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:

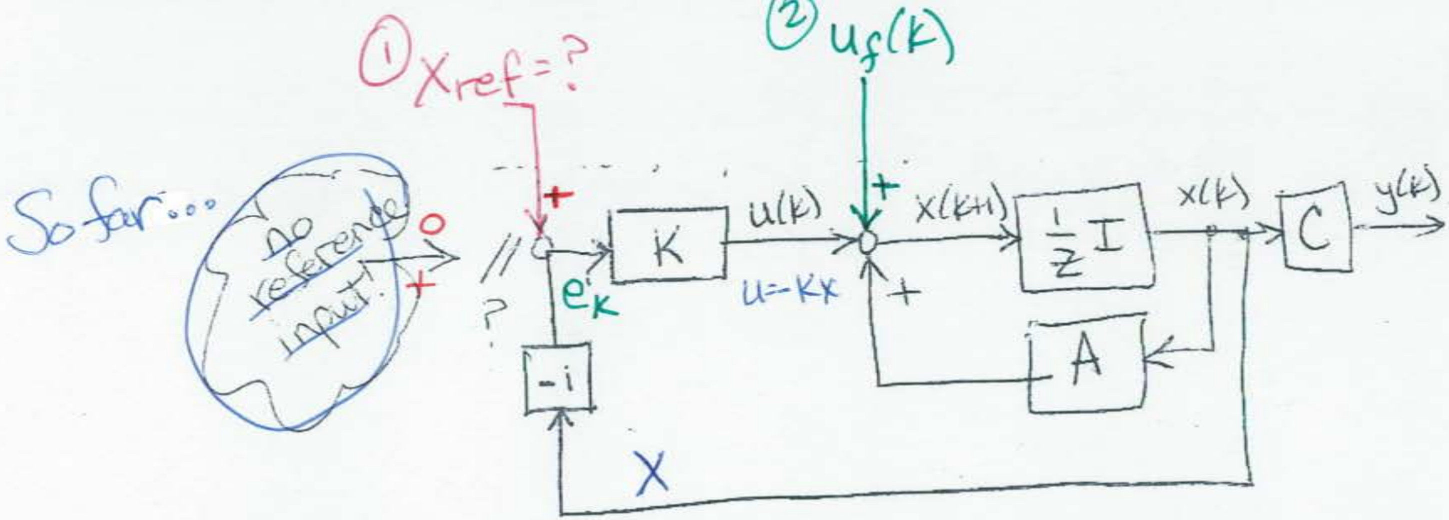
- Optimal Control and Estimation

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":

$\boxed{e_k \equiv x_{ref_k} - x_k}$



Questions: How can we

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

Given what we have is some $r_y(k)$ to match output $y(k)$.

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

Without feedforward, $u = -K(x - x_{ref})$, means $u=0$ when $(x - x_{ref}) = 0$... So if $x = x_{ref}$ requires some force (e.g., pushing a spring), we will have s.s. error.

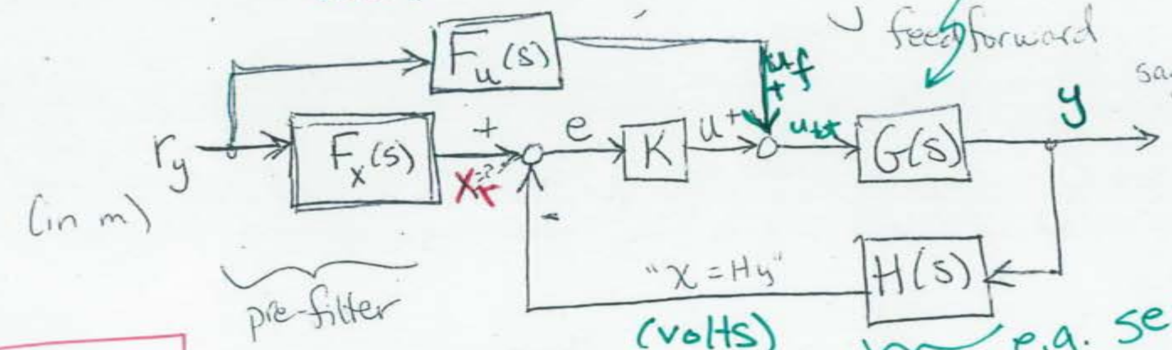
Reference Tracking

First, consider the familiar "SISO" system case:

Plant

feedforward

say y in (m)



$$\boxed{X_r = F_x r_y}$$

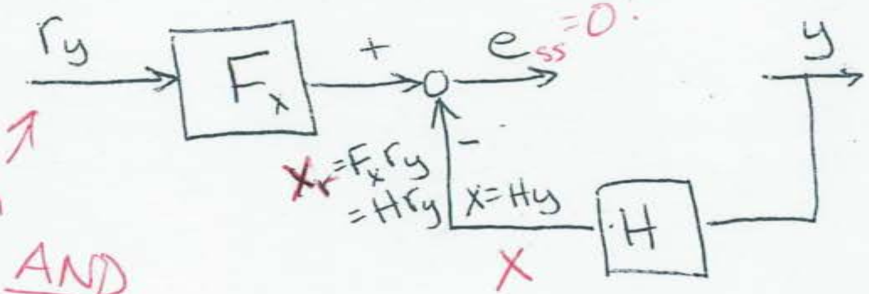
$$\boxed{u_f = F_u r_y}$$

e.g. sensor reads .05 meters for 1 volt

$$H \equiv \frac{1 \text{ volt}}{.05 \text{ m}} = 20 \, (\text{V/m})$$

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

[Note, for Type 1 (& higher) $G(s)$, $e_{ss} = 0$ w/out "feedforward".]



(match scaling AND dimensionality
(... later when we move to MIMO)

And we want $\boxed{y_{ss} = r_{y_{ss}}}$

$e_{ss} = 0$, desired

$$0 = F_x r_{y_{ss}} - H y_{ss}$$

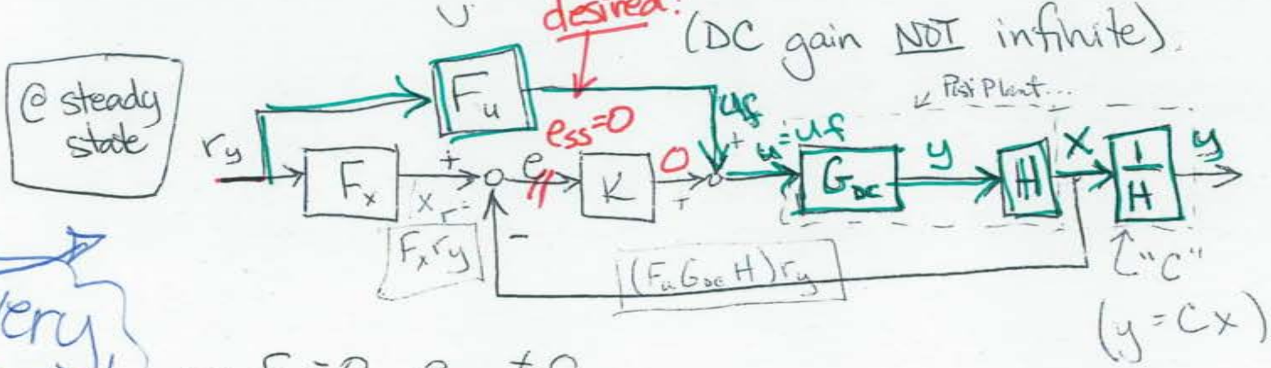
$$F_x r_{y_{ss}} = H y_{ss} \rightarrow \boxed{F_x = H}$$

Reference Tracking

G_{DC} = DC gain of $G(s)$ (Thinking about "steady state", ss.)

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

desired! (DC gain NOT infinite)



$F_x = H$, last page...

$$(F_x - F_u G_{DC} H) r_y = 0$$

$$(H - F_u G_{DC} H) r_y = 0$$

$$\therefore \boxed{F_u = \frac{1}{G_{DC}}}$$

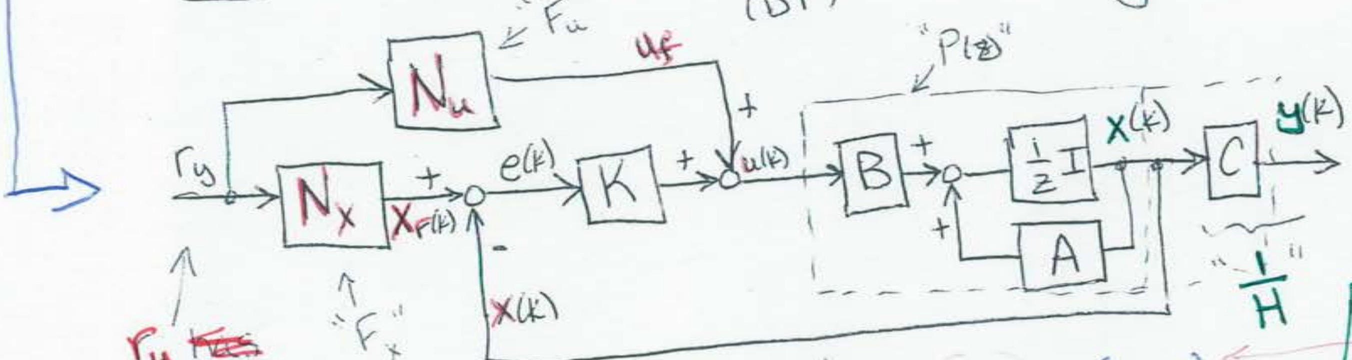
"invert" the plant (or at least its DC gain).

Very Similar!

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

MIMO equivalent:

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



r_y ~~Kes~~ (dimensionality & scaling) "matches" y

n states, x is $(n \times 1)$
m inputs, u is $(m \times 1)$
ℓ outputs, y is $(\ell \times 1)$

*Assume some r_y (to match) y is known (given).

$$\boxed{\underset{(n\times1)}{x_r} = \underset{(n\times\ell)}{N_x}\,\underset{(\ell\times1)}{r_y}}$$ to match y

to match x

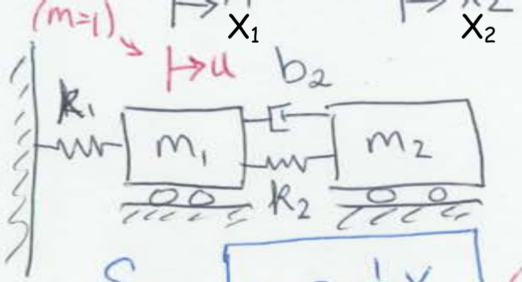
$$\boxed{\underset{(m\times1)}{u_f} = \underset{(m\times\ell)}{N_u}\,\underset{(\ell\times1)}{r_y}}$$

Reference Tracking

(n^{th} order, $\frac{n}{2}$ masses)

$m=1$ input, u

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



Say $\boxed{y = \frac{1}{2}X_1}$ $(\ell=1)$ output

& $r_y =$ unit step

$$X_{(n\times 1)} = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix}, \quad A_{(n\times n)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_1-k_2}{m_1}, & \frac{-b_2}{m_1}, & \frac{k_2}{m_1}, & \frac{b_2}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2}, & \frac{b_2}{m_2}, & \frac{-k_2}{m_2}, & \frac{-b_2}{m_2} \end{bmatrix}$$

$$m_1\ddot{x}_1 + b_2(\dot{x}_1 - \dot{x}_2) + k_2(x_1 - x_2) + k_1 x_1 = u$$

$$\ddot{x}_1 = \left(\frac{1}{m_1}\right) * [\text{stuff}]$$

$$B_{(n\times m)} = \begin{bmatrix} 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix}, \quad C_{(\ell\times n)} = \left[\frac{1}{2}, 0, 0, 0\right]$$

$\uparrow y = \frac{1}{2}x_1$

$P(s)$ [This is a SISO system...]

$$\frac{Y(s)}{U(s)} = \frac{K_{top}}{s^4 + \dots \dots + k_1}$$

$(K_{top} = \frac{1}{2})$

$$\boxed{G_{DC} = \frac{K_{top}}{k_1}}$$

No "free integrator"! Type 0. Needs feedforward to achieve zero steady-state error.

Recall, $F_u = \frac{1}{G_{DC}}$; here, $\boxed{N_u = \frac{1}{G_{DC}} = \frac{1}{\frac{1}{2k_1}} = 2k_1}$

For a spring $\boxed{F = -k\Delta X}$, So this makes sense. ☺

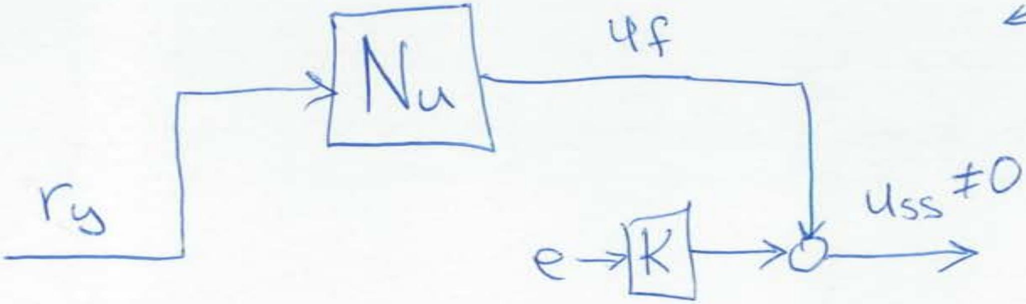
Also, $y_{ss} = 1 = Cx_{ss}$ means...

$(y_{ss} = \frac{1}{2}x_{1ss})$, $\boxed{x_{1,ss} = 2}$ & $\boxed{x_{2,ss} = x_{1,ss}}$

$$\therefore N_x = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$ $\leftarrow$ Both ss positions are 2. Vel's are 0.

Reference Tracking

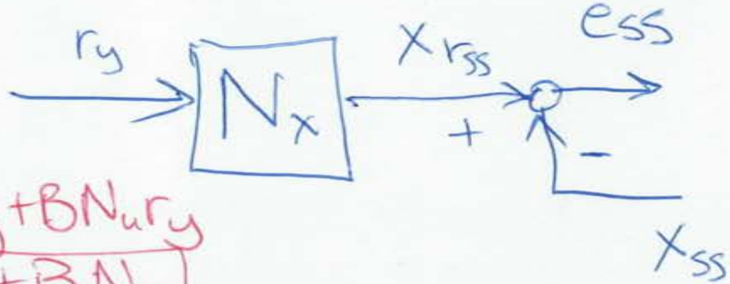
Recap, for the case of zero steady state error, we require:



← feedforward

$$\boxed{u_{ss} = N_u r_y}$$

Matching scaling & dimensionality for x_{ss} (vs. y_{ss}) requires a filter on r_y:
(pre-filter)



@ ss

$$0 = (A-I)N_x r_y + B N_u r_y$$

$$\boxed{0 = (A-I)N_x + B N_u}$$

$y_{ss} = r_y$, $y_{ss} = C x_{ss} = C N_x r_y$

$\therefore \boxed{r_y = C N_x r_y}$

$C N_x = I$

$$\begin{bmatrix} (A-I) & B \\ C & 0 \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

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

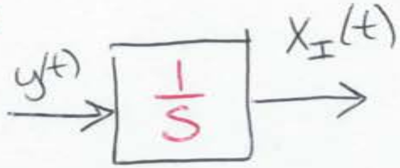
S-domain

Continuous Time: $x_I(\cancel{k})$ $X_I(t)$

$$X_I(t) = \int_0^T y(t)$$

$$X_I = \frac{1}{s} y$$

$$\frac{X_I}{y} = \frac{1}{s}$$



Discrete Time: $x_I(\cancel{T})$ $X_I(k)$

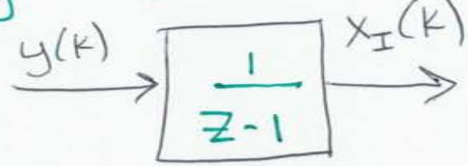
z-domain

$$X_I(k+1) = X_I(k) + y(k)$$

$$z \cdot X_I = X_I + y$$

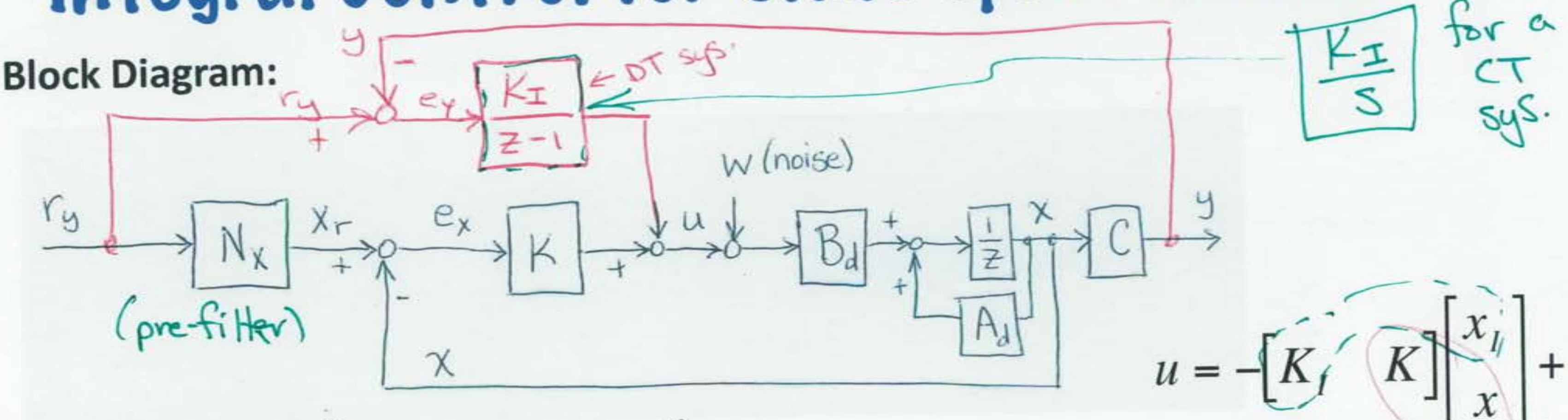
$$(z-1) \cdot X_I = y$$

$$\therefore \frac{X_I}{y} = \frac{1}{z-1}$$



Integral Control for State Space Models

Block Diagram:



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

"$u = -K_I x_I - Kx$" $+ K N_x r_y$

Discrete-time (DT) state space equations:

$$z\begin{bmatrix} x_I \\ x \end{bmatrix} = \begin{bmatrix} I & C \\ 0 & A \end{bmatrix} \begin{bmatrix} x_I \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} u - \begin{bmatrix} I \\ 0 \end{bmatrix} r$$

Was: $zX = AX + BU \ldots$ →

$y = Cx$

b/c we operate on "e_y"

for CT, this term (these terms) are 0.

Continuous-time (CT) state space equations:

$$s\begin{bmatrix} x_I \\ x \end{bmatrix} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix} \begin{bmatrix} x_I \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} u - \begin{bmatrix} I \\ 0 \end{bmatrix} r$$

$\frac{1}{s}$ vs $\frac{1}{z-1}$

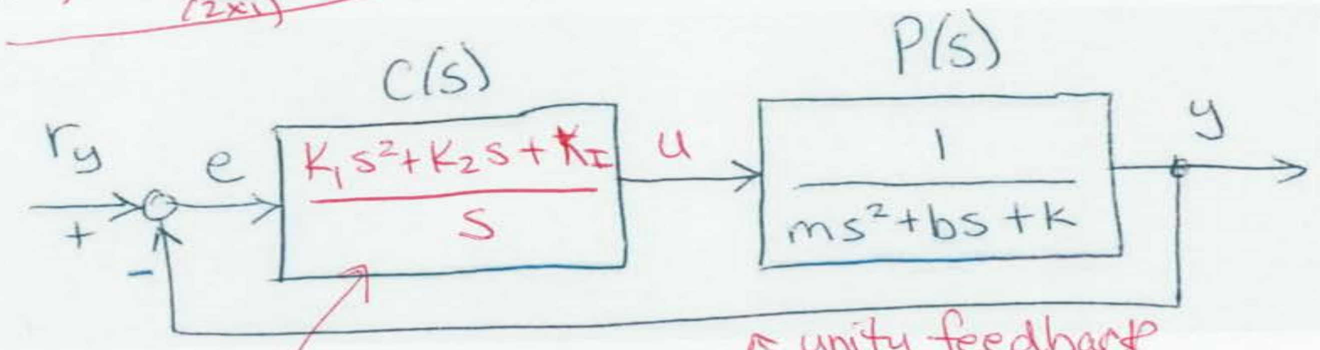
Integral Control for State Space Models

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).

$$X = \begin{bmatrix} \dot{x} \\ x \end{bmatrix} \qquad \underbrace{C = \begin{bmatrix} 0 & 1 \end{bmatrix}}_{y = CX = x \text{ (position)}} \qquad \underbrace{N_x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{(2\times 1)} \qquad A = \begin{bmatrix} -b/m & -k/m \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1/m \\ 0 \end{bmatrix}$$

control canonical form



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

$$C(s) = \frac{K_I}{s} + K_1 s + K_2$$

$$= \frac{K_1 s^2 + K_2 s + K_I}{s}$$

Closed-loop system?

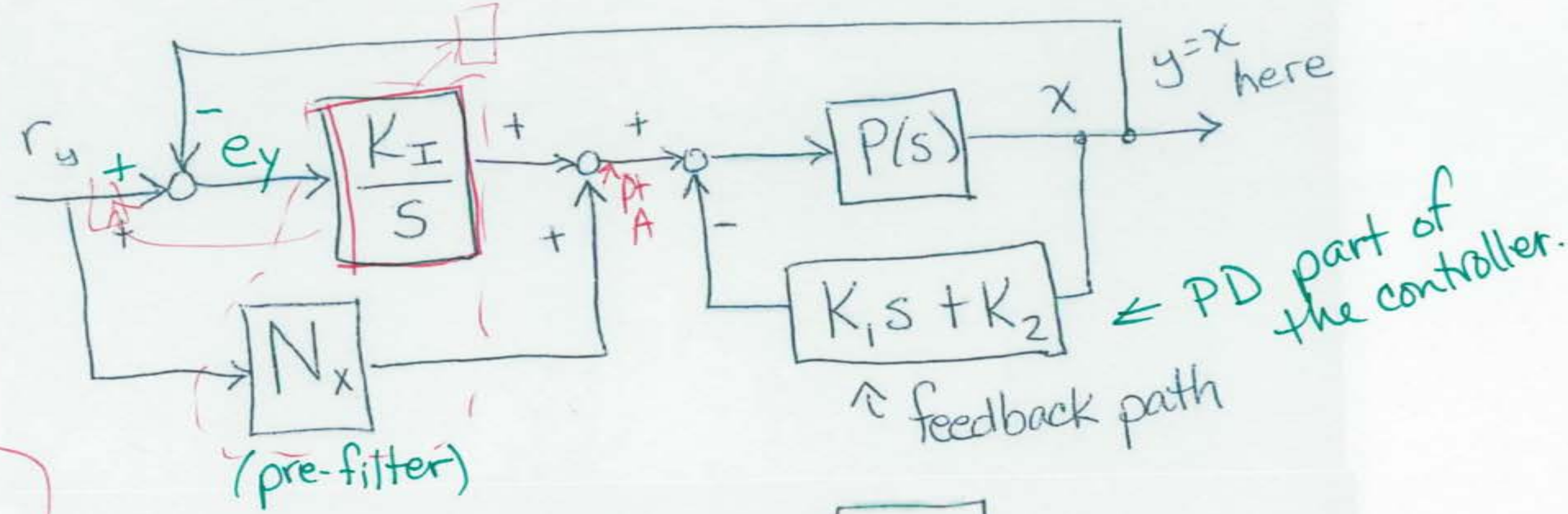
$$\frac{K_1 s^2 + K_2 s + K_I}{ms^3 + (b + K_1)s^2 + (k + K_2)s + K_I}$$

@ ss, response to a unit step?

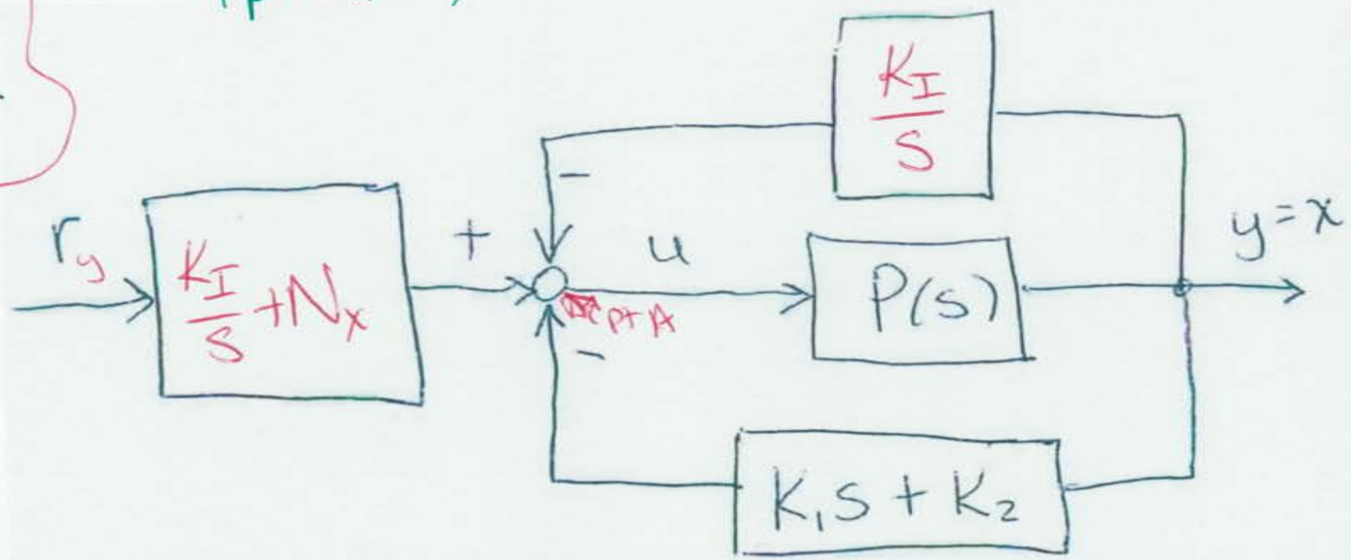
$$\frac{K_I}{K_I} = 1 \quad \left(\text{no error!}\right)$$

Integral Control for State Space Models

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

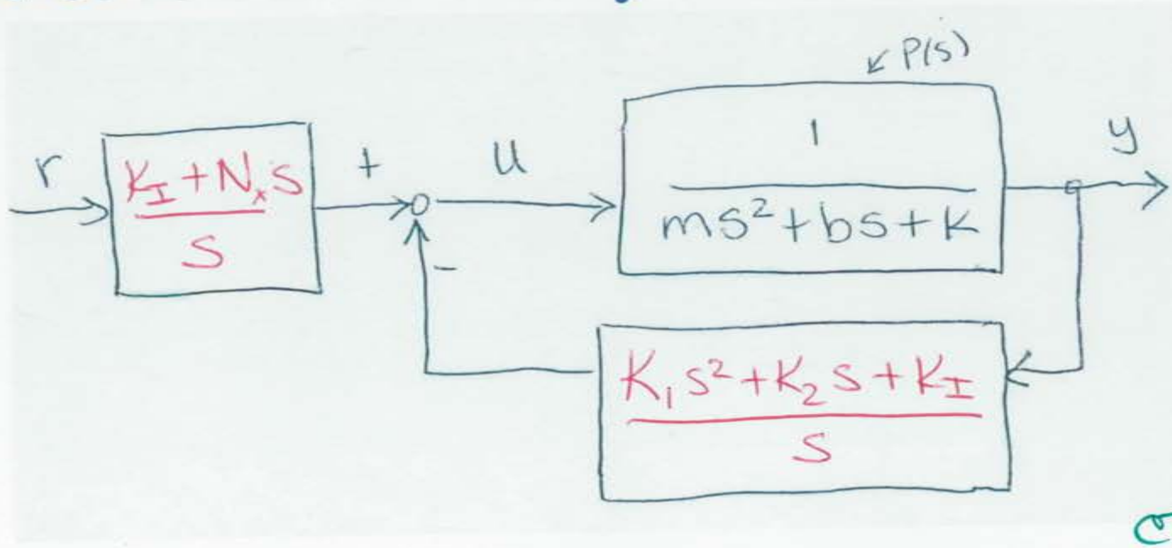


Block Diagram Algebra...



Integral Control for State Space Models

Alternate form, continued…



Closed-loop system?

For DT sys, zero is @ $\boxed{z = 1 - \frac{K_I}{N_x}}$

CT ~~pole~~ zero

zero @ $s = \frac{-K_I}{N_x}$

Different zeros.

$$\frac{(N_x s + K_I)}{\underbrace{ms^3 + (b + K_1)s^2 + (k + K_2)s + K_I}_{\text{same char. eqn}}}$$

same char. eqn

same poles

Integral Control for State Space Models

Recall… n states, ℓ l outputs: $\overset{\ell}{(n+l)}$ total poles. ← ℓ "extra" x_I states (internally).

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 $\overset{\ell}{\text{l}}$ zeros due to integral control, and then set the remaining n poles (via pole placement) as desired.

ζ & ω_n, as desired, for dominant poles.