Today: Pole Placement

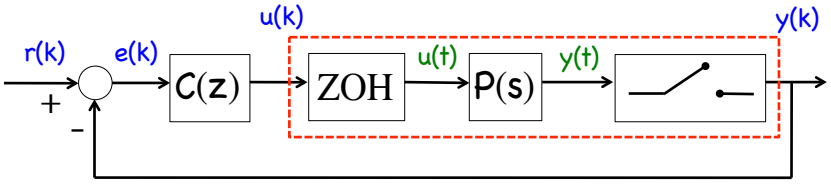
Lecture 12
ECE 147B
Digital Control

Reading: FPW 8.1
or: FV (9.1), 9.2

Topics:
- Modeling the ZOH / Review from last time
- State feedback: u=-Kx
- Control Canonical Form
- Pole Placement
- Controllability

Next Class: Estimator design via pole placement.
- FPW 8.2.1-8.2.4 <or> FV 9.5.1, (RS-12)

Modeling the ZOH - REVIEW



The Matrix Exponential:

$$e^M = \sum_{k=0}^{\infty} \frac{1}{k!} M^k$$

Example: $\ddot{x} + 2\dot{x} + 5x = u$
$(T = 0.01\text{sec})$

Forward Euler approximation:

State space: Block diagram of equations

1. Dynamics. (What internal system does, "open loop".)

2. Outputs. (Things we can MEASURE.)

3. Control: A way to MODIFY the dynamics. i.e., set u as a function of X, via a "control law", e.g., a "PD controller" (for example)...

State space: Poles and Zeros

1. Dynamics. (What internal system does, "open loop".) -> POLES

   Solutions for x≠0, such that u=0?

$$sx = Ax + Bu \qquad zx = A_d x + B_d u$$

2. Outputs. (Things we can MEASURE.) -> ZEROS

   Solutions for u≠0, such that y=0?

$$Y = Cx + Du \qquad Y = C_d x + D_d u$$

State space: CLOSED-LOOP Poles (and Zeros)

3. Control: A way to MODIFY the dynamics. i.e., set u as a function of X, via a "control law", e.g., a "PD controller" (for example)…

State feedback: u=-KX

$$sX = Ax + Bu = Ax - BKx \qquad zX = A_d x + B_d u = A_d x - B_d Kx$$

State feedback example

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

$$X \equiv \begin{bmatrix} \dot{x} \\ x \end{bmatrix}$$

$$K = \begin{bmatrix} ? \end{bmatrix}$$

- Say you are given the system dynamics and state variable definition at left. (m=1, b=2, k=5)
- What size must matrix "K" be?
- What values in K would result in closed loop poles with ω_n=10 (rad/s) and ζ =1?

"Control Canonical Form"

$$A = \begin{bmatrix} -a_1 & -a_2 & & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ & \vdots & \ddots & 0 & 0 \\ 0 & 0 & & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$K = \begin{bmatrix} K_1 & K_2 & \cdots & K_{n-1} & K_n \end{bmatrix}$$

$$A - BK = \begin{bmatrix} -a_1 - K_1 & -a_2 - K_2 & & -a_{n-1} - K_{n-1} & -a_n - K_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ & \vdots & \ddots & 0 & 0 \\ 0 & 0 & & 1 & 0 \end{bmatrix}$$

"Control Canonical Form"

Example:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

"Control Canonical Form"

Example:

$$A = \begin{bmatrix} 3 & -5 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

For state feedback control "u=-Kx", solve for K such that closed-loop poles are at s=-3, s=-3.

"Control Canonical Form"

Example:

$$A = \begin{bmatrix} 0 & 1 \\ -5 & 3 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

For state feedback control "u=-Kx", solve for K such that closed-loop poles are at s=-3, s=-3. HINT: Same dynamics but different definition of "X" here!

"Control Canonical Form"

Example:

$$A_d = \begin{bmatrix} 0.6 & -1.2 \\ 1 & 0 \end{bmatrix} \quad B_d = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

For state feedback control "u=-Kx", solve for K such that closed-loop poles are at z=0, z=0. This is "deadbeat" control!

Pole Placement

Comments:

1. By now, you should see (intuitively) how to solve for K such that "eig(A-BK)" [or "eig(A_d-B_dK)"] yield desired poles…
2. But our hand calculations would be TEDIOUS for large-dimensional, MIMO (multi-input multi-output) systems!

Is there a more efficient solution? → YES! This is "pole placement".

First, let's look at how to do this IN MATLAB:

```
A = rand(4,4); % some random, 4th-order open-loop system dynamics…
B = rand(4,1); % …including actuator inputs.
pdes = [-1 -2 -3 -4]; % desired poles (4 poles; 4th-order system…)
pnow = eig(A)  % open-loop poles
K = place(A,B,pdes) % "pole placement" command in matlab


K2 = acker(A,B,pdes) % alternate "pole placement" command


pact = eig(A-B*K)  % ACTUAL closed-loop poles!
```

Controllability

Given matrices A and B (for CT), or given A_d and B_d (for DT), the CONTROLLABILITY matrix is defined as:

$$C \equiv [B, \quad AB, \quad A^2B, \quad ..., \quad A^{n-1}B]$$

The system is CONTROLLABLE if and only if this matrix has FULL RANK (meaning it is INVERTIBLE).

Note the CHARACTERISTIC EQUATION for your desired CLOSED-LOOP (still nth-order) dynamics can be written as:

$$z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + ... + \alpha_n$$

Then, the GAIN MATRIX "K" needed will be: $K = [0...0,0,1]C^{-1}\alpha_c(A)$

Where $\alpha_c(A) \equiv A^n + \alpha_1 A^{n-1} + \alpha_2 A^{n-2} + ... + \alpha_n I$

Controllability - Example (see MATLAB)

```
n=4;
A = rand(n,n)  % n states
B = rand(n,1)  % single input here...
Co = ctrb(A,B)  % CONTROLLABILITY matrix
```

$$C \equiv [B, \quad AB, \quad A^2B, \quad ..., \quad A^{n-1}B]$$

```
RankCo = rank(Co)  % iff full rank -> Controllable (invertible)!
sdes = [1 2 3 4 5] % desired characteristic eqn
```

$$z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + ... + \alpha_n$$

```
pdes = roots(sdes) % desired poles
alpha_c = 0*A; % initialize
for ni = 1:length(sdes)
    alpha_c = alpha_c + sdes(ni)*A^(5-ni);
End
```

$$\alpha_c(A) \equiv A^n + \alpha_1 A^{n-1} + \alpha_2 A^{n-2} + ... + \alpha_n I$$

```
K = [zeros(1,n-1), 1] * inv(Co) * alpha_c

pact = eig(A-B*K)
```

$$K = [0...0,0,1]C^{-1}\alpha_c(A)$$

Controllability – More examples…

```
A = [-2 -5; 1 0];   % already control canonical form
B = [1; 0];
Co = ctrb(A,B)  % CONTROLLABILITY matrix
```

$$C \equiv [A, \quad AB] = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}$$

```
RankCo = rank(Co)  % iff full rank -> Controllable (invertible)!
sdes = [1 20 100] % desired characteristic eqn
```

$$s^2 + 20s + 100 = 0$$

```
pdes = roots(sdes) % desired poles
alpha_c = 0*A; % initialize
for ni = 1:length(sdes)
    alpha_c = alpha_c + sdes(ni)*A^(5-ni);
End
```

$$\alpha_c(A) \equiv A^2 + 20A + 100A^0 = \begin{bmatrix} 59 & -90 \\ 18 & 95 \end{bmatrix}$$

```
K = [zeros(1,n-1), 1] * inv(Co) * alpha_c
```

$$K = [0,1]C^{-1}\alpha_c(A) = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 59 & -90 \\ 18 & 95 \end{bmatrix}$$

```
pact = eig(A-B*K)
```