the differential equation describing the state error dynamics. For simplicity of notation, we omit the argument t of the time-varying functions. We have

$$\begin{aligned} \dot{e}_x &= \dot{x} - \dot{\hat{x}} \\ &= -ax + \sum_{k=1}^{K} w_k^* \phi(x, u, \boldsymbol{m}_k, \sigma_k) + e + a\hat{x} - \sum_{k=1}^{K} w_k \phi(x, u, \boldsymbol{m}_k, \sigma_k) \\ &= -ae_x + \sum_{k=1}^{K} \xi_k \phi(x, u, \boldsymbol{m}_k, \sigma_k) + e. \end{aligned} \tag{16.43}$$

We now use the second method of Lyapunov to devise a learning law for the RBF identifier that would yield a stable identification scheme in the sense that the state estimation error e_x decreases as the time increases and the weights do not diverge. For this purpose, we use the following Lyapunov function candidate,

$$V(e_x, \boldsymbol{\xi}) = \frac{1}{2}\left(e_x^2 + \frac{1}{\alpha}\sum_{k=1}^{K} \xi_k^2\right), \tag{16.44}$$

where $\alpha > 0$ is a design parameter and $\boldsymbol{\xi} = \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_K \end{bmatrix}^\top$. The time derivative of V evaluated on the trajectories of (16.43) is

$$\begin{aligned} \dot{V} &= e_x \dot{e}_x + \frac{1}{\alpha}\sum_{k=1}^{K} \xi_k \dot{\xi}_k \\ &= e_x\left(-ae_x + \sum_{k=1}^{K} \xi_k \phi(x, u, \boldsymbol{m}_k, \sigma_k) + e\right) + \frac{1}{\alpha}\sum_{k=1}^{K} \xi_k \dot{\xi}_k \\ &= -ae_x^2 + \frac{1}{\alpha}\sum_{k=1}^{K} \xi_k \left(\alpha e_x \phi(x, u, \boldsymbol{m}_k, \sigma_k) + \dot{\xi}_k\right) + e_x e. \end{aligned} \tag{16.45}$$

Let

$$\dot{\xi}_k = -\alpha e_x \phi(x, u, \boldsymbol{m}_k, \sigma_k). \tag{16.46}$$

Then (16.45) simplifies to

$$\dot{V} = -ae_x^2 + e_x e. \tag{16.47}$$

Because $\dot{w}_k = -\dot{\xi}_k$, we can represent (16.46) as

$$\boxed{\dot{w}_k = \alpha e_x \phi(x, u, \boldsymbol{m}_k, \sigma_k)} \tag{16.48}$$

We further evaluate (16.47) to obtain

$$\begin{aligned}\dot{V} &\leq -a e_x^2 + |e_x|\, e_K \\ &= -a\,|e_x| \left(|e_x| - \frac{e_K}{a}\right)\end{aligned} \tag{16.49}$$

It follows from (16.47) that if the state estimation error is zero, that is, $e_K = 0$, then

$$\dot{V} = -a e_x^2.$$

Thus in this case, $\dot{V}$ is negative semi-definite in the $\begin{bmatrix} e_x & \boldsymbol{\xi}^\top \end{bmatrix}^\top$-space, and hence the origin of the $\begin{bmatrix} e_x & \boldsymbol{\xi}^\top \end{bmatrix}^\top$-space is stable, which guarantees boundedness of e_x and the weights w_ks. We show, using the Lyapunov-like lemma stated on page 392, that when $e = 0$, then $e_x(t) \to 0$ as $t \to \infty$. We have to check if the assumptions of the Lyapunov-like lemma are satisfied. First, observe that V is bounded below and positive semi-definite. Next, because e_x, ξ_k as well as $\phi(x, u, \boldsymbol{m}_k, \sigma_k)$ are bounded, the derivative of the state estimation error, given by (16.43) for $e = 0$, is bounded as well, which implies that

$$\ddot{V} = -2a e_x \dot{e}_x$$

is bounded. This in turn implies that $\dot{V}$ is uniformly continuous. Hence, by the Lyapunov-like lemma, $e_x(t) \to 0$ as $t \to \infty$.

In the presence of the identification error, e, if $|e_x| < e_K/a$, then it is possible that $\dot{V} > 0$, which implies that the weights w_ks may drift to infinity with time. To avoid this problem, Liu, Kadirkamanathan, and Billings [218] propose to set $\dot{w}_k = 0$ if $|e_x| < e_K/a$. Also, because e_K is unknown, a bound is imposed on the weights as well as on the error. The learning rule becomes

$$\boxed{\dot{w}_k = \begin{cases} \alpha e_x \phi(x, u, \boldsymbol{m}_k, \sigma_k) & \text{if} \quad |e_x| \geq e_0 \quad \text{and} \quad \|\boldsymbol{w}\| \leq M \\ 0 & \text{otherwise} \end{cases}} \tag{16.50}$$

where the bounds e_0 and M are design parameters, and $\boldsymbol{w} = \begin{bmatrix} w_1 & \cdots & w_K \end{bmatrix}^\top$. With the above modified learning law, whenever $|e_x| \geq e_0 \geq e_K/a$, the Lyapunov derivative $\dot{V}$ is negative semi-definite. However, when $e_0 \leq |e_x| \leq e_K/a$, the weights may increase with time because it is possible that $\dot{V} > 0$. We impose an upper-bound on the weights. We also impose an upper bound on the estimation error, e_x. When any of the two bounds is reached, more basis functions may be needed to improve the approximation of g. New basis functions are added to reduce the error until $\dot{V} < 0$. In what follows, we describe a procedure for adding radial-basis functions to the identification network.

16.4.4 Growing RBF Network

As we mentioned above, when $\dot{V} > 0$, the identification scheme is no longer stable, but the learning rule (16.50) guarantees that the adjustable parameters are bounded. Once the parameter adjustments can no longer reduce the estimation error and the bounds on the weights or on the estimation error are reached, additional basis functions in the RBF network should be added to reduce the modeling error. This continues until the approximation leads to the error reduction so that $\dot{V} < 0$.

In the proposed identification scheme, we use Gaussian radial-basis functions, whose areas of "influence" are determined by the functions' centers and radii (widths). As the function to be approximated moves further away from a particular basis function, the contribution of that basis function decreases. This is due to the exponential form of the Gaussian basis functions. As $|z|$ increases $\exp(-z^2)$ decreases. The maximum contribution of a particular basis function to approximating a given function occurs when the center of the basis function and the approximated function are at the same point because in this case we have $\exp(0) = 1$. This heuristic argument leads to the following rule for determining the centers of new basis functions to be added at a time instance t,

$$\boldsymbol{m}_{K+1} = \begin{bmatrix} m_{x,K+1} & m_{u,K+1} \end{bmatrix}^\top = \begin{bmatrix} \delta_x \text{round}\left(x(t)/\delta_x\right) & \delta_u \text{round}\left(u(t)/\delta_u\right) \end{bmatrix}^\top, \tag{16.51}$$

where the round operator rounds the elements on which it operates to the nearest integer. For example, if $\boldsymbol{x} = \begin{bmatrix} -1.8 & -0.3 & 3.7 \end{bmatrix}$, then round$(\boldsymbol{x}) = \begin{bmatrix} -2 & 0 & 4 \end{bmatrix}$. The parameters, δ_x and δ_u represent the spacing of the centers of the basis functions. We can think of having a grid over the input-state space where the spacing between the vertices of the grid are determined by the parameters δ_x and δ_u. Each vertex of the grid represents the potential center of a basis function. The parameter σ_k of the new basis function is determined from the relation,

$$\sigma_k = \kappa \left\| \boldsymbol{m}_k - \boldsymbol{m}_k^* \right\|, \tag{16.52}$$

where $\kappa > 0$ is a design parameter and

$$\boldsymbol{m}_k^* = \arg \min_{\substack{i=1,\dots,K+1 \\ i \neq k}} \left\{ \left\| \boldsymbol{m}_k - \boldsymbol{m}_i \right\| \right\}. \tag{16.53}$$

Thus, $\boldsymbol{m}_k^*$ is the nearest center of all basis functions to the center of the k-th basis function.

With the above background, we are ready to present the rules for adding a new basis function to the RBF network. Initially, the network contains no basis functions. Then, once the following two conditions are satisfied, a new basis function is added. The first condition is:

$$\min_{k=1,\ldots,K} |x(t) - m_{xk}| > \delta_x/2 \quad \text{or} \quad \min_{k=1,\ldots,K} |u(t) - m_{uk}| > \delta_u/2. \tag{16.54}$$

The above is satisfied if the state or input, at time t, is closer to an available "unoccupied" grid vertex than to any of the existing centers of the basis functions. The basis function at the new vertex would be the closest to the existing state or input and would have the greatest influence in the approximation of g. The second condition is

$$|x(t) - \hat{x}(t)| > e_{\max}, \tag{16.55}$$

where $e_{\max}$ is the maximum allowable state estimation error. If (16.55) is satisfied, a new basis function is added to reduce the state estimation error by reducing the modeling error.

Example 16.6 The above technique for identification of dynamical systems using radial-basis function network is illustrated with the system,

$$\dot{x} = 1.1\left(1 - x + 2xu - x^2\right)\exp\left(-0.5x^2 - u^2\right), \quad x(0) = 0, \tag{16.56}$$

where $u = \cos(t/100)$. This model was adapted from [218]. There are many ways to simulate the above identification scheme. In our simulations, we discretized the above differential equation using Euler's method with the step size $dt = 0.01$ sec. The initial weights and the initial state of the identifier were set to zero. We used the following parameter values: $e_0 = 0.001$, $e_{\max} = 0.005$, $\alpha = 5$, $\kappa = 3$. In Figure 16.17, plots of the true state and its estimate versus time are shown. A plot of the state estimation error, e_x, between the true state and its estimate is shown in Figure 16.18. Initially, the error exceeds the bound $e_{\max}$. Adding additional basis functions drives the error back down. The plots shown in Figure 16.19 depict the true state and its estimate versus the input u. Finally, in Figure 16.20, plots of the derivative of the true state versus the true state and the derivative of the estimated state versus the state estimate are shown.

For a generalization of the above approach to real-time approximation of continuous-time dynamical systems using self-organizing radial basis function networks, we refer to Lian et al. [212].

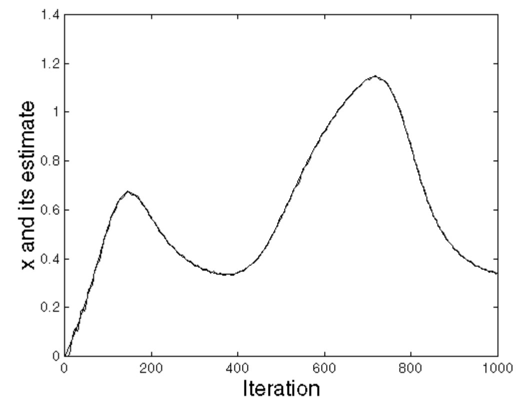


Figure 16.17: Plots of the true state and its estimate versus time. The estimated state "hugs" closely the actual state.

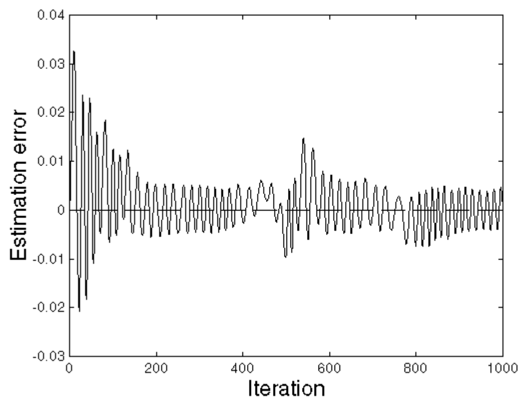


Figure 16.18: A plot of the error, e_x, between the true state and its estimate.

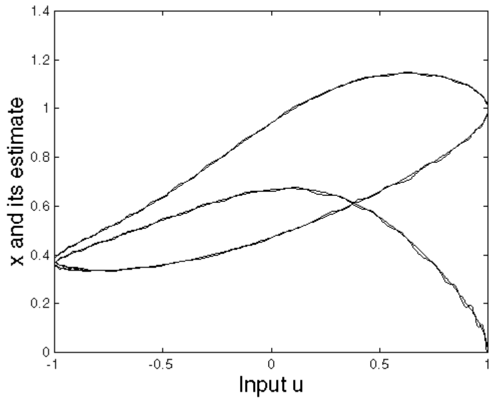


Figure 16.19: A plot of the true state and its estimate versus the input u.

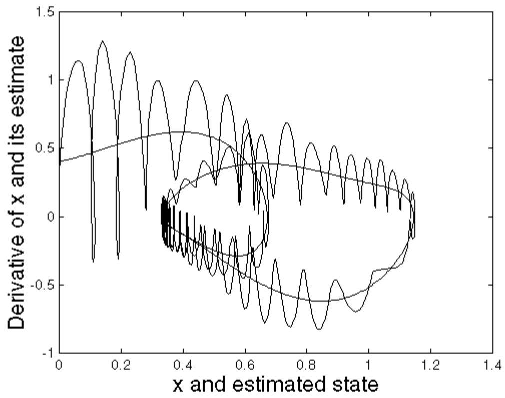


Figure 16.20: Plots of the derivative of the true state versus the true state and the derivative of the estimated state versus the state estimate.

16.4.5 Identification of Multivariable Systems

We extend the results of Subsection 16.4.2 to multi-input multi-state dynamical systems. In our derivation of the update law for the weights of the RBF network, we use a result from matrix algebra presented next. Recall the definition of the trace of a square matrix as given by (A.32). If $\boldsymbol{\Gamma} \in \mathbb{R}^{m \times n}$, then by (A.79) its Frobenius norm satisfies

$$\|\boldsymbol{\Gamma}\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \gamma_{ij}^2 = \operatorname{trace}\left(\boldsymbol{\Gamma}\boldsymbol{\Gamma}^\top\right).$$

If the elements, γ_{ij}, of $\boldsymbol{\Gamma}$ are functions of time, then

$$\frac{d}{dt}\left(\operatorname{trace}\left(\boldsymbol{\Gamma}\boldsymbol{\Gamma}^\top\right)\right) = 2\sum_{i=1}^{m} \sum_{j=1}^{n} \gamma_{ij}\dot{\gamma}_{ij} = 2\operatorname{trace}\left(\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right), \tag{16.57}$$

where $\dot{\boldsymbol{\Gamma}} = \frac{d\boldsymbol{\Gamma}}{dt}$. With the above background, we can proceed with an analysis of the RBF neural identifier for the multivariable systems.

The multivariable dynamical systems that we consider are modeled by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{16.58}$$

where $\boldsymbol{u}(t) \in \mathbb{R}^m$, $\boldsymbol{x}(t) \in \mathbb{R}^n$, and $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the function to be identified. Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be a Hurwitz matrix, that is, the matrix whose eigenvalues are all in the open left-half plane. Because $\boldsymbol{A}$ is a Hurwitz matrix, by the theorem of Lyapunov (Theorem 7.1), for any given real symmetric positive definite (r.s.p.d.) matrix $\boldsymbol{Q}$ the solution $\boldsymbol{P}$ to the continuous Lyapunov matrix equation,

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q} \tag{16.59}$$

is also (real symmetric) positive definite. We use this fact later when we develop a learning law for the RBF neural identifier.

We represent the model (16.58) in an equivalent form,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{16.60}$$

where

$$\boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) = -\boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)).$$

The k-th basis function can be represented as

$$\phi_k(\boldsymbol{x}, \boldsymbol{u}) = \exp\left(-\frac{1}{2\sigma_k^2}\left\| \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} - \boldsymbol{m}_k \right\|^2\right).$$

Let

$$\boldsymbol{\Phi}_K(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \phi_1(\boldsymbol{x}, \boldsymbol{u}) & \phi_2(\boldsymbol{x}, \boldsymbol{u}) & \cdots & \phi_K(\boldsymbol{x}, \boldsymbol{u}) \end{bmatrix}^\top$$

and

$$\hat{\boldsymbol{g}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{m}_k, \sigma_k) = \boldsymbol{W}_K \boldsymbol{\Phi}_K,$$

where $\boldsymbol{W}_K$ is the $n \times K$ weight matrix and $\boldsymbol{m}_k = \begin{bmatrix} \boldsymbol{m}_{\boldsymbol{x}k}^\top & \boldsymbol{m}_{\boldsymbol{u}k}^\top \end{bmatrix}^\top$ is the center of the k-th radial basis function in the $\begin{bmatrix} \boldsymbol{x}^\top & \boldsymbol{u}^\top \end{bmatrix}^\top$-space. The RBF neural identifier has the form,

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{g}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{m}_k, \sigma_k), \quad \hat{\boldsymbol{x}}(0) = \boldsymbol{x}_0. \tag{16.61}$$

Let $\boldsymbol{W}_K^*$ be the optimal weight matrix that minimizes the modeling error,

$$\boldsymbol{e}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) - \boldsymbol{W}_K^* \boldsymbol{\Phi}_K(\boldsymbol{x}(t), \boldsymbol{u}(t)).$$

Using the above definition, we represent the system model (16.60) as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{W}_K^* \boldsymbol{\Phi}_K(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{e}(t). \tag{16.62}$$

We define the state estimation error and the weight estimation error as

$$\boldsymbol{e}_{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}} \quad \text{and} \quad \boldsymbol{\Gamma}_K = \boldsymbol{W}_K^* - \boldsymbol{W}_K.$$

Next, we form the differential equation describing the state estimation error dynamics,

$$\begin{aligned} \dot{\boldsymbol{e}}_{\boldsymbol{x}} &= \dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}} \\ &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{W}_K^* \boldsymbol{\Phi}_K(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{e} - \boldsymbol{A}\hat{\boldsymbol{x}} - \hat{\boldsymbol{g}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{m}_k, \sigma_k) \\ &= \boldsymbol{A}\left(\boldsymbol{x} - \hat{\boldsymbol{x}}\right) + \boldsymbol{W}_K^* \boldsymbol{\Phi}_K(\boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{W}_K \boldsymbol{\Phi}_K(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{e} \\ &= \boldsymbol{A}\boldsymbol{e}_{\boldsymbol{x}} + \boldsymbol{\Gamma}_K \boldsymbol{\Phi}_K(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{e}. \end{aligned} \tag{16.63}$$

We employ the following Lyapunov function candidate,

$$V = V\left(\boldsymbol{e}_{\boldsymbol{x}}, \boldsymbol{\Gamma}_K\right) = \frac{1}{2}\left(\boldsymbol{e}_{\boldsymbol{x}}^\top \boldsymbol{P} \boldsymbol{e}_{\boldsymbol{x}} + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}_K \boldsymbol{\Gamma}_K^\top\right)\right). \tag{16.64}$$

Using (16.57), we evaluate the time derivative of V on the solutions of the state estimation error equation to obtain

$$\begin{aligned} \dot{V} &= \boldsymbol{e}_{\boldsymbol{x}}^\top \boldsymbol{P} \dot{\boldsymbol{e}}_{\boldsymbol{x}} + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}_K \dot{\boldsymbol{\Gamma}}_K^\top\right) \\ &= \boldsymbol{e}_{\boldsymbol{x}}^\top \left(\boldsymbol{P}\boldsymbol{A}\boldsymbol{e}_{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{\Gamma}_K \boldsymbol{\Phi}_K(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{P}\boldsymbol{e}\right) + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}_K \dot{\boldsymbol{\Gamma}}_K^\top\right) \\ &= \frac{1}{2}\boldsymbol{e}_{\boldsymbol{x}}^\top \left(\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}\right) \boldsymbol{e}_{\boldsymbol{x}} + \boldsymbol{e}_{\boldsymbol{x}}^\top \boldsymbol{P}\boldsymbol{\Gamma}_K \boldsymbol{\Phi}_K(\boldsymbol{x}, \boldsymbol{u}) + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}_K \dot{\boldsymbol{\Gamma}}_K^\top\right) \\ &\quad + \boldsymbol{e}_{\boldsymbol{x}}^\top \boldsymbol{P}\boldsymbol{e}. \end{aligned} \tag{16.65}$$

Using the property of the trace operator given by (A.33), on page 965, gives

$$\begin{aligned} \boldsymbol{e}_{\boldsymbol{x}}^{\top}\boldsymbol{P}\boldsymbol{\Gamma}_K\boldsymbol{\Phi}_K(\boldsymbol{x},\boldsymbol{u}) &= \operatorname{trace}\left(\boldsymbol{e}_{\boldsymbol{x}}^{\top}\boldsymbol{P}\boldsymbol{\Gamma}_K\boldsymbol{\Phi}_K(\boldsymbol{x},\boldsymbol{u})\right) \\ &= \operatorname{trace}\left(\boldsymbol{\Gamma}_K\boldsymbol{\Phi}_K(\boldsymbol{x},\boldsymbol{u})\boldsymbol{e}_{\boldsymbol{x}}^{\top}\boldsymbol{P}\right). \end{aligned} \tag{16.66}$$

Observe that

$$\dot{\boldsymbol{W}}_K = -\dot{\boldsymbol{\Gamma}}_K.$$

Using the above and (16.66) along with the Lyapunov equation (16.59) yields

$$\dot{V} = -\boldsymbol{e}_{\boldsymbol{x}}^{\top}\boldsymbol{Q}\boldsymbol{e}_{\boldsymbol{x}} + \frac{1}{\alpha}\left(\operatorname{trace}\left(\alpha\boldsymbol{\Gamma}_K\boldsymbol{\Phi}_K(\boldsymbol{x},\boldsymbol{u})\boldsymbol{e}_{\boldsymbol{x}}^{\top}\boldsymbol{P}\right) - \operatorname{trace}\left(\boldsymbol{\Gamma}_K\dot{\boldsymbol{W}}_K^{\top}\right)\right) + \boldsymbol{e}_{\boldsymbol{x}}^{\top}\boldsymbol{P}\boldsymbol{e}. \tag{16.67}$$

Consider the following weight matrix update law,

$$\boxed{\dot{\boldsymbol{W}}_K = \alpha\boldsymbol{P}\boldsymbol{e}_{\boldsymbol{x}}\boldsymbol{\Phi}_K^{\top}(\boldsymbol{x},\boldsymbol{u})} \tag{16.68}$$

With the above update law, the Lyapunov derivative (16.67) evaluates to

$$\begin{aligned} \dot{V} &= -\boldsymbol{e}_{\boldsymbol{x}}^{\top}\boldsymbol{Q}\boldsymbol{e}_{\boldsymbol{x}} + \boldsymbol{e}_{\boldsymbol{x}}^{\top}\boldsymbol{P}\boldsymbol{e} \\ &\le -\lambda_{\min}(\boldsymbol{Q})\|\boldsymbol{e}_{\boldsymbol{x}}\|^2 + \lambda_{\max}(\boldsymbol{P})\|\boldsymbol{e}\|\|\boldsymbol{e}_{\boldsymbol{x}}\| \\ &= -\lambda_{\min}(\boldsymbol{Q})\left(\|\boldsymbol{e}_{\boldsymbol{x}}\| - \frac{\lambda_{\max}(\boldsymbol{P})}{\lambda_{\min}(\boldsymbol{Q})}\|\boldsymbol{e}\|\right)\|\boldsymbol{e}_{\boldsymbol{x}}\|. \end{aligned} \tag{16.69}$$

Utilizing the same type of arguments as in the scalar case (analyzed in Subsection 16.4.2) and the Lyapunov-like lemma stated on page 392, we can show that when $\boldsymbol{e} = \boldsymbol{0}$, then $\boldsymbol{e}_x(t) \to \boldsymbol{0}$ as $t \to \infty$.

If the magnitude of the state estimation error is such that $\|\boldsymbol{e}_{\boldsymbol{x}}\| < \frac{\lambda_{\max}(\boldsymbol{P})}{\lambda_{\min}(\boldsymbol{Q})}\|\boldsymbol{e}\|$, then it may happen that $\dot{V} > 0$, which in turn may lead to the weights' drift to infinity over time. To prevent this, we modify (16.68) following the analysis of the single-input single-state case. The modified weight update law is

$$\boxed{\dot{\boldsymbol{W}}_K = \begin{cases} \alpha\boldsymbol{P}\boldsymbol{e}_{\boldsymbol{x}}\boldsymbol{\Phi}_K^{\top}(\boldsymbol{x},\boldsymbol{u}) & \text{if } \|\boldsymbol{e}_{\boldsymbol{x}}\| \ge e_0 \quad \text{and} \quad \|\boldsymbol{W}_K\| \le M \\ \boldsymbol{O} & \text{otherwise} \end{cases}} \tag{16.70}$$

Example 16.7 Consider an uncertain dynamical system model,

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -116 & -69 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} \right) \\ &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{u} + \boldsymbol{\zeta}), \end{aligned}$$

where the vector modeling the system uncertainties satisfies the norm bound condition,

$$\left\| \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} \right\| \leq 10.$$

The uncontrolled nominal system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ is asymptotically stable. The solution to the Lyapunov matrix equation $\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{I}_3$ is

$$\boldsymbol{P} = \begin{bmatrix} 9.2833 & 4.8451 & 0.0043 \\ 4.8451 & 3.9263 & 0.0775 \\ 0.0043 & 0.0775 & 0.0412 \end{bmatrix}.$$

Our objective is to design a stabilizing controller, that contains an **adaptive neural term** employing 10 radial basis functions, so that the state of the closed-loop system driven by your controller asymptotically converges to the origin.

We propose the following stabilizing controller,

$$\boldsymbol{u} = -\boldsymbol{u}_{neural} - \eta \frac{\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{x}}{\|\boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{x}\|}, \tag{16.71}$$

where

$$\boldsymbol{u}_{neural} = \boldsymbol{W}\boldsymbol{\Phi}$$

and the gain η will be determined later. The vector $\boldsymbol{\Phi}$ is composed of radial basis functions and is bounded. We assume that the "optimal" weights approximate the uncertainties modeled by the function $\boldsymbol{\zeta}$ so that

$$\boldsymbol{\zeta} = \boldsymbol{W}^*\boldsymbol{\Phi} + \boldsymbol{e}, \tag{16.72}$$

where $\boldsymbol{e}$ is the approximation error. We assume that the approximation error is bounded by a known constant ρ, that is,

$$\|\boldsymbol{e}\| \leq \rho.$$

In further discussion, we use the following notation,

$$\boldsymbol{\sigma} = \boldsymbol{B}^\top \boldsymbol{P}\boldsymbol{x} \text{ and let } \boldsymbol{\Gamma} = \boldsymbol{W}^* - \boldsymbol{W}.$$

Employing the above notation, we represent the proposed control strategy (16.71) as

$$\boldsymbol{u} = -\boldsymbol{W}\boldsymbol{\Phi} - \eta \frac{\boldsymbol{\sigma}}{\|\boldsymbol{\sigma}\|}. \tag{16.73}$$

Consider the following Lyapunov function candidate,

$$V = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} + \frac{1}{2\alpha}\text{trace}\left(\boldsymbol{\Gamma}\boldsymbol{\Gamma}^\top\right).$$

The Lyapunov derivative is

$$\begin{aligned}
\dot{V} &= \boldsymbol{x}^\top \boldsymbol{P}\dot{\boldsymbol{x}} + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right)\\
&= \boldsymbol{x}^\top \boldsymbol{P}\left(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{u}+\boldsymbol{\zeta})\right) + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right)\\
&= -\frac{1}{2}\|\boldsymbol{x}\|^2 + \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{u} + \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{B}\boldsymbol{\zeta} + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right).
\end{aligned}$$

Substituting into the above (16.72) and the control strategy given by (16.73) and manipulating gives

$$\begin{aligned}
\dot{V} &= -\frac{1}{2}\|\boldsymbol{x}\|^2 - \eta\|\boldsymbol{\sigma}\| - \boldsymbol{\sigma}^\top \boldsymbol{W}\boldsymbol{\Phi} + \boldsymbol{\sigma}^\top\left(\boldsymbol{W}^*\boldsymbol{\Phi} + \boldsymbol{e}\right) + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right)\\
&= -\frac{1}{2}\|\boldsymbol{x}\|^2 - \eta\|\boldsymbol{\sigma}\| + \boldsymbol{\sigma}^\top\left(\boldsymbol{W}^* - \boldsymbol{W}\right)\boldsymbol{\Phi} + \boldsymbol{\sigma}^\top\boldsymbol{e} + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right)\\
&\le -\frac{1}{2}\|\boldsymbol{x}\|^2 - \|\boldsymbol{\sigma}\|\left(\eta - \|\boldsymbol{e}\|\right) + \boldsymbol{\sigma}^\top\boldsymbol{\Gamma}\boldsymbol{\Phi} + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right)\\
&= -\frac{1}{2}\|\boldsymbol{x}\|^2 - \|\boldsymbol{\sigma}\|\left(\eta - \|\boldsymbol{e}\|\right) + \text{trace}\left(\boldsymbol{\sigma}^\top\boldsymbol{\Gamma}\boldsymbol{\Phi}\right) + \frac{1}{\alpha}\text{trace}\left(\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right)\\
&= -\frac{1}{2}\|\boldsymbol{x}\|^2 - \|\boldsymbol{\sigma}\|\left(\eta - \|\boldsymbol{e}\|\right) + \text{trace}\left(\boldsymbol{\Gamma}\boldsymbol{\Phi}\boldsymbol{\sigma}^\top + \frac{1}{\alpha}\boldsymbol{\Gamma}\dot{\boldsymbol{\Gamma}}^\top\right).
\end{aligned}$$

We select

$$\eta \ge \|\boldsymbol{e}\| \quad \text{and} \quad \dot{\boldsymbol{\Gamma}} = -\alpha\boldsymbol{\sigma}\boldsymbol{\Phi}^\top$$

to obtain

$$\dot{V} \le -\frac{1}{2}\|\boldsymbol{x}\|^2, \tag{16.74}$$

which is negative semi-definite in the augmented $(\boldsymbol{x}, \boldsymbol{\Gamma})$-space. Therefore $\boldsymbol{x}$ as well as $\boldsymbol{W}$ are bounded. Hence the proposed controller is also bounded, which implies that $\dot{\boldsymbol{x}}$ is bounded. This in turn implies that

$$\frac{d}{dt}\|\boldsymbol{x}\|^2 = 2\boldsymbol{x}^\top\dot{\boldsymbol{x}}$$

is bounded and hence $\|\boldsymbol{x}\|^2$ is uniformly continuous. Integrating both sides of (16.74) gives

$$\int_0^t \dot{V}(\tau)d\tau = V(t) - V(0) \le \int_0^t \left(-\frac{1}{2}\|\boldsymbol{x}\|^2\right)d\tau$$

and hence

$$\int_0^t \|\boldsymbol{x}\|^2 d\tau \le 2(V(0) - V(t)) \le 2V(0),$$

which means that

$$\lim_{t\to\infty}\int_0^t \|\boldsymbol{x}\|^2 d\tau$$

exists and is finite. By Barbalat's lemma,

$$\|\boldsymbol{x}\|^2 \to 0 \quad \text{as} \quad t\to\infty,$$

which implies that

$$\lim_{t\to\infty} \boldsymbol{x} = \boldsymbol{0}.$$

16.5 A Self-Organizing Network

Up to now we discussed neural networks that must have a teacher in order to learn presented patterns. This mode of learning is referred to as the supervised mode. In this section, we analyze a neural network that can learn in the unsupervised mode and the network undergoes the self-organization process. Specifically, the network is capable of clustering training patterns. It is assumed that the number of classes, p, is known. The network's number of weight vectors is equal to the number of classes in the training set. The weights are initialized and then normalized. After initialization, the training patterns, $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}$, are presented. In response to each presented pattern, we compute the winning neuron of the network, that is, we compute the network weight that is closest to the presented pattern. We call this computation the *similarity matching*. Let $\boldsymbol{x}$ be the presented pattern, where, for simplicity, we omitted its index. Then the similarity matching can be described as

$$\left\|\boldsymbol{x}-\boldsymbol{w}^{(i)}\right\| = \min_{1\le j\le p}\left\{\left\|\boldsymbol{x}-\boldsymbol{w}^{(j)}\right\|\right\},$$

where i is the index of the winning neuron. Thus, searching for the winning neuron corresponds to finding the maximum among the p scalar products,

$$\max_{j=1,2,\ldots,p}\left\langle \boldsymbol{w}^{(j)}, \boldsymbol{x}\right\rangle.$$

To see that this is indeed the case, note that because the weights and the patterns are normalized, that is, $\left\|\boldsymbol{w}^{(j)}\right\| = \|\boldsymbol{x}\| = 1$, we have

$$\begin{aligned}\left\|\boldsymbol{x}-\boldsymbol{w}^{(j)}\right\| &= \left(\|\boldsymbol{x}\|^2 - 2\left\langle \boldsymbol{w}^{(j)}, \boldsymbol{x}\right\rangle + \left\|\boldsymbol{w}^{(j)}\right\|^2\right)^{1/2}\\ &= \left(2-2\left\langle \boldsymbol{w}^{(j)}, \boldsymbol{x}\right\rangle\right)^{1/2}.\end{aligned}$$

Hence, minimizing $\left\|\boldsymbol{x}-\boldsymbol{w}^{(j)}\right\|$ is equivalent to maximizing the scalar product $\langle\boldsymbol{w}^{(j)},\boldsymbol{x}\rangle$. Let $\cos\left(\boldsymbol{w}^{(j)},\boldsymbol{x}\right)$ denote the cosine of the angle between $\boldsymbol{w}^{(j)}$ and $\boldsymbol{x}$. Then,

$$\langle\boldsymbol{w}^{(j)},\boldsymbol{x}\rangle=\left\|\boldsymbol{w}^{(j)}\right\|\|\boldsymbol{x}\|\cos\left(\boldsymbol{w}^{(j)},\boldsymbol{x}\right),$$

which means that the winning neuron is "more parallel" to the input pattern $\boldsymbol{x}$ than the other neurons.

After the winning neuron is declared, it is updated, while the remaining weights are not changed. This is the reason for the name of the network—the winner-take-all (WTA) network. This update algorithm is also known as the *Kohonen learning rule* after Kohonen who first proposed this learning rule. The role of the update is to reduce the distance between the winning neuron, $\boldsymbol{w}^{(i)}$, and the presented pattern, $\boldsymbol{x}$. One way to accomplish this task is to apply the descent gradient method to the objective function $\frac{1}{2}\left\|\boldsymbol{x}-\boldsymbol{w}^{(i)}\right\|^2$. The gradient of $\frac{1}{2}\left\|\boldsymbol{x}-\boldsymbol{w}^{(i)}\right\|^2$ with respect to $\boldsymbol{w}^{(i)}$ is

$$\frac{1}{2}\nabla_{\boldsymbol{w}^{(j)}}\left\|\boldsymbol{x}-\boldsymbol{w}^{(i)}\right\|^2=-\left(\boldsymbol{x}-\boldsymbol{w}^{(i)}\right).$$

Applying the descent gradient method to minimize $\frac{1}{2}\left\|\boldsymbol{x}-\boldsymbol{w}^{(i)}\right\|^2$ with a step size α_k yields

$$\boldsymbol{w}^{(i)}[k+1]=\boldsymbol{w}^{(i)}[k]+\alpha_k\left(\boldsymbol{x}-\boldsymbol{w}^{(i)}[k]\right).$$

The weights of other neurons are unchanged, that is,

$$\boldsymbol{w}^{(j)}[k+1]=\boldsymbol{w}^{(j)}[k],\quad \text{for}\quad j\neq i.$$

We summarize our discussion in the form of the following algorithm.

Algorithm 16.4 Winner-Take-All (WTA) Learning

1. **Initialize Weights**
   Generate the initial weights. Normalize each weight to have unit length: $\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$.

2. **Serially Present Training Patterns**
   Normalize each training pattern $\boldsymbol{x}^{(j)}$ to have unit length. The training patterns can be presented in a random order a fixed number of times.

3. **Select Winning Neuron**
   For each WTA weight, compute the scalar product with the presented pattern. Select a weight that gives the largest scalar product. Call this the winning weight.

4. **Update Weight of Winning Neuron**
   Let i be the index of the winning weight. Set

$$\begin{aligned}\boldsymbol{w}^{(i)}[k+1] &= \boldsymbol{w}^{(i)}[k] + \alpha_k \left(\boldsymbol{x}^{(j)} - \boldsymbol{w}^{(i)}[k]\right) \\ \boldsymbol{w}^{(j)}[k+1] &= \boldsymbol{w}^{(j)}[k], \quad \text{for} \quad j \neq i\end{aligned}$$

   Normalize $\boldsymbol{w}^{(i)}[k+1]$ to have unit length.

5. **Repeat by Returning to Step 2**
   Repeat until stopping criterion is met.

Example 16.8 We take the same set of training patterns as in Zurada [377, page 407]:

$$\left[\begin{array}{ccccc} \boldsymbol{x}^{(1)} & \boldsymbol{x}^{(2)} & \boldsymbol{x}^{(3)} & \boldsymbol{x}^{(4)} & \boldsymbol{x}^{(5)} \end{array}\right] = \left[\begin{array}{ccccc} 0.8 & 0.1736 & 0.707 & 0.342 & 0.6 \\ 0.6 & -0.9848 & 0.707 & -0.9397 & 0.8 \end{array}\right].$$

The above patterns are already normalized and are marked in Figure 16.21. As can be seen from Figure 16.21, we can classify the training patterns into two classes, that is, $p = 2$. The initial normalized weights are

$$\left[\begin{array}{cc} \boldsymbol{w}_0^{(1)} & \boldsymbol{w}_0^{(2)} \end{array}\right] = \left[\begin{array}{cc} 1 & -1 \\ 0 & 0 \end{array}\right].$$

The initial weights are also marked in Figure 16.21. The training patterns, $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(5)}$ are presented in ascending order and then recycled. After 200 presentations of the training set, we obtained the following final weights of the network:

$$\left[\begin{array}{cc} \boldsymbol{w}^{(1)} & \boldsymbol{w}^{(2)} \end{array}\right] = \left[\begin{array}{cc} 0.6628 & 0.2868 \\ 0.7488 & -0.9580 \end{array}\right].$$

The final weights are marked in Figure 16.21 with $*$.

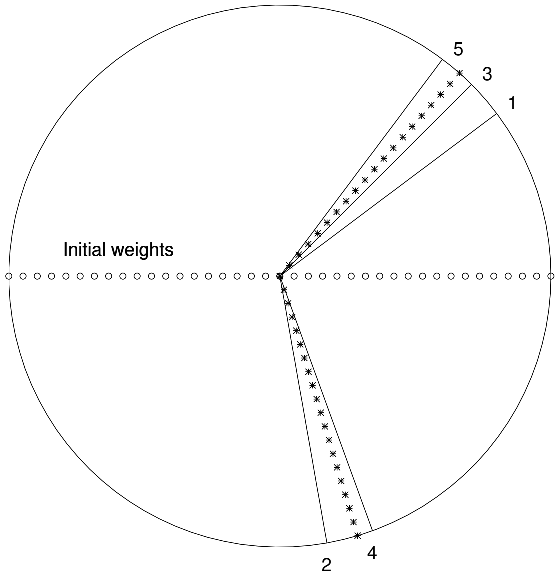


Figure 16.21: Training patterns, initial weights and the final weights of the WTA network of Example 16.8.

16.6 Hopfield Neural Network

16.6.1 Hopfield Neural Network Modeling and Analysis

The analog Hopfield neural network is a dynamical system whose schematic is shown in Figure 16.22. It is a nonlinear interconnected system proposed by Hopfield [140] as a model of biological neural networks. Each node, or neuron, in Figure 16.22 is represented by a circle. The structure of a neuron is depicted in Figure 16.23. The nonlinear amplifier in Figure 16.23 has an input-output characteristic described by a function $g : \mathbb{R} \to [0, 1]$. We assume that g is continuously differentiable and strictly monotonically increasing, that is, $g(u) > g(u')$ if $u > u'$. The function g is sometimes called, the *activation function*. A typical choice of g is the sigmoid function given by

$$x = g(u) = \frac{1}{1 + e^{-u/\theta}}, \tag{16.75}$$

where the positive parameter θ controls the slope of g. A circuit implementation of the Hopfield network is shown in Figure 16.24. To derive equations that model the Hopfield neural network, we use an equivalent representation of the single node of the Hopfield network. This equivalent representation is shown in Figure 16.25. Applying Kirchhoff's current law at the input of each amplifier of the equivalent node representation and taking into account that

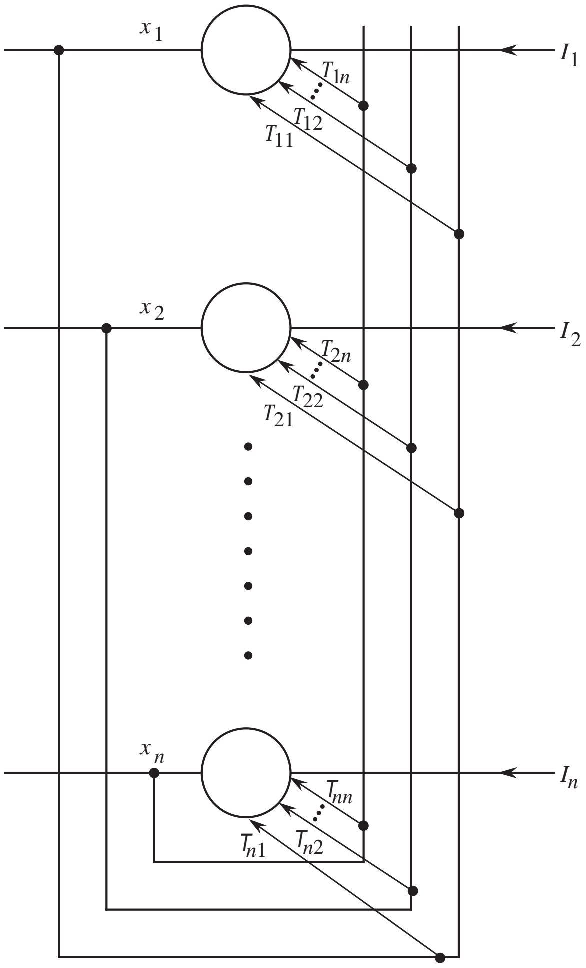


Figure 16.22: Hopfield neural network model.

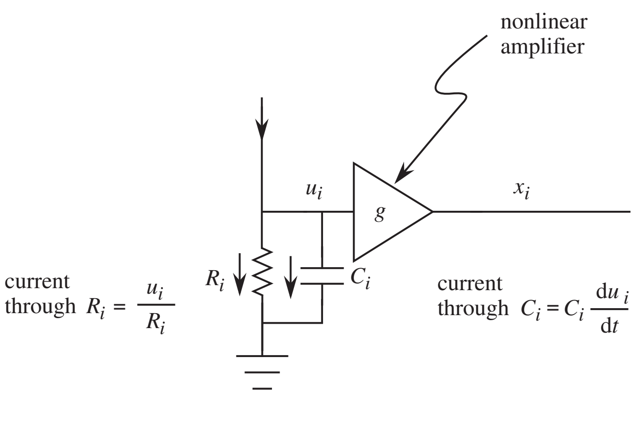


Figure 16.23: Model of a neuron in the Hopfield neural network.

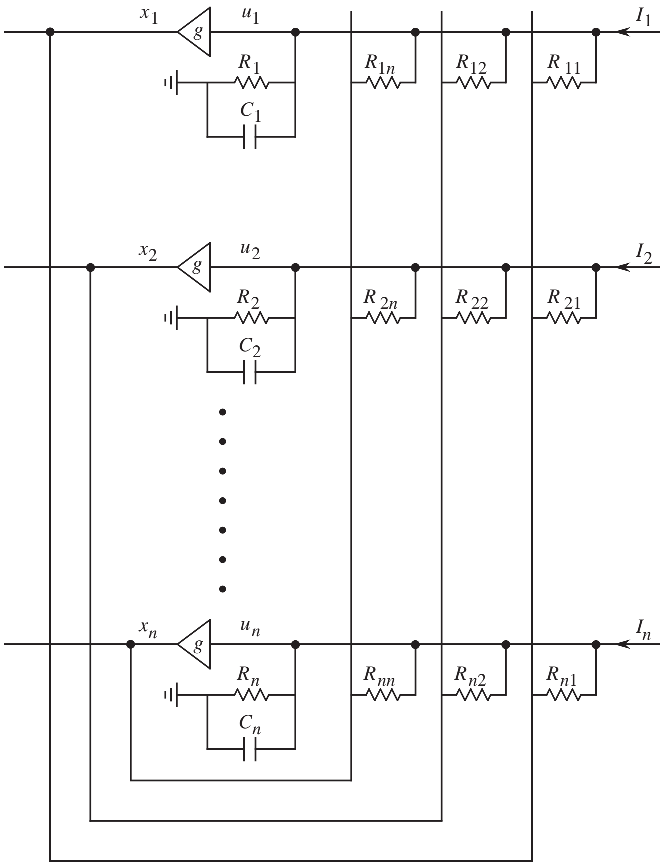


Figure 16.24: A circuit realization of the Hopfield neural network.

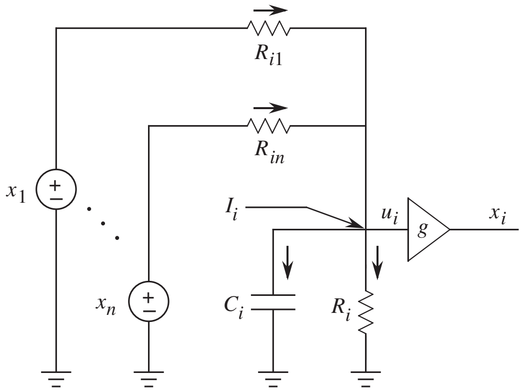


Figure 16.25: Equivalent representation of a single node in the circuit realization of the Hopfield neural network.

the current into the amplifier is negligibly small due to the amplifier's high input resistance, we get

$$C_i \frac{du_i}{dt} + \frac{u_i}{R_i} = \frac{x_1 - u_i}{R_{i1}} + \frac{x_2 - u_i}{R_{i2}} + \cdots + \frac{x_n - u_i}{R_{in}} + I_i, \quad i = 1, 2, \ldots, n, \tag{16.76}$$

where $x_i = g(u_i)$. Let

$$T_{ij} = \frac{1}{R_{ij}}, \quad \frac{1}{r_i} = \sum_{j=1}^{n} T_{ij} + \frac{1}{R_i}. \tag{16.77}$$

Substituting (16.77) into (16.76) yields

$$\boxed{C_i \frac{du_i}{dt} + \frac{u_i}{r_i} = \sum_{j=1}^{n} T_{ij} x_j + I_i, \quad i = 1, 2, \ldots, n} \tag{16.78}$$

The system n first-order nonlinear differential equations given by (16.78) constitutes a model of the Hopfield neural network.

The Hopfield neural network can be viewed as an associative memory. This is because it can have asymptotically stable states that attract the neighboring states to develop in time. This behavior can be interpreted as an evolution of an imperfect, corrupted pattern towards the correct, stored pattern—hence the term associative memory.

In further analysis, we use an equivalent representation of (16.78). To obtain this equivalent representation of the Hopfield model, we first note that the activation function g is, by assumption, strictly increasing and differentiable. Therefore g^{-1} exists and is also differentiable—see the inverse function theorem on page 608. In Figure 16.26, we show plots of $x = g(u)$ and $u = g^{-1}(x)$ for $\theta = 0.2$. We re-write equations (16.78) as

$$C_i \frac{dg^{-1}(x_i)}{dx_i} \frac{dx_i}{dt} + \frac{g^{-1}(x_i)}{r_i} = \sum_{j=1}^{n} T_{ij} x_j + I_i, \quad i = 1, 2, \ldots, n \tag{16.79}$$

Let $\boldsymbol{T} \in \mathbb{R}^{n \times n}$ be the matrix whose elements are T_{ij} and $\boldsymbol{v} = \begin{bmatrix} I_1 & I_2 & \cdots I_n \end{bmatrix}^\top \in \mathbb{R}^n$. Using this notation, we write (16.78) in matrix form,

$$\begin{bmatrix} C_1 \frac{dg^{-1}(x_1)}{dx_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_n \frac{dg^{-1}(x_n)}{dx_n} \end{bmatrix} \dot{\boldsymbol{x}}(t) = \boldsymbol{T}\boldsymbol{x} + \boldsymbol{v} - \begin{bmatrix} \frac{g^{-1}(x_1)}{r_1} \\ \vdots \\ \frac{g^{-1}(x_n)}{r_n} \end{bmatrix}. \tag{16.80}$$

Associated with the Hopfield neural network is its computational energy function, $E : \mathbb{R}^n \to$

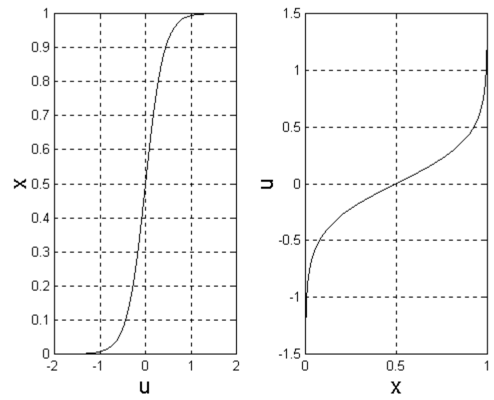


Figure 16.26: Plots of $x = g(u)$ and $u = g^{-1}(x)$.

$\mathbb{R}$, given by

$$\begin{aligned} E(\boldsymbol{x}) &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{ij}x_i x_j - \sum_{i=1}^{n} I_i x_i + \sum_{i=1}^{n}\int_0^{x_i}\frac{g^{-1}(s)}{r_i}ds \\ &= -\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{T}\boldsymbol{x} - \boldsymbol{v}^\top \boldsymbol{x} + \sum_{i=1}^{n}\int_0^{x_i}\frac{g^{-1}(s)}{r_i}ds. \end{aligned} \tag{16.81}$$

Suppose now that $\boldsymbol{T} = \boldsymbol{T}^\top$. Then, the gradient of E is

$$\nabla E(\boldsymbol{x}) = -\boldsymbol{T}\boldsymbol{x} - \boldsymbol{v} + \begin{bmatrix} \frac{g^{-1}(x_1)}{r_1} & \cdots & \frac{g^{-1}(x_n)}{r_n} \end{bmatrix}^\top. \tag{16.82}$$

Let

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} C_1\frac{dg^{-1}(x_1)}{dx_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_n\frac{dg^{-1}(x_n)}{dx_n} \end{bmatrix}. \tag{16.83}$$

We now combine (16.82) and (16.83) to obtain an equivalent form of the Hopfield network modeling equations,

$$\boxed{\boldsymbol{S}(\boldsymbol{x})\dot{\boldsymbol{x}}(t) = -\nabla E(\boldsymbol{x}(t))} \tag{16.84}$$

We analyze the dynamics of the Hopfield network modeled by (16.84) using the computational energy function. We first show that E is bounded. Taking norms of (16.81) yields

$$|E(\boldsymbol{x})| \leq \frac{1}{2}\|\boldsymbol{T}\|\,\|\boldsymbol{x}\|^2 + \|\boldsymbol{v}\|\|\boldsymbol{x}\| + \sum_{i=1}^{n}\frac{1}{r_i}\left|\int_0^{x_i} g^{-1}(s)ds\right|. \tag{16.85}$$

The components of the vector $\boldsymbol{x}$ are the outputs of the nonlinear amplifiers and they take values from the interval $[0,1]$. Hence, $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 \leq \sqrt{n}$. Thus, to show that $E(\boldsymbol{x})$ is bounded it is enough to show that the integral $\int_0^{x_i} g^{-1}(s)ds$ is bounded. We first find the expression for g^{-1}. Recall that $g(u)$ is given by (16.75). Therefore,

$$\frac{1}{x} = 1 + e^{-u/\theta},$$

or

$$\frac{1}{x} - 1 = e^{-u/\theta}, \tag{16.86}$$

Taking logarithms of both sides of the above yields

$$\ln\left(\frac{1-x}{x}\right) = -\frac{u}{\theta}.$$

Hence,

$$u = g^{-1}(x) = \theta\left(\ln(x) - \ln(1-x)\right). \tag{16.87}$$

Using the symmetry property of g^{-1}, we obtain

$$\begin{aligned}\left|\int_0^{x_i} g^{-1}(s)ds\right| &\leq \int_0^{x_i}\left|g^{-1}(s)\right|ds \\ &\leq -2\int_0^{1/2} g^{-1}(s)ds. \end{aligned} \tag{16.88}$$

Recall that

$$\int \ln(s)ds = s\ln(s) - s.$$

Hence,

$$\int \ln(1-s)ds = -(1-s)\ln(1-s) - s.$$

Using the above, we obtain

$$\begin{aligned}\int_0^{1/2} g^{-1}(s)ds &= \int_0^{1/2}\theta\left(\ln(s) - \ln(1-s)\right)ds \\ &= \theta\left(s\ln(s) - s + (1-s)\ln(1-s) + s\right)\Big|_0^{1/2} \\ &= \theta\left(s\ln(s) + (1-s)\ln(1-s)\right)\Big|_0^{1/2}. \end{aligned} \tag{16.89}$$

Note that

$$s \ln(s) = \frac{\ln(s)}{1/s}.$$

Applying L'Hospital's rule to the above yields

$$\lim_{s\to 0} \frac{\ln(s)}{1/s} = \lim_{s\to 0} \frac{1/s}{-1/s^2} = 0.$$

Applying the above to (16.89), we get

$$\int_0^{1/2} g^{-1}(s)ds = \theta \ln(1/2). \tag{16.90}$$

Combining (16.88) and (16.90) yields

$$\left| \int_0^{x_i} g^{-1}(s)ds \right| \le -2\theta \ln(1/2), \tag{16.91}$$

which completes the proof of the statement that E is bounded.

Let $\boldsymbol{x}(t)$ denote the trajectory of the Hopfield network model (16.84). We show that if $\boldsymbol{T} = \boldsymbol{T}^\top$, then

$$\frac{d}{dt}E(\boldsymbol{x}(t)) \le 0$$

on the Hopfield network trajectories. Using the chain rule gives

$$\frac{d}{dt}E(\boldsymbol{x}(t)) = \nabla E(\boldsymbol{x}(t))^\top \dot{\boldsymbol{x}}(t). \tag{16.92}$$

It follows from (16.84) that

$$\nabla E(\boldsymbol{x}(t)) = -\boldsymbol{S}(\boldsymbol{x}(t))\dot{\boldsymbol{x}}(t). \tag{16.93}$$

Substituting (16.93) into (16.92), we obtain

$$\begin{aligned} \frac{d}{dt}E(\boldsymbol{x}(t)) &= -\dot{\boldsymbol{x}}^\top(t)\boldsymbol{S}(\boldsymbol{x}(t))\dot{\boldsymbol{x}}(t) \\ &= -\sum_{i=1}^{n} C_i \frac{dg^{-1}(x_i(t))}{dx_i}\dot{x}_i^2, \end{aligned} \tag{16.94}$$

where $C_i > 0$ and

$$\frac{dg^{-1}(x_i)}{dx_i} = \theta\left(1/x_i + 1/(1-x_i)\right) > 0 \tag{16.95}$$

because $x_i \in (0,1)$. Hence,

$$\frac{d}{dt}E(\boldsymbol{x}(t)) \le 0$$

on the trajectories of the Hopfield model (16.84).

In the following subsection, we use the above result to construct Hopfield network based analog-to-digital converters.

16.6.2 Analog-to-Digital Converter

Our objective now is to show how the Hopfield neural network can be used to solve some engineering problems. Specifically, we will discuss the construction of an analog-to-digital (A/D) converter using the Hopfield neural network. The idea was put forward by Tank and Hopfield [324]. In A/D conversion, an analog value is transformed into its corresponding binary representation. The conversion that we discuss consists of two steps. The first step is to round the given analog value to the nearest integer. The second step is to convert the integer into its binary equivalent. For example, 3.4 is first rounded to 3, then 3 is converted into its binary equivalent "11" if we have a 2-bit converter, "011" if we have a 3-bit converter, or "0011" if we have a 4-bit converter. Therefore, an N-bit A/D converter can be viewed as a device that assigns to a given analog input a corresponding vertex of the hypercube $[0,1]^N$.

The first step in A/D conversion, rounding of a given number to its nearest integer, can be organized as solving an optimization problem by a neural network. This entails formulation of a suitable cost, or penalty, function to be minimized. In fact, finding a suitable cost function is the first step in constructing a neural A/D converter.

To better understand the idea behind solving A/D conversion problems using neural networks, we first consider constructing a 2-bit A/D converter. Let V_1 be the most significant bit (MSB), V_0 be the least significant bit (LSB) and x an analog value to be converted. In the actual neural network, V_1 and V_0 are the outputs of the two neurons in the neural A/D converter. We first consider a very simple cost function,

$$E = \frac{1}{2}\left(V_1 2^1 + V_0 2^0 - x\right)^2. \tag{16.96}$$

A plot of the above function, for an analog input of $x = 1.5$, is shown in Figure 16.27(a). Note that E is defined over the two-dimensional hypercube $[0,1]^2$. However, constructing a circuit minimizing the cost function (16.96) would not yield good results because V_0 and V_1 would take values from the interval $[0,1]$ rather than the set $\{0,1\}$. To overcome the problem, we add to (16.96) two terms whose purpose is to penalize for non-integer values of V_0 and V_1. The modified cost function is

$$E = \frac{1}{2}\left(V_1 2^1 + V_0 2^0 - x\right)^2 + aV_1(V_1 - 1) + bV_0(V_0 - 1), \tag{16.97}$$

where a and b are weighting coefficients. Tank and Hopfield [324] suggest selecting a and b so that the quadratic terms V_0^2 and V_1^2 are cancelled out. In particular, if we set $a = -2$ and $b = -0.5$, then (16.97) becomes

$$E = \frac{1}{2}x^2 + 2V_0V_1 - 2V_1x - V_0x + 2V_1 + \frac{1}{2}V_0. \tag{16.98}$$

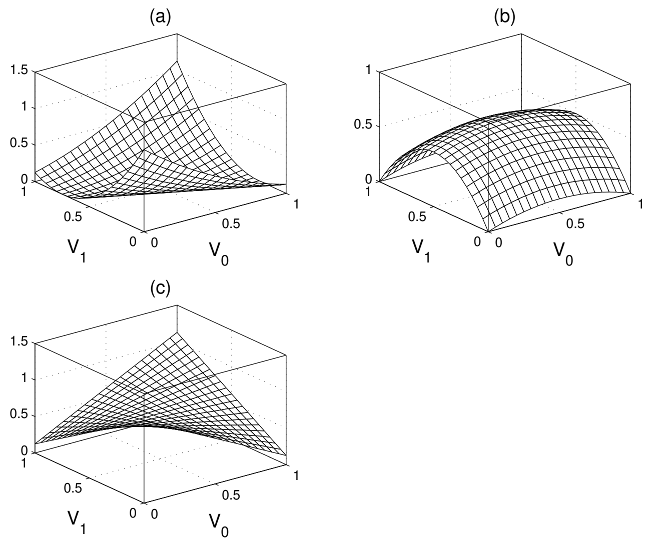


Figure 16.27: Constructing the cost function for a 2-bit A/D converter—plots of the cost function and its components; (a) plot of (16.96) for an analog input of $x = 1.5$; (b) plot of the term used to cancel the quadratic terms in (16.96); (c) plot of the cost function (16.98).

The term used to cancel the quadratic terms is plotted in Figure 16.27(b). The final cost function given by (16.98) is plotted in Figure 16.27(c). When an analog input x is converted to its binary equivalent, the cost function (16.98) is minimized. One can interpret this process as follows. A given analog input x defines a corresponding cost function E. We can view E as a landscape. Now, a ball placed somewhere on this landscape will move towards the lowest point, which should correspond to one of the vertices of the hypercube $[0,1]^2$.

Our next task is to construct differential equations that model a neural network implementing an A/D converter. The parameters of the state equations of the network will be chosen so that the time derivative of E evaluated on the solutions of the equations describing the network is negative. We consider a general case of an N-bit A/D converter. After this, we derive state equations for a 2-bit A/D converter. Note that

$$\frac{dE}{dt} = \frac{\partial E}{\partial V_0}\dot{V}_0 + \frac{\partial E}{\partial V_1}\dot{V}_1 + \cdots + \frac{\partial E}{\partial V_{N-1}}\dot{V}_{N-1},$$

where $V_i = g_i(u_i)$. Hence, similarly as in (16.94), it is enough to set

$$\frac{\partial E}{\partial V_i} = -C_i\dot{u}_i = -C_i\frac{dg_i^{-1}(V_i)}{dV_i}\dot{V}_i, \quad i = 0, 1, \ldots, N-1 \tag{16.99}$$

to obtain

$$\frac{dE}{dt} = -\sum_{i=0}^{N-1} C_i\frac{dg_i^{-1}(V_i)}{dV_i}\dot{V}_i^2 \le 0, \tag{16.100}$$

where as in (16.95), $\frac{dg_i^{-1}(V_i)}{dV_i} > 0$. Thus, if the state equations of an N-bit A/D converter satisfy (16.99), then $dE/dt \leq 0$ on the trajectories of the converter. Using (16.99), we can write the state equations that model the 2-bit A/D converter consisting of two neurons. Let $V_0 = g_0(u_0)$ and $V_1 = g_1(u_1)$. Then, substituting (16.98) into (16.99) and performing required differentiation, we obtain

$$\left.\begin{array}{rcl} C_0\dot{u}_0 & = & -\frac{\partial E}{\partial V_0} = -\left(2V_1 - x + 0.5\right) = -(2g_1(u_1) - x + 0.5) \\ C_1\dot{u}_1 & = & -\frac{\partial E}{\partial V_1} = -2\left(V_0 - x + 1\right) = -2(g_0(u_0) - x + 1) \end{array}\right\} \tag{16.101}$$

For simplicity, we assume that $C_0 = C_1 = 1$ and that $g_0 = g_1 = g$. Note that because C is present in both state equations, assigning $C_0 = C_1 = 1$ does not distort the dynamical behavior of the network; it only changes the time scale of both equations by the same factor. Observing that for $i = 1, 2$,

$$\dot{u}_i = \frac{dg_i^{-1}(V_i)}{dV_i}\dot{V}_i,$$

we represent (16.101) as

$$\left.\begin{array}{rcl} \dot{V}_0 & = & -\left(2V_1 - x + 0.5\right)\left(\frac{dg^{-1}(V_0)}{dV_0}\right)^{-1} \\ \dot{V}_1 & = & -2\left(V_0 - x + 1\right)\left(\frac{dg^{-1}(V_1)}{dV_1}\right)^{-1} \end{array}\right\} \tag{16.102}$$

We display a plot of the cost function E for the analog value $x = 1.5$ in Figure (16.28)(a) and a phase portrait of the 2-bit A/D converter in Figure (16.28)(b). We chose the activation function

$$V_i = g(u_i) = \frac{1}{1 + e^{-4u_i}}.$$

Using (16.87), we find

$$u_i = g^{-1}(V_i) = \frac{1}{4}\left(\ln(V_i) - \ln(1 - V_i)\right).$$

Thus,

$$\frac{dg^{-1}(V_i)}{dV_i} = \frac{1}{4V_i(1 - V_i)}.$$

Note that for $x = 1.5$, the cost function has two minimizers in the hypercube $[0, 1]^2$ that are separated by a "ridge". Initial conditions on each side of this ridge yield a different final value of the state vector $[V_0, V_1]$. Depending on the initial conditions of the state vector of the A/D converter, the analog value $x = 1.5$ will either be converted to the binary "01" or "10". Thus, initial conditions influence the dynamical behavior of the A/D converter. Also, the activation nonlinearity influences the dynamical behavior of the A/D converter via the multiplicative term $\left(\frac{dg^{-1}(V_i)}{dV_i}\right)^{-1}$ present in the state equations (16.102). One can see

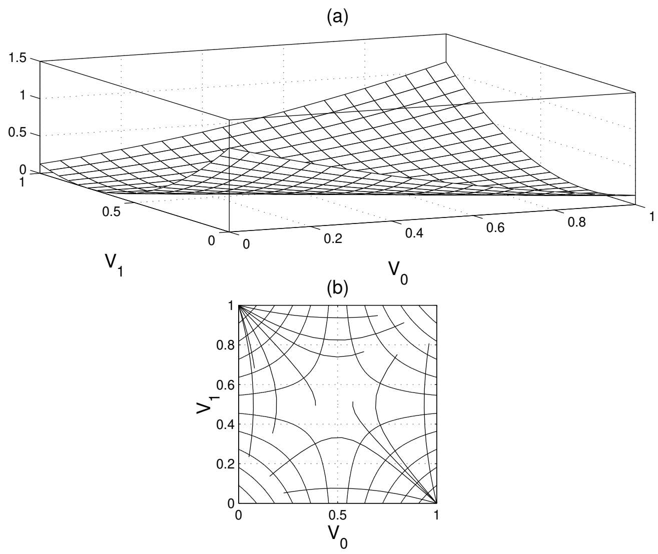


Figure 16.28: A plot of the cost function and a phase portrait of the 2-bit A/D converter for the analog value $x = 1.5$.

that the multiplicative factor influences the direction in which the converter state trajectory evolves.

Employing the above technique, we can construct higher order A/D converters. For example, to construct a 3-bit A/D converter, we begin with the cost function

$$E = \frac{1}{2}\left(V_2 2^2 + V_1 2^1 + V_0 2^0 - x\right)^2 + aV_2(V_2 - 1) + bV_1(V_1 - 1) + cV_0(V_0 - 1).$$

To cancel the quadratic terms, select $a = -8$, $b = -2$, $c = -1/2$. The state equations of the converter are obtained from the equations

$$\dot{V}_i = -\frac{1}{C_i}\frac{\partial E}{\partial V_i}\left(\frac{dg_i^{-1}(V_i)}{dV_i}\right)^{-1} \qquad i = 0, 1, 2.$$

In our analysis of A/D neural converters, we observed that while the 2-bit converter yielded correct results, the 3-bit and 4-bit converters showed some deficiency in their performance. The errors observed for these two converters were quite significant and increased with the number of bits and consequently with network size.

16.7 Notes

This chapter gives an introduction to computing using artificial neural networks from a dynamical system perspective. The disclaimer that we found in the book by Hertz, Krogh, and

Palmer [135, page xix] also applies to this chapter as well as to the whole field of artificial neural networks, "The adjective "neural" is used because much of the inspiration for such networks comes from neuroscience, *not* because we are concerned with networks of real neurons. Brain modeling is a different field and, though we sometimes describe biological analogies, our prime concern is with what *artificial* networks can do, and why." The readers interested in principles underlying the organization of biological neurons into circuits and networks within different regions of the brain may consult the book edited by Shepherd [298]. For textbooks with an engineering approach to neural networks, we recommend Hassoun [132], Hagan, Demuth, and Beale [125], Zurada [377], Haykin [134], Fausett [99], Khanna [179], or Harvey [131]. Applications of artificial neural networks to signal processing and solving optimization problems are presented in the book by Cichocki and Unbehauen [63]. A psychological slant to the subject is presented in the book by Rumelhart, McClelland, and the PDP Research Group [290], and Anderson [10]. Impatient readers wishing to get a feel of artificial neural networks may consult a tutorial by Lippmann [217], and also by Hush and Horne [154].

According to Haykin [134, page 38], "The modern area of neural networks began with the pioneering work of McCulloch and Pitts (1943). McCulloch was a psychiatrist and neuroanatomist by training; he spent some 20 years thinking about the representation of an event in the nervous system. Pitts was a mathematical prodigy, who joined McCulloch in 1942." In their paper [236], McCulloch and Pitts, assuming that the activity of the neuron is an "all-or-none" process, developed a logical calculus of neural events. They argued that, with addition of memory, a neural network consisting of a sufficiently large number of their neurons can compute anything that can be computed on any digital computer. McCulloch and Pitts' paper influenced John von Neumann who used idealized switch-delay elements derived from the McCulloch-Pitts neuron model in the construction of the EDVAC (Electronic Discrete Variable Automatic Computer) that was developed out of the ENIAC (Electronic Numerical Integrator and Computer). The ENIAC was the first general purpose computer and it was invented by J. Presper Eckert and John Mauchly.

More recent investigators of the computational processes in the nervous system compare the brain to a hybrid, analog-digital computer [185]. According to Koch [185, page 210], individual nerve cells communicate between each other via streams of digital pulses. Information is, on the other hand, processed in the analog domain. The resulting signal is then converted back into digital pulses and conveyed to the following neurons.

16.8 Exercises

Exercise 16.1 Write a MATLAB program that implements the TLU (Perceptron) learning algorithm for generating a separating line for points on a plane. Test your program on the following data. For $i = 1, \ldots, 4$ and $d_i = 1$,

$$\begin{bmatrix} \boldsymbol{x}^{(1)} & \boldsymbol{x}^{(2)} & \boldsymbol{x}^{(3)} & \boldsymbol{x}^{(4)} \end{bmatrix} = \begin{bmatrix} 5 & -7 & 8 & -1 \\ 5 & 6 & -3 & 2 \end{bmatrix}.$$

The above points constitute class $\mathcal{C}_1$. You can label the above points using, for example, "x". Class $\mathcal{C}_2$ consists of the points corresponding to $d_i = -1$. They are,

$$\begin{bmatrix} \boldsymbol{x}^{(5)} & \boldsymbol{x}^{(6)} & \boldsymbol{x}^{(7)} \end{bmatrix} = \begin{bmatrix} -6 & -4 & 7 \\ 1 & -4 & -5 \end{bmatrix}$$

You can label these points using "o". The initial weight vector and threshold are

$$\boldsymbol{w}[0] = \begin{bmatrix} -5 & -1 \end{bmatrix}^\top \quad \text{and} \quad t[0] = 5.$$

Plot the initial separating line as well as the final separating line.

Exercise 16.2 Write a MATLAB program that implements the WTA algorithm. Test your program on the following data. The initial weights of the WTA network are:

$$\begin{bmatrix} \boldsymbol{w}_0^{(1)} & \boldsymbol{w}_0^{(2)} & \boldsymbol{w}_0^{(3)} \end{bmatrix} = \begin{bmatrix} 0.4403 & 0.7411 & -0.8866 \\ 0.8979 & -0.6714 & 0.4625 \end{bmatrix}, \qquad \alpha = 0.5.$$

The training patterns are:

$$\boldsymbol{X} = \begin{bmatrix} -0.2 & 0.2 & -.1 & 0.98 & 0.98 & .8 & -0.6 & -0.8 & -0.7 \\ 0.98 & 0.98 & .9 & 0.2 & -0.2 & 0.1 & -0.8 & -0.6 & -0.7 \end{bmatrix}$$

Present the training set 200 times. Find the weights of the network after training.

Exercise 16.3 Use the identification scheme depicted in Figure 16.11 and the neural fuzzy on-line identification algorithm from Section 16.3 to identify the plant described by

$$y[k+1] = \frac{y[k](y[k-1]+2)y[k]+2.5)}{8.5 + y[k]^2 + y[k-1]^2} + u[k],$$

where

$$u[k] = 2\cos(2\pi k/100) \quad \text{for} \quad k \le 200$$

and

$$u[k] = 1.2\sin(2\pi/20) \quad \text{for} \quad k > 200.$$

The identifier is to be described by

$$\hat{y}[k+1] = f[k] + u[k],$$

where f has the form of (16.27) with $M = 100$. Choose $\alpha = 0.5$, $y[0] = y[1] = 0$, $f[1] = f[2] = 0$, and start with $u[1] = 0$. (The above plant was also used by Lin and Lee [215, page 731] to test their identification scheme.)

Exercise 16.4 Write a MATLAB program that implements the learning algorithm for the RBF identifier discussed in Section 16.4.

Chapter 17

Genetic and Evolutionary Algorithms

> The word "algorithm" itself is quite interesting; at first glance it may look as though someone intended to write "logarithm" but jumbled up the first four letters. The word did not appear in *Webster's New World Dictionary* as late as 1957; we find only the older form "algorism" with its ancient meaning, i.e., the process of doing arithmetic using Arabic numerals."…"Finally, historians of mathematics found the true origin of the word algorism: it comes from the name of a famous Persian textbook author, Abu Ja'far Mohammed ibn Mûsâ al-Khowârizmî (c. 825)—literally, "Father of Ja'far, Mohammed, son of Moses, native of Khowârizm." "…"Al-Khowârizmî wrote the celebrated book *Kitab al jabr w'al-muqabala* ("Rules of restoration and reduction"); another word, "algebra", stems from the title of his book, although the book wasn't really very algebraic."…"The modern meaning for algorithm is quite similar to that of *recipe*, *process*, *method*, *technique*, *procedure*, *routine*, except that the word "algorithm" connotes something just a little different. [184, pages 1–3]

Genetic algorithms, introduced by Holland [139] in the 1960s, have been originally proposed to model adaptive processes, both natural and man-made. In this chapter, we discuss applications of genetic algorithms to solving optimization problems. Genetic algorithms were inspired by biological genetics and evolutionary processes. This is the reason for biological terms being used in the context of genetic algorithms. One of the variants of genetic algorithms, called the evolutionary algorithm is also presented.

17.1 Genetics as an Inspiration for an Optimization Approach

The *gene* is the basic unit of inheritance. Genes are pieces of DNA, which is the material inside of a nucleus of a cell that carries the genetic instructions for making living organisms. Many traits are determined by genes inherited from the parents of the individual. Consider gene $\mathcal{A}$ that determines a certain trait, such as seed coat texture—round or wrinkled. Thus gene $\mathcal{A}$ has two alternative forms. These different forms of the same gene are called *alleles*, or *allelic genes*. The allelic forms of each gene are symbolized by the uppercase and lowercase of the same letter. In our example, gene $\mathcal{A}$ has two alleles, allele A and allele a. One of the alleles controls the round seed coat texture while, the other allele controls the wrinkled texture. Genes are organized into *chromosomes*, which are small packages of genes in the nucleus of a cell. Different kinds of organisms have different numbers of chromosomes. Humans have 23 pairs of chromosomes including two sex chromosomes. In mating each parent contributes one chromosome to each pair of the chromosomes of an offspring, so the offspring gets half of his or her chromosomes from his or her mother and half from his or her father. The assemblage of genes, encrypted as a string of DNA molecules, is called the *genome*. The human genome consists of about 35,000 genes. The particular gene constitution is called the *genotype*. The expression of the genotype is the *phenotype*, which is the way we look and behave.

Genetic algorithms are a result of an effort to model adaptation phenomena in natural and artificial systems. The term *adaptation* can be viewed as the composition of two words: *ad* + *aptare* meaning "to fit to" [139, page 1]. "Just what are adaptation's salient features? We can see at once that adaptation, whatever its context, involves a progressive modification of some structure or structures. These structures constitute the grist of the adaptive process, being largely determined by the field of study. Careful observation of successive structural modifications generally reveals a basic set of structural modifiers or operators; repeated action of these operators yields the observed modification sequences." [139, page 3]

In this chapter, we describe two genetic algorithms and apply them to solving optimization problems. An optimization problem of a real-valued function, $f : \Omega \to \mathbb{R}$, requires finding an element $\boldsymbol{x} \in \Omega$ such that f is maximized or minimized. The function f is called the *fitness function*. Suppose that we wish to maximize f over the parameter space Ω. The parameter space, Ω, of an underlying optimization problem is also called the *phenotype space*. A classical genetic algorithm works with binary representations of the points from the parameter space, rather than directly with the elements of Ω. We refer to the classical genetic algorithms as the canonical algorithms. The strings of binary digits (1s and

0s) corresponding to these points are called chromosomes. The space of chromosomes is called the *genotype space*. In general, the chromosomes in the genotype space are abstract mathematical objects on which the so called genetic operators work. After a stopping criterion for the algorithm is satisfied, a decoding function transforms a solution chromosome from the genotype space into the phenotype space. Genetic algorithms are derivative-free optimization algorithms that use only the fitness function. They do not use the fitness function derivatives. The higher the fitness of a chromosome, the better the candidate solution encoded by the chromosome. Genetic algorithms work with populations of chromosomes rather than individual chromosomes. A genetic algorithm is an iterative process that at each iteration uses the information obtained from testing a current population of chromosomes to direct its search into promising areas of the search space. Thus, generating a suitable population of chromosomes at each iteration is the key to the algorithm performance. In the simplest form, creating a new population of chromosomes to be tested involves three genetic operators: selection, crossover, and mutation. These genetic operators work on genotypes that encode phenotypes.

The role of the selection operator is to pick a population of chromosomes for further processing, called reproduction. Chromosomes are selected into a mating pool for reproduction with a probability proportional to their fitness.

The crossover operator randomly exchanges substrings between two parent chromosomes to create two offspring. The simplest crossover operator is the one-point crossover, where a crossing site is randomly picked and the corresponding substrings are exchanged, as illustrated in Figure 17.1. For example, the two parent chromosomes

$$1010101 \quad \text{and} \quad 1111010,$$

could have the crossing site after the fourth bit in each to produce the two offspring

$$1010010 \quad \text{and} \quad 1111101.$$

The mutation operator randomly flips an arbitrary bit in a chromosome. For example, the chromosome 010101 might undergo a mutation in its third position to yield a new chromosome 011101.

We now provide a description of the canonical genetic algorithm using the above components.

Algorithm 17.1 Canonical Genetic (GA)

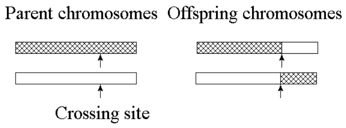


Figure 17.1: The one-point crossover.

For a given optimization problem characterized by a fitness function f:

Step 1. Choose a method of encoding candidate solutions into chromosomes, that is, decide upon a map between the phenotype and genotype spaces. (A commonly used encoding method is to concatenate the binary strings representing each real parameter value x_i.)

Step 2. Start with a randomly generated population of chromosomes.

Step 3. Evaluate the fitness of each chromosome in the population.

Step 4. Replace the current population with a new population by applying selection, crossover, and mutation operations on the current population.

Step 5. If a stopping criterion is satisfied, stop and choose the fittest chromosome from the population as a solution, else go to **Step 3.**

Another type of a genetic algorithm that we analyze in this chapter attempts to avoid the need for a decoding function by using the chromosome representation as close as possible to their representation in the phenotype space.

17.2 Implementing a Canonical Genetic Algorithm

We now discuss implementation issues of the canonical genetic algorithm (GA) when applied to solving optimization problems. We consider optimization problems of the form:

$$\text{maximize} \quad g(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^n. \tag{17.1}$$

A maximizing point or maximizer is denoted $\boldsymbol{x}^*$. We note that there is no loss of generality by considering only maximization problems, for if we are asked to minimize g, we can transform the minimization problem into the maximization problem by means of a simple transformation,

$$\min_{\boldsymbol{x}} g(\boldsymbol{x}) = -\max_{\boldsymbol{x}}(-g(\boldsymbol{x})). \tag{17.2}$$

Also note that if $\boldsymbol{x}^*$ is a minimizer of $g(\boldsymbol{x})$, then $\boldsymbol{x}^*$ is a maximizer of $-g(\boldsymbol{x})$, and vice versa. Once an objective function of the maximization problem is given, the next step is to transform the objective function into a fitness function. For example, we can use the transformation,

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) - g_{\min}, \tag{17.3}$$

where $f(\boldsymbol{x})$ denotes a fitness function and $g_{\min}$ is the minimum value of $g(\boldsymbol{x})$ for the current population of chromosomes. Note that the constructed fitness function is indeed non-negative on the chromosomes of the current population.

We now discuss the implementation of **Step 1** of the canonical GA presented in the previous section. We first analyze the case when the fitness function, f, is a function of only one parameter x. Suppose that

$$x \in [x_{\min}, x_{\max}] = \Omega.$$

A common encoding method, that transforms points from the phenotype space into a genotype space, is the binary encoding. In our example the phenotype space is the interval $[x_{\min}, x_{\max}]$. If we decide to use l bits to represent a point x, then this corresponds to mapping the interval $[x_{\min}, x_{\max}]$ into the discrete genotype space $\left[0, 2^l - 1\right]$ consisting of 2^l binary strings. Thus, the number of bits l determines the size of the genotype space and hence the resolution, or accuracy, of the subsequent search. We can control the resolution by appropriately choosing the parameter l. Let r denote the desired resolution. Then, the parameter l that results in the required resolution can be found by solving the inequality

$$r \geq \frac{x_{\max} - x_{\min}}{2^l - 1}. \tag{17.4}$$

If the fitness function depends on many parameters, then we can encode the elements of the phenotype space by concatenating the binary strings representing each component of $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top$. For example, let $f = f(x_1, x_2)$, where $x_1 \in [x_{1,\min}, x_{1,\max}]$ and $x_2 \in [x_{2,\min}, x_{2,\max}]$. If we use 10 bits to encode each variable, then the genotype space will consist of 20-bit binary strings and the size of the genotype space is $2^{10} \times 2^{10} = 1,048,576$.

After initializing randomly a population of chromosomes, we evaluate the fitness $f(\boldsymbol{x})$ of each chromosome $\boldsymbol{x}$ in the population. This requires decoding from the genotype space into the phenotype space. Suppose, as before, that the fitness function depends on two parameters x_1 and x_2 and that we used the standard encoding method described above. We now have to convert a given binary string into two real numbers $x_1 \in [x_{1,\min}, x_{1,\max}]$ and $x_2 \in [x_{2,\min}, x_{2,\max}]$. We can perform the required decoding process in three steps. First note that the first l bits of the chromosome $\boldsymbol{x}$ correspond to x_1 while the remaining bits correspond to x_2. We first convert the strings of l bits into integers. Thus, for example, if $l = 4$ and the chromosome is 10010111, which is an 8-bit string, then the first 4 bits yield

$$x_{1,int} = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9,$$

while the remaining 4 bits yield

$$x_{2,int} = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 7.$$

Then, we convert $x_{1,int}$ from the range $[0, 2^l - 1]$ into the range $[0, (x_{1,\max} - x_{1,\min})]$ by multiplying $x_{1,int}$ by

$$\frac{x_{1,\max} - x_{1,\min}}{2^l - 1}. \tag{17.5}$$

To obtain the corresponding value of x_1, we add $x_{1,\min}$ to (17.5) to get

$$x_1 = x_{1,\min} + x_{1,int} \frac{x_{1,\max} - x_{1,\min}}{2^l - 1}. \tag{17.6}$$

We obtain x_2 proceeding in an analogous manner. In general, if $\boldsymbol{x}$ has n components, the mapping from the genotype space into the phenotype space has the form

$$\boxed{x_i = x_{i,\min} + x_{i,int} \frac{x_{i,\max} - x_{i,\min}}{2^l - 1}, \quad i = 1, 2, \ldots, n} \tag{17.7}$$

The next step of the GA algorithm is the application of the selection operator to pick chromosomes from the current population for reproduction to create the new population. In the canonical GA, a pair of chromosomes is selected from the current population with the probability of selection being proportional to the fitness values of the chromosomes. Selection is done with replacement, which means that the same chromosome can be selected more than once for reproduction. A common method of selecting parent chromosomes is the so called *roulette-wheel method*. In this method each chromosome is given a slice of a wheel. The area of the wheel slice is proportional to the chromosome's fitness. The wheel is spun, and the chromosome is selected on whose wheel slice the ball comes to rest. The roulette-wheel method can be implemented in the following steps:

1. Calculate the fitness of each chromosome in the current population;

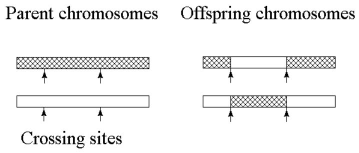


Figure 17.2: The two-point crossover.

2. Calculate the total fitness of the current population by summing the fitnesses of each chromosome in the current population;

3. Divide the fitness of each chromosome by the total fitness;

4. Calculate the cumulative fitness, labeled for our purposes as `cum_fitness`, of the current population using the scaled fitness of each chromosome found in the previous step. You can use MATLAB's command `cumsum` for this purpose.

5. Return the first population member whose corresponding cumulative fitness is greater than the random number from the interval $(0, 1)$. The following MATLAB command can be used here: `find(cum_fitness-rand>0)`, where `cum_fitness` is the vector of the scaled cumulative fitnesses found in step 4.

6. Repeat the above step to get the second parent chromosome.

Using the parent chromosomes picked by the selection operator, we form two offspring by crossing over, with probability p_c, at a randomly chosen crossing site. The uniform probability p_c is called the crossover probability or crossover rate. If no crossover takes place, two offspring chromosomes are formed that are exact copies of their parent chromosomes. One can also use a multi-point crossover operator. The two-point crossover operation is illustrated in Figure 17.2.

The offspring chromosomes obtained from the crossover, are then acted upon by the mutation operator. Mutation is performed at each bit of the offspring chromosomes with probability p_m. The probability p_m is called the mutation probability or mutation rate. The mutation operator randomly flips some of the bits in a chromosome. For example, the chromosome 101010 might be mutated in its third bit to yield 100010. The mutation probability is usually very small, for example $p_m = 0.01$.

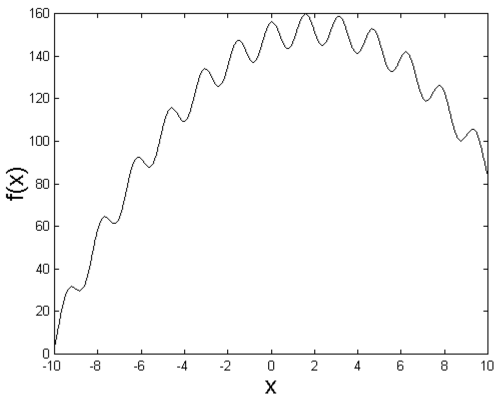


Figure 17.3: A plot of the function given by (17.8).

Each iteration, also called a *generation*, of the GA replaces the current population with a new one. A typical number of iterations of a GA algorithm is from 50 to 500 or more. The entire set of generations is called a *run*.

Example 17.1 We apply the canonical GA described above to maximize the function,

$$f(x) = -15\sin^2(2x) - (x-2)^2 + 160, \quad |x| \le 10. \tag{17.8}$$

This function has many local maximizers and minimizers as can be seen from its plot shown in Figure 17.3. The canonical GA was applied to maximize the function given by (17.8) with the chromosomes composed of 12 bits, $p_c = 0.75$, and $p_m = 0.01$. In Figure 17.4 a summary of a typical run is shown. In this run the best candidate solution was $x = 1.565324$ whose corresponding objective value is $f(x) = 159.809259$. The best solution that was observed in other runs resulted in $f(1.575092) = 159.818346$.

In the next section, we attempt to obtain answers to some whys and hows of genetic algorithms.

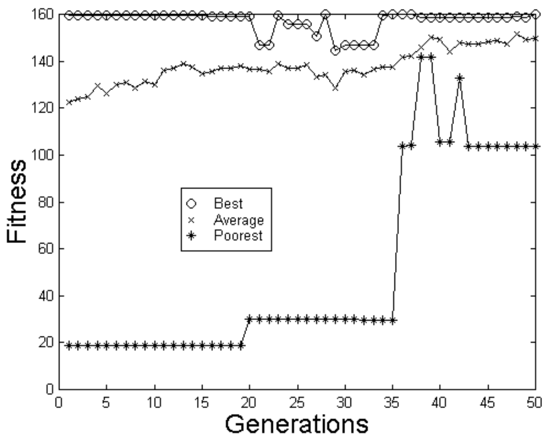


Figure 17.4: Summary of a sample run of the canonical GA when searching for a maximizer of the function given by (17.8).

17.3 Analysis of the Canonical Genetic Algorithm

In the analysis of genetic algorithms, the notion of a *schema* is useful. Plural of "schema" is *schemas* or *schemata*. A schema is a set of bit strings. It can be described using a template made up of ones, zeros, and asterisks that represent "don't cares". Thus, the asterisk $*$ matches either 0 or 1. For example, the schema $H = *0*$ contains the binary strings 000, 001, 100, and 101. Note that a bit string of length l belongs to 2^l different schemas. For example, the string 10 of length $l = 2$ belongs to the schemas $**$, $*0$, $1*$, and 10. Similarly, when $l = 3$, then any bit string of length $l = 3$ belongs to $2^3 = 8$ schemas. Elements of a set of N bit strings of length l are members of between 2^l and $N \times 2^l$ schemas. If all the strings are identical, then they belong to 2^l schemas, otherwise there may be up to $N \times 2^l$ schemas to which the N strings of length l can belong.

The parameters that characterize a given schema are its *order* and *defining length*. The order of a schema H, denoted $o(H)$, is the number of its fixed bits. For example, the order of $H = *01*1$ is $o(H) = 3$ because H has three bits fixed. The defining length of the schema H, denoted $d(H)$, is the distance between its first and the last fixed bits, which is the difference between the positions of the rightmost fixed symbol and the leftmost fixed symbol. The defining length of $H = *0*1$ is $d(H) = 4 - 2 = 2$. In further discussion, the term schema is used to denote a set of binary strings represented by a template, as well as the template itself.

The capability of a chromosome to solve a given optimization problem, that is, the chromosome performance, is measured by its fitness. Similarly, we can characterize the performance of a schema by its average fitness. We now derive an expression for the average fitness of a given schema. Let $\boldsymbol{x}$ be a chromosome that matches the template H, that is, $\boldsymbol{x} \in H$. Then, we say that "$\boldsymbol{x}$ is an instance of H." Let H be a schema with at least one instant in the k-th generation. Denote by $e(H,k)$ the number of instances of H in the k-th generation, and let $\bar{u}(H,k)$ be the average fitness of H in the k-th generation, that is,

$$\bar{u}(H,k) = \frac{\sum_{\boldsymbol{x} \in H} f(\boldsymbol{x})}{e(H,k)}. \tag{17.9}$$

When the genetic algorithm is explicitly evaluating the fitness of N chromosomes from the current generation, it is implicitly evaluating the average fitness of up to $N \times 2^l$ schemas to which the N chromosomes belong.

We now find $\mathcal{E}\,(H, k+1)$, the expected number of instances of the schema H in the $(k+1)$-st generation, given the k-th generation. We first analyze the effect of the selection operator only. Assume that the roulette-wheel method is used in selecting parent chromosomes to the mating pool for reproduction. The probability that the chromosome $\boldsymbol{x}$ is selected to the mating pool, $M(k)$, is

$$\frac{f(\boldsymbol{x})}{\sum_{i=1}^{N} f\left(\boldsymbol{x}_i^{(k)}\right)},$$

where $\boldsymbol{x}_i^{(k)}$ refers to the i-th chromosome in the k-th generation and $f(\boldsymbol{x})$ is the fitness of the chromosome $\boldsymbol{x}$. Therefore, the expected number of chromosomes equal to $\boldsymbol{x}$ that are selected to the mating pool is

$$N \frac{f(\boldsymbol{x})}{\sum_{i=1}^{N} f\left(\boldsymbol{x}_i^{(k)}\right)} = \frac{f(\boldsymbol{x})}{\bar{f}(k)},$$

where

$$\bar{f}(k) = \frac{1}{N} \sum_{i=1}^{N} f\left(\boldsymbol{x}_i^{(k)}\right)$$

is the average fitness of the k-th generation. Let $m(k)$ be the number of chromosomes in $M(k)$ that are instances of H. The expected value, $\mathcal{M}(H,k)$, of the number of chromosomes in the mating pool that are instances of H is

$$\frac{\sum_{\boldsymbol{x} \in H} f(\boldsymbol{x})}{\bar{f}(k)} = \frac{\sum_{\boldsymbol{x} \in H} f(\boldsymbol{x})}{\bar{f}(k)} \frac{e(H,k)}{e(H,k)}. \tag{17.10}$$

Substituting (17.9) into (17.10) yields an alternative expression for the expected value of the number of chromosomes in the mating pool that are instances of H,

$$\boxed{\mathcal{M}(H,k) = \frac{\bar{u}\,(H,k)}{\bar{f}(k)} e\,(H,k)} \tag{17.11}$$

It follows from the above equation that a schema whose average fitness is greater than the population average, will be found in an increasing number of chromosomes.

We next quantify the effects of the one-point crossover on a given schema. Given two chromosomes, where at least one of them is an instance of the schema H, we say that the schema H survives crossover operation if at least one of the offspring is an instance of H. We will give a lower bound on the probability $S_c(H)$ that the schema H will survive the one-point crossover. A schema survives if the crossing site occurs outside the defining length. Crossovers that take place within the defining length can destroy the schema. For example, let $H = *00**$, and let the parent chromosomes be $p_1 = 10011$ and $p_2 = 11100$. Note that $p_1 \in H$, while $p_2 \notin H$. For the crossing site after the third position, there are two offspring chromosomes: $s_1 = 10000$, which is an instance of H, and $s_2 = 11111$, which is not an instance of H. Thus, the schema H survived this crossover. Suppose now that the crossing site is after the second position, which is within the defining length of the schema H. The offspring chromosomes are $s_1 = 10100$ and $s_2 = 11011$. None of the offspring chromosome is an instance of H, which means that H was destroyed by the crossover. If both parent chromosomes are instances of H, then the offspring chromosomes are also instances of H irrespective of the location of the crossing site. An upper bound on the probability $D_c(H)$ that the schema H will be destroyed by the one-point crossover is

$$D_c(H) \leq p_c \frac{d(H)}{l-1}. \tag{17.12}$$

Therefore, a lower bound on the probability $S_c(H)$ that H survives is

$$\boxed{S_c(H) = 1 - D_c(H) \geq 1 - p_c \frac{d(H)}{l-1}} \tag{17.13}$$

Thus, the probability of survival under one-point crossover is higher for short schemas.

A schema H survives under mutation of its element, instance, if all defined bits of the instance of H remain unchanged. Let $S_m(H)$ denote the probability that the schema H survives under mutation of an instance of H. For each defined bit, the probability that it will not be mutated is $1 - p_m$. The probability that no defined bit of H will be mutated is obtained by multiplying $1 - p_m$ by itself $o(H)$ times. Therefore,

$$\boxed{S_m(H) = (1 - p_m)^{o(H)}} \tag{17.14}$$

We combine (17.11), (17.13), and (17.14) to obtain a lower bound on the expected number of instances of H in the $(k+1)$-st generation, given the k-generation,

$$\boxed{\mathcal{E}(H, k+1) \geq \frac{\bar{u}(H,k)}{\bar{f}(k)} e(H,k) \left(1 - p_c \frac{d(H)}{l-1}\right) (1-p_m)^{o(H)}} \tag{17.15}$$

The above is known as the *schema theorem* of Holland. It follows from our discussion that the canonical genetic algorithm implicitly processes, in parallel, schemas. This implicit computation is performed with no operation being explicitly performed on schemas. To illustrate the last statement, consider the string 101. As the fitness of this string is evaluated, partial information is obtained about the expected fitness of all possible schemas to which the evaluated string belongs. In our example, computing the fitness of the string 101 gives partial information about the fitness of strings in the schemas $***$, $**1$, $*0*$, $1**$, $*01$, $1*1, \ldots$.

17.4 Simple Evolutionary Algorithm

Canonical genetic algorithms use binary representations of the elements of the parameter space. The reason behind this is the schema theory, discussed in the previous section, which allows one to analyze canonical genetic algorithms. When using a binary representation of the elements of the parameter space, canonical algorithms use encoding and decoding functions. A binary representation may have some disadvantages though. Indeed, let $h : \{0,1\}^l \to \Omega$ denote a decoding function used in the canonical genetic algorithm implementation. Then the original problem of optimizing $f : \Omega \to \mathbb{R}$ is transformed into the problem of optimizing the combined objective function

$$f(h) : \{0,1\}^l \to \mathbb{R}.$$

The combined objective function to be optimized may be more complex than the original objective function. For example, it may introduce extra nonlinearities or add extra optimizers, which can make the search process for a global optimizer more difficult.

In contrast to the canonical genetic algorithms, in the optimization procedure discussed next, the parameter space is no longer coded using the binary representation. The genetic operators work directly on real-valued vectors. A term used to describe an optimization algorithm using genetic-type operators working on the real-valued vectors is an *evolutionary algorithm*. In what follows, we present a very simple evolutionary algorithm for solving optimization problems of the form given by (17.1).

Algorithm 17.2 Simple Evolutionary Algorithm (EA)

For a given optimization problem of the form (17.1) perform the following steps:

Step 1. Initialize a population of candidate solutions (chromosomes) to the problem. Each chromosome is a real-valued vector of the same length as the vector $\boldsymbol{x}$ of decision variables.

Step 2. Evaluate the fitness of each candidate solution in the population.

Step 3. Replace the current population with a new population by applying selection, crossover, and mutation operations on the current population. The roulette-wheel selection method can be used for selection. A possible implementation of the crossover operation is to use the average crossover operator. This operator takes two parents, say $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ and returns one offspring

$$\boldsymbol{x} = \frac{1}{2}\left(\boldsymbol{x}_j + \boldsymbol{x}_k\right).$$

Another implementation of the crossover operator is a convex combination of two parents. Specifically, let $r \in (0, 1)$. Then, the offspring is

$$\boldsymbol{x} = r\boldsymbol{x}_j + (1 - r)\boldsymbol{x}_k.$$

Note that if the feasible set is convex, then the above crossover operators produce chromosomes that are elements of the feasible set. Otherwise conditions should be imposed to ensure that candidate solutions are the elements of the feasible set.

As for the mutation operation, let $\boldsymbol{x}$ be a chromosome to be mutated. Then, an implementation of the mutation operation could have the form,

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{d},$$

where $\boldsymbol{d}$ is a randomly generated feasible vector so that $\boldsymbol{x}'$ is in the feasible set.

Step 4. If a stopping criterion is satisfied, terminate the algorithm and choose the fittest chromosome from the population as a solution, else go to **Step 2.**

In addition, both in the canonical GA, as well as in the evolutionary algorithm, the elitist strategy, introduced by K. DeJong in 1975, can be incorporated. This strategy copies the best chromosomes from the old population into the next population to prevent losing the best solutions in the succeeding iterations. As observed by Mitchell [243], among others, the elitist strategy may lead to the domination of a population by a "super chromosome". However, the elitist strategy improves genetic and evolutionary algorithms performance. Rudolph [286] showed in 1994 that a canonical GA will never converge to the global optimum unless the best solution in the population, either before or after selection, is copied into the next population.

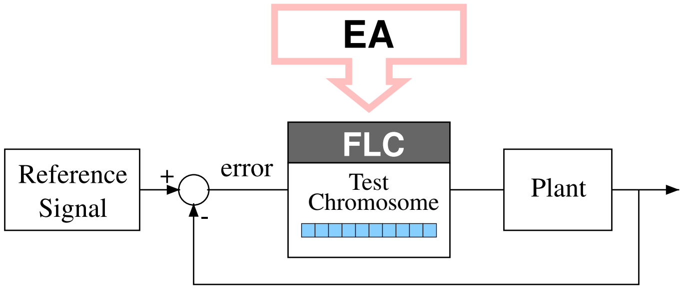


Figure 17.5: A configuration used to design a fuzzy logic controller using an evolutionary algorithm.

17.5 Evolutionary Fuzzy Logic Controllers

In this section, we employ evolutionary algorithms to design tracking controllers. We describe three design techniques for evolutionary fuzzy logic controllers for the step-lane-change maneuver of a ground vehicle. The presentation of this section is based on [207]. We mention that the step-lane-change maneuver is an important aspect in the design of the Intelligent Vehicle/Highway System (IVHS), the highway of the future.

An essential element in combining an evolutionary algorithm (EA) and a fuzzy logic controller (FLC) is the encoding method used to represent a fuzzy rule base as a chromosome on which the EA operates. Selecting an appropriate encoding procedure is a critical factor that affects the efficiency of the EA and the final solution. Another essential element is the fitness function. Each chromosome defines a fuzzy rule base. After conducting a simulation of the structure depicted in Figure 17.5 and measuring the tracking error, the fitness of the chromosome representing the fuzzy rule base of the simulated FLC is calculated. With all the chromosomes assigned a fitness value, genetic operators yield a new generation and the entire procedure is repeated until a stopping criterion, usually the number of iterations or the fitness value, is satisfied.

17.5.1 Vehicle Model and Control Objective

We use a “bicycle model” of a front-wheel-steering vehicle. Modeling of a ground vehicle is covered in detail in Subsection 2.4.2. The standard coordinate system of the Society of Automotive Engineers (SAE) is used. A schematic of the model used is shown in Figure 17.6. The modeling equations are:

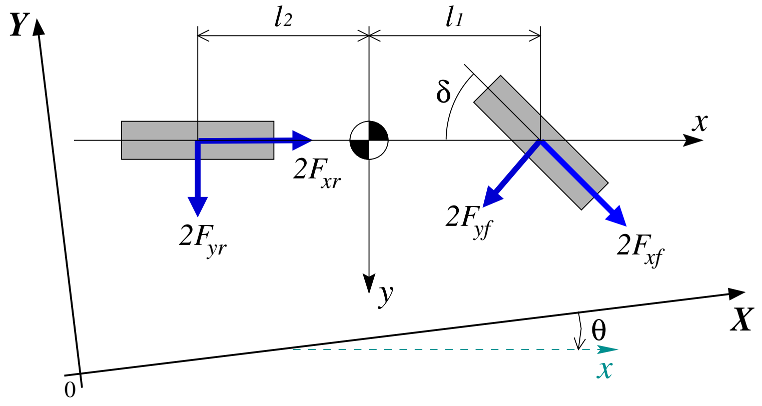


Figure 17.6: Schematic of a bicycle model of a front-wheel-steering vehicle.

Table 17.1: Vehicle data and symbols used in modeling equations.

δ	steering angle of the front wheel	
F_{yf}	lateral force of the front tire	
F_{yr}	lateral force of the rear tire	
m	total mass of the vehicle	1280 kg
I	moment of the inertia of the vehicle	2500 kg·m^2
l_1	distance from center of gravity to front axle	1.2 m
l_2	distance from center of gravity to rear axle	1.22 m
v_x	longitudinal velocity of the vehicle	18.3 m/s

$$\left\{\begin{array}{lcl} \dot{x}_1 & = & -v_x x_3 + \frac{1}{m}\left(F_{yf} + F_{yr}\right) \\ \dot{x}_2 & = & x_3 \\ \dot{x}_3 & = & \frac{2}{I}(l_1 F_{yf} - l_2 F_{yr}) \\ \dot{x}_4 & = & -v_x x_2 - x_1, \end{array}\right.$$

where $x_1 = \dot{y}$ is the lateral velocity of the vehicle, $x_2 = \theta$ is the yaw angle, $x_3 = \dot{\theta}$ is the yaw velocity, and $x_4 = Y$ is the lateral position of the vehicle. Other parameters and their values are given in Table 17.1. The lateral forces $F_{yf} = F_{yf}(\alpha_f)$ and $F_{yr} = F_{yr}(\alpha_r)$ are nonlinear functions of respective slip angles with a typical plot of a lateral force versus tire slip angle shown in Figure 2.10. Recall that the slip angles for the front and rear tires are

$$\alpha_f = \delta - \frac{l_1 x_3 + x_1}{v_x} \quad \text{and} \quad \alpha_r = \frac{l_2 x_3 - x_1}{v_x}.$$

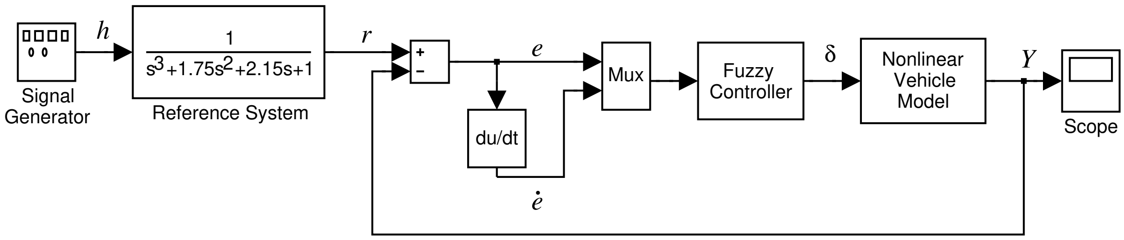


Figure 17.7: SIMULINK block diagram of the closed-loop vehicle system driven by a fuzzy logic controller.

The input is the steering angle δ of the front wheel and the output is the lateral position of the vehicle, $x_4 = Y$.

The step-lane-change maneuver along with cruising, traveling path selection, highway exiting and entering is one of the essential elements of automated vehicle control. However, automation of these tasks requires a roadway reference system that involves sensing for lane detection. In our simulation, we assume that the road reference system is already installed. Our objective is to design a fuzzy logic controller that forces the vehicle to track a given reference lateral position signal r. The reference signal is generated by a reference signal generator whose transfer function is $1/(s^3 + 1.75s^2 + 2.15s + 1)$. The tracking error e is the difference between the reference lateral position signal r and the actual lateral position Y of the vehicle. The role of the fuzzy controller is to produce the steering wheel angle δ using the information of the tracking error e and the change in the tracking error $\dot{e}$ so that the vehicle tracks the reference signal. A SIMULINK block diagram of the closed-loop vehicle system is depicted in Figure 17.7. In the following, we describe three different methods for an evolutionary algorithm (EA) aided fuzzy logic controller (FLC) design.

17.5.2 Case 1: EA Tunes Fuzzy Rules

In this case, we assume that the membership functions are fixed. In our simulation experiment, we use the membership functions shown in Figure 17.8. The labels of fuzzy sets LN, N, Z, P, and LP denote large negative, negative, zero, positive, and large positive. There are five fuzzy sets for e and five fuzzy sets for $\dot{e}$, hence we can have 25 different rules of the form:

$$\text{IF } e \text{ is LN AND } \dot{e} \text{ is LN then } \delta \text{ is SP}$$
$$\vdots$$
$$\text{IF } e \text{ is LP AND } \dot{e} \text{ is LP then } \delta \text{ is SN.}$$

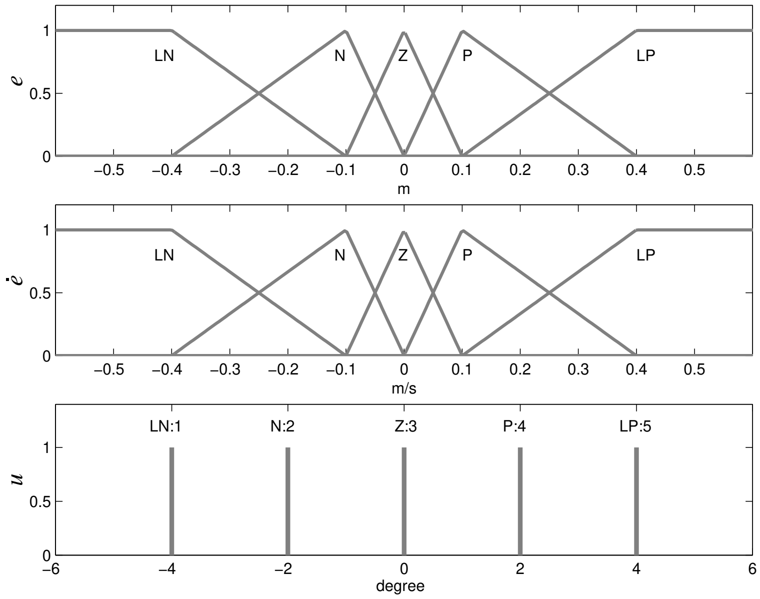


Figure 17.8: Fuzzy membership functions for e, $\dot{e}$, and u.

These rules can be arranged into a rules table as shown in Figure 17.9(a). We mention here that the rules shown in Figure 17.9(a) were arrived at heuristically. Using the rules table, we construct a FLC using the center average defuzzifier given by (15.2). The control surface derived using the rules in Figure 17.9(a) is shown in Figure 17.9(b). Tracking performance for two different initial conditions $Y(0) = 5$ m and $Y(0) = -5$ m is illustrated in Figure 17.9(c). We calculated tracking errors for each initial condition and labeled the tracking error for the first initial condition as e_1 while the tracking error of the second initial condition was labeled as e_2. We then constructed the performance index (PI) as the sum of the integrals of the square of the error (ISE) over the simulation time,

$$PI = \int_0^{50} \left(e_1^2(t) + e_2^2(t)\right) dt.$$

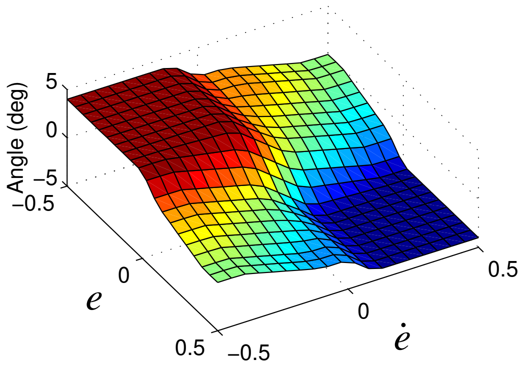
The value of the PI in the above experiment is 107.04. The performance index, PI, was evaluated using the circuit shown in Figure 17.10.

Of course, we could include more error trajectories in the expression for the PI. However, for the sake of simplicity, we decided to use only two tracking error trajectories in our simulation experiments.

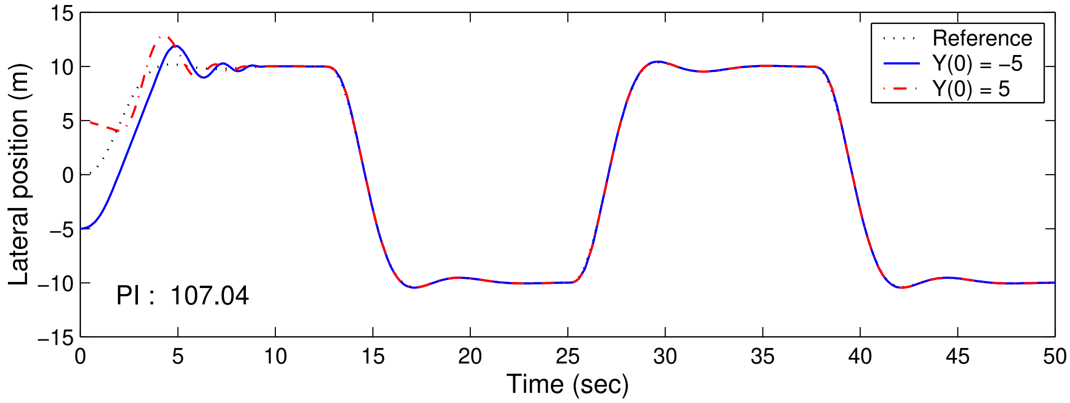
We next employ an EA to generate a fuzzy rules table automatically. In order to represent the whole fuzzy rules table as a chromosome, we encode the labels of the fuzzy singletons corresponding to the controller outputs as integers, where LN is represented by 1, N by

e \ $\dot{e}$	LN	N	Z	P	LP
LN	LP	LP	LP	P	Z
N	LP	LP	P	Z	N
Z	P	P	Z	N	N
P	P	Z	N	LN	LN
LP	Z	N	LN	LN	LN

(a)



(b)



(c)

Figure 17.9: FLC constructed using a fuzzy rule base obtained heuristically: (a) fuzzy rules table (b) control surface (c) tracking performance.

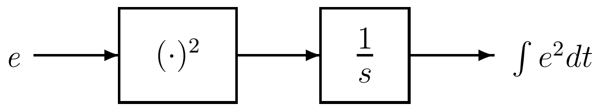


Figure 17.10: Circuit used to evaluate the performance index, PI.

2, Z by 3, P by 4, and LP by 5. This encoding of the fuzzy singletons of the control signal u is depicted in Figure 17.8. The actual control signal, however, is produced as the weighted sum of the fuzzy singletons using the center average defuzzifier. Because there are 25 rules, each chromosome is a string composed of 25 integers. For example, the rules table of Figure 17.9(a) can be represented as a chromosome of the form,

5	5	5	4	3	5	5	4	3	2	4	4	3	2	2	4	3	2	1	1	3	2	1	1	1

In our experiment, we decided on the population size of 30. The roulette wheel method was used to select chromosomes to the mating pool. We employed the single-point crossover operator with the crossover rate $p_c = 0.9$. The role of the mutation operator was to replace each element of a chromosome randomly with an integer from the set $\{1, \ldots, 5\}$. We used the mutation rate $p_m = 0.05$.

We also employed an elitist strategy where two best individuals of the current generation were passed unchanged to the next generation. The fitness function employed was the inverse of the PI discussed above, that is,

$$Fitness = \frac{1}{PI} = \frac{1}{\int_0^{50} \left(e_1^2(t) + e_2^2(t)\right) dt}. \tag{17.16}$$

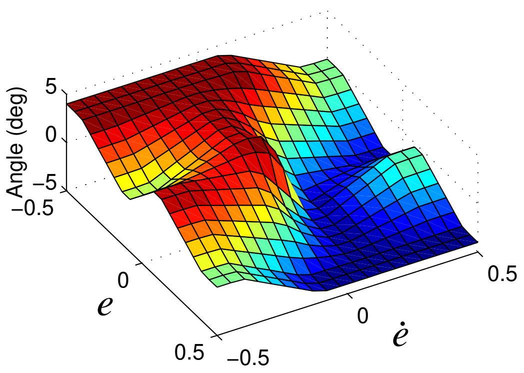
Recall that $e_1(t)$ is the tracking error with the initial lateral position at 5 m, and $e_2(t)$ is the tracking error with the initial lateral position at -5 m. The input to the reference system was a square wave with the amplitude of 10 m and the frequency of 0.04 Hz. The rules table of the best FLC, after 50 iterations of the EA, is shown in Figure 17.11(a). Figure 17.11(b) shows the fuzzy control surface based on the rules table of Figure 17.11(a). Two different initial conditions, $Y(0) = 5$ m and $Y(0) = -5$ m were used to test the tracking performance of the fuzzy controller; Figure 17.11(c) depicts the results. The vehicle lateral position becomes almost indistinguishable from the reference position after about seven seconds. The value of the PI is 97.10, which is an improvement over the performance of the FLC obtained using heuristically generated fuzzy rules.

17.5.3 Case 2: EA Tunes Fuzzy Membership Functions

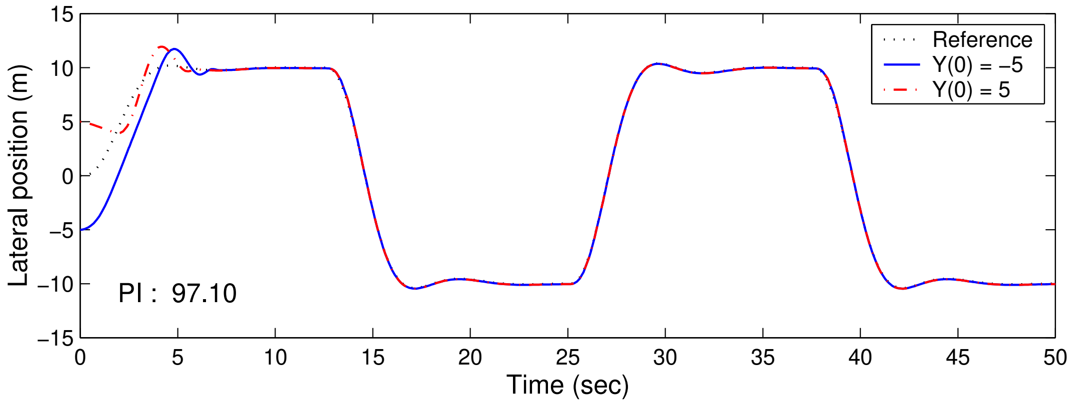
In this case, fuzzy rules are assumed to be known and fixed. The role of an EA is to tune the input and output membership functions. We use the heuristically obtained rules table of Figure 17.9(a) as the given fuzzy rule base. The fuzzy sets for each input can be completely described using two real numbers if the corresponding fuzzy membership functions are symmetric with respect to the origin and, at each point of their universe of discourse, the

e \ $\dot{e}$	LN	N	Z	P	LP
LN	LP	Z	P	LP	Z
N	LP	LP	LP	P	LN
Z	LP	P	Z	N	LN
P	LP	N	LN	LN	LN
LP	Z	LN	N	Z	LN

(a)



(b)



(c)

Figure 17.11: The best FLC of the 50-th generation using a fuzzy rule base automatically generated by the EA: (a) fuzzy rules table (b) control surface (c) tracking performance.

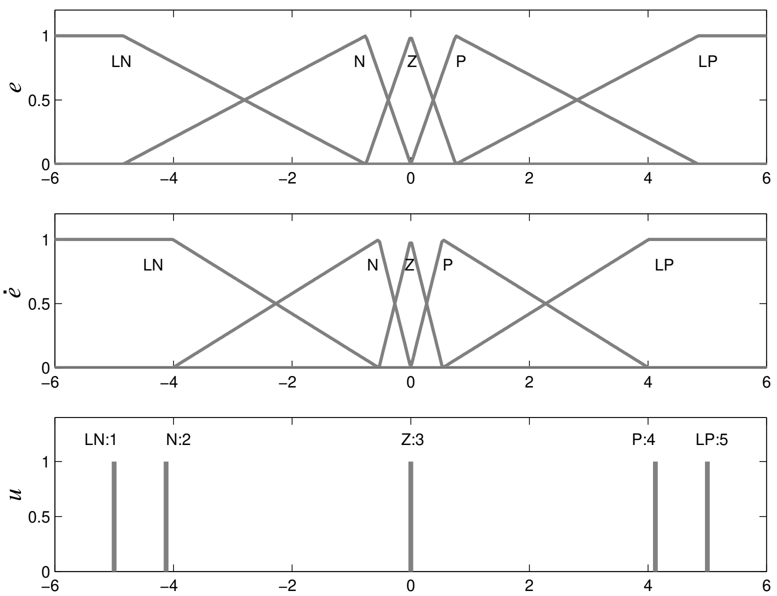


Figure 17.12: Input and output fuzzy membership functions of the best FLC of the 50-th generation obtained by tuning fuzzy membership functions.

sum of the values of the membership functions equals 1. The controller output fuzzy sets are also assumed to be symmetric with respect to the origin. Under the above assumptions, the controller output fuzzy singletons can be completely characterized using just two parameters. Hence, six real parameters can completely characterize all the input and output membership functions. So the chromosomes are strings of just six real parameters. For example, the fuzzy sets of Figure 17.8 can be represented as the string,

0.1	0.4	0.1	0.4	2	4

We proceed with a description the EA that operates on the above described fuzzy rule base. The population size, the selection method, the crossover operator and its rate, the mutation operator and its rate, and the fitness function, as well as other parameters, are the same as in the previous case. The resulting fuzzy sets, after 50 iterations of the EA, are shown in Figure 17.12. The control surface and the tracking performance of the closed-loop system driven by the FLC constructed using fuzzy sets of Figure 17.12 are depicted in Figure 17.13. The PI value is 43.51. The tracking error of the lateral position almost disappears in about four seconds.

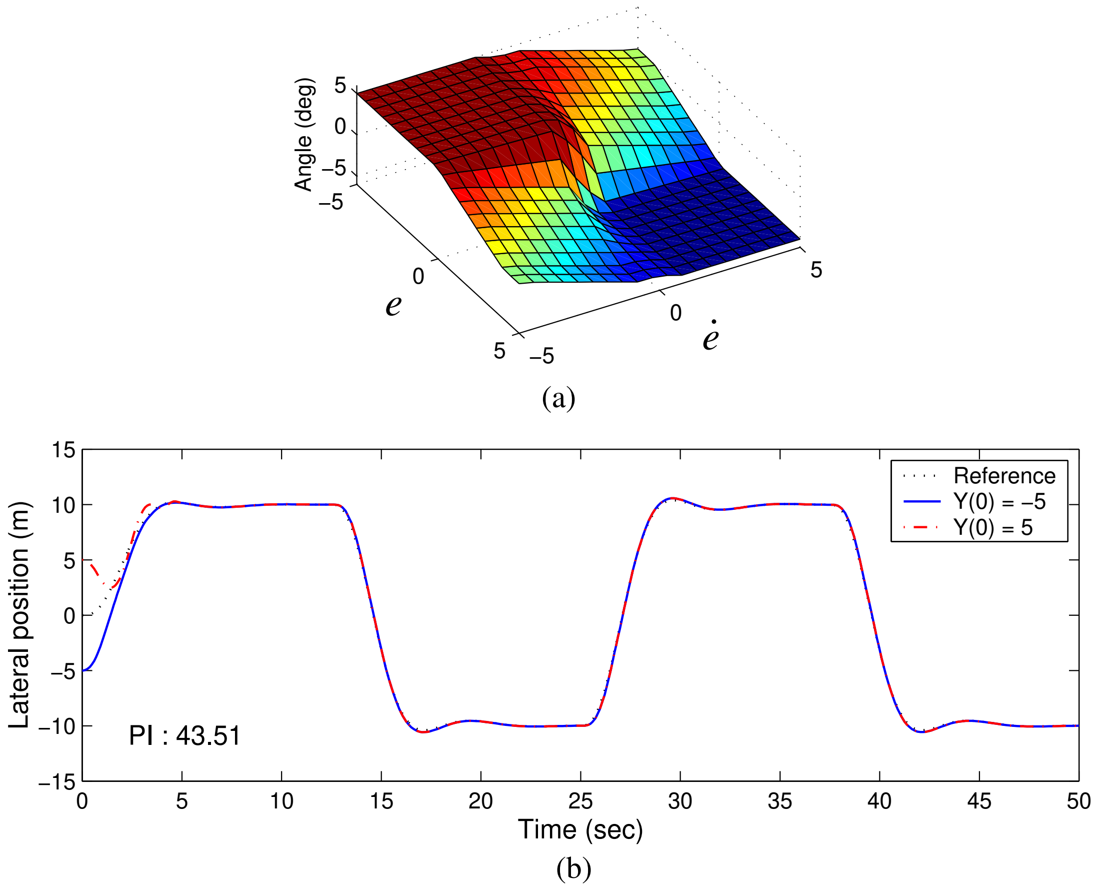


Figure 17.13: The best FLC of the 50-th generation obtained by tuning fuzzy membership functions: (a) control surface (b) tracking performance.

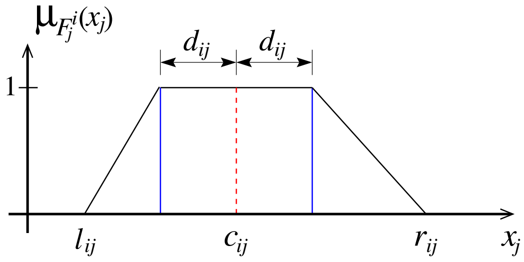


Figure 17.14: An example of a trapezoidal membership function.

17.5.4 Case 3: EA Tunes Fuzzy Rules and Membership Functions

We observed in the previous cases that changing fuzzy rules or modifying the shape of membership functions can improve the performance of the resulting FLC. So let us analyze an EA that simultaneously operates on fuzzy rules and fuzzy membership functions. This approach is a modification of the technique described in [155, 174].

In general, an i-th fuzzy rule has the form,

$$\text{IF } x_1 \text{ IS } F_1^i \text{ AND } \cdots \text{ AND } x_n \text{ IS } F_n^i \text{ THEN } u_1 \text{ IS } \theta_1^i \text{ AND } \cdots \text{ AND } u_m \text{ IS } \theta_m^i$$

where $1 \leq i \leq r$, x_js are the state variables where $1 \leq j \leq n$, and u_ks being the controller's outputs, where $1 \leq k \leq m$. F_j^is are fuzzy sets with trapezoidal membership functions, and θ_k^is are the controller's output variables whose membership functions are singletons. A membership function, $\mu_{F_j^i}(x_j)$, of the fuzzy set, F_j^i, can be completely characterized using four parameters c_{ij}, d_{ij}, l_{ij}, and r_{ij}. It can be described as

$$\mu_{F_j^i}(x_j) = \begin{cases} \frac{x_j - l_{ij}}{c_{ij} - l_{ij} - d_{ij}} & \text{if } \; l_{ij} < x_j < c_{ij} - d_{ij} \\ 1 & \text{if } \; c_{ij} - d_{ij} \leq x_j < c_{ij} + d_{ij} \\ -\frac{x_j - r_{ij}}{r_{ij} - c_{ij} - d_{ij}} & \text{if } \; c_{ij} + d_{ij} < x_j < r_{ij} \\ 0 & \text{otherwise,} \end{cases} \tag{17.17}$$

or equivalently,

$$\mu_{F_j^i}(x_j) = \max\left(0, \min\left(1, \frac{x_j - l_{ij}}{c_{ij} - l_{ij} - d_{ij}}, -\frac{x_j - r_{ij}}{r_{ij} - c_{ij} - d_{ij}}\right)\right). \tag{17.18}$$

Figure 17.14 shows a generic fuzzy membership function of the form given by (17.17) or, equivalently by (17.18). The fuzzy singletons, on the other hand, are described as

$$\mu_{\theta_k^i}(z) = \begin{cases} 1 & \text{if } z = \theta_k^i \\ 0 & \text{otherwise.} \end{cases} \tag{17.19}$$

The controller's signal u_k is calculated using the center average defuzzification formula,

$$u_k = \frac{\sum_{i=1}^{r} w_i \theta_k^i}{\sum_{i=1}^{r} w_i}, \tag{17.20}$$

where r is the number of fuzzy rules and the firing strength, $w_i = \mu_{F_1^i} \times \mu_{F_2^i} \times \cdots \times \mu_{F_n^i}$.

We present a method of representing a given fuzzy rule base as a chromosome. In order to tune fuzzy rules and to alter the shape of membership functions, we require that a chromosome contains all the information pertaining to the structure of the fuzzy rules and the shape of membership functions explicitly. A chromosome is composed of two substructures: one substructure contains the fuzzy rules' structure matrix while the other substructure contains the information concerning the parameters of the membership functions of the fuzzy rule base. The fuzzy rules' structure matrix contains the information related to the absence or presence of the premise part of the fuzzy rule corresponding to the particular controller input. Hence the elements of this matrix are from the set $\{0, 1\}$.

The parameter matrix contains the parameters that describe fuzzy membership functions. As previously mentioned, there are four parameters for each fuzzy trapezoidal membership function and one for each fuzzy singleton. The parameter matrix has the same number of rows as the number of rules and the number of columns is equal four times the number of the controller inputs plus the number of the controller's outputs. Figure 17.15(a) shows an example of the rules' structure matrix and the parameter matrix for the controller with two inputs and one output whose operation is described by two IF-THEN rules. The fuzzy rules and membership functions corresponding to the rules' structure matrix and the parameter matrix in Figure 17.15(a) are depicted in Figure 17.15(b). Note that a 0 in the second column of the first row of the rules' structure matrix means that the premise part of the first fuzzy rule corresponding to the second controller input is absent; that is, the first rule does not involve the controller's second input.

The fitness function has the form,

$$\begin{aligned} Fitness &= \frac{1}{PI + \alpha \times (\text{number of rules})} \\ &= \frac{1}{\int_0^{50} \left(e_1^2(t) + e_2^2(t)\right) dt + \alpha \times (\text{number of rules})}, \end{aligned} \tag{17.21}$$

where $e_1(t)$ is the tracking error with the initial lateral position at 5 m, and $e_2(t)$ is the tracking error with the initial lateral position at -5 m. The reference signal used has the amplitude of 10 m and the frequency of 0.04 Hz over 50 seconds. The design parameter α is employed to penalize for the excessive number of fuzzy rules. In our simulations, $\alpha = 1$.

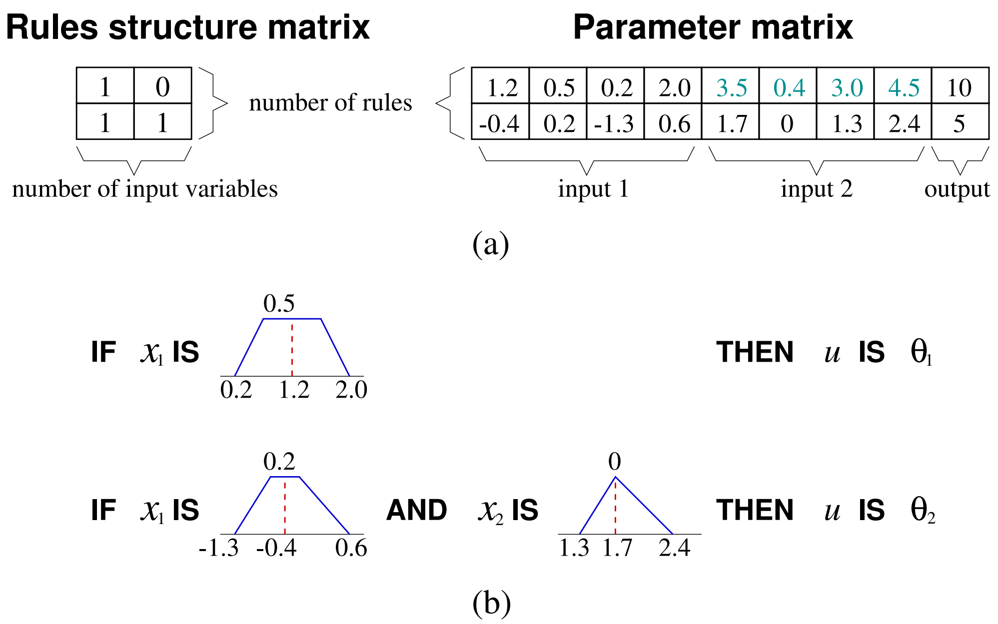


Figure 17.15: An example of a chromosome structure and its corresponding fuzzy rule base where the controller has two inputs, x_1 and x_2, and one output u.

We next describe genetic operators utilized by the EA. Three genetic operators are used with the probability p_g. We first discuss a genetic operator that modifies the number of fuzzy rules. Let r_{new} be the number of fuzzy rules after the genetic operator is applied while r_{old} be the number of fuzzy rules before the genetic operator is applied. Then,

$$r_{new} = r_{old} + \Delta r,$$

where Δr is the correction term that we analyze next. The number of fuzzy rules cannot exceed a prespecified bound r_{max}. It is reasonable to assume that on the average r_{old} is about $r_{max}/2$. Therefore, we wish to construct the correction term so that

$$\Delta r \in [-r_{max}/2, r_{max}/2].$$

Let Ψ be a random variable with standard normal distribution $N(0,1)$. The probability that Ψ will take its values in the interval $(-3,3)$ is 0.99. Let integer$(\cdot)$ be the operator that produces the nearest integer value of its argument. Then, the random variable, integer$(\Psi r_{max}/6)$, will take its value in the interval $(-\text{integer}(r_{max}/2), \text{integer}(r_{max}/2))$ with the probability 0.99. In view of the above, we use the following expression for updating the number of fuzzy rules,

$$r_{new} = r_{old} + \text{integer}(\Psi r_{max}/6). \tag{17.22}$$

This particular form of the genetic operator ensures that the variation range of the rules' number is reasonable with respect to the maximum number r_{max} of the rules allowed. If the number of new rules, r_{new}, falls outside the allowed range, then we set $r_{new} = r_{old}$.

The second genetic operator acts upon the elements of the rules' structure matrix by taking the binary complement of each element, that is, $1 \rightarrow 0$ and $0 \rightarrow 1$ with probability p_g.

The third genetic operator acts upon the parameter matrix as per the following expression,

$$p_{ij}^{new} = p_{ij}^{old} + \eta_{ij} \Psi e^{1-F_{old}/F_{max}}, \tag{17.23}$$

where p_{ij}^{old} is the current value of the (ij)-th element of the parameter matrix, p_{ij}^{new} is the updated value of the (ij)-th element of the parameter matrix, Ψ is a random variable with $N(0,1)$, F_{old} is the fitness value of the chromosome being operated upon, and F_{max} is the maximum fitness value of the chromosomes of the current population. Each design parameter η_{ij} was chosen according to the following recipe,

$$\eta_{ij} = (\text{maximal range of } p_{ij} - \text{minimal range of } p_{ij})/20.$$

The term $e^{1-F_{old}/F_{max}}$, in equation (17.23), allows a chromosome with low fitness to vary the elements of the parameter matrix in a larger range than for the chromosomes with high fitness values.

In our implementation of the EA, we employ the elitist strategy to ensure that 20% of the fittest chromosomes in each generation are passed directly to the next generation without being acted upon by the genetic operators. The genetic operators, described above, are applied to the top 20% of the chromosomes in the current generation to produce the remaining 80% of the chromosomes of the new population.

After applying genetic operators to the chromosomes of the current generation, post-processing may be required to ensure that the resulting chromosomes are feasible. If a zero row in the rules' structure matrix is created, then that row is deleted and the corresponding row in the parameter matrix is also deleted. The following conditions must be satisfied by the parameter matrix for the chromosome containing this matrix to be feasible,

$$\left.\begin{array}{lcl} d_{ij} & \geq & 0 \\ l_{ij} & \leq & c_{ij} - d_{ij} \\ r_{ij} & \geq & c_{ij} + d_{ij}. \end{array}\right\}$$

If the above conditions are violated by a parameter, then that parameter is assigned a boundary value. For example, if $d_{ij} < 0$, then d_{ij} is set to 0, and if $l_{ij} > c_{ij} - d_{ij}$, then l_{ij} is set to $c_{ij} - d_{ij}$, etc.

The above described EA is used to construct a fuzzy logic controller for the front-wheel-steering ground vehicle model, whose dynamics are characterized by a certain type of symmetry that allows us to simplify computations. Specifically, suppose that in the rule base we have a fuzzy rule of the form,

$$\text{IF } x_1 \text{ is } F_1 \text{ and } x_2 \text{ is } F_2 \text{ THEN } u \text{ is } \theta,$$

where the fuzzy sets F_1 and F_2 are specified by the membership functions $\mu_{F_1}(c_1, d_1, r_1, l_1)$ and $\mu_{F_2}(c_2, d_2, r_2, l_2)$, respectively. The parameters c_i, d_i, r_i, and l_i, $i = 1, 2$, characterize membership functions of the form shown in Figure 17.14. Because of the symmetry in the vehicle dynamics, the rule base also contains the rule

$$\text{IF } x_1 \text{ is } \tilde{F}_1 \text{ and } x_2 \text{ is } \tilde{F}_2 \text{ THEN } u \text{ is } -\theta,$$

where $\tilde{F}_1$ and $\tilde{F}_2$ have membership functions

$$\mu_{\tilde{F}_1} = \mu_{F_1}(-c_1, d_1, -l_1, -r_1) \quad \text{and} \quad \mu_{\tilde{F}_2} = \mu_{F_2}(-c_2, d_2, -l_2, -r_2),$$

respectively.

The size of the population in our simulations was 30, the maximum number of rules was $r_{max} = 10$, and the probability p_g was equal to 0.1. A fuzzy rule base obtained after 100 iterations is shown in Figure 17.16. Only four rules were generated. The fuzzy rule base shown in Figure 17.16 was then used to construct a fuzzy logic controller that was connected to the vehicle model in the closed-loop configuration shown in Figure 17.7. The fuzzy control surface constructed using the obtained fuzzy rule base is shown in Figure 17.17(a). The results of a typical simulation run, the fitness values of the best, average, and worst chromosomes in each generation are shown in Figure 17.17(b), while the tracking performance is illustrated in Figure 17.17(c). As in the previous cases, the simulations were performed for two different initial conditions, and excellent tracking was achieved in both cases. The value of the PI was reduced to 38.06, which is lower than in the two previous designs.

17.6 Notes

Basic principles of genetics have their roots in the experiments performed by a nineteenth-century monk, Gregor Mendel in his monastery garden in Brno now in the Czech Republic. Mendel worked with the garden pea (*Pisum sativum*) because it possesses traits that are easily recognizable and appear in widely contrasting forms. In addition the garden pea lends itself well to controlled crosses. Mendel published his findings in 1866 in a paper titled,

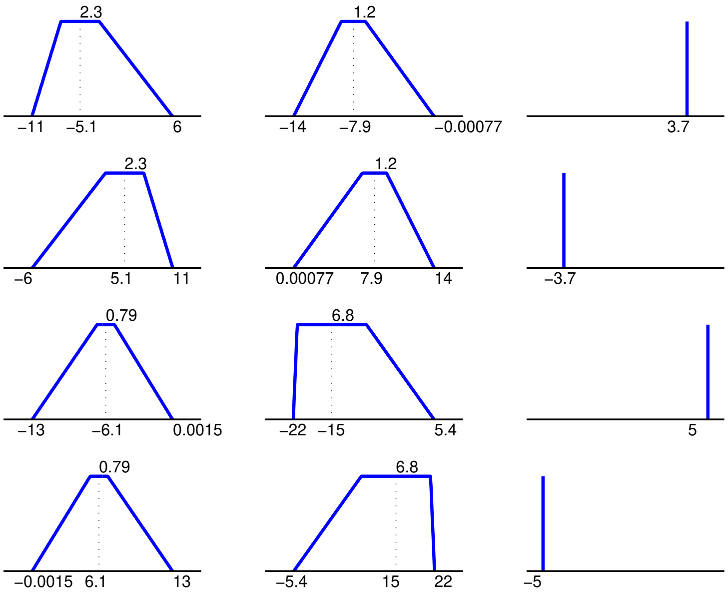


Figure 17.16: The best fuzzy rule base after 100 iterations of the EA operating on fuzzy membership functions and fuzzy rule base.

"Experiments in Plant Hybridization". According to Elseth and Baumgardner [95, page 3], "The publication of this paper marked the birth of genetics as a science. The principles stated in the paper still stand today as the cornerstone of modern genetics." The importance of Mendel's work went completely unrecognized for 34 years. "In 1900, three botanists, Hugo de Vries (Dutch), Carl Correns (German), and Erich von Tschermak-Seysenegg (Austrian), simultaneously discovered Mendel's paper on heredity, recognized its importance, and cited it in their own publications." [95, page 7]. We add that the term *gene* was coined by W. L. Johannsen in 1903.

A seminal text on genetic algorithms by Holland [139] is definitely worth perusing. However, Holland's monograph is not intended as an introduction to genetic algorithms. A very well written introductory text to genetic algorithms is Mitchell's book [243]. A number of engineering design examples using genetic algorithms can be found in Gen and Cheng [113]. Bäck, Hammel, and Schwefel [17] survey the history as well as the current state of evolutionary computation which is the term that encompasses evolution strategies, evolutionary programming, and genetic algorithms. Applications of genetic algorithms to automatic programming using evolving computer programs are described by Koza [189]. Automatic programming with genetic algorithms is called by Koza "genetic programming" (GP). Robust controller design with genetic algorithms are reported by Patton and Liu [266], and Ge, Lee,

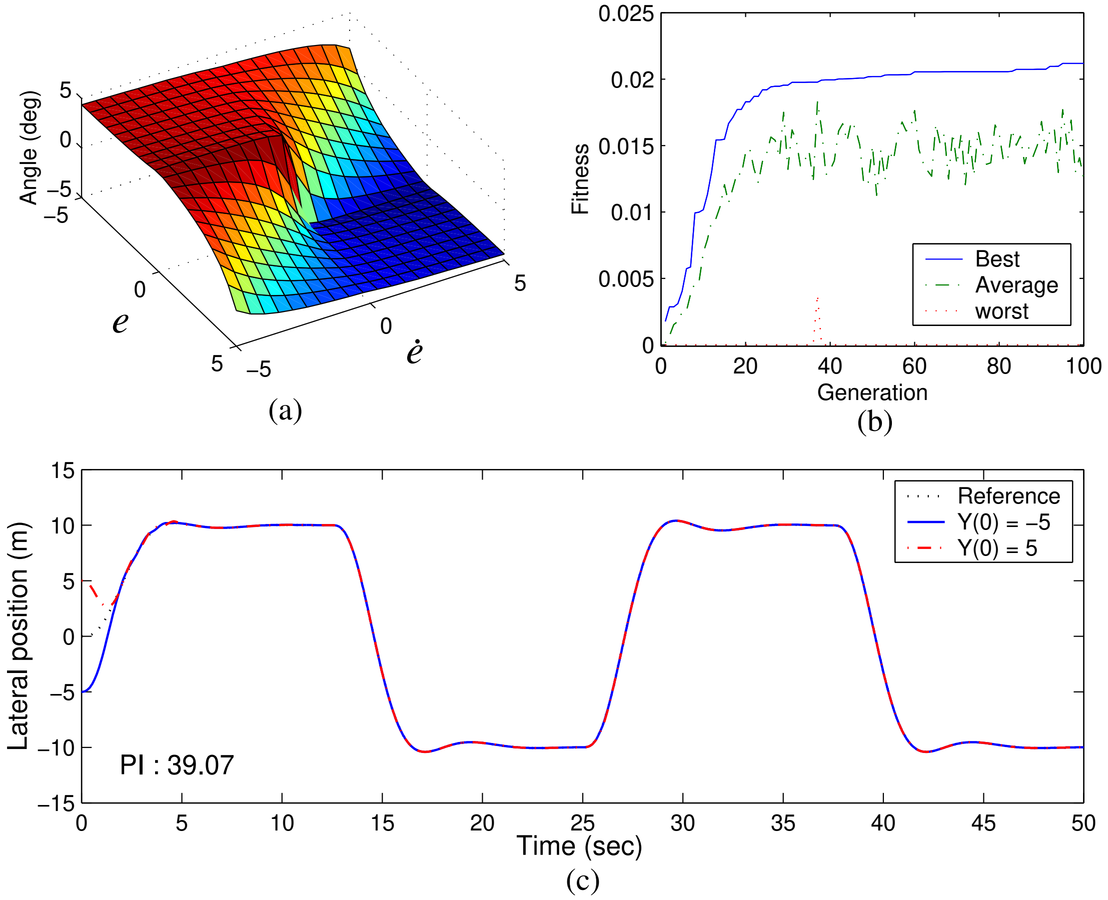


Figure 17.17: Summary of the simulation run of the closed-loop system driven by the fuzzy logic controller obtained from the fuzzy rule base shown in Figure 17.16: (a) control surface (b) fitness values of the best, average, and worst chromosomes (c) tracking performance.

and Zhu [111]. Kozek, Roska, and Chua [191], Pal and Bhandari [261], and Zamparelli [371] use genetic algorithms to train cellular neural networks. Kim and Myung [180] apply genetic algorithms to solve constrained optimization problems, while Koza et al. [190] employ genetic algorithms for the automatic design of electrical circuits. Yang, Hachino, and Tsuji [362] combine least-squares techniques with a genetic algorithm to on-line identification of time-delay systems. More recently, Loop et al. [220] use genetic algorithms to estimate regions of asymptotic stability of power electronics systems.

17.7 Exercises

Exercise 17.1 Recall that the "elitism" is a genetic operator that forces the GA, or EA, to retain some number of the best individuals from the current population in the new population. Incorporate the elitism operator into your GA program and test it on the function (17.8). You may try, for example, to keep the best two individuals from the current population in the new population.

Exercise 17.2 Implement the simple EA from Section 17.4 and test it on:

(i) the function (17.8);

(ii) the function $f(x) = x + |\sin(32x)|$, where $0 \le x < \pi$.

The function in (ii) comes from Mitchell [243, page 9].

Exercise 17.3 Implement the simple EA from Section 17.4 that maximizes the MATLAB's peaks function

$$\begin{aligned} f(x,y) &= 3(1-x)^2 \exp(-x^2-(y+1)^2) - 10\left(\frac{x}{5} - x^3 - y^5\right)\exp(-x^2-y^2) \\ &\quad -\frac{1}{3}\exp(-(x+1)^2-y^2), \end{aligned}$$

where $-3 \le x, y, \le 3$.

Exercise 17.4 Implement the simple EA from Section 17.4 that maximizes the function

$$f(x,y) = 3 + y\sin(0.7x) + x\cos(1.5y);$$

where $-3.0 \le x \le 12.1$ and $4.1 \le y \le 5.8$.

Exercise 17.5 Consider the classic traveling salesman problem (TSP): Suppose a salesman must visit clients in different cities, and then return home. What is the shortest tour through those cities, visiting each one once and only once? Implement in MATLAB a simple variant of the genetic algorithm that optimizes the salesman route for 10 cities. How many different possible paths are there for 10 cities? Following Fogel [105, page 29], use the following representation of any candidate solution: for a given map of the cities to be visited (including the salesman's home base), assign to each city a positive integer. Thus, for example, if there were six cities, one possible solution might be $\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$. The above represents an order of progression of the salesman's trip. Note that because the problem requires a round trip, the first coordinate of the vector representing the first city of the trip is also the last city visited by the salesman. Because the trip is a closed loop, it does not matter from which city the salesman starts his or her trip. As far as a crossover operator implementation is concerned, we recommend using Fogel's idea [105, page 29], where a single-parent operator produces an offspring by inverting the visiting order between two randomly chosen coordinates of the parent vector. As an example of this implementation of the single-parent crossover operator, suppose that the parent is as above and that the randomly chosen coordinates, that is, the inversion points, are the second and fifth ones. The offspring would then be $\begin{bmatrix} 1 & 5 & 3 & 4 & 2 & 6 \end{bmatrix}$.

Plot a map with the cities and mark the obtained salesman's route. The coordinates of the cities are given in the following table.

Table 17.2: Locations of the cities in Exercise 17.5.

x	0.4306	3.7094	6.9330	9.3582	4.7758	1.2910	4.83831	9.4560	3.6774	3.2849
y	7.7288	2.9727	1.7785	6.9080	2.6394	4.5774	8.43692	8.8150	7.0002	7.5569

Exercise 17.6 Consider a closed-loop system with a fuzzy logic controller (FLC) in the loop shown in Figure 17.18. Suppose that we use five membership functions for the FLC inputs and its output. A possible set of rules is given in Table 17.3, where LN denotes large negative, N means negative, ZE—zero, P—positive, and LP refers to large positive. Implement a variant of the simple EA from Section 17.4 that generates the set of rules to be used to synthesize the FLC. Assign to each fuzzy set a positive integer, for example, let LN be encoded as 1, N as 2, ZE as 3, P as 4, and LP as 5. Then, a candidate solution, that is shown in Table 17.3, can be represented in a vector form as

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 1 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 5 & 2 & 3 & 4 & 5 & 5 & 3 & 4 & 4 & 5 & 5 \end{bmatrix},$$

that is, a vector representation of the candidate solution given by the rules matrix is obtained using the stacking operation of the columns of the rules matrix. Use the one-point crossover operator similar to the one in the canonical GA. The mutation operation can be implemented by altering a randomly chosen coordinate in the given chromosome, that is, the vector representing a candidate solution. You can use the roulette-wheel method to determine the replacement. For example, if the seventh component was selected to be altered, then you can

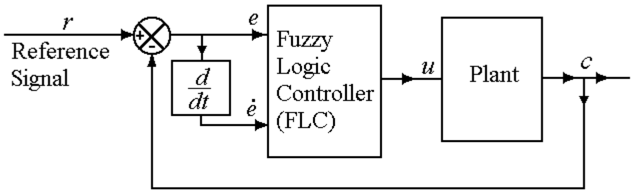


Figure 17.18: Closed-loop system with a fuzzy logic controller in the loop.

use the roulette-wheel to determine an integer from the set $\{1, 2, 3, 4, 5\}$ that will replace the "1" that occupies the seventh coordinate of the above chromosome. Chan, Lee, and Leung [54] used a similar version of the EA to generate fuzzy rules to synthesize a fuzzy logic target tracking algorithm.

Table 17.3: Rules matrix to be used to construct a fuzzy logic controller in Exercise 17.6.

$\dot{e}$ / e	LN	N	ZE	P	LP
LN	LN	LN	LN	N	ZE
N	LN	LN	N	ZE	P
ZE	N	N	ZE	P	P
P	N	ZE	P	LP	LP
LP	ZE	P	LP	LP	LP

Chapter 18

Chaotic Systems and Fractals

> An apparent paradox is that chaos is deterministic, generated by fixed rules which do not themselves involve any elements of change. We even speak of deterministic chaos. In principle, the future is completely determined by the past; but in practice small uncertainties, much like minute errors of measurement which enter into calculations, are amplified, with the effect that even though the behavior is predictable in the short term, it is unpredictable over the long run. [270, page 11]

18.1 Chaotic Systems Are Dynamical Systems With Wild Behavior

The objective of this chapter is to show the reader "the wild things that simple nonlinear equations can do" [232, page 459]. These simple nonlinear models with very complicated dynamics are examples of *chaotic systems*, which are deterministic nonlinear dynamical systems that exhibit a random-like behavior. A discrete-time model of population dynamics discussed in the next section, Newton's method for finding roots of certain polynomials, or a jetting balloon after its release are examples of dynamical chaotic systems. Trajectories of chaotic dynamical systems are sensitive to initial conditions in the sense that starting from slightly different initial conditions the trajectories diverge exponentially. Another property of a chaotic dynamical system is *loss of information* about initial conditions. We follow Moon [246, page 4] to explain this property. Suppose that we are given a state-plane portrait of a chaotic dynamical system in the (x, y)-plane. Assume that we specify the x-coordinate with accuracy Δx and the y-coordinate with accuracy Δy. Then, we could form a grid in

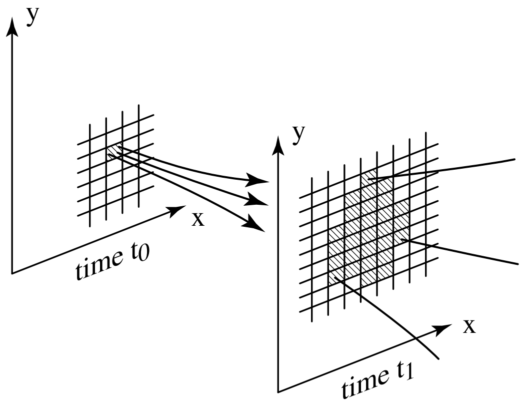


Figure 18.1: An illustration of the growth of uncertainty in a chaotic dynamical system.

the (x, y)-plane, shown in Figure 18.1, where each area would be of size $\Delta x\,\Delta y$. Thus, if we select initial conditions to the stated accuracy, then we know that these initial conditions are located somewhere in the shaded box in the (x, y)-plane corresponding to the initial time t_0. For a chaotic system this uncertainty grows in time to, say, N boxes at a later time t_1, as illustrated in Figure 18.1. The uncertainty growth, as mentioned above, is exponential and for continuous-time systems can be described by

$$N(t) \approx e^{\mu(t-t_0)} N(t_0), \tag{18.1}$$

where the constant μ is a measure of sensitivity with respect to initial conditions. Lyapunov exponents, that we discuss in a subsequent section, can be used to measure the initial condition sensitivity. Chaotic behavior can arise in low-order nonlinear dynamical systems. In the case of continuous-time systems, chaotic behavior can only occur for systems of dimension three or higher. However, in discrete-time systems, chaos can occur even in one-dimensional systems, as we show in the following section.

18.2 Chaotic Behavior of the Logistic Equation

18.2.1 The Logistic Equation—An Example From Ecology

The so called logistic equation, or logistic map, whose behavior we analyze in this section, arises, for example, in the analysis of the growth of a population.

Definition 18.1 *A map is a function whose domain (input) space and range (output) space are the same. Let $\boldsymbol{x}(0)$ be a point in $\mathbb{R}^n$ and $\boldsymbol{f}$ be a map on $\mathbb{R}^n$. The orbit, or trajectory, of*

$\boldsymbol{x}(0)$ *under* $\boldsymbol{f}$ *is the set of points*

$$\left\{\boldsymbol{x}(0), \boldsymbol{f}(\boldsymbol{x}(0)), \boldsymbol{f}^2(\boldsymbol{x}(0)), \ldots, \boldsymbol{f}^k(\boldsymbol{x}(0)), \ldots\right\},$$

where $\boldsymbol{f}^k$ *is the k-th iterate of* $\boldsymbol{f}$. *The starting point,* $\boldsymbol{x}(0)$, *of the orbit is called the initial value of the orbit.*

◇

We first analyze a situation where there are no predators and there is an unlimited supply of food. For the sake of argument, we consider a population of rabbits. Assume that on the average each rabbit gives birth to σ new rabbits in one unit of time, and that no rabbit dies. Let $N(k)$ be the number of rabbits at time $t = k$, then the change in the population over a time period, between k and $k+1$ is

$$N(k+1) - N(k) = \sigma N(k). \tag{18.2}$$

Thus,

$$N(k+1) = (1+\sigma)N(k). \tag{18.3}$$

Equation (18.3) is a very simple linear model of population growth. Given the number $N(0)$ of rabbits at time $k = 0$, the number of rabbits at time k is

$$N(k) = (1+\sigma)^k N(0). \tag{18.4}$$

One can see from (18.4) that the linear model given by (18.3) predicts the exponential grow of population. We now try to make the above model more realistic. We assume, as before, that the number of births is proportional to the size of the population. Let b be the birth rate, then the number of births at time k is $bN(k)$. We also assume that the number of deaths is proportional to the population size. Let d be the death rate. Then, the number of deaths at time k is $dN(k)$. The change in population in a time period between times k and $k+1$ is

$$N(k+1) - N(k) = bN(k) - dN(k). \tag{18.5}$$

We rewrite equation (18.5) as

$$N(k+1) = (1+b-d)N(k). \tag{18.6}$$

Let $\sigma = b - d$. Then, equation (18.6) has the same form as equation (18.3). In both cases, the population grows exponentially. We refer to the constant σ as the *unrestricted growth rate*. Over short periods of time, the above equation may be sufficient for purposes of an analysis of a population. Over long periods of time, the growth rates may change as the size of a population changes.

We alter the above linear models by introducing the concept of saturation. We first assume that the environment of the population can only support a certain number of species. We denote this number by K, and call it the *carrying capacity*. Thus, if $N(k) > K$, then there will not be enough food or space available, and more species will die than will be born. In such a situation the growth rate should be negative. If the population at time k is $N(k) < K$, then the growth rate should be positive. Finally, if $N(k)$ is much smaller than the carrying capacity, then the growth rate should be close to the unrestricted growth rate σ. A possible function that satisfies the above requirements has the form

$$\sigma\left(1-\frac{N(k)}{K}\right). \tag{18.7}$$

The change in the size of population then can be described as

$$N(k+1)-N(k)=\sigma\left(1-\frac{N(k)}{K}\right)N(k), \tag{18.8}$$

or equivalently

$$N(k+1)=(1+\sigma)N(k)-\frac{\sigma}{K}N^2(k). \tag{18.9}$$

Equation (18.9) is called the *logistic equation*. The term $\frac{\sigma}{K}N^2(k)$ is referred to as the *damping term*, because it dampens the growth of the population. To simplify (18.9), we scale the variable $N(k)$ by introducing the new variable

$$x(k)=\frac{\sigma}{\rho K}N(k),$$

where

$$\rho=1+\sigma.$$

Then, equation (18.9) becomes

$$\boxed{x(k+1)=\rho x(k)(1-x(k))} \tag{18.10}$$

We will show that if $0<\rho\le 4$ and $0\le x(0)\le 1$, then $0\le x(k)\le 1$ for $k=1,2,\ldots$. Let

$$f(x)=\rho x(1-x).$$

Then,

$$\frac{df(x)}{dx}=\rho-2\rho x=0$$

for $x=1/2$ and $f_{max}(x)=\rho/4$. Hence, for $k=1,2,\ldots,$

$$0\le x(k)\le 1 \quad \text{if} \quad 0<\rho\le 4 \quad \text{and} \quad 0\le x(0)\le 1.$$

18.2.2 Stability Analysis of the Logistic Map

We begin our investigation of the logistic equation (18.10) by seeking its equilibrium states (fixed points). These can be found by setting $x(k+1)$ equal to $x(k)$ in (18.10) to obtain

$$x(k)\left(\rho x(k) + (1-\rho)\right) = 0.$$

It follows that the equilibrium solutions are

$$x^* = 0 \quad \text{and} \quad x^* = \frac{\rho - 1}{\rho}.$$

We study attracting properties of the fixed points as a function of the parameter ρ. Roughly speaking, a fixed point $x^* = f(x^*)$ of the system $x(k+1) = f(x(k))$ is attracting if, for a starting point close to x^*, the next iterate is still closer. Formally, a fixed point x^* is attracting if there is an $\varepsilon > 0$ such that when $|x(0) - x^*| < \varepsilon$, then $\lim_{k\to\infty} x(k) = x^*$.

A fixed point x^* is repelling if for any $\varepsilon > 0$ no matter how small, if $0 < |x(0) - x^*| < \varepsilon$, then $|x(k) - x^*| > \varepsilon$ for some k.

Theorem 18.1 *Suppose $x^* = f(x^*)$ is a fixed point of $x(k+1) = f(x(k))$, where f is continuously differentiable. Then, x^* is attracting if*

$$\left|f'(x^*)\right| < 1,$$

and is repelling if

$$\left|f'(x^*)\right| > 1.$$

The case when $\left|f'(x^*)\right| = 1$ is inconclusive.

Proof We first show that if $x^* = f(x^*)$ is such that $\left|f'(x^*)\right| < 1$, then x^* is attracting. Since by assumption $\left|f'(x^*)\right| < 1$ and f' is continuous, there exists an interval $(x^* - \varepsilon, x^* + \varepsilon)$ about x^* such that for some $L > 0$,

$$\left|f'(x^*)\right| < L < 1 \quad \text{and} \quad \left|f'(x)\right| < L \quad \text{for every} \quad x \in (x^* - \varepsilon, x^* + \varepsilon).$$

Let $x(k) \in (x^* - \varepsilon, x^* + \varepsilon)$. We will show that $x(k+1)$ is closer to x^* than $x(k)$. We have

$$|x(k+1) - x^*| = |f(x(k)) - f(x^*)|$$

because $x^* = f(x^*)$ is a fixed point. Using the mean value theorem yields

$$|x(k+1) - x^*| = |f(x(k)) - f(x^*)| = \left|f'(\tilde{x})\right| |x(k) - x^*|$$

for some $\tilde{x} \in (x(k), x^*)$ or $\tilde{x} \in (x^*, x(k))$ if $x^* < x(k)$. Since $x(k) \in (x^* - \varepsilon, x^* + \varepsilon)$, then $\tilde{x} \in (x^* - \varepsilon, x^* + \varepsilon)$ also. Thus, $\left|f'(\tilde{x})\right| < L < 1$, and therefore

$$|x(k+1) - x^*| < L\,|x(k) - x^*|\,.$$

From the above, we conclude that if $x(0) \in (x^* - \varepsilon, x^* + \varepsilon)$, then

$$|x(k) - x^*| < L^k\,|x(0) - x^*|\,.$$

Because $L < 1$, we have $\lim_{k\to\infty} |x(k) - x^*| = 0$, that is,

$$\lim_{k\to\infty} x(k) = x^*,$$

which means that x^* is an attracting fixed point.

Similarly, if $x^* = f(x^*)$ is such that $\left|f'(x^*)\right| > 1$, then it is a repelling fixed point. Indeed, if $\left|f'(x^*)\right| > 1$, then because f' is continuous, there is an interval $(x^* - \varepsilon, x^* + \varepsilon)$ about x^* and a constant $L > 1$ such that $\left|f'(x)\right| > L$ for all $x \in (x^* - \varepsilon, x^* + \varepsilon)$. We will now show that if $x(k) \in (x^* - \varepsilon, x^* + \varepsilon)$, then $x(k+1)$ will be further away from x^* than $x(k)$. Using the mean value theorem, we obtain

$$\begin{aligned} |x(k+1) - x^*| &= |f(x(k)) - f(x^*)| \\ &= \left|f'(\tilde{x})\right| |x(k) - x^*| \\ &> L\,|x(k) - x^*|\,. \end{aligned} \tag{18.11}$$

If $x(k+1)$ is no longer in $(x^* - \varepsilon, x^* + \varepsilon)$, we are done. Otherwise, we repeat the process to get

$$\begin{aligned} |x(k+2) - x^*| &= |f(x(k+1)) - f(x^*)| \\ &> L\,|x(k+1) - x^*| \\ &> L^2\,|x(k) - x^*|\,. \end{aligned}$$

If $x(k+2)$ is no longer in the interval $(x^* - \varepsilon, x^* + \varepsilon)$, we are done. If not, we repeat the process to obtain

$$|x(k+m) - x^*| > L^m\,|x(k) - x^*|\,.$$

Since $(x^* - \varepsilon, x^* + \varepsilon)$ is a finite interval and $L^m \to \infty$ as $m \to \infty$, at some time, $x(k+m)$ must leave the interval, which means that x^* is a repelling fixed point. The proof of the theorem is complete.

□

Equation (18.11) tells us that the trajectory $\{x(k)\}$ diverges exponentially from x^* as $k \to \infty$. Thus, a trajectory with an initial condition close to x^* for the case when $\left|f'(x^*)\right| = L > 1$ will separate from x^* at a multiplicative rate of approximately $L > 1$ per iteration until it moves significantly away from x^*. In other words, the distance between $f(x(k))$ starting from $x(0)$ close to x^* will grow exponentially, that is, this distance will be magnified approximately by L for each iteration. To quantify this multiplicative rate of separation of points close to x^*, the Lyapunov exponent term is introduced that we discuss in Section 18.4.

We now consider the logistic equation (18.10). Recall that $f = \rho x(1-x)$ denotes the right hand side of the logistic equation. The derivative of f with respect to x is

$$f'(x) = \rho(1-2x).$$

Therefore, $f'(0) = \rho$. Since $\left|f'(0)\right| < 1$ when $\rho < 1$, by Theorem 18.1, the fixed point $x^* = 0$ is attracting for $\rho < 1$ and repelling for $\rho > 1$. The fixed point $x^* = \frac{\rho-1}{\rho}$ is outside of the range of x for $\rho < 1$. Thus, if $0 < \rho < 1$, we have just one attractive fixed point. For $\rho > 1$, the fixed point $x^* = 0$ is repelling and $x^* = \frac{\rho-1}{\rho}$ comes in within the range of the variable x. The derivative of f evaluated at $x^* = \frac{\rho-1}{\rho}$ is

$$f'\left(\frac{\rho-1}{\rho}\right) = 2-\rho.$$

By Theorem 18.1, the fixed point $x^* = \frac{\rho-1}{\rho}$ is attracting if $|2-\rho| < 1$, that is, if

$$1 < \rho < 3.$$

Thus, for $\rho = 1$ there is an exchange of attracting properties from one fixed point to the other. This change in the qualitative behavior at a certain parameter value is called a *bifurcation*. In Figure 18.2, the fixed points $x^* = 0$ and $x^* = \frac{\rho-1}{\rho}$ are shown and the ranges of ρ for which they are attracting. For $\rho > 3$, there are no attracting fixed points, and the solutions of the logistic equation become very complex. We discuss the logistic equation dynamics for $\rho > 3$ after displaying its solutions for $\rho = 0.7$, $\rho = 1.4$, $\rho = 2.8$, and $\rho = 3.6$ in Figure 18.3. The initial condition in all four cases was $x(0) = 0.9$. Another way of displaying a solution of a first-order difference equation is a *cobweb*. To construct a cobweb for the given first-order discrete system, we first draw the function $y = f(x)$ representing the right-hand side of the difference equation. Then, we draw the straight line $y = x$. The fixed points correspond to the points of intersection of these two graphs. The solution sequence, represented by the cobweb, staring at $x(0)$ is obtained as follows. Starting from $x(0)$, draw upward vertical line to intersect the graph of $f(x)$. This corresponds to the calculation of $x(1)$. The value of $x(1)$

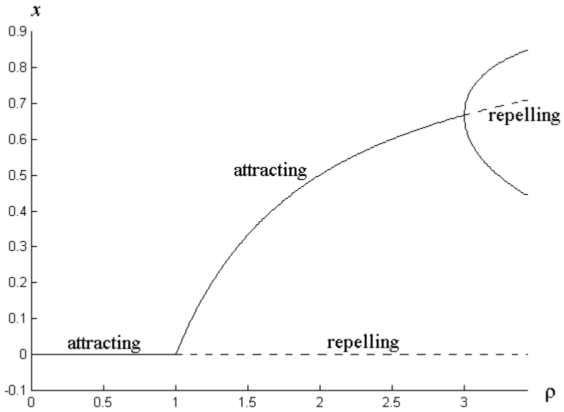


Figure 18.2: Exchange of attracting properties of the fixed points for $0 < \rho \leq 3.44$.

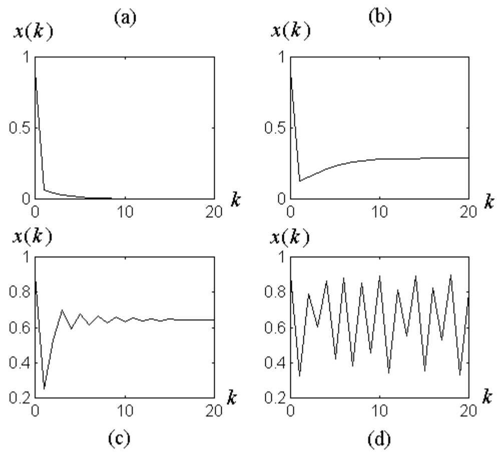


Figure 18.3: Solutions of the logistic equation: (a) $\rho = 0.7$, (b) $\rho = 1.4$, (c) $\rho = 2.8$, and (d) $\rho = 3.6$.

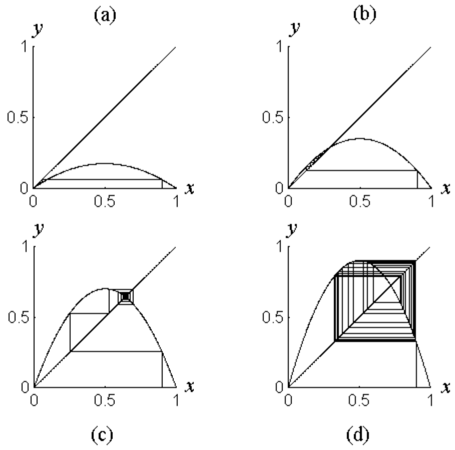


Figure 18.4: Cobwebs for the logistic equation: (a) $\rho = 0.7$, (b) $\rho = 1.4$, (c) $\rho = 2.8$, and (d) $\rho = 3.6$.

is transferred to the straight line $y = x$. The x-coordinate of the intersection is $x(1)$ which is then used to generate $x(2)$ using the above procedure. The process is repeated over and over again. In Figure 18.4, the cobwebs corresponding to the solutions displayed in Figure 18.3 are presented.

As we mentioned before, for $\rho > 3$ there are no attracting fixed points. As ρ increases, the solutions of the logistic equation exhibit increasing complexity. For ρ just slightly larger than 3, the solution sequence settles down into a steady oscillation of period 2 as shown in Figure 18.5. Then, as ρ is further increased, periodic solutions of period $4, 8, 16, \ldots$ appear. In Figure 18.6, a solution of period 4 is shown corresponding to $\rho = 3.5$. For $\rho = 3.9$, the solution is shown in Figure 18.7.

We next analyze the solutions for $\rho > 3$. To simplify the analysis, we first make the substitution

$$x(k) = z(k) + \frac{\rho - 1}{\rho}, \tag{18.12}$$

which is equivalent to shifting the fixed point $x^* = \frac{\rho-1}{\rho}$ to the origin in the (ρ, z)-coordinates. Note that

$$x(k+1) = z(k+1) + \frac{\rho - 1}{\rho}. \tag{18.13}$$

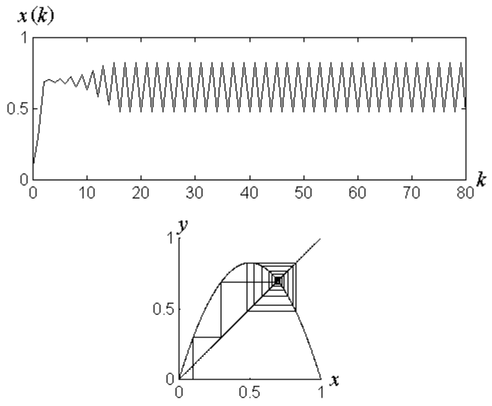


Figure 18.5: A two-cycle solution and its cobweb for $\rho = 3.3$ and $x(0) = 0.1$.

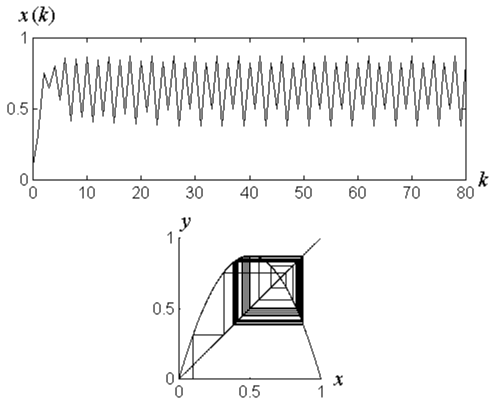


Figure 18.6: A period-four solution and its cobweb for $\rho = 3.5$ and $x(0) = 0.1$.

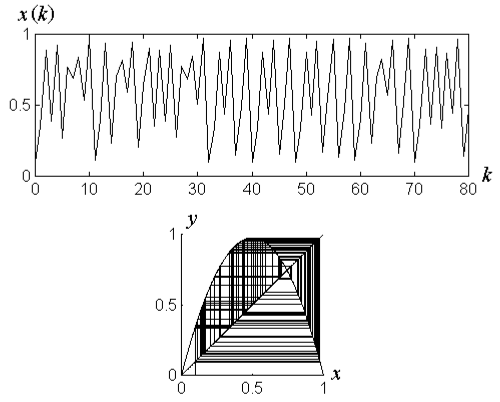


Figure 18.7: A periodic solution and its cobweb for $\rho = 3.9$ and $x(0) = 0.1$.

Substituting (18.12) and (18.13) into the logistic equation (18.10) yields

$$z(k+1) + \frac{\rho - 1}{\rho} = \rho \left(z(k) + \frac{\rho - 1}{\rho} \right) \left(1 - z(k) - \frac{\rho - 1}{\rho} \right). \tag{18.14}$$

After performing some manipulations, we represent (18.14) as

$$\boxed{z(k+1) = (2 - \rho) z(k) - \rho z^2(k)} \tag{18.15}$$

The fixed points of (18.15) are $z^* = 0$ corresponding to $x^* = \frac{\rho-1}{\rho}$, and $z^* = \frac{1-\rho}{\rho}$ corresponding to $x^* = 0$. We investigate the attracting properties of the fixed point $z^* = 0$ for $\rho \geq 0$. We see from Figure 18.5 that for ρ just greater than 3, the solution of the logistic equation goes to an attracting period-two cycle. Two numbers p and q form a period-2 cycle for a first-order discrete systems if, when $x(k) = p$, then $x(k+1) = q$, $x(k+2) = p$, and so on. A period-2 cycle is attracting if there are intervals (p_1, p_2) and (q_1, q_2) about p and q, respectively, such that

(i) if $x(0) \in (p_1, p_2)$, then $\lim_{k\to\infty} x(2k) = p$ and $\lim_{k\to\infty} x(2k+1) = q$;

(ii) if $x(0) \in (q_1, q_2)$, then $\lim_{k\to\infty} x(2k) = q$ and $\lim_{k\to\infty} x(2k+1) = p$.

The fixed points p and q of the period-2 cycle of equation (18.15) can be found by solving the equation

$$z = g(g(z)), \tag{18.16}$$

where $g = (2-\rho)z - \rho z^2$. We can use MATLAB's function `compose` to form (18.16), and then use the function `solve` to solve (18.16). We obtain four roots. Two of them are the fixed points that we analyzed above. The remaining two roots are the fixed points of the period-2 cycle. They are:

$$p = -\frac{1}{2\rho}\left(\rho - 3 + \sqrt{\rho^2 - 2\rho - 3}\right), \tag{18.17}$$

and

$$q = -\frac{1}{2\rho}\left(\rho - 3 - \sqrt{\rho^2 - 2\rho - 3}\right). \tag{18.18}$$

Once we have found the 2-cycle, we next determine the range of ρ for which it is attracting. A 2-cycle p and q of the system $z(k+1) = g(z(k))$ is attracting if the fixed points p and q of the map $y(k+1) = h(y(k))$ are attracting, where $h(y) = g(g(y))$. By Theorem 18.1, the fixed points $p = h(p)$ and $q = h(q)$ are attracting if $\left|h'(p)\right| < 1$ and $\left|h'(q)\right| < 1$. Using the chain rule, we obtain

$$h'(y) = \frac{d}{dy}g(g(y)) = g'(g(y))g'(y). \tag{18.19}$$

Since $g(p) = q$ and $g(q) = p$, we have

$$h'(p) = g'(g(p))g'(p) = g'(q)g'(p). \tag{18.20}$$

Similarly,

$$h'(q) = g'(p)g'(q). \tag{18.21}$$

Therefore, $\left|h'(p)\right| < 1$ and $\left|h'(q)\right| < 1$ when

$$\left|g'(p)g'(q)\right| < 1. \tag{18.22}$$

Using Theorem 18.1 and the arguments similar to the ones above, we conclude that the 2-cycle represented by p and q is repelling if

$$\left|g'(p)g'(q)\right| > 1. \tag{18.23}$$

We next substitute (18.17) and (18.18) into $g'(p)g'(q)$, and evaluate the resulting expression using MATLAB's function `subs`, to obtain

$$g'(p)g'(q) = (2 - \rho - 2\rho p)(2 - \rho - 2\rho q) = 4 - \rho^2 + 2\rho. \tag{18.24}$$

Thus, the 2-cycle represented by the pair (p, q) is attracting if

$$-1 < 4 - \rho^2 + 2\rho < 1, \tag{18.25}$$

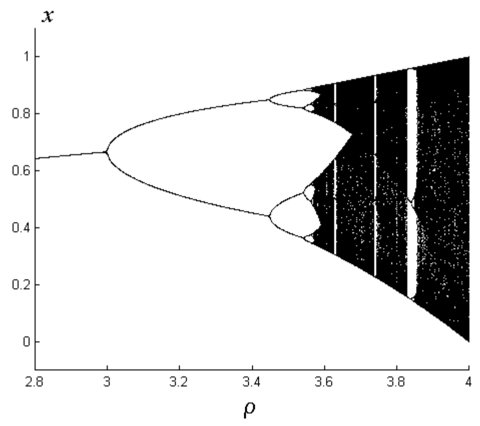


Figure 18.8: Bifurcation diagram of the logistic equation.

where $\rho > 3$. Solving (18.25) yields

$$3 < \rho < 1 + \sqrt{6}. \tag{18.26}$$

When $\rho > 1 + \sqrt{6}$, an attracting 4-cycle appears. As can be seen from the bifurcation diagram shown in Figure 18.8, that as the value of ρ increases the period doubling process takes place. Each of the branches of the bifurcation diagram produces a period-2 cycle of its own. The process continues as the parameter ρ approaches the value $\rho = 3.5699$, until no periodic cycles are created. For $\rho = 3.5699$ the behavior is not cyclic anymore. The behavior of the solutions of the logistic equation for $\rho \geq 3.5699$ is complex indeed, as can be seen, for example, by inspecting the bifurcation diagram shown in Figure 18.8. Various windows of ranges of $\rho > 3.5699$ correspond to different periodicities of solutions as well as to aperiodic solutions. These aperiodic solutions are called *chaotic solutions* or *chaotic motions*. Here, we quote a descriptive definition of chaotic motion due to Thompson and Stewart [327, page 197]. "... one way to define chaos is in the negative—recurrent behavior that is not an equilibrium, a cycle or even a quasi-periodic motion—there is more to be said about chaos. Chaotic motion has some aspect that is provably as random as a coin toss. The randomness arises from sensitive dependence on imperfectly known initial conditions, resulting for example in broadband noise in the power spectra of observable time series. This seems remarkable because the dynamical system needs no stochastic input to achieve this". A precise definition of chaotic motion is given in Section 18.4.

18.2.3 Period Doubling to Chaos

We describe one of the ways in which chaotic motion can arise—by repeated flip bifurcations, that is, through period doubling. Such a behavior arises for $3 \leq \rho \leq 3.5699$. We show that there is a sequence of numbers

$$c_1 < c_2 < \cdots < c_n < \cdots \tag{18.27}$$

such that when

$$c_n < \rho < c_{n+1},$$

then the logistic equation has an attracting 2^n-cycle. Thus, for example, if $c_8 < \rho < c_9$, then the solution will settle into an attracting 2^8-cycle. We already have $c_1 = 3$ and $c_2 = 1+\sqrt{6} = 3.4495$. We are interested in finding the remaining elements of the sequence (18.27). The sequence (18.27) is convergent, where

$$\lim_{n\to\infty} c_n = c_\infty = 3.5699.$$

To find the elements of the sequence (18.27), we could, in principle, form composite maps $f^{2^n}(x)$ and use stability results from the previous subsection. We have observed that if the parameter ρ is slightly bigger than $1 + \sqrt{6}$, then the 2-cycle is repelling and the solution goes to an attracting 4-cycle. A cobweb of an attracting 4-cycle is shown in Figure 18.9. We could find the four points corresponding to an attracting 4-cycle by solving the algebraic equation

$$x = f(f(f(f(x)))), \tag{18.28}$$

where $f(x) = \rho x(1 - x)$. A graph of the function $y = f^4(x)$ is shown in Figure 18.10. The solution to (18.28) gives eight points. Four of these points, corresponding to $y = f^4(x)$ having negative slopes when intersecting the line $y = x$, form an attracting 4-cycle. Using results from the previous subsection, we could find the range of ρ resulting in attracting 4-cycle solutions. We could then proceed to find the range of ρ corresponding to solutions going to attracting 8-cycles, then 16-cycles, and so forth. However, this approach is only feasible for $n = 1$ and $n = 2$. We have to take a different approach to compute the elements of the sequence (18.27). We use an iteration technique consisting of four steps:

(1) Finding the second iterate, $z(k + 1) = g(g(z(k))$, of the logistic map (18.15).

(2) Shifting one of the points of the 2-cycle of the second iterate to the origin.

(3) Neglecting cubic and quartic terms in the shifted map.

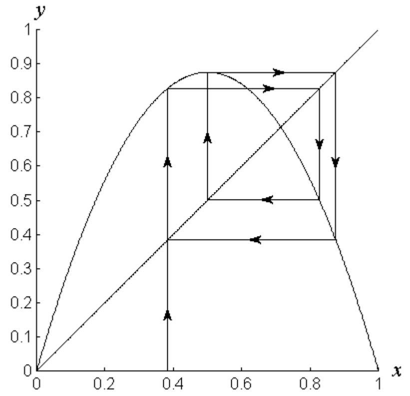


Figure 18.9: Cobweb of an attracting 4-cycle for $\rho = 3.5$.

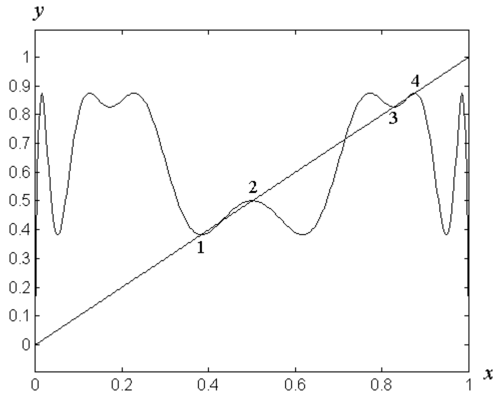


Figure 18.10: A graphical solution of $x = f^4(x)$, where $f(x) = 3.5x(1 - x)$. Marked points, corresponding to slopes less than one, form an attracting 4-cycle.

(4) Scaling the map obtained in step (3).

Performing iterations consisting of the above four steps will allow us to find the elements of the sequence (18.27).

The second iterate of (18.15) is

$$\begin{aligned} z(k+2) &= (2-\rho)z(k+1) - \rho z^2(k+1) \\ &= (2-\rho)\left((2-\rho)z(k) - \rho z^2(k)\right) - \rho\left((2-\rho)z(k) - \rho z^2(k)\right)^2 \\ &= (2-\rho)^2 z(k) - \left((2-\rho)\rho + \rho(2-\rho)^2\right) z^2(k) \\ &\quad +2\rho^2(2-\rho)z^3(k) - \rho^3 z^4(k). \end{aligned} \tag{18.29}$$

The points p and q, given by (18.17) and (18.18), are fixed points of the map (18.29). We select one of the points, p or q, to work with. Suppose that we choose the point p. Consider the transformation

$$v(k) = z(k) - p. \tag{18.30}$$

Note that

$$v(k+2) = z(k+2) - p. \tag{18.31}$$

Substituting (18.30) and (18.31) into (18.29), performing manipulations, and neglecting terms of order three and four, we obtain

$$v(k+2) = \left(4 + 2\rho - \rho^2\right) v(k) - \rho\left(\rho^2 - 2\rho - 3 - 3\sqrt{\rho^2 - 2\rho - 3}\right) v^2(k). \tag{18.32}$$

One can easily verify the above result using MATLAB's function `subs` applied to (18.29)–(18.31). We will be studying fixed points of (18.32). We replace $k+2$, the argument of v on the left-hand side of (18.32), with $k+1$. Let

$$C_1(p) = \rho^2 - 2\rho - 3 - 3\sqrt{\rho^2 - 2\rho - 3}. \tag{18.33}$$

Then, (18.32) can be represented as

$$v(k+1) = \left(4 + 2\rho - \rho^2\right) v(k) - \rho C_1(p) v^2(k). \tag{18.34}$$

Note that we if we used q in (18.30) instead of p, we would get

$$v(k+1) = \left(4 + 2\rho - \rho^2\right) v(k) - \rho C_1(q) v^2(k), \tag{18.35}$$

where now $C_1(q) = \rho^2 - 2\rho - 3 + 3\sqrt{\rho^2 - 2\rho - 3}$. In the next step, we transform the v-map given by (18.34) into the same format as the z-map given by (18.15). This allows us to apply stability properties of the z-map to the v-map. We introduce an iterated parameter ρ_1 such that

$$2-\rho_1 = 4+2\rho_0-\rho_0^2, \tag{18.36}$$

and represent (18.34) as

$$v(k+1) = (2-\rho_1)\,v(k) - \rho_0 C_1(p) v^2(k). \tag{18.37}$$

Finally, we scale (18.37) using the transformation

$$s(k) = \frac{\rho_0 C_1(p)}{\rho_1} v(k). \tag{18.38}$$

Substituting (18.38) into (18.37) gives

$$\frac{\rho_1}{\rho_0 C_1(p)} s(k+1) = \frac{\rho_1(2-\rho_1)}{\rho_0 C_1(p)} s(k) - \frac{\rho_1^2 \rho_0 C_1(p)}{\rho_0^2 C_1^2(p)} s^2(k). \tag{18.39}$$

Simplifying (18.39) yields

$$\boxed{s(k+1) = (2-\rho_1)\,s(k) - \rho_1 s^2(k)} \tag{18.40}$$

The above s-map has the same format as the z-map given by (18.15) and analyzed in the previous subsection. It follows from these results that the s-map given by (18.40) has an attracting 2-cycle for $3 < \rho_1 < 1+\sqrt{6}$, which corresponds to an attracting 4-cycle of the z-map given by (18.15). Using (18.36), we conclude that $3<\rho_1<1+\sqrt{6}$ corresponds to

$$1+\sqrt{6} < \rho_0 < 1 + \sqrt{4+\sqrt{6}} = 3.5396.$$

Indeed, representing (18.36) as

$$\rho_0^2 - 2\rho_0 - (2+\rho_1) = 0,$$

and using the quadratic formula gives

$$\rho_0 = 1+\sqrt{3+\rho_1},$$

where we took the solution that gives $\rho_0 > 0$. The above concludes the first iteration. The result of this iteration is the element $c_3 = 1+\sqrt{4+\sqrt{6}} = 3.5396$ of the sequence (18.27) and the scaling parameter $C_1(p)$ whose properties will be investigated in the following subsection.

We then proceed with the next iteration. In particular, executing the steps of the iteration method on the s-map yields the new map that we call the t-map. This map has the form

$$t(k+1) = \left(4+2\rho_1-\rho_1^2\right) t(k) - \rho_1 C_2 t^2(k). \tag{18.41}$$

Introducing the iterated parameter, ρ_2, such that

$$\rho_2 = \rho_1^2 - 2\rho_1 - 2, \tag{18.42}$$

and scaling the t-map using the transformation

$$u(k) = \frac{\rho_1 C_2}{\rho_2} t(k)$$

yields the u-map of the form

$$u(k+1) = (2 - \rho_2)\, u(k) - \rho_2 u^2(k). \tag{18.43}$$

The above map has an attracting 2-cycle for $3 < \rho_2 < 1 + \sqrt{6}$ which corresponds to an attracting 4-cycle of the s-map, which in turn corresponds to an attracting 8-cycle of the z-map. We calculate the range of the parameter ρ corresponding to attracting 8-cycles of the z-map by working the recursive relation (18.42) to get

$$c_4 = 1 + \sqrt{3 + \rho_1} = 1 + \sqrt{3 + 1 + \sqrt{3 + \rho_2}} = 1 + \sqrt{4 + \sqrt{4 + \sqrt{6}}} = 3.5573.$$

In general, after $i + 1$ iterations we obtain

$$\rho_{i+1} = \rho_i^2 - 2\rho_i - 2, \tag{18.44}$$

and work backwards to find the term c_{i+3} of the sequence (18.27). We can also compute the elements of the sequence (18.27) using the recursive relation

$$\rho_i = \rho_{i+1}^2 - 2\rho_{i+1} - 2, \quad i = 1, 2, \ldots. \tag{18.45}$$

Solving the above, using the quadratic formula, yields

$$\rho_{i+1} = 1 + \sqrt{3 + \rho_i}.$$

This means that the elements of the sequence (18.27) satisfy the difference equation

$$c_{i+1} = 1 + \sqrt{3 + c_i}, \quad c(1) = 3. \tag{18.46}$$

The equation (18.46) has two equilibrium points that are obtained by solving the algebraic equation

$$c = 1 + \sqrt{3 + c}.$$

The equilibrium point

$$c_{eq} = \frac{3+\sqrt{17}}{2}$$

is attracting, and we have

$$\lim_{i\to\infty} c_i = c_\infty = c_{eq} = 3.5616.$$

An analysis without truncation when computing second iterates of the logistic equation yields $c_\infty = 3.5699$. Such an analysis can be found in the paper by Feigenbaum [100]. After the parameter ρ reaches the limiting value $\rho = 3.5699$, no stable solution patterns of the logistic equation emerge. We obtain a chaotic behavior.

18.2.4 The Feigenbaum Numbers

The sequence (18.27) and the scaling factors

$$\frac{\rho_i C_{i+1}}{\rho_{i+1}} \tag{18.47}$$

have certain properties that are universal for all quadratic maps. Here, we perform an approximate analysis of these properties following Kahn [167, Chapter 16].

We begin by rewriting equation (18.45) as

$$\left(\rho_{i+1} - 1\right)^2 = 3 + \rho_i. \tag{18.48}$$

Hence,

$$\rho_{i+1} - 1 = \sqrt{3+\rho_i}, \tag{18.49}$$

and

$$\rho_i - 1 = \sqrt{3+\rho_{i-1}}. \tag{18.50}$$

Subtracting (18.50) from (18.49) gives

$$\rho_{i+1} - \rho_i = \sqrt{3+\rho_i} - \sqrt{3+\rho_{i-1}}. \tag{18.51}$$

Summing (18.50) and (18.49) yields

$$\rho_{i+1} + \rho_i - 2 = \sqrt{3+\rho_i} + \sqrt{3+\rho_{i-1}}. \tag{18.52}$$

Multiplying (18.51) and (18.52), we get

$$\left(\rho_{i+1} - \rho_i\right)\left(\rho_{i+1} + \rho_i - 2\right) = \rho_i - \rho_{i-1}. \tag{18.53}$$

We rewrite (18.53) as

$$\rho_{i+1} + \rho_i - 2 = \frac{\rho_i - \rho_{i-1}}{\rho_{i+1} - \rho_i}. \tag{18.54}$$

Let

$$\boxed{\delta_i = \frac{\rho_i - \rho_{i-1}}{\rho_{i+1} - \rho_i}} \tag{18.55}$$

The limiting value of δ_i as i tends to infinity is called the *Feigenbaum number* δ. To find its value, we use (18.55) to represent equation (18.54) as

$$\delta_i = \rho_{i+1} + \rho_i - 2. \tag{18.56}$$

We obtain an approximate value for the Feigenbaum number δ because of the truncation procedure that we used to obtain (18.47). We have

$$\delta_{approx} = \lim_{i\to\infty} (\rho_{i+1} + \rho_i - 2) = 2c_\infty - 2 = 5.1231 \tag{18.57}$$

The exact value of the Feigenbaum number δ is $\delta_{exact} = 4.669202\ldots$.

The other parameter that characterizes our iteration method is the scaling factor given by (18.47). The limiting value of (18.47) as i tends to infinity is called the *Feigenbaum number* α. Its approximate value is

$$\begin{aligned}
\alpha_{approx} &= \lim_{i\to\infty} \frac{\rho_i C_{i+1}(p)}{\rho_{i+1}} \\
&= \frac{c_\infty C_\infty(p)}{c_\infty} \\
&= c_\infty^2 - 2c_\infty - 3 - 3\sqrt{c_\infty^2 - 2c_\infty - 3} \\
&= -2.24
\end{aligned} \tag{18.58}$$

The exact value of the Feigenbaum number α is $\alpha_{exact} = -2.50\ldots$.

18.3 Fractals

This section is devoted to modeling of irregular shapes. These non-Euclidean objects are called *fractals*. The term *fractal* was coined by Mandelbrot [229] and it comes from the Latin adjective *fractus*, meaning “irregular”. The corresponding Latin verb is *frangere*, which means “to break to create irregular fragments.” An example of a simple fractal is the so called the von Koch snowflake curve that is shown in Figure 18.11. In this Figure, an iterative procedure for constructing the curve is also shown. We start with dividing a line

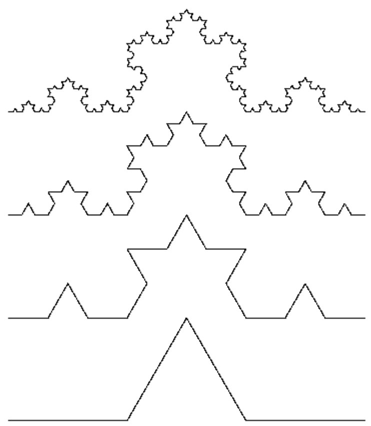


Figure 18.11: The first four stages in constructing the von Koch snowflake curve.

segment into thirds and then replacing the middle segment with two equal segments that form two sides of an equilateral triangle. The outcome of this stage is a shape consisting of four segments. In the next stage, each of the four segments is divided into thirds and the middle segments are replaced with two equal segments that form two sides of equilateral triangle. The process is repeated over and over again. The resulting curve is characterized by *self-similarity* or *scaling*.

One of the characteristics of a fractal is its dimension. A line segment is a one-dimensional Euclidean object and we can divide this line segment into N identical parts. In such a case, the individual parts can be viewed as scaled down, by the factor $r = 1/N$, versions of the original object. Similarly a square, a two-dimensional Euclidean object, can be divided into N self-similar parts. These parts can be viewed as scaled down copies by the factor $r = 1/\sqrt{N}$ of the original square. In general, a D-dimensional object can be divided into N copies of itself. Each copy can be viewed as a scaled down version by the factor

$$r = \frac{1}{N^{1/D}} \tag{18.59}$$

of the parent object. Equation (18.59) can be represented as

$$r^D N = 1. \tag{18.60}$$

We now generalize the above arguments to the objects that are not necessarily Euclidean. Taking logarithms of both sides of (18.60) yields the expression for the *fractal dimension* of

Figure 18.12: The first four iterations in constructing Cantor's ternary set.

the form

$$\boxed{D = \frac{\log N}{\log(1/r)}} \tag{18.61}$$

We use (18.61) to compute the dimension of the von Koch curve. Any segment of the curve is composed of four sub-segments, that is, $N = 4$. Each sub-segment is a scaled down copy of its parent by the factor $r = 1/3$. Hence, the dimension of the von Koch curve is

$$D = \frac{\log N}{\log(1/r)} = \frac{\ln 4}{\ln 3} = 1.26 \tag{18.62}$$

We can see from the above that the fractal dimension, unlike the dimension of a common Euclidean pattern, need not be an integer.

Another example of a fractal is the *Cantor set*, sometimes called *Cantor's ternary set*. The Cantor set consists of the points of the unit interval $[0, 1]$ that remain after "middle third" intervals have been successively removed. This set is illustrated in Figure 18.12. It is an example of so called *disconnected fractal* or *fractal dust*. For Cantor's set, we have $N = 2$ and $1/r = 3$. Hence, the fractal dimension of the Cantor set is

$$D = \frac{\log N}{\log(1/r)} = \frac{\ln 2}{\ln 3} = 0.631 \tag{18.63}$$

18.3.1 The Mandelbrot Set

The Mandelbrot set, denoted $\mathcal{M}$, is the set of all complex $c \in \mathbb{C}$ for which the sequence

$$0, c, c^2+c, (c^2+c)^2+c, \ldots \tag{18.64}$$

does not tend to ∞ as n tends to ∞. Thus, the Mandelbrot set is generated by the map in the complex plane,

$$z \mapsto z^2 + c, \quad c \in \mathbb{C}, \tag{18.65}$$

or equivalently,

$$z(k+1) = z^2(k) + c, \quad c \in \mathbb{C}. \tag{18.66}$$

Let

$$z(k) = x(k) + jy(k) \quad \text{and} \quad c = a + jb.$$

Then,

$$\begin{aligned} z(k+1) &= x(k+1) + jy(k+1) \\ &= x^2(k) - y^2(k) + a + j\left(2x(k)y(k) + b\right) \\ &= x^2(k) + 2jx(k)y(k) + (jy(k))^2 + a + jb \\ &= z^2(k) + c. \end{aligned}$$

Thus, we can express the first-order quadratic complex dynamical system (18.66) as a second-order real dynamical system of the form

$$\left.\begin{aligned} x(k+1) &= x^2(k) - y^2(k) + a \\ y(k+1) &= 2x(k)y(k) + b \end{aligned}\right\} \tag{18.67}$$

The representations (18.66) and (18.67) are indeed equivalent.

In terms of the dynamical system (18.67), the Mandelbrot set is the set of parameter values, a and b, for which the solutions of (18.67) starting at $x(0) = y(0) = 0$ remain bounded. The Mandelbrot set is shown in Figure 18.13.

18.3.2 Julia Sets

A filled Julia set corresponding to the given values of the parameters a and b of the dynamical system (18.67) is the set of initial conditions $(x(0), y(0))$ resulting in bounded system trajectories. For each value of $c = a + jb$ there is a different Julia set. In order to construct a Julia set, pick first a point (a, b), say in the Mandelbrot set. Then, pick an initial

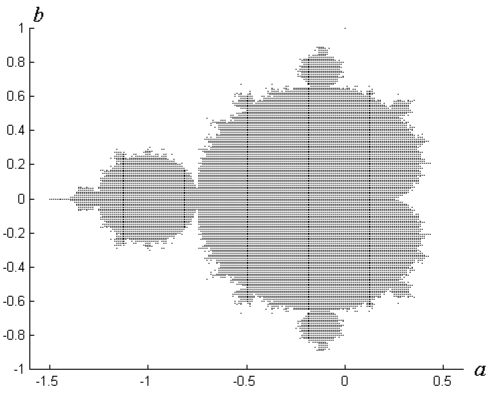


Figure 18.13: The Mandelbrot set for $x(0) = y(0) = 0$.

condition $(x(0), y(0))$ and compute the resulting trajectory of (18.67) starting from the chosen initial condition $(x(0), y(0))$. Plot only the initial conditions $(x(0), y(0))$ that give rise to bounded trajectories. The set of plotted points is the filled Julia set corresponding to the particular parameter values (a, b). The boundary of the filled Julia set is the Julia set. Two examples of filled Julia sets are shown in Figures 18.14 and 18.15. The filled Julia set shown in Figure 18.14 is called by Keen ([82, page 61]), "Douady's Rabbit". It is observed by Peitgen in [271, page 199] that in a sense the Mandelbrot set can be considered as a one-page-dictionary of all Julia sets. This means that the Mandelbrot set stores infinitely many different Julia sets.

18.3.3 Iterated Function Systems

An iterated function system (IFS) can be used to model natural objects such as clouds, smoke, leaves, or ferns. An IFS uses affine transformations that we discuss next. A two-dimensional affine transformation $\boldsymbol{w} : \mathbb{R}^2 \to \mathbb{R}^2$ from $\mathbb{R}^2$ into itself is defined by

$$\boldsymbol{w}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \tag{18.68}$$

where $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{b} \in \mathbb{R}^2$. The affine transformation (18.68) satisfies

$$\|\boldsymbol{w}(\boldsymbol{x}) - \boldsymbol{w}(\boldsymbol{y})\| \le L\|\boldsymbol{x} - \boldsymbol{y}\|, \tag{18.69}$$

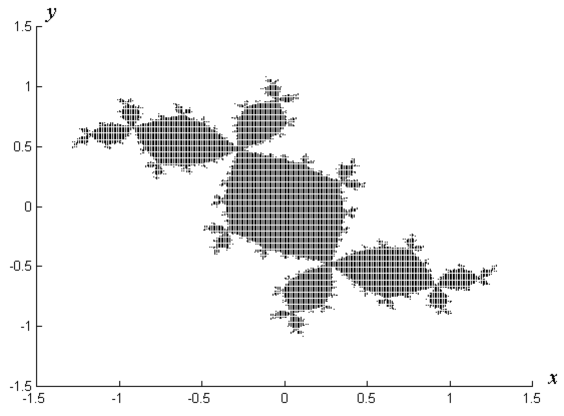


Figure 18.14: Filled Julia set for $c = -0.12256117 + j0.74486177$.

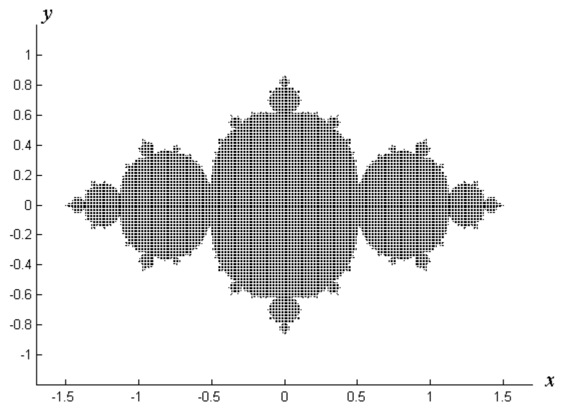


Figure 18.15: Filled Julia set for $c = -0.75$.

where the Lipschitz constant L is

$$L = \|\boldsymbol{A}\|.$$

The affine transformation $\boldsymbol{w}$ is contractive if $L < 1$. It is a symmetry if

$$\|\boldsymbol{w}(\boldsymbol{x}) - \boldsymbol{w}(\boldsymbol{y})\| = \|\boldsymbol{x} - \boldsymbol{y}\|.$$

The affine transformation $\boldsymbol{w}$ is expansive if $L > 1$.

We now define an IFS. A two-dimensional IFS consists of a set of N affine transformations denoted

$$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N\},$$

each transformation mapping $\mathbb{R}^2$ into itself, together with a set of probabilities

$$\{p_1, p_2, \ldots, p_N\},$$

where each $p_i \geq 0$ and

$$p_1 + p_2 + \cdots + p_N = 1.$$

An IFS is the set

$$\{\boldsymbol{w}_n, p_n : n = 1, 2, \ldots, N\}.$$

Let L_n, $n = 1, 2, \ldots, N$, denote the Lipschitz constant for $\boldsymbol{w}_n$. Then, we say that the IFS obeys the average contractivity condition if

$$L_1^{p_1} L_2^{p_2} \cdots L_N^{p_N} < 1. \tag{18.70}$$

Interesting mathematical properties of the IFS that satisfy the condition (18.70) are reported by Barnsley [24].

An IFS operates as follows. A set of N affine transformations is chosen and an initial point is picked. Then, the affine transformations are iteratively applied over and over again in a random order with probabilities p_n, for example, proportional to the determinants of $\boldsymbol{A}_n$. The result is a sequence of points from $\mathbb{R}^2$,

$$\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots$$

As an example, suppose that $N = 4$, where

$$\begin{aligned}
\boldsymbol{A}_1 &= \begin{bmatrix} 0.02 & 0 \\ 0 & 0.21 \end{bmatrix} & \boldsymbol{b}_1 &= \begin{bmatrix} -0.15 \\ -0.8 \end{bmatrix} \\
\boldsymbol{A}_2 &= \begin{bmatrix} 0.8496 & 0.0255 \\ -0.0255 & 0.8496 \end{bmatrix} & \boldsymbol{b}_2 &= \begin{bmatrix} 0 \\ 0.15 \end{bmatrix} \\
\boldsymbol{A}_3 &= \begin{bmatrix} -0.1554 & 0.235 \\ 0.1958 & 0.1865 \end{bmatrix} & \boldsymbol{b}_3 &= \begin{bmatrix} 0.05 \\ -0.55 \end{bmatrix} \\
\boldsymbol{A}_4 &= \begin{bmatrix} 0.1554 & -0.235 \\ 0.1958 & 0.1865 \end{bmatrix} & \boldsymbol{b}_4 &= \begin{bmatrix} -0.35 \\ -0.4 \end{bmatrix}.
\end{aligned}$$

The above data comes from Bunde and Havlin [47]. The elements of the set of probabilities

$$\{p_1, p_2, p_3, p_4\}$$

were constructed as

$$p_n = \frac{|\det(\boldsymbol{A}_n)|}{\sum_{i=1}^{N} |\det(\boldsymbol{A}_i)|} \qquad n = 1, 2, 3, 4.$$

The initial point was chosen to be $\boldsymbol{x}(0) = \begin{bmatrix} -0.1 & -0.2 \end{bmatrix}^\top$. Then, the roulette wheel method was used to select the affine transformation to operate on $\boldsymbol{x}(0)$. The roulette wheel method operates as follows. Each affine transformation is assigned a slice of a wheel, where the slice is proportional to the probability p_n of the particular transformation $\boldsymbol{w}_n$. A random number from the interval $[0, 1]$ is generated, which corresponds to spinning the wheel. The transformation under the wheel's marker is selected. Suppose, $\boldsymbol{w}_k$ was selected. Then $\boldsymbol{w}_k$ operates on $\boldsymbol{x}(0)$ and the result is

$$\boldsymbol{x}(1) = \boldsymbol{A}_k \boldsymbol{x}(0) + \boldsymbol{b}_k.$$

The roulette wheel is spun again to select the affine transformation to act upon $\boldsymbol{x}(1)$ in order to obtain $\boldsymbol{x}(2)$, etc.

The result of $30,000$ iterations is shown in Figure 18.16. The object is called the Barnsley fern because it was M. Barnsley [24] who introduced the method.

18.4 Lyapunov Exponents

Trajectories of a chaotic dynamical system are characterized by a sensitive dependence upon initial conditions, which means that the distance between the adjacent trajectories grows

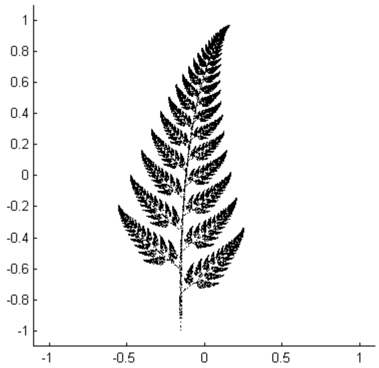


Figure 18.16: The Barnsley fern.

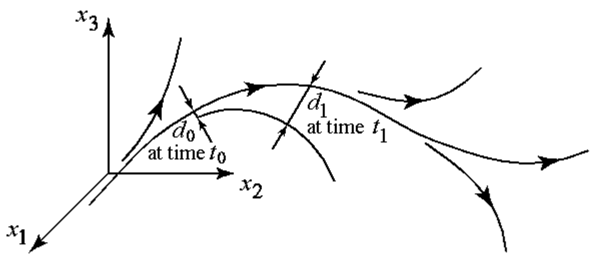


Figure 18.17: The change in distance between adjacent chaotic trajectories that is used to compute the largest Lyapunov exponent.

exponentially with time as illustrated in Figure 18.17. For a discrete dynamical system, if x^* is a fixed point of a one-dimensional map f and if $|f'(x^*)| = a > 1$, then by Theorem 18.1, trajectories starting near x^* will separate from x^* at a multiplicative rate of approximately a per iteration until they move significantly far away from x^*. Let x_1 be a point that constitutes the k-cycle. Then, for a k-cycle, the trajectory starting from a point close to x_1, will separate from x_1 at a rate approximately equal to the product of the derivatives of the k points of the k-cycle. Let this product be denoted A and suppose that $A > 1$. Then, it takes k iterations to separate a trajectory starting close to x_1 by a distance A from x_1. We can say that in the case of a k-cycle the average rate of separation is $A^{1/k}$ per iteration. We now are ready to introduce the term *Lyapunov number* to describe the average multiplicative rate of separation of points close to x_1. For a k-cycle, the Lyapunov number $L(x_1)$ is

$$\begin{aligned} L(x_1) &= \left(\left| \left(f^k\right)'(x_1) \right| \right)^{1/k} \\ &= \left(|f'(x_1)|\,|f'(x_2)| \cdots |f'(x_k)|\right)^{1/k} \end{aligned} \tag{18.71}$$

The concept of Lyapunov number is now extended to non-periodic trajectories.

Definition 18.2 *Let f be a differentiable map on the real line $\mathbb{R}$ and let $\{x(0), x(1), \ldots\}$ be a trajectory starting from $x(0)$. Then, the Lyapunov number $L(x(0))$ of the trajectory starting from $x(0)$ is*

$$L(x(0)) = \lim_{N\to\infty} \left(|f'(x(0))| \cdots |f'(x(N))|\right)^{1/N}$$

if the limit exists.

◇

The Lyapunov exponent, $\mu(x(0))$, is then defined as the natural logarithm of the Lyapunov number, that is,

$$\mu(x(0)) = \ln L(x(0)) = \lim_{N\to\infty} \frac{\ln|f'(x(0))| + \cdots + \ln|f'(x(N))|}{N} \tag{18.72}$$

if the limit exists.

We now look at the Lyapunov exponent from another perspective. Let d_0 denote the distance between the two initial states of a continuous dynamical system. Then, for a chaotic motion, at a later time,

$$d(t) = e^{\mu t} d_0. \tag{18.73}$$

The measure of the trajectories' divergence is obtained by averaging the exponential growth at many points along a trajectory. This is because the chaotic trajectories' divergence can only be measured locally because if the trajectories are bounded, $d(t)$ in (18.73) cannot go to infinity. To define the measure of the divergence of chaotic trajectories, first a reference trajectory is obtained. Next a point on an adjacent trajectory is selected and $d(t)/d_0$ is measured. This is illustrated in Figure 18.17. Then, a new $d_0(t)$ is selected on new adjacent trajectory. Using the collected data, the Lyapunov exponent is computed as

$$\mu(x(0)) = \lim_{N\to\infty} \frac{1}{t_N - t_0} \sum_{k=1}^{N} \ln \frac{d(t_k)}{d_0(t_{k-1})}. \tag{18.74}$$

There are n Lyapunov exponents for an n-dimensional nonlinear dynamical system, just as there are n eigenvalues that govern behavior of an n-dimensional linear system. To define these Lyapunov exponents for an n-dimensional nonlinear dynamical system model, first choose an n-dimensional sphere centered at a point on a reference trajectory. Let the radius of this sphere be δ_0. We can view this sphere as an n-dimensional ellipsoid with n semi-axes of equal length $\delta_i(0) = \delta_0$. Next, at a later time t construct an n-dimensional ellipsoid with the property that all the trajectories emanating from the previously chosen sphere, pass through this ellipsoid. Let the n semi-axes of the ellipsoid be denoted $\delta_i(t)$. Then, we can compute the Lyapunov exponents as

$$\boxed{\mu_i(\boldsymbol{x}(0)) = \lim_{t\to\infty}\left(\lim_{\delta_i(0)\to 0}\left\{\frac{1}{t}\ln\frac{\delta_i(t)}{\delta_i(0)}\right\}\right), \quad i = 1, 2, \ldots, n} \tag{18.75}$$

if the limits exist. It is common to order the Lyapunov exponents so that

$$\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n.$$

Thus, the Lyapunov exponents are numbers that are used to measure the exponential attraction or separation in time of two neighboring trajectories of a dynamical system.

Example 18.1 This example was adapted from Moon [246, page 193]. Suppose that we are given a one-dimensional map

$$x(k+1) = f(x(k)). \tag{18.76}$$

We demonstrate again that in the regions where f is differentiable, the Lyapunov exponent is determined by $|f'|$. Indeed, let $x(k)$ and $x(k)+\varepsilon$ be two initial conditions. Then, we have

$$d_0(k-1) = \varepsilon \quad \text{and} \quad d(k) = |f(x(k)+\varepsilon) - f(x(k))| \approx \left| f'|_{x=x(k)} \right| \varepsilon.$$

Using (18.74), we obtain

$$
\begin{aligned}
\mu(x(0)) &= \lim_{N\to\infty} \frac{1}{t_N - t_0} \sum_{k=1}^{N} \ln \frac{d(t_k)}{d_0(t_{k-1})} \\
&= \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \ln |f'(x(k))|,
\end{aligned}
$$

which is the same as (18.72)

We use the notion of Lyapunov exponent to formally define chaotic motion. The other term that we use in this definition is an asymptotically periodic trajectory that we define next. This definition is adapted from Alligood et al. [7, page 108].

Definition 18.3 *A trajectory* $\{\boldsymbol{x}(0), \ldots, \boldsymbol{x}(N), \ldots\}$ *of a smooth map* $\boldsymbol{f}$ *is called* asymptotically periodic *if it converges to a periodic trajectory as* $N \to \infty$*; that is, there exists a periodic trajectory* $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots\}$ *such that*

$$
\lim_{N\to\infty} \|\boldsymbol{x}(N) - \boldsymbol{y}(N)\| = 0.
$$

◇

Using the definition of Lyapunov exponent and the above definition, we can define a chaotic trajectory. This definition is adapted from Alligood et al. [7, page 110].

Definition 18.4 *Let* f *be a map of the real line* $\mathbb{R}$*, and let* $\{x(0), x(1), \ldots\}$ *be a bounded trajectory of the map* f*. This trajectory is chaotic if*

1. *the trajectory is not asymptotically periodic,*

2. *the Lyapunov exponent* $\mu(x(0))$ *is greater then zero.*

◇

For a map on $\mathbb{R}^n$, each trajectory has n Lyapunov numbers. The Lyapunov numbers, in this case, measure the rates of separation from the current point of the trajectory along n orthogonal directions. The first direction is the one where the separation between the nearby points is the greatest. The second direction is the one in which the separation is the greatest from all directions perpendicular to the first direction. The third direction will be the one where the separation is the greatest among directions perpendicular to the first two

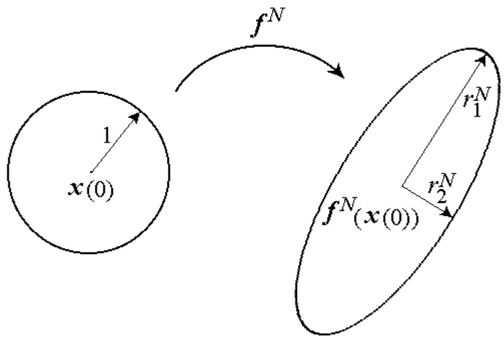


Figure 18.18: The unit disk is mapped into an ellipse by a two-dimensional map.

directions, and so on. We illustrate the above in two dimensions in Figure 18.18, where a unit disk is centered at $\boldsymbol{x}(0)$. After N iterates of the two-dimensional map $\boldsymbol{f}$, the disk evolves into an ellipse. The per-iteration changes of the ellipse axes are the Lyapunov numbers of $\boldsymbol{x}(0)$. Thus, the Lyapunov numbers measure the contraction or expansion near the trajectory of $\boldsymbol{x}(0)$ during the first N iterations. The natural logarithm of each Lyapunov number is a Lyapunov exponent.

Let $D\boldsymbol{f}(\boldsymbol{x}(0))$ be the first derivative of $\boldsymbol{f}$ evaluated at $\boldsymbol{x}(0)$. Let $J_N = D\boldsymbol{f}^N(\boldsymbol{x}(0))$ be the first derivative of the N-th iterate of $\boldsymbol{f}$. We denote by B the unit sphere centered at $\boldsymbol{x}(0)$. With the above notation, we are ready to give the definition of Lyapunov exponents for a map on $\mathbb{R}^n$. This definition is adapted from Alligood et al. [7, page 195].

Definition 18.5 *Let $\boldsymbol{f}$ be a smooth map on $\mathbb{R}^n$ and let r_k^N be the length of the k-th longest orthogonal axis of the ellipsoid $J_N B$ for a trajectory with the initial state $\boldsymbol{x}(0)$. The k-th Lyapunov number is*

$$L_k(\boldsymbol{x}(0)) = \lim_{N\to\infty} \left(r_k^N\right)^{1/N}$$

if the limit exists. The k-th Lyapunov exponent of $\boldsymbol{x}(0)$ is

$$\mu_k(\boldsymbol{x}(0)) = \ln L_k(\boldsymbol{x}(0)).$$

◇

Using the above definition, we can extend the definition of chaotic trajectories for maps on $\mathbb{R}^n$. This definition is adapted from Alligood et al. [7, page 196]

Definition 18.6 *Let $\boldsymbol{f}$ be a map on $\mathbb{R}^n$, $n \geq 1$, and let $\{\boldsymbol{x}(0), \boldsymbol{x}(1), \boldsymbol{x}(2), \ldots\}$ be a bounded trajectory of the map $\boldsymbol{f}$. The trajectory is chaotic if*

1. *it is not asymptotically periodic,*
2. *no Lyapunov exponent is equal to zero,*
3. $\mu_1(\boldsymbol{x}(0)) > 0$.

◇

For continuous systems, the above definition of a chaotic trajectory has to be modified—see Alligood at al. [7, pages 385–386].

To numerically compute Lyapunov exponents, a reference trajectory of a given nonlinear system is constructed and the ellipsoid semi-axes lengths r_k-s are found using linearized models. We now analyze this process in some detail. For this, suppose that we are given an n-th order continuous dynamical system model

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{18.77}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the system's state at time t, and $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is called the vector field. The solution to (18.77) subject to initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ is called a trajectory and is denoted $\boldsymbol{\phi}_t(\boldsymbol{x}_0)$. For every t, the mapping

$$\boldsymbol{\phi}_t : \mathbb{R}^n \to \mathbb{R}^n,$$

maps the state space $\mathbb{R}^n$ into itself. The mapping $\boldsymbol{\phi}_t$ is called the *flow* of the system (18.77). For example, if (18.77) is linear, that is, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, then

$$\boldsymbol{\phi}_t(\boldsymbol{x}_0) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}_0.$$

Consider now an equilibrium state $\boldsymbol{x}_e$ of (18.77). The local behavior of (18.77) in a neighborhood of $\boldsymbol{x}_e$ is determined by the linearized model of (18.77) about the equilibrium state $\boldsymbol{x}_e$. The linearized model is

$$\frac{d}{dt}\Delta\boldsymbol{x} = D\boldsymbol{f}(\boldsymbol{x}_e)\Delta\boldsymbol{x}, \tag{18.78}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_e$. We can also linearize the system (18.77) about its trajectory $\boldsymbol{\phi}_t(\boldsymbol{x}_0)$. The linearized model about the trajectory $\boldsymbol{\phi}_t(\boldsymbol{x}_0)$ is

$$\frac{d}{dt}\Delta\boldsymbol{x} = D\boldsymbol{f}(\boldsymbol{\phi}_t(\boldsymbol{x}_0))\Delta\boldsymbol{x}, \quad \Delta\boldsymbol{x}(t_0) = \Delta\boldsymbol{x}_0. \tag{18.79}$$

The solution to (18.79) is

$$\Delta\boldsymbol{x}(t) = \boldsymbol{\Phi}_t(\boldsymbol{x}_0)\Delta\boldsymbol{x}_0.$$

We call the $n \times n$ matrix $\boldsymbol{\Phi}_t(\boldsymbol{x}_0)$ the *state transition matrix* of the system (18.79)—see Section 6.2 for more information on the state transition matrix properties. The transition matrix satisfies the equation

$$\frac{d}{dt}\boldsymbol{\Phi} = D\boldsymbol{f}(\boldsymbol{\phi}_t(\boldsymbol{x}_0))\boldsymbol{\Phi}, \quad \boldsymbol{\Phi}_0 = \boldsymbol{I}_n.$$

Let

$$\{m_i(t)\}_{i=1}^n$$

be the eigenvalues of the transition matrix $\boldsymbol{\Phi}_t(\boldsymbol{x}_0)$. Then, the Lyapunov exponents of $\boldsymbol{x}_0$ are

$$\boxed{\mu_i\left(\boldsymbol{x}_0\right) = \lim_{t\to\infty}\frac{1}{t}\ln\left|m_i(t)\right|, \quad i = 1, 2, \ldots, n} \tag{18.80}$$

if the limits exist. The above is an alternative to (18.75) representation of the Lyapunov exponents.

Example 18.2 The Lyapunov exponents of an equilibrium state $\boldsymbol{x}_e$ of the system (18.77), with $t_0 = 0$, can be found by first linearizing (18.77) about the equilibrium state to obtain a linearized model of the form (18.78). The transition matrix of (18.78) is

$$\boldsymbol{\Phi}_t(\boldsymbol{x}_e) = e^{D\boldsymbol{f}(\boldsymbol{x}_e)t}.$$

The i-th eigenvalue of the transition matrix $\boldsymbol{\Phi}_t(\boldsymbol{x}_e)$ is $m_i(t) = e^{\lambda_i t}$, where λ_i is the i-th eigenvalue of the matrix $D\boldsymbol{f}(\boldsymbol{x}_e)$. Therefore, the Lyapunov exponents of $\boldsymbol{x}_e$ are

$$\begin{aligned}
\mu_i\left(\boldsymbol{x}_e\right) &= \lim_{t\to\infty}\frac{1}{t}\ln\left|m_i(t)\right| \\
&= \lim_{t\to\infty}\frac{1}{t}\ln\left|e^{\Re\lambda_i t}e^{j\Im\lambda_i t}\right| \\
&= \lim_{t\to\infty}\frac{1}{t}\ln\left(\left|e^{\Re\lambda_i t}\right|\left|\cos\left(\Im\lambda_i t\right) + j\sin\left(\Im\lambda_i t\right)\right|\right) \\
&= \lim_{t\to\infty}\frac{1}{t}t\,\Re\lambda_i \\
&= \Re\lambda_i.
\end{aligned} \tag{18.81}$$

Thus, the Lyapunov exponents in this case are the real parts of the eigenvalues of the linearized system about the equilibrium state. Note that if $\boldsymbol{x}_0$ is in the basin of attraction of $\boldsymbol{x}_e$, that is, if

$$\lim_{t\to\infty}\boldsymbol{\phi}_t(\boldsymbol{x}_0) = \boldsymbol{x}_e,$$

then the Lyapunov exponents of $\boldsymbol{x}_0$ are the same as the Lyapunov exponents of $\boldsymbol{x}_e$.

Example 18.3 Suppose that we are given a discrete-time dynamical system model,

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{18.82}$$

Let $\boldsymbol{x}_e$ be the equilibrium state of the model (18.82), that is,

$$\boldsymbol{x}_e = \boldsymbol{f}\left(\boldsymbol{x}_e\right).$$

The linearized model is

$$\Delta\boldsymbol{x}(k+1) = D\boldsymbol{f}(\boldsymbol{x}_e)\Delta x(k).$$

The state transition matrix of the linearized model is

$$\boldsymbol{\Phi}_k(\boldsymbol{x}_e) = \left(D\boldsymbol{f}\left(\boldsymbol{x}_e\right)\right)^k \tag{18.83}$$

If λ_i is the i-th eigenvalue of the matrix $D\boldsymbol{f}\left(\boldsymbol{x}_e\right)$, then λ_i^k is the i-th eigenvalue of the transition matrix (18.83). The Lyapunov exponents of $\boldsymbol{x}_e$, in this case, are

$$\begin{aligned} \mu_i(\boldsymbol{x}_e) &= \lim_{k\to\infty} \frac{1}{k} \ln |m_i(k)| \\ &= \lim_{k\to\infty} \frac{1}{k} \ln |\lambda_i|^k \\ &= \ln |\lambda_i| . \end{aligned} \tag{18.84}$$

In summary, Lyapunov exponents can be used to characterize a chaotic motion. Specifically, a necessary condition for a chaotic motion to occur in a dynamical system with bounded trajectories is that the largest Lyapunov exponent be positive. An efficient numerical method for evaluating the largest Lyapunov exponent can be found, for example, in the paper by Parker and Chua [263], see also their book [264] on practical numerical algorithms for chaotic systems.

In Figure 18.19, we show a plot of the Lyapunov exponent versus the parameter ρ for the logistic equation (18.10).

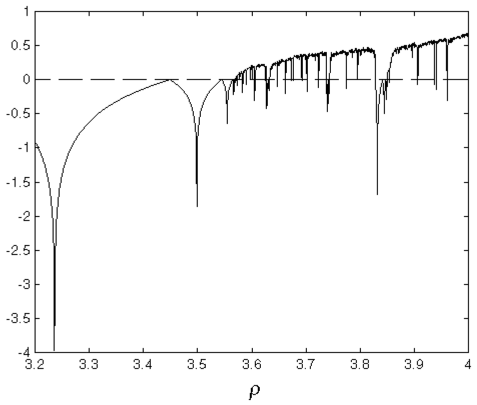


Figure 18.19: Lyapunov exponent versus parameter ρ for the logistic equation (18.10).

18.5 Discretization Chaos

In this section, we consider examples of chaotic systems that are discrete-time analogs of continuous-time systems, where the latter systems do not exhibit any chaotic behavior. We begin by considering a method of obtaining discrete models from continuous models. Suppose that we are given a scalar differential equation

$$\dot{x} = ax, \quad x(t_0) = x_0. \tag{18.85}$$

The solution to the above differential equation is

$$x(t) = e^{a(t-t_0)} x(t_0). \tag{18.86}$$

Let $\Delta t = t - t_0$. Then, we can express (18.86) as

$$x(1) = e^{a\Delta t} x(0), \tag{18.87}$$

where $x(1) = x(t_0 + \Delta t)$ and $x(0) = x(t_0)$. More generally,

$$x(k+1) = e^{a\Delta t} x(k), \tag{18.88}$$

where $x(k+1) = x((k+1)\Delta t)$ and $x(k) = x(k\Delta t)$. The discrete model (18.88) of the continuous differential equation (18.85) is exact in the sense that the values of (18.86) and (18.88) are the same at the sampling times. We generalize the above method to a class of nonlinear differential equations. Consider a scalar differential equation

$$\dot{x}(t) = x(t)h(x(t)). \tag{18.89}$$

Dividing both sides of the above by x and then integrating yields

$$\int \frac{dx}{x} = \int h(x(t))dt. \tag{18.90}$$

Suppose now that we hold $h(x(t))$ constant on the interval $\left[\ k\Delta t,\quad (k+1)\Delta t\ \right)$. Then, evaluating (18.90) over the above interval gives

$$\ln\left(\frac{x((k+1)\Delta t)}{x(k\Delta t)}\right) = h(x(k\Delta t))\Delta t.$$

Using the the notation introduced after (18.88), we represent the above as

$$\boxed{x(k+1) = e^{h(x(k))\Delta t}x(k)} \tag{18.91}$$

Example 18.4 Consider the system of differential equations

$$\left.\begin{array}{rcl} \dot{x}_1 & = & \gamma x_1 (1 - x_1 - 0.5x_2) \\ \dot{x}_2 & = & 2\gamma x_2 (1 - 0.5x_1 - x_2) \end{array}\right\} \tag{18.92}$$

where $\gamma = 2.5$. The above system models a pair of competing species. For a discussion of modeling of competing species population dynamics, the reader is referred to Subsection 2.4.7. The system (18.92) was analyzed by Grantham and Athalye [119]. A trajectory of the system (18.92) for $x_1(0) = x_2(0) = 0.5$ is shown in Figure 18.20. We discretize (18.92) using the formula (18.91). For $\Delta t = 1$, we obtain the following discrete-time analog

$$\left.\begin{array}{rcl} x_1(k+1) & = & x_1(k)\exp\left(\gamma\left(1 - x_1(k) - 0.5x_2(k)\right)\right) \\ x_2(k+1) & = & x_2(k)\exp\left(2\gamma\left(1 - 0.5x_1(k) - x_2(k)\right)\right) \end{array}\right\} \tag{18.93}$$

The trajectory of the discrete model (18.93) starting from the same initial condition as its continuous-time analog is shown in Figure 18.21. The continuous-time model trajectory asymptotically converges to the equilibrium state $\left[\ 2/3 \quad 2/3\ \right]^\top$, while the discrete-time trajectory wanders around chaotically.

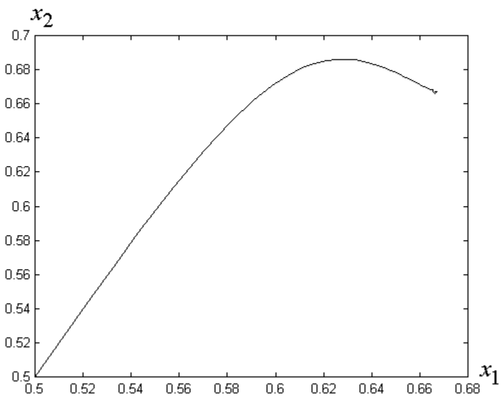


Figure 18.20: A trajectory of the system (18.92).

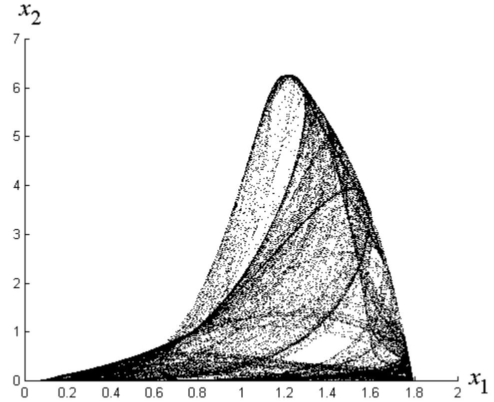


Figure 18.21: A trajectory of the system (18.92).

18.6 Controlling Chaotic Systems

A trajectory of a chaotic system behaves in a random-like fashion. The set on which the chaotic motion is taking place attracts trajectories starting from a set of initial conditions. Roughly speaking, this attracting set is called a *chaotic attractor*. It is not possible to predict when a trajectory will pass through a given point in the chaotic attractor. However, any given point in the chaotic attractor will be visited by the chaotic system's trajectory in a finite time. This property of a chaotic system is referred to as *ergodicity*. Using the ergodicity property one can propose a control algorithm whose task it is to bring the chaotic system's trajectory to a prespecified state in the system's state space. The idea behind the chaotic control algorithm, that we discuss next, is to set up "fishing nets" in the chaotic attractor in a neighborhood of an operating point to "catch" the system trajectory. Once the trajectory is "caught", a control law is used to bring the trajectory to the desired operating point. The controller is constructed using linear quadratic regulator (LQR) technique. The chaotic algorithm is illustrated with two examples; the Hénon map and the bouncing ball system.

18.6.1 Ingredients of Chaotic Control Algorithm

Chaotic motion is characterized by extreme sensitivity to initial conditions. This might suggest that it would be impossible, for example, to stabilize a chaotic system with respect to a desired operating point, because any perturbation resulting from applying a control action would lead to a chaotic motion's exponential grow in time. However, it turns out that precisely this property of a chaotic motion can be used to convert a chaotic attractor to a periodic motion. A chaotic attractor, as observed by Ott, Grebogi, and Yorke [259] typically has imbedded within it an infinite number of unstable periodic orbits. These periodic orbits are dense in the attractor in the sense that they pass through any neighborhood, however small, of any point in the attractor. Ott et al. [259] noticed that these properties could be advantageous in achieving control objectives. They proposed a method, now called the OGY method, which involves converting, using small time-dependent perturbations of a system parameter, a chaotic attractor into a periodic orbit. The idea behind this method is to exploit the ergodicity property of a chaotic system. Specifically, all of the unstable periodic orbits of the chaotic attractor will eventually be visited by the system trajectory. Once the trajectory gets to a neighborhood of the desired unstable periodic orbit, a small perturbation is applied to stabilize the orbit of interest.

The chaotic control algorithm, that we discuss next, can be viewed as a variant of the

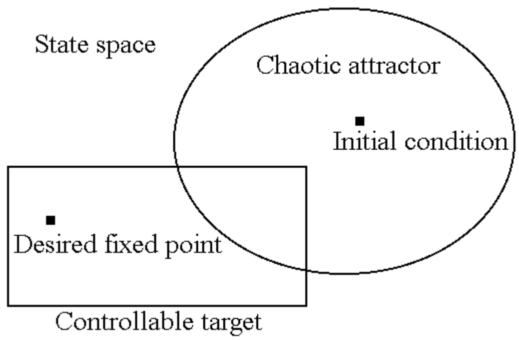


Figure 18.22: A pictorial representation of the chaotic algorithm.

OGY method. This algorithm was presented by Vincent [337] and is characterized by two ingredients. They are: a chaotic attractor and a controllable target. These components of the chaotic algorithm are illustrated in Figure 18.22. We assume that either the system under consideration is chaotic or that it can be made chaotic using an open-loop control. Next, we select a fixed point of interest and assume that a feedback law can be constructed such that the closed-loop system is non-chaotic and the chosen fixed point is asymptotically stable. The region of attraction of the fixed point should have a non-empty intersection with the chaotic attractor. A controllable target can be any subset of the region of attraction of the fixed point that has a non-empty intersection with the chaotic attractor. Note that the fixed point does not have to be located in the chaotic attractor. If we start the system from any initial state in the chaotic attractor, the resulting chaotic trajectory will at some time arrive at the intersection region of the chaotic attractor and the controllable target. The chaotic algorithm keeps track when the trajectory reaches this intersection region. When this happens, the open-loop control, that was used to create chaotic motion, is turned off and the feedback control is turned on to drive the system towards the fixed point. The region of attraction intersecting the chaotic attractor plays the role of a "fishing net" for the chaotic trajectory.

18.6.2 Chaotic Control Algorithm

We consider two classes of single-input dynamical systems. The first class is modeled by nonlinear difference equation,

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), u(k)), \tag{18.94}$$

and the other class is modeled by nonlinear differential equation,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), u(t)), \tag{18.95}$$

The control u, in both cases, will be bounded. The control u will be either a specified function of time or a specified function of the system state. In the first case, we have an open-loop control, in the latter case we have a closed-loop control. We assume that for a given system there exists an open-loop control such that the system has a chaotic attractor. In addition, we assume that for a specified constant control $u = u_e$ there is a corresponding equilibrium state located in a vicinity of the chaotic attractor. The equilibrium state satisfies the equation

$$\boldsymbol{x}_e = \boldsymbol{f}(\boldsymbol{x}_e, u_e)$$

for the discrete-time system, and it satisfies the relation

$$\boldsymbol{0} = \boldsymbol{f}(\boldsymbol{x}_e, u_e)$$

for the continuous-time system.

We now briefly describe the way we obtain a controllable target for the system of interest. We first linearize the system about the selected equilibrium state. The linearized discrete-time system has the form,

$$\Delta\boldsymbol{x}(k+1) = \boldsymbol{A}\Delta\boldsymbol{x}(k) + \boldsymbol{b}\Delta u(k),$$

where $\Delta\boldsymbol{x}(k) = \boldsymbol{x}(k) - \boldsymbol{x}_e$ and $\Delta u(k) = u(k) - u_e$. The linearized continuous-time system has the form

$$\frac{d}{dt}\Delta\boldsymbol{x}(t) = \boldsymbol{A}\Delta\boldsymbol{x}(t) + \boldsymbol{b}\Delta u(t),$$

where $\Delta\boldsymbol{x}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_e$ and $\Delta u(t) = u(t) - u_e$. Then, the LQR method, described in Chapter 9, is used to construct a state variable feedback control law, $\Delta u(\boldsymbol{\Delta x}) = -\boldsymbol{k}\Delta\boldsymbol{x}$, that guarantees the origin of the linearized system,

$$\Delta\boldsymbol{x}(k+1) = (\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k})\Delta\boldsymbol{x}(k), \tag{18.96}$$

or

$$\frac{d}{dt}\Delta \boldsymbol{x}(t) = (\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k})\Delta \boldsymbol{x}(t), \tag{18.97}$$

to be asymptotically stable. This implies that for the corresponding nonlinear system driven by the state feedback controller, the equilibrium state about which the system was linearized, is asymptotically stable. We use a Lyapunov function obtained from the linearized system to estimate the region of asymptotic stability (RAS) of the closed-loop nonlinear system. The Lyapunov function has the form, $V(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{P}_l \boldsymbol{x}$, where $\boldsymbol{P}_l$ is obtained by solving the corresponding Lyapunov equation for the optimal closed-loop linear system given by (18.96) or by (18.97). This Lyapunov function is then used to determine a controllable target for the nonlinear system driven by a bounded controller of the form,

$$u = \operatorname{sat}(\boldsymbol{k}\boldsymbol{x}) + u_e, \tag{18.98}$$

where

$$\operatorname{sat}(v) = \begin{cases} u_{\max} - u_e & \text{if} \quad v > u_{\max} - u_e \\ v & \text{if} \quad u_{\min} - u_e \le v \le u_{\max} - u_e \\ u_{\min} - u_e & \text{if} \quad v < u_{\min} - u_e, \end{cases}$$

and the bounds $u_{\min}$ and $u_{\max}$ are specified by a designer. Using the above control law, we then seek the largest level curve $V(\boldsymbol{x}) = V_{\max}$ such that for the points $\{\boldsymbol{x} : V(\boldsymbol{x}) \le V_{\max}\}$, for the discrete-time system,

$$\begin{aligned} \Delta V(k) &= V(\boldsymbol{x}(k+1)) - V(\boldsymbol{x}(k)) \\ &= (\boldsymbol{x}(k+1) - \boldsymbol{x}_e)^\top \boldsymbol{P}_l(\boldsymbol{x}(k+1) - \boldsymbol{x}_e) - (\boldsymbol{x}(k) - \boldsymbol{x}_e)^\top \boldsymbol{P}_l(\boldsymbol{x}(k) - \boldsymbol{x}_e) \\ &= (\boldsymbol{f}(\boldsymbol{x}(k), u(k)) - \boldsymbol{x}_e)^\top \boldsymbol{P}_l \left(\boldsymbol{f}(\boldsymbol{x}(k), u(k)) - \boldsymbol{x}_e\right) - (\boldsymbol{x}(k) - \boldsymbol{x}_e)^\top \boldsymbol{P}_l \left(\boldsymbol{x}(k) - \boldsymbol{x}_e\right) \\ &< 0, \end{aligned}$$

or for the continuous-time system,

$$\begin{aligned} \dot{V}(\boldsymbol{x}) &= 2\left(\boldsymbol{x} - \boldsymbol{x}_e\right)^\top \boldsymbol{P}_l \dot{\boldsymbol{x}} \\ &= 2\left(\boldsymbol{x} - \boldsymbol{x}_e\right)^\top \boldsymbol{P}_l \boldsymbol{f}(\boldsymbol{x}, u) \\ &< 0. \end{aligned}$$

We are now ready to describe the chaotic algorithm. When the given nonlinear system trajectory is in the region $\{\boldsymbol{x} : V(\boldsymbol{x}) > V_{\max}\}$, an open-loop control is applied to keep the system in the chaotic attractor until the system reaches a point in the set $\{\boldsymbol{x} : V(\boldsymbol{x}) \le V_{\max}\}$, where the open-loop control is turned off and the feedback saturating control law is applied to

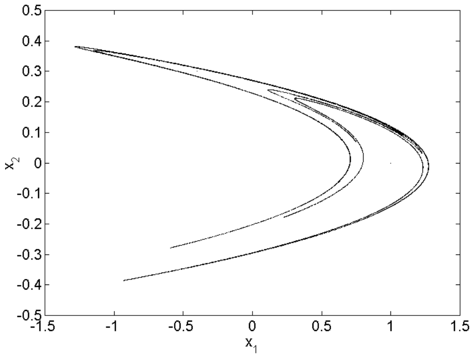


Figure 18.23: An orbit of the Hénon map.

force the system trajectory to asymptotically approach the equilibrium state. More formally, we state the chaotic algorithm in the form of the following algorithm:

IF $V(\boldsymbol{x}) > V_{\max}$
THEN $u =$ open-loop control to keep the system in the chaotic attractor
ELSE $u = \text{sat}(\boldsymbol{kx}) + u_e$
END

We illustrate the above algorithm when applied to the Hénon map.

Example 18.5 The dynamical system known as the Hénon map is described by the difference equations,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 + x_2(k) - 1.4x_1^2(k) \\ 0.3x_1(k) \end{bmatrix}. \tag{18.99}$$

It was named after the French astronomer M. Hénon who discovered chaotic behavior of the above system in 1974. The orbit of $\boldsymbol{x}(0) = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top$ of the Hénon map is shown in Figure 18.23. We modify the Hénon map by adding the control input into the first equation so that the map becomes,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} -1.4x_1^2(k) + x_2(k) + 1 + u \\ 0.3x_1(k) \end{bmatrix}. \tag{18.100}$$

There are two fixed states corresponding to $u = u_e = 0$. They are obtained by solving two algebraic equations,

$$\begin{cases} x_{1,eq} &= -1.4x_{1,e}^2 + x_{2,e} + 1 \\ x_{2,eq} &= 0.3x_{1,e} \end{cases}$$

Solving the above gives

$$\begin{bmatrix} 0.6314 & 0.1894 \end{bmatrix}^\top \quad \text{and} \quad \begin{bmatrix} -1.1314 & -0.3394 \end{bmatrix}^\top .$$

We select to work with the first of the above fixed states. We linearize the system (18.100) about the first fixed state to obtain

$$\boldsymbol{A} = \left.\begin{bmatrix} -2.8x_1 & 1 \\ 0.3 & 0 \end{bmatrix}\right|_{x_1=0.6314} = \begin{bmatrix} -1.7678 & 1 \\ 0.3 & 0 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

We then use the discrete LQR design method to find the state feedback. The weight matrices of the associated quadratic performance index are chosen to be $\boldsymbol{Q} = \boldsymbol{I}_2$ and $r = 1$. Solving the discrete algebraic Riccati equation yields

$$\boldsymbol{P} = \begin{bmatrix} 3.9351 & -1.5008 \\ -1.5008 & 1.7974 \end{bmatrix} \quad \text{and} \quad \boldsymbol{k} = \begin{bmatrix} -1.5008 & 0.7974 \end{bmatrix}.$$

Hence,

$$\boldsymbol{A}_c = \boldsymbol{A} - \boldsymbol{b}\boldsymbol{k} = \begin{bmatrix} -0.2670 & 0.2026 \\ 0.3000 & 0 \end{bmatrix}.$$

Solving the discrete Lyapunov equation,

$$\boldsymbol{A}_c^\top \boldsymbol{P}_l \boldsymbol{A}_c - \boldsymbol{P}_l = -\boldsymbol{I}_2$$

gives

$$\boldsymbol{P}_l = \begin{bmatrix} 1.1902 & -0.0686 \\ -0.0686 & 1.0489 \end{bmatrix}.$$

The next step is to calculate $V_{\max}$ corresponding to the largest level curve such that the Lyapunov first difference $\Delta V < 0$ on this level curve for the nonlinear model. We obtain level curves by trial and error as follows. We first evaluate

$$\begin{aligned} V &= (\boldsymbol{x} - \boldsymbol{x}_e)^\top \boldsymbol{P}_l (\boldsymbol{x} - \boldsymbol{x}_e) \\ &= \begin{bmatrix} x_1 - 0.6314 & x_2 - 0.1894 \end{bmatrix} \begin{bmatrix} 1.1902 & -0.0686 \\ -0.0686 & 1.0489 \end{bmatrix} \begin{bmatrix} x_1 - 0.6314 \\ x_2 - 0.1894 \end{bmatrix} \\ &= 1.1902(x_1 - 0.6314)^2 - 0.1372(x_1 - 0.6314)(x_2 - 0.1894) \\ &\quad +1.0489(x_2 - 0.1894)^2. \end{aligned} \tag{18.101}$$

We next choose an initial state $\boldsymbol{x}(0)$ such that $V(\boldsymbol{x}(0)) = V_0$, where V_0 is a constant selected by the designer. Let $x_2(0) = 0.1894$, which is the value of the second component of the

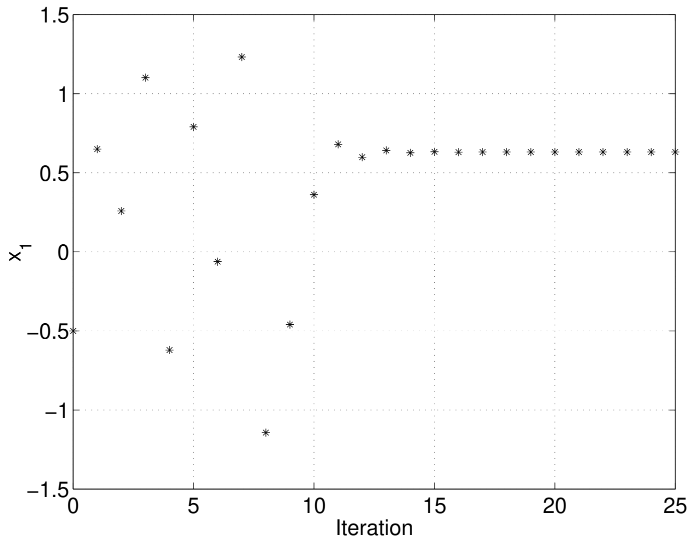


Figure 18.24: A plot of x_1 versus time for the controlled Hénon map of Example 18.5.

equilibrium state of interest. Substituting this value into (18.101) allows us to compute the first component of $\boldsymbol{x}(0)$ such that $V(\boldsymbol{x}(0)) = V_0$. We have

$$\boldsymbol{x}(0) = \begin{bmatrix} \sqrt{0.8402V_0} + 0.6314 & 0.1894 \end{bmatrix}^\top .$$

Then, a level curve is constructed by solving the differential equation,

$$\nabla^\top V \dot{\boldsymbol{x}} = 0,$$

subject to the initial condition $\boldsymbol{x}(0)$. To satisfy the above equation, we can select

$$\begin{cases} \dot{x}_1 &= \frac{\partial V}{\partial x_2} \\ \dot{x}_2 &= -\frac{\partial V}{\partial x_1} \end{cases} \tag{18.102}$$

After constructing a level curve, we evaluate ΔV at these points and check if $\Delta V < 0$ at every point on the level curve. For the control law

$$u = 0.3\,\mathrm{sat}(\boldsymbol{k}\boldsymbol{x}/0.3), \tag{18.103}$$

applied to the nonlinear system (18.100), we found $V_{\max} = 0.0109$. Plots of x_1 and x_2 versus time for the closed-loop system (18.100), (18.103) are shown in Figures 18.24 and 18.25, respectively, for the initial condition $\boldsymbol{x}(0) = \begin{bmatrix} -0.5 & 0 \end{bmatrix}^\top$. The orbit of $\boldsymbol{x}(0) = \begin{bmatrix} -0.5 & 0 \end{bmatrix}^\top$ of the controlled Hénon map is shown in Figure 18.26, where the controllable target is also depicted as the set of points inside the ellipse $V(\boldsymbol{x} - \boldsymbol{x}_e) = 0.0109$.

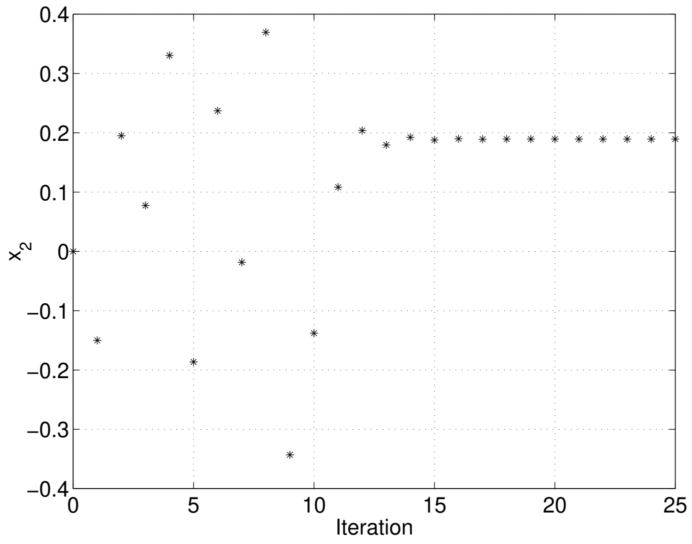


Figure 18.25: A plot of x_2 versus time for the controlled Hénon map of Example 18.5.

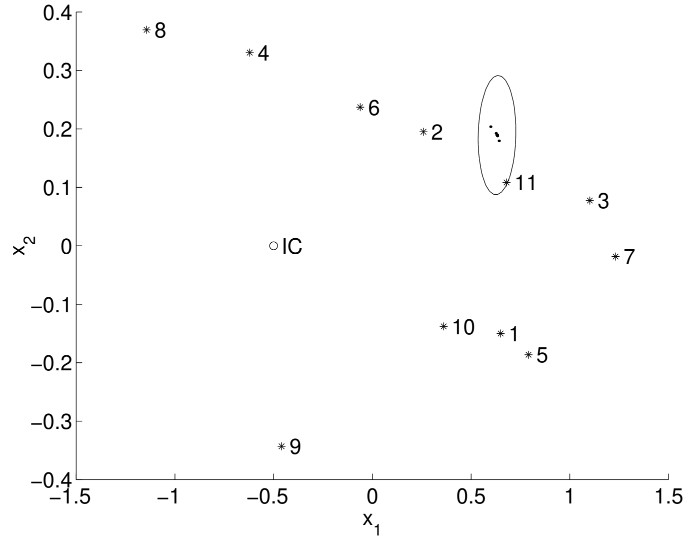


Figure 18.26: An orbit of the controlled Hénon map of Example 18.5, where the initial condition is denoted by "○" and labelled "IC". The subsequent iterations prior to hitting the controllable target are denoted by "⋆". The iterations in the controllable target are denoted by "·".

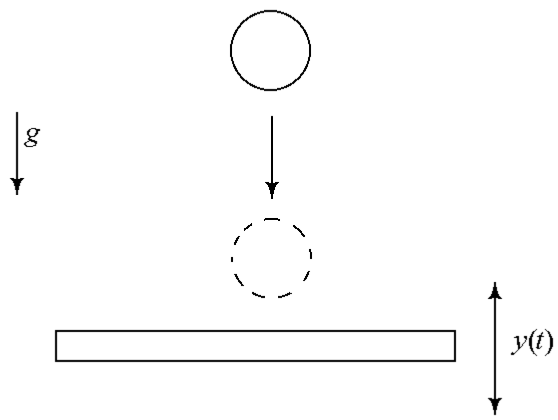


Figure 18.27: Bouncing ball on a vibrating plate.

Example 18.6 Controlling the bouncing ball on a vibrating plate
The bouncing ball on a vibrating plate is an another example of a chaotic system. In this example, we use the chaotic algorithm to design a control law for the vibrating plate, as shown in Figure 18.27, so that the ball, starting at rest on the plate can be bounced up to and then maintained bouncing at a prescribed height. The amplitude of the vibrating plate will be kept fixed. The control input is the frequency of the plate vibrations. To proceed, we need a mathematical model of the ball-plate system, which we derive next.

Modeling a Bouncing Ball on a Vibrating Plate

Let the displacement of the plate in the vertical direction be

$$y(t) = A\sin(\omega t),$$

where A is the amplitude of the plate displacement, ω is the angular frequency of vibrations and t is time. Then, the velocity of the plate is

$$W(t) = \frac{dy(t)}{dt} = A\omega\cos(\omega t). \tag{18.104}$$

We denote by U the velocity of the ball just before impact and let V denote the velocity of the ball just after impact. In our modeling process, we use the principle of conservation of momentum, which states that momentum of the system before and after collisions is constant, that is,

$$\sum(\text{masses} \times \text{velocities})_{before} = \sum(\text{masses} \times \text{velocities})_{after}.$$

Applying the conservation of momentum principle to our system consisting of the bouncing ball and the vibrating plate yields

$$m_{ball}U + m_{plate}W_{before} = m_{ball}V + m_{plate}W_{after}. \tag{18.105}$$

To proceed, we define the so called *coefficient of restitution*

$$e = -\frac{V - W_{after}}{U - W_{before}}. \tag{18.106}$$

Performing simple manipulations on (18.106), we obtain

$$W_{after} = Ue - W_{before}e + V. \tag{18.107}$$

Substituting the above into (18.105) and rearranging gives

$$m_{ball}U + m_{plate}W_{before} = (m_{ball} + m_{plate})\,V + em_{plate}U - em_{plate}W_{before}. \tag{18.108}$$

Dividing both sides of the above by $(m_{ball} + m_{plate})$ and rearranging, we get

$$V = \frac{m_{ball} - em_{plate}}{m_{ball} + m_{plate}}U + \frac{m_{plate} + em_{plate}}{m_{ball} + m_{plate}}W_{before}. \tag{18.109}$$

Let

$$M = \frac{m_{ball}}{m_{plate}}.$$

Substituting the above into (18.109) gives

$$V = \frac{M - e}{M + 1}U + \frac{e + 1}{M + 1}W_{before}. \tag{18.110}$$

The ball is acted upon by a constant acceleration g and therefore $V(t) = gt$. Hence, from one bounce to the next,

$$t(k+1) = t(k) + \frac{2V(k)}{g}. \tag{18.111}$$

We assume that the amplitude, A, of the plate vibrations is negligible compared with the height of the bounce. Referring to Figure 18.28, we obtain

$$U(k+1) = -V(k). \tag{18.112}$$

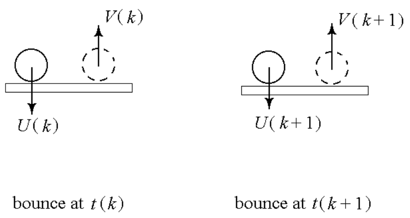


Figure 18.28: Velocities of the bouncing ball from one bounce to the next.

Let

$$a_1 = \frac{e+1}{M+1} \quad \text{and} \quad a_2 = \frac{M-e}{M+1}.$$

Taking into account the above notation and (18.110), we obtain a relation between velocities of the bouncing ball and the vibrating plate at the bounce at a time $t(k+1)$,

$$V(k+1) = a_2 U(k+1) + a_1 W_{before}(k+1). \tag{18.113}$$

We use (18.112) to represent (18.113) as

$$V(k+1) = -a_2 V(k) + a_1 W(k+1). \tag{18.114}$$

Furthermore, by (18.104) and (18.111),

$$W(k+1) = A\omega \cos(\omega t(k+1)) = A\omega \cos\left(\omega\left(t(k) + \frac{2V(k)}{g}\right)\right).$$

Hence, we can represent (18.114) as

$$V(k+1) = -a_2 V(k) + a_1 A\omega \cos\left(\omega\left(t(k) + \frac{2V(k)}{g}\right)\right). \tag{18.115}$$

Let

$$\phi(k) = \omega t(k) \quad \text{and} \quad \psi(k) = 2\omega V(k)/g, \tag{18.116}$$

where $\phi(k)$ is the current phase angle of the plate and $\psi(k)$ is the phase angle change between the current bounce of the ball and the next. Then, we can write

$$\boxed{\begin{array}{rcl} \phi(k+1) & = & \phi(k) + \psi(k) \\ \psi(k+1) & = & -a_2\psi(k) + \hat{a}_1\omega^2\cos(\phi(k)+\psi(k)), \end{array}} \tag{18.117}$$

Table 18.1: Simulation parameters of the ball map.

M	1/26
e	0.8
A	0.013 m

where $\hat{a}_1 = 2Aa_1/g$. The above dynamical system is known as the *ball map*. In our simulations, we use the same values of the ball map parameters as those used by Vincent [337], which are given in Table 18.1. Using these parameters, we compute $a_1 = 1.73333$, $\hat{a}_1 = 0.004594\ \text{sec}^2$, and $a_2 = -0.733333$.

There are many possible periodic equilibrium solutions to the ball map (18.117). For instance, the ball can bounce to a fixed height at every n cycles of the plate, or it can bounce to m different heights at nm cycles of the plate. Here, we will only be concerned with the case when $m = 1$. There are bands of frequencies of the vibrating plate that will not support periodic motion for $m = 1$. Examples of such frequency bands, as listed by Vincent [337], are:

$$\begin{aligned} 32.3966 &< \omega < 33.0782 \\ 36.4575 &< \omega < 38.1954 \\ 40.5278 &< \omega < 42.7038 \\ 44.4249 &< \omega < 46.7796 \end{aligned}$$

A frequency of the plate vibration from one of the above ranges will either result in a chaotic motion of the ball or a chaotic transient to a periodic motion with $m > 1$. In our example, we use $\omega = 45$ rad/sec to generate chaotic motion.

The ball map (18.117) may produce negative values for ψ corresponding to negative bounce heights, which is physically impossible. To overcome this problem, whenever the map produces a negative value for ψ, this value is multiplied by a random negative number between zero and one in order to simulate the fact that the ball cannot go through the plate. In Figure 18.29, an orbit of the modified ball map for the initial conditions $\phi(0) = 0$ and $\psi(0) = 9.1743$ rads and $\omega = 45$ rad/sec is shown. Note that the values of ψ are divided by 2π before plotting. The chaotic attractor is located between the upper and lower bounds as depicted in the figure. The upper bound is given by

$$\psi_{upper} = 2\hat{a}_1\omega^2 + \hat{a}_1\omega^2\cos(\phi(k) + \psi(k)),$$

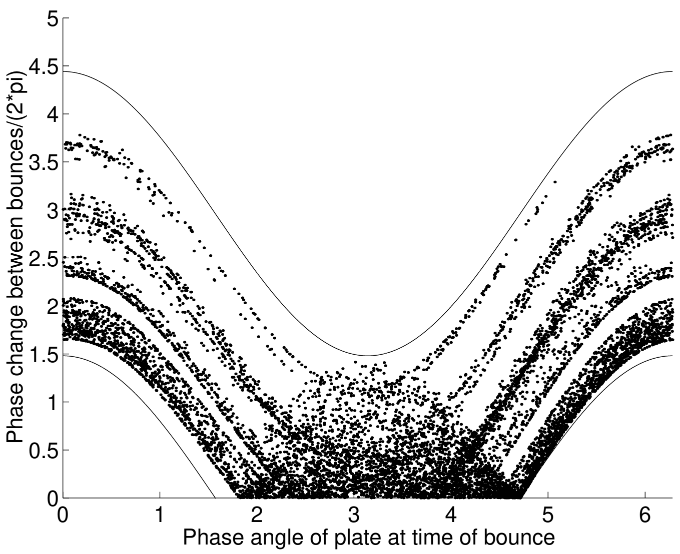


Figure 18.29: An orbit of the ball map.

while the lower bound is

$$\psi_{lower} = \hat{a}_1\omega^2\cos(\phi(k) + \psi(k)),$$

When ψ_{lower} goes negative it is set equal to zero.

Controller Construction for a Bouncing Ball on a Vibrating Plate

To proceed with the controller design, we first find fixed points of the ball map. They are found by solving the equations,

$$\left\{ \begin{array}{rcl} \bar{\phi} & = & \bar{\phi} + \bar{\psi} \\ \bar{\psi} & = & -a_2\bar{\psi} + \hat{a}_1\omega^2\cos(\bar{\phi} + \bar{\psi}). \end{array} \right. \tag{18.118}$$

The fixed points are obtained by evaluating the right-hand side of the first above equation modulo $2n\pi$, where n is the number of cycles of the plate between bounces. We obtain,

$$\left\{ \begin{array}{rcl} \bar{\psi} & = & 2n\pi \\ \cos(\bar{\phi}) & = & \frac{2n\pi(1+a_2)}{\hat{a}_1\omega^2}. \end{array} \right. \tag{18.119}$$

Linearizing the ball map about (18.119) gives

$$\left[\begin{array}{c} x_1(k+1) \\ x_2(k+1) \end{array} \right] = \left[\begin{array}{cc} 1 & 1 \\ a_{21} & a_{22} \end{array} \right] \left[\begin{array}{c} x_1(k) \\ x_2(k) \end{array} \right], \tag{18.120}$$

where

$$\begin{aligned} x_1(k) &= \phi(k) - \bar{\phi} \\ x_2(k) &= \psi(k) - \bar{\psi} \\ a_{21} &= -\hat{a}_1\omega^2 \sin(\bar{\phi}) \\ a_{22} &= -a_2 + a_{21}. \end{aligned}$$

We next construct a control law for selecting frequencies of the vibrating plate in such a way that the steady-state constant height of the bouncing ball is reached. Vincent [337] showed that for $n = 1$ the range of frequencies for which there exists an asymptotically stable equilibrium is $[19.0977 \text{ rad/sec}, 28.9505 \text{ rad/sec}]$. The corresponding range of the bounce heights is $[0.0578 \text{ m}, 0.1327 \text{ m}]$. Thus, for example the height, h, of 0.1 m is in the above range and from the energy conservation law,

$$\frac{1}{2} m_{ball} V^2 = m_{ball} g h,$$

we obtain

$$V = 1.3994 \text{ m/sec}^2.$$

Using the above, (18.116), and (18.119), we get

$$\bar{\psi} = 2\pi = \frac{2\omega V}{g} \quad \text{and} \quad \cos(\bar{\phi}) = \frac{2\pi(1 + a_2)}{\hat{a}_1\omega^2},$$

from which we compute,

$$\omega = 22 \text{ rad/sec} \quad \text{and} \quad \bar{\phi} = 41.1^\circ = 0.7173 \text{ rad}.$$

The linearized ball map about the above equilibrium state is

$$\boldsymbol{x}(k+1) = \begin{bmatrix} 1 & 1 \\ -1.4617 & -0.7283 \end{bmatrix} \boldsymbol{x}(k),$$

which has its poles in the open unit circle. Solving the discrete Lyapunov equation, $\boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P} = -\boldsymbol{I}_2$, gives

$$\boldsymbol{P} = \begin{bmatrix} 8.6414 & 4.6978 \\ 4.6978 & 5.9598 \end{bmatrix}.$$

Employing the same method as in the previous example, we obtain $V_{\max} = 5.006$. The controllable target is

$$\left\{ \begin{bmatrix} \phi \\ \psi \end{bmatrix} : V(\phi - \bar{\phi}, \psi - \bar{\psi}) = \begin{bmatrix} \phi - 0.7173 & \psi - 2\pi \end{bmatrix} \boldsymbol{P} \begin{bmatrix} \phi - 0.7173 \\ \psi - 2\pi \end{bmatrix} \leq V_{max} \right\}.$$

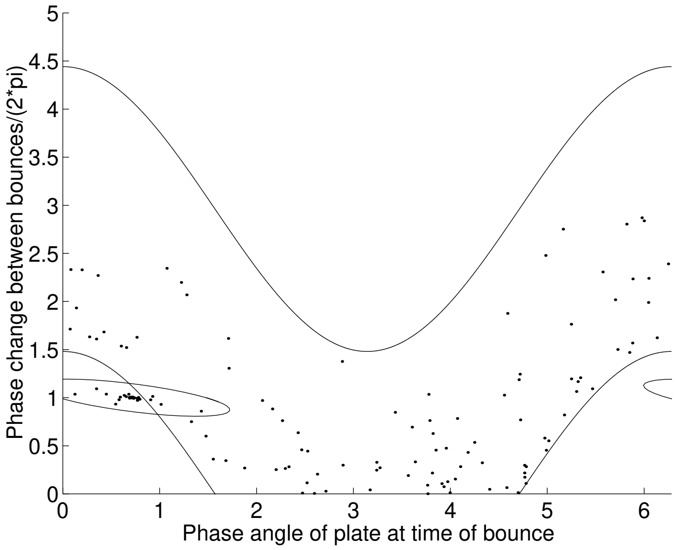


Figure 18.30: An orbit of the controlled modified ball map.

The controllable target is depicted in Figure 18.30. We use $\omega = 45$ rad/sec when the system trajectory is outside of the controllable target, and apply $\omega = 22$ rad/sec when the system trajectory hits the controllable target, that is, we use the algorithm:

IF $V(\phi - \bar{\phi}, \psi - \bar{\psi}) > V_{\max}$
THEN $\omega = 45$ rad/sec
ELSE $\omega = 22$ rad/sec
END

An orbit of $\phi(0) = 0$, $\psi(0) = 9.1743$ of the controlled modified ball map using the above algorithm is shown in Figure 18.30. A plot of $\phi(k)$ versus time is shown in Figure 18.31. A plot of $\psi(k)$ versus time is shown in Figure 18.32.

18.7 An Application to Secure Communication

The purpose of secure communication is to limit the availability of the transmitted information to a third party even if the communication channel is public or has been compromised. With an ever increasing reliance on wireless and internet communication of sensitive information, including data related to individuals, electronic commerce and other financial transactions, secure communication is of utmost importance.

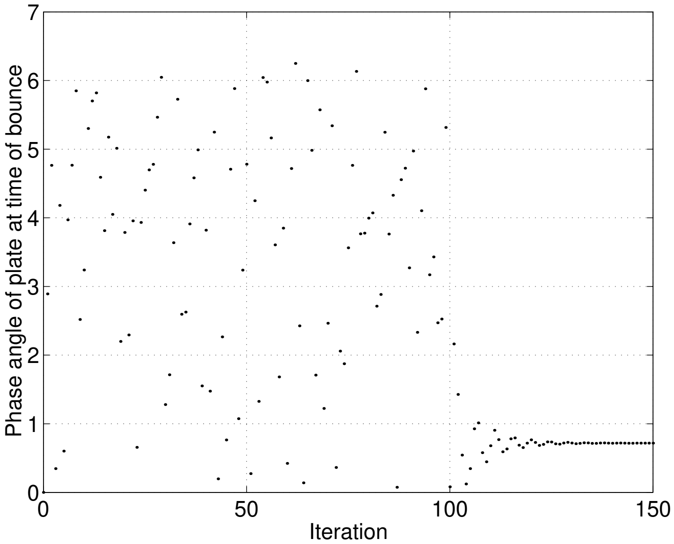


Figure 18.31: A plot of $\phi(k)$ versus time of the controlled modified ball map.

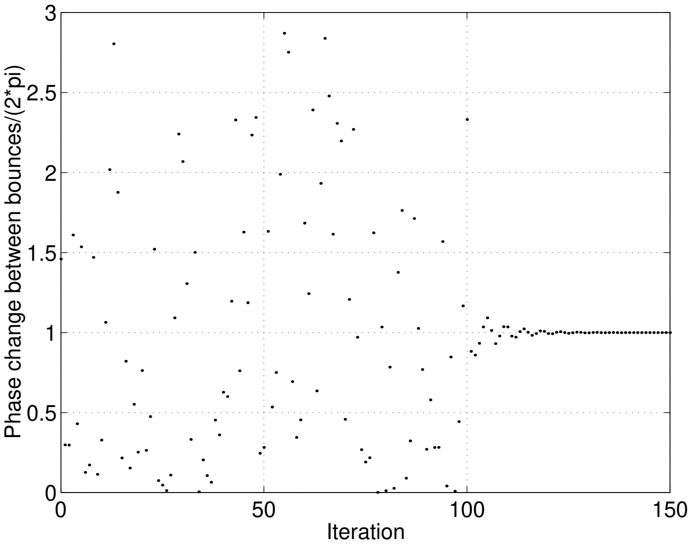


Figure 18.32: A plot of $\psi(k)$ versus time of the controlled modified ball map.

One approach to secure transfer of signals is to hide them in broad spectrum noise-like signals. Chaotic signals are characterized by spread spectrum, where the power of a chaotic signal is spread across a broad range of frequencies. Thus, chaotic signals are particularly suitable for secure communication because the transmitted signal masked by a chaotic signal may appear to a third party as a noise. The concept of synchronized systems, that we define next, plays important role when using chaotic systems in secure communication.

Definition 18.7 *Two dynamical systems,*

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \quad \textit{and} \quad \dot{\tilde{\boldsymbol{x}}} = \boldsymbol{g}(\tilde{\boldsymbol{x}}), \quad \boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathbb{R}^n,$$

are said to be synchronized if

$$\boldsymbol{e}(t) = (\boldsymbol{x}(t) - \tilde{\boldsymbol{x}}(t)) \to \mathbf{0} \quad \textit{as} \quad t \to \infty.$$

◇

Cuomo, Oppenheim, and Strogatz [70] used the concept of synchronized chaotic systems (SCS's) to secure communications with the Lorenz circuit implemented in both the transmitter and receiver. Specifically, a chaotic masking signal was added at the transmitter to the message. Then, at the receiver, the masking was regenerated and subtracted from the received signal.

Synchronization is widely used in electronics, for example, in television, radar, and electronic watches [323]. The general idea behind synchronization is the adaptive generation of a reference signal that is used to estimate the frequency and phase of the received signal. The problem of designing a dynamical system whose behavior mimics that of a given chaotic system is referred to as chaotic synchronization. This is a challenging design problem because chaotic systems are characterized by a sensitive dependence on initial conditions. Thus, one may be tempted to conclude that chaotic synchronization is not feasible because even infinitesimal errors in initial conditions or system parameters will result in the exponential divergence of the systems' trajectories.

Although chaotic systems would seem to defy synchronization, Pecora and Carroll [267, 52] demonstrated that chaotic system synchronization is possible. Morgül and Solak [247] proposed a systematic approach to chaotic system synchronization using observers. Other observer-based approaches to this problem were presented by Grassi and Masscolo [120], Azemi and Yaz [16], Lin et al. [216], and more recently by Jiang, Tang, and Chen [163]. Lian, Chiu, Chaing, and Liu [214] proposed a systematic design methodology using fuzzy observer-based approach.

In this Section, we show how to combine fuzzy observers, chaos, and LMIs to develop systems for secure communication. Specifically, we construct a fuzzy observer-based secure communication system using a chaotic system. The design is accomplished with LMIs. Our exposition is based on the paper by Lian et al. [214].

18.7.1 Fuzzy Representation of Chaotic Systems

In our construction of a secure communication system, we employ a chaotic system represented as a fuzzy system. We use the Lorentz's system described by a system of three differential equations,

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} a(-x_1+x_2) \\ bx_1 - x_2 - x_1x_3 \\ -cx_3 + x_1x_2 \end{bmatrix} \\ y &= x_1, \end{aligned}$$

where $a = 10$, $b = 28$, and $c = 8/3$; see also Exercise 18.1. To design Lorentz's system fuzzy representation, we use x_1 to construct fuzzy rules. We note that $x_1(t) \in [-d,\ d]$, where $d = 30$. We use two fuzzy rules:

$$\text{Rule i}: \quad \text{IF } x_1(t) \text{ is } F_i(x_1(t)) \text{ THEN}$$
$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i\boldsymbol{x}(t) + \boldsymbol{b}_i \\ \boldsymbol{y}(t) = \boldsymbol{c}_i\boldsymbol{x}(t), \quad i = 1, 2, \end{cases}$$

where

$$F_1(x_1(t)) = \frac{1}{2}\left(1 + \frac{x_1(t)}{d}\right) \quad \text{and} \quad F_2(x_1(t)) = \frac{1}{2}\left(1 - \frac{x_1(t)}{d}\right),$$

$$\boldsymbol{A}_1 = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -d \\ 0 & d & -\frac{8}{3} \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & d \\ 0 & -d & -\frac{8}{3} \end{bmatrix},$$

$$\boldsymbol{c}_1 = \boldsymbol{c}_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{b}_1 = \boldsymbol{b}_2 = \boldsymbol{0}.$$

Note that on the interval $[-d, d]$, we have $F_i(x_1(t)) \geq 0$, $i = 1, 2$. In addition,

$$F_i(x_1) + F_2(x_1) = 1.$$

The fuzzy Lorentz system takes the form

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \left(\frac{F_1}{F_1+F_2}\boldsymbol{A}_1 + \frac{F_2}{F_1+F_2}\boldsymbol{A}_2\right)\boldsymbol{x}(t) \\ &= (F_1\boldsymbol{A}_1 + F_2\boldsymbol{A}_2)\,\boldsymbol{x}(t). \end{aligned} \tag{18.121}$$

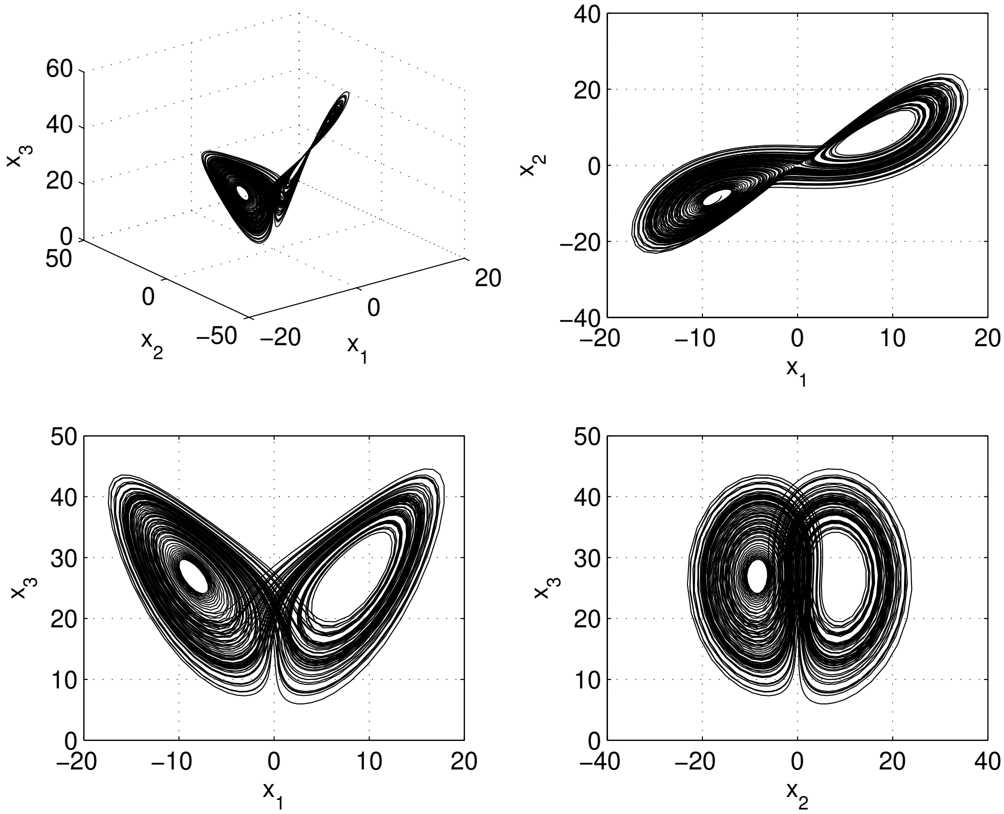


Figure 18.33: Plots of trajectories of the fuzzy Lorentz system (18.121).

In Figure 18.33, we show plots of the fuzzy Lorentz system over the time inerval $[0, 100]$ for the initial condition $\boldsymbol{x}(0) = \begin{bmatrix} -8 & -8 & 24 \end{bmatrix}^\top$.

18.7.2 Constructing Secure Communication System

We first construct a transmitter. Our approach is to hide the message to be transmitted in a noise-like signal being generated by a fuzzy chaotic system of the form

$$\left.\begin{array}{rcl} \dot{\boldsymbol{x}} & = & \sum_{i=1}^r \mu_i \left(\boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{b}_i + \boldsymbol{l}_i m\right) \\ y & = & \boldsymbol{c}\boldsymbol{x} + m, \end{array}\right\} \tag{18.122}$$

where

$$\mu_i = \frac{F_i}{\sum_{i=1}^r F_i},$$

m is the message to be transmitted, and $\boldsymbol{l}_i$, $i = 1, 2, \ldots, r$, are gain vectors selected in such a way that the matrices

$$\boldsymbol{A}_i - \boldsymbol{l}_i \boldsymbol{c}, \quad i = 1, 2, \ldots, r,$$

are asymptotically stable.

The receiver is a fuzzy observer of the transmitter,

$$\left.\begin{array}{rcl}\dot{\tilde{\boldsymbol{x}}} & = & \sum_{i=1}^{r}\mu_i\left(\boldsymbol{A}_i\tilde{\boldsymbol{x}}+\boldsymbol{b}_i+\boldsymbol{l}_i(y-\tilde{y})\right)\\ \tilde{y} & = & \boldsymbol{c}\tilde{\boldsymbol{x}},\end{array}\right\} \tag{18.123}$$

Note that

$$\begin{aligned} y-\tilde{y} &= \boldsymbol{c}\boldsymbol{x}+m-\boldsymbol{c}\tilde{\boldsymbol{x}}\\ &= \boldsymbol{c}(\boldsymbol{x}-\tilde{\boldsymbol{x}})+m.\end{aligned}$$

Let

$$\boldsymbol{e}=\boldsymbol{x}-\tilde{\boldsymbol{x}},$$

be the estimation error of the transmitter's state $\boldsymbol{x}$ by the receiver's state $\tilde{\boldsymbol{x}}$. Then

$$y-\tilde{y}=\boldsymbol{c}\boldsymbol{e}+m \tag{18.124}$$

Thus if

$$\boldsymbol{e}(t)\to 0 \quad \text{as} \quad t\to\infty,$$

then we can recover the message because

$$y-\tilde{y}\to m \quad \text{as} \quad t\to\infty.$$

Our task then is to formulate conditions guaranteeing that $\boldsymbol{e}(t)\to 0 \quad \text{as} \quad t\to\infty$. Let us analyze the dynamics of the estimation error,

$$\begin{aligned}\dot{\boldsymbol{e}} &= \dot{\boldsymbol{x}}-\dot{\tilde{\boldsymbol{x}}}\\ &= \sum_{i=1}^{r}\mu_i\boldsymbol{A}_i\boldsymbol{x}+\sum_{i=1}^{r}\mu_i\boldsymbol{b}_i+\sum_{i=1}^{r}\mu_i\boldsymbol{l}_i m-\sum_{i=1}^{r}\mu_i\boldsymbol{A}_i\tilde{\boldsymbol{x}}-\sum_{i=1}^{r}\mu_i\boldsymbol{b}_i-\sum_{i=1}^{r}\mu_i\boldsymbol{l}_i(y-\tilde{y})\\ &= \sum_{i=1}^{r}\mu_i\boldsymbol{A}_i\boldsymbol{e}+\sum_{i=1}^{r}\mu_i\boldsymbol{l}_i m-\sum_{i=1}^{r}\mu_i\boldsymbol{l}_i(\boldsymbol{c}\boldsymbol{x}+m-\boldsymbol{c}\tilde{\boldsymbol{x}})\\ &= \sum_{i=1}^{r}\mu_i\boldsymbol{A}_i\boldsymbol{e}-\sum_{i=1}^{r}\mu_i\boldsymbol{l}_i\boldsymbol{c}(\boldsymbol{x}-\tilde{\boldsymbol{x}})\\ &= \sum_{i=1}^{r}\mu_i(\boldsymbol{A}_i-\boldsymbol{l}_i\boldsymbol{c})\boldsymbol{e}.\end{aligned} \tag{18.125}$$

We are guaranteed that $\boldsymbol{e}(t)\to 0 \quad \text{as} \quad t\to\infty$ if can select the gains $\boldsymbol{l}_i$, $r=1,2,\ldots,r$ such that there exists a common $\boldsymbol{P}=\boldsymbol{P}^\top>0$ so that

$$(\boldsymbol{A}_i-\boldsymbol{l}_i\boldsymbol{c})^\top\boldsymbol{P}+\boldsymbol{P}(\boldsymbol{A}_i-\boldsymbol{l}_i\boldsymbol{c})<0, \quad r=1,2,\ldots,r. \tag{18.126}$$

The problem of solving simultaneously the above r Lyapunov inequalities can be represented as an LMI feasibility problem. Indeed, performing multiplications in (18.126) yields

$$\boldsymbol{A}_i^\top \boldsymbol{P} - \boldsymbol{c}^\top \boldsymbol{l}_i^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i - \boldsymbol{P}\boldsymbol{l}_i\boldsymbol{c} < 0 \tag{18.127}$$

Let

$$\boldsymbol{M}_i = \boldsymbol{P}\boldsymbol{l}_i.$$

Then we also have that $\boldsymbol{M}_i^\top = \boldsymbol{l}_i^\top \boldsymbol{P}$. Taking the above into account, we represent (18.126) in the form of r LMIs,

$$\boldsymbol{A}_i^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i - (\boldsymbol{c}^\top \boldsymbol{M}_i^\top + \boldsymbol{M}_i\boldsymbol{c}) < 0, \quad r = 1, 2, \ldots, r. \tag{18.128}$$

We summarize the above development in the form of the following proposition that appears in [214, p. 214].

Proposition 18.1 *If there exist $\boldsymbol{P} = \boldsymbol{P}^\top > 0$ and $\boldsymbol{l}_i$, $r = 1, 2, \ldots, r$ such that for $r = 1, 2, \ldots, r$,*

$$\boldsymbol{A}_i^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i - (\boldsymbol{c}^\top \boldsymbol{M}_i^\top + \boldsymbol{M}_i\boldsymbol{c}) < 0, \quad \boldsymbol{M}_i = \boldsymbol{P}\boldsymbol{l}_i,$$

then

$$\boldsymbol{l}_i = \boldsymbol{P}^{-1}\boldsymbol{M}_i, \quad r = 1, 2, \ldots, r$$

and the estimation error $\boldsymbol{e}(t) \to 0$ as $t \to \infty$.

Our fuzzy Lorentz-based secure communication system consists of the transmitter (18.122) and the receiver (18.123). The message to be transmitted is

$$m(t) = 0.01\,\mathrm{square}(2\pi t),$$

where square($\cdot$) generates the square wave. In Figure 18.34 we show the transmitted signal, $y = \boldsymbol{c}\boldsymbol{x} + m$.

In Figure 18.35, we compare the message signal m against its recovered version. As can be seen, after transients decay the recovered signal matches the message signal perfectly.

Finally, in Figure 18.36, we show a state-space trajectory of of the transmitted signal, that is, the signal generated by the transmitter.

We add that the initial condition of the transmitter was $\boldsymbol{x}(0) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top$. We set the zero initial condition for the receiver.

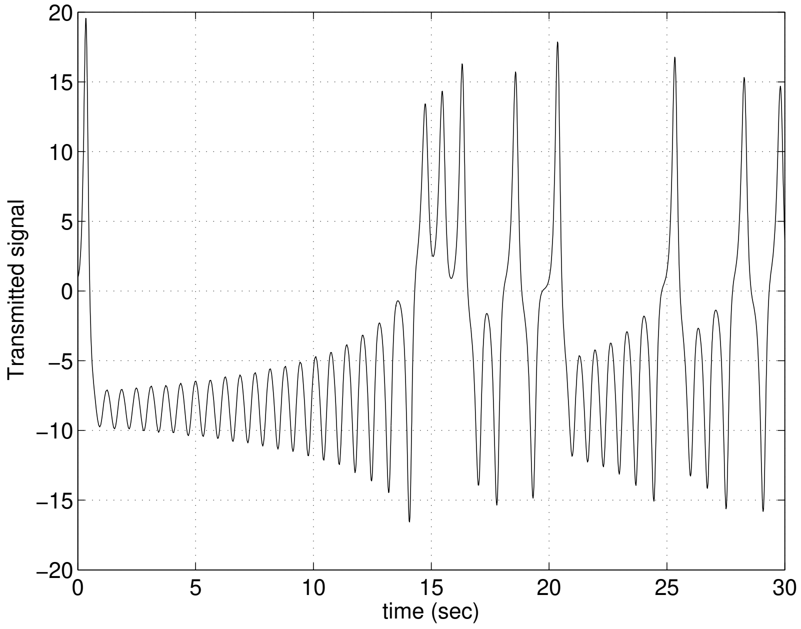


Figure 18.34: Transmitted signal, $\tilde{y}$.

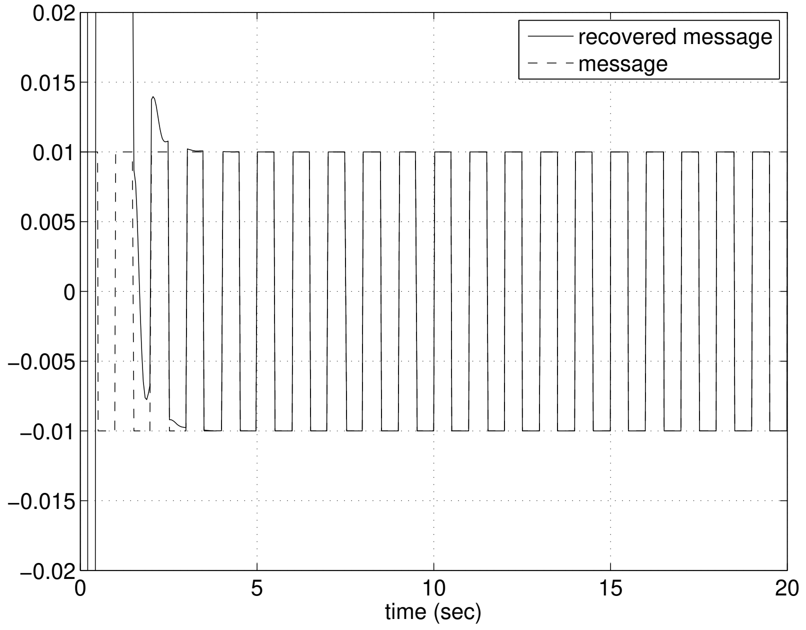


Figure 18.35: Plots of the message signal, m, and its recovered version at the receiver's end obtained from the difference, $y - \tilde{y}$; see (18.124).

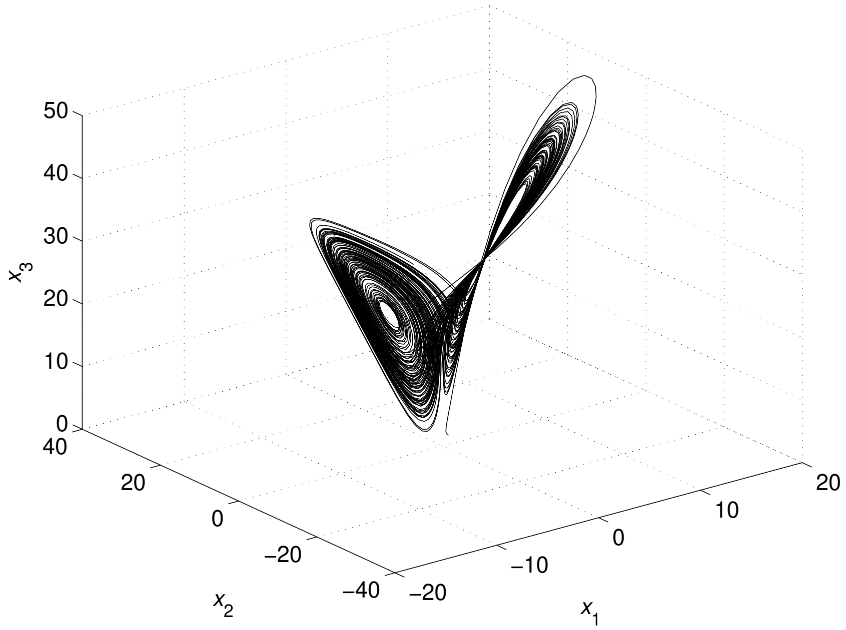


Figure 18.36: State-space trajectory of the transmitter (18.121).

18.8 Notes

Sandefur [293] is a lucid elementary introduction to chaotic discrete-time dynamical systems. Peitgen, Jürgens and Saupe [270] and Alligood, Sauer and Yorke [7] are superb introductions to chaos and dynamical systems. Papers by May [232] and Feigenbaum [100] are recommended for their originality and clarity. Introductory texts to nonlinear dynamics and chaos for scientists and engineers are the books by Moon [246] and Thompson and Stewart [327]. Numerical algorithms for chaotic systems can be found in the book by Parker and Chua [264]. Fractals are covered in Mandelbrot [229], Peitgen and Saupe [271], Devaney and Keen [82], and Bunde and Havlin [47]. Barnsley [24], and Peitgen and Saupe [271] are recommended for their coverage of iterated function systems. A reader who is interested in a mathematical approach to chaotic behavior of dynamical systems should consult the book by Arrowsmith and Place [14]. Peterson [273] gives a remarkable account of the development of models of celestial mechanics leading to the discovery of chaos in the solar system. The use of fractals in education, art, music, fashion, chess and medicine is described in an edited volume by Pickover [274]. Other more recent papers on controlling chaotic systems, in addition to the ones already referenced in the previous section, are: [363], [373], [361]. Applications of chaos in information processing are discussed in the papers by Dmitriev, Panas and Starkov [84] and Andreyev et al. [11]. Aihara, Takabe and Toyoda [5] and Adachi and Aihara [3] discuss the presence of chaos in biological neurons and propose mathematical models of chaotic neural networks. Applications of chaotic neural networks in the complex memory search for patterns with a sought feature are discussed by Nara, Davis and Totsuji [251]; see also [53].

Edward Lorenz was a meteorologist at the Massachusetts Institute of Technology (MIT). His discovery of chaotic behavior of his mathematical model of weather is a fascinating one. As reported by Gleick [115, Chapter 1] and Alligood et al [7, Chapter 9], to speed up his computer output which was simulating a weather model, Lorenz used just three significant digits to print out the results of simulations although the calculations were performed using several more digits of accuracy. One day in the winter of 1961, Lorenz came across a simulation that he decided to repeat. He typed in an initial condition for the new simulation from the printout using the three digit format. He then performed the simulation expecting the same behavior as before of the repeated simulation. Instead, "he saw something unexpected, something that planted a seed for a new science." [115, page 16]. The resulting simulation was totally different from the previous one. The difference between the two solutions was the result of the Lorenz's weather model sensitive dependence on initial conditions. In the computer memory, according to Gleick [115], the initial condition for the first run was 0.506127, while to save space, Lorenz entered 0.506 for the second run. This seemingly inconsequential difference caused the two simulations to flow apart until there was no resemblance at all between them. Later, Lorenz greatly simplified his weather model to just three nonlinear equations that are given in Exercise 18.1. The phase portrait of these famous equations resemble butterfly's wings, which became a symbol for the early activity in chaotic systems.

18.9 Exercises

Exercise 18.1 Simulate the system of differential equations

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} a(-x_1 + x_2) \\ bx_1 - x_2 - x_1x_3 \\ -cx_3 + x_1x_2 \end{bmatrix}, \quad \boldsymbol{x}(0) = \begin{bmatrix} -8 \\ -8 \\ 24 \end{bmatrix}, \tag{18.129}$$

where $a = 10$, $b = 28$, and $c = 8/3$. The set of differential equations (18.129) is commonly referred to as the *Lorenz equations*, after the American meteorologist E. N. Lorenz who first published these equations in 1963. They model fluid flows in the atmosphere. Plot x_2 versus x_1, then x_3 versus x_1, and also x_3 versus x_2. Use MATLAB's command `plot3` to obtain a 3-D plot of the solution of the Lorenz equations.

Exercise 18.2 Simulate the system of the difference equations

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 - x_2(k) - |x_1(k)| \\ x_1(k) \end{bmatrix}, \quad \boldsymbol{x}(0) = \begin{bmatrix} -0.1 \\ 0 \end{bmatrix}. \tag{18.130}$$

The solution "fills" a region that looks like a "gingerbread man". For this reason, Devaney, in [271, page 149], refers to this example of a chaotic set in the plane as a *chaotic gingerbreadman.*

Exercise 18.3 Simulate the system of differential equations

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -x_2 - x_3 \\ x_1 + ax_2 \\ b + x_3(x_1 - c) \end{bmatrix}, \tag{18.131}$$

where $a = 0.398$, $b = 2$, and $c = 4$. The set of differential equations (18.131) is referred to as the *Rössler equations*, after O. E. Rössler who reported on their chaotic behavior in 1976. Rössler called the chaotic attractor of the above system the "spiral chaos". This type of chaotic attractor is also called a *folded band.* Select an initial condition, integrate the equations, then plot x_2 versus x_1, x_3 versus x_1, and x_3 versus x_2. Use MATLAB's command `plot3` to obtain a 3-D plot of the folded band attractor.

Exercise 18.4 For the dynamical system

$$x(k+1) = x(k) + r\left(0.8 - \operatorname{atan}(x(k))\right), \tag{18.132}$$

plot a bifurcation diagram for the parameter r in the interval $[3, 12]$. The map (18.132) was analyzed in [150].

Exercise 18.5 For the dynamical system

$$x(k+1) = x(k) + r\left(0.8 - \frac{1 - e^{-x(k)}}{1 + e^{-x(k)}}\right), \tag{18.133}$$

plot a bifurcation diagram for the parameter r in the interval $[10, 55]$.

Exercise 18.6 For the dynamical system

$$x(k+1) = x(k) + r\left(0.8 - \frac{1}{1 + e^{-x(k)}}\right), \tag{18.134}$$

plot a bifurcation diagram for the parameter r in the interval $[10, 45]$.

Exercise 18.7 Write a MATLAB program of the following "chaos game": Select three points that are vertices of an equilateral triangle. Choose at random one of the vertices as a starting point. Then, choose at random another vertex and move halfway to it from the starting vertex. Mark this halfway point with a dot. This halfway point is your new starting point. Choose at random one of the vertices and move halfway to the chosen vertex from the new starting point. Mark the new halfway point with the dot. Repeat the process. The outcome of this chaos game is a fractal called the Sierpiński gasket. Generate a plot of the Sierpiński gasket. Other methods of constructing the Sierpiński gasket are described in [270] and in Exercise 18.8. The term "chaos game" was coined by M. F. Barnsley, the author of [24].

Exercise 18.8 The result of the chaos game described in Exercise 18.7 is the Sierpiński gasket that can also be constructed using the following procedure. Start with an equilateral triangle. Divide it into four equal sub-triangles, quarters, and throw away the middle one, upside-down, triangle. This step leaves three equilateral triangles half the size of the original triangle. In the second step, divide each of the remaining three triangles into four equal quarter-triangles and throw away the middle, upside-down, triangle in each of the three triangles. This step leaves nine triangles one-quarter the size of the original triangle. Repeat the process. The remaining area as the number of iterations tends to infinity is called the Sierpiński gasket.

(i) Show that the area of the Sierpiński gasket is zero.

(ii) Determine the fractal dimension of the Sierpiński gasket.

Exercise 18.9 In this exercise, we concern ourselves with another fractal, the Sierpiński carpet. Start with a square. Divide it into nine subsquares, each one-third the size of the original square. Remove the central square. Repeat the procedure. The remaining area as the number of iterations goes to infinity is called the Sierpiński carpet.

(i) Show that the area of the Sierpiński carpet is zero.

(ii) Determine the fractal dimension of the Sierpiński carpet.

Exercise 18.10 For the double pendulum system described in Subsection 2.4.6 set the input torques to zero and simulate the system behavior starting from the initial state $\boldsymbol{x}(0) = \begin{bmatrix} 0 & 0 & \pi & .001 \end{bmatrix}^\top$. Verify that one of the equilibrium states corresponding to zero input torques is $\boldsymbol{x}_e = \begin{bmatrix} 0 & 0 & \pi & 0 \end{bmatrix}^\top$. Linearize the state-space model about this equilibrium point. Then, use the LQR approach to compute the state feedback matrix $\boldsymbol{K}$ for the weight matrices

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad \text{and} \quad \boldsymbol{R} = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}.$$

Solve the Lyapunov equation

$$(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})^\top \boldsymbol{P}_l + \boldsymbol{P}_l(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}) = -\boldsymbol{Q}.$$

Suppose that the bound on the input torques are: $|T_1| \leq 5$ and $|T_2| \leq 2$. Use the method of Example 18.5 to find $V_{\max}$ and apply the chaotic control law to stabilize the closed-loop system with respect to $\begin{bmatrix} 0 & 0 & \pi & 0 \end{bmatrix}^\top$. Plot x_1, x_2, and $V = (\boldsymbol{x} - \boldsymbol{x}_e)^\top \boldsymbol{P}_l(\boldsymbol{x} - \boldsymbol{x}_e)$ versus time.

Appendix A

Math Review

> Kroemers Lemma of New Technology: “The principal applications of any sufficiently new and innovative technology always have been and will continue to be applications created by that technology.”
> In other words: The principal applications have not been of the kind where one improves on something that could already be done on an acceptable level; they have been applications that could previously not been met at all or were not even recognized.
> As a rule, such applications have indeed arisen, although usually not immediately. The problem with truly new applications is that they are extraordinarily hard to predict, and were historically rarely predicted. Ultimately, progress in applications is not deterministic, but opportunistic, exploiting for new applications whatever new science and technology happen to be coming along.—H. Kroemer [194, p. 958]

> “The last word, when all is heard: Fear God and keep his commandments, for this is man’s all; because God will bring to judgment every work, with all its hidden qualities, whether good or bad.”
> *Ecclesiastes 12:13–14*

The appendix reviews mathematical techniques that are used in the book. The appendix also serves as a glossary of symbols, notation, and concepts employed in the text. For more information on the topics covered here, the reader should consult references supplied at the end of the appendix.

In an attempt to keep the presentation as clear as possible, we employ the theorem-proof style whenever appropriate. Major results of independent interest are encapsulated as theorems. A lemma is used to denote an auxiliary result that is used as a stepping stone towards a theorem. We use the term proposition as a way to announce a minor theorem. A corollary is a direct consequence of a theorem or proposition. A hypothesis is a statement that is taken for further reasoning in a proof. For more information on the subject of writing for mathematical or technical sciences, we recommend a very clear and useful book by Higham [137].

A.1 Notation and Methods of Proof

We often use the term set. The great German mathematician G. Cantor (1845–1918), the creator of the theory of sets, defines a set as any collection of definite, distinguishable objects of our thought, to be conceived as a whole. [227, page 1] The objects, in the definition, are the elements or members of the set. Usually, we use capital letters to denote sets and lower case letters to denote elements of the set. Thus, if X is a set, then we write $x \in X$ to mean that x is an element of X. When an object x is not an element of a set X, we write $x \notin X$. Often, we use the curly bracket notation for sets. For example, $\{x : x < 7\}$ denotes the set of all x less than 7. The colon, :, following x is read "such that". The following sets are used in the book:

$\mathbb{N} = \{1, 2, 3, \ldots\}$, the set of all positive integers,
$\mathbb{Z} = \{0, 1, -1, 2, -2, \ldots\}$, the set of all integers,
$\mathbb{R}$, the set of all real numbers,
$\mathbb{C}$, the set of all complex numbers.

Sometimes, we take a fixed set, and then carry out manipulations with reference to the fixed set. We call this fixed set the *universe of discourse*, or just the *universe*, for short.

Throughout the text, we work with statements that are combinations of individual statements. It is necessary to distinguish between sentences and the statements. A sentence is a part of a language. Sentences are used to make different statements. In classical logic, statements are either true or false. Consider the following two statements:

$A : x > 7;$

$B : x^2 > 49.$

We can combine the above two statements into one statement, called a *conditional*, that has the form: "IF A THEN B". Thus, a conditional is a compound statement obtained by placing the word "IF" before the first statement and inserting the word "THEN" before the second statement. We use the symbol "$\Rightarrow$" to represent the conditional "IF A THEN B" as $A \Rightarrow B$. The statement $A \Rightarrow B$ also reads as "A implies B", or "A only if B", or "A is sufficient for B", or "B is necessary for A". Statements A and B may either be true or false. The relationship between the truth or falsity of A and B and the conditional $A \Rightarrow B$ can be illustrated by means of a diagram called a *truth table* shown in Table A.1. In the table, T stands for "true", and F stands for "false". It is intuitively clear that if A is true,

Table A.1: Truth table for conditional.

A	B	$A \Rightarrow B$
F	F	T
F	T	T
T	F	F
T	T	T

then B must also be true for the statement $A \Rightarrow B$ to be true. If A is not true, then the sentence $A \Rightarrow B$ does not have an obvious meaning in everyday language. We can interpret $A \Rightarrow B$ to mean that we cannot have A true and B not true. One can check that the truth values of the statements $A \Rightarrow B$ and "not (A and not B)" are the same. We can also check that the truth values of $A \Rightarrow B$ and "not $B \Rightarrow$ not A" are the same. The above is the basis for three methods, inference processes, of proving statements of the form $A \Rightarrow B$. *Inference* is a process by which one statement is arrived at, using reasoning, on the basis of one or more other statements accepted as a starting point. *Reasoning* is a special kind of thinking in which inference takes place using the laws of thought. By a *proof* we mean the process of establishing the truth of a statement by arriving at it from other statements using principles of reasoning. Most of the time we are concerned with the statements of the form $A \Rightarrow B$. We use the following types of proofs of such statements:

(i) the direct method;

(ii) the proof by contraposition;

(iii) the proof by contradiction.

In the first case, the direct method, one starts with A, then tries to deduce B. In the proof by contraposition, one starts with "not B", then deduces a chain of consequences to end with "not A". This method of proving is based on the fact that the truth values of $A \Rightarrow B$ and "not $B \Rightarrow$ not A" are the same. The proof by contradiction is based on the fact that the truth values of $A \Rightarrow B$ and "not (A and not B)" are the same. In this method, one starts with "A and not B" and tries to arrive at a contradiction, which is a statement that is always false.

It is possible to combine two conditionals to form a *biconditional*, which is a statement of the form "A if and only if B". Occasionally, we use the symbol "$\Leftrightarrow$" to write $A \Leftrightarrow B$ to mean "A if and only if B". Another way of expressing $A \Leftrightarrow B$ is "A is equivalent to B", or "A is necessary and sufficient for B".

A.2 Vectors

We define an n-dimensional column vector to be an n-tuple of numbers

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \tag{A.1}$$

where n is a positive integer. Thus a vector can be viewed as an ordered set of scalars. We use lower-case bold letters, for example $\boldsymbol{x}$, to denote such n-tuples. We call the numbers $x_1, x_2, \ldots, x_n$ the *coordinates* or *components* of the vector $\boldsymbol{x}$. The number of components of a vector is its dimension. When referring to a particular component of a vector, we use its component, for example x_i, or the subscripted name of the vector, for example $(\boldsymbol{x})_i$ to denote x_i. The set of all n-tuples, that is n-vectors, with real coefficients will be denoted $\mathbb{R}^n$ and called the real n-space. The transpose of the vector $\boldsymbol{x}$, denoted $\boldsymbol{x}^\top$, is an n-dimensional row vector

$$\boldsymbol{x}^\top = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^\top = \begin{bmatrix} x_1 & x_2 \cdots x_n \end{bmatrix}. \tag{A.2}$$

The sum of two n-vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is an n-vector, denoted $\boldsymbol{x}+\boldsymbol{y}$, whose i-th component is $x_i + y_i$. The following rules are satisfied:

1. $(\boldsymbol{x}+\boldsymbol{y})+\boldsymbol{z} = \boldsymbol{x}+(\boldsymbol{y}+\boldsymbol{z})$, that is, vector addition is associative.

2. $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x}$, which means that vector addition is commutative.

3. Let $\mathbf{0} = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^\top$ denote the zero vector. Then,
$$\mathbf{0} + \boldsymbol{x} = \boldsymbol{x} + \mathbf{0} = \boldsymbol{x}.$$

4. Let $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top$ and let $-\boldsymbol{x} = \begin{bmatrix} -x_1 & -x_2 & \cdots & -x_n \end{bmatrix}^\top$. Then,
$$\boldsymbol{x} + (-\boldsymbol{x}) = \mathbf{0}.$$

The product of an n-vector $\boldsymbol{x}$ by a scalar c is an n-th vector, denoted $c\boldsymbol{x}$, whose i-th component is cx_i.

For two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ from $\mathbb{R}^n$, we define their *scalar product*, also called the *inner product*, to be
$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n. \tag{A.3}$$
The scalar product is commutative,
$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle,$$
and distributivity over vector addition holds,
$$\langle \boldsymbol{x}, \boldsymbol{y} + \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{x}, \boldsymbol{z} \rangle.$$
Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are said to be *orthogonal* if $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$.

A *vector norm*, denoted $\|\cdot\|$, satisfies the following properties:

VN 1. $\|\boldsymbol{x}\| > 0$ for any nonzero vector.

VN 2. For any scalar c, $\|c\boldsymbol{x}\| = |c|\|\boldsymbol{x}\|$.

VN 3. For any two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$—*the triangle inequality.*

The *Euclidean norm*, or the *2-norm*, of a vector $\boldsymbol{x} \in \mathbb{R}^n$, denoted $\|\boldsymbol{x}\|_2$, is defined as
$$\|\boldsymbol{x}\|_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}. \tag{A.4}$$
Note that $\|\boldsymbol{x}\|_2^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \boldsymbol{x}^\top \boldsymbol{x}$. The Euclidean norm is a special class of norms called the *Hölder* or *p-norms* defined by
$$\|\boldsymbol{x}\|_p = \left(|x_1|^p + \cdots + |x_n|^p\right)^{1/p}, \quad p \geq 1. \tag{A.5}$$

Other often used norms that are special cases of the above class of norms are the *1-norm* and the *infinity norm* defined respectively as

$$\|\boldsymbol{x}\|_1 = |x_1| + \cdots + |x_n| \tag{A.6}$$

and

$$\|\boldsymbol{x}\|_\infty = \max_i |x_i|. \tag{A.7}$$

Most of the time, we use the Euclidean norm for vectors. So for the sake of brevity, we write $\|\boldsymbol{x}\|$ to mean $\|\boldsymbol{x}\|_2$.

Theorem A.1 (Pythagoras) *If $\boldsymbol{x}$ and $\boldsymbol{y}$ are orthogonal, then*

$$\|\boldsymbol{x}+\boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2.$$

Proof We have,

$$\begin{aligned}\|\boldsymbol{x}+\boldsymbol{y}\|^2 &= (\boldsymbol{x}+\boldsymbol{y})^\top(\boldsymbol{x}+\boldsymbol{y})\\ &= \|\boldsymbol{x}\|^2 + 2\langle\boldsymbol{x},\boldsymbol{y}\rangle + \|\boldsymbol{y}\|^2\\ &= \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2,\end{aligned}$$

since by assumption $\langle\boldsymbol{x},\boldsymbol{y}\rangle = 0$. The proof of the theorem is complete.

□

We now introduce the notion of *projection.* Let $\boldsymbol{x}$, $\boldsymbol{y}$ be two vectors, where $\boldsymbol{y} \neq \boldsymbol{0}$. Let $\boldsymbol{z} = c\boldsymbol{y}$, $c \in \mathbb{R}$, be the vector such that the vector $\boldsymbol{x} - \boldsymbol{z}$ is orthogonal to $\boldsymbol{y}$—see Figure A.1 for an illustration of the above construction. Thus,

$$(\boldsymbol{x}-\boldsymbol{z})^\top \boldsymbol{y} = 0. \tag{A.8}$$

Substituting $\boldsymbol{z} = c\boldsymbol{y}$ into (A.8) yields

$$(\boldsymbol{x}-\boldsymbol{z})^\top \boldsymbol{y} = (\boldsymbol{x}-c\boldsymbol{y})^\top \boldsymbol{y} = 0. \tag{A.9}$$

Hence,

$$c = \frac{\langle\boldsymbol{x},\boldsymbol{y}\rangle}{\|\boldsymbol{y}\|^2}. \tag{A.10}$$

The number c given by (A.10) is called the component of $\boldsymbol{x}$ along $\boldsymbol{y}$. The projection of $\boldsymbol{x}$ along $\boldsymbol{y}$ is the vector

$$\boldsymbol{z} = c\boldsymbol{y} = \frac{\langle\boldsymbol{x},\boldsymbol{y}\rangle}{\|\boldsymbol{y}\|^2}\boldsymbol{y}. \tag{A.11}$$

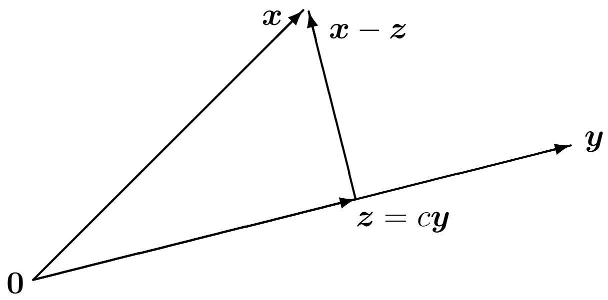


Figure A.1: Computing the projection of $\boldsymbol{x}$ along $\boldsymbol{y}$.

Theorem A.2 (Cauchy-Schwarz Inequality) *Let* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$*. Then,*

$$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \leq \|\boldsymbol{x}\| \|\boldsymbol{y}\|. \tag{A.12}$$

Proof If $\boldsymbol{y} = \mathbf{0}$, then both sides of (A.12) are equal to 0, and the assertion holds. Suppose that $\boldsymbol{y} \neq \mathbf{0}$, and let c denote the component of $\boldsymbol{x}$ along $\boldsymbol{y}$. Hence, c is given by (A.10). We then write

$$\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{z} + \boldsymbol{z} = \boldsymbol{x} - c\boldsymbol{y} + c\boldsymbol{y}. \tag{A.13}$$

For an illustration of (A.13), consult Figure A.1. Applying the Pythagoras theorem to (A.13), we obtain

$$\|\boldsymbol{x}\|^2 = \|\boldsymbol{x} - c\boldsymbol{y}\|^2 + \|c\boldsymbol{y}\|^2 = \|\boldsymbol{x} - c\boldsymbol{y}\|^2 + c^2\|\boldsymbol{y}\|^2. \tag{A.14}$$

Hence,

$$c^2\|\boldsymbol{y}\|^2 \leq \|\boldsymbol{x}\|^2. \tag{A.15}$$

On the other hand, by (A.10),

$$c^2\|\boldsymbol{y}\|^2 = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2}{\|\boldsymbol{y}\|^4}\|\boldsymbol{y}\|^2 = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2}{\|\boldsymbol{y}\|^2}. \tag{A.16}$$

Combining (A.15) and (A.16) yields

$$\frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2}{\|\boldsymbol{y}\|^2} \leq \|\boldsymbol{x}\|^2. \tag{A.17}$$

Multiplying both sides of (A.17) by $\|\boldsymbol{y}\|^2$ and taking the square root, concludes the proof.

□

Theorem A.3 (Triangle inequality) *Let $\boldsymbol{x}$, $\boldsymbol{y}$ be two vectors in $\mathbb{R}^n$. Then,*

$$\|\boldsymbol{x}+\boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|. \tag{A.18}$$

Proof Both sides of (A.18) are nonnegative. Hence, to prove that (A.18) holds it suffices to show that the squares of the both sides satisfy the desired inequality,

$$(\boldsymbol{x}+\boldsymbol{y})^\top(\boldsymbol{x}+\boldsymbol{y}) \leq (\|\boldsymbol{x}\|+\|\boldsymbol{y}\|)^2. \tag{A.19}$$

Expanding $(\boldsymbol{x}+\boldsymbol{y})^\top(\boldsymbol{x}+\boldsymbol{y})$ and applying the Cauchy-Schwarz inequality yields

$$\begin{aligned}(\boldsymbol{x}+\boldsymbol{y})^\top(\boldsymbol{x}+\boldsymbol{y}) &= \|\boldsymbol{x}\|^2 + 2\langle\boldsymbol{x},\boldsymbol{y}\rangle + \|\boldsymbol{y}\|^2 \\ &\leq |\boldsymbol{x}\|^2 + 2\|\boldsymbol{x}\|\|\boldsymbol{y}\| + \|\boldsymbol{y}\|^2 \\ &= (\|\boldsymbol{x}\|+\|\boldsymbol{y}\|)^2, \end{aligned} \tag{A.20}$$

which completes the proof of the theorem.

□

Theorem A.4 *Let $\boldsymbol{x}$, $\boldsymbol{y}$ be two vectors in $\mathbb{R}^n$. Then,*

$$|\|\boldsymbol{x}\| - \|\boldsymbol{y}\|| \leq \|\boldsymbol{x}-\boldsymbol{y}\|.$$

Proof We have

$$\|\boldsymbol{x}\| = \|(\boldsymbol{x}-\boldsymbol{y})+\boldsymbol{y}\| \leq \|\boldsymbol{x}-\boldsymbol{y}\| + \|\boldsymbol{y}\|.$$

Hence,

$$\|\boldsymbol{x}\| - \|\boldsymbol{y}\| \leq \|\boldsymbol{x}-\boldsymbol{y}\|. \tag{A.21}$$

The above implies that

$$\|\boldsymbol{x}-\boldsymbol{y}\| = \|\boldsymbol{y}-\boldsymbol{x}\| \geq \|\boldsymbol{y}\| - \|\boldsymbol{x}\| = -(\|\boldsymbol{x}\|-\|\boldsymbol{y}\|). \tag{A.22}$$

Combining (A.21) and (A.22) yields

$$|\|\boldsymbol{x}\| - \|\boldsymbol{y}\|| \leq \|\boldsymbol{x}-\boldsymbol{y}\|,$$

thus completing the proof.

□

A.3 Matrices and Determinants

Let m, n be two integers such that $m, n \geq 1$. An array of numbers

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

is called a *matrix*. Thus a matrix is a set of scalars subject to two orderings. We say that it is an m by n matrix, or $m \times n$ matrix. This matrix has m rows and n columns. If $m = n$, then we say that it is a square matrix. An $m \times n$ matrix $\boldsymbol{A}$ whose elements are real numbers will be denoted as $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. If the elements of the matrix $\boldsymbol{A}$ are complex numbers, then we write $\boldsymbol{A} \in \mathbb{C}^{m \times n}$. We define the zero matrix, denoted $\boldsymbol{O}$, to be the matrix such that $a_{ij} = 0$ for all i, j. We define the $n \times n$ identity matrix, denoted $\boldsymbol{I}_n$, as the matrix having diagonal components all equal to 1 and all other components all equal to 0.

Given an $m \times n$ matrix $\boldsymbol{A}$. The $n \times m$ matrix $\boldsymbol{B}$ such that $b_{ji} = a_{ij}$ is called the transpose of $\boldsymbol{A}$ and is denoted $\boldsymbol{A}^\top$. Thus, the transpose of a matrix is obtained by changing rows into columns and vice versa. A matrix $\boldsymbol{A}$ which is equal to its transpose, that is, $\boldsymbol{A} = \boldsymbol{A}^\top$, is called symmetric.

We now define the sum of two matrices. Given two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same size. We define $\boldsymbol{A} + \boldsymbol{B}$ to be the matrix whose elements are $a_{ij} + b_{ij}$, that is, we add matrices of the same size componentwise.

Let c be a number and $\boldsymbol{A}$ be a matrix. We define $c\boldsymbol{A}$ to be the matrix whose components are ca_{ij}. Thus, to obtain $c\boldsymbol{A}$, we multiply each component of $\boldsymbol{A}$ by c.

We now define the product of two matrices. Let $\boldsymbol{A}$ be an $m \times n$ matrix and $\boldsymbol{B}$ be an $n \times s$ matrix. Let $\boldsymbol{a}_i^\top$ denote the i-th row of $\boldsymbol{A}$, and $\boldsymbol{b}_j$ the j-th column of $\boldsymbol{B}$, Then, the product of $\boldsymbol{AB}$ is an $m \times s$ matrix $\boldsymbol{C}$ defined as

$$\boldsymbol{C} = \boldsymbol{AB} = \begin{bmatrix} \boldsymbol{a}_1^\top \boldsymbol{b}_1 & \boldsymbol{a}_1^\top \boldsymbol{b}_2 & \cdots & \boldsymbol{a}_1^\top \boldsymbol{b}_s \\ \boldsymbol{a}_2^\top \boldsymbol{b}_1 & \boldsymbol{a}_2^\top \boldsymbol{b}_2 & \cdots & \boldsymbol{a}_2^\top \boldsymbol{b}_s \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_m^\top \boldsymbol{b}_1 & \boldsymbol{a}_m^\top \boldsymbol{b}_2 & \cdots & \boldsymbol{a}_m^\top \boldsymbol{b}_s \end{bmatrix}. \tag{A.23}$$

It is not always true that $\boldsymbol{AB} = \boldsymbol{BA}$, that is, matrix multiplication is not necessarily commutative.

Let $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ be matrices such that $\boldsymbol{A}$, $\boldsymbol{B}$ can be multiplied, $\boldsymbol{B}$, $\boldsymbol{C}$ can be multiplied, and $\boldsymbol{B}$, $\boldsymbol{C}$ can be added. Then,

$$\boldsymbol{A}(\boldsymbol{B}+\boldsymbol{C}) = \boldsymbol{AB}+\boldsymbol{AC}. \tag{A.24}$$

The above property is called the distributive law. If x is a number, then

$$\boldsymbol{A}(x\boldsymbol{B}) = x(\boldsymbol{AB}). \tag{A.25}$$

Furthermore, matrix multiplication is associative because

$$(\boldsymbol{AB})\boldsymbol{C} = \boldsymbol{A}(\boldsymbol{BC}). \tag{A.26}$$

Suppose now that $\boldsymbol{A}$, $\boldsymbol{B}$ can be multiplied. Then,

$$(\boldsymbol{AB})^\top = \boldsymbol{B}^\top\boldsymbol{A}^\top. \tag{A.27}$$

Thus the transpose of a product is the product of transposed matrices in reverse order.

Let $\boldsymbol{A}$ be an $n\times n$ matrix. An inverse of $\boldsymbol{A}$, if it exists, is an $n\times n$ matrix $\boldsymbol{B}$ such that

$$\boldsymbol{AB} = \boldsymbol{BA} = \boldsymbol{I}_n. \tag{A.28}$$

If an inverse exists, then it is unique. A square matrix that is invertible is said to be *nonsingular*. We denote the inverse of $\boldsymbol{A}$ by $\boldsymbol{A}^{-1}$. Hence, we have

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}_n.$$

The transpose of an inverse is the inverse of the transpose,

$$\left(\boldsymbol{A}^{-1}\right)^\top = \left(\boldsymbol{A}^\top\right)^{-1}, \tag{A.29}$$

and we denote it $\boldsymbol{A}^{-T}$.

We now present the Sherman-Morrison-Woodbury formula for inverting a special kind of matrices.

Lemma A.1 *If $\boldsymbol{A}$, $\boldsymbol{D}$, and $\boldsymbol{D}^{-1}+\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}$ are invertible, then*

$$(\boldsymbol{A}+\boldsymbol{BDC})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{D}^{-1}+\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{C}\boldsymbol{A}^{-1}.$$

Proof We prove the Sherman-Morrison-Woodbury formula by verification. We have

$$
\begin{aligned}
&(\boldsymbol{A}+\boldsymbol{BDC})\left(\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{D}^{-1}+\boldsymbol{CA}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{CA}^{-1}\right)\\
&\quad=(\boldsymbol{A}+\boldsymbol{BDC})\,\boldsymbol{A}^{-1}\\
&\qquad-(\boldsymbol{A}+\boldsymbol{BDC})\,\boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{D}^{-1}+\boldsymbol{CA}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{CA}^{-1}\\
&\quad=\boldsymbol{I}+\boldsymbol{BDCA}^{-1}\\
&\qquad-\left(\boldsymbol{B}+\boldsymbol{BDCA}^{-1}\boldsymbol{B}\right)\left(\boldsymbol{D}^{-1}+\boldsymbol{CA}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{CA}^{-1}\\
&\quad=\boldsymbol{I}+\boldsymbol{BDCA}^{-1}\\
&\qquad-\boldsymbol{BD}\left(\boldsymbol{D}^{-1}+\boldsymbol{CA}^{-1}\boldsymbol{B}\right)\left(\boldsymbol{D}^{-1}+\boldsymbol{CA}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{CA}^{-1}\\
&\quad=\boldsymbol{I}+\boldsymbol{BDCA}^{-1}-\boldsymbol{BDCA}^{-1}\\
&\quad=\boldsymbol{I},
\end{aligned}
$$

as desired.

□

In the case when $\boldsymbol{D}=\boldsymbol{I}$, the Sherman-Morrison-Woodbury formula takes the form

$$(\boldsymbol{A}+\boldsymbol{BC})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{I}+\boldsymbol{CA}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{CA}^{-1}. \tag{A.30}$$

Formula (A.30) is sometimes referred to as the *matrix inversion formula.*

A set of nonzero vectors $\{\boldsymbol{q}_1,\boldsymbol{q}_2,\ldots,\boldsymbol{q}_r\}$ is said to be orthogonal if

$$\langle\boldsymbol{q}_i,\boldsymbol{q}_j\rangle=0,\quad \text{for}\quad i\neq j.$$

If each vector in an orthogonal set is of unity length, that is, $\|\boldsymbol{q}_i\|=1$ for all $i=1,2\ldots,r$, then the set of vectors is also called *orthonormal.* A square matrix with orthogonal columns is called an *orthogonal matrix.* Sometimes in the literature, an orthogonal matrix is called an *unitary matrix.* If $\boldsymbol{Q}\in\mathbb{R}^{n\times n}$ is an orthogonal matrix, then

$$\boldsymbol{Q}^\top\boldsymbol{Q}=\boldsymbol{Q}\boldsymbol{Q}^\top=\boldsymbol{I}_n. \tag{A.31}$$

For a square, $n\times n$, matrix $\boldsymbol{A}$, its trace, denoted trace$\boldsymbol{A}$, is the sum of its diagonal elements,

$$\operatorname{trace}\boldsymbol{A}=\sum_{i=1}^n a_{ii}. \tag{A.32}$$

If $\boldsymbol{B}$ is also an $n\times n$ matrix, then

$$\operatorname{trace}(\boldsymbol{AB})=\operatorname{trace}(\boldsymbol{BA}). \tag{A.33}$$

The *determinant* of a square matrix $\boldsymbol{A}$, denoted $\det\boldsymbol{A}$, can be evaluated using the Laplace expansion as follows. Let c_{ij} denote the cofactor corresponding to the element a_{ij} of an $n\times n$ matrix $\boldsymbol{A}$. The cofactor c_{ij} is $(-1)^{i+j}$ times the determinant of the $(n-1)\times(n-1)$ submatrix of $\boldsymbol{A}$ that is obtained by deleting the i-th row and j-th column of $\boldsymbol{A}$. Then, for any fixed i, $1\le i\le n$, we have

$$\det\boldsymbol{A}=\sum_{j=1}^{n}a_{ij}c_{ij}. \tag{A.34}$$

This is the expansion of the determinant along the i-th row. In a similar fashion, we can expand the determinant along a column. The determinant has the following properties:

D 1. $\det\boldsymbol{A}=\det\boldsymbol{A}^{\top}$

D 2. Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be square matrices of the same size, then $\det(\boldsymbol{AB})=\det\boldsymbol{A}\det\boldsymbol{B}=\det(\boldsymbol{BA})$

D 3. $\det\boldsymbol{O}=0$

D 4. $\det\boldsymbol{I}_n=1$.

We now list additional properties of determinants. Let $\boldsymbol{a}_i$ denote the i-th column of $\boldsymbol{A}$. Then, we write

$$\det\boldsymbol{A}=\det(\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\boldsymbol{a}_n).$$

D 5. Suppose that the i-th column $\boldsymbol{a}_i$ is written as a sum

$$\boldsymbol{a}_i=\boldsymbol{b}+\boldsymbol{c}.$$

Then, we have

$$\begin{aligned}\det(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_i,\ldots,\boldsymbol{a}_n)&=\det(\boldsymbol{a}_1,\ldots,\boldsymbol{b}+\boldsymbol{c},\ldots,\boldsymbol{a}_n)\\&=\det(\boldsymbol{a}_1,\ldots,\boldsymbol{b},\ldots,\boldsymbol{a}_n)+\det(\boldsymbol{a}_1,\ldots,\boldsymbol{c},\ldots,\boldsymbol{a}_n).\end{aligned}$$

D 6. If x is a number, then

$$\det(\boldsymbol{a}_1,\ldots,x\,\boldsymbol{a}_i,\ldots,\boldsymbol{a}_n)=x\det(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_i,\ldots,\boldsymbol{a}_n).$$

D 7. If two columns of the matrix are equal, then the determinant is equal to zero.

D 8. If we add a multiple of one column to another, then the value of the determinant does not change. In other words, if x is a number, then

$$\det(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_i,\ldots,\boldsymbol{a}_j,\ldots,\boldsymbol{a}_n)=\det(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_i+x\boldsymbol{a}_j,\ldots,\boldsymbol{a}_j,\ldots,\boldsymbol{a}_n).$$

Properties **D 5**–**D 8** hold also for rows instead of columns because $\det \boldsymbol{A} = \det \boldsymbol{A}^\top$.

Using the cofactors of $\boldsymbol{A}$ and its determinant, we can give a formula for $\boldsymbol{A}^{-1}$. First, we define the adjugate of $\boldsymbol{A}$, denoted adj$\boldsymbol{A}$ to be the matrix whose (i,j)-th entry is the cofactor c_{ji}, that is, adj$\boldsymbol{A}$ is the transpose of the cofactor matrix of $\boldsymbol{A}$. Then,

$$\boldsymbol{A}^{-1} = \left(\frac{1}{\det \boldsymbol{A}}\right) \mathrm{adj}\boldsymbol{A} = \frac{\mathrm{adj}\boldsymbol{A}}{\det \boldsymbol{A}}. \tag{A.35}$$

If $\boldsymbol{A}$ is $n \times n$ and $\boldsymbol{v}$ is a nonzero $n \times 1$ vector such that for some scalar λ,

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}, \tag{A.36}$$

then $\boldsymbol{v}$ is an *eigenvector* corresponding to the *eigenvalue* λ. In order that λ be an eigenvalue of $\boldsymbol{A}$, it is necessary and sufficient for the matrix $\lambda\boldsymbol{I}_n - \boldsymbol{A}$ to be singular, that is, $\det(\lambda \boldsymbol{I}_n - \boldsymbol{A}) = 0$. This leads to an n-th order polynomial equation in λ of the form

$$\det(\lambda \boldsymbol{I}_n - \boldsymbol{A}) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = 0. \tag{A.37}$$

This equation, called the *characteristic polynomial equation*, must have n, possibly nondistinct, complex roots which are the eigenvalues of $\boldsymbol{A}$. The *spectral radius* of $\boldsymbol{A}$, denoted $\rho(\boldsymbol{A})$ is defined as

$$\rho(\boldsymbol{A}) = \max_{1 \le i \le n} |\lambda_i(\boldsymbol{A})|,$$

where $\lambda_i(\boldsymbol{A})$ denotes the i-th eigenvalue of $\boldsymbol{A}$.

Any $m \times n$ matrix $\boldsymbol{A}$ can be represented as

$$\boldsymbol{A} = \boldsymbol{U} \begin{bmatrix} \boldsymbol{S}, & \boldsymbol{O} \end{bmatrix} \boldsymbol{V}^\top, \quad \text{if} \quad m \le n, \tag{A.38}$$

or

$$\boldsymbol{A} = \boldsymbol{U} \begin{bmatrix} \boldsymbol{S} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{V}^\top, \quad \text{if} \quad m > n, \tag{A.39}$$

where $\boldsymbol{U}$ is an $m \times m$ orthogonal matrix, $\boldsymbol{V}$ is an $n \times n$ orthogonal matrix, and $\boldsymbol{S}$ is a diagonal matrix of the form

$$\boldsymbol{S} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p), \quad \text{where} \quad p = \min(m,n) \quad \text{and} \quad \sigma_i \ge 0, \quad i = 1, 2, \ldots, p.$$

The nonnegative numbers $\{\sigma_i\}$ are called the *singular values* of $\boldsymbol{A}$, and (A.38), or (A.39), is called the *singular value decomposition* (SVD) of $\boldsymbol{A}$. Following the convention, we write

$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_p,$$

so that $\sigma_1 = \sigma_1(\boldsymbol{A})$ denotes the largest singular value of $\boldsymbol{A}$. The singular values of $\boldsymbol{A}$ are the square roots of the eigenvalues of $\boldsymbol{A}\boldsymbol{A}^\top$ if $m \le n$ or of $\boldsymbol{A}^\top\boldsymbol{A}$ if $m > n$. If $\boldsymbol{A}$ is symmetric, its singular values are the absolute values of its eigenvalues.

A.4 Quadratic Forms

A real *quadratic form* is a special type of a real-valued function of n variables. It is the product $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$, where $\boldsymbol{Q}$ is a specified $n \times n$ matrix with real coefficients and $\boldsymbol{x}$ is an $n \times 1$ real, column, vector. Without loss of generality, the matrix $\boldsymbol{Q}$ can be taken to be symmetric, that is, $\boldsymbol{Q} = \boldsymbol{Q}^\top$, for if $\boldsymbol{Q}$ is not symmetric, we can always replace it with a symmetric matrix without changing the quadratic form values. Indeed, because $z = \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$ is a scalar, hence $z^\top = z$, and we have

$$\left(\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}\right)^\top = \boldsymbol{x}^\top \boldsymbol{Q}^\top \boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}.$$

Therefore, we can write

$$\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} = \boldsymbol{x}^\top \left(\frac{1}{2}\boldsymbol{Q} + \frac{1}{2}\boldsymbol{Q}^\top\right)\boldsymbol{x} = \boldsymbol{x}^\top \left(\frac{\boldsymbol{Q} + \boldsymbol{Q}^\top}{2}\right)\boldsymbol{x},$$

where $\frac{1}{2}\left(\boldsymbol{Q} + \boldsymbol{Q}^\top\right)$ is a symmetric matrix. Thus, the quadratic form values are unchanged if $\boldsymbol{Q}$ is replaced with the symmetric matrix $\frac{1}{2}\left(\boldsymbol{Q} + \boldsymbol{Q}^\top\right)$. From now on, in our study of quadratic forms, we assume that $\boldsymbol{Q}$ is symmetric.

A quadratic form $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$, or equivalently, the matrix $\boldsymbol{Q}$, is called *positive semi-definite* if $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} \ge 0$ for all $\boldsymbol{x}$. It is *positive definite* if $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} > 0$ for all $\boldsymbol{x} \ne \boldsymbol{0}$. If $\boldsymbol{Q}$ is positive semi-definite, then we write $\boldsymbol{Q} \ge 0$, if it is positive definite, we write $\boldsymbol{Q} > 0$. We write $\boldsymbol{Q}_1 \ge \boldsymbol{Q}_2$ to mean that $\boldsymbol{Q}_1 - \boldsymbol{Q}_2 \ge 0$. We say that $\boldsymbol{Q}$ is *negative semi-definite* if $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} \le 0$ for all $\boldsymbol{x}$, it is *negative definite* if $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} < 0$ for all $\boldsymbol{x} \ne \boldsymbol{0}$. Note that $\boldsymbol{Q} \le 0$ if and only if $-\boldsymbol{Q} \ge 0$.

We next present tests for definiteness properties of symmetric matrices. We state these tests in terms of *principal minors* of a given real, symmetric, $n \times n$ matrix $\boldsymbol{Q}$ with elements q_{ij}. We will also give tests for definiteness in terms of eigenvalues of $\boldsymbol{Q}$. The principal minors of $\boldsymbol{Q}$ are $\det \boldsymbol{Q}$ itself and the determinants of submatrices of $\boldsymbol{Q}$ obtained by removing successively an i-th row and i-th column. Specifically, for $p = 1, 2, \ldots, n$, the principal minors are

$$\Delta_p \begin{pmatrix} i_1, & i_2, & \ldots, & i_p \\ i_1, & i_2, & \ldots, & i_p \end{pmatrix} = \det \begin{bmatrix} q_{i_1 i_1} & q_{i_1 i_2} & \cdots & q_{i_1 i_p} \\ q_{i_2 i_1} & q_{i_2 i_2} & \cdots & q_{i_2 i_p} \\ \vdots & \vdots & \ddots & \vdots \\ q_{i_p i_1} & q_{i_p i_2} & \cdots & q_{i_p i_p} \end{bmatrix} \quad 1 \le i_1 < i_2 < \cdots < i_p \le n.$$

The above are called the principal minors of order p.

In the following, we use the fact that if $\boldsymbol{Q} = \boldsymbol{Q}^\top$, then its eigenvalues are all real.

Theorem A.5 *The following are equivalent:*

(i) The symmetric matrix $\boldsymbol{Q}$ *is positive semi-definite.*

(ii) Principal minors of $\boldsymbol{Q}$ *are nonnegative, that is, for* $p = 1, 2, \ldots, n$,

$$\Delta_p \begin{pmatrix} i_1, & i_2, & \ldots, & i_p \\ i_1, & i_2, & \ldots, & i_p \end{pmatrix} \geq 0, \quad 1 \leq i_1 < i_2 < \cdots < i_p \leq n.$$

(iii) Eigenvalues of $\boldsymbol{Q}$ *are nonnegative,*

$$\lambda_i(\boldsymbol{Q}) \geq 0, \quad i = 1, 2, \ldots, n.$$

We give two tests for $\boldsymbol{Q} = \boldsymbol{Q}^\top$ to be positive definite; one in terms of its leading principal minors, and the other in terms of its eigenvalues. The *leading principal minors* of $\boldsymbol{Q}$ are

$$\Delta_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} = q_{11}, \quad \Delta_2 \begin{pmatrix} 1, & 2 \\ 1, & 2 \end{pmatrix} = \det \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix},$$

$$\Delta_3 \begin{pmatrix} 1, & 2, & 3 \\ 1, & 2, & 3 \end{pmatrix} = \det \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}, \ldots, \Delta_n \begin{pmatrix} 1, & 2, & \ldots, & n \\ 1, & 2, & \ldots, & n \end{pmatrix} = \det \boldsymbol{Q}.$$

Theorem A.6 *The following are equivalent:*

(i) The symmetric matrix $\boldsymbol{Q}$ *is positive definite.*

(ii) Leading principal minors of $\boldsymbol{Q}$ *are positive, that is,*

$$\Delta_p \begin{pmatrix} 1, & 2, & \ldots, & p \\ 1, & 2, & \ldots, & p \end{pmatrix} > 0, \quad p = 1, 2, \ldots, n.$$

(iii) Eigenvalues of $\boldsymbol{Q}$ *are positive,*

$$\lambda_i(\boldsymbol{Q}) > 0, \quad i = 1, 2, \ldots, n.$$

If a given matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ is symmetric, then its eigenvalues are all real and there exists an orthogonal matrix $\boldsymbol{U}$ such that

$$\boldsymbol{U}^\top \boldsymbol{Q} \boldsymbol{U} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} = \boldsymbol{\Lambda}. \tag{A.40}$$

Recall that $\boldsymbol{U}$ is orthogonal if $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_n$. Using the above, we show that a symmetric positive semi-definite matrix $\boldsymbol{Q}$ has a positive semi-definite symmetric square root, denoted $\boldsymbol{Q}^{1/2}$ or $\sqrt{\boldsymbol{Q}}$, that satisfies

$$\boldsymbol{Q}^{1/2}\boldsymbol{Q}^{1/2} = \boldsymbol{Q}. \tag{A.41}$$

Indeed, it follows from Theorem A.5 that if $\boldsymbol{Q} = \boldsymbol{Q}^\top \geq 0$ then its eigenvalues are all nonnegative. Hence,

$$\boldsymbol{\Lambda}^{1/2} = \begin{bmatrix} \lambda_1^{1/2} & 0 & \cdots & 0 \\ 0 & \lambda_2^{1/2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n^{1/2} \end{bmatrix} \tag{A.42}$$

is well defined. Using the fact that $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_n$, we define

$$\boldsymbol{Q}^{1/2} = \boldsymbol{U}\boldsymbol{\Lambda}^{1/2}\boldsymbol{U}^\top, \tag{A.43}$$

which has the desired properties.

If $\boldsymbol{Q} = \boldsymbol{Q}^\top$ is positive semi-definite, and not positive definite, then some of its eigenvalues are equal to zero. Suppose that $\boldsymbol{Q}$ has r nonzero eigenvalues. Then, there is an orthogonal matrix, say $\boldsymbol{U}$, such that

$$\boldsymbol{U}^\top\boldsymbol{Q}\boldsymbol{U} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{A.44}$$

Let $\boldsymbol{V} \in \mathbb{R}^{n\times r}$ be a matrix that consists of the first r columns of the matrix $\boldsymbol{U}$, and let

$$\boldsymbol{C} = \begin{bmatrix} \lambda_1^{1/2} & 0 & \cdots & 0 \\ 0 & \lambda_2^{1/2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_r^{1/2} \end{bmatrix} \boldsymbol{V}^\top. \tag{A.45}$$

Note that the matrix $\boldsymbol{C}$ is of full rank. We have

$$\boldsymbol{Q} = \boldsymbol{C}^\top\boldsymbol{C}. \tag{A.46}$$

We refer to (A.46) as a full rank factorization of $\boldsymbol{Q}$ because $\boldsymbol{C}$ is of full rank. We can generalize the above, using (A.38), to obtain a full rank factorization of a non-square matrix. Specifically, given an $m \times n$ matrix $\boldsymbol{A}$ of rank r, we can represent it as

$$\boldsymbol{A} = \boldsymbol{B}\boldsymbol{C},$$

where $\boldsymbol{B}$ is an $m \times r$ matrix and $\boldsymbol{C}$ is a $r \times n$ matrix such that

$$\operatorname{rank}(\boldsymbol{B}) = \operatorname{rank}(\boldsymbol{C}) = \operatorname{rank}(\boldsymbol{A}) = r.$$

A.5 The Kronecker Product

Let $\boldsymbol{A}$ be an $m \times n$ and $\boldsymbol{B}$ be a $p \times q$ matrices. The *Kronecker product* of $\boldsymbol{A}$ and $\boldsymbol{B}$, denoted $\boldsymbol{A} \otimes \boldsymbol{B}$, is an $mp \times nq$ matrix defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1n}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots & a_{2n}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \cdots & a_{mn}\boldsymbol{B} \end{bmatrix}, \tag{A.47}$$

where the symbol $\otimes$ reads "otimes." Thus, the matrix $\boldsymbol{A} \otimes \boldsymbol{B}$ consists of mn blocks. Using the definition of the Kronecker product, we can verify the following two properties:

$$(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = \boldsymbol{A}\boldsymbol{C} \otimes \boldsymbol{B}\boldsymbol{D}, \tag{A.48}$$

$$(\boldsymbol{A} \otimes \boldsymbol{B})^{\top} = \boldsymbol{A}^{\top} \otimes \boldsymbol{B}^{\top}. \tag{A.49}$$

Let

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1, & \boldsymbol{x}_2, & \ldots, & \boldsymbol{x}_m \end{bmatrix}$$

be a $n \times m$ matrix, where $\boldsymbol{x}_i$, $i = 1, 2, \ldots, m$ are the columns of $\boldsymbol{X}$. Each $\boldsymbol{x}_i$ consists of n elements. Then, the *vectorization operator* or *stacking operator* is defined as

$$\begin{aligned} \operatorname{vec}(\boldsymbol{X}) &= \begin{bmatrix} \boldsymbol{x}_1^{\top}, & \boldsymbol{x}_2^{\top}, & \ldots, & \boldsymbol{x}_m^{\top} \end{bmatrix}^{\top} \\ &= \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{n1} & x_{12} & x_{22} & \cdots & x_{n2} & \cdots & x_{1m} & x_{2m} & \cdots & x_{nm} \end{bmatrix}^{\top} \end{aligned}$$

is the column nm-vector formed from the columns of $\boldsymbol{X}$ taken in order. Let now $\boldsymbol{A}$ be an $n \times n$, and $\boldsymbol{C}$ and $\boldsymbol{X}$ be $n \times m$ matrices. Then, the matrix equation

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{C} \tag{A.50}$$

can be written as

$$(\boldsymbol{I}_m \otimes \boldsymbol{A}) \operatorname{vec}(\boldsymbol{X}) = \operatorname{vec}(\boldsymbol{C}). \tag{A.51}$$

Let $\boldsymbol{B}$ be an $m \times m$ matrix, then the matrix equation

$$\boldsymbol{X}\boldsymbol{B} = \boldsymbol{C} \tag{A.52}$$

can be written as

$$\left(\boldsymbol{B}^\top \otimes \boldsymbol{I}_n\right) \operatorname{vec}(\boldsymbol{X}) = \operatorname{vec}(\boldsymbol{C}). \tag{A.53}$$

Using the above two facts, we can verify that the Sylvester matrix equation,

$$\boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{B} = \boldsymbol{C}, \tag{A.54}$$

where $\boldsymbol{A}$ is $n \times n$, $\boldsymbol{B}$ is $m \times m$, and $\boldsymbol{C}$ is $n \times m$, can be written as

$$\left(\boldsymbol{I}_m \otimes \boldsymbol{A} + \boldsymbol{B}^\top \otimes \boldsymbol{I}_n\right) \operatorname{vec}(\boldsymbol{X}) = \operatorname{vec}(\boldsymbol{C}). \tag{A.55}$$

Let λ_i, $\boldsymbol{v}_i$ be the eigenvalues and eigenvectors, respectively, of $\boldsymbol{A}$, and μ_j and $\boldsymbol{w}_j$ the eigenvalues and eigenvectors of $m \times m$ matrix $\boldsymbol{B}$. Then,

$$\begin{aligned}
(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{v}_i \otimes \boldsymbol{w}_j) &= \boldsymbol{A}\boldsymbol{v}_i \otimes \boldsymbol{B}\boldsymbol{w}_j \\
&= \lambda_i \boldsymbol{v}_i \otimes \mu_j \boldsymbol{w}_j \\
&= \lambda_i \mu_j (\boldsymbol{v}_i \otimes \boldsymbol{w}_j). \qquad \text{(A.56)}
\end{aligned}$$

Thus, the eigenvalues of $\boldsymbol{A} \otimes \boldsymbol{B}$ are $\lambda_i \mu_j$, and their respective eigenvectors are $\boldsymbol{v}_i \otimes \boldsymbol{w}_j$ for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

We use the above result to study the Sylvester equation, given by (A.54), that we represent as

$$\boldsymbol{M} \operatorname{vec}(\boldsymbol{X}) = \operatorname{vec}(\boldsymbol{C}), \tag{A.57}$$

where

$$\boldsymbol{M} = \boldsymbol{I}_m \otimes \boldsymbol{A} + \boldsymbol{B}^\top \otimes \boldsymbol{I}_n. \tag{A.58}$$

The solution to (A.57) is unique if, and only if, the $mn \times mn$ matrix $\boldsymbol{M}$ is nonsingular. To find the condition for this to hold, consider the following matrix

$$\left(\boldsymbol{I}_m + \varepsilon \boldsymbol{B}^\top\right) \otimes (\boldsymbol{I}_n + \varepsilon \boldsymbol{A}) = \boldsymbol{I}_m \otimes \boldsymbol{I}_n + \varepsilon \boldsymbol{M} + \varepsilon^2 \boldsymbol{B}^\top \otimes \boldsymbol{A} \tag{A.59}$$

whose eigenvalues are

$$(1 + \varepsilon \mu_j)(1 + \varepsilon \lambda_i) = 1 + \varepsilon(\mu_j + \lambda_i) + \varepsilon^2 \mu_j \lambda_i \tag{A.60}$$

because for a square matrix $\boldsymbol{Q}$,

$$\lambda_i\left(\boldsymbol{I}_n + \varepsilon \boldsymbol{Q}\right) = 1 + \varepsilon \lambda_i\left(\boldsymbol{Q}\right).$$

Comparing terms in ε in (A.59) and (A.60), we conclude that the eigenvalues of $\boldsymbol{M}$ are $\lambda_i + \mu_j$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. Hence, $\boldsymbol{M}$ is nonsingular if and only if

$$\lambda_i + \mu_j \neq 0. \tag{A.61}$$

The above is the necessary and sufficient condition for the solution $\boldsymbol{X}$ of the matrix equation $\boldsymbol{AX} + \boldsymbol{XB} = \boldsymbol{C}$ to be unique.

Another useful identity involving the Kronecker product is

$$\operatorname{vec}(\boldsymbol{ABC}) = \left(\boldsymbol{C}^\top \otimes \boldsymbol{A}\right) \operatorname{vec}(\boldsymbol{B}) \tag{A.62}$$

For more information on the subject of the Kronecker product, we refer the reader to Brewer [42].

A.6 Upper and Lower Bounds

Consider a set $\mathcal{S}$ of real numbers. A number M is called an *upper bound* of the set $\mathcal{S}$ if

$$x \leq M \quad \text{for all} \quad x \in \mathcal{S}.$$

A set of real numbers that has an upper bound is said to be *bounded above.*

The Least Upper Bound Axiom Every nonempty set $\mathcal{S}$ of real numbers that has an upper bound has a *least upper bound*, also called *supremum*, and is denoted

$$\sup\{x : x \in \mathcal{S}\}.$$

Examples

1.

$$\sup\left\{\frac{1}{2}, \frac{2}{3}, \ldots, \frac{n}{n+1}, \ldots\right\} = 1;$$

2.

$$\sup\left\{-\frac{1}{2}, -\frac{1}{8}, -\frac{1}{27}, \ldots, -\frac{1}{n^3}, \ldots\right\} = 0;$$

3.

$$\sup\left\{x : x^2 < 3\right\} = \sup\left\{x : -\sqrt{3} < x < \sqrt{3}\right\} = \sqrt{3}.$$

Theorem A.7 *If $M = \sup\{x : x \in \mathcal{S}\}$ and $\varepsilon > 0$, then there is at least one number x in $\mathcal{S}$ such that*

$$M - \varepsilon < x \leq M.$$

Proof The condition $x \leq M$ is satisfied by all numbers x in $\mathcal{S}$ by virtue of the Least Upper Bound Axiom. We have to show that for an $\varepsilon > 0$ there is a number $x \in \mathcal{S}$ such that

$$M - \varepsilon < x.$$

We prove the statement by contradiction. We suppose that there is no such number in $\mathcal{S}$. Then,

$$x \leq M - \varepsilon \quad \text{for all} \quad x \in \mathcal{S}$$

and hence $M - \varepsilon$ would be an upper bound of $\mathcal{S}$ that is less than M which is the least upper bound. This contradicts the assumption, and thus the proof is complete.

$\square$

To illustrate the above theorem, we consider the following set of real numbers:

$$\mathcal{S} = \left\{ \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \ldots, \frac{n}{n+1}, \ldots \right\}.$$

Note that

$$\sup\{x : x \in \mathcal{S}\} = 1.$$

Let $\varepsilon = 0.1$. Then, for example, the element of the set $\mathcal{S}$ that is equal to 99/100 satisfies

$$1 - \varepsilon < \frac{99}{100} \leq 1.$$

A number m is called a *lower bound* of $\mathcal{S}$ if

$$m \leq x \quad \text{for all} \quad x \in \mathcal{S}.$$

Sets that have lower bounds are said to be *bounded below.*

Theorem A.8 *Every nonempty set of real numbers that has a lower bound has a* greatest lower bound*, also called* infimum*, denoted as*

$$\inf\{x : x \in \mathcal{S}\}.$$

Proof By assumption $\mathcal{S}$ is nonempty and has a lower bound that we call s. Thus

$$s \leq x \quad \text{for all} \quad x \in \mathcal{S}.$$

Hence

$$-x \leq -s \quad \text{for all} \quad x \in \mathcal{S},$$

that is, the set $\{-x : x \in \mathcal{S}\}$ has an upper bound $-s$. From the Least Upper Bound Axiom, we conclude that the set $\{-x : x \in \mathcal{S}\}$ has a least upper bound, supremum. We call it $-s_0$. Because

$$-x \leq -s_0 \quad \text{for all} \quad x \in \mathcal{S},$$

we have

$$s_0 \leq x \quad \text{for all} \quad x \in \mathcal{S},$$

and thus s_0 is a lower bound for $\mathcal{S}$. We now prove, by contradiction, that s_0 is the greatest lower bound, infimum, of the set $\mathcal{S}$. Indeed, if there was a number $\tilde{x}$ satisfying

$$s_0 < \tilde{x} \leq x \quad \text{for all} \quad x \in \mathcal{S},$$

then we would have

$$-x \leq -\tilde{x} < -s_0 \quad \text{for all} \quad x \in \mathcal{S},$$

which contradicts the fact that $-s_0$ is the supremum of $\{-x : x \in \mathcal{S}\}$.

□

Using arguments similar to those we used in the proof of Theorem A.8, we can prove the following theorem:

Theorem A.9 *If $m = \inf\{x : x \in \mathcal{S}\}$ and $\varepsilon > 0$, then there is at least one number x in $\mathcal{S}$ such that*

$$m \leq x < m + \varepsilon.$$

A.7 Sequences

A *sequence of real numbers* is a function whose domain is the set of natural numbers $1, 2, \ldots, n, \ldots$ and whose range is contained in $\mathbb{R}$. Thus, a sequence of real numbers can be viewed as a set of points $(1, a_1), (2, a_2), \ldots, (n, a_n), \ldots$. We denote a sequence of real numbers

$$a_1, a_2, \ldots, a_n, \ldots$$

as $\{a_n\}$.

A sequence $\{a_n\}$ is increasing if $a_1 < a_2 < \cdots < a_n \cdots$. In general, a sequence is increasing if $a_n < a_{n+1}$. If $a_n \leq a_{n+1}$, then we say that the sequence is nondecreasing. Similarly, we can define decreasing and nonincreasing sequences. Nonincreasing or nondecreasing sequences are called monotone sequences.

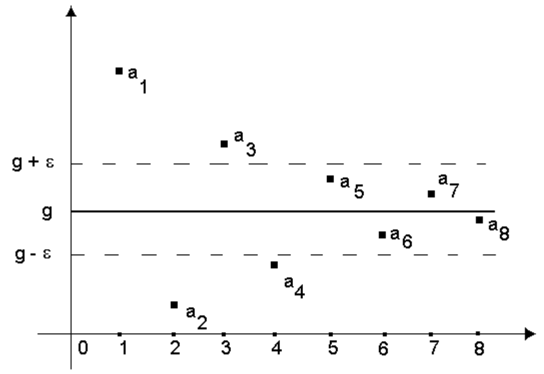


Figure A.2: An illustration of the notion of the limit of a sequence.

A number g is called the limit of the infinite sequence $a_1, a_2, \ldots, a_n, \ldots$ if for any positive ε there is a number $k = k(\varepsilon)$ such that for all $n > k$

$$|a_n - g| < \varepsilon,$$

that is, a_n lies between $g - \varepsilon$ and $g + \varepsilon$ for all $n > k$. In other words, for any $\varepsilon > 0$ we have $|a_n - g| < \varepsilon$ for sufficiently large n's, see Figure A.2, and we write

$$g = \lim_{n\to\infty} a_n,$$

or

$$a_n \to g.$$

A sequence that has a limit is called a *convergent* sequence. A sequence that has no limit is called *divergent*.

The notion of the sequence can be extended to sequences with elements in $\mathbb{R}^p$. Specifically, a sequence in $\mathbb{R}^p$ is a function whose domain is the set of natural numbers $1, 2, \ldots, n, \ldots$ and whose range is contained in $\mathbb{R}^p$.

Suppose that $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n, \ldots$, denoted $\{\boldsymbol{a}_n\}$, is a sequence in $\mathbb{R}^p$. Then, $\boldsymbol{g} \in \mathbb{R}^p$ is called the limit of this sequence if for each $\varepsilon > 0$ there exists a natural number $k = k(\varepsilon)$ such that for all $n > k$

$$\|\boldsymbol{a}_n - \boldsymbol{g}\| < \varepsilon,$$

and we write

$$\boldsymbol{g} = \lim_{n\to\infty} \boldsymbol{a}_n,$$

or

$$\boldsymbol{a}_n \to \boldsymbol{g}.$$

Theorem A.10 *If a limit of a sequence in $\mathbb{R}^p$ exists, then it is unique.*

Proof We prove this theorem by contradiction. Let us assume that a sequence $\{\boldsymbol{a}_n\}$ has two different limits, say $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. Then, we have

$$\|\boldsymbol{g}_1 - \boldsymbol{g}_2\| > 0.$$

Let

$$\varepsilon = \frac{1}{2}\|\boldsymbol{g}_1 - \boldsymbol{g}_2\|.$$

From the definition of a limit, there exist $k_1 = k_1(\varepsilon)$ and $k_2 = k_2(\varepsilon)$ such that for $n > k_1$ we have $\|\boldsymbol{a}_n - \boldsymbol{g}_1\| < \varepsilon$, and for $n > k_2$ we have $\|\boldsymbol{a}_n - \boldsymbol{g}_2\| < \varepsilon$. Let $m = \max(k_1, k_2)$. Then, if $n > m$, we have $\|\boldsymbol{a}_n - \boldsymbol{g}_1\| < \varepsilon$ and $\|\boldsymbol{a}_n - \boldsymbol{g}_2\| < \varepsilon$. Adding $\|\boldsymbol{a}_n - \boldsymbol{g}_1\| < \varepsilon$ and $\|\boldsymbol{a}_n - \boldsymbol{g}_2\| < \varepsilon$ yields

$$\|\boldsymbol{a}_n - \boldsymbol{g}_1\| + \|\boldsymbol{a}_n - \boldsymbol{g}_2\| < 2\varepsilon. \tag{A.63}$$

It follows from the triangle inequality that

$$\|\boldsymbol{a} - \boldsymbol{b}\| \le \|\boldsymbol{a}\| + \|\boldsymbol{b}\|.$$

Using the above inequality, we obtain

$$\begin{aligned} \|-\boldsymbol{g}_1 + \boldsymbol{g}_2\| &= \|\boldsymbol{a}_n - \boldsymbol{g}_1 - \boldsymbol{a}_n + \boldsymbol{g}_2\| \\ &= \|(\boldsymbol{a}_n - \boldsymbol{g}_1) - (\boldsymbol{a}_n - \boldsymbol{g}_2)\| \\ &\le \|\boldsymbol{a}_n - \boldsymbol{g}_1\| + \|\boldsymbol{a}_n - \boldsymbol{g}_2\|. \end{aligned} \tag{A.64}$$

Applying (A.63) to (A.64) yields

$$\|-\boldsymbol{g}_1 + \boldsymbol{g}_2\| = \|\boldsymbol{g}_1 - \boldsymbol{g}_2\| < 2\varepsilon.$$

However, the above contradicts the assumption that $\|\boldsymbol{g}_1 - \boldsymbol{g}_2\| = 2\varepsilon$, which completes the proof.

$\square$

A sequence in $\mathbb{R}^p$ is bounded if there exists a number $M \ge 0$ such that $\|\boldsymbol{a}_n\| \le M$ for all $n = 1, 2, \ldots$. The set of all bounded sequences is sometimes denoted l_∞.

Theorem A.11 *Every convergent sequence is bounded.*

Proof Let

$$\boldsymbol{g} = \lim_{n\to\infty} \boldsymbol{a}_n.$$

Choose $\varepsilon = 1$. Then, there exists a natural number $k = k(\varepsilon)$ such that

$$\|\boldsymbol{a}_n - \boldsymbol{g}\| < 1 \quad \text{for all} \quad n > k.$$

By Theorem A.4,

$$\|\boldsymbol{a}\| - \|\boldsymbol{b}\| \le \|\boldsymbol{a} - \boldsymbol{b}\|.$$

Using the above inequality, we get

$$\|\boldsymbol{a}_n\| - \|\boldsymbol{g}\| \le \|\boldsymbol{a}_n - \boldsymbol{g}\| < 1 \quad \text{for all} \quad n > k.$$

Therefore,

$$\|\boldsymbol{a}_n\| < \|\boldsymbol{g}\| + 1 \quad \text{for all} \quad n > k.$$

Let

$$M = \max\left(\|\boldsymbol{a}_1\|, \|\boldsymbol{a}_2\|, \ldots, \|\boldsymbol{a}_k\|, \|\boldsymbol{g}\| + 1\right),$$

then we have

$$M \ge \|\boldsymbol{a}_n\| \quad \text{for all} \quad n,$$

which means that a convergent sequence is bounded, and the proof is complete.

□

Suppose we are given a sequence $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots$ in $\mathbb{R}^p$, and an increasing sequence of positive natural numbers

$$m_1, m_2, \ldots, m_n, \ldots.$$

The sequence

$$\boldsymbol{a}_{m_1}, \boldsymbol{a}_{m_2}, \ldots, \boldsymbol{a}_{m_n}, \ldots$$

is called a *subsequence* of the sequence $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots$ in $\mathbb{R}^p$.

We have

$$m_n \ge n.$$

Indeed, this is obvious for $n = 1$, that is, $m_1 \ge 1$ because m_1 is a positive integer. We now apply the principle of induction. We assume that $m_n \ge n$ for a given n. Then, we have

$$m_{n+1} > m_n \ge n.$$

Hence,

$$m_{n+1} \geq n+1.$$

Thus, indeed $m_n \geq n$ for all n.

We can say that a subsequence is obtained from the given sequence by neglecting a number of elements in the sequence. We denote a subsequence $\boldsymbol{a}_{m_1}, \boldsymbol{a}_{m_2}, \ldots, \boldsymbol{a}_{m_n}, \ldots$ as

$$\{\boldsymbol{a}_{m_n}\}.$$

Theorem A.12 *A subsequence of a convergent sequence in $\mathbb{R}^p$ is convergent to the same limit, that is, if*

$$\lim_{n\to\infty} \boldsymbol{a}_n = \boldsymbol{g},$$

and if

$$m_1 < m_2 < \ldots < m_n < \ldots,$$

then

$$\lim_{n\to\infty} \boldsymbol{a}_{m_n} = \boldsymbol{g}.$$

Proof Let $\varepsilon > 0$ be given. Then, there exists a natural number $k = k(\varepsilon)$ such that $\|\boldsymbol{a}_n - \boldsymbol{g}\| < \varepsilon$ for any $n > k$. From the fact that $m_n \geq n$, it follows that

$$m_n > n > k,$$

and thus

$$\|\boldsymbol{a}_{m_n} - \boldsymbol{g}\| < \varepsilon$$

for any $m_n > n > k$, which means that

$$\lim_{n\to\infty} \boldsymbol{a}_{m_n} = \boldsymbol{g}.$$

The proof is complete.

□

We now state, without a proof, an important result due to Bolzano and Weierstrass.

Theorem A.13 (Bolzano-Weierstrass) *Every bounded sequence in $\mathbb{R}^p$ contains a convergent subsequence.*

Theorem A.14 (Cauchy) *A sequence $\{\boldsymbol{a}_n\}$ in $\mathbb{R}^p$ is convergent if, and only if for every $\varepsilon > 0$ there exists an $r = r(\varepsilon)$ such that*

$$\|\boldsymbol{a}_n - \boldsymbol{a}_r\| < \varepsilon$$

holds for all $n > r$.

Proof (⇒) Let

$$\lim_{n\to\infty} \boldsymbol{a}_n = \boldsymbol{g},$$

and let $\varepsilon > 0$ be given. Then, for some $l = l(\varepsilon)$, we have

$$\|\boldsymbol{a}_n - \boldsymbol{g}\| < \frac{1}{2}\varepsilon \quad \text{for all} \quad n > l.$$

Let $r = l + 1$, then

$$\|\boldsymbol{a}_r - \boldsymbol{g}\| < \frac{1}{2}\varepsilon.$$

Adding the above inequalities we obtain

$$\|\boldsymbol{a}_n - \boldsymbol{g}\| + \|\boldsymbol{a}_r - \boldsymbol{g}\| < \varepsilon \quad \text{for all} \quad n > r.$$

On the other hand

$$\|\boldsymbol{a}_n - \boldsymbol{a}_r\| \le \|\boldsymbol{a}_n - \boldsymbol{g}\| + \|\boldsymbol{a}_r - \boldsymbol{g}\|.$$

Hence,

$$\|\boldsymbol{a}_n - \boldsymbol{a}_r\| < \varepsilon \quad \text{for all} \quad n > r.$$

(⇐) We now assume that for every $\varepsilon > 0$ there exists an $r = r(\varepsilon)$ such that

$$\|\boldsymbol{a}_n - \boldsymbol{a}_r\| < \varepsilon \quad \text{for all} \quad n > r.$$

We then show that the above condition implies

$$\lim_{n\to\infty} \boldsymbol{a}_n = \boldsymbol{g}.$$

First, we show that $\{\boldsymbol{a}_n\}$ is a bounded sequence. Let $\varepsilon = 1$. Then, an $r = r(\varepsilon)$ exists such that $\|\boldsymbol{a}_n - \boldsymbol{a}_r\| < 1$ for all $n > r$. Hence,

$$\|\boldsymbol{a}_n\| - \|\boldsymbol{a}_r\| \le \|\boldsymbol{a}_n - \boldsymbol{a}_r\| < 1,$$

which means that

$$\|\boldsymbol{a}_n\| < \|\boldsymbol{a}_r\| + 1.$$

Let

$$M = \max\{\|\boldsymbol{a}_1\|, \|\boldsymbol{a}_2\|, \ldots, \|\boldsymbol{a}_{r-1}\|, \|\boldsymbol{a}_r\| + 1\},$$

then we have

$$\|\boldsymbol{a}_n\| \le M \quad \text{for all} \quad n,$$

which implies that the sequence $\{\boldsymbol{a}_n\}$ is bounded. By the Bolzano-Weierstrass theorem the sequence $\{\boldsymbol{a}_n\}$ contains a convergent subsequence. Let

$$\lim_{n\to\infty} \boldsymbol{a}_{m_n} = \boldsymbol{g}, \quad m_1 < m_2 < \ldots .$$

We will show that

$$\lim_{n\to\infty} \boldsymbol{a}_n = \boldsymbol{g}.$$

From the hypothesis of the theorem it follows that for some $r = r\,(\varepsilon/3)$, we have

$$\|\boldsymbol{a}_n - \boldsymbol{a}_r\| < \frac{1}{3}\varepsilon \quad \text{for all} \quad n > r. \tag{A.65}$$

On the other hand, the assumption that the subsequence $\{\boldsymbol{a}_{m_n}\}$ is convergent implies that for some $k = k\,(\varepsilon/3)$,

$$\|\boldsymbol{a}_{m_n} - \boldsymbol{g}\| < \frac{1}{3}\varepsilon \quad \text{for all} \quad n > k. \tag{A.66}$$

We can select k so that $k > r$. In such a case, (A.65) and (A.66) are satisfied for all $n > k$. Since $m_n \geq n > r$, we have for all $n > k$,

$$\|\boldsymbol{a}_{m_n} - \boldsymbol{a}_r\| < \frac{1}{3}\varepsilon,$$

or equivalently,

$$\|\boldsymbol{a}_r - \boldsymbol{a}_{m_n}\| < \frac{1}{3}\varepsilon. \tag{A.67}$$

Adding (A.65), (A.66), and (A.67) yields

$$\|\boldsymbol{a}_n - \boldsymbol{a}_r\| + \|\boldsymbol{a}_{m_n} - \boldsymbol{g}\| + \|\boldsymbol{a}_r - \boldsymbol{a}_{m_n}\| < \varepsilon.$$

Hence,

$$\begin{aligned}
\|\boldsymbol{a}_n - \boldsymbol{g}\| &= \|(\boldsymbol{a}_n - \boldsymbol{a}_r) + (\boldsymbol{a}_r - \boldsymbol{a}_{m_n}) + (\boldsymbol{a}_{m_n} - \boldsymbol{g})\| \\
&\leq \|\boldsymbol{a}_n - \boldsymbol{a}_r\| + \|\boldsymbol{a}_{m_n} - \boldsymbol{g}\| + \|\boldsymbol{a}_r - \boldsymbol{a}_{m_n}\| \\
&< \varepsilon \quad \text{for all} \quad n > k,
\end{aligned}$$

which implies that

$$\lim_{n\to\infty} \boldsymbol{a}_n = \boldsymbol{g}.$$

The proof is complete.

□

Theorem A.15 *Every monotone bounded sequence in* $\mathbb{R}$ *is convergent.*

Proof We will prove the theorem for nondecreasing sequences. The proof for nonincreasing sequences is analogous.

We assume that we are given a sequence $\{a_n\}$ in $\mathbb{R}$ that is bounded and nondecreasing. Let Z be the set of numbers belonging to the sequence, and let g be the least upper bound of Z. Hence

$$g \geq a_n, \quad n = 1, 2, \ldots.$$

From the properties of the least upper bound—see Theorem A.7, it follows that for any $\varepsilon > 0$ there exists a k such that

$$g - \varepsilon < a_k.$$

Since the sequence $\{a_n\}$ is nondecreasing, the inequality $n > k$ implies that

$$a_k \leq a_n,$$

and thus

$$g - \varepsilon < a_n.$$

We then conclude that for any $\varepsilon > 0$, there exists a k such that for $n > k$ we have

$$g - \varepsilon < a_n < g + \varepsilon.$$

Hence,

$$|a_n - g| < \varepsilon,$$

which means that

$$\lim_{n\to\infty} a_n = g.$$

Thus, a monotone bounded sequence in $\mathbb{R}$ is convergent. The proof is complete.

□

Given a sequence $\{a_n\}$ of real numbers. Then the sequence $\{s_N\}$ defined by

$$s_N = \sum_{n=1}^{N} a_n = a_1 + a_2 + \cdots + a_N$$

is called the sequence of *partial sums* of the series

$$\sum_{n=1}^{\infty} a_n. \tag{A.68}$$

If the sequence of the partial sums has a limit, that is, $s_N \to s$ as $N \to \infty$, then we say that the series converges to the sum s, and write

$$s = \sum_{n=1}^{\infty} a_n.$$

The above formula can only make sense when the series (A.68) converges.

Theorem A.16 *If the series $\sum_{n=1}^{\infty} a_n$ converges, then*

$$a_n \to 0 \quad as \quad n \to \infty.$$

Proof Let s be the sum of the series $\sum_{n=1}^{\infty} a_n$. Then,

$$s_N = \sum_{n=1}^{N} a_n \to s \quad \text{as} \quad N \to \infty. \tag{A.69}$$

We also have

$$s_{N-1} \to s \quad \text{as} \quad N \to \infty. \tag{A.70}$$

We have,

$$\begin{aligned} a_N &= (a_1 + a_2 + \cdots + a_N) - (a_1 + a_2 + \cdots + a_{N-1}) \\ &= s_N - s_{N-1}. \end{aligned}$$

Therefore, by (A.69) and (A.70),

$$a_N \to (s - s) = 0 \quad \text{as} \quad N \to \infty,$$

and the proof is complete.

□

The converse of Theorem A.16 is false. The terms of the series

$$\sum_{n=1}^{\infty} \frac{1}{n}$$

tend to zero, but the series diverges.

A.8 Functions

We discuss functions of one and several variables and their basic properties. A function f from a set $\mathcal{A}$ to a set $\mathcal{B}$, denoted

$$f : \mathcal{A} \to \mathcal{B},$$

is a rule that assigns to each $\boldsymbol{x} \in \mathcal{A}$ a unique element $\boldsymbol{y} \in \mathcal{B}$. The set $\mathcal{A}$ is the domain of f and the range of f is a subset of $\mathcal{B}$, not necessarily the whole of $\mathcal{B}$.

Let $f : \mathcal{S} \to \mathbb{R}$, where $\mathcal{S}$ is a subset of $\mathbb{R}$. Then, f is said to be continuous at $z_0 \in \mathcal{S}$ if for every $\varepsilon > 0$, there exists $\delta = \delta(\varepsilon, z_0)$ such that $z \in \mathcal{S}$, $|z - z_0| < \delta$ imply $|f(z) - f(z_0)| < \varepsilon$.

This is also written as $f(z) \to f(z_0)$ as $z \to z_0$. If f is continuous at every point of $\mathcal{S}$, then we say that f is continuous on $\mathcal{S}$.

We say that the function f is bounded on a set $\mathcal{S}$ if there is $M \geq 0$ such that for any $\boldsymbol{z} \in \mathcal{S}$ we have

$$|f(\boldsymbol{z})| \leq M.$$

Lemma A.2 *If f is continuous on $[a, b] \subset \mathbb{R}$, then f is bounded on $[a, b]$.*

Proof Consider the set

$$\{x : x \in [a, b], \text{ and } f \text{ is bounded on } [a, x]\}.$$

This set is nonempty and bounded above by b. Let

$$c = \sup\{x : f \text{ is bounded on } [a, x]\}.$$

We will show that $c = b$. Suppose that $c < b$. From the continuity of f at c it follows that f is bounded on $[c - \delta, c + \delta]$ for some sufficiently small $\delta > 0$. Being bounded on $[a, c - \delta]$ and on $[c - \delta, c + \delta]$, it is bounded on $[a, c + \delta]$. This contradicts our choice of c. We therefore conclude that $c = b$. This means that f is bounded on $[a, x]$ for all $x < b$. From the continuity of f, we know that f is bounded on some interval $[b - \delta, b]$. Hence, f is bounded on $[a, b - \delta]$ and on $[b - \delta, b]$, which means that f is bounded on $[a, b]$.

□

With the help of the above lemma, we can now prove the maximum-minimum theorem of Weierstrass.

Theorem A.17 (The Maximum-Minimum Theorem) *If f is continuous on an interval $[a, b]$, then it takes on both its supremum and infimum values on $[a, b]$.*

Proof By Lemma A.2 above, the function f is bounded on $[a, b]$. Let

$$M = \sup\{f(x) : x \in [a, b]\}.$$

We show that there is a $c \in [a, b]$ such that $f(c) = M$. Let

$$g(x) = \frac{1}{M - f(x)}.$$

If f does not take on the value of M, then g is continuous on $[a, b]$, and by Lemma A.2 the function g is bounded on $[a, b]$. But g cannot be bounded on $[a, b]$. Thus, the assumption that f does not take on the value M has led us to a contradiction. The other case that f takes on its infimum value can be proved in a similar manner.

□

A *sphere*, ball, around $\boldsymbol{x}$ in $\mathbb{R}^p$ is a set of the form $\{\boldsymbol{y} : \|\boldsymbol{x} - \boldsymbol{y}\| < \varepsilon\}$ for some $\varepsilon > 0$. A set Ω in $\mathbb{R}^p$ is said to be *open* if around every point $\boldsymbol{x}$ in Ω there is a sphere that is contained in Ω.

If $\boldsymbol{x} \in \mathbb{R}^p$, then any set that contains an open set containing $\boldsymbol{x}$ is called a *neighborhood* of $\boldsymbol{x}$. A sphere around $\boldsymbol{x}$ is a neighborhood of $\boldsymbol{x}$.

A point $\boldsymbol{x} \in \mathbb{R}^p$ is called a *boundary point* of a set $\Omega \subseteq \mathbb{R}^p$ if every neighborhood of $\boldsymbol{x}$ contains a point of Ω and a point that is not in Ω. The set of all boundary points of Ω is called the *boundary* of Ω. A set is said to be *closed* if it contains its boundary. Equivalently, a set Ω is closed if $\boldsymbol{x}_k \to \boldsymbol{g}$ with $\boldsymbol{x}_k \in \Omega$ implies $\boldsymbol{g} \in \Omega$. A set is *bounded* if it is contained within some sphere of finite radius. A set is *compact* if it is both closed and bounded. Any finite set $\Omega = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$ in $\mathbb{R}^p$ is compact.

We now generalize Lemma A.2 for real-valued functions of several variables.

Lemma A.3 *Let $f : \mathbb{R}^p \to \mathbb{R}$ be a continuous function on a compact set $\Omega \subset \mathbb{R}^p$. Then, f is bounded on Ω.*

Proof We prove the lemma by contradiction. We assume that f is continuous on a compact set Ω, but f is not bounded on Ω. Without loss of generality, suppose that f is unbounded above on Ω. Since f is unbounded above on Ω, we can find a sequence $\{\boldsymbol{x}_n\} \subset \Omega$ such that $f(\boldsymbol{x}_n) > r(n)$, where $r(n) \to \infty$ as $n \to \infty$, that is, $f(\boldsymbol{x}_n) \to \infty$ as $n \to \infty$. The sequence $\{\boldsymbol{x}_n\}$ must be bounded because, by assumption, the set Ω is compact. Therefore, by the Bolzano-Weierstrass theorem, on page 979, there is a convergent subsequence, say, $\{\boldsymbol{x}_{n_k}\}$ that converges to a point $\boldsymbol{x}^* \in \Omega$. The function is continuous on Ω. Hence,

$$\lim_{k\to\infty} f(\boldsymbol{x}_{n_k}) = f\left(\lim_{k\to\infty} \boldsymbol{x}_{n_k}\right) = f\left(\boldsymbol{x}^*\right). \tag{A.71}$$

We arrived at a contradiction because by assumption, the left-hand side of (A.71) is divergent, while $f(\boldsymbol{x}^*) < \infty$ because f is continuous on Ω. Therefore, a continuous function on a compact set in $\mathbb{R}^p$ must be bounded.

□

We next prove the theorem of Weierstrass for real-valued functions of several variables.

Theorem A.18 (Weierstrass) *If $f : \mathbb{R}^p \to \mathbb{R}$ is a continuous function on a compact set $\Omega \subset \mathbb{R}^p$, then f achieves its supremum and infimum.*

Proof The proof follows that of Theorem A.17 with the interval $[a, b]$ being replaced with the compact set Ω, and by referring to Lemma A.3 rather than to Lemma A.2.

□

There is a special type of continuity that we will use later.

Definition A.1 (Uniform continuity) *Let $f : \mathcal{X} \to \mathcal{Y}$. Then, f is called uniformly continuous on $\mathcal{X}$ if for every $\varepsilon > 0$ there exists $\delta = \delta(\varepsilon) > 0$, depending only on ε, such that $|x - x'| < \delta$ implies $|f(x) - f(x')| < \varepsilon$, where x, x' are in $\mathcal{X}$*

◇

Every uniformly continuous function is continuous, but not necessarily conversely. Indeed, $f(x) = 1/x$ is continuous on $(0, 1)$ but not uniformly continuous.

Another type of continuous functions that we will be using later are absolutely continuous functions.

Definition A.2 (Absolutely continuous function) *Let f be a function defined on the closed interval $[a, b]$. Then, the function f is said to be absolutely continuous on $[a, b]$ if for every $\varepsilon > 0$, there exists a $\delta > 0$ such that*

$$\sum_{k=1}^{n} (b_k - a_k) < \delta \tag{A.72}$$

for all numbers $a_1, b_1, \ldots, a_n, b_n$ such that $a_1 < b_1 \le a_2 < b_2 \le \cdots \le a_n < b_n$, we have

$$\sum_{k=1}^{n} |f(b_k) - f(a_k)| < \varepsilon. \tag{A.73}$$

◇

Note that every absolutely continuous function is continuous, because the case $n = 1$ is not excluded in the above definition. A function f that satisfies the Lipschitz condition,

$$|f(x) - f(y)| \le L|x - y|,$$

is absolutely continuous.

A.9 Linear Operators

A function

$$F : \mathbb{R}^n \to \mathbb{R}^m$$

is said to be *linear* if it satisfies the following two properties:

LM 1. For any $\boldsymbol{x}$, $\boldsymbol{y}$ in $\mathbb{R}^n$, we have

$$F(\boldsymbol{x}+\boldsymbol{y}) = F(\boldsymbol{x}) + F(\boldsymbol{y}),$$

LM 2. If c is a number, then

$$F(c\,\boldsymbol{x}) = cF(\boldsymbol{x}).$$

Linear functions are sometimes referred to as *linear maps* or *linear operators*.

Linear functions from $\mathbb{R}^n$ into $\mathbb{R}^m$ can be identified with the $m \times n$ real matrices by

$$\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^m \quad \text{is linear if and only if} \quad \boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$.

If $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^m$ is linear, then there exists a constant $L \geq 0$ such that

$$\|\boldsymbol{F}(\boldsymbol{x})\| \leq L\|\boldsymbol{x}\| \tag{A.74}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$. The smallest such constant L for which this is true is denoted $\|\boldsymbol{F}\|$ and called the norm of the linear map $\boldsymbol{F}$. If $\boldsymbol{F}$ is the linear map induced by the matrix $\boldsymbol{A}$, then we define $\|\boldsymbol{A}\| = \|\boldsymbol{F}\|$. For any matrix $\boldsymbol{A}$ and vector norm $\|\cdot\|$, we can compute the matrix norm using the formula

$$\boxed{\|\boldsymbol{A}\| = \sup_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}} \tag{A.75}$$

The matrix norm, that is, the norm of the linear map $\boldsymbol{F}$ given by (A.75) is equivalent to

$$\boxed{\|\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|} \tag{A.76}$$

A matrix norm produced using (A.75) or (A.76) is called the *induced norm.* Note that

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{y}\| = \|\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{y})\| \leq \|\boldsymbol{A}\|\|\boldsymbol{x}-\boldsymbol{y}\|,$$

that is, the linear map, $\boldsymbol{F}$, represented by the matrix $\boldsymbol{A}$ is uniformly continuous.

A couple of comments on the notation used in (A.75) and (A.76) are needed to clarify the matter. First, the same symbol is used for the matrix norm as for the norm of a vector. Second, the norms appearing on the right side of (A.75) and (A.76) are the norms of the vectors $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^m$. Thus, if $\mathbb{R}^n$ is equipped with the vector norm $\|\cdot\|_s$ while $\mathbb{R}^m$ is equipped with the vector norm $\|\cdot\|_t$, then (A.76) should be interpreted as

$$\|\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|_s=1} \|\boldsymbol{A}\boldsymbol{x}\|_t.$$

Vector norms and a matrix norm are said to be *compatible*, or *consistent*, if they satisfy,

$$\|\boldsymbol{A}\boldsymbol{x}\|_t \leq \|\boldsymbol{A}\|\|\boldsymbol{x}\|_s. \tag{A.77}$$

We next show that (A.76) is well defined. For this, we need the following lemma.

Lemma A.4 *A vector norm* $\|\cdot\| : \mathbb{R}^p \to \mathbb{R}$ *is an uniformly continuous function.*

Proof By Theorem A.4, on page 962,

$$|\|\boldsymbol{x}\| - \|\boldsymbol{y}\|| \leq \|\boldsymbol{x} - \boldsymbol{y}\|.$$

Hence, for any $\varepsilon > 0$, if we take $\delta = \varepsilon$ then for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, if $\|\boldsymbol{x} - \boldsymbol{y}\| < \delta$ then $|\|\boldsymbol{x} - \boldsymbol{y}\|| < \varepsilon$, as was to be shown.

□

By the continuity of the vector norm and the theorem of Weierstrass, on page 985, a vector $\boldsymbol{x}_0$ can be found such that $\|\boldsymbol{x}_0\| = 1$ and $\|\boldsymbol{A}\| = \|\boldsymbol{A}\boldsymbol{x}_0\| = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|$. Thus, the expression (A.76) is well defined.

We show that (A.76) satisfies (A.77). Suppose that we are given a vector $\boldsymbol{y} \neq \mathbf{0}$. Then,

$$\boldsymbol{x} = \frac{1}{\|\boldsymbol{y}\|}\boldsymbol{y}$$

satisfies the condition $\|\boldsymbol{x}\| = 1$. Hence,

$$\|\boldsymbol{A}\boldsymbol{y}\| = \|\boldsymbol{A}\left(\|\boldsymbol{y}\|\boldsymbol{x}\right)\| = \|\boldsymbol{y}\|\|\boldsymbol{A}\boldsymbol{x}\|.$$

Applying now (A.76) to the right-hand side of the above yields

$$\|\boldsymbol{A}\boldsymbol{y}\| \leq \|\boldsymbol{y}\|\|\boldsymbol{A}\|.$$

Hence, any matrix norm produced by (A.76) is consistent.

Next, we prove that a matrix norm obtained using (A.76) satisfies the following conditions:

MN 1. $\|\boldsymbol{A}\| > 0$ if $\boldsymbol{A} \neq \boldsymbol{O}$ and $\|\boldsymbol{O}\| = 0$.

MN 2. $\|c\boldsymbol{A}\| = |c|\|\boldsymbol{A}\|$, where c is a constant.

MN 3. $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$.

MN 4. $\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\|$.

Proof of MN 1. Let $\boldsymbol{A} \neq \boldsymbol{O}$. Then a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\| = 1$, can be found such that $\boldsymbol{A}\boldsymbol{x} \neq \boldsymbol{0}$, and thus $\|\boldsymbol{A}\boldsymbol{x}\| \neq 0$. Hence, $\|\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\| \neq 0$. If, on the other hand, $\boldsymbol{A} = \boldsymbol{O}$, then $\|\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{O}\boldsymbol{x}\| = 0$.

Proof of MN 2. By definition, $\|c\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|=1} \|c\boldsymbol{A}\boldsymbol{x}\|$. Using a property of the vector norm, we write $\|c\boldsymbol{A}\boldsymbol{x}\| = |c|\|\boldsymbol{A}\boldsymbol{x}\|$, and hence

$$\|c\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|=1} |c|\|\boldsymbol{A}\boldsymbol{x}\| = |c| \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\| = |c|\|\boldsymbol{A}\|.$$

Proof of MN 3. For the sum of two matrices $\boldsymbol{A}+\boldsymbol{B}$, there exists a vector $\boldsymbol{x}_0$, $\|\boldsymbol{x}_0\| = 1$, such that

$$\|\boldsymbol{A} + \boldsymbol{B}\| = \|(\boldsymbol{A} + \boldsymbol{B})\boldsymbol{x}_0\|.$$

Then, using properties of the vector norm, we obtain

$$\|\boldsymbol{A} + \boldsymbol{B}\| = \|\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{B}\boldsymbol{x}_0\| \leq \|\boldsymbol{A}\boldsymbol{x}_0\| + \|\boldsymbol{B}\boldsymbol{x}_0\|.$$

Using (A.76) yields

$$\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{x}_0\| + \|\boldsymbol{B}\|\|\boldsymbol{x}_0\| = \|\boldsymbol{A}\| + \|\boldsymbol{B}\|.$$

Proof of MN 4. For the product of two matrices $\boldsymbol{A}\boldsymbol{B}$, there exists a vector $\boldsymbol{x}_0$, $\|\boldsymbol{x}_0\| = 1$, such that $\|\boldsymbol{A}\boldsymbol{B}\| = \|\boldsymbol{A}\boldsymbol{B}\boldsymbol{x}_0\|$. Then, using (A.76), we have

$$\|\boldsymbol{A}\boldsymbol{B}\| = \|\boldsymbol{A}(\boldsymbol{B}\boldsymbol{x}_0)\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\boldsymbol{x}_0\|.$$

Using (A.76) again yields

$$\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\|\|\boldsymbol{x}_0\| = \|\boldsymbol{A}\|\|\boldsymbol{B}\|.$$

□

The most frequently used matrix norms are the p-norms

$$\|\boldsymbol{A}\|_p = \max_{\|\boldsymbol{x}\|_p=1} \|\boldsymbol{A}\boldsymbol{x}\|_p.$$

The induced matrix norms corresponding to the one-, two-, and infinity-norm for vectors are:

- $\|\boldsymbol{A}\|_1 = \max_j \|\boldsymbol{a}_j\|_1$, that is, $\|\boldsymbol{A}\|_1$ is the maximum of absolute column sum;
- $\|\boldsymbol{A}\|_2 = \sigma_1(\boldsymbol{A})$, which is the largest singular value of $\boldsymbol{A}$;

- $\|\boldsymbol{A}\|_\infty = \max_i \|\boldsymbol{a}_i^\top\|_1$, which is the maximum of absolute row sum.

A matrix norm that is not induced by a vector norm is the *Frobenius norm*, defined for an $m \times n$ matrix $\boldsymbol{A}$ as

$$\|\boldsymbol{A}\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)^{1/2}. \tag{A.78}$$

The Frobenius norm satisfies the relation

$$\|\boldsymbol{A}\|_F^2 = \operatorname{trace}\left(\boldsymbol{A}\boldsymbol{A}^\top\right) = \operatorname{trace}\left(\boldsymbol{A}^\top\boldsymbol{A}\right). \tag{A.79}$$

The Frobenius norm is compatible with the Euclidean vector norm, that is,

$$\|\boldsymbol{A}\boldsymbol{x}\|_2 \le \|\boldsymbol{A}\|_F\|\boldsymbol{x}\|_2. \tag{A.80}$$

A.10 Vector Spaces

A given set of vectors of the same dimension $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r\}$ is *linearly dependent* if the zero vector, $\boldsymbol{0}$, can be written as a non-trivial *linear combination* of the given vectors,

$$\alpha_1\boldsymbol{x}_1 + \alpha_2\boldsymbol{x}_2 + \cdots + \alpha_r\boldsymbol{x}_r = \boldsymbol{0}, \quad \text{with} \quad \alpha_i \neq 0 \quad \text{for at least one } i.$$

In other words, the set of vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r\}$ is *linearly independent* if

$$\alpha_1\boldsymbol{x}_1 + \alpha_2\boldsymbol{x}_2 + \cdots + \alpha_r\boldsymbol{x}_r = \boldsymbol{0}, \quad \text{implies} \quad \alpha_1 = \alpha_2 = \cdots = \alpha_r = 0.$$

A set $\mathcal{S}$ containing vectors of dimension p is a *subspace* of $\mathbb{R}^p$ if for any scalars α and β,

$$\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S} \quad \text{implies} \quad \alpha\boldsymbol{x} + \beta\boldsymbol{y} \in \mathcal{S}. \tag{A.81}$$

It follows from the above that a subspace must contain the zero vector, $\boldsymbol{0}$. The set containing only the zero vector is a subspace. This set, $\{\boldsymbol{0}\}$, is called a *trivial* subspace.

A matrix-vector product

$$\boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2 + \cdots + x_n\boldsymbol{a}_n \tag{A.82}$$

is a linear combination of the columns of the matrix. Thus, linear dependence of the columns of $\boldsymbol{A}$ is equivalent to the condition

$$\boldsymbol{A}\boldsymbol{z} = \boldsymbol{0} \quad \text{for some vector} \quad \boldsymbol{z} \neq \boldsymbol{0}. \tag{A.83}$$

Linear independence of the columns of $\boldsymbol{A}$ is equivalent to the condition

$$\boldsymbol{A}\boldsymbol{z} = \boldsymbol{0} \quad \text{implies} \quad \boldsymbol{z} = \boldsymbol{0}. \tag{A.84}$$

The set of all vectors that are linear combinations of the columns of $\boldsymbol{A}$ is called the *range*, or *image*, of $\boldsymbol{A}$ and will be denoted $\operatorname{range}(\boldsymbol{A})$. Thus,

$$\operatorname{range}(\boldsymbol{A}) = \{\boldsymbol{v} : \boldsymbol{v} = \boldsymbol{A}\boldsymbol{x} \quad \text{for some} \quad \boldsymbol{x}\}. \tag{A.85}$$

For a given $m \times n$ matrix $\boldsymbol{A}$ and an m-vector $\boldsymbol{b}$, we have $\boldsymbol{b} \in \operatorname{range}(\boldsymbol{A})$ if and only if there exists an n-vector $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. We say that a system of linear equations represented by $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is compatible if $\boldsymbol{b} \in \operatorname{range}(\boldsymbol{A})$.

The range of $\boldsymbol{A}$ is a subspace of $\mathbb{R}^m$. Indeed, for any $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \operatorname{range}(\boldsymbol{A})$, we have $\boldsymbol{b}_1 = \boldsymbol{A}\boldsymbol{x}_1$ and $\boldsymbol{b}_2 = \boldsymbol{A}\boldsymbol{x}_2$ for some $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Using the fact that $\boldsymbol{A}$ represents a linear map, for any scalars α_1 and α_2,

$$\alpha_1\boldsymbol{b}_1 + \alpha_2\boldsymbol{b}_2 = \alpha_1\boldsymbol{A}\boldsymbol{x}_1 + \alpha_2\boldsymbol{A}\boldsymbol{x}_2 = \boldsymbol{A}\left(\alpha_1\boldsymbol{x}_1 + \alpha_2\boldsymbol{x}_2\right). \tag{A.86}$$

Hence, if $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \operatorname{range}(\boldsymbol{A})$, then $\alpha_1\boldsymbol{b}_1 + \alpha_2\boldsymbol{b}_2 \in \operatorname{range}(\boldsymbol{A})$, which means that the range of $\boldsymbol{A}$ is a subspace of $\mathbb{R}^m$.

Similarly, we define the range of $\boldsymbol{A}^\top$ to be the set

$$\operatorname{range}(\boldsymbol{A}^\top) = \left\{\boldsymbol{y} : \boldsymbol{y} = \boldsymbol{A}^\top\boldsymbol{z} \quad \text{for some} \quad \boldsymbol{z}\right\}. \tag{A.87}$$

For a non-trivial subspace $\mathcal{S}$ there is a unique smallest positive integer called the *dimension* of the subspace $\mathcal{S}$ and denoted $\dim \mathcal{S}$. The dimension of the subspace is equal to the smallest number of linearly independent vectors of the subspace such that every vector of the subspace can be expressed as a linear combination of these vectors. Any such set of vectors is called a *basis* of the subspace. The vectors forming a basis are not unique.

The number of linearly independent columns of a matrix $\boldsymbol{A}$ is called the *rank* of $\boldsymbol{A}$ and denoted $\operatorname{rank}(\boldsymbol{A})$. The rank of $\boldsymbol{A}$ is also equal to the number of linearly independent rows of $\boldsymbol{A}$.

For the product of a $m\times n$ matrix $\boldsymbol{A}$ and a $n\times p$ matrix $\boldsymbol{B}$, we have *Sylvester's inequalities*:

$$\operatorname{rank}(\boldsymbol{A}) + \operatorname{rank}(\boldsymbol{B}) - n \leq \operatorname{rank}(\boldsymbol{A}\boldsymbol{B}) \leq \min\{\operatorname{rank}(\boldsymbol{A}), \operatorname{rank}(\boldsymbol{B})\}. \tag{A.88}$$

For an $m \times n$ matrix $\boldsymbol{A}$, the set of all n-vectors orthogonal to the rows of $\boldsymbol{A}$ is called the *kernel*, or *null space*, of $\boldsymbol{A}$ and denoted $\ker(\boldsymbol{A})$, that is,

$$\ker(\boldsymbol{A}) = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}. \tag{A.89}$$

The kernel of $\boldsymbol{A}$ is a subspace of $\mathbb{R}^n$. Indeed, if $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \ker(\boldsymbol{A})$, then $\boldsymbol{A}\boldsymbol{x}_1 = \boldsymbol{0}$ and $\boldsymbol{A}\boldsymbol{x}_2 = \boldsymbol{0}$. Hence, for any scalars α_1 and α_2, we have

$$\boldsymbol{A}\left(\alpha_1 \boldsymbol{x}_1 + \alpha_2 \boldsymbol{x}_2\right) = \alpha_1 \boldsymbol{A}\boldsymbol{x}_1 + \alpha_2 \boldsymbol{A}\boldsymbol{x}_2 = \boldsymbol{0}. \tag{A.90}$$

Thus, if $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \ker(\boldsymbol{A})$, then $\alpha_1\boldsymbol{x}_1 + \alpha_2\boldsymbol{x}_2 \in \ker(\boldsymbol{A})$, which means that $\ker(\boldsymbol{A})$ is a subspace of $\mathbb{R}^n$. Similarly, we define the kernel of $\boldsymbol{A}^\top$ to be the set

$$\ker\left(\boldsymbol{A}^\top\right) = \left\{\boldsymbol{w} : \boldsymbol{A}^\top \boldsymbol{w} = \boldsymbol{0}\right\}. \tag{A.91}$$

The intersection of $\operatorname{range}(\boldsymbol{A})$ and $\ker\left(\boldsymbol{A}^\top\right)$ contains only the zero vector, that is,

$$\operatorname{range}(\boldsymbol{A}) \cap \ker\left(\boldsymbol{A}^\top\right) = \{\boldsymbol{0}\}. \tag{A.92}$$

For a given subspace $\mathcal{S}$ of $\mathbb{R}^n$, the orthogonal complement $\mathcal{S}^\perp$ of $\mathcal{S}$ is defined by

$$\mathcal{S}^\perp = \left\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}^\top \boldsymbol{y} = 0 \quad \text{for all} \quad \boldsymbol{y} \in \mathcal{S}\right\}. \tag{A.93}$$

The vectors in $\ker\left(\boldsymbol{A}^\top\right)$ are orthogonal to the vectors in $\operatorname{range}(\boldsymbol{A})$, and vice versa. We say that $\operatorname{range}(\boldsymbol{A})$ and $\ker\left(\boldsymbol{A}^\top\right)$ are *orthogonal complements* of one another and write

$$\operatorname{range}(\boldsymbol{A})^\perp = \ker\left(\boldsymbol{A}^\top\right), \tag{A.94}$$

and

$$\ker\left(\boldsymbol{A}^\top\right)^\perp = \operatorname{range}(\boldsymbol{A}). \tag{A.95}$$

Similarly,

$$\operatorname{range}\left(\boldsymbol{A}^\top\right)^\perp = \ker\left(\boldsymbol{A}\right), \tag{A.96}$$

and

$$\ker\left(\boldsymbol{A}\right)^\perp = \operatorname{range}\left(\boldsymbol{A}^\top\right). \tag{A.97}$$

If $\boldsymbol{A}$ is an $m \times n$ matrix with linearly independent columns, which implies that $m \geq n$, then

$$\operatorname{rank}(\boldsymbol{A}) = n = \dim \operatorname{range}(\boldsymbol{A}),$$

and

$$\dim \ker\left(\boldsymbol{A}^\top\right) = m - n.$$

We now define a normed space.

Definition A.3 *A vector space with a norm is called a* normed space.

◇

By the normed vector space $\mathbb{R}^n$, we usually mean the Euclidean space, which is $\mathbb{R}^n$ with the norm $\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}$. The vector space $\mathbb{R}^n$ equipped with any of the p-norm forms a normed vector space. The absolute value is a norm in $\mathbb{R}$.

Let Ω be a compact subset of $\mathbb{R}^n$ and let $\mathcal{C}(\Omega)$ denote the set of continuous functions $f : \mathbb{R}^n \to \mathbb{R}$ on Ω. Thus, we view functions as vectors in the function space $\mathcal{C}(\Omega)$. Then, the function $\|\cdot\| : \mathcal{C}(\Omega) \to \mathbb{R}$ defined as

$$\|f\| = \max_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x})|, \tag{A.98}$$

satisfies the norm axioms. The function space $\mathcal{C}(\Omega)$, with scalars from $\mathbb{R}$ and with the norm (A.98) is a normed linear space. In particular, the space of all real-valued functions, $f(x)$, continuous on some interval $[a, b]$ of the real line with the norm defined by

$$\|f\| = \max_{x\in[a,b]} |f(x)|, \tag{A.99}$$

is an example of a normed linear space. This space is denoted $\mathcal{C}([a, b])$.

We will also be interested in sequences of elements of $\mathcal{C}([a, b])$. Let $f_1, f_2, \ldots$ be a sequence of elements of $\mathcal{C}([a, b])$. Denote this sequence by $\{f_k\}$. In the following, $f_k(x)$, means the value of the function f_k at x, while f_k means the function itself. We say that the sequence $\{f_k\}$ converges (pointwise) on the interval $[a, b]$ if there exists a function f such that for every $x \in [a, b]$, the sequence $\{f_k(x)\}$ converges to $f(x)$. In other words, the sequence $\{f_k\}$ converges (pointwise) if there exists a function f such that for a given $x \in [a, b]$ and a given $\varepsilon > 0$, we can find $N = N(x, \varepsilon)$ such that

$$|f(x) - f_k(x)| < \varepsilon \quad \text{for} \quad k > N. \tag{A.100}$$

The sequence $\{f_k\}$ converges uniformly, on the interval $[a, b]$, if there is a function f such that for any $\varepsilon > 0$ there exists $N = N(\varepsilon)$, depending on ε but not on x, such that (A.100) holds for every $x \in [a, b]$. The distinction between convergence and uniform convergence is that in the later case, the same N can be used for any value of x to show the convergence of the sequence $\{f_k(x)\}$. One can show that the limit of an uniformly convergent sequence of continuous functions is continuous. It can be verified that the norm (A.99) defines the uniform convergence because $\{f_k\}$ converges uniformly to f if and only if $\max_{x\in[a,b]} |f(x) - f_k(x)| \to 0$. This is the reason the norm (A.99) is sometimes called the *norm of uniform convergence*. It can be shown that there is no norm on $\mathcal{C}([a, b])$ that defines pointwise convergence to an element in $\mathcal{C}([a, b])$—see Debnath and Mikusiński [74, p. 12].

Definition A.4 (Banach Space) *A normed linear space is complete, if every Cauchy sequence with elements form the space converges to an element in the space. A complete normed linear space is called a Banach space.*

◇

The Euclidean space, as well as $\mathcal{C}([a,b])$ with the norm (A.99), are examples of Banach spaces.

A.11 Least Squares

Consider the system of equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{A.101}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{b} \in \mathbb{R}^m$. A necessary and sufficient condition for the system (A.101) to have a solution is

$$\operatorname{rank}(\boldsymbol{A}) = \operatorname{rank}\left[\begin{array}{ccc} \boldsymbol{A} & \vdots & \boldsymbol{b} \end{array}\right]. \tag{A.102}$$

In particular, if the matrix $\boldsymbol{A}$ is invertible, then there is a unique solution to (A.101) given by

$$\boldsymbol{x}^* = \boldsymbol{A}^{-1}\boldsymbol{b}.$$

Here, we analyze the case when $m > n$ and $\operatorname{rank}(\boldsymbol{A}) < \operatorname{rank}\left[\begin{array}{ccc} \boldsymbol{A} & \vdots & \boldsymbol{b} \end{array}\right]$, that is, $\boldsymbol{b} \notin \operatorname{range}(\boldsymbol{A})$, which means that the system (A.101) does not have an exact solution. We modify (A.101) by incorporating an error vector $\boldsymbol{e}$ to obtain

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} = \boldsymbol{b}. \tag{A.103}$$

This is illustrated in Figure A.3. Because

$$\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}, \tag{A.104}$$

we have

$$\begin{aligned} \|\boldsymbol{e}\|^2 &= \boldsymbol{e}^\top\boldsymbol{e} \\ &= (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) \\ &= \boldsymbol{x}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{x}^\top\boldsymbol{A}^\top\boldsymbol{b} + \boldsymbol{b}^\top\boldsymbol{b}. \end{aligned} \tag{A.105}$$

The gradient of $\|\boldsymbol{e}\|^2$ with respect to $\boldsymbol{x}$ is

$$\nabla_{\boldsymbol{x}}\|\boldsymbol{e}\|^2 = 2\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{A}^\top\boldsymbol{b}. \tag{A.106}$$

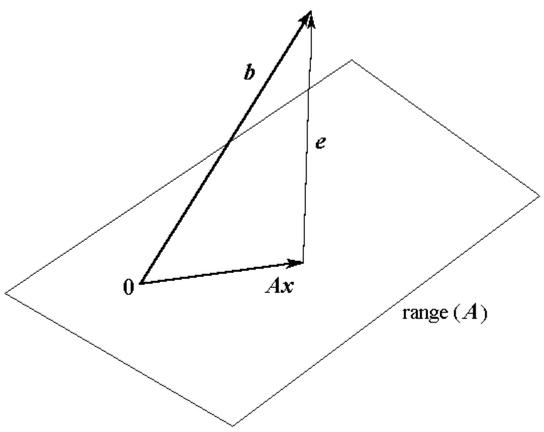


Figure A.3: Geometrical interpretation of the least squares solution to a system of linear equations.

Solving

$$\nabla_{\boldsymbol{x}} \|\boldsymbol{e}\|^2 = \mathbf{0}$$

yields candidate minimizer points of (A.105). If the matrix $\boldsymbol{A}$ has full column rank, then the matrix $\boldsymbol{A}^\top \boldsymbol{A}$ is invertible and the only critical point is

$$\boxed{\boldsymbol{x}^* = \left(\boldsymbol{A}^\top \boldsymbol{A}\right)^{-1} \boldsymbol{A}^\top \boldsymbol{b}} \tag{A.107}$$

The above point satisfies the second order sufficient condition to be the minimizer of (A.105) because the Hessian of (A.105) is

$$2\boldsymbol{A}^\top \boldsymbol{A},$$

which is positive definite. The solution given by (A.107) is called the least squares solution of the inconsistent linear system of equations, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, because it minimizes the sum of squares $\|\boldsymbol{e}\|^2 = \boldsymbol{e}^\top \boldsymbol{e} = \sum_{i=1}^m e_i^2$. For a graphical interpretation of the least squares solution, we refer to Figure A.3.

Suppose now that we interpret the i-th row of the matrix $\boldsymbol{A}$ and the i-th component of the vector $\boldsymbol{b}$ as the i-th data pair. Thus, the i-th data pair has the form $\left(\boldsymbol{a}_i^\top; b_i\right)$. Suppose that the k data pairs are represented by the pair $(\boldsymbol{A}; \boldsymbol{b})$. We know that the least squares solution to the system of equations represented by the data pair $(\boldsymbol{A}; \boldsymbol{b})$ is given by (A.107). If the new data pair, $\left(\boldsymbol{a}_{k+1}^\top; b_{k+1}\right)$, becomes available, then we could find the least squares solution to the system of equations

$$\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{b} \\ b_{k+1} \end{bmatrix} \tag{A.108}$$

by applying the formula (A.107) to (A.108). Let $\boldsymbol{x}_{k+1}$ denote the least squares solution to the above problem. Then, we have

$$\boldsymbol{x}_{k+1} = \left(\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix} \right)^{-1} \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{b} \\ b_{k+1} \end{bmatrix}. \tag{A.109}$$

In the case of large k the above method of recomputing the new least squares solution from scratch may be time consuming. We present a recursive method of computing least squares solutions as the data pairs become available. To proceed, let

$$\boldsymbol{P}_k = \left(\boldsymbol{A}^\top \boldsymbol{A} \right)^{-1}. \tag{A.110}$$

In view of the above, we can write

$$\begin{aligned} \boldsymbol{P}_{k+1}^{-1} &= \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}^\top & \boldsymbol{a}_{k+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix} \\ &= \boldsymbol{A}^\top \boldsymbol{A} + \boldsymbol{a}_{k+1} \boldsymbol{a}_{k+1}^\top \\ &= \boldsymbol{P}_k^{-1} + \boldsymbol{a}_{k+1} \boldsymbol{a}_{k+1}^\top. \end{aligned} \tag{A.111}$$

Hence,

$$\boldsymbol{P}_{k+1} = \left(\boldsymbol{P}_k^{-1} + \boldsymbol{a}_{k+1} \boldsymbol{a}_{k+1}^\top \right)^{-1}.$$

Applying to the above the matrix inversion formula (A.30), we obtain

$$\boxed{\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \frac{\boldsymbol{P}_k \boldsymbol{a}_{k+1} \boldsymbol{a}_{k+1}^\top \boldsymbol{P}_k}{1 + \boldsymbol{a}_{k+1}^\top \boldsymbol{P}_k \boldsymbol{a}_{k+1}}} \tag{A.112}$$

Using (A.110), we write

$$\begin{aligned} \boldsymbol{x}_k &= \left(\boldsymbol{A}^\top \boldsymbol{A} \right)^{-1} \boldsymbol{A}^\top \boldsymbol{b} \\ &= \boldsymbol{P}_k \boldsymbol{A}^\top \boldsymbol{b}. \end{aligned} \tag{A.113}$$

Using (A.111), we can write

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \left(\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix} \right)^{-1} \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{a}_{k+1}^\top \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{b} \\ b_{k+1} \end{bmatrix} \\ &= \boldsymbol{P}_{k+1} \left(\boldsymbol{A}^\top \boldsymbol{b} + \boldsymbol{a}_{k+1} b_{k+1} \right). \end{aligned} \tag{A.114}$$

Computing $\boldsymbol{A}^\top \boldsymbol{b}$ from (A.113), we get

$$\boldsymbol{A}^\top \boldsymbol{b} = \boldsymbol{P}_k^{-1} \boldsymbol{x}_k. \tag{A.115}$$

Substituting the above into (A.114) yields

$$\boldsymbol{x}_{k+1} = \boldsymbol{P}_{k+1} \left(\boldsymbol{P}_k^{-1} \boldsymbol{x}_k + \boldsymbol{a}_{k+1} b_{k+1} \right). \tag{A.116}$$

Computing $\boldsymbol{P}_k^{-1}$ from (A.111) gives

$$\boldsymbol{P}_k^{-1} = \boldsymbol{P}_{k+1}^{-1} - \boldsymbol{a}_{k+1} \boldsymbol{a}_{k+1}^\top$$

Substituting the above into (A.116) we have

$$\boldsymbol{x}_{k+1} = \boldsymbol{P}_{k+1} \left(\left(\boldsymbol{P}_{k+1}^{-1} - \boldsymbol{a}_{k+1} \boldsymbol{a}_{k+1}^\top \right) \boldsymbol{x}_k + \boldsymbol{a}_{k+1} b_{k+1} \right). \tag{A.117}$$

Taking into account that $\boldsymbol{a}_{k+1}^\top \boldsymbol{x}_k$ is a scalar while performing simple manipulations, we obtain

$$\boxed{\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \left(b_{k+1} - \boldsymbol{a}_{k+1}^\top \boldsymbol{x}_k \right) \boldsymbol{P}_{k+1} \boldsymbol{a}_{k+1}} \tag{A.118}$$

The two equations (A.112) and (A.118) constitute the *recursive least squares* (RLS) algorithm. To initialize the RLS, we can set

$$\boldsymbol{P}_0 = \left(\boldsymbol{A}_n^\top \boldsymbol{A}_n \right)^{-1} \quad \text{and} \quad \boldsymbol{x}_0 = \boldsymbol{P}_n \boldsymbol{A}_n^\top \boldsymbol{b}_n,$$

where the pair $(\boldsymbol{A}_n; \boldsymbol{b}_n)$ represents the first n data pairs.

A.12 Contraction Maps

An operator $\boldsymbol{F}$ that maps Euclidean space $\mathbb{R}^p$ into itself is called a *contraction map* if for any two elements $\boldsymbol{x}$ and $\boldsymbol{y}$ of $\mathbb{R}^p$,

$$\|\boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{F}(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \quad 0 < L < 1. \tag{A.119}$$

A point $\boldsymbol{x}^* \in \mathbb{R}^p$ is called a *fixed point* if

$$\boldsymbol{F}(\boldsymbol{x}^*) = \boldsymbol{x}^*. \tag{A.120}$$

The importance of the contraction operators lies in the fact that there is a unique fixed point. Furthermore, this point may be found by successive iterations as stated in the following theorem.

Theorem A.19 *Let $\boldsymbol{F} : \mathbb{R}^p \to \mathbb{R}^p$ be a contraction map with the contraction constant $0 < L < 1$. Then $\boldsymbol{F}$ has a unique fixed point, and the sequence*

$$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots$$

generated by $\boldsymbol{x}^{(n+1)} = \boldsymbol{F}\left(\boldsymbol{x}^{(n)}\right)$, $n = 0, 1, 2, \ldots$, converges to $\boldsymbol{x}^*$ for any initial choice of $\boldsymbol{x}^{(0)} \in \mathbb{R}^p$.

Proof To show the uniqueness of the fixed point $\boldsymbol{x}^*$, suppose that there exist two distinct fixed points $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$. We then have

$$\begin{aligned} \|\boldsymbol{x}^* - \boldsymbol{y}^*\| &= \|\boldsymbol{F}(\boldsymbol{x}^*) - \boldsymbol{F}(\boldsymbol{y}^*)\| \\ &\leq L\,\|\boldsymbol{x}^* - \boldsymbol{y}^*\|. \end{aligned}$$

Thus,

$$\|\boldsymbol{x}^* - \boldsymbol{y}^*\| \leq L\,\|\boldsymbol{x}^* - \boldsymbol{y}^*\|, \quad 0 < L < 1,$$

which is impossible unless $\|\boldsymbol{x}^* - \boldsymbol{y}^*\| = 0$. Hence, $\boldsymbol{x}^* = \boldsymbol{y}^*$ which means that $\boldsymbol{F}$ has at most one fixed point.

We next show that a fixed point exists, and that the sequence

$$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots$$

generated by $\boldsymbol{x}^{(n+1)} = \boldsymbol{F}\left(\boldsymbol{x}^{(n)}\right)$, $n = 0, 1, 2, \ldots$, converges to $\boldsymbol{x}^*$ for any initial choice of $\boldsymbol{x}^{(0)} \in \mathbb{R}^p$. If $\boldsymbol{F}\left(\boldsymbol{x}^{(0)}\right) = \boldsymbol{x}^{(0)}$, then the theorem is proved. In general $\boldsymbol{F}\left(\boldsymbol{x}^{(0)}\right) \neq \boldsymbol{x}^{(0)}$. For any n, m with $m > n$, we have

$$\begin{aligned} \left\|\boldsymbol{x}^{(m)} - \boldsymbol{x}^{(n)}\right\| &= \left\|\boldsymbol{F}^m\left(\boldsymbol{x}^{(0)}\right) - \boldsymbol{F}^n\left(\boldsymbol{x}^{(0)}\right)\right\| \\ &= \left\|\boldsymbol{F}\left(\boldsymbol{F}^{m-1}\left(\boldsymbol{x}^{(0)}\right)\right) - \boldsymbol{F}\left(\boldsymbol{F}^{n-1}\left(\boldsymbol{x}^{(0)}\right)\right)\right\| \\ &\leq L\left\|\boldsymbol{F}^{m-1}\left(\boldsymbol{x}^{(0)}\right) - \boldsymbol{F}^{n-1}\left(\boldsymbol{x}^{(0)}\right)\right\| \\ &\;\;\vdots \\ &\leq L^n\left\|\boldsymbol{x}^{(m-n)} - \boldsymbol{x}^{(0)}\right\| \\ &= L^n\left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^{(m-n)}\right\|. \end{aligned} \tag{A.121}$$

We write

$$\begin{aligned}\left\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(m-n)}\right\| &= \left\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}+\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(2)}+\boldsymbol{x}^{(2)}-\right.\\ &\qquad \left.\cdots-\boldsymbol{x}^{(m-n-1)}+\boldsymbol{x}^{(m-n-1)}-\boldsymbol{x}^{(m-n)}\right\|. \qquad (A.122)\end{aligned}$$

Substituting (A.122) into (A.121) and then applying the triangle inequality yields

$$\left\|\boldsymbol{x}^{(m)}-\boldsymbol{x}^{(n)}\right\| \le L^n\left(\left\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\right\|+\left\|\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(2)}\right\|+\cdots+\left\|\boldsymbol{x}^{(m-n-1)}-\boldsymbol{x}^{(m-n)}\right\|\right). \qquad (A.123)$$

Observe that

$$\begin{aligned}\left\|\boldsymbol{x}^{(j)}-\boldsymbol{x}^{(j+1)}\right\| &= \left\|\boldsymbol{F}\left(\boldsymbol{x}^{(j-1)}\right)-\boldsymbol{F}\left(\boldsymbol{x}^{(j)}\right)\right\|\\ &\le L\left\|\boldsymbol{x}^{(j-1)}-\boldsymbol{x}^{(j)}\right\|\\ &\vdots\\ &\le L^j\left\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\right\|, \qquad (A.124)\end{aligned}$$

We use the above to further evaluate the right hand side of (A.123) to obtain

$$\begin{aligned}\left\|\boldsymbol{x}^{(m)}-\boldsymbol{x}^{(n)}\right\| &\le L^n\left\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\right\|\left(1+L+\cdots+L^{m-n-1}\right)\\ &\le L^n\left\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\right\|\left(1+L+\cdots+L^{m-n-1}+L^{m-n}+\cdots\right)\\ &= \frac{L^n}{1-L}\left\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\right\|\end{aligned}$$

because $0<L<1$. Recall that a sequence $\boldsymbol{x}^{(0)},\boldsymbol{x}^{(1)},\ldots$ is a Cauchy sequence if for each $\varepsilon>0$ there exists a positive integer $n=n(\varepsilon)$ such that $\left\|\boldsymbol{x}^{(m)}-\boldsymbol{x}^{(n)}\right\|<\varepsilon$ for all $m>n$. Given an $\varepsilon>0$, let n be such that

$$L^n<\frac{\varepsilon(1-L)}{\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\|}.$$

Then,

$$\left\|\boldsymbol{x}^{(m)}-\boldsymbol{x}^{(n)}\right\| \le \frac{L^n}{1-L}\left\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\right\|<\varepsilon, \quad \text{for all} \quad m>n. \qquad (A.125)$$

Therefore, the sequence $\boldsymbol{x}^{(0)},\boldsymbol{x}^{(1)},\ldots$ is a Cauchy sequence, and we show that the limit $\boldsymbol{g}$ of this sequence is the fixed point $\boldsymbol{x}^*$ of the map $\boldsymbol{F}$. Note that $\boldsymbol{F}$ is uniformly continuous. Indeed, for each $\varepsilon>0$ there exists a $\delta=\delta(\varepsilon)$ such that if $\|\boldsymbol{x}-\boldsymbol{y}\|<\delta$, then $\|\boldsymbol{F}(\boldsymbol{x})-\boldsymbol{F}(\boldsymbol{y})\|<\varepsilon$. Since $\|\boldsymbol{F}(\boldsymbol{x})-\boldsymbol{F}(\boldsymbol{y})\|\le L\|\boldsymbol{x}-\boldsymbol{y}\|$, given an $\varepsilon>0$, we can choose $\delta=\varepsilon/L$. Because F is continuous,

$$\lim_{n\to\infty}\boldsymbol{F}\left(\boldsymbol{x}^{(n)}\right)=\boldsymbol{F}\left(\lim_{n\to\infty}\boldsymbol{x}^{(n)}\right)=\boldsymbol{F}(\boldsymbol{g}).$$

However, $\boldsymbol{x}^{(n+1)} = \boldsymbol{F}\left(\boldsymbol{x}^{(n)}\right)$. Therefore,

$$\begin{aligned}
\boldsymbol{g} &= \lim_{n\to\infty} \boldsymbol{x}^{(n+1)} \\
&= \lim_{n\to\infty} \boldsymbol{F}\left(\boldsymbol{x}^{(n)}\right) \\
&= \boldsymbol{F}\left(\lim_{n\to\infty} \boldsymbol{x}^{(n)}\right) \\
&= \boldsymbol{F}(\boldsymbol{g}).
\end{aligned}$$

Hence, $\boldsymbol{g} = \boldsymbol{x}^*$ is a fixed point of $\boldsymbol{F}$, and the proof is complete.

□

Using the above proof we can give an estimate of the norm of $\left\| \boldsymbol{x}^{(n)} - \boldsymbol{x}^* \right\|$. Indeed, if we fix n in (A.125) and allow $m \to \infty$, then

$$\left\| \boldsymbol{x}^{(n)} - \boldsymbol{x}^* \right\| \leq \frac{L^n}{1-L} \left\| \boldsymbol{x}^{(0)} - \boldsymbol{F}\left(\boldsymbol{x}^{(0)}\right) \right\| . \tag{A.126}$$

A.13 First-Order Differential Equation

Let $\mathbb{R}$ denote the set of real numbers and let $D \subseteq \mathbb{R}^2$ be a domain, that is, an open connected nonempty subset of $\mathbb{R}^2$. Let f be a real valued function defined and continuous on D. Let $\dot{x} = dx/dt$ denote the derivative of x with respect to t. We call

$$\dot{x} = f(t, x) \tag{A.127}$$

an *ordinary differential equation of the first order*. By a *solution* of the differential equation (A.127) on an open interval $I = \{t \in \mathbb{R} : t_1 < t < t_2\}$, we mean a real valued continuously differentiable function ϕ defined on I such that $(t, \phi(t)) \in D$ for all $t \in I$ and

$$\dot{\phi}(t) = f(t, \phi(t)) \quad \text{for all} \quad t \in I.$$

Definition A.5 *Given* $(t_0, x_0) \in D$*, the* initial value problem *for (A.127) is*

$$\dot{x} = f(t, x), \quad x(t_0) = x_0. \tag{A.128}$$

◇

A function ϕ is a solution to (A.128) if ϕ is a solution of the differential equation (A.127) on some interval I containing t_0 and $x(t_0) = x_0$.

We can represent the initial value problem (A.128) equivalently by an integral equation,

$$\phi(t) = x_0 + \int_{t_0}^{t} f(s, \phi(s))ds. \tag{A.129}$$

To prove this equivalence, let ϕ be a solution of the initial value problem (A.128). Then $x(t_0) = x_0$ and

$$\dot{\phi}(t) = f(t, \phi(t)) \quad \text{for all} \quad t \in I.$$

Integrating the above from t_0 to t, we obtain

$$\int_{t_0}^{t} \dot{\phi}(s)ds = \int_{t_0}^{t} f(s, \phi(s))ds.$$

Hence,

$$\phi(t) - x_0 = \int_{t_0}^{t} f(s, \phi(s))ds.$$

Therefore, ϕ is a solution of the integral equation (A.129).

Conversely, let ϕ be a solution of the integral equation (A.129). Then $\phi(t_0) = x_0$, and differentiating both sides of (A.129) with respect to t gives $\dot{\phi}(t) = f(t, \phi(t))$. Therefore, ϕ is also a solution of the initial value problem (A.128).

The general *ordinary differential linear equation of the first order* has the form

$$\dot{x}(t) = a(t)x(t) + b(t), \qquad t \in (t_1,\ t_2), \tag{A.130}$$

where $a(t)$ and $b(t)$ are assumed to be continuous on the interval $(t_1,\ t_2)$. We associate with equation (A.130) the initial condition

$$x(t_0) = x_0, \qquad x_0 \in \mathbb{R}.$$

To find the solution to (A.130), we write it in the form

$$\dot{x}(t) - a(t)x(t) = b(t). \tag{A.131}$$

We then multiply (A.131) by the "integrating factor"

$$e^{-\int_{t_0}^{t} a(s)ds} \tag{A.132}$$

to get

$$\dot{x}(t)e^{-\int_{t_0}^{t} a(s)ds} - a(t)x(t)e^{-\int_{t_0}^{t} a(s)ds} = e^{-\int_{t_0}^{t} a(s)ds}b(t). \tag{A.133}$$

The above equation can be represented in an equivalent form as

$$\frac{d}{dt}\left\{x(t)e^{-\int_{t_0}^{t} a(s)ds}\right\} = e^{-\int_{t_0}^{t} a(s)ds}b(t). \tag{A.134}$$

Integrating both sides of (A.134) yields

$$x(t)e^{-\int_{t_0}^{t} a(s)ds} - x_0 = \int_{t_0}^{t} e^{-\int_{t_0}^{\tau} a(s)ds}b(\tau)d\tau. \tag{A.135}$$

Hence,

$$x(t) = e^{\int_{t_0}^{t} a(s)ds} \left[x_0 + \int_{t_0}^{t} e^{-\int_{t_0}^{\tau} a(s)ds} b(\tau) d\tau \right], \tag{A.136}$$

or in an equivalent form

$$\boxed{x(t) = e^{\int_{t_0}^{t} a(s)ds} x_0 + \int_{t_0}^{t} e^{\int_{\tau}^{t} a(s)ds} b(\tau) d\tau} \tag{A.137}$$

A.14 Integral and Differential Inequalities

A.14.1 The Bellman-Gronwall Lemma

Lemma A.5 *Let $v(t)$ and $w(t)$ be continuous real functions of t. Let $w(t) \geq 0$, and c be a real constant. If*

$$v(t) \leq c + \int_{0}^{t} w(s)v(s)ds, \tag{A.138}$$

then

$$v(t) \leq ce^{\int_0^t w(s)ds}. \tag{A.139}$$

Proof Denote the right-hand side of (A.138) by $x(t)$, that is, let

$$x(t) = c + \int_{0}^{t} w(s)v(s)ds, \tag{A.140}$$

and let

$$z(t) = x(t) - v(t). \tag{A.141}$$

Note that $z(t) \geq 0$, and $x(t)$ is differentiable. Differentiating both sides of (A.140) and using (A.141) yields

$$\begin{aligned} \dot{x}(t) &= w(t)v(t) \\ &= w(t)x(t) - w(t)z(t). \end{aligned} \tag{A.142}$$

We use (A.137) to represent (A.142) in the equivalent integral form as

$$x(t) = e^{\int_0^t w(s)ds} x_0 - \int_{0}^{t} e^{\int_{\tau}^{t} w(s)ds} w(\tau)z(\tau)d\tau. \tag{A.143}$$

Since $z(t) \geq 0$ and $w(t) \geq 0$, we have

$$x(t) \leq e^{\int_0^t w(s)ds} x_0. \tag{A.144}$$

From the definition of $x(t)$ given by (A.140), we have $x(0) = c$. Since

$$v(t) = x(t) - z(t) \leq x(t), \tag{A.145}$$

because $z(t) \geq 0$, hence,

$$v(t) \leq ce^{\int_0^t w(s)ds},$$

and the proof is completed.

□

The generalized Bellman-Gronwall lemma allows us to convert the investigation of integral equations into integral inequalities.

A.14.2 A Comparison Theorem

We state and prove a theorem that allows one to reduce the study of some problems involving differential equations to the study of corresponding problems using differential inequalities instead.

We consider the differential inequality,

$$\dot{x}(t) \leq \omega(t, x(t)), \qquad x(t_0) = x_0,$$

where $\omega : \mathbb{R}^2 \to \mathbb{R}$ is a continuous function in a domain $\Delta \subset \mathbb{R}^2$. We assume that in Δ, the function ω satisfies the Lipschitz condition,

$$|\omega(t, x) - \omega(t, y)| \leq L|x - y|,$$

where $L \geq 0$ is a Lipschitz constant. Consider an associated differential equation

$$\dot{\tilde{x}}(t) = \omega(t, \tilde{x}(t)), \qquad \tilde{x}(t_0) = \tilde{x}_0.$$

We refer to the above equation as the *equation of comparison*. Denote the solution to the equation of comparison by

$$\tilde{x}(t) = \tilde{x}(t; t_0, \tilde{x}_0).$$

The following theorem, that can be found, for example, in Corduneanu [67, page 49], relates the solutions of the differential inequality and the associated differential equation.

Theorem A.20 *If $x(t)$ is a differentiable solution to the differential inequality and $\tilde{x}(t)$ is the solution to the associated differential equation with the corresponding initial conditions such that*

$$x_0 \leq \tilde{x}_0,$$

then

$$x(t) \leq \tilde{x}(t)$$

for all $t \geq t_0$.

Proof We prove the theorem by contradiction. We suppose that $x_0 \leq \tilde{x}_0$ but $x(t) \leq \tilde{x}(t)$ does not hold. Thus, there exists $t_1 > t_0$ such that

$$x(t_1) > \tilde{x}(t_1).$$

Let

$$y(t) = x(t) - \tilde{x}(t).$$

Note that $y(t_0) = x(t_0) - \tilde{x}(t_0) \leq 0$, and $y(t_1) = x(t_1) - \tilde{x}(t_1) > 0$. Denote by τ the largest $t \in [t_0,\, t_1]$ such that $y(\tau) = 0$. Such a value of t exists by virtue of the continuity of y. Thus,

$$y(t) > 0 \quad \text{for all} \quad t \in (\tau, t_1]\,.$$

On the other hand, for $t \in [\tau, t_1]$, we have

$$\begin{aligned} \dot{y}(t) &= \dot{x}(t) - \dot{\tilde{x}}(t) \\ &\leq \omega(t, x(t)) - \omega(t, \tilde{x}(t)) \\ &\leq |\omega(t, x(t)) - \omega(t, \tilde{x}(t))| \\ &\leq L|x(t) - \tilde{x}(t)|. \end{aligned}$$

In the interval $[\tau, t_1]$, we have

$$y(t) = x(t) - \tilde{x}(t) \geq 0.$$

Hence,

$$\begin{aligned} \dot{y}(t) &\leq L|x(t) - \tilde{x}(t)| \\ &= L(x(t) - \tilde{x}(t)) \\ &= Ly(t). \end{aligned}$$

Therefore, in the interval $[\tau, t_1]$, the function y satisfies

$$y(t) \leq y(\tau)e^{L(t-\tau)} = 0$$

since $y(\tau) = 0$. But this is a contradiction to the previous conclusion that $y(t) > 0$ for all $t \in (\tau, t_1]$, which completes the proof.

□

A.15 The Laplace Transform

The Laplace transform is an operator that transforms a function of time, $f(t)$, into a new function of complex variable, $F(s)$, where $s = \sigma + j\omega$, as illustrated in Figure A.4. The operator $\mathcal{L}$ denotes that the time function $f(t)$ has been transformed to its Laplace transform, denoted $F(s)$. The Laplace transform is very useful in solving linear differential equations

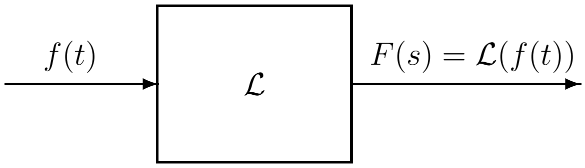


Figure A.4: Schematic representation of the Laplace transform operator.

and hence in analyzing control systems.

To obtain the Laplace transform of the given function of time, $f(t)$,

1. multiply $f(t)$ by a converging factor e^{-st}. This is a factor that decreases to zero as time increases to infinity;

2. Integrate $f(t)e^{-st}$ with respect to time between the limits 0^- and ∞ to obtain the Laplace transform of $f(t)$,

$$\boxed{F(s) = \mathcal{L}(f(t)) = \int_{0^-}^{\infty} f(t)e^{-st}dt}$$

The lower limit of integration is 0^-, rather than 0, to account for the effect of "instantaneous energy transfer," for example, in electric circuits that contain inductors or capacitors.

The above definition of the Laplace transform is also referred to as the *one-sided* or *unilateral* Laplace transform. In the two-sided, or bilateral, Laplace transform, the lower limit is $-\infty$. For our purposes the one-sided Laplace transform is sufficient.

If we wish to reverse the operation and take the inverse transform, back to the time domain, we write

$$\mathcal{L}^{-1}(F(s)) = f(t).$$

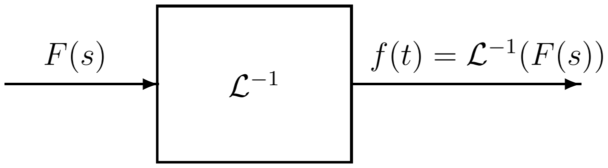


Figure A.5: Schematic representation of the inverse Laplace transform operation.

Taking the inverse Laplace transform is illustrated in Figure A.5.

Because we are using the one-sided Laplace transform, we define all functions, whose Laplace transforms we compute, to be zero for $t < 0^-$. To proceed, we recall the definition of the unit step function denoted $1(t)$,

$$1(t) = \begin{cases} 1 & \text{if} \quad t \geq 0 \\ 0 & \text{if} \quad t < 0. \end{cases}$$

The unit step function is also called the Heaviside function.

Example A.1 Find the Laplace transform of

$$f(t) = e^{-at}1(t),$$

where a is a real constant. A graph of $f(t)$ for $a = 3$ is shown in Figure A.6. We have

$$\mathcal{L}(e^{-at}1(t)) = \int_{0^-}^{\infty} e^{-at}e^{-st}1(t)dt = \int_{0^-}^{\infty} e^{-(a+s)t}dt.$$

The above integral exists if

$$\Re(a + s) > 0.$$

The region of the s-plane for which the Laplace transform exists is called the *Region of Convergence* and abbreviated ROC. Since $s = \sigma + j\omega$, the ROC, in our example, is the region in the s-plane where

$$0 < \Re(a + s) = \Re(a + \sigma + j\omega) = a + \sigma,$$

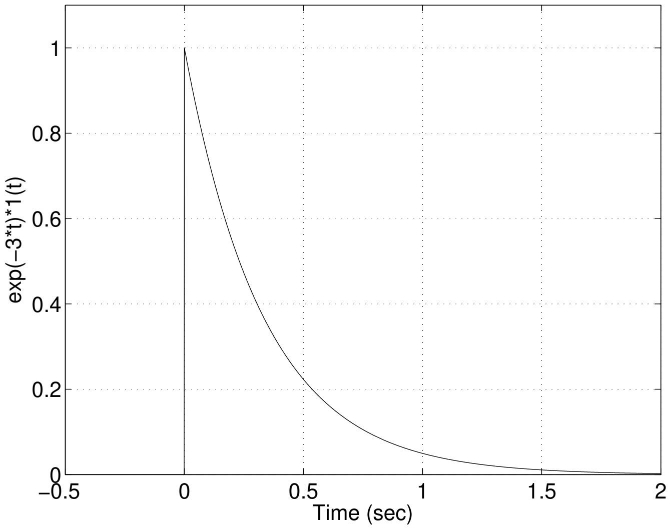


Figure A.6: A plot of $e^{-3t}1(t)$.

that is, the region where $\sigma > -a$. Proceeding with the integration, we obtain

$$\begin{aligned}\mathcal{L}(e^{-at}1(t)) &= \left(-\frac{1}{a+s}e^{-(a+s)t}\right)\Bigg|_{0^-}^{\infty} \\ &= \left(-\frac{1}{a+s}e^{-(a+s)\infty}\right) - \left(-\frac{1}{a+s}e^{-(a+s)0^-}\right).\end{aligned}$$

For s in the ROC, the first term tends to zero. Hence,

$$\boxed{\mathcal{L}(e^{-at}1(t)) = \frac{1}{s+a}}$$

Using the Laplace transform of the exponential function, we can easily find the Laplace transform of the unit step. Indeed, if $a = 0$, then $f(t) = e^{-at}1(t) = 1(t)$. Hence,

$$\boxed{\mathcal{L}(1(t)) = \frac{1}{s}} \tag{A.146}$$

A.15.1 Linearity Property of the Laplace Transform

The Laplace transform of the sum, or difference, of two functions of time is equal to the sum, or difference, of the transforms of each function, that is,

$$\mathcal{L}\left(f_1(t) \pm f_2(t)\right) = \mathcal{L}\left(f_1(t)\right) \pm \mathcal{L}\left(f_2(t)\right) \tag{A.147}$$

Indeed,

$$\begin{aligned}\mathcal{L}\left(f_1(t) \pm f_2(t)\right) &= \int_{0^-}^{\infty} \left(f_1(t) \pm f_2(t)\right) e^{-st} dt \\ &= \int_{0^-}^{\infty} f_1(t) e^{-st} dt \pm \int_{0^-}^{\infty} f_2(t) e^{-st} dt \\ &= \mathcal{L}\left(f_1(t)\right) \pm \mathcal{L}\left(f_2(t)\right).\end{aligned}$$

The Laplace transform of the product of a real or complex constant K and a time function $f(t)$ is equal to the product of the constant and the transform of the time function, that is,

$$\mathcal{L}\left(Kf(t)\right) = K\mathcal{L}\left(f(t)\right). \tag{A.148}$$

Indeed,

$$\mathcal{L}\left(Kf(t)\right) = \int_{0^-}^{\infty} Kf(t) e^{-st} dt = K \int_{0^-}^{\infty} f(t) e^{-st} dt = K\mathcal{L}\left(f(t)\right).$$

The above two properties can be represented in the form

$$\boxed{\mathcal{L}\left(Kf_1(t) \pm f_2(t)\right) = K\mathcal{L}\left(f_1(t)\right) \pm \mathcal{L}\left(f_2(t)\right)}$$

Thus the Laplace transform is a linear operator.

Example A.2 To find the Laplace transform of

$$f(t) = K1(t),$$

where K is a constant, we can use (A.148) and the Laplace transform of the unit step, given by (A.146), to obtain

$$\mathcal{L}\left(K1(t)\right) = K\mathcal{L}\left(1(t)\right) = \frac{K}{s}.$$

Example A.3 We use the linear property of the Laplace transform to find the Laplace transform of

$$f(t) = \sin \omega t.$$

First, recall the Euler formula,

$$e^{j\omega t} = \cos \omega t + j \sin \omega t.$$

Hence,

$$e^{-j\omega t} = \cos \omega t - j \sin \omega t.$$

Subtracting the second of the above expressions from the first one and dividing the result by $2j$ gives

$$\sin\omega t = \frac{e^{j\omega t} - e^{-j\omega t}}{2j}.$$

Therefore,

$$\mathcal{L}(\sin\omega t) = \mathcal{L}\left(\frac{e^{j\omega t} - e^{-j\omega t}}{2j}\right) = \mathcal{L}\left(\frac{1}{2j}\left(e^{j\omega t} - e^{-j\omega t}\right)\right).$$

Applying to the above (A.148) yields

$$\mathcal{L}(\sin\omega t) = \frac{1}{2j}\mathcal{L}\left(e^{j\omega t} - e^{-j\omega t}\right).$$

Applying now (A.147), we obtain

$$\mathcal{L}(\sin\omega t) = \frac{1}{2j}\left(\mathcal{L}\left(e^{j\omega t}\right) - \mathcal{L}\left(e^{-j\omega t}\right)\right).$$

We then use twice the formula for the Laplace transform of the exponential to get

$$\mathcal{L}(\sin\omega t) = \frac{1}{2j}\left(\frac{1}{s - j\omega} - \frac{1}{s + j\omega}\right).$$

Performing algebraic manipulations gives

$$\boxed{\mathcal{L}(\sin\omega t) = \frac{\omega}{s^2 + \omega^2}}$$

Similarly, we obtain

$$\mathcal{L}(\cos\omega t) = \mathcal{L}\left(\frac{e^{j\omega t} + e^{-j\omega t}}{2}\right).$$

Hence,

$$\boxed{\mathcal{L}(\cos\omega t) = \frac{s}{s^2 + \omega^2}}$$

A.15.2 Laplace Transforms of Derivatives

We will now show that

$$\boxed{\mathcal{L}\left(\frac{df(t)}{dt}\right) = sF(s) - f\left(0^-\right)}$$

where $F(s) = \mathcal{L}(f(t))$ and $f\left(0^-\right)$ is the initial value of f, that is, the value of f at 0^-. Thus, differentiating in the time domain corresponds to multiplying $F(s)$ by s and then subtracting the initial value of $f(t)$. To derive the above formula, we apply the definition of the Laplace transform,

$$\mathcal{L}\left(\frac{df(t)}{dt}\right) = \int_{0^-}^{\infty} \frac{df(t)}{dt} e^{-st} dt, \tag{A.149}$$

and then evaluate the integral by integrating by parts. Recall that the formula for integrating by parts can easily be derived from the formula for the derivative of the product of two functions. We use the notation,

$$u' = \frac{du(t)}{dt}.$$

Then, we have

$$(uv)' = u'v + uv'.$$

Integrating the above and rearranging gives

$$\int u'v = uv - \int uv'. \tag{A.150}$$

We let

$$u' = \frac{df}{dt} \quad \text{and} \quad v = e^{-st}.$$

Then applying (A.150) to (A.149), we obtain

$$\mathcal{L}\left(\frac{df(t)}{dt}\right) = e^{-st} f(t) \,\Big|_{0^-}^{\infty} - \int_{0^-}^{\infty} f(t) \left(-se^{-st}\right) dt.$$

Assuming that $f(t)$ is Laplace transformable, the value of $e^{-st}f(t)$ at $t = \infty$ is zero. Hence, the right-hand side of the above reduces to

$$\mathcal{L}\left(\frac{df(t)}{dt}\right) = -f\left(0^-\right) + s\int_{0^-}^{\infty} f(t)e^{-st}dt = sF(s) - f\left(0^-\right).$$

Thus, we showed that differentiation in the time domain corresponds to an algebraic operation in the s domain.

Using the Laplace transform of the first derivative of a time function, we can easily determine the Laplace transform of higher-order derivatives. Indeed, to find the Laplace transform of $\frac{d^2 f(t)}{dt^2}$, we let

$$g(t) = \frac{df(t)}{dt}.$$

Then,

$$\mathcal{L}(g(t)) = G(s) = \mathcal{L}\left(\frac{df(t)}{dt}\right) = sF(s) - f\left(0^-\right). \tag{A.151}$$

Note that

$$\frac{d^2 f(t)}{dt^2} = \frac{dg(t)}{dt}.$$

Hence,

$$\mathcal{L}\left(\frac{df^2(t)}{dt^2}\right) = \mathcal{L}\left(\frac{dg(t)}{dt}\right) = sG(s) - g\left(0^-\right).$$

Substituting into the above (A.151) gives

$$\mathcal{L}\left(\frac{df^2(t)}{dt^2}\right) = s\left(sF(s) - f\left(0^-\right)\right) - g\left(0^-\right).$$

Observing that $g\left(0^-\right) = \frac{df\left(0^-\right)}{dt}$, we get

$$\boxed{\mathcal{L}\left(\frac{df^2(t)}{dt^2}\right) = s^2F(s) - sf\left(0^-\right) - \frac{df\left(0^-\right)}{dt}}$$

Successively applying the above arguments, we can obtain the Laplace transform of the n-th time derivative,

$$\boxed{\mathcal{L}\left(\frac{df^n(t)}{dt^n}\right) = s^nF(s) - s^{n-1}f\left(0^-\right) - s^{n-2}\frac{df\left(0^-\right)}{dt} - \cdots - s\frac{d^{n-2}f\left(0^-\right)}{dt^{n-2}} - \frac{d^{n-1}f\left(0^-\right)}{dt^{n-1}}}$$

A.15.3 Solving Differential Equations Using Laplace Transform

We now illustrate how to use the Laplace transform and inverse Laplace transform to solve linear ordinary differential equations. After taking the Laplace transform of both sides of a differential equation and performing required manipulations in the s domain, we need to reconstruct the solution in the time domain. This is achieved using the inverse Laplace transform.

Example A.4 We will solve the following differential equation using the Laplace transform,

$$2\frac{d^2x(t)}{dt^2} + 7\frac{dx(t)}{dt} + 5x(t) = 10 \times 1(t),$$

subject to the initial conditions

$$\frac{dx\left(0^-\right)}{dt} = 1, \quad x\left(0^-\right) = -2,$$

where $1(t)$ is the unit step function.

We take the Laplace transform of both sides of the differential equation to get

$$2\mathcal{L}\left(\frac{d^2x(t)}{dt^2}\right) + 7\mathcal{L}\left(\frac{dx(t)}{dt}\right) + 5\mathcal{L}\left(x(t)\right) = 10\mathcal{L}\left(1(t)\right).$$

Using Laplace transforms of time derivatives of a function of time and the Laplace transform of the unit step function, we obtain

$$2\left(s^2X(s) - sx\left(0^-\right) - \frac{dx\left(0^-\right)}{dt}\right) + 7\left(sX(s) - x\left(0^-\right)\right) + 5X(s) = \frac{10}{s}.$$

Substituting into the above the initial conditions and rearranging gives

$$\left(2s^2 + 7s + 5\right) X(s) = \frac{10}{s} - 4s - 12 = \frac{10 - 4s^2 - 12s}{s}.$$

Hence,

$$X(s) = \frac{-4s^2 - 12s + 10}{s\left(2s^2 + 7s + 5\right)} = \frac{-2s^2 - 6s + 5}{s\left(s^2 + 3.5s + 2.5\right)} = \frac{-2s^2 - 6s + 5}{s\left(s + 2.5\right)\left(s + 1\right)}. \tag{A.152}$$

We now need to reconstruct the solution of the differential equation in the time domain. In other words, given $X(s)$ we wish to obtain $x(t)$ in which $x(t)$ is zero for $t < 0$ such that $X(s) = \mathcal{L}(x(t))$. Thus,

$$x(t) = \mathcal{L}^{-1}\left(X(s)\right).$$

To proceed, note that $X(s)$ as given by (A.152) is a rational function of s, that is, $X(s)$ is a ratio of two polynomials in s, in which the degree of the numerator with respect to s is smaller than the degree of the denominator. Such a rational function is called strictly proper and can be expanded into a sum of partial fractions by writing a term or a series of terms for each zero of the denominator. We obtain

$$X(s) = \frac{K_1}{s} + \frac{K_2}{s + 2.5} + \frac{K_3}{s + 1}. \tag{A.153}$$

We next compute the constants K_i's. We can do so by representing the right-hand side of the above as a rational function and then comparing the resulting numerator with the numerator of (A.152). We represent (A.153) as

$$\begin{aligned} X(s) &= \frac{K_1\left(s + 2.5\right)\left(s + 1\right) + K_2 s\left(s + 1\right) + K_3 s\left(s + 2.5\right)}{s\left(s + 2.5\right)\left(s + 1\right)} \\ &= \frac{\left(K_1 + K_2 + K_3\right)s^2 + \left(3.5K_1 + K_2 + 2.5K_3\right)s + 2.5K_1}{s\left(s + 2.5\right)\left(s + 1\right)} \end{aligned} \tag{A.154}$$

Comparing coefficients of like powers of the numerators of (A.152) and (A.154) gives three algebraic equations in three unknowns,

$$\left.\begin{aligned} K_1 + K_2 + K_3 &= -2 \\ 3.5K_1 + K_2 + 2.5K_3 &= -6 \\ 5 &= 2.5K_1 \end{aligned}\right\}$$

Solving the above system of equations gives

$$K_1 = 2, \quad K_2 = 2, \quad K_3 = -6.$$

Substituting the above into (A.153) yields

$$X(s) = \frac{2}{s} + \frac{2}{s+2.5} - \frac{6}{s+1}.$$

Applying the inverse Laplace transform to the above, we obtain

$$\begin{aligned} x(t) &= \mathcal{L}\left(\frac{2}{s} + \frac{2}{s+2.5} - \frac{6}{s+1}\right) \\ &= \mathcal{L}\left(\frac{2}{s}\right) + \mathcal{L}\left(\frac{2}{s+2.5}\right) - \mathcal{L}\left(\frac{6}{s+1}\right) \\ &= \left(2 + 2e^{-2.5t} - 6e^{-t}\right) 1(t). \end{aligned}$$

A.15.4 Laplace Transforms of Integrals

We first consider taking the Laplace transform of

$$\int_{0^-}^{t} f(x)dx. \tag{A.155}$$

Let $F(s)$ denote the Laplace transform of $f(t)$, that is, $F(s) = \mathcal{L}(f(t))$. We find the Laplace transform of (A.155) using integration by parts to obtain

$$\mathcal{L}\left(\int_{0^-}^{t} f(x)dx\right) = \int_{0^-}^{\infty}\left(\int_{0^-}^{t} f(x)dx\right)e^{-st}dt. \tag{A.156}$$

Recall the formula for integration by parts,

$$\int u'v = uv - \int uv'.$$

Let

$$u = -\frac{1}{s}e^{-st} \quad \text{and} \quad v = \int_{0^-}^{t} f(x)dx.$$

Note that

$$\frac{dv}{dt} = \frac{d}{dt}\int_{0^-}^{t} f(x)dx = f(t).$$

Integrating (A.156) by parts gives

$$\mathcal{L}\left(\int_{0^-}^{t} f(x)dx\right) = -\frac{1}{s}e^{-st}\int_{0^-}^{t} f(x)dx\,\bigg|_{0^-}^{\infty} + \frac{1}{s}\int_{0^-}^{\infty} e^{-st}f(t)dt.$$

The first term on the right-hand side is zero at both the lower and upper limits. The value at the lower limit, $t = 0^-$, is clearly zero. The value at the upper limit, $t = \infty$, is zero because we assumed that $f(t)$ is Laplace transformable. Hence,

$$\boxed{\mathcal{L}\left(\int_{0^-}^{t} f(x)dx\right) = \frac{F(s)}{s}} \tag{A.157}$$

We next find the Laplace transform of

$$\int_{-\infty}^{t} f(x)dx. \tag{A.158}$$

Applying the definition and using the fact that the Laplace transform is linear yields

$$\mathcal{L}\left(\int_{-\infty}^{t} f(x)dx\right) = \mathcal{L}\left(\int_{-\infty}^{0^-} f(x)dx\right) + \mathcal{L}\left(\int_{0^-}^{t} f(x)dx\right) = F_1(s) + F_2(s).$$

The limits of the first integral are constant, therefore the integral will be a constant. Hence,

$$F_1(s) = \frac{\int_{-\infty}^{0^-} f(x)dx}{s},$$

while by (A.157),

$$F_2(s) = \frac{F(s)}{s}.$$

Therefore,

$$\boxed{\mathcal{L}\left(\int_{-\infty}^{t} f(x)dx\right) = \frac{F(s)}{s} + \frac{\int_{-\infty}^{0^-} f(x)dx}{s}} \tag{A.159}$$

Example A.5 We find the Laplace transform of the voltage, $v_C(t)$, across a capacitor C. We have

$$v_C(t) = \frac{1}{C}\int_{-\infty}^{t} i(x)dx.$$

Using (A.159) gives

$$\mathcal{L}\left(v_C(t)\right) = \frac{1}{C}\frac{I(s)}{s} + \frac{\frac{1}{C}\int_{-\infty}^{0^-} i(x)dx}{s}.$$

Since $\int_{-\infty}^{0^-} i(x)dx$ is the charge, q, on the capacitor at $t = 0^-$ and $v = q/C$, we obtain

$$\boxed{\mathcal{L}\left(v_C(t)\right) = \frac{1}{C}\frac{I(s)}{s} + \frac{v_C\left(0^-\right)}{s}}$$

More Properties of the Laplace Transform

We will now show that translation in the time domain corresponds to multiplication by an exponential in the s domain, that is,

$$\boxed{\mathcal{L}\left(f(t-a)1(t-a)\right) = e^{-sa}F(s), \qquad a > 0}$$

where $F(s) = \mathcal{L}(f(t))$. We have

$$\mathcal{L}\left(f(t-a)1(t-a)\right) = \int_{0^-}^{\infty} f(t-a)1(t-a)e^{-st}dt = \int_{a^-}^{\infty} f(t-a)e^{-st}dt,$$

because $1(t-a) = 0$ for $t < a$. Next, we change the variable of integration. We let

$$x = t - a.$$

Then, $x = 0^-$ when $t = a^-$, $x = \infty$ when $t = \infty$, and $dx = dt$. Substituting the above into (A.160) yields

$$\mathcal{L}\left(f(t-a)1(t-a)\right) = \int_{0^-}^{\infty} f(x)e^{-s(x+a)}dx = e^{-sa}\int_{0^-}^{\infty} f(x)e^{-sx}dx = e^{-as}F(s),$$

which is the desired result.

Next, we prove the time/frequency scaling property,

$$\boxed{\mathcal{L}(f(at)) = \frac{1}{a}F\left(\frac{s}{a}\right), \quad a > 0}$$

Indeed, applying the Laplace transform yields

$$\mathcal{L}(f(at)) = \int_{0^-}^{\infty} f(at)e^{-st}dt. \tag{A.160}$$

Let $x = at$. Then, $dt = \frac{1}{a}dx$. Substituting the above into (A.160) gives

$$\mathcal{L}(f(at)) = \frac{1}{a}\int_{0^-}^{\infty} f(x)e^{-\left(\frac{s}{a}\right)x}dx = \frac{1}{a}F\left(\frac{s}{a}\right).$$

Using the sifting property, given by (1.1) on page 15, we show that

$$\boxed{\mathcal{L}(\delta(t)) = 1}$$

Indeed, applying the definition of the Laplace transform and using the sifting property (1.1), we obtain

$$\mathcal{L}(\delta(t)) = \int_{0^-}^{\infty} \delta(t)e^{-st}dt = e^0\int_{0^-}^{0^+} \delta(t)dt = 1.$$

A.16 The $\mathcal{Z}$-Transform

A.16.1 Definition and Simple Examples

The $\mathcal{Z}$-transform is an operator denoted $\mathcal{Z}(\cdot)$ that, when applied to a sequence of numbers produces a function of a variable z.

Definition A.6 *For a given sequence of numbers*

$$y(0), y(1), \ldots, y(k), \ldots$$

its $\mathcal{Z}$-transform is the series defined as

$$Y(z) = \mathcal{Z}(y(k)) = \sum_{k=0}^{\infty} \frac{y(k)}{z^k}.$$

◇

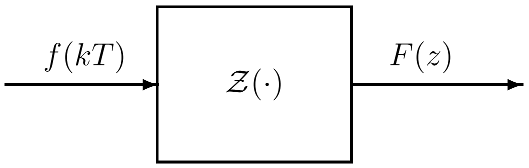


Figure A.7: The $\mathcal{Z}$-transform is an operator acting on a sequence of numbers resulting in a function of the variable z.

Example A.6 We find the $\mathcal{Z}$-transform of the sampled unit step, $f(kT) = 1$. The $\mathcal{Z}$-transform of this sequence is

$$\begin{aligned} F(z) &= \sum_{k=0}^{\infty} f(kT) z^{-k} \\ &= 1 + \frac{1}{z} + \frac{1}{z^2} + \cdots \end{aligned}$$

If z is a complex variable such that $|z| > 1$, then the above sequence converges and we have

$$\begin{aligned} F(z) &= \frac{1}{1 - 1/z} \\ &= \frac{z}{z-1}. \end{aligned}$$

Example A.7 We find the $\mathcal{Z}$-transform of the sampled unit ramp, $f(kT) = kT$. The $\mathcal{Z}$-transform of this sequence is

$$\begin{aligned} F(z) &= \sum_{k=0}^{\infty} kTz^{-k} \\ &= T\left(z^{-1} + 2z^{-2} + 3z^{-3} + \cdots\right) \end{aligned} \tag{A.161}$$

Multiplying both sides of the above by z gives

$$zF(z) = T\left(1 + 2z^{-1} + 3z^{-2} + \cdots\right) \tag{A.162}$$

Subtracting (A.161) from (A.162) yields

$$\begin{aligned} zF(z) - F(z) &= (z-1)F(z) \\ &= T\left(1 + z^{-1} + z^{-2} + \cdots\right) \\ &= \frac{1}{1 - z^{-1}} \\ &= \frac{z}{z-1}. \end{aligned} \tag{A.163}$$

Hence, the $\mathcal{Z}$-transform of $f(kT) = kT$ is

$$\boxed{\mathcal{Z}(kT) = T\frac{z}{(z-1)^2}} \tag{A.164}$$

Example A.8 We now find the $\mathcal{Z}$-transform of the geometric sequence

$$f(k) = a^{kT}. \tag{A.165}$$

We obtain

$$\boxed{\mathcal{Z}(a^{kT}) = \sum_{k=0}^{\infty} \frac{a^{kT}}{z^k} = \frac{z}{z - a^T}} \tag{A.166}$$

for $|z| > \left|a^T\right|$.

A.16.2 Some Properties of the $\mathcal{Z}$-Transform

We analyze now three basic properties that are especially useful when computing transforms of more complicated sequences.

Linearity property: The $\mathcal{Z}$-transform is a linear operator, that is,

(i) if $F_1(z) = \mathcal{Z}(f_1(kT))$ and $F_2(z) = \mathcal{Z}(f_2(kT))$ then

$$\mathcal{Z}(f_1(kT) + f_2(kT)) = F_1(z) + F_2(z); \text{ and}$$

(ii) if $F(z) = \mathcal{Z}(f(kT))$ then

$$\mathcal{Z}(cf(kT)) = cF(z),$$

where c is a real or complex number.

We represent the above two properties as: if $F_1(z) = \mathcal{Z}(f_1(kT))$ and $F_2(z) = \mathcal{Z}(f_2(kT))$, and c is a real or complex number, then

$$\mathcal{Z}(f_1(kT) + cf_2(kT)) = F_1(z) + cF_2(z).$$

$\mathcal{Z}$-transform of a delayed sequence: If the sequence $f(kT)$ has the $\mathcal{Z}$-transform $F(z)$, then the unit delayed sequence

$$g(kT) = \begin{cases} f((k-1)T) & k \geq 1 \\ 0 & k = 0 \end{cases}$$

has the $\mathcal{Z}$-transform

$$G(z) = z^{-1}F(z). \tag{A.167}$$

We use the definition of the $\mathcal{Z}$-transform to prove the above property;

$$\begin{aligned} G(z) &= \sum_{k=0}^{\infty} z^{-k} g(kT) \\ &= \sum_{k=1}^{\infty} z^{-k} f((k-1)T) \\ &= z^{-1} \sum_{k=1}^{\infty} z^{-(k-1)} f((k-1)T) \\ &= z^{-1} \sum_{j=0}^{\infty} z^{-j} f(jT) \\ &= z^{-1} F(z). \end{aligned}$$

$\mathcal{Z}$-transform of an advanced sequence: If the sequence $f(kT)$ has the $\mathcal{Z}$-transform $F(z)$, then the unit advanced sequence $h(kT) = f((k+1)T)$, $k \geq 0$, has the transform

$$\mathcal{Z}(f((k+1)T)) = zF(z) - zf(0). \tag{A.168}$$

To prove the above, we apply the definition of the $\mathcal{Z}$-transform to obtain,

$$\begin{aligned} H(z) &= \sum_{k=0}^{\infty} z^{-k} h(kT) \\ &= \sum_{k=0}^{\infty} z^{-k} f((k+1)T) \\ &= z\sum_{k=0}^{\infty} z^{-k-1} f((k+1)T) \\ &= z\sum_{j=1}^{\infty} z^{-j} f(jT) \\ &= z\sum_{j=0}^{\infty} z^{-j} f(jT) - zf(0) \\ &= zF(z) - zf(0). \end{aligned}$$

Example A.9 We find the $\mathcal{Z}$-transform of the delayed geometric sequence,

$$f(kT) = \begin{cases} 0 & \text{for} \quad k = 0 \\ a^{(k-1)T} & \text{for} \quad k > 0. \end{cases} \tag{A.169}$$

Applying (A.167) to the geometric sequence (A.165) yields

$$\boxed{\mathcal{Z}(f(kT) = \frac{1}{z - a^T}} \tag{A.170}$$

Solving Difference Equations Using the $\mathcal{Z}$-Transform

The $\mathcal{Z}$-transform is a very effective tool for solving linear, constant-coefficients difference equations. The basis for the method is the $\mathcal{Z}$-transform of an advanced sequence given by (A.168). Repeated application of this property gives

$$\mathcal{Z}(f((k+2)T)) = z^2F(z) - z^2f(0) - zf(T). \tag{A.171}$$

Applying again (A.168) to (A.171) gives

$$\mathcal{Z}(f((k+3)T)) = z^3F(z) - z^3f(0) - z^2f(T) - zf(2T) \tag{A.172}$$

and so on.

Example A.10 We solve the difference equation

$$x[k+1] + 3x[k] = 2^k, \quad k = 0, 1, \ldots,$$

where $x(0) = 0$. Applying the $\mathcal{Z}$-transform to both sides of the above difference equation gives

$$zX(z) - zx(0) + 3X(z) = \frac{z}{z-2}.$$

Hence,

$$X(z) = \frac{z}{(z+3)(z-2)},$$

which is a strictly proper rational function of z. We represent $X(z)$ in partial fraction form as

$$X(z) = \frac{A}{z+3} + \frac{B}{z-2},$$

where

$$A = (z+3)X(z)\big|_{z=-3} = \frac{3}{5} \quad \text{and} \quad B = (z-2)X(z)\big|_{z=2} = \frac{2}{5}.$$

Therefore,

$$X(z) = \frac{3/5}{z+3} + \frac{2/5}{z-2}.$$

We use (A.170) when applying the inverse $\mathcal{Z}$-transform to the above,

$$x[k] = \begin{cases} 0 & \text{for} \quad k = 0 \\ \frac{3}{5}(-3)^{k-1} + \frac{2}{5}2^{k-1} & \text{for} \quad k \geq 1 \end{cases}$$

Example A.11 We solve the difference equation

$$x[k+1] + 3x[k] = 2^k, \quad k = 0, 1, \ldots,$$

where $x(0) = 5$. Applying the $\mathcal{Z}$-transform to both sides of the above difference equation gives

$$zX(z) - zx(0) + 3X(z) = \frac{z}{z-2}.$$

Hence,

$$X(z) = \frac{5z^2 - 9z}{(z+3)(z-2)} = \frac{z(5z-9)}{(z+3)(z-2)}.$$

The above rational function is not strictly proper so we use a different approach than in Example A.10. In this example z is a factor of the numerator so we can represent the above as

$$\frac{X(z)}{z} = \frac{5z-9}{(z+3)(z-2)}$$

and express the right hand side in partial fraction format as

$$\frac{X(z)}{z} = \frac{24/5}{z+3} + \frac{1/5}{z-2}.$$

Thus

$$X(z) = \frac{24}{5}\frac{z}{z+3} + \frac{1}{5}\frac{z}{z-2}.$$

Taking into account (A.166) when applying the inverse $\mathcal{Z}$-transform yields

$$x[k] = \frac{24}{5}(-3)^k + \frac{1}{5}2^k, \quad k \geq 0$$

A.17 Curves and Surfaces

In the following, we review selected results from calculus of several variables concerning curves, surfaces, vector fields, and curve integrals.

Consider a point moving along a curve in n-dimensional space. The position of the point at time t can be described as,

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}. \tag{A.173}$$

For example, if the point is moving along a straight line passing through fixed points $\boldsymbol{y}$ and $\boldsymbol{z}$ in $\mathbb{R}^n$, then

$$\boldsymbol{x}(t) = \boldsymbol{y} + t\boldsymbol{z}. \tag{A.174}$$

As another example, consider

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} \in \mathbb{R}^2. \tag{A.175}$$

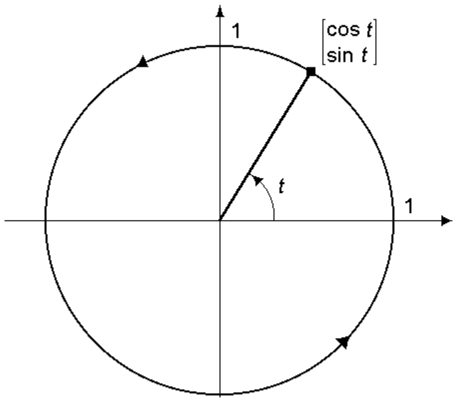


Figure A.8: An example of a curve in two-dimensional space.

Then, the point moves around a circle of radius 1 in counterclockwise direction as shown in Figure A.8. We used t as the variable corresponding to the angle corresponding to the position of the point on the circle. We are now ready for a formal definition of a curve.

Definition A.7 *Let $I \in \mathbb{R}$ be an interval. A parameterized curve defined on the interval I is an association which to each point of I associates a vector, where $\boldsymbol{x}(t)$ denotes the vector associated to $t \in I$ by $\boldsymbol{x}$. We write the association $t \mapsto \boldsymbol{x}(t)$ as*

$$\boldsymbol{x} : I \to \mathbb{R}^n.$$

We also call this association the parameterization of a curve. We call $\boldsymbol{x}(t)$, the position vector at time t.

$\diamond$

The position vector can be represented in terms of its coordinates as in (A.173), where each component $x_i(t)$ is a function of time t. We say that this curve is differentiable if each function $x_i(t)$ is a differentiable function of time t. We define the derivative $d\boldsymbol{x}(t)/dt$ to be

$$\dot{\boldsymbol{x}}(t) = \frac{d\boldsymbol{x}(t)}{dt} = \begin{bmatrix} \frac{dx_1(t)}{dt} \\ \frac{dx_2(t)}{dt} \\ \vdots \\ \frac{dx_n(t)}{dt} \end{bmatrix},$$

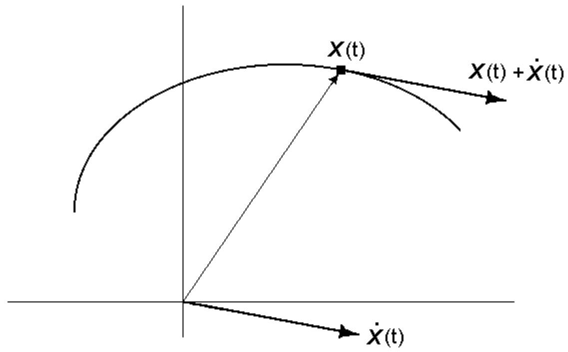


Figure A.9: Velocity vector of a curve at time t is parallel to a tangent vector at time t.

and call it the velocity vector of the curve at time t. The velocity vector is located at the origin. However, when we translate it to the point $\boldsymbol{x}(t)$, as in Figure A.9, then we visualize it as tangent to the curve $\boldsymbol{x}(t)$.

We define the tangent line to a curve $\boldsymbol{x}$ at time t to be the line passing through $\boldsymbol{x}(t)$ in the direction of $\dot{\boldsymbol{x}}(t)$, provided $\dot{\boldsymbol{x}}(t) \neq \mathbf{0}$. If $\dot{\boldsymbol{x}}(t) = \mathbf{0}$, we do not define a tangent line. The speed of the curve $\boldsymbol{x}(t)$, denoted $v(t)$, is

$$v(t) = \|\dot{\boldsymbol{x}}(t)\|.$$

To illustrate the above, we consider the point moving on the circle. The point position at time t is

$$\boldsymbol{x}(t) = \begin{bmatrix} \cos t & \sin t \end{bmatrix}^\top.$$

The velocity vector of the curve at time t is

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} -\sin t & \cos t \end{bmatrix}^\top,$$

and the speed of the curve $\boldsymbol{x}(t)$ at time t is

$$v(t) = \|\dot{\boldsymbol{x}}(t)\| = \sqrt{(-\sin t)^2 + \cos^2 t} = 1.$$

The set of points in $\mathbb{R}^3$ such that for some $F : \mathbb{R}^3 \to \mathbb{R}$, we have

$$F(\boldsymbol{x}) = F(x_1, x_2, x_3) = 0,$$

and

$$DF(x_1, x_2, x_3) = \nabla F(x_1, x_2, x_3)^\top = \begin{bmatrix} \frac{\partial F}{\partial x_1} & \frac{\partial F}{\partial x_2} & \frac{\partial F}{\partial x_3} \end{bmatrix} \neq \mathbf{0}^\top,$$

is called a surface in $\mathbb{R}^3$, where $\nabla\, F = \operatorname{grad} F$. For example, let

$$F(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 - 1,$$

then, the surface is the sphere of radius 1, centered at the origin of $\mathbb{R}^3$.

Let

$$\boldsymbol{C}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_n(t) \end{bmatrix}^\top$$

be a differentiable curve. We say that the curve lies on the surface described by

$$\left.\begin{array}{rcl} F_1(x_1, x_2, \ldots, x_n) & = & 0 \\ & \vdots & \\ F_m(x_1, x_2, \ldots, x_n) & = & 0, \end{array}\right\}$$

if, for all t, we have

$$\left.\begin{array}{rcl} F_1(x_1(t), x_2(t), \ldots, x_n(t)) & = & 0 \\ & \vdots & \\ F_m(x_1(t), x_2(t), \ldots, x_n(t)) & = & 0. \end{array}\right\}$$

Differentiating the above equations, using the chain rule, we obtain

$$DF_i\left(\boldsymbol{C}(t)\right)\dot{\boldsymbol{C}}(t) = \nabla F_i\left(\boldsymbol{C}(t)\right)^\top \dot{\boldsymbol{C}}(t) = 0, \quad i = 1, 2, \ldots, m.$$

Let $\boldsymbol{x}_0$ be a point of the surface, and let $\boldsymbol{C}(t)$ be a curve on the surface passing through $\boldsymbol{x}_0$, that is, for some t_0 we have

$$\boldsymbol{C}(t_0) = \boldsymbol{x}_0.$$

For this t_0 we have

$$\nabla F_i\left(\boldsymbol{x}_0\right)^\top \dot{\boldsymbol{C}}(t_0) = 0, \quad i = 1, 2, \ldots, m,$$

where we assumed that $\dot{\boldsymbol{C}}(t_0) \neq \mathbf{0}$. Thus, the gradients ∇F_i at $\boldsymbol{x}_0$ are orthogonal to the tangent vector of the curve at the point $\boldsymbol{x}_0$. This is true for every differentiable curve on the surface passing through $\boldsymbol{x}_0$. For an illustration of the above properties in three-dimensional space, we refer to Figure A.10.

Definition A.8 *The tangent space to the surface $\{\boldsymbol{x} : \boldsymbol{F}(\boldsymbol{x}) = \mathbf{0}\}$ at the point $\boldsymbol{x}_0$ is the set of vectors at $\boldsymbol{x}_0$ that are tangent to differentiable curves on the surface passing through $\boldsymbol{x}_0$.*

◇

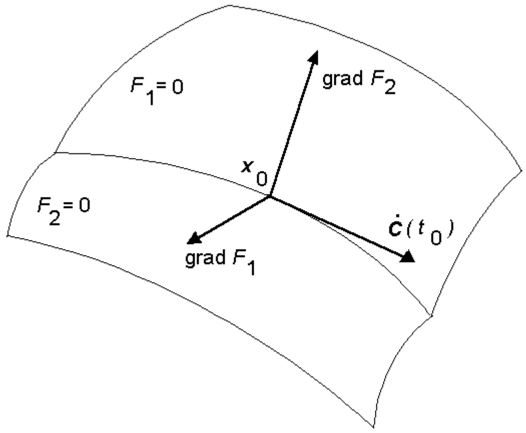


Figure A.10: Tangent space to a surface is the set of vectors orthogonal to the gradients of the functions defining the surface.

Vector Fields and Curve Integrals

Definition A.9 *Let $\mathcal{U}$ be an open subset of $\mathbb{R}^n$. By a vector field on $\mathcal{U}$ we mean an association*

$$\boldsymbol{F} : \mathcal{U} \to \mathbb{R}^n$$

which to every point of $\mathcal{U}$ associates a vector of the same dimension.

$\diamond$

We can interpret a vector field as a field of forces. We can visualize a vector field as a field of arrows, which to each point associates an arrow as shown in Figure A.11. If $f : \mathcal{U} \to \mathbb{R}$ is a function, then

$$\boldsymbol{F} = \nabla f$$

is a vector field, and we have

$$\boldsymbol{F}(\boldsymbol{x}) = \nabla f(\boldsymbol{x})$$

for all $\boldsymbol{x} \in \mathcal{U}$. We now consider an inverse problem: When is a given vector field in $\mathbb{R}^n$ a gradient?

Definition A.10 *Let $\boldsymbol{F}$ be a vector field on an open set $\mathcal{U}$. If f is a differentiable function on $\mathcal{U}$ such that*

$$\boldsymbol{F} = \nabla f,$$

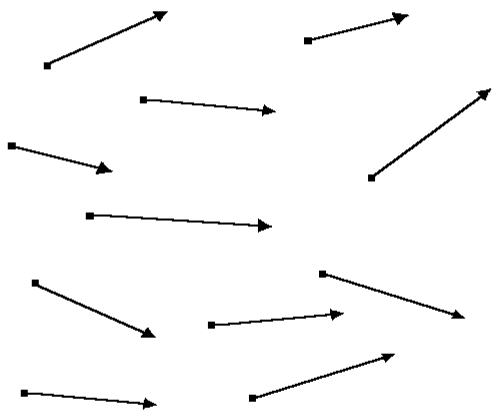


Figure A.11: A vector field.

then we say that f is a potential function for $\boldsymbol{F}$.

$\diamond$

We then can ask the question if the potential functions exist. To answer the question, we recast the problem into the one that involves solving n first order partial differential equations:

$$\frac{\partial f}{\partial x_i} = F_i(\boldsymbol{x}), \quad i = 1, 2, \ldots, n,$$

with the initial condition

$$\boldsymbol{F}(\boldsymbol{x}_0) = \boldsymbol{F}_0.$$

To answer the above question we need one more definition.

Definition A.11 *An open set $\mathcal{U}$ is said to be connected if, given two points in $\mathcal{U}$, there is a differentiable curve in $\mathcal{U}$ that joins the two points.*

$\diamond$

If $\mathcal{U} = \mathbb{R}^n$, then any two points can be joined by a straight line. Of course, it is not always the case that two points of an open set can be joined by a straight line, as it is illustrated in Figure A.12. We say that the function $\boldsymbol{F} = \begin{bmatrix} F_1 & F_2 & \cdots & F_n \end{bmatrix}^\top$ is a $\mathcal{C}^1$ function if all partial derivatives of all functions F_i exist and are continuous.

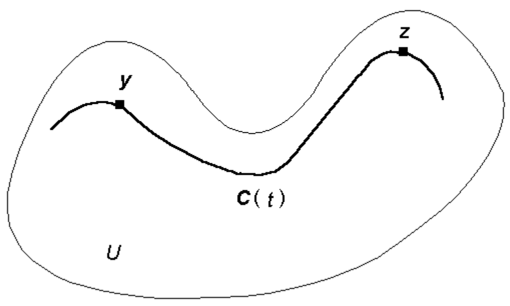


Figure A.12: A connected open set in which there are points that cannot be joined by a straight line.

Theorem A.21 *Let $\boldsymbol{F}(\boldsymbol{x})$ be $\mathcal{C}^1$ function defined on an open connected region $\mathcal{U}$ of $\mathbb{R}^n$. Then, there is a solution to the differential equation*

$$\nabla f(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x}), \quad \boldsymbol{F}(\boldsymbol{x}_0) = \boldsymbol{F}_0,$$

if and only if

$$\frac{\partial F_i}{\partial x_j} = \frac{\partial F_j}{\partial x_i},$$

that is, the Jacobian matrix of $\boldsymbol{F}$ is symmetric.

Proof We prove the theorem for $n = 2$, where the connected region $\mathcal{U}$ is a rectangle. We select a point $\boldsymbol{x}_0 = \begin{bmatrix} x_{10} & x_{20} \end{bmatrix}^\top$ in the rectangle. We assume that $\boldsymbol{F} = \begin{bmatrix} F_1 & F_2 \end{bmatrix}^\top$ and that

$$D_2 F_1 = D_1 F_2, \quad \text{that is,} \quad \frac{\partial F_1}{\partial x_2} = \frac{\partial F_2}{\partial x_1}.$$

Our goal is to find a potential function $f(x_1, x_2)$. We assume that the potential function has the form

$$f(x_1, x_2) = \int_{x_{10}}^{x_1} F_1(s, x_2) ds + u(x_2),$$

where $u(x_2)$ is a function of x_2 only and is determined later in the proof. We illustrate the integration path in Figure A.13. Applying the fundamental theorem of calculus, we obtain

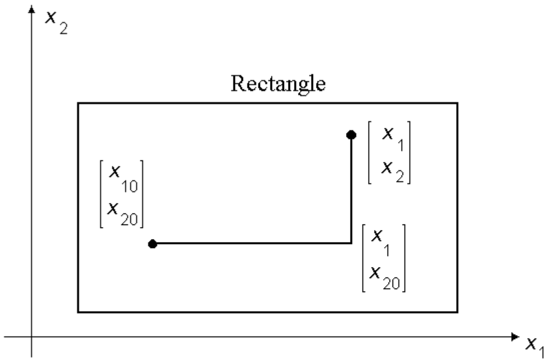


Figure A.13: Integration path used in the proof of Theorem A.21.

$$\begin{aligned}\frac{\partial f}{\partial x_1} &= \frac{\partial}{\partial x_1}\int_{x_{10}}^{x_1} F_1(s,x_2)ds + \frac{\partial u(x_2)}{\partial x_1}\\ &= F_1(x_1,x_2)\end{aligned}$$

because

$$\frac{\partial u(x_2)}{\partial x_1} = 0.$$

Hence,

$$D_1 f(x_1,x_2) = F_1(x_1,x_2)$$

as required. We now differentiate $f(x_1,x_2)$ with respect to x_2 to get

$$\begin{aligned}\frac{\partial f}{\partial x_2} &= \int_{x_{10}}^{x_1} \frac{\partial F_1(s,x_2)}{\partial x_2}ds + \frac{\partial u(x_2)}{\partial x_2}\\ &= \int_{x_{10}}^{x_1} \frac{\partial F_2(s,x_2)}{\partial x_1}ds + \frac{\partial u(x_2)}{\partial x_2}\end{aligned}$$

because, by assumption, $D_2F = D_1F$. Therefore,

$$\begin{aligned}\frac{\partial f}{\partial x_2} &= F_2(s,x_2)|_{x_{10}}^{x_1} + \frac{\partial u(x_2)}{\partial x_2}\\ &= F_2(x_1,x_2) - F_2(x_{10},x_2) + \frac{\partial u(x_2)}{\partial x_2}.\end{aligned}$$

Let

$$-F_2(x_{10},x_2) + \frac{\partial u(x_2)}{\partial x_2} = 0,$$

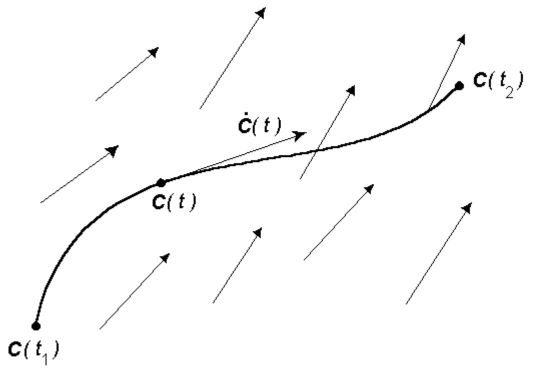


Figure A.14: A particle moving along a curve in vector field.

that is, $\frac{\partial u(x_2)}{\partial x_2} = F_2(x_{10}, x_2)$. Then, $D_2 f = F_2$ as wanted. To conclude the proof, we let

$$u(x_2) = \int_{x_{20}}^{x_2} F_2(x_{10}, x_2)dx_2,$$

which yields

$$\boxed{f(x_1, x_2) = \int_{x_{10}}^{x_1} F_1(s, x_2)ds + \int_{x_{20}}^{x_2} F_2(x_{10}, t)dt} \tag{A.176}$$

The proof is complete for $n = 2$.

□

We note that if the connected region $\mathcal{U}$ is an arbitrary open set, then the above proof may not work. This is because we needed to integrate over intervals that were contained in the region $\mathcal{U}$.

Suppose we are given a vector field $\boldsymbol{F}$ on an open set U in the plane. We can interpret $\boldsymbol{F}$ as a field of forces. We wish to calculate the work done when moving a particle from a point $\boldsymbol{C}(t_1)$ to a point $\boldsymbol{C}(t_2)$ along the curve $\boldsymbol{C}(t)$ in U, as illustrated in Figure A.14. Let

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} F_1(x_1, x_2) & F_2(x_1, x_2) \end{bmatrix}^\top.$$

We assume that each component of $\boldsymbol{F}$ is differentiable, that is, $\boldsymbol{F}$ is a differentiable vector field. We next assume that the curve $\boldsymbol{C}(t)$ is defined on a closed interval $[t_1, t_2]$ with $t_1 < t_2$. For each $t \in [t_1, t_2]$ the value $\boldsymbol{C}(t)$ is a point in $\mathbb{R}^2$. The curve $\boldsymbol{C}$ lies in U if $\boldsymbol{C}(t)$ is a point in U for all $t \in [t_1, t_2]$. We further assume that the curve $\boldsymbol{C}$ is continuously differentiable and its derivative $\dot{\boldsymbol{C}}(t) = \frac{d\boldsymbol{C}(t)}{dt}$ exists and is continuous, that is, we assume that $\boldsymbol{C} \in \mathcal{C}^1$.

We define the integral of $\boldsymbol{F}$ along $\boldsymbol{C}$ to be

$$\begin{aligned}\int_{\boldsymbol{C}} \boldsymbol{F} &= \int_{t_1}^{t_2} \boldsymbol{F}(\boldsymbol{C}(t))^\top \frac{d\boldsymbol{C}}{dt} dt \\ &= \int_{t_1}^{t_2} \boldsymbol{F}(\boldsymbol{C}(t))^\top d\boldsymbol{C},\end{aligned}$$

which is the work done against the force field $\boldsymbol{F}$ along the curve $\boldsymbol{C}$. Let

$$\boldsymbol{C}(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^\top.$$

Then, we can write

$$\int_{\boldsymbol{C}} \boldsymbol{F} = \int_{\boldsymbol{C}} F_1 dx_1 + F_2 dx_2. \tag{A.177}$$

Theorem A.22 (Green) *Let the vector field $\boldsymbol{F}$ be given on a region A that is the interior of a closed path $\boldsymbol{C}$ parameterized counterclockwise. Then,*

$$\int_{\boldsymbol{C}} F_1 dx_1 + F_2\, dx_2 = \iint_A \left(\frac{\partial F_2}{\partial x_1} - \frac{\partial F_1}{\partial x_2} \right) dx_1 dx_2.$$

A.18 Matrix Calculus Formulas

Below, we list useful formulas from matrix calculus that involve computing the gradient (with respect to $\boldsymbol{x}$). Recall that in our convention the gradient is a column vector. We have:

$$\nabla \left(\boldsymbol{x}^\top \boldsymbol{y}\right) = \nabla \left(\boldsymbol{y}^\top \boldsymbol{x}\right) = \boldsymbol{y}, \tag{A.178}$$

$$\nabla \left(\boldsymbol{x}^\top \boldsymbol{x}\right) = 2\boldsymbol{x}, \tag{A.179}$$

$$\nabla \left(\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}\right) = \boldsymbol{A}\boldsymbol{y}, \tag{A.180}$$

$$\nabla \left(\boldsymbol{y}^\top \boldsymbol{A} \boldsymbol{x}\right) = \boldsymbol{A}^\top \boldsymbol{y}, \tag{A.181}$$

$$\nabla \left(\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}\right) = \left(\boldsymbol{A} + \boldsymbol{A}^\top\right) \boldsymbol{x}. \tag{A.182}$$

Let $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$ be two maps, and let $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ and $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$ be the Jacobian matrices of $\boldsymbol{f}$ and $\boldsymbol{g}$, respectively. Then,

$$\nabla \left(\boldsymbol{f}(\boldsymbol{x})^\top \boldsymbol{g}(\boldsymbol{x})\right) = \nabla \left(\boldsymbol{g}(\boldsymbol{x})^\top \boldsymbol{f}(\boldsymbol{x})\right) = \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)^\top \boldsymbol{g} + \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^\top \boldsymbol{f}. \tag{A.183}$$

For an $n \times n$ symmetric matrix $\boldsymbol{Q}$,

$$\nabla\left(\boldsymbol{f}(\boldsymbol{x})^\top \boldsymbol{Q}\boldsymbol{f}(\boldsymbol{x})\right) = 2\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)^\top \boldsymbol{Q}\boldsymbol{f}. \tag{A.184}$$

A.19 Notes

For other proof techniques, we refer to Velleman [335]. For further reading on linear algebra, and in particular matrix theory, we recommend classical texts of Gelfand [112] and Gantmacher [110]. Analyses of numerical methods for matrix computations can be found in Faddeeva [98], Householder [144], Golub and Van Loan [117], as well as in Gill, Murray, and Wright [114]. Lang [203] and Seeley [297] are recommended for a review of calculus of several variables. A straightforward approach to mathematical analysis is presented in Binmore [35]. Principles of real analysis can be found in Natanson [253], Bartle [25], and Rudin [285]. Some facts from basic functional analysis that we touched upon in this chapter, are discussed, for example, by Maddox [227], and Debnath and Mikusiński [74]. Lucid expositions of differential equations are in Driver [88], Miller and Michel [242], and in Boyce and DiPrima [40].

The famous mathematician, G. H. Hardy (1877–1947) argues in [130, page 113] that in truly great theorems and their proofs, "there is a very high degree of *unexpectedness*, combined with *inevitability* and *economy*. The arguments take so odd and surprising a form; the weapons used seem so childishly simple when compared with the far-reaching results; but there is no escape from the conclusions." He further states that "A mathematical proof should resemble a simple and clear-cut constellation, not a scattered cluster in the Milky Way."

A.20 Exercises

Exercise A.1 The open unit disc with respect to the norm $\|\cdot\|_p$, is the set

$$\Delta_p = \{\boldsymbol{x} \in \mathbb{R}^2 : \|\boldsymbol{x}\|_p < 1\}.$$

The unit discs are quite different for different choices of the norm $\|\cdot\|_p$. Sketch Δ_1, Δ_2, Δ_3, and Δ_∞.

Exercise A.2 Let $v_1, v_2, \ldots, v_n$ be non-zero, mutually orthogonal elements of the space V and let v be an arbitrary element of V.

(i) Find the component of v along v_j.

(ii) Let c_j denote the component of v along v_j. Show that the element

$$v - c_1 v_1 - \cdots c_n v_n$$

is orthogonal to $v_1, v_2, \ldots, v_n$.

(iii) Verify the Bessel inequality,

$$\sum_{i=1}^{n} c_i^2 \leq \|v\|^2.$$

Exercise A.3 (M.C.M. Teixeira) Given a linear combination of matrices,

$$\boldsymbol{M} = \alpha_1 \boldsymbol{M}_1 + \cdots + \alpha_r \boldsymbol{M}_r, \tag{A.185}$$

where for $i = 1, \ldots, r$, $\alpha_i \in \mathbb{R}$, and $\boldsymbol{M}_i \in \mathbb{R}^{n \times n}$. Let $\boldsymbol{m}_i^j$ be the j-th row of the matrix $\boldsymbol{M}_i$. Thus,

$$\boldsymbol{M}_i = \begin{bmatrix} \boldsymbol{m}_i^1 \\ \boldsymbol{m}_i^2 \\ \vdots \\ \boldsymbol{m}_i^n \end{bmatrix}.$$

Show that

$$\det \boldsymbol{M} = \sum_{s_1=1}^{r} \cdots \sum_{s_n=1}^{r} \alpha_{s_1} \cdots \alpha_{s_n} \det \begin{bmatrix} \boldsymbol{m}_{s_1}^1 \\ \vdots \\ \boldsymbol{m}_{s_n}^n \end{bmatrix}. \tag{A.186}$$

Exercise A.4 Let $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$. Show that

$$\det(\boldsymbol{I}_n + \boldsymbol{X}\boldsymbol{Y}) = \det(\boldsymbol{I}_m + \boldsymbol{Y}\boldsymbol{X}).$$

Exercise A.5 Let $\lambda_i(\boldsymbol{A})$ denote the i-th eigenvalue of a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and let α be a scalar. Show that

(i) $\lambda_i(\alpha \boldsymbol{A}) = \alpha \lambda_i(\boldsymbol{A})$;

(ii) $\lambda_i(\boldsymbol{I}_n \pm \alpha \boldsymbol{A}) = 1 \pm \alpha \lambda_i(\boldsymbol{A})$.

Exercise A.6 Show that the formulas (A.75) and (A.76) for matrix norms are equivalent.

Exercise A.7 Show that for any square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$,

$$\|\boldsymbol{A}\| \geq \max_{1 \leq i \leq n} |\lambda_i(\boldsymbol{A})| . \tag{A.187}$$

Exercise A.8 Show that if $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ is symmetric, then

$$\lambda_{\min}(\boldsymbol{P})\|\boldsymbol{x}\|_2^2 \leq \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} \leq \lambda_{\max}(\boldsymbol{P})\|\boldsymbol{x}\|_2^2.$$

Exercise A.9 Show that a sufficient condition for

$$\lim_{k\to\infty} \boldsymbol{A}^k \to \boldsymbol{O}.$$

is

$$\|\boldsymbol{A}\| < 1.$$

Exercise A.10 Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. Show that a necessary and sufficient condition for

$$\lim_{k\to\infty} \boldsymbol{A}^k \to \boldsymbol{O}.$$

is

$$|\lambda_i(\boldsymbol{A})| < 1, \quad i = 1, 2, \ldots, n$$

Exercise A.11 Let $\operatorname{eig}(\boldsymbol{M})$ denote the set of eigenvalues of the square matrix $\boldsymbol{M}$ and let Δ be the open unit disk in the complex plane. Prove that if $\operatorname{eig}(\boldsymbol{M}) \subset \Delta \cup \{1\}$, then $\lim_{k\to\infty} \boldsymbol{M}^k$ exists if and only if the dimension of the eigenspace corresponding to the eigenvalue 1 is equal to the multiplicity of the eigenvalue 1, that is, the geometric multiplicity of the eigenvalue 1 is equal to its algebraic multiplicity.

Exercise A.12 (a) Show that the series

$$\sum_{n=1}^{\infty} \frac{1}{n}$$

diverges to $+\infty$.

(a) Show that if α is a rational number such that $\alpha > 1$, then the series

$$\sum_{n=1}^{\infty} \frac{1}{n^\alpha}$$

converges.

(c) Use MATLAB's Symbolic Math Toolbox to compute

$$\sum_{n=1}^{\infty} \frac{1}{n^2}.$$

Exercise A.13 (a) Let $0 < x < 1$. Show that

$$\ln(1-x) \leq -x.$$

(b) Let $0 < x \leq 1/2$. Show that

$$\ln(1-x) \geq -2x.$$

Exercise A.14 Let $0 < a_k < 1$ for $k \geq 0$. Use a proof by contraposition, Theorem A.16, and Exercise A.13 to show that

$$\prod_{k=0}^{\infty}(1-a_k) = 0, \quad \text{if and only if} \quad \sum_{k=0}^{\infty} a_k = \infty.$$

Exercise A.15 Prove Sylvester's inequalities (A.88).

Exercise A.16 Compute $e^{\boldsymbol{A}t}$ for

$$\text{(a) } \boldsymbol{A} = \begin{bmatrix} 0 & -3 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \text{(b) } \boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix}.$$

Exercise A.17 Find $\boldsymbol{x}(t)$ for the dynamical system model,

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad \boldsymbol{x}(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where $u(t)$ is the unit step function, that is, $u(t) = 1$ for $t \geq 0$ and $u(t) = 0$ for $t < 0$.

Exercise A.18 For

$$\boldsymbol{A} = \begin{bmatrix} -2 & -6 \\ 2 & -10 \end{bmatrix},$$

find constants $M > 0$ and $\alpha > 0$ such that

$$\left\| e^{\boldsymbol{A}t} \right\|_2 \leq M e^{-\alpha t}, \quad t \geq 0.$$

Exercise A.19 Show that the function space $\mathcal{C}([a,b])$ of real-valued functions, $f : \mathbb{R} \to \mathbb{R}$, with the norm defined by (A.99) is a normed linear space. Show that this space is a Banach space.

Exercise A.20 Derive a sufficient condition for the convergence of the iteration process

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{A}\boldsymbol{x}^{(k)} + \boldsymbol{b}, \quad k = 0, 1, \ldots,$$

for any initial $\boldsymbol{x}(0)$, where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$.

Bibliography

[1] P. P. Acarnley. *Stepping Motors: A Guide to Modern Theory and Practice.* IEE Control Engineering Series 19. Peter Peregrinus Ltd., London, 1982.

[2] J. Ackermann. Robust control prevents car skiding. *IEEE Control Systems Magazine*, 17(3):23–31, June 1997.

[3] M. Adachi and K. Aihara. Associative dynamics in a chaotic neural network. *Neural Networks*, 10(1):83–98, January 1997.

[4] J. A. Adam and N. Bellomo, editors. *A Survey of Models for Tumor-Immune System Dynamics.* Birkhäuser, Boston, 1997.

[5] K. Aihara, T. Takabe, and M. Toyoda. Chaotic neural networks. *Physics Letters A*, 144(6, 7):333–340, 12 March 1990.

[6] A. W. Al-Khafaji and J. R. Tooley. *Numerical Methods in Engineering Practice.* Holt, Rinehart and Winston, Inc., New York, 1986.

[7] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *CHAOS: An Introduction to Dynamical Systems.* Springer-Verlag, New York, 1997.

[8] H. Alwi and C. Edwards. Fault detection and fault-tolerant control of a civil aircraft using a sliding-mode-based scheme. *IEEE Transactions on Control Systems Technology*, 16(3):499–510, May 2008.

[9] H. Alwi, C. Edwards, and C. P. Tan. Sliding mode estimation schemes for incipient sensor faults. *Automatica*, 45(7):1679–1685, July 2009.

[10] J. A. Anderson. *An Introduction to Neural Networks.* A Bradford Book, The MIT Press, Cambridge, Massachusetts, 1995.

[11] Y. V. Andreyev, Y. L. Belsky, A. S. Dmitriev, and D. A. Kuminov. Information processing using dynamical chaos: Neural networks implementation. *IEEE Transactions on Neural Networks*, 7(2):290–299, March 1996.

[12] P. J. Antsaklis and A. N. Michel. *Linear Systems*. McGraw-Hill, New York, 1997.

[13] P. J. Antsaklis and A. N. Michel. *A Linear Systems Primer*. Birkhäuser, Boston, 2007.

[14] D. K. Arrowsmith and C. M. Place. *An Introduction to Dynamical Systems*. Cambridge University Press, Cambridge, Great Britain, 1990.

[15] K. J. Åström and T. Hägglund. PID Control. In W. S. Levine, editor, *The Control Handbook*, chapter 10.5, pages 198–209. CRC Press in cooperation with IEEE Press, Boca Raton, Florida, 1996.

[16] A. Azemi and E. E. Yaz. Sliding-mode adaptive observer appraoch to chaotic synchronization. *Journal of Dynamic Systems, Measurement, and Control*, 122:758–765, December 2000.

[17] T. Bäck, U. Hammel, and H.-P. Schwefel. Evolutionary computation: comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, April 1997.

[18] F. N. Bailey. The application of Lyapunov's second method to interconnected systems. *Journal of SIAM on Control*, 3(3):443–462, 1966.

[19] G. J. Bakker and H. Nijmeijer. Comments on 'Control design for multivariable nonlinear time-varying systems'. *Int. J. Control*, 41(6):1627–1629, 1985.

[20] P. Ball, editor. *A Biography of Water: Life's Matrix*. Farra, Straus and Giroux, New York, 1999.

[21] S. P. Banks. *Control Systems Engineering: Modelling and Simulation, Control Theory and Microprocessor Implementation*. Prentice-Hall International, Englewood Cliffs, NJ 07632, 1986.

[22] B. R. Barmish and A. R. Galimidi. Robustness of Luenberger observers: Linear systems stabilized via non-linear control. *Automatica*, 22(4):413–423, July 1986.

[23] S. Barnett. *Introduction to Mathematical Control Theory*. Clarendon Press, Oxford, 1975.

[24] M. Barnsley. *Fractals Everywhere*. Academic Press, San Diego, 1988.

[25] R. G. Bartle. *The Elements of Real Analysis*. John Wiley & Sons, New York, second edition, 1976.

[26] E. F. Beckenbach and R. Bellman. *Inequalities*. Springer-Verlag, New York Inc., Berlin, 1965. Second revised printing.

[27] R. Bellman. *Introduction to Matrix Analysis*. McGraw-Hill Book Company, Inc., New York, 1960.

[28] R. Bellman. Vector Lyapunov functions. *Journal of SIAM on Control*, 1(1):32–34, 1962.

[29] A. Ben-Zvi, S. D. Vernon, and G. Broderick. Model-based therapeutic correction of hypothalamic-pituitary-adrenal axis dysfunction. *PLoS Computational Biology*, 5(1):1–10, January 23, 2009.

[30] S. Bennett. Development of the PID controller. *IEEE Conrol Systems Magazine*, 13(6):58–65, December 1993.

[31] D. Bestle and M. Zeitz. Canonical form observer design for non-linear time-variable systems. *Int. J. Control*, 38(2):419–431, 1983.

[32] S. P. Bhattacharyya. Observer design for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, AC-23(3):483–484, June 1978.

[33] A. Bhaya and E. Kaszkurewicz. *Control Perspectives on Numerical Algorithms and Matrix Problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2006.

[34] L. Bingzheng, Z. Zhenye, and C. Liansong. A mathematical model of the regulation system of the secretion of glucocorticoids. *J. Biological Physics*, 17(4):221–233, December 1990.

[35] K. G. Binmore. *Mathematical Analysis: A Straightforward Approach*. Cambridge University Press, Cambridge, Great Britain, second edition, 1982.

[36] C. Bissell. A history of automatic control. In S. Y. Nof, editor, *Springer Handbook of Automation*, chapter 4, pages 53–69. Springer-Verlag, Berlin Heidelberg, 2009.

[37] D. L. Boley. On Kalman's procedure for the computation of the controllable/observable canonical form. *SIAM J. Control and Optimization*, 18(6):624–626, November 1980.

[38] V. G. Boltyanskii. *Mathematical Methods of Optimal Control*. Holt, Rinehart and Winston, Inc., New York, 1971.

[39] H. Bourlès and B. Marinescu. *Linear Time-Varying Systems: Algebraic-Analytic Approach*. Springer-Verlag, Berlin Heidelberg, 2011. Lecture Notes in Control and Information Sciences 410.

[40] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. John Wiley & Sons, Inc., New York, sixth edition, 1997.

[41] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[42] J. W. Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems*, CAS-25(9):772–781, September 1978.

[43] R. W. Brockett. *Finite Dimensional Linear Systems*. J. Wiley & Sons, Inc., New York, 1970.

[44] W. L. Brogan. *Modern Control Theory*. Prentice Hall, Englewood Cliffs, New Jersey 07632, third edition, 1991.

[45] A. E. Bryson and Y. C. Ho. *Applied Optimal Control*. Blaisdell Publishing Co., Waltham, Mass., 1975.

[46] A. E. Bryson, Jr. *Dynamic Optimization*. Addison Wesley Longman, Inc., Menlo Park, CA, 1999.

[47] A. Bunde and S. Havlin, editors. *Fractals in Science*. Springer-Verlag, Berlin, 1994.

[48] J. B. Burl. *Linear Optimal Control; $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Methods*. Addison Wesley Longman, Inc., Menlo Park, CA, 1999.

[49] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, London, second edition, 2004.

[50] R. H. Cannon, Jr. *Dynamics of Physical Systems*. McGraw-Hill, New York, 1967.

[51] S. G. Cao, N. W. Rees, and G. Feng. Quadratic stability analysis and design of continuous-time fuzzy control systems. *Int. J. System Science*, 27(2):193–203, 1996.

[52] T. L. Carroll and L. M. Pecora. Synchronizing chaotic circuits. *IEEE Transactions on Circuits and Systems*, 38(4):453–456, April 1991.

[53] H. Y. Chan and S. H. Żak. Chaotic neural fuzzy associative memory. *Int. J. Bifurcation and Chaos*, 9(8):1597–1617, August 1999.

[54] K. C. C. Chan, V. Lee, and H. Leung. Generating fuzzy rules for target tracking using a steady-state genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 1(3):189–200, September 1997.

[55] C.-T. Chen. *Linear System Theory and Design.* Oxford University Press, New York, third edition, 1999.

[56] J. Chen and R. J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems.* Kluwer Academic Publishers, Boston, 1999.

[57] J. Chen, R. J. Patton, and H.-Y. Zhang. Design of unknown input observers and robust fault detection filters. *Int. J. Control*, 63(1):85–105, 1996.

[58] M. Chen and W. Min. Unknown input observer based chaotic secure communication. *Physics Letters A*, 372(10):1595–1600, 3 March 2008.

[59] X. Chen and H. Kano. State observer for a class of nonlinear systems and its application to machine vision. *IEEE Transactions on Automatic Control*, 49(11):2085–2091, November 2004.

[60] J. Chiasson and B. Vairamohan. Estimating the state of charge of a battery. In *Proceedings of the 2003 American Control Conference, Denver, Colorado*, pages 2863–2868, June 4–6, 2003.

[61] J. Chiasson and B. Vairamohan. Estimating the state of charge of a battery. *IEEE Transactions on Control Systems Technology*, 13(3):465–470, May 2005.

[62] E. K. P. Chong and S. H. Żak. *An Introduction to Optimization.* Wiley-Interscience, John Wiley & Sons, Inc., New York, third edition, 2008.

[63] A. Cichocki and R. Unbehauen. *Neural Networks for Optimization and Signal Processing.* John Wiley, Chichester, England, 1993.

[64] M. Conrad, C. Hubold, B. Fischer, and A. Peters. Modeling the hypothalamus-pituitary-adrenal system: homeostasis by interacting positive and negative feedback. *J. Biological Physics*, 35(2):149–162, May 2009.

[65] S. D. Conte and C. de Boor. *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill Book Company, New York, third edition, 1980.

[66] I. M. Copi. *Introduction to Logic*. Macmillan Publishing Co., Inc., New York, fifth edition, 1978.

[67] C. Corduneanu. *Principles of Differential and Integral Equations*. Allyn and Bacon, Inc., Boston, Massachusetts, 1971.

[68] M. Corless and J. Tu. State and input estimation for a class of uncertain systems. *Automatica*, 34(6):757–764, June 1998.

[69] M. J. Corless and G. Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, AC-26(5):1139–1144, October 1981.

[70] K. M. Cuomo, A. V. Oppenheim, and S. H. Strogatz. Synchronization of Lorenz-based chaotic circuits with applications to communications. *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, 40(10):626–633, October 1993.

[71] M. Darouach, M. Zasadzinski, and S. J. Xu. Full-order observers for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, 39(3):606–609, March 1994.

[72] D. M. Dawson, J. Hu, and T. C. Burg. *Nonlinear Control of Electric Machinery*. Marcel Dekker, New York, 1998.

[73] J. J. D'Azzo and C. H. Houpis. *Linear Control System Analysis and Design: Conventional and Modern*. The McGraw-Hill, Inc., New York, fourth edition, 1995.

[74] L. Debnath and P. Mikusiński. *Introduction to Hilbert Spaces with Applications*. Academic Press, Inc., San Diego, 1990.

[75] R. A. DeCarlo. *Linear Systems: A State Variable Approach With Numerical Implementation*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1989.

[76] R. A. DeCarlo, S. H. Żak, and S. V. Drakunov. Variable structure, sliding-mode controller design. In W. S. Levine, editor, *The Control Handbook: Control System Advanced Methods*, chapter 50, pages 50–1 – 50–22. CRC Press, Boca Raton, Florida, 2011.

[77] R. A. DeCarlo, S. H. Żak, and G. P. Matthews. Variable structure control of nonlinear multivariable systems: A tutorial. *Proceedings of the IEEE*, 76(3):212–232, March 1988.

[78] V. Del Toro. *Electric Machines and Power Systems.* Prentice Hall, Inc., Englewood Cliffs, N.J. 07632, 1985.

[79] M. J. Denham. Canonical forms for the identification of multivariable linear systems. *IEEE Transactions on Automatic Control*, AC-19(6):646–656, December 1974.

[80] A. Derdiyok. Speed-sensorless control of induction motor using a continuous control approach of sliding-mode and flux observer. *IEEE Transactions on Industrial Electronics*, 52(4):1170–1176, August 2005.

[81] A. Derdiyok, M. K. Güven, H. Rehman, N. Inanc, and L. Xu. Design and implementation of a new sliding-mode observer for speed-sensorless control of induction machine. *IEEE Transactions on Industrial Electronics*, 49(5):1177–1182, October 2002.

[82] R. L. Devaney and L. Keen, editors. *Chaos and Fractals; The Mathematics Behind the Computer Graphics*, Providence, Rhode Island, 1989. American Mathematical Society. Proceedings of Symposia in Applied Mathematics, Volume 39.

[83] B. M. Diong and J. V. Medanic. Dynamic output feedback variable structure control for system stabilization. *Int. J. Control*, 56(3):607–630, September 1992.

[84] A. S. Dmitriev, A. I. Panas, and S. O. Starkov. Storing and recognizing information based on stable cycles of one-dimensional maps. *Physics Letters A*, 155(8, 9):494–499, 27 May 1991.

[85] R. C. Dorf and R. H. Bishop. *Modern Control Systems.* Prentice Hall, Upper Saddle River, New Jersey, twelfth edition, 2011.

[86] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control.* Springer-Verlag, Berlin, 1993.

[87] M. Driels. *Linear Control Systems Engineering.* McGraw-Hill, Inc., New York, 1996.

[88] R. D. Driver. *Ordinary and Delay Differential Equations.* Springer-Verlag, New York, 1977.

[89] W. Dunham. *Journey Through Genius: The Great Theorems of Mathematics.* Wiley Science Editions, J. Wiley & Sons, Inc., New York, 1990.

[90] C. Edwards and S. K. Spurgeon. On the development of discontinuous observers. *Int. J. Control*, 59(5):1211–1229, May 1994.

[91] C. Edwards and S. K. Spurgeon. Compensator based output feedback sliding mode controller design. *Int. J. Control*, 71(4):601–614, 10 November 1998.

[92] C. Edwards and S. K. Spurgeon. *Sliding Mode Control: Theory and Applications.* Systems and Control Book Series. Taylor & Francis Ltd, London, England, 1998.

[93] C. Edwards, S. K. Spurgeon, and R. J. Patton. Sliding mode observers for fault detection and isolation. *Automatica*, 36(4):541–553, April 2000.

[94] G. Ellis. *Observers in Control Systems; A Practical Guide.* Academic Press, San Diego, 2002.

[95] G. D. Elseth and K. D. Baumgardner. *Genetics.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

[96] L. E. Elsgolc. *Calculus of Variations.* Pergamon Press Ltd., London, 1961.

[97] K. D. Elworthy, W. N. Everitt, and E. B. Lee, editors. *Differential Equations, Dynamical Systems, and Control Science: A Festschrift in Honor of Lawrence Markus.* Marcel Dekker, Inc., New York, 1994. Lecture Notes in Pure and Applied Mathematics, Vol. 152.

[98] V. N. Faddeeva. *Computationl Methods of Linear Algebra.* Dover Publications, Inc., New York, 1959.

[99] L. Fausett. *Fundamentals of Neural Networks; Architectures, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, NJ 07632, 1994.

[100] M. J. Feigenbaum. Universal behavior in nonlinear systems. *Physica D*, 7D(1–3):16–39, May 1983.

[101] G. Feng, S. G. Cao, N. W. Rees, and C. K. Chak. Design of fuzzy control systems with guaranteed stability. *Fuzzy Sets and Systems*, 85(1):1–10, January 1, 1997.

[102] A. F. Filippov. Differential equations with discontinuous right-hand side. In *American Mathematical Society Translations*, pages 199–231. American Mathematical Society, Providence, Rhode Island, 1964.

[103] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Mathematics and Its Applications (Soviet Series). Kluwer Academic Publishers, Dordrecht, The Netherlands, 1988.

[104] K. Fischle and D. Schröder. An improved stable adaptive fuzzy control method. *IEEE Transactions on Fuzzy Systems*, 7(1):27–40, February 1999.

[105] D. B. Fogel. What is evolutionary computation? *IEEE Spectrum*, 37(2):26–32, February 2000.

[106] G. F. Franklin and J. D. Powell. *Digital Control of Dynamic Systems*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.

[107] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458, fourth edition, 2002.

[108] B. Friedland. *Advanced Control System Design*. Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1996.

[109] A. T. Fuller. Maxwell's treatment of governors fitted with differential gears. *Int. J. Control*, 65(3):385–408, October 1996.

[110] F. R. Gantmacher. *The Theory of Matrices*, volume 1. Chelsea Publishing Company, New York, second edition, 1990.

[111] S. S. Ge, T. H. Lee, and G. Zhu. Genetic algorithm tuning of Lyapunov-based controllers: An application to a single-link flexible robot system. *IEEE Transactions on Industrial Electronics*, 43(5):567–574, October 1996.

[112] I. M. Gel'fand. *Lectures on Linear Algebra*. Interscience Publishers, Inc., New York, 1961.

[113] M. Gen and R. Cheng. *Genetic Algorithms and Engineering Design.* A Wiley-Interscience Publication, John Wiley & Sons, Inc., New York, 1997.

[114] P. E. Gill, W. Murray, and M. H. Wright. *Numerical Linear Algebra and Optimization*, volume 1. Addison-Wesley Publishing Company, Redwood City, CA, 1991.

[115] J. Gleick. *Chaos: Making a New Science.* Penguin Books, New York, 1988.

[116] I. Glynn, editor. *An Anatomy of Thought: The Origin and Machinery of Mind.* Oxford University Press, New York, 1999.

[117] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, Maryland, 1983.

[118] D. Graham and D. McRuer. *Analysis of Nonlinear Control Systems.* J. Wiley & Sons, Inc., New York, 1961.

[119] W. J. Grantham and A. M. Athalye. Discretization chaos: Feedback control and transition to chaos. In C. T. Leondes, editor, *Control and Dynamic Systems; Advances in Theory and Applications, Vol. 34: Advances in Control Mechanics, Part 1 of 2*, pages 205–277. Academic Press, San Diego, California 92101, 1990.

[120] G. Grassi and S. Mascolo. Nonlinear observer design to synchronize hyperchaotic systems via a scalar signal. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 44(10):1011–1014, October 1997.

[121] D. Greenfield. When is PID not the answer? *Control Engineering*, 57(3):54–57, March 2010.

[122] S. Gupta, E. Aslakson, B. M. Gurbaxani, and S. D. Vernon. Inclusion of the glucocorticoid receptor in a hypothalamic pituitary adrenal axis model reveals bistability. *Theoretical Biology and Medical Modeling*, 4(8):1–12, 14 February 2007.

[123] Q. P. Ha, H. Trinh, H. T. Nguyen, and H. D. Tuan. Dynamic output feedback sliding-mode control using pole placement and linear functional observers. *IEEE Transactions on Industrial Electronics*, 50(5):1030–1037, October 2003.

[124] V. B. Haas. EE 675 Class Notes. Purdue University, School of Electrical Engineering, West Lafayette, IN., Fall 1980.

[125] M. T. Hagan, H. B. Demuth, and M. Beale. *Neural Network Design*. PWS Publishing Co., Boston, MA, 1996.

[126] T. Hägglund and K. J. Åström. Automatic tuning of PID controllers. In W. S. Levine, editor, *The Control Handbook: Control System Advanced Methods*, chapter 32, pages 32–1 – 32–20. CRC Press, Boca Raton, Florida, 2011.

[127] W. Hahn. Über die Anwendung der Methode von Ljapunov auf Differenzengleichungen. *Mathematische Annalen*, 136(1):430–441, 1958.

[128] W. Hahn. *Theory and Application of Liapunov's Direct Method*. Prentice-Hall International Series in Applied Mathematics. Prentice-Hall, Englewood Cliffs, N.J., 1963.

[129] W. Hahn. *Stability of Motion*. Springer Verlag, New York, 1967.

[130] G. H. Hardy. *A Mathematician's Apology*. Cambridge University Press, Cambridge, Great Britain, 1992. Canto edition with a foreword by C. P. Snow.

[131] R. L. Harvey. *Neural Network Principles*. Prentice Hall, Englewood Cliffs, NJ 07632, 1994.

[132] M. H. Hassoun. *Fundamentals of Artificial Neural Networks*. A Bradford Book, The MIT Press, Cambridge, MA, 1995.

[133] S. W. Hawking. *A Brief History of Time: From the Big Bang to Black Holes*. Bantam Books, New York, 1988.

[134] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ 07458, second edition, 1999.

[135] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Lecture Notes Volume I in the Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley Publishing, Redwood City, CA, 1991.

[136] M. R. Hestenes. *Calculus of Variations and Optimal Control Theory*. John Wiley & Sons, Inc., New York, 1966.

[137] N. J. Higham. *Handbook of Writing for the Mathematical Sciences*. SIAM, Philadelphia, 1993.

[138] H. Hochstadt. *Differential Equations: A Modern Approach.* Dover Publications, Inc., New York, 1975.

[139] J. H. Holland. *Adaptation in Natural and Artificial Systems; An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* A Bradford Book, The MIT Press, Cambridge, Massachusetts, 1992.

[140] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA, Biophysics*, 81:3088–3092, May 1984.

[141] G. Hostetter and J. S. Meditch. Observing systems with unmeasurable inputs. *IEEE Transactions on Automatic Control*, AC-18(3):307–308, June 1973.

[142] M. Hou and P. C. Müller. Design of observers for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, 37(6):871–875, June 1992.

[143] M. Hou, A. C. Pugh, and P. C. Müller. Disturbance decoupled functional observers. *IEEE Transactions on Automatic Control*, 44(2):382–386, February 1999.

[144] A. S. Householder. *The Theory of Matrices in Numerical Analysis.* Dover Publications, Inc., New York, 1975.

[145] J. C. Hsu and A. U. Meyer. *Modern Control Principles and Applications.* McGraw-Hill Book Company, New York, 1968.

[146] S. Hui and S. H. Żak. Control and observation of uncertain systems: A variable structure systems approach. In C. T. Leondes, editor, *Control and Dynamic Systems; Advances in Theory and Applications, Vol. 34: Advances in Control Mechanics, Part 1 of 2*, pages 175–204. Academic Press, San Diego, California 92101, 1990.

[147] S. Hui and S. H. Żak. Robust control synthesis for uncertain/nonlinear dynamical systems. *Automatica*, 28(2):289–298, 1992.

[148] S. Hui and S. H. Żak. Low-order state estimators and compensators for dynamical systems with unknown inputs. *Systems & Control Letters*, 21(6):493–502, December 1993.

[149] S. Hui and S. H. Żak. Robust output feedback stabilization of uncertain dynamic systems with bounded controllers. *Int. J. of Robust and Nonlinear Control*, 3(2):115–132, August 1993.

[150] S. Hui and S. H. Żak. The Widrow-Hoff algorithm for McCulloch-Pitts type neurons. *IEEE Transactions on Neural Networks*, 5(6):924–929, November 1994.

[151] S. Hui and S. H. Żak. Observer design for systems with unknown inputs. *Int. J. Appl. Math. Comput. Sci*, 15(4):431–446, 2005.

[152] J. Y. Hung, W. Gao, and J. C. Hung. Variable structure control: A survey. *IEEE Transactions on Industrial Electronics*, 40(1):2–22, February 1993.

[153] L. R. Hunt, R. Su, and G. Meyer. Global transformations of nonlinear systems. *IEEE Transactions on Automatic Control*, AC-28(1):24–31, January 1983.

[154] D. R. Hush and B. G. Horne. Progress in supervised neural networks: What's new since Lippmann? *IEEE Signal Processing Magazine*, 10(1):8–39, January 1993.

[155] H. S. Hwang. Automatic design of fuzzy rule base for modelling and control using evolutionary programming. *IEE Proceedings—Control Theory Appl.*, 146(1):9–16, January 1999.

[156] P. Ignaciuk and A. Bartoszewicz. Discrete-time sliding-mode congestion control in multisource communication networks with time-varying delay. *IEEE Transactions on Control Systems Technology*, 19(4):852–867, July 2011.

[157] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, London, third edition, 1995.

[158] A. Isidori and M. D. Di Benedetto. Feedback linearization of nonlinear systems. In W. S. Levine, editor, *The Control Handbook: Control System Advanced Methods*, chapter 46, pages 46–1 – 46–15. CRC Press, Boca Raton, Florida, 2011.

[159] U. Itkis. *Control Systems of Variable Structure*. A Halsted Press Book, a division of John Wiley & Sons, New York, 1976.

[160] M. Jamshidi, M. Tarokh, and B. Shafai. *Computer-Aided Analysis and Design of Linear Control Systems*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1992.

[161] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Engineering: A Computational Approach to Learning and Machine Intelligence*. MATLAB Curriculum Series. Prentice-Hall, Upper Saddle River, NJ 07458, 1997.

[162] B. Jiang, M. Staroswiecki, and V. Cocquempot. Fault estimation in nonlinear uncertain systems using robust/sliding-mode observers. *IEE Proceedings—Control Theory Appl.*, 151(1):29–37, February 2004.

[163] G.-P. Jiang, W. K.-S. Tang, and G. Chen. A state-observer-based approach for synchronization in complex dynamical networks. *IEEE Transactions on Circuits and Systems—I*, 53(12):2739–2745, December 2006.

[164] T. Kaczorek. *Teoria Sterowania i Systemów.* Wydawnictwo Naukowe PWN—Polish Scientific Publishers, Warszawa, 1993.

[165] T. Kaczorek. *Wektory i Macierze w Automatyce i Elektrotechnice.* Wydawnictwa Naukowe-Techniczne—Scientific-Technical Publishers, Warszawa, second edition, 1998.

[166] T. Kaczorek, A. Dzieliński, W. Dąbrowski, and R. Łopatka. *Podstawy Teorii Sterowania.* Wydawnictwa Naukowe-Techniczne—Scientific-Technical Publishers, Warszawa, 2005.

[167] P. B. Kahn. *Mathematical Methods for Scientists and Engineers: Linear and Nonlinear Systems.* Wiley-Interscience, J. Wiley & Sons, New York, 1990.

[168] T. Kailath. *Linear Systems.* Prentice Hall, Englewood Cliffs, New Jersey 07632, 1980.

[169] R. E. Kalman. Mathematical description of linear dynamical systems. *J. SIAM Control, Ser. A*, 1(2):152–192, 1963.

[170] R. E. Kalman. On the computation of the reachable/observable canonical form. *SIAM J. Control and Optimization*, 20(2):258–260, March 1982.

[171] K. Kalsi, J. Lian, S. Hui, and S. H. Żak. Sliding-mode observers for systems with unknown inputs: A high-gain approach. *Automatica*, 46(2):347–353, February 2010.

[172] K. Kalsi, J. Lian, and S. H. Żak. Decentralized dynamic output feedback control of nonlinear interconnected systems. *IEEE Transactions on Automatic Control*, 55(8):1964–1970, August 2010.

[173] E. W. Kamen. Fundamentals of linear time-varying systems. In W. S. Levine, editor, *The Control Handbook: Control System Advanced Methods*, chapter 3, pages 3–1 – 3–33. CRC Press, Boca Raton, Florida, 2011.

[174] S.-J. Kang, C.-H. Woo, H.-S. Hwang, and K. B. Woo. Evolutionary design of fuzzy rule base for nonlinear system modeling and control. *IEEE Transactions on Fuzzy Systems*, 8(1):37–45, February 2000.

[175] K. Kelly. *Out of Control: The New Biology of Machines, Social Systems and the Economic World*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.

[176] T. Kenjo. *Stepping Motors and Their Microprocessor Controls*. Clarendon Press, Oxford, 1984.

[177] T. Kenjo and S. Nagamori. *Permanent-Magnet and Brushless DC Motors*. Clarendon Press, Oxford, 1985.

[178] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ 07458, second edition, 1996.

[179] T. Khanna. *Foundations of Neural Networks*. Addison-Wesley Publishing, Reading, MA, 1990.

[180] E. Kim, M. Park, S. Ji, and M. Park. A new approach to fuzzy modeling. *IEEE Transactions on Fuzzy Systems*, 5(3):328–337, August 1997.

[181] D. E. Kirk. *Optimal Control Theory: An Introduction*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1970.

[182] G. J. Klir, U. St.Clair, and B. Yuan. *Fuzzy Set Theory: Foundations and Applications*. Prentice Hall PTR, Upper Saddle River, NJ 07458, 1997.

[183] C. Knospe. PID control—Introduction to the special section. *IEEE Conrol Systems Magazine*, 26(1):30–31, February 2006.

[184] D. E. Knuth. *The Art of Computer Programming*, volume 1. Fundamental Algorithms. Addison-Wesley, Reading, Massachusetts, second edition, 1973.

[185] C. Koch. Computation and the single neuron. *Nature*, 385(6613):207–210, 16 January 1997.

[186] A. J. Koshkouei and A. S. I. Zinober. Sliding mode state observation for non-linear systems. *Int. J. Control*, 77(2):118–127, 2004.

[187] B. Kosko. *Fuzzy Engineering*. Prentice-Hall, Upper Saddle River, NJ 07458, 1997.

[188] B. Kouvaritakis and M. Cannon, editors. *Nonlinear Predictive Control: Theory and Practice*. The Institution of Electrical Engineers, London, 2001.

[189] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. A Bradford Book, The MIT Press, Cambridge, Massachusetts, 1992.

[190] J. R. Koza, F. H. Bennett, III, D. Andre, M. A. Keane, and F. Dunlap. Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Transactions on Evolutionary Computation*, 1(2):109–128, July 1997.

[191] T. Kozek, T. Roska, and L. O. Chua. Genetic algorithm for CNN template learning. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 40(6):392–402, June 1993.

[192] P. C. Krause and O. Wasynczuk. *Electromechanical Motion Devices*. McGraw-Hill Book Company, New York, 1989.

[193] A. J. Krener and W. Respondek. Nonlinear observers with linearizable error dynamics. *SIAM J. Control and Optimization*, 23(2):197–216, March 1985.

[194] H. Kroemer. Nano-whatever: Do we *really* know where we are heading? *Physica Status Solidi (a)*, 202(6):957–964, May 2005.

[195] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović. *Nonlinear and Adaptive Control Design*. John Wiley & Sons, Inc., New York, 1995.

[196] V. Kučera. A contribution to matrix quadratic equations. *IEEE Transactions on Automatic Control*, AC-17(3):344–347, June 1972.

[197] P. Kudva, N. Viswanadham, and A. Ramakrishna. Observers for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, AC-25(1):113–115, February 1980.

[198] J. E. Kurek. The state vector reconstruction for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, AC-28(12):1120–1122, December 1983.

[199] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, a Division of J. Wiley & Sons, Inc., New York, 1972.

[200] C. M. Kwan. On variable structure output feedback controllers. *IEEE Transactions on Automatic Control*, 41(11):1691–1693, November 1996.

[201] V. Kyrylov, L. A. Severyanova, and A. Vieira. Modeling robust oscillatory behavior of the hypothalamic-pituitary-adrenal axis. *IEEE Transactions on Biomedical Engineering*, 52(12):1977–1983, December 2005.

[202] B. Lamb. *Booknotes: America's Finest Authors on Reading, Writing, and the Power of Ideas*. Times Books, New York, 1997.

[203] S. Lang. *Calculus of Several Variables*. Springer-Verlag, New York, third edition, 1987.

[204] J. P. LaSalle. *The Stability and Control of Discrete Processes*. Springer-Verlag, New York, 1986.

[205] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory*. Robert E. Krieger Publishing Company, Malabar, Florida, 1986.

[206] Y. Lee, J. Q. Gong, B. Yao, and S. H. Żak. Fuzzy adaptive robust tracking control of a class of nonlinear systems. In *Proc. 2001 American Control Conf.*, pages 4040–4045, 2001. Arlington, VA, June 25–27, 2001.

[207] Y. Lee and S. H. Żak. Designing a genetic neural fuzzy antilock-brake-system controller. *IEEE Transactions on Evolutionary Computation*, 6(2):198–211, April 2002.

[208] J. R. Leigh. *Modelling and Simulation*. IEE Topics in Control Series 1. Peter Peregrinus Ltd., London, 1983.

[209] Y. Lenbury and P. Pornsawad. A delay-differential equation model of the feedback-controlled hypothalamus-pituitary-adrenal axis in humans. *Mathematical Medicine and Biology*, 22(1):15–33, March 2005.

[210] F. L. Lewis and V. L. Syrmos. *Optimal Control*. A Wiley-Interscience Publication, John Wiley & Sons, Inc., New York, second edition, 1995.

[211] Y. Li, K. H. Ang, and G. C. Y. Chong. PID control system analysis and design: Problems, remedies, and future directions. *IEEE Conrol Systems Magazine*, 26(1):32–41, February 2006.

[212] J. Lian, Y. Lee, S. D. Sudhoff, and S. H. Żak. Self-organizing radial basis function network for real-time approximation of continuous-time dynamical systems. *IEEE Transactions on Neural Networks*, 19(3):460–474, March 2008.

[213] J. Lian and S. H. Żak. Controller design for a class of uncertain systems with guaranteed performance. *SIAM J. Control and Optimization*, 49(3):1239–1261, May 2011.

[214] K.-Y. Lian, C.-S. Chiu, T.-S. Chiang, and P. Liu. Secure communications of chaotic systems with robust performance via fuzzy observer-based design. *IEEE Transactions on Fuzzy Systems*, 9(1):212–220, February 2001.

[215] C.-T. Lin and C. S. G. Lee. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems.* Prentice-Hall PTR, Upper Saddle River, NJ 07458, 1996.

[216] J.-S. Lin, T.-L. Liu, J.-J. Yan, and H.-T. Yau. Synchronization of unidirectional coupled chaotic systems with unknown channel time-delay: Adaptive robust observer-based appraoch. *Chaos, Solitons and Fractals*, 26:971–978, November 2005.

[217] R. P. Lippmann. An introductuion to computing with neural nets. *IEEE ASSP Magazine*, 4(2):4–22, April 1987.

[218] G. P. Liu, V. Kadirkamanathan, and S. A. Billings. Stable sequential identification of continuous nonlinear dynamical systems by growing radial basis function networks. *Int. J. Control*, 65(1):53–69, 10 September 1996.

[219] Y.-W. Liu, Z.-H. Hu, J.-H. Peng, and B.-Z. Liu. A dynamical model for the pulsatile secretion of the hypothalamo-pituitary-adrenal axis. *Mathematical and Computer Modeling*, 29(4):103–110, February 1999.

[220] B. P. Loop, S. D. Sudhoff, S. H. Żak, and E. L. Zivi. Estimating regions of asymptotic stability of power electronics systems using genetic algorithms. *IEEE Transactions on Control Systems Technology*, 18(5):1011–1022, September 2010.

[221] D. G. Luenberger. Observers for multivariable systems. *IEEE Transactions on Automatic Control*, AC-11(2):190–197, April 1966.

[222] D. G. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, AC-16(6):596–602, December 1971.

[223] D. G. Luenberger. *Introduction to Dynamic Systems: Theory, Models, and Applications.* John Wiley & Sons, New York, 1979.

[224] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley Publishing, Reading, Massachusetts, second edition, 1984.

[225] X.-J. Ma, Z.-Q. Sun, and Y.-Y. He. Analysis and design of fuzzy controller and fuzzy observer. *IEEE Transactions on Fuzzy Systems*, 6(1):41–51, February 1998.

[226] A. G. J. MacFarlane. The development of frequency-response methods in automatic control. *IEEE Transactions on Automatic Control*, AC-24(2):250–265, April 1979.

[227] I. J. Maddox. *Elements of Functional Analysis.* Cambridge University Press, Cambridge, Great Britain, second edition, 1988.

[228] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Machine Studies*, 7(1):1–13, January 1975.

[229] B. B. Mandelbrot. *The Fractal Geometry of Nature.* W. H. Freeman and Company, New York, 1983.

[230] R. Marino and P. Tomei. *Nonlinear Control Design: Geometric, Adaptive and Robust.* Prentice Hall Information and System Sciences Series. Prentice Hall Europe, London, 1995.

[231] J. C. Maxwell. On governors. *Proceedings of the Royal Society of London*, XVI:270–283, March 1868. Printed by Taylor and Francis, London, Red Lion Court, Fleet Street.

[232] R. M. May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467, June 10, 1976.

[233] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, July 1990.

[234] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.

[235] O. Mayr. *The Origins of Feedback Control.* The M.I.T. Press, Cambridge, Massachusetts, 1970.

[236] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[237] B. McEwen with E. Norton Lashley. *The End of Stress as We Know It.* Joseph Henry Press, Washington, DC, 2002.

[238] M. McIntyre, T. Burg, D. Dawson, and B. Xian. Adaptive state of charge (SOC) estimator for a battery. In *Proceedings of the 2006 American Control Conference, Minneapolis, Minnesota*, pages 5740–5744, June 14–16, 2006.

[239] J. M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall PTR, Upper Saddle River, NJ 07458, 2001.

[240] H. Michalska and D. Q. Mayne. Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40(6):995–1006, June 1995.

[241] A. N. Michel, J. A. Farrell, and W. Porod. Qualitative analysis of neural networks. *IEEE Transactions on Circuits and Systems*, 36(2):229–243, February 1989.

[242] R. K. Miller and A. N. Michel. *Ordinary Differential Equations*. Academic Press, Inc., New York, 1982.

[243] M. Mitchell. *An Introduction to Genetic Algorithms*. A Bradford Book, The MIT Press, Cambridge, Massachusetts, 1996.

[244] R. R. Mohler. *Nonlinear Systems; Volume I, Dynamics and Control*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1991.

[245] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20(4):801–836, October 1978.

[246] F. C. Moon. *Chaotic Vibrations: An Introduction for Applied Scientists and Engineers*. Wiley-Interscience, J. Wiley & Sons, New York, 1987.

[247] Ö. Morgül and E. Solak. Observer based synchronization of chaotic systems. *Physical Review E*, 54(5):4803–4811, November 1996.

[248] A. G. Nalecz. Sensitivity analysis of vehicle design attributes in frequency domain. *Vehicle System Dynamics*, 17:141–163, 1988.

[249] A. G. Nalecz and A. C. Bindemann. Analysis of the dynamic response of four wheel steering vehicles at high speed. *Int. J. of Vehicle Design*, 9(2):179–202, 1988.

[250] A. G. Nalecz and A. C. Bindemann. Investigation into the stability of four wheel steering vehicles. *Int. J. of Vehicle Design*, 9(2):159–178, 1988.

[251] S. Nara, P. Davis, and H. Totsuji. Memory search using complex dynamics in a recurrent neural network model. *Neural Networks*, 6(7):963–973, 1993.

[252] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming.* McGraw-Hill, New York, 1996.

[253] I. P. Natanson. *Theory of Functions of a Real Variable.* Frederick Ungar, New York, 1955.

[254] Nusawardhana, S. H. Żak, and W. A. Crossley. Nonlinear synergetic optimal control. *AIAA Journal of Guidance, Control, and Dynamics*, 30(4):1134–1147, July-August 2007.

[255] K. Ogata. *Modern Control Engineering.* Prentice Hall, Upper Saddle River, New Jersey 07458, third edition, 1997.

[256] K. Ogata. *Solutions Manual to Modern Control Engineering.* Prentice Hall, Upper Saddle River, New Jersey 07458, third edition, 1997.

[257] K. Ogata. *Modern Control Engineering.* Prentice Hall, Upper Saddle River, New Jersey 07458, fourth edition, 2002.

[258] J. O'Reilly. *Observers for Linear Systems.* Academic Press, London, 1983.

[259] E. Ott, C. Grebogi, and J. A. Yorke. Controlling chaos. *Physical Review Letters*, 64(11):1196–1199, 12 March 1990.

[260] D. H. Owens. *Multivariable and Optimal Systems.* Academic Press, London, 1981.

[261] S. K. Pal and D. Bhandari. Genetic algorithms with fuzzy fitness function for object extraction using cellular networks. *Fuzzy Sets and Systems*, 65(2/3):129–139, August 10, 1994.

[262] S. Pang, J. Farrell, J. Du, and M. Barth. Battery state-of-charge estimation. In *Proceedings of the 2001 American Control Conference, Arlington, VA*, pages 1644–1649, June 25–27, 2001.

[263] T. S. Parker and L. O. Chua. Chaos: A tutorial for engineers. *Proceedings of the IEEE*, 75(8):982–1008, August 1987.

[264] T. S. Parker and L. O. Chua. *Practical Numerical Algorithms for Chaotic Systems.* Springer-Verlag, New York, 1989.

[265] R. V. Patel and M. Toda. Quantitative measures of robustness for multivariable systems. In *Proceedings of Joint Automatic Control Conference (JACC), San Francisco, CA*, pages TP8–A, 1980.

[266] R. J. Patton and G. P. Liu. Robust control design via eigenstructure assignment, genetic algorithms and gradient-based optimization. *IEE Proceedings—Control Theory Appl.*, 141(3):202–208, May 1994.

[267] L. M. Pecora and T. L. Carroll. Synchronization in chaotic systems. *Physical Review Letters*, 64(8):821–824, 19 February 1990.

[268] W. Pedrycz. *Fuzzy Control and Fuzzy Sytems.* Research Studies Press Ltd., Taunton, Somerset, England, second, extended edition, 1993.

[269] W. Pedrycz, editor. *Fuzzy Modeling: Paradigms and Practice.* Kluwer Academic Publishers, Boston, 1996.

[270] H.-O. Peitgen, H. Jürgens, and D. Saupe, editors. *Chaos and Fractals: New Frontiers of Science.* Springer-Verlag, New York, 1992.

[271] H.-O. Peitgen and D. Saupe, editors. *The Science of Fractal Images.* Springer-Verlag, New York, 1988.

[272] A. L. Peressini, F. E. Sullivan, and J. J. Uhl, Jr. *The Mathematics of Nonlinear Programming.* Springer-Verlag, New York, 1988.

[273] I. Peterson. *Newton's Clock: Chaos in the Solar System.* W. H. Freeman and Co., New York, 1993.

[274] C. A. Pickover, editor. *Fractal Horizons: The Future Use of Fractals.* St. Martin's Press, New York, 1996.

[275] D. A. Pierre. *Optimization Theory With Applications.* Dover Publications, Inc., New York, 1986.

[276] L. S. Pontryagin. *Ordinary Differential Equations.* Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1962.

[277] V. M. Popov. Invariant description of linear, time-invariant controllable systems. *SIAM J. Control*, 10(2):252–264, May 1972.

[278] H. M. Power and R. J. Simpson. *Introduction to Dynamics and Control*. McGraw-Hill Book Company (UK) Limited, London, 1978.

[279] R. Pratap. *Getting Started with MATLAB 7: A Quick Introduction for Scientists and Engineers*. Oxford University Press, New York, 2006.

[280] F. H. Raven. *Automatic Control Engineering*. The McGraw-Hill Book Company, New York, fourth edition, 1987.

[281] Reader's Digest Practical Problem Solver. Substitutes, Shortcuts, and Ingenious Solutions for Making Life Easier. The Reader's Digest Association, Inc., Pleasantville, New York, 1991.

[282] K. Röbenack and A. F. Lynch. An efficient method for observer design with approximately linear error dynamics. *Int. J. Control*, 77(7):607–612, 10 May 2004.

[283] C. E. Rohrs, J. L. Melsa, and D. G. Schultz. *Linear Control Systems*. McGraw-Hill, Inc., New York, 1993.

[284] B. Ronen and S. Pass. *Focused Operations Management: Achieving More With Existing Resources*. John Wiley & Sons, Hoboken, New Jersey, 2008.

[285] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, Inc., New York, third edition, 1976.

[286] G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1):96–101, January 1994.

[287] W. J. Rugh. *Linear System Theory*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1993.

[288] W. J. Rugh. *Linear System Theory*. Prentice Hall, Upper Saddle River, New Jersey 07458, second edition, 1996.

[289] W. J. Rugh and G. J. Murphy. A new approach to the solution of linear optimal control problems. *Transactions of the ASME: Journal of Basic Engineering*, 91(2):149–154, June 1969.

[290] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. A Bradford Book, The MIT Press, Cambridge, Massachusetts, 1986.

[291] M. Saif and Y. Xiong. Sliding mode observers and their application in fault diagnosis. In F. Caccavale and L. Villani, editors, *Fault Diagnosis and Fault Tolerance for Mechatronic Systems*, pages 1–57. Springer-Verlag, Berlin, 2003.

[292] M. G. Salvadori and M. L. Baron. *Numerical Methods in Engineering*. Prentice-Hall, Inc., Englewood Cliffs, N.J., second edition, 1961.

[293] J. T. Sandefur. *Discrete Dynamical Systems; Theory and Applications*. Clarendon Press, Oxford, 1990.

[294] E. Santi, A. Monti, D. Li, K. Proddutur, and R.A. Dougal. Synergetic control for DC-DC boost converter: Implementation options. *IEEE Transactions on Industry Applications*, 39(6):1803–1813, November/December 2003.

[295] E. Santi, A. Monti, D. Li, K. Proddutur, and R.A. Dougal. Synergetic Control for Power Electronics Applications: A Comparison with the Sliding Mode Approach. *Journal of Circuits, Systems, and Computers*, 13(4):737–760, 2004.

[296] D. Sbarbaro and R. Peña. A non-linear wind velocity observer for a small wind energy system. In *Proceedings of the 39-th IEEE Conference on Decsions and Control*, pages 3086–3087, Sydney, Australia, December 2000.

[297] R. T. Seeley. *Calculus of Several Variables: An Introduction*. Scott, Foresman and Company, Glenview, Illinois 60025, 1970.

[298] G. M. Shepherd, editor. *The Synaptic Organization of the Brain*. Oxford University Press, New York, third edition, 1990.

[299] K.-Y. Siu, V. Roychowdhury, and T. Kailath. *Discrete Neural Computation; A Theoretical Fmundation*. Prentice Hall PTR, Englewood Cliffs, NJ 07632, 1995.

[300] R. E. Skelton. *Dynamic Systems Control: Linear Systems Analysis and Synthesis*. John Wiley & Sons, Inc., New York, 1988.

[301] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1991.

[302] D. E. Smith and J. M. Starkey. Effects of model complexity on the performance of automated vehicle steering controllers: controller development and evaluation. *Vehicle System Dynamics*, 23(8):627–645, November 1994.

[303] D. E. Smith and J. M. Starkey. Effects of model complexity on the performance of automated vehicle steering controllers: model development, validation and comparison. *Vehicle System Dynamics*, 24(2):163–181, March 1995.

[304] L. Smith. *Linear Algebra*. Springer-Verlag, New York, second edition, 1984.

[305] M. C. Smith. Special issue on control education—The United Kingdom. *IEEE Control Systems*, 16(2):51–56, April 1996.

[306] W. E. Snyder. *Industrial Robots: Computer Interfacing and Control*. Prentice Hall, Inc., Englewood Cliffs, New Jersey 07632, 1985.

[307] O. A. Solheim. Design of optimal control systems with prescribed eigenvalues. *Int. J. Control*, 15(1):143–160, 1972.

[308] J. A. Solsona and M. I. Valla. Disturbance and nonlinear Luenberger observers for estimating mechanical variables in permanent magnet synchronous motors under mechanical parameters uncertainties. *IEEE Transactions on Industrial Electronics*, 50(4):717–725, August 2003.

[309] R. Sommer. Control design for multivariable non-linear time-varying systems. *Int. J. Control*, 31(5):883–891, 1980.

[310] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer-Verlag, New York, second edition, 1998.

[311] R. Su. On the linear equivalents of nonlinear systems. *Systems & Control Letters*, 2(1):48–52, July 1982.

[312] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33, October 1988.

[313] K. K. Sundareswaran, P. J. McLane, and M. M. Bayoumi. Observers for linear systems with arbitrary plant disturbances. *IEEE Transactions on Automatic Control*, AC-22(5):870–871, October 1977.

[314] M. Sunwoo and K. C. Cheok. An application of explicit self-tuning controller to vehicle active suspension systems. In *Proceedings of the 29th Conference on Decision and Control, Honolulu, Hawaii*, pages 2251–2257, December 1990.

[315] H. J. Sussmann and J. C. Willems. 300 years of optimal control: From the brachystochrone to the maximum principle. *IEEE Control Systems*, 17(3):32–44, June 1997.

[316] K. Takaba. Robust servomechanism with preview action for polytopic uncertain systems. *Int. J. Robust and Nonlinear Control*, 10(2):101–111, February 2000.

[317] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:116–132, January/February 1985.

[318] C. P. Tan, F. Crusca, and M. Aldeen. Extended results on robust state estimation and fault detection. *Automatica*, 44(8):2027–2033, August 2008.

[319] C. P. Tan and C. Edwards. Sliding mode observers for robust detection and reconstruction of actuator and sensor faults. *Int. J. Robust and Nonlinear Control*, 13(5):443–463, 30 April 2003.

[320] C. P. Tan and C. Edwards. Robust fault reconstruction in uncertain linear systems using multiple sliding observers in cascade. *IEEE Transactions on Automatic Control*, 55(4):855–867, April 2010.

[321] K. Tanaka. Stability and stabilizability of fuzzy-neural-linear control systems. *IEEE Transactions on Fuzzy Systems*, 3(4):438–447, November 1995.

[322] K. Tanaka and M. Sugeno. Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems*, 45(2):135–156, January 24, 1992.

[323] Y. S. Tang, A. I. Mees, and L. O. Chua. Synchronization and chaos. *IEEE Transactions on Circuits and Systems*, CAS-30(9):620–626, September 1983.

[324] D. W. Tank and J. J. Hopfield. Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems*, CAS-36(5):533–541, 1986.

[325] M. C. M. Teixeira, E. Assunção, and R. G. Avellar. On relaxed LMI-based designs for fuzzy regulators and fuzzy observers. *IEEE Transactions on Fuzzy Systems*, 11(5):613–623, October 2003.

[326] M. C. M. Teixeira and S. H. Żak. Stabilizing controller design for uncertain nonlinear systems using fuzzy models. *IEEE Transactions on Fuzzy Systems*, 7(2):133–142, April 1999.

[327] J. M. T. Thompson and H. B. Stewart. *Nonlinear Dynamics and Chaos.* John Wiley and Sons, Chichester, 1986.

[328] Y. I. Toptscheev and A. P. Tsyplyakov. *Problem Solver in the Theory of Automatic Control.* Mashinostroyeniee, Moscow, 1977. In Russian.

[329] J. G. Truxal. *Introductory System Engineering.* McGraw-Hill Book Company, New York, 1972.

[330] V. Utkin, J. Guldner, and J. Shi. *Sliding Mode Control in Electromechanical Systems.* Taylor & Francis, London, 1999.

[331] V. I. Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, AC-22(2):212–222, April 1977.

[332] V. I. Utkin. *Sliding Modes and Their Application in Variable Structure Systems.* Mir Publishers, Moscow, 1978.

[333] V. I. Utkin. *Sliding Modes in Control and Optimization.* Springer-Verlag Berlin, Heidelberg, 1992.

[334] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.

[335] D. J. Velleman. *How to Prove it: A Structured Approach.* Cambridge University Press, Cambridge, 1994.

[336] M. Vidyasagar. *Nonlinear Systems Analysis.* Prentice Hall, Englewood Cliffs, New Jersey 07632, second edition, 1993.

[337] T. L. Vincent. Control using chaos. *IEEE Control Systems*, 17(6):65–76, December 1997.

[338] B. L. Walcott, M. J. Corless, and S. H. Żak. Comparative study of non-linear state-observation techniques. *Int. J. Control*, 45(6):2109–2132, 1987.

[339] B. L. Walcott and S. H. Żak. State observation of nonlinear uncertain dynamical systems. *IEEE Transactions on Automatic Control*, AC-32(2):166–170, February 1987.

[340] B. L. Walcott and S. H. Żak. Combined observer-controller synthesis for uncertain dynamical systems with applications. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):88–104, January/February 1988.

[341] H. O. Wang, K. Tanaka, and M. F. Griffin. An approach to fuzzy control of nonlinear systems: Stability and design issues. *IEEE Transactions on Fuzzy Systems*, 4(1):14–23, February 1996.

[342] L. Wang. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, London, 2009.

[343] L.-X. Wang. Stable adaptive fuzzy control of nonlinear systems. *IEEE Transactions on Fuzzy Systems*, 1(2):146–155, May 1993.

[344] L.-X. Wang. *Adaptive Fuzzy Systems and Control; Design and Stability Analysis*. PTR Prentice Hall, Englewood Cliffs, New Jersey 07632, 1994.

[345] L.-X. Wang. *A Course in Fuzzy Systems and Control*. Prentice Hall PTR, Upper Saddle River, NJ 07458, 1997.

[346] S.-H. Wang, E. J. Davison, and P. Dorato. Observing the states of systems with unmeasurable disturbances. *IEEE Transactions on Automatic Control*, AC-20(5):716–717, October 1975.

[347] R. Weinstock. *Calculus of Variations: With Applications to Physics and Engineering*. Dover Publications, Inc., New York, 1974.

[348] P. Welander. Understanding derivative in PID control. *Control Engineering*, 57(2):24–27, February 2010.

[349] J. R. Wertz, editor. *Spacecraft Attitude Determination and Control*. Astrophysics and Space Science Library, Vol. 73. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[350] A. B. Will and S. H. Żak. Modeling and control of an automated vehicle. *Vehicle System Dynamics*, 27(3):131–155, March 1997.

[351] J. L. Willems. *Stability Theory of Dynamical Systems*. John Wiley & Sons, New York, 1970.

[352] W. A. Wolovich. *Linear Multivariable Systems*. Springer-Verlag, New York, 1974.

[353] W. A. Wolovich. *Automatic Control Systems: Basic Analysis and Design*. Saunders College Publishing, Harcourt Brace College Publishers, Fort Worth, 1994.

[354] J. Y. Wong. *Theory of Ground Vehicles*. J. Wiley & Sons, New York, second edition, 1993.

[355] W. M. Wonham. On pole assignment in multi-input controllable linear systems. *IEEE Transactions on Automatic Control*, AC-12(6):660–665, December 1967.

[356] J. Xiang, H. Su, and J. Chu. On the design of Walcott-Zak sliding mode observer. In *Proceedings of the 2005 American Control Conference, Portland, OR*, pages 2451–2456, June 8–10, 2005.

[357] X.-G. Yan and C. Edwards. Nonlinear robust fault reconstruction and estimation using a sliding mode observer. *Automatica*, 43(9):1605–1614, September 2007.

[358] X.-G. Yan and C. Edwards. Robust decentralized actuator fault detection and estimation for large-scale systems using a sliding mode observer. *Int. J. Control*, 81(4):591–606, April 2008.

[359] X.-G. Yan and C. Edwards. Robust sliding mode observer-based actuator fault detection and isolation for a class of nonlinear systems. *Int. J. Systems Science*, 39(4):349–359, April 2008.

[360] F. Yang and R. W. Wilde. Observers for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, 33(7):677–681, July 1988.

[361] T. Yang, C.-M. Yang, and L.-B. Yang. A detailed study of adaptive control of chaotic systems with unknown parameters. *Dynamics and Control*, 8(3):255–267, July 1998.

[362] Z.-J. Yang, T. Hachino, and T. Tsuji. On-line identification of continuous time-delay systems combining least-squares techniques with a genetic algorithm. *Int. J. Control*, 66(1):23–42, 10 January 1997.

[363] X. Yu. Controlling chaos using input-output linearization approach. *Int. J. Bifurcation and Chaos*, 7(7):1659–1664, July 1997.

[364] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.

[365] S. H. Żak. *Systems and Control*. Oxford University Press, New York, 2003.

[366] S. H. Żak and S. Hui. On variable structure output feedback controllers for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, 38(10):1509–1512, October 1993.

[367] S. H. Żak and S. Hui. Output feedback variable structure controllers and state estimators for uncertain/nonlinear dynamic systems. *IEE Proceedings—Control Theory Appl.*, 140(1):41–50, January 1993.

[368] S. H. Żak and C. A. MacCarley. State-feedback control of non-linear systems. *Int. J. Control*, 43(5):1497–1514, 1986.

[369] S. H. Żak and B. L. Walcott. State observation of nonlinear control systems via the method of Lyapunov. In A. S. I. Zinober, editor, *Deterministic Control of Uncertain Systems*, chapter 16, pages 333–350. Peter Peregrinus Ltd., London, UK, 1990.

[370] S. H. Żak, B. L. Walcott, and S. Hui. Variable structure control and observation of nonlinear/uncertain systems. In K.-K. D. Young, editor, *Variable Structure Control for Robotics and Aerospace Applications*, chapter 4, pages 59–88. Elsevier, Amsterdam, 1993.

[371] M. Zamparelli. Genetically trained cellular neural networks. *Neural Networks*, 10(6):1143–1151, August 1997.

[372] M. Zeitz. Comments on 'Comparative study of non-linear state-observation techniques'. *Int. J. Control*, 46(5):1861–1863, 1987.

[373] Y. Zeng and S. N. Singh. Adaptive control of chaos in Lorenz system. *Dynamics and Control*, 7(2):143–154, April 1997.

[374] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Transactions of the American Society of Mechanical Engineers*, 64:759–768, November 1942.

[375] A. S. I. Zinober, editor. *Deterministic Control of Uncertain Systems*. Peter Peregrinus Ltd., London, 1990.

[376] M. Zribi and J. Chiasson. Position control of a PM stepper motor by exact linearization. *IEEE Transactions on Automatic Control*, 36(5):620–625, May 1991.

[377] J. M. Zurada. *Introduction to Artificial Neural Systems.* West Publishing, St. Paul, MN, 1992.

Index