Today: Intro to State Space (ss)

Lecture 8
ECE 147B
Digital Control

Reading: FPW 4.3.3, 7.5
or: FV 7.1, 7.2, 6.6

Topics:
- Root locus (Review from Lecture 7)
- Step response calculation: CT (ECE 147A Review)
- Direct design
- e^{sT} approximation (for small "sT")
- State space equations

Next Class: State space (ss).
- FPW: 6.1

or: - FV: 7.6, 7.7

Root Locus

The root locus rules for z-plane and s-plane are identical!

What is different is the meaning of the resulting pole locations, i.e., stable poles in the left-hand plane (CT) versus within the unit circle (DT).

$$P(s) = \frac{1}{(s+1)(s+3)}$$

$$G(z) = \frac{1}{(z-.4)(z-.8)}$$

Root Locus

The root locus rules for z-plane and s-plane are identical!

$$P(s) = \frac{1}{(s+0.2)^4}$$

$$G(z) = \frac{1}{(z+0.2)^4}$$

Root Locus

Root locus intuition...

$$G(z) = \frac{1}{(z^n + Mz^{n-1} + \ldots + K)}$$

CT Step Response: Analytic Solution

Last lecture, we discussed how to do this for a DT system. As review, here is a methodology for 2nd-order CT systems:

<u>Review</u> – How to calculate time response (e.g., step response) for CT system.

Example: $G(s) = \dfrac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \dfrac{Y(s)}{U(s)}$

① Form will be: $y(t) = K + C_1 e^{s_1 t} + C_2 e^{s_2 t}$

...which can also be: $y(t) = K\left(1 + A\sin(\omega_d t + \phi)e^{-\sigma t}\right)$

where: $\begin{cases} \omega_d = \omega_n\sqrt{1-\zeta^2} \\ \sigma = \zeta\omega_n \end{cases}$

CT Step Response: Analytic Solution

② @ $t=0$, $\dot{y}(t)=0$ (init. cond is zero. 2nd order system.) $t=0$ $t=0$

$\therefore\ 0 = A\omega_d\cos(\omega_d t + \phi)e^{-\sigma t} - A\sigma\sin(\omega_d t + \phi)e^{-\sigma t}$ $t=0$ $t=0$

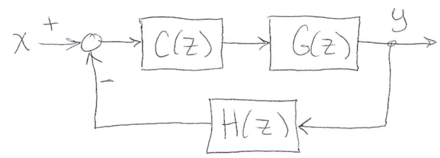
$0 = A\omega_d\cos(\phi) - A\sigma\sin(\phi)$

$\dfrac{\omega_d}{\sigma} = \dfrac{\sin(\phi)}{\cos(\phi)} \rightarrow \therefore \boxed{\phi = \operatorname{atan}\left(\dfrac{\omega_d}{\sigma}\right)}$

③ @ $t=0$, $y=0 = K(1 + A\sin(\phi))$

$\therefore\ A\sin(\phi) = -1 \rightarrow \boxed{A = \dfrac{-1}{\sin(\phi)}}$

"Direct Design"



$$P = \frac{Y(z)}{X(z)}$$

Example (MATLAB demo):

$$G(z) = \frac{1}{z - 0.8}$$

$$P(z) = \frac{1}{z}$$

$$H(z) = 1$$

"Direct Design"

Important Notes:

1. Controller "C(z)" cannot have unstable poles!

2. C(z) must also be CAUSAL.

e^{sT} Approximation (for small sT)

A hint when using "pole zero matching"…

$$z \Leftrightarrow e^{sT}$$

When sT << 1 (small in magnitude):

$$e^{sT} \approx 1 + sT$$

Analogously, where |z-1| << 1
("z-1" small in mag):

$$\log(z) \approx z - 1$$

e^{sT} Approximation (for small sT)

$z \Leftrightarrow e^{sT}$

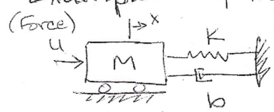
MATLAB examples…

$$e^{sT} \approx 1 + sT \approx z$$

$$\log(z) \approx z - 1 \approx s$$

State Space: Introduction

CT Example: spring-mass-damper (2nd-order) system



$$m\ddot{x} + b\dot{x} + kx = u$$

(equation(s) of motion...)

Let $X = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$, $Y = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C} X + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{D} u$

$$\dot{X} = AX + Bu$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{A} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{B} u$$

State Space: Introduction

vs Alternate form: $X = \begin{bmatrix} \dot{x} \\ x \end{bmatrix}$, $Y = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$

$$Y = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{C} X + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{D} u$$

$$\dot{X} = \begin{bmatrix} \ddot{x} \\ \dot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} -b/m & -k/m \\ 1 & 0 \end{bmatrix}}_{A} \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + \underbrace{\begin{bmatrix} 1/m \\ 0 \end{bmatrix}}_{B} u$$

Matrix form Equations of Motion

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

DT: $\boxed{\begin{aligned}x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k)\end{aligned}}$

matrix equations, for multi-input, multi-output systems ("MIMO")

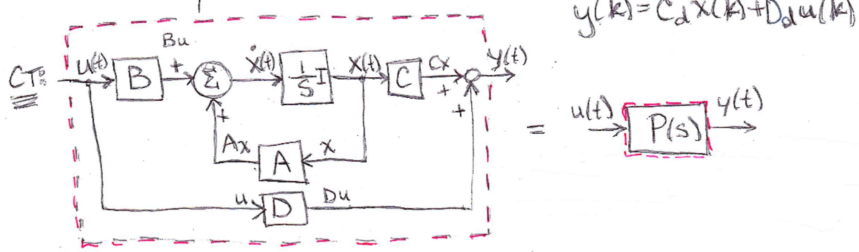
State Space: CT vs DT Block Diagrams

- CT vs DT ss models (Matrix equations...)

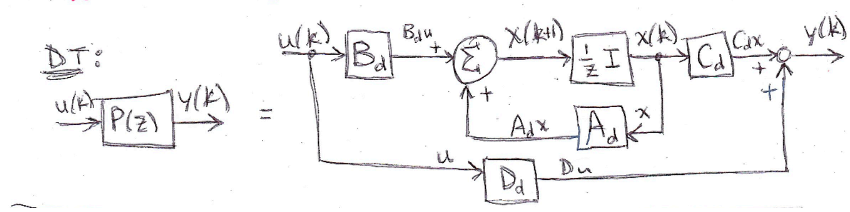
$\dot{x} = Ax + Bu$
$y = Cx + Du$

$x(k+1) = A_d x(k) + B_d u(k)$
$y(k) = C_d x(k) + D_d u(k)$

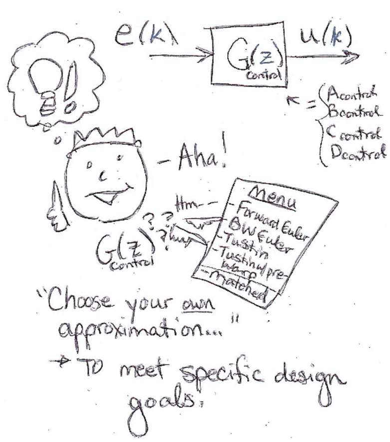
CT:

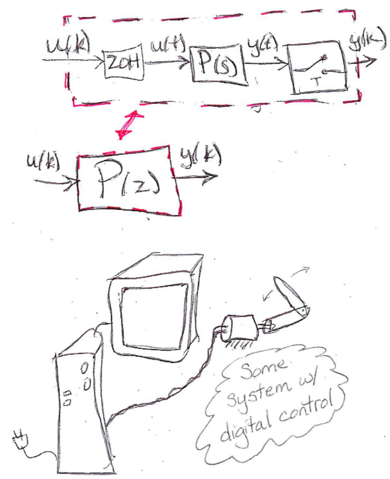


DT:



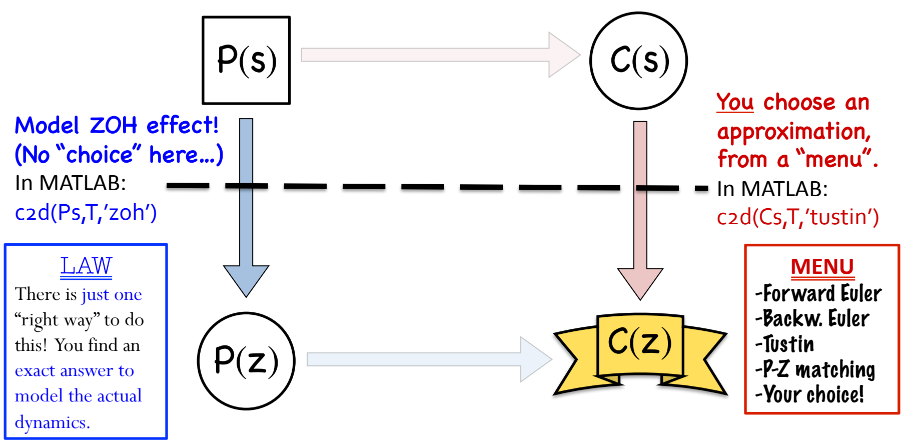
SS Representation: 2 cases…





Yes! This is EXACTLY as before!!!

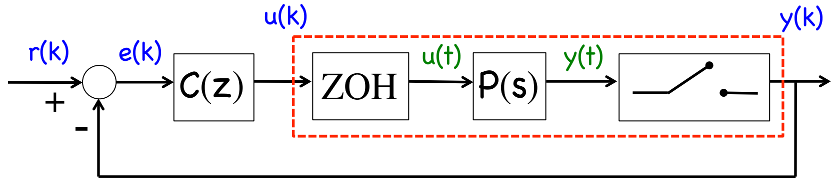
- There are two DIFFERENT ways we will represent dynamics via DT SS models:



Case 1: Numeric Integration

Case 1: Numeric Integration

Case 2: Modeling the ZOH plus sampling



Case 2: Modeling the ZOH plus sampling