ECE 675—Linear Matrix Inequalities

Stan Żak

School of Electrical and Computer Engineering
Purdue University

zak@purdue.edu

March 9, 2022

Outline

- Motivation
- Definitions of convex set and convex function
- Linear matrix inequality (LMI)
- Canonical LMI
- Example of LMIs
- LMI solvers

The Lyapunov theorem

Lyapunov's thm:

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

The Lyapunov thm re-stated

Lyapunov's thm:

The 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

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} \prec 0$$

Equivalently,

$$-\boldsymbol{A}^\top \boldsymbol{P} - \boldsymbol{P}\boldsymbol{A} \succ 0$$

$A^\top P + PA \prec 0$ re-stated

- Let

$$P = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ x_2 & x_{n+1} & \cdots & x_{2n-1} \\ \vdots & & & \vdots \\ x_n & x_{2n-1} & \cdots & x_q \end{bmatrix}$$

- Define

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Defining P_is

- $$\boldsymbol{P}_q = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$
- Note that each $\boldsymbol{P}_i$ has only non-zero elements corresponding to x_i in $\boldsymbol{P}$
- Let
$$\boldsymbol{F}_i = -\boldsymbol{A}^\top \boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{A}, \quad i = 1, 2, \ldots, q$$

Manipulate

- $$\begin{aligned} \boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} &= x_1\left(\boldsymbol{A}^\top \boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A}\right) \\ &\quad + x_2\left(\boldsymbol{A}^\top \boldsymbol{P}_2 + \boldsymbol{P}_2\boldsymbol{A}\right) \\ &\quad + \cdots + x_q\left(\boldsymbol{A}^\top \boldsymbol{P}_q + \boldsymbol{P}_q\boldsymbol{A}\right) \\ &= -x_1\boldsymbol{F}_1 - x_2\boldsymbol{F}_2 - \cdots - x_q\boldsymbol{F}_q \\ &\prec 0 \end{aligned}$$
- Let $\boldsymbol{F}(\boldsymbol{x}) = x_1\boldsymbol{F}_1 + x_2\boldsymbol{F}_2 + \cdots + x_q\boldsymbol{F}_q$

Lyapunov's Inequality restated

- $$\boldsymbol{P} = \boldsymbol{P}^\top \succ 0 \quad \text{and} \quad \boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} \prec 0$$

  if and only if

  $$\boldsymbol{F}(\boldsymbol{x}) \succ 0$$

- Equivalently,

  $$\begin{bmatrix} \boldsymbol{P} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{A}^\top \boldsymbol{P} - \boldsymbol{P}\boldsymbol{A} \end{bmatrix} \succ 0$$

Linear Matrix Inequality

- 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 $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top$
- Construct an affine function

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{x}) &= \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \ldots + x_n\boldsymbol{F}_n \\ &= \boldsymbol{F}_0 + \sum_{i=1}^{n} x_i \boldsymbol{F}_i \end{aligned}$$

Linear Matrix Inequality—Definition

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \cdots + x_n\boldsymbol{F}_n \succ 0$$

Find a set of vectors $\boldsymbol{x}$ such that

$$\boldsymbol{z}^\top \boldsymbol{F}(\boldsymbol{x})\boldsymbol{z} > 0 \text{ for all } \boldsymbol{z} \in \mathbb{R}^m, \ \ \boldsymbol{z} \neq \boldsymbol{0},$$

that is, $\boldsymbol{F}(\boldsymbol{x})$ is positive definite

Linear Matrix Inequality—Another Definition

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \cdots + x_n\boldsymbol{F}_n \succeq 0$$

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 semidefinite

Convex Set

Definition

A set $\boldsymbol{\Omega} \subseteq \mathbb{R}^n$ is convex if for any $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\boldsymbol{\Omega}$, the line segment between $\boldsymbol{x}$ and $\boldsymbol{y}$ lies in $\boldsymbol{\Omega}$, that is,

$$\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y} \in \boldsymbol{\Omega} \quad \text{for any} \quad \alpha \in (0,1)$$

Convex Function

Definition

A real-valued function

$$f : \boldsymbol{\Omega} \to \mathbb{R}$$

defined on a convex set $\boldsymbol{\Omega} \subseteq \mathbb{R}^n$ is convex if for all $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{\Omega}$ and all $\alpha \in (0, 1)$,

$$f(\alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y}) \leq \alpha f(\boldsymbol{x}) + (1 - \alpha) f(\boldsymbol{y})$$

Convex Optimization Problem

Definition

A convex optimization problem is the one where the objective function to be minimized is convex and the constraint set, over which we optimize the objective function, is a convex set.

Warning: If f is a convex function, then

$$\begin{aligned} &\max f(\boldsymbol{x}) \\ &\text{subject to } \boldsymbol{x} \in \boldsymbol{\Omega} \end{aligned}$$

is NOT a convex optimization problem!

Another LMI Example

A system of LMIs,

$$\boldsymbol{F}_1(\boldsymbol{x}) \succeq 0,\ \boldsymbol{F}_2(\boldsymbol{x}) \succeq 0, \ldots, \boldsymbol{F}_k(\boldsymbol{x}) \succeq 0$$

can be represented as one single LMI

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{F}_1(\boldsymbol{x}) & & & \\ & \boldsymbol{F}_2(\boldsymbol{x}) & & \\ & & \ddots & \\ & & & \boldsymbol{F}_k(\boldsymbol{x}) \end{bmatrix} \succeq 0$$

Yet Another LMI Example

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 $\boldsymbol{A}$

Example # 2 Contd

- View each scalar inequality as an LMI
- Represent m LMIs as one LMI,

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} b_1 - \boldsymbol{a}_1^\top \boldsymbol{x} & & & \\ & b_2 - \boldsymbol{a}_2^\top \boldsymbol{x} & & \\ & & \ddots & \\ & & & b_m - \boldsymbol{a}_m^\top \boldsymbol{x} \end{bmatrix} \succeq 0$$

Semidefinite Programming Program

- A semidefinite programming problem is a convex optimization problem of the form

$$\begin{array}{rl} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{F}(\boldsymbol{x}) \succeq 0 \end{array}$$

- Note that the linear objective function $\boldsymbol{c}^\top \boldsymbol{x}$ is a convex function and the constraint set, $\{\boldsymbol{x} : \boldsymbol{F}(\boldsymbol{x}) \succeq 0\}$, is a convex set

Notation $>$ versus $\geq$

- 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 $\geq$

Converting LMIs Into Equivalent LMISs—Background Results

Lemma

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,

$$\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} \geq 0 \text{ if and only if } \boldsymbol{z}^\top \boldsymbol{M}^\top \boldsymbol{P}\boldsymbol{M}\boldsymbol{z} \geq 0,$$

that is,

$$\boldsymbol{P} \succeq 0 \text{ if and only if } \boldsymbol{M}^\top \boldsymbol{P}\boldsymbol{M} \succeq 0.$$

- Similarly

$$\boldsymbol{P} \succ 0 \text{ if and only if } \boldsymbol{M}^\top \boldsymbol{P}\boldsymbol{M} \succ 0.$$

The Schur Complements—Backround Results

- Suppose have a square block matrix

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix},$$

where $\boldsymbol{A} = \boldsymbol{A}^\top$ and $\boldsymbol{D} = \boldsymbol{D}^\top$

- Then, by the Lemma,

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix} \succeq 0$$

if and only if

$$\begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{O} \end{bmatrix} \succeq 0,$$

where $\boldsymbol{I}$ is an identity matrix of appropriate dimension

Background Results—Contd.

In other words,

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix} \succeq 0 \text{ if and only if } \begin{bmatrix} \boldsymbol{D} & \boldsymbol{B}^\top \\ \boldsymbol{B} & \boldsymbol{A} \end{bmatrix} \succeq 0.$$

The Schur Complements

- Given a square block matrix,

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix},$$

  where $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{22}$ are square and symmetric submatrices, and $\boldsymbol{A}_{12} = \boldsymbol{A}_{21}^{\top}$
- Suppose that the matrix $\boldsymbol{A}_{11}$ is invertible
- Then

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ -\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{21}^{\top} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{21}^{\top} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix}$$
$$= \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{21}^{\top} \end{bmatrix}$$

The Schur Complements—Contd.

- Let

$$\mathbf{\Delta}_{11} = \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{21}^{\top}.$$

- The matrix $\mathbf{\Delta}_{11}$ is called the Schur complement of $\boldsymbol{A}_{11}$
- By the Lemma

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{21}^{\top} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \succ 0 \text{ if and only if } \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{O} \\ \boldsymbol{O} & \mathbf{\Delta}_{11} \end{bmatrix} \succ 0$$

- That is,

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{21}^{\top} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \succ 0 \text{ if and only if } \boldsymbol{A}_{11} \succ 0 \text{ and } \mathbf{\Delta}_{11} \succ 0$$

The Schur Complement of $\boldsymbol{A}_{22}$

- Let
$$\boldsymbol{\Delta}_{22} = \boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{12}^{\top}.$$
- The matrix $\boldsymbol{\Delta}_{22}$ is called the Schur complement of $\boldsymbol{A}_{22}$
- By the Lemma
$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{12}^{\top} & \boldsymbol{A}_{22} \end{bmatrix} \succ 0 \text{ if and only if } \begin{bmatrix} \boldsymbol{\Delta}_{22} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{bmatrix} \succ 0$$
- That is,
$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{12}^{\top} & \boldsymbol{A}_{22} \end{bmatrix} \succ 0 \text{ if and only if } \boldsymbol{A}_{22} \succ 0 \text{ and } \boldsymbol{\Delta}_{22} \succ 0$$

Solving LMIs

- $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \cdots + x_n\boldsymbol{F}_n \succeq 0$ 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

LMI Solvers

- Three types of LMI solvers
- To test whether or not there exists a solution $\boldsymbol{x}$ to $\boldsymbol{F}(\boldsymbol{x}) \succ 0$ is called a **feasibility problem**
- Minimization of a linear objective under LMI constraints
- Generalized eigenvalue minimization problem

Solving the feasibility problem

- Can solve LMIs of the form

$$\boldsymbol{N}^\top \mathcal{L}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)\boldsymbol{N} \prec \boldsymbol{M}^\top \mathcal{R}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)\boldsymbol{M}$$

- $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k$—matrix variables
- $\boldsymbol{N}$—left outer factor, $\boldsymbol{M}$—right outer factor
- $\mathcal{L}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$—left inner factor, $\mathcal{R}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$—right inner factor

Left-hand side vs. the right-hand side

- The term "left-hand side" refers to what is on the "smaller" side of the inequality $0 \prec \boldsymbol{X}$
- In $\boldsymbol{X} \succ 0$, the matrix $\boldsymbol{X}$ is on the right-hand side—it is on the "larger" side of the inequality

A general structure for finding a feasible soln

```
setlmis([])
lmivar
lmiterm
.
.
.
lmiterm
getlmis
feasp
dec2mat
```

X=lmivar(type,structure)

- The input `type` specifies the structure of the variable `X`
- Three structures of matrix variables
- `type=1`—symmetric block diagonal matrix variable
- `type=2`—full rectangular matrix variable
- `type=3`—other cases

Second input of X=lmivar(type,structure)

- Additional info on the structure of the matrix variable $\boldsymbol{X}$
- Example

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{D}_2 & \cdots & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{D}_r \end{bmatrix}$$

- Each $\boldsymbol{D}_i$ is a square symmetric matrix—`type=1`
- r blocks—`structure` is $r \times 2$

The input structure

- The first component of each row of the input structure—corresponding block size
- The second element of each row—the block type
- `X=lmivar(1,[3 1])`
  full symmetric 3×3 matrix variable
- `X=lmivar(2,[2 3])`
  rectangular 2×3 matrix variable

Scalar block

- $$\boldsymbol{S} = \left[\begin{array}{cc|cc} s_1 & 0 & 0 & 0 \\ 0 & s_1 & 0 & 0 \\ -- & -- & -- & -- \\ 0 & 0 & s_2 & s_3 \\ 0 & 0 & s_3 & s_4 \end{array}\right],$$
- `S=lmivar(1,[2 0;2 1])` describes a scalar block matrix, $\boldsymbol{D}_1 = s_1 \boldsymbol{I}_2$
- The second block is a 2×2 symmetric full block

lmiterm(termid,A,B,flag)

- `termid`—row with four elements specify the terms of each LMI of the LMI system
- `termid(1)=n` to specify the left-hand side of the *n*-th LMI
- `termid(1)=-n` to specify the right-hand side of the *n*-th LMI
- `termid(2,3)=[i j]` specifies the term of the (i, j) block of the LMI specified by the first component

More on lmiterm(termid,A,B,flag)

- `termid(4)=0` for the constant term
- `termid(4)=X` for the variable term in the form $\boldsymbol{AXB}$
- `termid(4)=-X` for the variable term in the form $\boldsymbol{AX}^\top\boldsymbol{B}$

Second and third inputs in lmiterm(termid,A,B,flag)

- `A` and `B` give the value of the constant outer factors in the variable terms, $\boldsymbol{AXB}$ or in $\boldsymbol{AX}^\top\boldsymbol{B}$
- the flag input to `lmiterm` serves as a compact way to specify the expression

$$\boldsymbol{AXB} + (\boldsymbol{AXB})^\top$$

Using flag in lmiterm(termid,A,B,flag)

- `flag='s'` use for symmetrized expression
- $\boldsymbol{PA} + \boldsymbol{A}^\top \boldsymbol{P} \prec 0$

  ```
  lmiterm([1 1 1 P],1,A)
  lmiterm([1 1 1 -P],A',1)
  ```
- Note that $\boldsymbol{PA} + \boldsymbol{A}^\top \boldsymbol{P} = \boldsymbol{PA} + (\boldsymbol{PA})^\top$
- `lmiterm([1 1 1 P],1,A,'s')`

[tmin,xfeas]=feasp(lmis)

- Feasibility problem

$$\begin{array}{rl} \text{find} & \boldsymbol{x} \\ \text{such that} & \boldsymbol{L}(\boldsymbol{x}) \prec \boldsymbol{R}(\boldsymbol{x}) \end{array}$$

- `feasp` solves the auxiliary convex

$$\begin{array}{rl} \text{minimize} & t \\ \text{subject to} & \boldsymbol{L}(\boldsymbol{x}) \prec \boldsymbol{R}(\boldsymbol{x}) + t\boldsymbol{I}. \end{array}$$

P=dec2mat(lmis,xfeas,P)

- The system of LMIs is feasible if the minimal $t < 0$
- `P=dec2mat(lmis,xfeas,P)` converts the output of the LMI solver into matrix variables

Minimizer of a Linear Objective Subject to LMI Constraints

- Invoked using the function `mincx`
-

$$\begin{array}{rl} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}(\boldsymbol{x}) \prec \boldsymbol{B}(\boldsymbol{x}). \end{array}$$

- $\boldsymbol{A}(\boldsymbol{x}) \prec \boldsymbol{B}(\boldsymbol{x})$ is a shorthand notation for general structured LMI systems

State/Output Feedback Control

LTI System with output feedback control:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \\ u &= -Ky \end{aligned}$$

Goal: Design K to ensure asymptotic stability of $(A - BKC)$

- Matrix inequality for output-feedback controller design:

$$(A - BKC)^\top P + P(A - BKC) \prec 0, \ P \succ 0$$

- Simpler case: state-feedback ($C = I$)

$$(A - BK)^\top P + P(A - BK) \prec 0, \ P \succ 0$$

Simpler Case: State-Feedback Control

$$(A - BK)^\top P + P(A - BK) \prec 0, \; P \succ 0$$

- To-do: Find K, P
- Problem: Bi-linear matrix inequality in K and P
- **Technique #2**: Congruence transformation with $S \triangleq P^{-1}$ and $Z \triangleq KS$
- New inequalities

$$SA^\top + AS - SK^\top B^\top - BKS \prec 0$$

- LMIs:

$$\underbrace{SA^\top + AS}_{\text{linear in } S} - \underbrace{Z^\top B^\top - BZ}_{\text{linear in } Z} \prec 0, \; P \succ 0$$

- Get back $P = S^{-1}$, $K = ZP$

General Structure of CVX Code in MATLAB as summarized by Dr. Ankush Chakrabarty

```
cvx_begin sdp quiet
% sdp: semi-definite programming mode
% quiet: no display during computing
include CVX [variables]
% very intuitive variable initialization
% for example: variable P(3,3) symmetric
minimize([cost]) % convex function
subject to
[affine constraints] % preferably non-strict inequalities

cvx_end
disp(cvx_status) % solution status
```

Snippet in CVX by Dr. Ankush Chakrabarty

```
cvx_begin sdp

% Variable definition
variable S(n, n) symmetric
variable Z(m, n)

% LMIs
sys.A*S + S*sys.A' - sys.B*Z - Z'*sys.B' <= -eps*eye(n)
S >= eps*eye(n)

cvx_end
sys.K = Z/S; % compute K matrix
```

Output-Feedback Control

$$A^\top P + PA - C^\top K^\top B^\top P - PBKC \prec 0, \; P \succ 0$$

- To-do: Find K, P
- Problem: Bi-linear matrix inequality in K and P
- **Technique #3**: Choose M such that $BM = PB$ and $N \triangleq MK$
- New inequalities: $A^\top P + PA - C^\top K^\top MB^\top - BMKC \prec 0$
- Linear matrix (in)equalities:

$$\underbrace{A^\top P + PA}_{\text{linear in } P} - \underbrace{C^\top N^\top B^\top - BNC}_{\text{linear in } N} \prec 0, \; BM = PB, \; P \succ 0$$

- Get back $K = M^{-1}N$ (M is invertible if B has full column rank)

Snippet in CVX by Dr. Ankush Chakrabarty

Cool fact: CVX/YALMIP can handle *equality constraints*!

```
cvx_begin sdp quiet

% Variable definition
variable P(n, n) symmetric
variable N(m, p)
variable M(m, m)

% LMIs
P*sys.A + sys.A'*P - sys.B*N*sys.C ...
 - sys.C'*N'*sys.B' <= -eps*eye(n)
sys.B*M == P*sys.B
P >= eps*eye(n);

cvx_end
sys.K = M\N % compute K matrix
```

Observer Design

Plant model:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}$$

Linear observer:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

Goal: Design L to ensure asymptotic stability of the error dynamics

- Matrix inequality for observer design:

$$(A - LC)^\top P + P(A - LC) \prec 0, \ P = P^\top \succ 0$$

Observer Design—Contd.

$$A^\top P + PA - C^\top L^\top P - PLC \prec 0, \; P \succ 0$$

- To-do: Find L, P
- Problem: Bi-linear matrix inequality in L and P
- **Technique #1**: Choose $Y = PL$
- LMIs:

$$\underbrace{A^\top P + PA}_{\text{linear in } P} - \underbrace{C^\top Y^\top - YC}_{\text{linear in } Y} \prec 0, \; P \succ 0$$

- For robustness of solution, rewrite as

$$A^\top P + PA - C^\top Y^\top - YC + 2\alpha P \preceq 0, \; P \succ 0$$

  with fixed $\alpha > 0$
- Get back $L = P^{-1} Y$ ($P \succ 0$, hence invertible)

Snippet in CVX by Dr. Ankush Chakrabarty

```
cvx_begin sdp

% Variable definition
variable P(n, n) symmetric
variable Y(n, p)

% LMIs
P*sys.A + sys.A'*P - Y*sys.C - sys.C'*Y' + P <= 0
P >= eps*eye(n) % eps is a very small number in MATLAB

cvx_end
sys.L = P\Y; % compute L matrix
```