LECTURE 33

DYNAMIC COMPENSATION

LEAD COMPENSATION:

- APPROXIMATES PD CONTROL
- SPEEDS UP RESPONSE
- LOWERS RISE TIME
- REDUCES TRANSIENT OVERSHOOT

LAG COMPENSATION:

APPROXIMATES PI CONTROL

- IMPROVE STEADY-STATE ACCURACY

NOTCH COMPENSATION:

ACHIEVE STABILITY

$$D(s) = K\frac{s+z}{s+p} \quad \begin{cases} z < p \Rightarrow \text{LEAD} \\ z > p \Rightarrow \text{LAG} \end{cases}$$

(ADAPTED FROM SATISH NAIR)

5.4 DESIGN USING DYNAMIC COMPENSATION

What are deficiencies of a proportional-only controller?

Assume that a customer comes to you and wants you the smart MU graduate to improve her system characteristics so that it has a settling time of 1.2 sec, and rise time of 0.26 sec.

Your first step is to determine a model for the system. Assume you did open loop tests and determined the OLTF to be

G(s) = 1/[s(s+1)]. What do you to next?

5.4 DESIGN USING DYNAMIC COMPENSATION

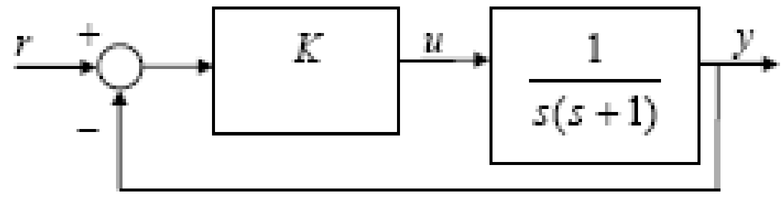
What are deficiencies of a proportional-only controller?

Assume that a customer comes to you and wants you the smart MU graduate to improve her system characteristics so that it has a settling time of 1.2 sec, and rise time of 0.26 sec.

Your first step is to determine a model for the system. Assume you did open loop tests and determined the OLTF to be

G(s) = 1/[s(s+1)]. What do you to next?

Add the LOWEST cost controller – proportional controller, K, and see if it can do the job. What do you think?



5.4 DESIGN USING DYNAMIC COMPENSATION

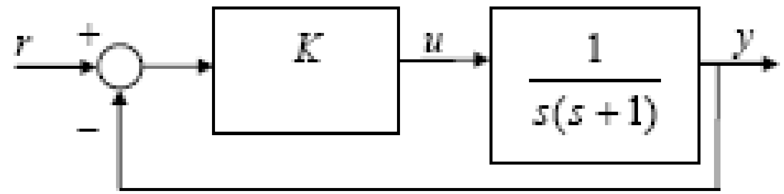
What are deficiencies of a proportional-only controller?

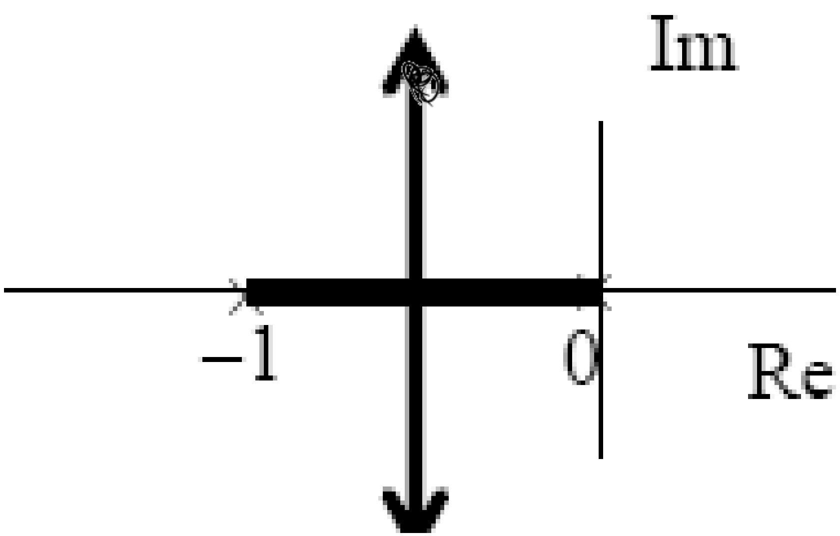
Assume that a customer comes to you and wants you the smart MU graduate to improve her system characteristics so that it has a settling time of 1.2 sec, and rise time of 0.26 sec.

Your first step is to determine a model for the system. Assume you did open loop tests and determined the OLTF to be

G(s) = 1/[s(s+1)]. What do you to next?

Add the LOWEST cost controller – proportional controller, K, and see if it can do the job. What do you think?





Does it give you the design you want?

5.4 DESIGN USING DYNAMIC COMPENSATION

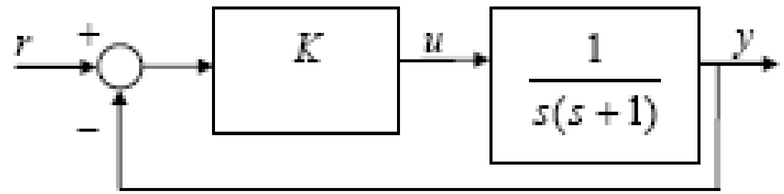
What are deficiencies of a proportional-only controller?

Assume that a customer comes to you and wants you the smart MU graduate to improve her system characteristics so that it has a settling time of 1.2 sec, and rise time of 0.26 sec.

Your first step is to determine a model for the system. Assume you did open loop tests and determined the OLTF to be

G(s) = 1/[s(s+1)]. What do you to next?

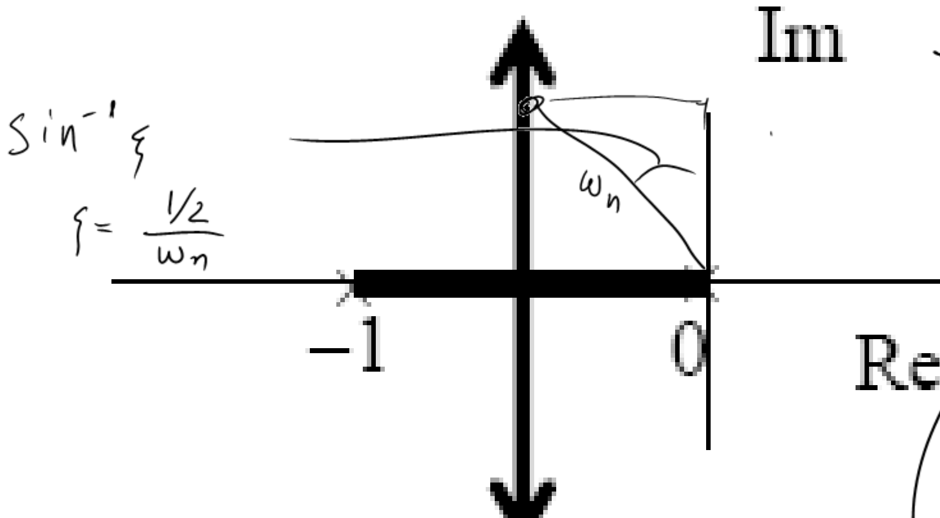
Add the LOWEST cost controller – proportional controller, K, and see if it can do the job. What do you think?



Recall $T(s) = \frac{DG}{1+DG}$

$T(s) = \frac{K}{s^2 + s + K}$

$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n}$



$t_r = \frac{1.8}{\omega_n} = 0.26s$

$t_s = \frac{4.6}{\zeta\omega_n} = 1.2s$

$K = \omega_n^2 \quad \zeta = \frac{1}{2\omega_n}$

$\omega_n = 1.8/0.26 = 3.846$

$\zeta = 0.9967$

Does it give you the design you want?

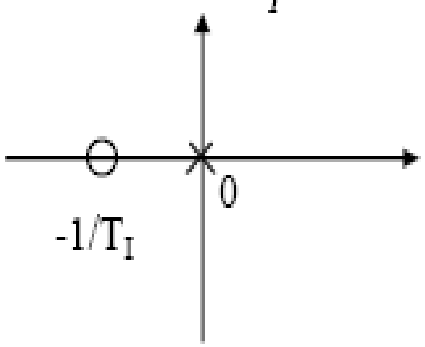
5.4 DESIGN USING DYNAMIC COMPENSATION

PI vs. LAG (APPROXIMATE PI) and PD vs. LEAD (APPOXIMATE PD)

- **PI compensation**

The primary reason for integral control is to reduce or eliminate constant steady-state error

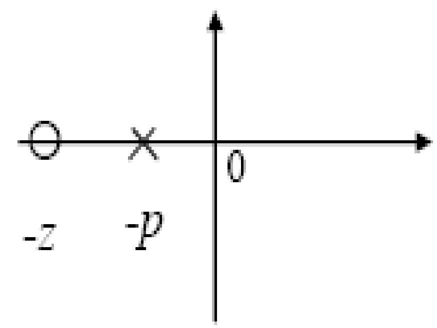
$$D(s) = K\frac{T_I s + 1}{T_I s}$$



PI feedback control cannot be realized using passive elements.

- **Lag compensation (approximate PI)**

$$D(s) = K\frac{s+z}{s+p} (z > p)$$

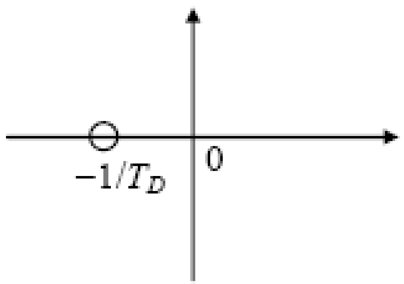


Lead compensation can be realized using passive elements.

- **PD compensation**

Derivative feedback is used in conjunction with proportional feedback to improve transient characteristics.

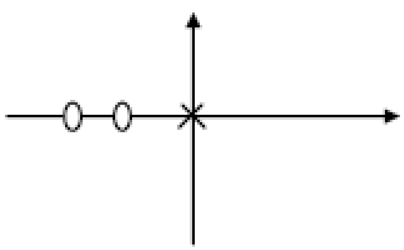
$$D(s) = K(1 + T_D s)$$



PD control amplifies noise.

- **PID compensation**

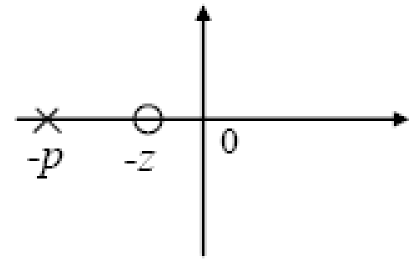
$$D(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right)$$



PID control eliminates steady state and transient errors.

- **Lead compensation (approximate PD)**

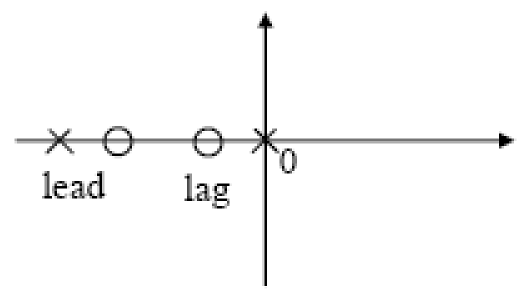
$$D(s) = K\frac{s+z}{s+p} \quad (z < p)$$



Lead compensation does not amplify noise due to the presence of the additional pole.

- **Lead- Lag compensation**

$$D(s) = K\left(\frac{s+z_1}{s+p_1}\right)\left(\frac{s+z_2}{s+p_2}\right)$$

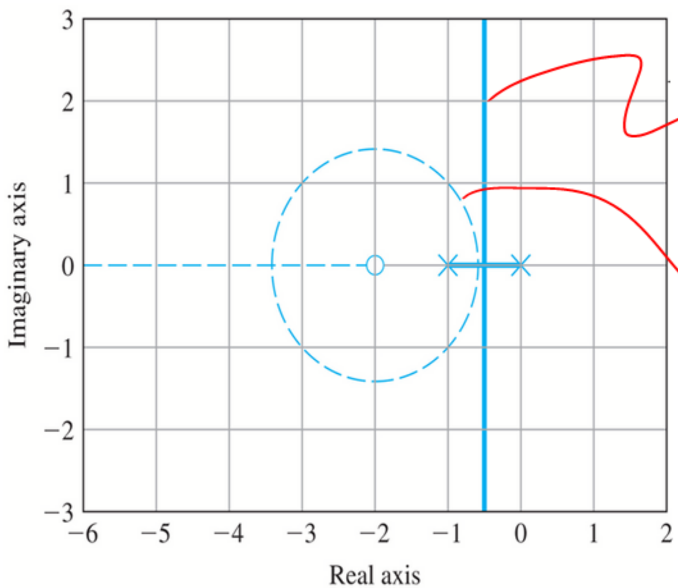


Lead-lag compensation approximates PID controller.

Ex

$D(s) = K(s+z) \qquad G(s) = \frac{1}{s(s+p)} \qquad T(s) = \frac{DG}{1+DG}$

$$T(s) = \frac{K(s+z)/s(s+p)}{1 + K(s+z)/s(s+p)} = \frac{K(s+z)}{s^2 + sp + K(s+z)} = \frac{K(s+z)}{s^2 + (K+p)s + Kz}$$



$G = \frac{1}{s(s+p)}$

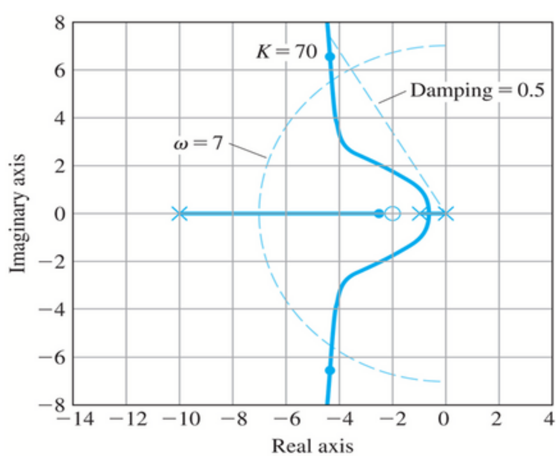
$D = K$

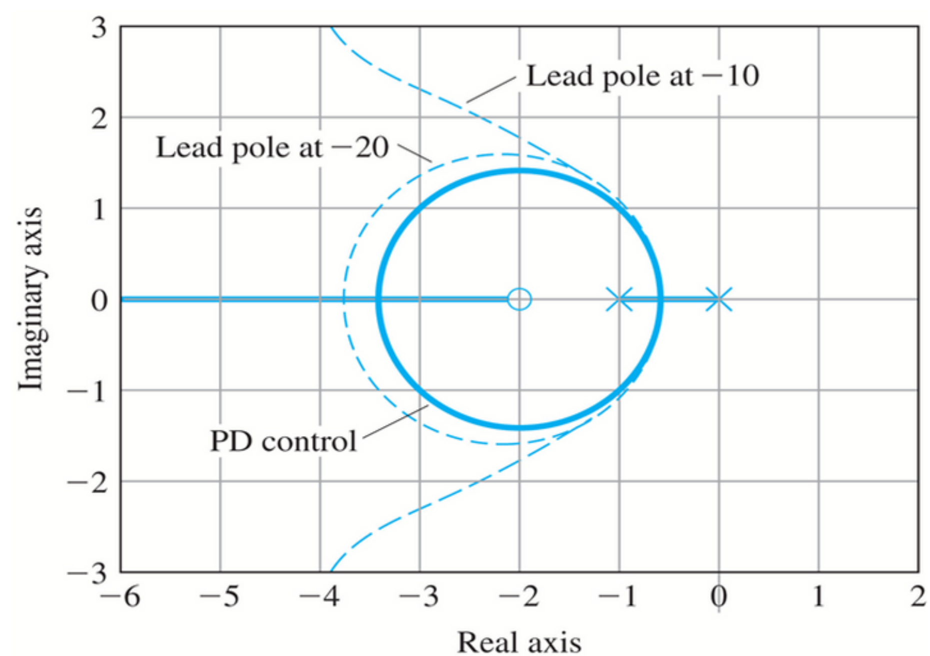
$G = \frac{1}{s(s+p)} \quad z = 2$

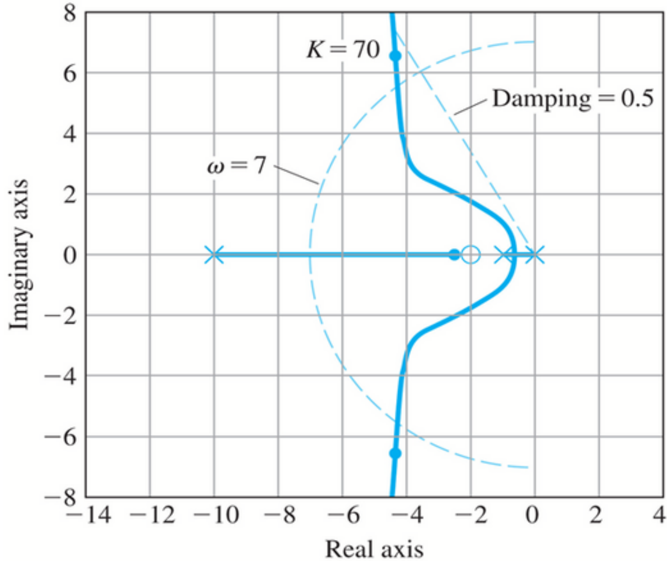
$D = K(s+z) \quad p =$

Keep z c^te (2)

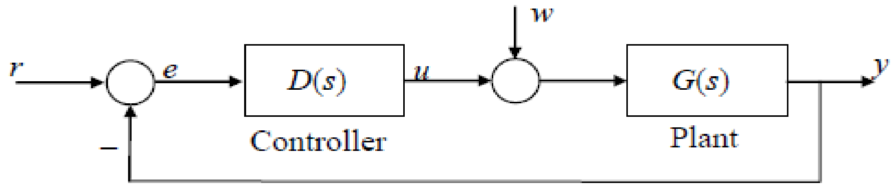
K = Varies






8
6
4
2
0
−2
−4
−6
−8
−14
−12
−10
−8
−6
−4
−2
0
2
4
K = 70
Damping = 0.5
ω = 7
Imaginary axis
Real axis

UNITY FEEDBACK SYSTEMS REVISITED



RECALL -- for the Unity Feedback case, if *D(s)G(s)* has n free *s* in the denominator, then the system is of type *n*

e.g., D(s)G(s) = 1/[s(s+1)] is a type 1 system and will have a zero ss error for a step input and a constant steady state error for a ramp input.

<u>How do we calculate ss error for such systems</u>?

Note that for UFB systems only, the error constants are as follows:

position error constant $K_p = \lim_{s \to 0} D(s)G(s)$;

velocity error constant $K_v = \lim_{s \to 0} sD(s)G(s)$; **and**

acceleration error constant $K_a = \lim_{s \to 0} s^2 D(s)G(s)$ [*these terms are used in industry*]

TYPES? Type 0 – the system has a constant error to a step; Type 1 – system has a constant error to a ramp input,….and so on.

So, for type 0 systems, ss error to a step input $r(t) = 1(t)$ **is** $e_{ss} = \frac{1}{1+K_p}$

for type 1 systems, ss error to a ramp input $r(t) = t \cdot 1(t)$ **is** $e_{ss} = \frac{1}{K_v}$

for type 2 systems, ss error to parabolic input $r(t) = t^2 \cdot 1(t)$ **is** $e_{ss} = \frac{1}{K_a}$

<u>Find the ss error to the appropriate inputs for the following systems</u>:

H(s) = 1/(s+5); H(s) = 1/[s(s+7)]; H(s) = (s+4)/[s(s+10)]; H(s) = (s+2)/[s (s+3)(s+12)]