Today: **Reference Tracking / Integral Control** **Lecture 17**

ECE 147B
Digital Control

Reading: FPW 8.4.1, 8.5.1 <or> FV 9.3, 10.4.3, p.370
<AND> RS-14, RS-15

Topics:
- Reference Tracking
- Integral Control

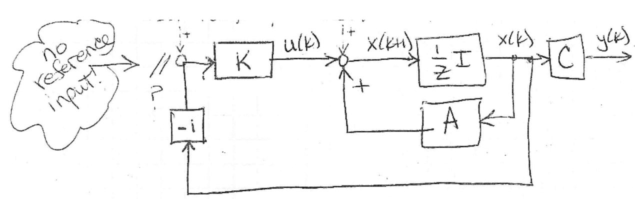
Next Class:

- Optimal Control and Estimation

FPW 9.3.5, 9.4.1 <or> FV 10.2-10.4
<AND> RS-16

Reference Tracking

So far, we have only considered state space control for a "regulator":



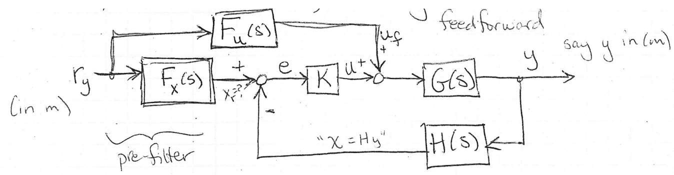
Questions: How can we

1) Compare x(k) (or an ESTIMATE of x(k)) with a desired REFERENCE, $x_r(k)$?

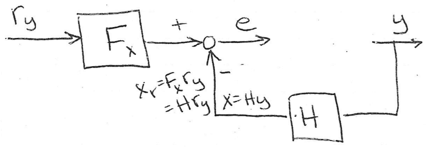
2) Generate a feedforward term, $u_f(k)$, to eliminate steady state error?

Reference Tracking

First, consider the familiar “SISO” system case:

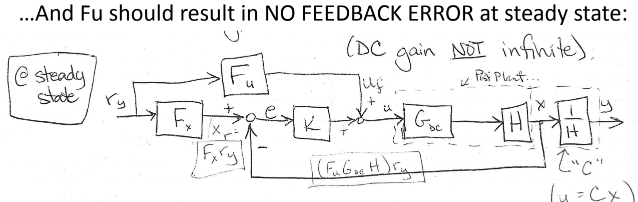


F_x should scale output “y” to match (in units) the feedback term:



Reference Tracking

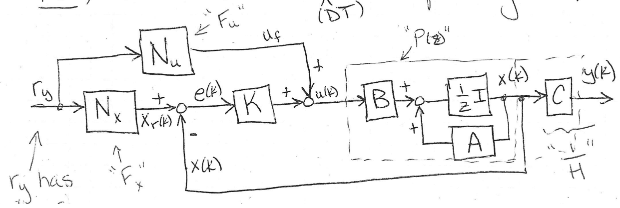
…And Fu should result in NO FEEDBACK ERROR at steady state:



- w/ $F_u = 0$, $e_{ss} \neq 0$.
- If we want $e_{ss} = 0$, $F_x r_y - (F_u G_{DC} H) r_y = 0$

MIMO equivalent:

Now, consider a MIMO (DT) state space system form:



Reference Tracking

Example: 4^{th}-order spring-mass system

Reference Tracking

Recap, for the case of zero steady state error, we require:

Integral Control for State Space Models

State space control, "u = -Kx", involves multiplication of state vectors, x, by a matrix of gains, K. Thus, we can capture a "PD" (proportional plus derivative) controller if there are states to represent position and velocity: u = -K_1*x - K_2*dx/dt

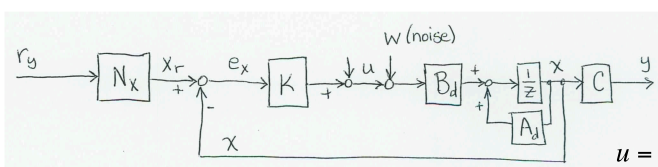
To include an "integral action", for example "PID" control, we need an (internal) state that is the INTEGRAL of another (actual) state of the dynamic system.

Answer: Augment the system with an additional state! Let us call this new state: x_I

<u>Continuous Time:</u> $x_I(k)$ <u>Discrete Time:</u> $x_I(T)$

Integral Control for State Space Models

Block Diagram:



$$u = -\begin{bmatrix} K_I & K \end{bmatrix}\begin{bmatrix} x_I \\ x \end{bmatrix} + KN_x r$$

Discrete-time (DT) state space equations:

$$z\begin{bmatrix} x_I \\ x \end{bmatrix} = \begin{bmatrix} & \\ & A \end{bmatrix}\begin{bmatrix} x_I \\ x \end{bmatrix} + \begin{bmatrix} \\ B \end{bmatrix}u - \begin{bmatrix} \\ \end{bmatrix}r$$

Continuous-time (CT) state space equations:

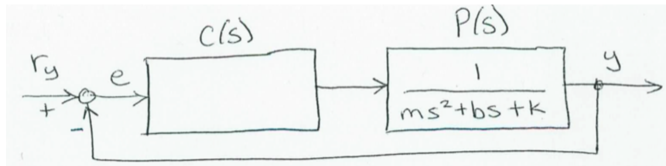
$$s\begin{bmatrix} x_I \\ x \end{bmatrix} = \begin{bmatrix} & \\ & A \end{bmatrix}\begin{bmatrix} x_I \\ x \end{bmatrix} + \begin{bmatrix} \\ B \end{bmatrix}u - \begin{bmatrix} \\ \end{bmatrix}r$$

Integral Control for State Space Models

Example: spring-mass-damper system (2nd order). Continuous time.
n states, l outputs: n+l total poles (since integral of each output is a NEW state now).

$$X = \begin{bmatrix} \dot{x} \\ x \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \end{bmatrix} \qquad N_x = \begin{bmatrix} \quad \end{bmatrix} \qquad A = \begin{bmatrix} -b/m & -k/m \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1/m \\ 0 \end{bmatrix}$$

$$P(s) = \left(\frac{1}{ms^2 + bs + k} \right)$$

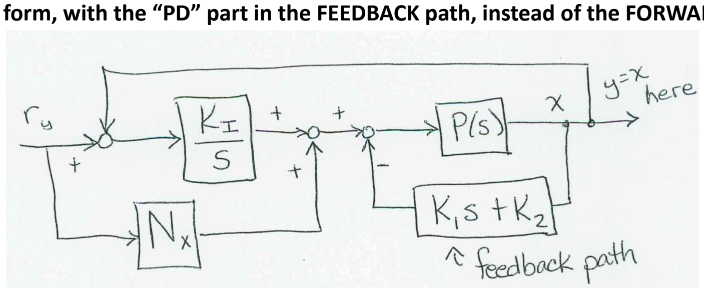


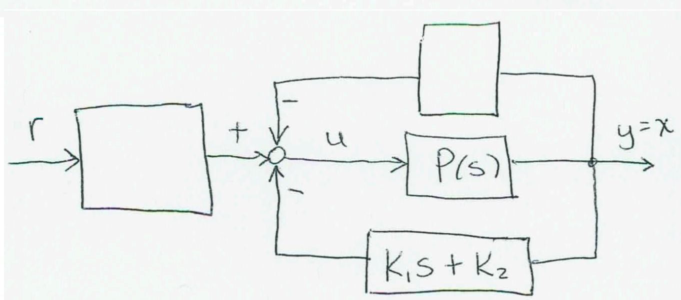
$C(s) =$

Closed-loop system?

Integral Control for State Space Models

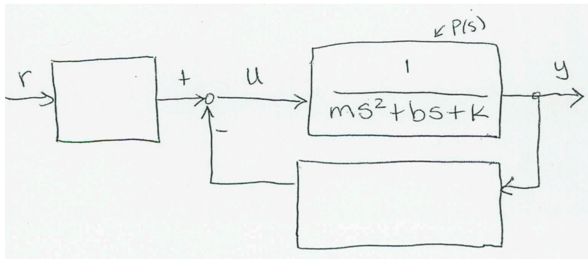
Alternate form, with the "PD" part in the FEEDBACK path, instead of the FORWARD path:





Integral Control for State Space Models

Alternate form, continued…



Closed-loop system?

Integral Control for State Space Models

Recall… n states, l outputs: (n+l) total poles.

Notes:

1. Putting control dynamics (PD part) in the feedback path can reduce overshoot. (i.e., Avoid some or all of the "zeros".)

2. Can place pole(s) to 'cancel out' the l zeros due to integral control, and then set the remaining n poles (via pole placement) as desired.