DIGITAL CONTROL

PRACTICAL EXPERIMENTAL SETUP
One important application area of advanced control systems is the precision motion control of various devices. As an example, the linear motor driven high-speed/high-accuracy X-Y table controlled by a digital signal processor (DSP) based digital control system is shown below. The entire system has been set-up in the School of Mechanical Engineering at Purdue University. The two axes (X and Y directions respectively) are driven by two different types of leading edge precision linear motors. Each axis of the table can move at a speed more than 2m/s with a position measurement resolution less than 1 micron and acceleration of several gs. The entire system is controlled by an advanced dSPACE product based digital servo control system. A Pentium PC functions as the user interface. For more information regarding the application, you can visit https://engineering.purdue.edu/~byao under the research item.

CONTINUOUS CONTROLLER vs DIGITAL CONTROLLER:
Continuous Control System (e.g., analog circuit):

Digital Control System Using a Microprocessor

Experimental Setup

Advantages of Digital Control System
Flexible and Universal:
Can implement any feedback gains and higher order controller. Overall system set-up is quite fixed.
Powerful:
Can implement advanced control strategies such as nonlinear controllers and complicated switching logic.
**Sampler**

The sampler is used to obtain the values of a continuous signal at each sampling instant $t_k = kT$, $k = 0, 1, 2, \cdots$, where $T$ is the sampling period. The sampled value is denoted as $x(kT)$, $k = 0, 1, 2, \cdots$.

**A/D Converter**

The A/D converter converts $x(kT)$, the sampled analog signal value at each sampling instant, into $x_d(kT)$, a digital number that a microprocessor can accept. $x_d(kT)$ is stored in bits and is normally chosen to be the digital number closest the analog value $x(kT)$. For example, for a 12 bits A/D convert that accepts an analog value between $\pm 10$ V, $x(kT)$ can be any value between $\pm 10$ V but a 12 bit A/D can only store $2^{12} = 4096$ digital numbers. So such an A/D convert only has a resolution of $20V/4096 = 0.0048$ V. The more bits the A/D convert has, the smaller the resolution, and the closer $x_d(kT)$ to $x(kT)$.

For simplicity, in this course, we think A/D’s resolution is very small and assume that $x_d(kT) = x(kT)$.

**D/A Converter**

D/A converter converts a digital number $u_d(kT)$, the digital control input, to an analog signal $u(kT)$, which has the same value, i.e., $u(kT) = u_d(kT)$.

**Hold Circuit**

The Hold Circuit generates a continuous control input signal that corresponds to the digital control input sequence $u(kT)$ at each sampling instant as:

$$u(t) = u(kT) \text{ when } kT \leq t < (k+1)T$$

i.e., holds the value $u(kT)$ until next sampling instant. Such a hold circuit is normally referred to as zero-order hold (ZOH).
DIGITAL CONTROLLER DESIGN via EMULATION

The easiest way to design a digital controller is by emulation or digital implementation of continuous controller, which proceeds as follows:

**Step 1:**
Design a continuous compensation or controller as done in previous lectures.

**Step 2:**
Digitize the continuous compensation or transfer function via certain approximation techniques such as the Euler’s method, Tustin’s Method, Matched Pole-zero (MPZ) method, etc.

**Step 3:**
Using discrete analysis, simulation, or experimentation to verify the design.

DIGITIZATION via Euler’s METHOD

Euler’s method is to approximate the derivative by a forward rectangular rule:

Approximate \( \dot{x}(kT) \) by:

\[
\dot{x}(kT) \approx \frac{x((k+1)T) - x(kT)}{T}
\]

Short hand notation:

\[
x(k) \equiv x(kT)
\]

\[
\dot{x}(k) \equiv \frac{x(k + 1) - x(k)}{T}
\]

**Ex.D1:** Digitization of Transfer Function

\[
e \rightarrow \begin{vmatrix} 2(s + 1) \\ s + 3 \end{vmatrix} u \quad \text{or} \quad C(s) = \frac{2(s + 1)}{s + 3}
\]

\[
E(s) = C(s) \Rightarrow \dot{u} + 3u = 2(\dot{e} + e)
\]

\[
\Rightarrow \dot{u}(kT) + 3u(kT) = 2 \left[ \dot{e}(kT) + e(kT) \right]
\]

Use Euler’s Method to approximate \( \dot{u}(kT) \) and \( \dot{e}(kT) \):

\[
\frac{u(k + 1) - u(k)}{T} + 3u(k) = 2 \left[ \frac{e(k + 1) - e(k)}{T} + e(k) \right]
\]

\[
\Rightarrow u(k + 1) = (1 - 3T)u(k) + 2e(k + 1) + 2(-1 + T)e(k)
\]

Or \( u(k) = (1 - 3T)u(k - 1) + 2e(k) + 2(-1 + T)e(k - 1) \)

The above difference equation can be used to calculate the control input at next sampling instant \( u(k+1) \) once given \( u(k), e(k), \) and \( e(k+1) \).

**Ex.D2:** Digitization of State-Space Model

Differential Equation in Continuous Domain:

\[
\begin{cases}
\dot{x}_c = Ax_c + Be \\
u = Cx_c
\end{cases} \Rightarrow \begin{cases}
\dot{x}_c(kT) = Ax_c(kT) + Be(kT) \\
u(kT) = Cx_c(kT)
\end{cases}
\]

Use the Euler’s Method to approximate \( \dot{x}_c(kT) \):

\[
\begin{cases}
\frac{x_c(k + 1) - x_c(k)}{T} = Ax_c(k) + Be(k) \\
u(k) = Cx_c(k)
\end{cases}
\]

\[
\Rightarrow \begin{cases}
x_c(k + 1) = (1 + TA)x_c(k) + TB e(k) \\
u(k) = Cx_c(k)
\end{cases}
\]

The above difference equation can be used to calculate the controller output \( u(k) \) to be commanded and the controller state \( x_c(k + 1) \) at the next sampling instance once given the feedback signal \( e(k) \) and the initial condition of the controller state \( x_c(0) \).