Today: Estimator design via pole placement

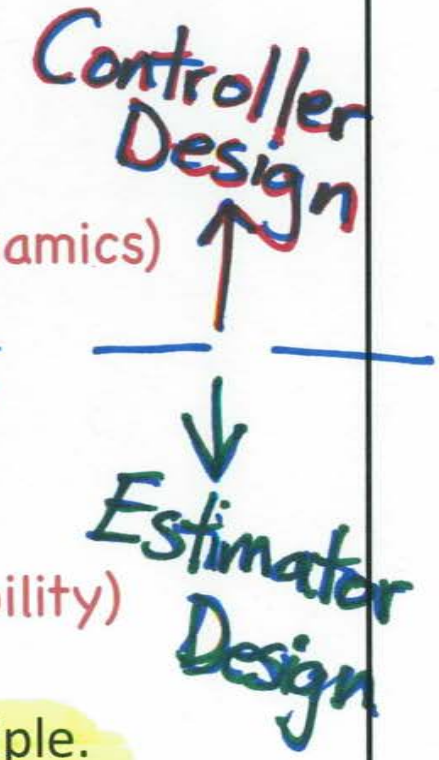
Lecture 13

ECE 147B
Digital Control

Reading: FPW 8.2.1-8.2.4
or: FV 9.5.1

Topics:
- Pole Placement (to control system dynamics)
- Controllability

- State estimation: open-loop approach
- State estimation: feedback approach
- Estimator design via pole placement
- Observability (compared to Controllability)



Next Class: Reduced-order estimators. Separation Principle.
- FPW 8.2.5, 8.3 <or> FV 9.5.2, 9.6 (RS-13)

"Control Canonical Form"

Coefficients go left-to-right.

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 & -a_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

L2R

(nx1) A (nxn) B

1. Off-diagonal "ONES", and otherwise all ZEROS below the first row.
2. B matrix ONLY non-zero in top row. – only affects "highest deriv" state. (CT)
3. All states are interrelated directly via either "s" (CT) or "z" (DT) operator:
   - Highest deriv state (CT) [largest "k" index (DT)] is "x1".
   - Lowest deriv state (CT) [lowest "k" index state (DT)] is "xn".

CT $X = \begin{bmatrix} \dddot{x} \\ \ddot{x} \\ \dot{x} \\ x \end{bmatrix} = \begin{bmatrix} s^3 X \\ s^2 X \\ sX \\ X \end{bmatrix}$ DT $X = \begin{bmatrix} x_{k+3} \\ x_{k+2} \\ x_{k+1} \\ x_k \end{bmatrix} = \begin{bmatrix} z^3 X \\ z^2 X \\ zX \\ X \end{bmatrix}$

most recent (z^{n-1}) ↑ oldest (z^0)

4. First row (therefore) gives NEGATIVE coeffs of char. eqn. (open-loop sys).

$$s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4 = 0$$

(or $z^4 + \dots$

Now: $\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dddot{x} \\ \ddddot{x} \end{bmatrix}$ vs "ROTATED" canonical form

If A^{L2R} & B^{L2R} each ROTATED 180°...

vs

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_4 & -a_3 & -a_2 & -a_1 \end{bmatrix} X + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

flipped! then

R2L

Rotate A & B by 180°... Coeffs now go "Right to Left"

C.L. char. eqn: $(s+3)(s+3)=0 = \boxed{s^2+6s+9=0}$ ↑ desired, C.L.

"Control Canonical Form" - REVIEW

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

$$K=\begin{bmatrix}K_1 & K_2\end{bmatrix}$$

True Canonical form...

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

L2R coeff's of CL. char. eqn.

$$A-BK=\begin{bmatrix}3-K_1 & -5-K_2\\ 1 & 0\end{bmatrix}$$

(C.L.: "closed loop"...)

↳ C.L. dyn: $s^2-(3-K_1)s-(-5-K_2)=0$

$s^2+(K_1-3)s+(K_2+5)=0$

$K_1-3=6 \rightarrow \boxed{K_1=9}$ ← 6

$K_2+5=9$ ← =9

$\boxed{K_2=4}$

$\boxed{K=[9,4]}$

...VS ROTATED canonical form...

$\boxed{K=[4,9]}$ C.L. dyn: $s^2+(K_2-3)s+(K_1+5)=0$

...K is Also rotated 180°...

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

Since A & B rotated 180°

$$K=\begin{bmatrix}K_1 & K_2\end{bmatrix}$$

$$A-BK=\begin{bmatrix}0 & 1\\ -5-K_1 & 3-K_2\end{bmatrix}$$

"R2L" ↔ (right-to-left) R2L coeff's of C.L. ch. eqn. ←

"Control Canonical Form" – DT example

Example:

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

$$K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$$

L2R → negative of coefs.

$$A - BK = \begin{bmatrix} 0.6 - K_1 & -1.2 - K_2 \\ 1 & 0 \end{bmatrix}$$

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

By inspection: O.L. dyn: $z^2 - 0.6z + 1.2 = 0$

C.L. dyn: $z^2 - (0.6 - K_1)z - (-1.2 - K_2) = 0$

$= 0$ $\qquad = 0$

We want:

$$\boxed{z^2 = 0}$$

$\therefore K_1 - 0.6 = 0$

$$\boxed{K_1 = 0.6}$$

$K_2 + 1.2 = 0$

$$\boxed{K_2 = -1.2}$$

For deadbeat: K is just the Top Row of A_d!

Pole Placement - for system dynamics

1. For systems written in CONTROL CANONICAL form:
   - EASY to solve for "K" for (arbitrary) closed-loop poles! You are DIRECTLY setting coefficients of the characteristic equation!
2. For "MESSIER" cases (i.e., A matrix NOT with an off-diagonal of "ones"):
   - Solving for "K" BY HAND would be HARD and TEDIOUS.

General solution? → "pole placement". ← Means taking a "messy", real-world definition of open-loop dynamics and FORCE this into "control canonical": A_c & B_c

First, let's look at MATLAB commands:

```
A = rand(4,4);  % some random, 4th-order sys…
B = rand(4,1);  % …including actuator inputs
pnow = eig(A)   % open-loop poles
pdes = [-1 -2 -3 -4]; % desired poles (4 poles; 4th-order system…)
```

(1) `K = place(A,B,pdes)` % "pole placement" command in matlab

↖ vector of desired closed-loop poles.

c: "canonical form"

place: (A) Generally better for "higher-order" (n > 10) systems.
(B) Poles cannot have multiplicity greater than rank(B).

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

acker. Can have repeated poles, BUT only "good" for $n \lesssim 10$.

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

Controllability

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

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

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

inv(C) or C^{-1}

Note the CHARACTERISTIC EQUATION for your desired CLOSED-LOOP (still nth-order) dynamics can be written as: ← You pick $\alpha_1 \to \alpha_n$ to set n poles.

$$z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + \ldots + \alpha_n = 0$$

GAIN MATRIX "K" needed will be:

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

depends only on I.C.

Where: $\alpha_c(A)$ is an (nxn) matrix!

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

Example: Start w/ some x_k. (initial condition)

? → How can we get to some desired end state? (x_{k+n}).

(Assume one input: B is (nx1)).

$$x_{k+1} = (Ax_k + Bu_k)$$

just 1 less (prev. time step)

$$x_{k+2} = A(x_{k+1}) + Bu_{k+1}$$

$$= A(Ax_k + Bu_k) + Bu_{k+1}$$

$$x_{k+3} = A\left[A(Ax_k + Bu_k) + Bu_{k+1}\right] + Bu_{k+2}$$

$$x_{k+4} = A\left\{A\left[A(Ax_k + Bu_k) + Bu_{k+1}\right] + Bu_{k+2}\right\} + Bu_{k+3}$$

$$x_{k+4} = A^4 x_k + A^3 Bu_k + A^2 Bu_{k+1} + \ldots$$
$$ABu_{k+2} + Bu_{k+3}$$

$$(x_{k+4} - A^4 x_k) = \underbrace{[B, AB, A^2B, A^3B]}_{C} \begin{bmatrix} u_{k+3} \\ u_{k+2} \\ u_{k+1} \\ u_k \end{bmatrix} \text{ "U"}$$

$$\therefore U = C^{-1}(x_{k+4} - A^4 x_k)$$

Can solve for U iff C invertible ... i.e. FULL RANK!

Controllability

What is the definition of "CONTROLLABILITY"?

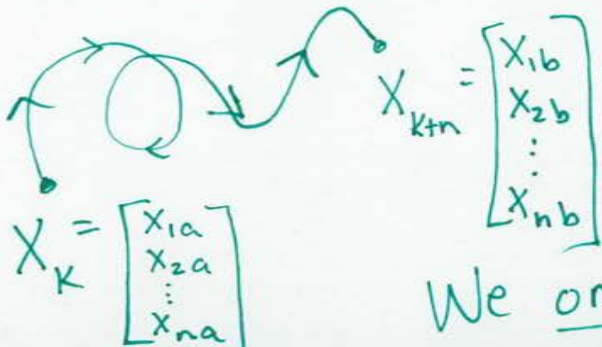
For a LINEAR system, there are two, equivalent definitions:

1. We can put n closed-loop poles anywhere (arbitrarily), by setting n values (or more) in gain matrix K:

   Wanted: $z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + \dots + \alpha_n = 0$

   ↑ Need at least n params in K to set n α values

2. We can take n values in state vector X from any arbitrary init. state to " " final state



↑ path not necc. "straight line"!

We only care about X_{k+n} at one instant in time.

Controllability - Pole Placement equations

On web: (1) google "byl". (2) Click "ECE Robotics at UCSB" link. (3) Click "ECE 147B" link. (4) goto "Lectures" tab. (5) Under "Lecture 13", download "example_controllability.m"

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

$\mathcal{C}$ is (nxn)

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

test for controllability

```
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
```

$\alpha_c(A)$ is (nxn) (like A)

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

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

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

inv(C)

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

pulls out last row, only.

Controllability – More MATLAB examples...

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

RankCo = rank(Co)  % iff full rank -> Controllable (invertible)!
sdes = [1 20 100]
% sdes: desired char eqn coef's
pdes = roots(sdes)
% pdes: desired poles
alpha_c = 0*A; % initialize
for ni = 1:length(sdes)
    alpha_c = alpha_c + sdes(ni)*A^(5-ni);
end

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

pact = eig(A-B*K)
% pact: CL poles
```

$A = \begin{bmatrix} -2 & -5 \\ 1 & 0 \end{bmatrix}$

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

1st row of A

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

↑ desired C.L. char. eqn.

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

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

just selects last row.

C^{-1}

$\alpha_c(A)$

$\therefore K = [18, \; 95]$ ← by inspection here, due to "control canonical" form...

State Estimation – Problem Statement

Sometimes, feedback control requires we "estimate" some of the states because, for example:

1. We might not sense all states in the output vector y, e.g.:

$$X = \begin{bmatrix} \dot{x} \\ x \end{bmatrix} \quad y = \underbrace{[0 \quad 1]}_{C} X + \underbrace{[0]}_{D} u = x$$

velocity ↑ position ↑ position only

$y = x$ ← position is measured.

Velocity is NOT. (As in lab, where we measure θ w/ encoder, but do NOT measure $\dot{\theta}$.)

2. Our measurements might be noisy, e.g.:

$$Y = CX + DU + V$$

V ← NOISE

PROBLEM STATEMENT: If we start with some "initial value" as an ESTIMATE of the state vector (for k=0), how can ESTIMATE the states for k>=0 (over time)?

intuitively (1) Reason about KNOWN system dynamics – "$sX = Ax + Bu$"

(2) Also "feed back" some info from sensors (measurements).

State Estimation – "Open loop" approach

← Use model (①) only.

Start with some "initial value" as an ESTIMATE of the states...
...and assume this vector of value evolves via "Ax+Bu" dynamics.

State: $x(k+1) = A_d x(k) + B_d u(k)$

Estimate of state: $\bar{x}(k+1) = A_d \bar{x}(k) + B_d u(k)$

error = (actual) − (estimate)

Error in the esimate: $e_x(k+1) = x(k+1) - \bar{x}(k+1)$ or: $\tilde{x}(k+1)$

↑ common notation.

↖ error : e_x – in our lectures...

Error dynamics?

$$x_{k+1} = A_d x_k + B_d u_k$$

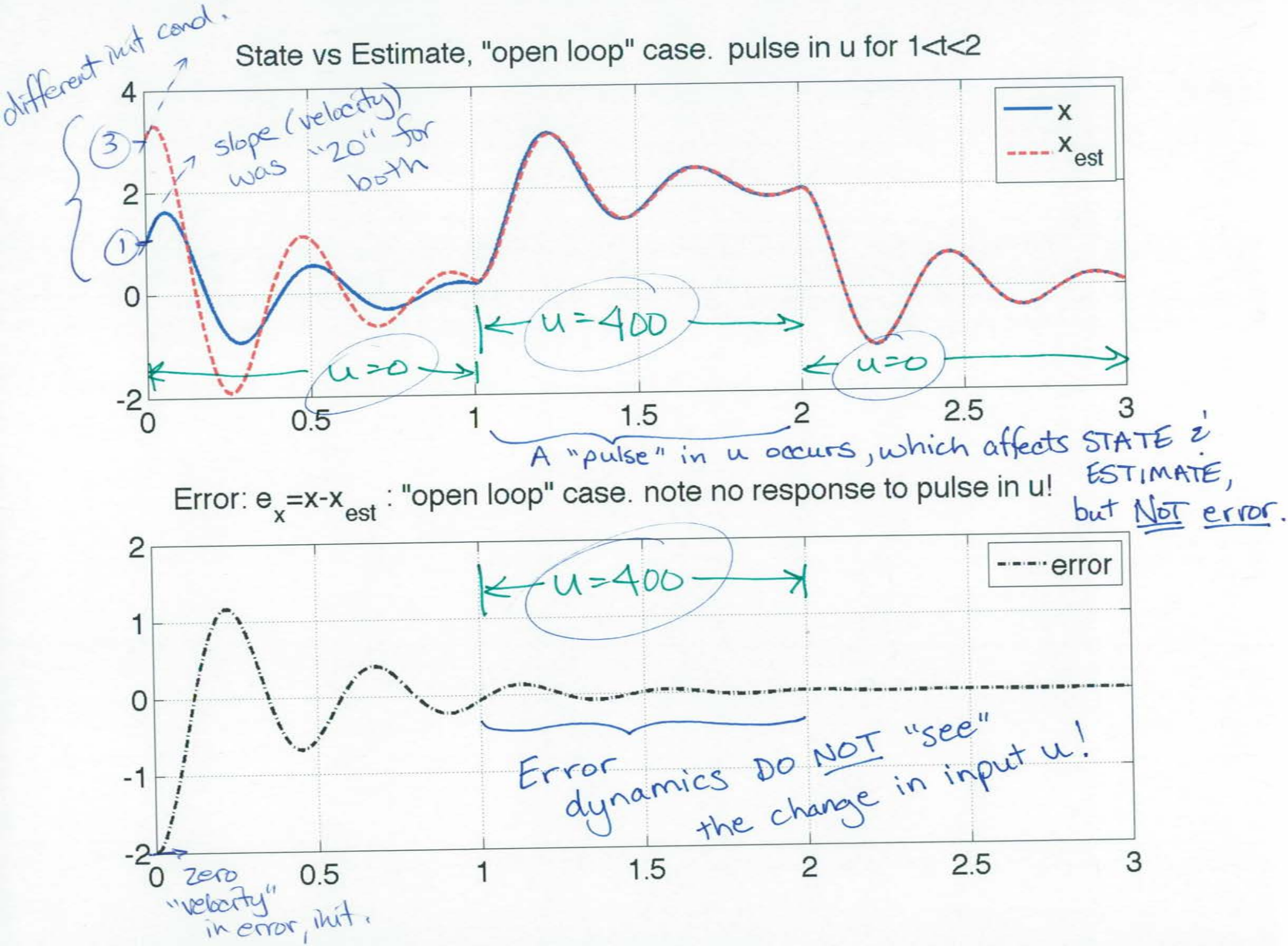
$$-\left(\bar{x}_{k+1} = A_d \bar{x}_k + B_d u_k\right)$$

$$(x_{k+1} - \bar{x}_{k+1}) = A_d(x_k - \bar{x}_k) + B_d(u_k - u_k)$$

$(x_{k+1} - \bar{x}_{k+1}) = e_x(k+1)$; $(x_k - \bar{x}_k) = e_x(k)$; $(u_k - u_k) \to 0$

"Error dynamics" → $\therefore$ $\boxed{e_x(k+1) = A_d e(k)}$

Same poles (dynamics) as open-loop system.


State vs Estimate, "open loop" case. pulse in u for 1<t<2
different init cond.
slope (velocity) was "20" for both
x
x_est
u=0
u=400
u=0
A "pulse" in u occurs, which affects STATE & ESTIMATE, but NOT error.
Error: e_x=x-x_est : "open loop" case. note no response to pulse in u!
error
u=400
Error dynamics DO NOT "see" the change in input u!
zero "velocity" in error, init.

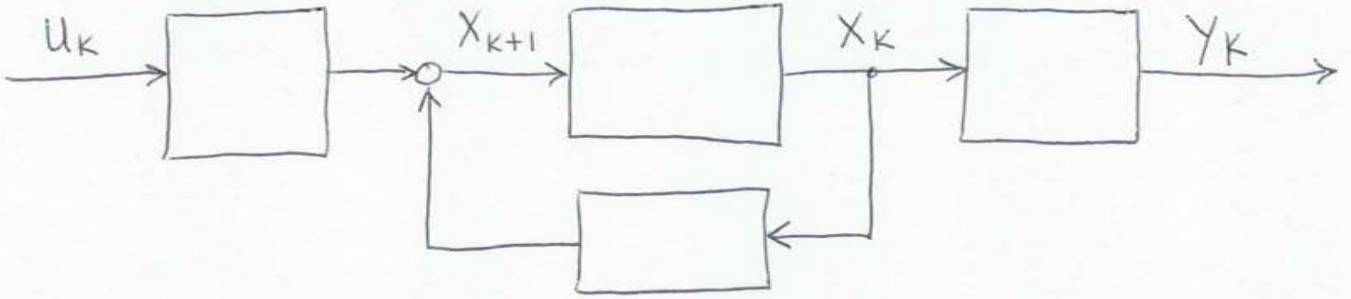
State Estimation – "feedback" approach

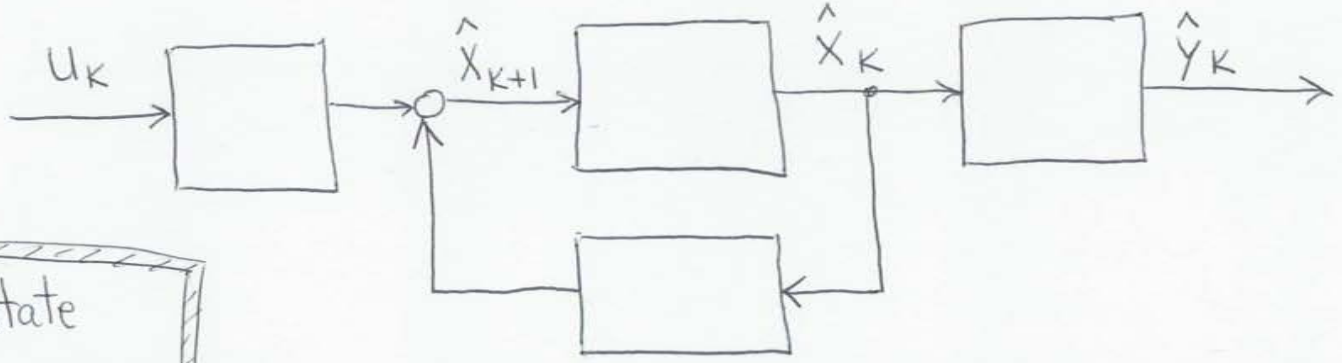
* NEXT CLASS *

Start with some "initial value" as an ESTIMATE of the states...
...use BOTH a MODEL ("Ax+Bu") and a MEASUREMENT ("y=Cx+Du").

Estimate of state WITH MEASUREMENT FEEDBACK : $\hat{x}(k+1)$

But how do we "feed back" MEASUREMENT info?





x: actual state

$\hat{x}$: estimated state

"x hat"