8/1 ME132 Discussion!

Larry

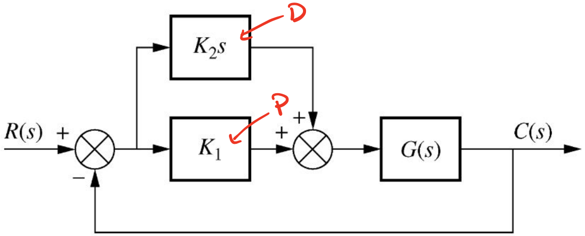
Compensators

IDEAL: use pure $\frac{d}{dt}$ and $\int$

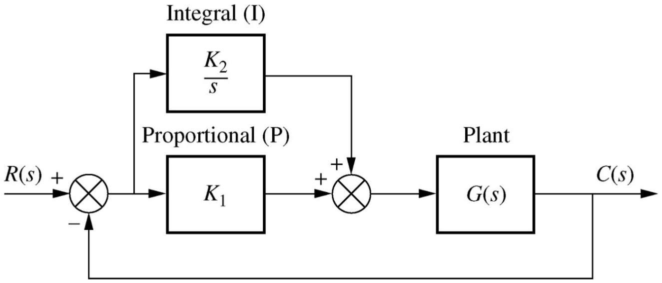
NON-IDEAL: approximate $\frac{d}{dt}$ and $\int$

Compensator: a subsystem represented as a TF inserted into the feedforward/feedback path to improve transient response or steady-state error

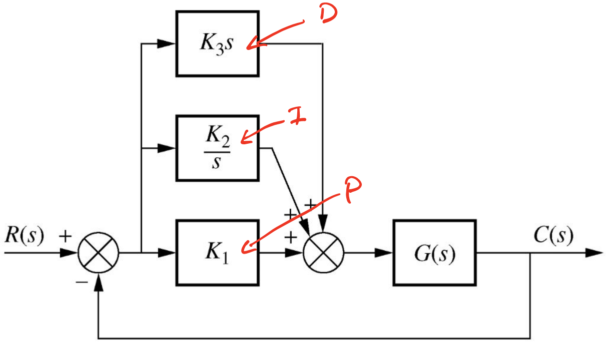
- changes the OL poles and zeros, thereby creating a new RL that goes through the desired CL pole locations.



PD

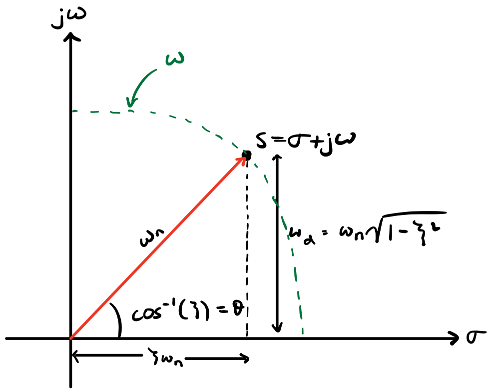


PI



PID

Performance Characteristics (RL)



1. Percent Overshoot (%PO): $\%PO \approx e^{-\pi\zeta/\sqrt{1-\zeta^2}} \times 100\%$

2. Settling Time (T_s): $T_s \approx \dfrac{4}{\zeta\omega_n}$

3. Rise Time (T_r): $T_r \; (10\text{–}90\%) \; \approx \; \dfrac{1}{\omega_n}(1.768\zeta^3 - 0.417\zeta^2 + 1.039\zeta + 1)$

4. Damping Frequency (ω_d): $\omega_d = \omega_n\sqrt{1-\zeta^2}$

constant ζ (θ-rays) $\Rightarrow$ $\zeta = \cos\theta$

constant ω_n (circles) $\Rightarrow$ $r = \omega_n$

5. **Dominant Poles:** high-order systems have many CL poles; we arrange for 2 **dominant** poles at $s_{1,2} = -\zeta\omega_n \pm j\,\omega_n\sqrt{1-\zeta^2} = \sigma \pm j\omega$ (at least 5 times farther in the LHP), it can be approximated as 2nd order!)

PI Controller (ideal integral, active)

$$G_c(s) = K_1 + K_2 s = \frac{K_1\left(s + \frac{K_2}{K_1}\right)}{s}$$

OL pole at origin

$$K\frac{s+z_c}{s}$$

- Eliminates the steady-state error to 0 by increasing the system type using PURE integration! Implement by cascading controller and the plant!

ensures CL poles roughly don't move

- Compensator zero $-z_c$ is small and negative.

- Active circuits are required to implement (op-amp)

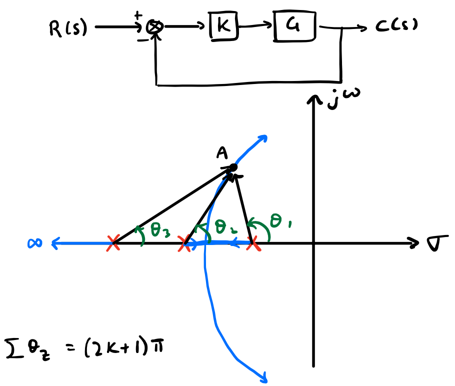
Method:

① Original transient response determined by location of the original OL poles

② Add a pole at origin
↳ RL no longer goes through location of prev. poles

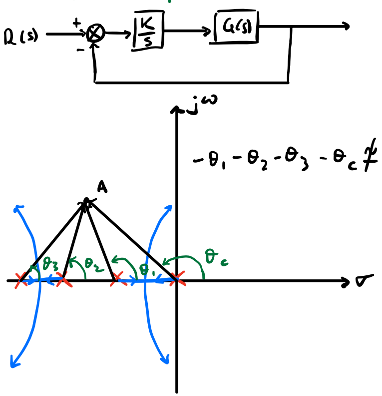
③ Add a zero close to pole at the origin
↳ zero location can be tuned to cause RL to go through location of previous poles

uncompensated



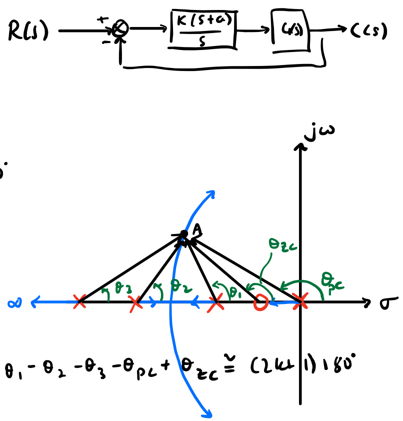
$$\sum\theta_p - \sum\theta_z = (2k+1)\pi$$

compensator pole added



$$-\theta_1 - \theta_2 - \theta_3 - \theta_c \neq (2k+1)180°$$

PI controller



$$-\theta_1 - \theta_2 - \theta_3 - \theta_{pc} + \theta_{zc} \approx (2k+1)180°$$

Lag Compensator (passive)

$$K\frac{s+z_c}{s+p_c}$$

- Improves steady-state error but does not drive it to 0 by APPROXIMATING integration by adding a pole near the origin. Implement by cascading controller and the plant!

- Poles at $-p_c$ is small negative; zeros at $-z_c$ is close to, and to the left of, the pole at $-p_c$.

- Active circuits are not required to implement; passive is fine.

Static error constant for uncompensated system

let $G_c(s) = K$ and $G(s) = \Pi(s+z_i) / \Pi(s+p_i)$

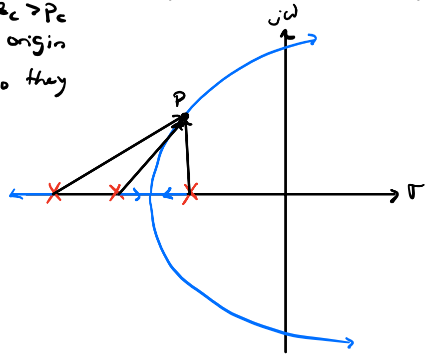
$$K_p = \lim_{s\to 0} G(s) = \frac{K z_1 z_2 \cdots}{p_1 p_2 \cdots}$$

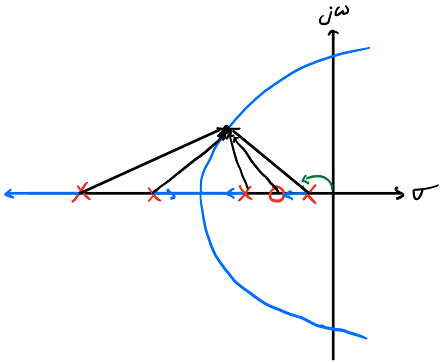
add pole and zero

$$K_p = \frac{(K z_1 z_2 \cdots)}{(p_1 p_2 \cdots)} \cdot \frac{z_c}{p_c}$$

To improve ss-error const., select $z_c > p_c$ by some factor and a p_c close to origin

↳ we still want z_c, p_c to be close so they net 0 angular contribution





PI Controller Example (Check Athul's Notes for Algebraic Version)

32. Design a PI controller to drive the step-response error to zero for the unity feedback system shown in Figure P9.1, where

$$G(s) = \frac{K}{(s+1)(s+3)(s+10)}$$

The system operates with a damping ratio of 0.5.

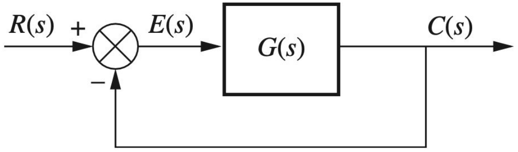
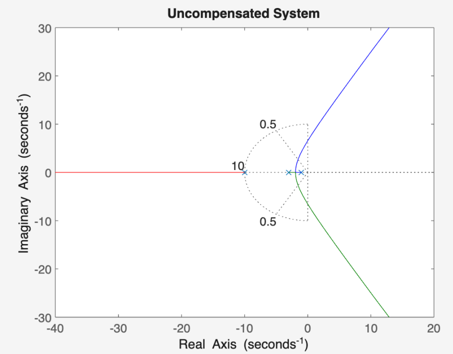


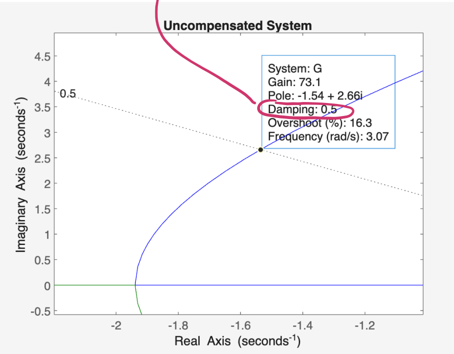
FIGURE P9.1

```
% PI CONTROLLER EXAMPLE

% Uncompensated
G = zpk([], [-1, -3, -10], 1)
rlocus(G)
title("Uncompensated System")
sgrid(0.5, 10)

% Compensated
G = zpk([-0.1], [-1, -3, -10, 0], 1)
rlocus(G)
title("Compensated System")
sgrid(0.5, 10)
```

① PLOT ROOT LOCUS

since we wanted $\zeta = 0.5$





② PERFORMANCE RELATION

$$\%OS = \exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)\cdot 100\%$$

$$= \exp\left(\frac{-(0.5)\pi}{\sqrt{1-0.5^2}}\right)\cdot 100\% = \underline{16.3\%}$$

$$T_s = \frac{4}{\zeta\omega_n} = \frac{4}{1.54} = 2.6$$

remember this is just the distance to the point on the Re axis

$$K_{p,\text{original}} = \lim_{s\to 0} G(s) = \lim_{s\to 0}\frac{K}{(s+1)(s+3)(s+10)}$$

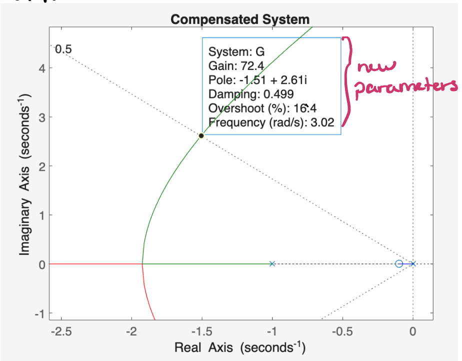
$$= \frac{K}{30} = \frac{73.1}{30} = 2.44$$

③ COMPENSATOR

↳ add pole at origin

↳ add small −ve zero i.e. −0.1

↳ Plot RL!



new parameters

$\%OS = 16.3\%$ (SAME!)

$$T_s = \frac{4}{\zeta\omega_n} = \frac{4}{1.51} = 2.64$$

(slower!)

$$K_{p,\text{new}} = \lim_{s\to 0} G_c(s)$$

$$= \lim_{s\to 0}\frac{K(s+0.1)}{s(s+1)(s+3)(s+10)}$$

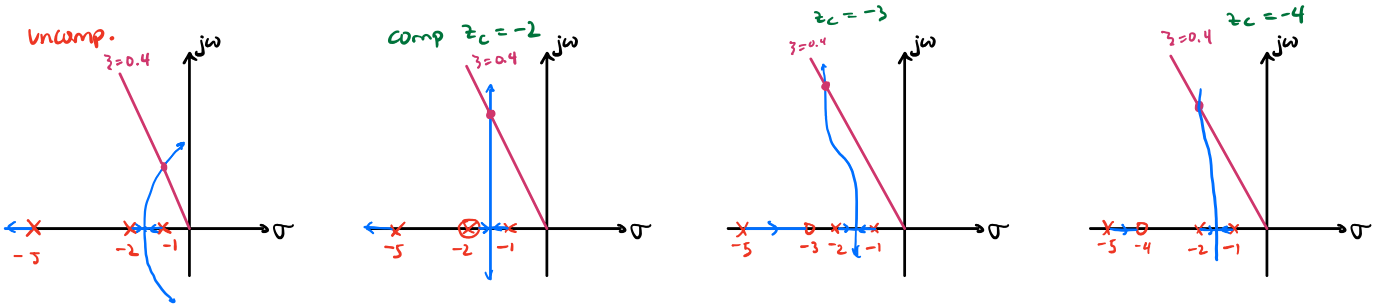
$$= \infty$$

$\Rightarrow$ if $K_p = \infty \Rightarrow$ error $= 0$ since type 1 system!

PD Controller (ideal derivative, active)

$$K(s+z_c)$$

- Improves the transient response using PURE differentiation.
- Zero at $-z_c$ is selected to put design point on root locus.



- Active circuits are required to implement. Can cause noise and saturation; implement with rate feedback or with a pole (lead)

Method:

① uncompensated system transient response is unacceptable

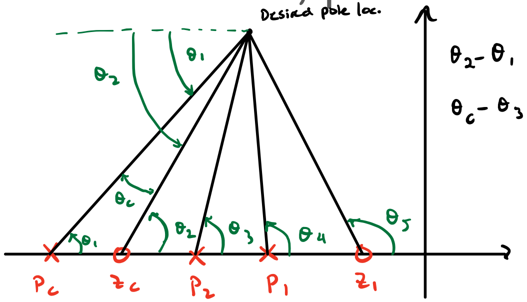
② compensated system transient response varies based on the zero location
- ↳ same ζ ∝ %OS
- ↳ larger -ve Real part ∝ shorter T_s
- ↳ larger Im part ∝ shorter T_p

Lead Compensator (passive)

$$K\frac{s+z_c}{s+p_c}$$

- Improves the transient response by APPROXIMATING differentiation

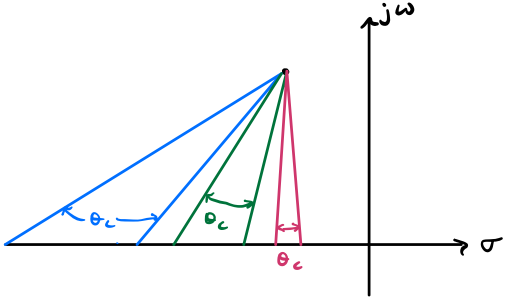
- Zero at $-z_c$ and pole at $-p_c$ are selected to put design point on root locus; pole at $-p_c$ is more negative than zero at $-z_c$.



$\theta_2 - \theta_1 = \theta_c$

$\theta_c - \theta_3 - \theta_4 + \theta_5 = (2k+1)180°$

Compensated



- Active circuits are not required to implement; passive is fine.

Method:

1. select a dominant 2nd order pole on s-plane
2. Angle condition for uncompensated
3. Diff. b/w. 180° and the Σ angles MUST be the angular contribution required by the compensator
4. An ∞ number of lead comps. could be used to meet transient req.

PD Controller Example

6. The unity feedback system shown in Figure P9.1 with

$$G(s) = \frac{K(s+6)}{(s+2)(s+3)(s+5)}$$

is operating with a dominant-pole damping ratio of 0.707. Design a PD controller so that the settling time is reduced by a factor of 2. Compare the transient and steady-state performance of the uncompensated and compensated systems. Describe any problems with your design. [Section: 9.3]

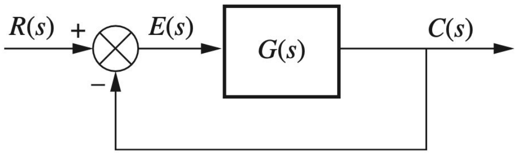
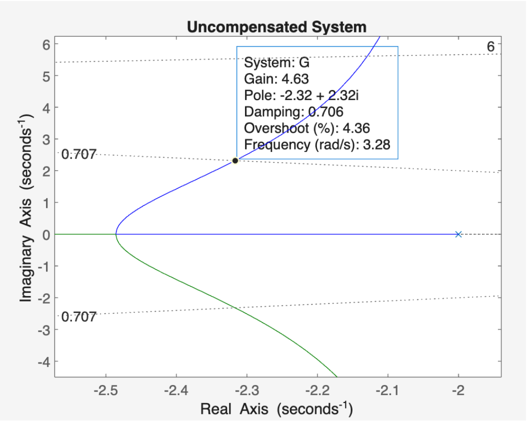
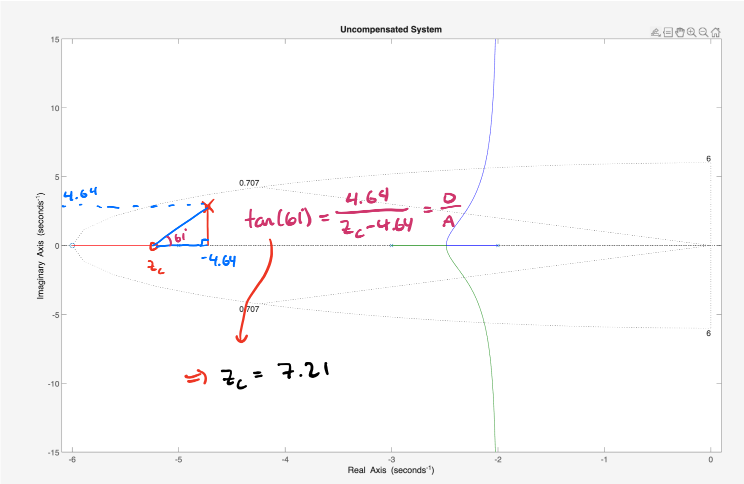


FIGURE P9.1

① PLOT ROOT LOCUS





② Look at performance Relations

$$T_s = \frac{4}{\zeta\omega_n} = \frac{4}{2.32} = 1.724\ s$$

we need to reduce by a factor of 2

so poles need to be at $-4.64 \pm 4.64j$

Angle criterion:

$$\sum_j \angle(s-p_j) - \sum_i \angle(s-z_i) = (2k+1)\pi$$

(just use matlab...)

$$\text{residual} = 180° + \sum_i \angle(s-z_i) - \sum_j \angle(s-p_j)$$

$$\angle(s+6) = 73.66°$$

$$\angle(s+2) = 119.64°$$

$$\angle(s+3) = 109.47°$$

$$\angle(s+5) = 85.56°$$

$$\sum_j \angle(s-p_j) = 314.67°$$

$$\text{res} = 180° + 73.66° - 314.67°$$

$$= -61°$$

③ Cascade Compensator + replot RL

$$\frac{K(s+7.21)(s+6)}{(s+2)(s+7)(s+5)} = G_c(s)$$


```
G = zpk([-6, -7.21], [-2, -3, -5], 1);
figure
rlocus(G)
title("Uncompensated System")
sgrid(0.707, 6)

z = -6;
p = [-2, -3, -5];        % poles
sd = -4.64 + 1j*4.64;    % desired closed-loop pole

angle_zero = angle(sd - z) * 180/pi;      % degrees
angle_poles = angle(sd - p) * 180/pi;

sum_zero   = angle_zero;
sum_poles  = sum(angle_poles);

fprintf("Angle from zero: %6.2f°\n", sum_zero);
fprintf("Angles from poles: [%6.2f°, %6.2f°, %6.2f°]\n", angle_poles);
fprintf(" Sum of pole angles: %6.2f°\n", sum_poles);

% (sum zeros) - (sum poles) = (2k+1)*180 ==> angle needed from the PD zero = 180 + sum_zeros - sum_poles
needed = 180 + (sum_zero - sum_poles);
fprintf("Angle needed from PD zero: %6.2f°\n", needed);
```

PD Controller Example

```
zeta = sqrt(2)/2
theta = acos(zeta)

s = tf('s')

G = (s+6)/(s+2)/(s+3)/(s+5)
G_cl = (s+6)/((s+2)*(s+3)*(s+5)+(s+6))
Ts_old = stepinfo(G_cl).SettlingTime

omega_old = 4/(Ts_old*zeta) %we know this is the distance of the pole from the origin

p_old = -omega_old*sqrt(2)/2 + omega_old*sqrt(2)/2*i

omega_new = 4/(Ts_old/2*zeta) % half the settling time!

p_new = -omega_new*sqrt(2)/2 + omega_new*sqrt(2)/2*i %this needs to be on the RL
a = real(p_new) %real part of the pole
b = imag(p_new)
%current angle contribution
currang = pi-atan(b/(-6-a)) - (pi-atan(b/(-2-a)) - pi-atan(b/(-3-a)) - pi-atan(b/(-5-a)))

%necessary contribution of additional zero
zero_contribution = pi-currang

%calculate zero location - we know the tan(angle) = a/(z-a)

z = -a-a/tan(zero_contribution) %OL zero at -7.0322!

Kgain = abs( (p_new+2)*(p_new+3)*(p_new+5) / ((p_new+6)*(p_new+z)) )

C = Kgain*(s+z)

T_cl = feedback(G*C, 1)

r = rlocus(T_cl/Kgain, Kgain)

stepinfo(T_cl).SettlingTime

T_new_est = Ts_old/2
```

```
Continuous-time transfer function.
Model Properties
G_cl =

           s + 6
  -----------------------
  s^3 + 10 s^2 + 32 s + 36

Continuous-time transfer function.
Model Properties
Ts_old = 1.9049
omega_old = 2.9696
p_old = -2.0998 + 2.0998i
omega_new = 5.9392
p_new = -4.1996 + 4.1996i
a = -4.1996
b = 4.1996
currang = 8.4474
zero_contribution = -5.3058
z = 7.0322
Kgain = 3.8245
C =

  3.825 s + 26.89

Continuous-time transfer function.
Model Properties
T_cl =

     3.825 s^2 + 49.84 s + 161.4
  ---------------------------------
  s^3 + 13.82 s^2 + 80.84 s + 191.4

Continuous-time transfer function.
Model Properties
r = 3x1 complex
    -6.0472 + 5.1899i
    -6.0472 - 5.1899i
    -5.5545 + 0.0000i

ans = 0.8731

T_new_est = 0.9525
```

6. The unity feedback system shown in Figure P9.1 with

$$G(s) = \frac{K(s+6)}{(s+2)(s+3)(s+5)}$$

is operating with a dominant-pole damping ratio of 0.707. Design a PD controller so that the settling time is reduced by a factor of 2. Compare the transient and steady-state performance of the uncompensated and compensated systems. Describe any problems with your design. [Section: 9.3]

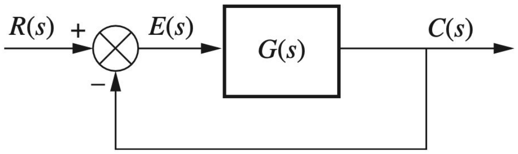


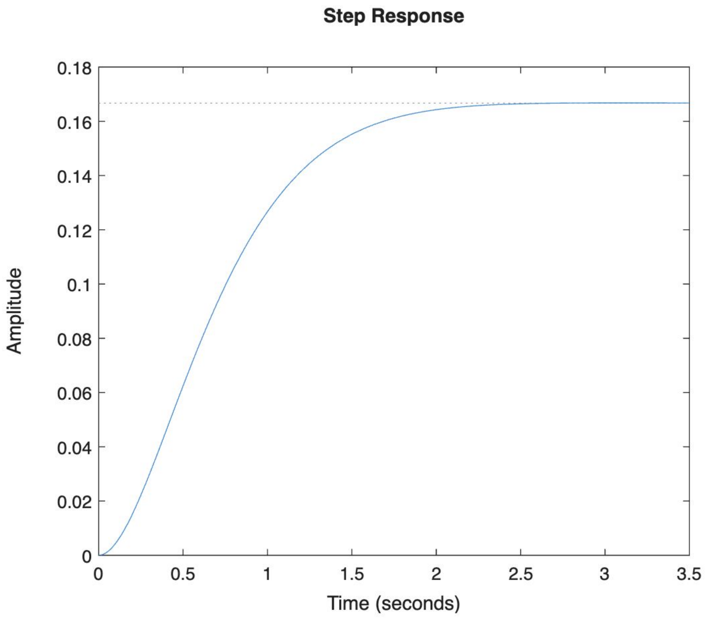
FIGURE P9.1

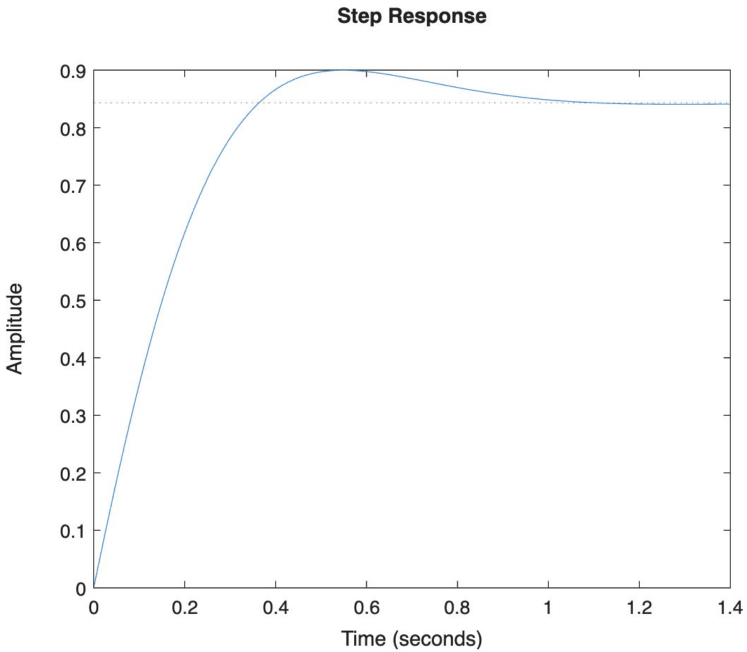
PD Controller Example

6. The unity feedback system shown in Figure P9.1 with

$$G(s) = \frac{K(s+6)}{(s+2)(s+3)(s+5)}$$

is operating with a dominant-pole damping ratio of 0.707. Design a PD controller so that the settling time is reduced by a factor of 2. Compare the transient and steady-state performance of the uncompensated and compensated systems. Describe any problems with your design. [Section: 9.3]





PID Controller (active)

$$K\frac{(s+z_{\text{lag}})(s+z_{\text{lead}})}{s}$$

- Improves steady-state error and transient response by cascading a PI and PD controller simultaneously.

- Lag zero at $-z_{lag}$ and pole at origin improve steady-state error; lag zero at $-z_{lag}$ is close to, and left of, the origin.

- Lead zero at $-z_{lead}$ improves transient response; lead zero at $-z_{lead}$ is selected to put design point on root locus.

- Active circuits required to implement
  - Active → requires an external power source (e.g. op-amp)

- Can cause noise and saturation; implement with rate feedback or with an additional pole.

Lag Lead Compensator (passive)

$$K \frac{(s + z_{\text{lag}})(s + z_{\text{lead}})}{(s + p_{\text{lag}})(s + p_{\text{lead}})}$$

- Improves steady-state error and transient response by cascading a lag and lead controller simultaneously.
- Lag pole at $-p_{lag}$ (small and negative) and lag zero at $-z_{lag}$ (close to and left of $-p_{lag}$) are used to improve steady-state error.
- Lead pole at $-p_{lead}$ and lead zero at $-z_{lead}$ are used to improve transient response (both selected to put design point on root locus)
- Active circuits are not required to implement.

PID Controller Design Example

25. For the unity feedback system in Figure P9.1, with

$$G(s) = \frac{K}{(s+1)(s+3)}$$

design a PID controller that will yield a peak time of 1.122 seconds and a damping ratio of 0.707, with zero error for a step input. [Section: 9.4]

$\sqrt{2}/2 = 3\,dB = 45^\circ$

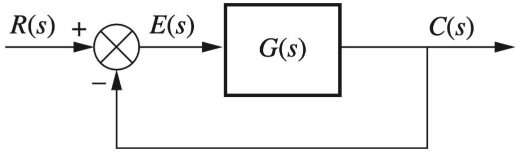


FIGURE P9.1

① check givens!

$T_p = 1.122 = \dfrac{\pi}{\omega_n\sqrt{1-\zeta^2}}$ imaginary part of CL compensated poles!

$s = \zeta\omega_n + j\omega_n\sqrt{1-\zeta^2}$

$s \in \mathbb{C}$

$\mathrm{Im}\{s\} = \dfrac{\pi}{1.122} = 2.8$

Recall from slide 2: $\dfrac{\mathrm{Im}\{s\}}{\mathrm{Re}\{s\}} = \tan(\cos^{-1}(\zeta))$

$\dfrac{2.8}{\mathrm{Re}\{s\}} = \cancel{\tan\left(\cos^{-1}\left(\frac{\sqrt{2}}{2}\right)\right)}\ 1$

$\mathrm{Re}\{s\} = 2.8$

$\Rightarrow s = 2.8 \pm 2.8j$

② Assume PI controller

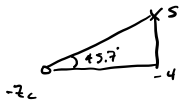
$G_c(s) = \dfrac{s+0.1}{s}$ $\Rightarrow$ reduces ss error to 0!

use system pole + zero of PI and plot Root Locus!

↳ Σ angles to the $s = 2.8 \pm 2.8j$

$\Rightarrow \Sigma$ angles $- 180^\circ = 45.7^\circ$

PD



solve z_c using tan $= 7.71$

$\Rightarrow \mathrm{PID} = \dfrac{(s+7.71)(s+0.1)}{s}$

plot new Root Locus ✓

gain $= 1.683$

Atill will post

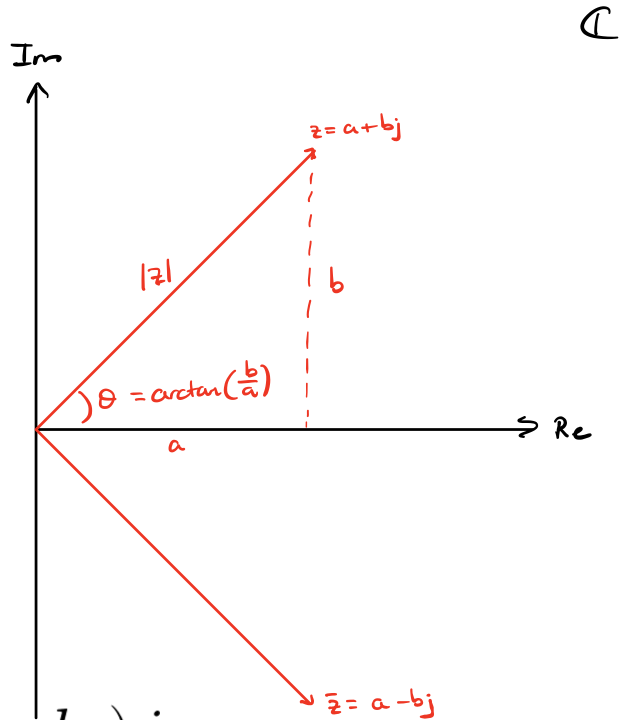
General Design Methodology

Go look at Athul's Root Locus Note! But as a quick reference:

1. Translate Design Specs using Performance Relations to ζ and ω_n and plot out the template region
2. Plot the Root Locus and locate dominant poles
3. Shape with Compensators/Controllers
4. Re-plot and iterate until dominant poles lie in template region
5. Validate in time domain and frequency domain!

Complex Arithmetic and Phasors

1. Set of Complex Numbers: $\mathbb{C}$
2. Complex Number: $z = a + bj = \text{Re}\{z\} + \text{Im}\{z\}$
   a. real part + imaginary part
3. Complex Conjugate: $z = a + bj \rightarrow \bar{z} = a - bj$
4. Magnitude: $|z| = r = \sqrt{a^2 + b^2}$
5. Phase: $\angle z = \theta = \arctan \dfrac{b}{a}$
6. Euler's Identity: $e^{j\theta} = \underbrace{\cos(\theta)}_{a} + j\,\underbrace{\sin(\theta)}_{b}$



For products of complex numbers we have:

$$z_1 z_2 = r_1 r_2 \angle(\theta_1 + \theta_2) = a_1 a_2 - b_1 b_2 + (a_1 b_1 + a_2 b_2) j$$

For ratios of complex numbers we have:

$$\frac{z_1}{z_2} = \frac{r_1}{r_2} \angle(\theta_1 - \theta_2)$$

Complex Arithmetic and Phasors

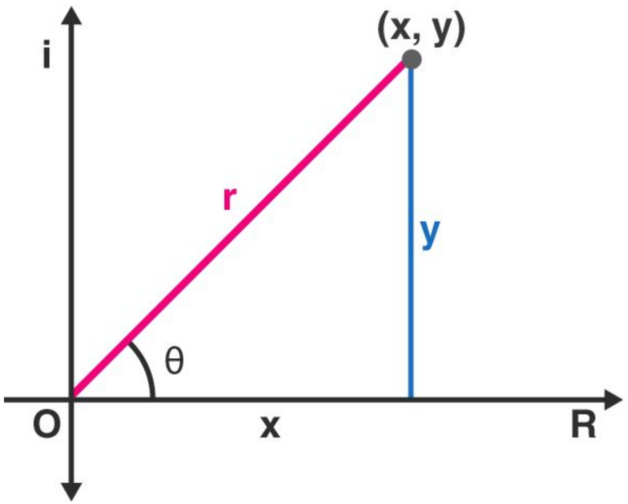
Given: $z = a + bj = \mathrm{Re}\{z\} + \mathrm{Im}\{z\}$

Polar/Phasor Form: $z = r\angle\theta$

$\sim A\cos(\omega t + \phi)$
$\equiv A\angle\phi$

Rectangular Form: $z = a + bj$

Natural Logarithm/ Exponential Form: $z = re^{j\theta}$



Fundamental Theorem of Algebra: Any polynomial of degree *n* has exactly *n* roots in the complex plane. If the coefficients of the polynomial are all read, then the roots concur in conjugate pairs. If it has odd degree, it **must** have a real root.

↳ allows us to write TF as products i.e. $\dfrac{\prod(s-z_c)}{\prod(s-p_c)}$

Bode Plots

visualizes FR of a system across the entire spectrum!

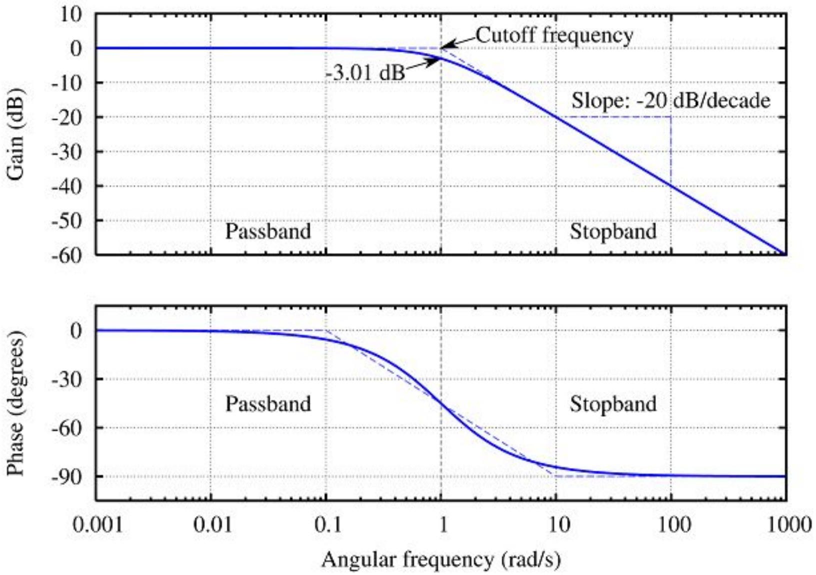
Overview

- Relates the phase and gain of the transfer function to the frequency of the input signal passed in

- Gain on plots is reported in dB, which is equal to:

$$20\log_{10}|H(s)|$$

log based on human hearing!



Given an input signal x(t), TF H(s), and an output signal y(t):

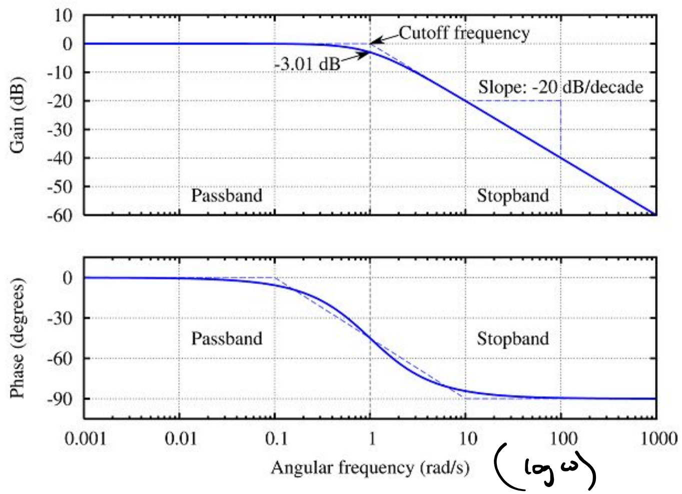
- Let $x(t) = \sum_i a_i \cos(\omega_i t)$

$$\rightarrow y(t) = \sum_i a_i |H(j\omega_i)| \cos(\omega_i t + \angle H(j\omega_i))$$

Drawing Bode Plots - Straight Line Approximation

1. Logarithmic Frequency Axis
2. Initial Points
   a. On the y-axis:
      i. For Magnitude: calculate
      ii. For Phase: calculate
3. Drawing
   a. Magnitude Rules (for stable p/z only)
      i. nth order OL zero: slope → slope + 20n
      ii. nth order OL pole: slope → slope - 20n
   b. Phase Rules (for stable p/z, unstable has the rules flipped)
      i. nth order OL zero at frequency ω → from 0.1ω to 10ω, phase changes by +90n linearly, then straightens out
      ii. nth order OL pole at frequency ω → from 0.1ω to 10ω, phase changes by -90n linearly, then straightens out
4. Overlap - if the effects of any pole/zero coincide with another, sum them up.

i.e. $\frac{1}{(s+1)^2}$



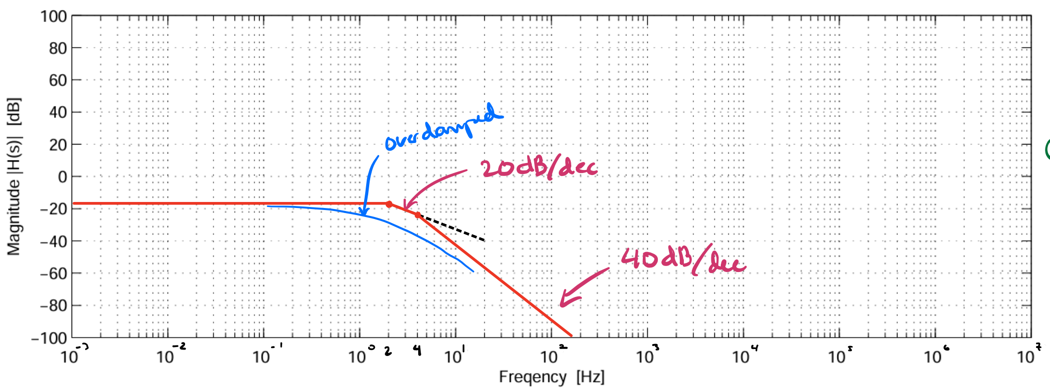
Bode Plot Example

any form $(s-a)$ generates a break frequency!

Find analytical expressions for the magnitude and phase responses of

$$G(s) = \frac{1}{(s+2)(s+4)}$$

Make plots of the log-magnitude and the phase, using log-frequency in rad/s as the ordinate.



① INITIAL PTS

$$|G(j\omega)|\big|_{\omega=0} = |G(0)| = \frac{1}{(0+2)(0+4)} = \frac{1}{8}$$

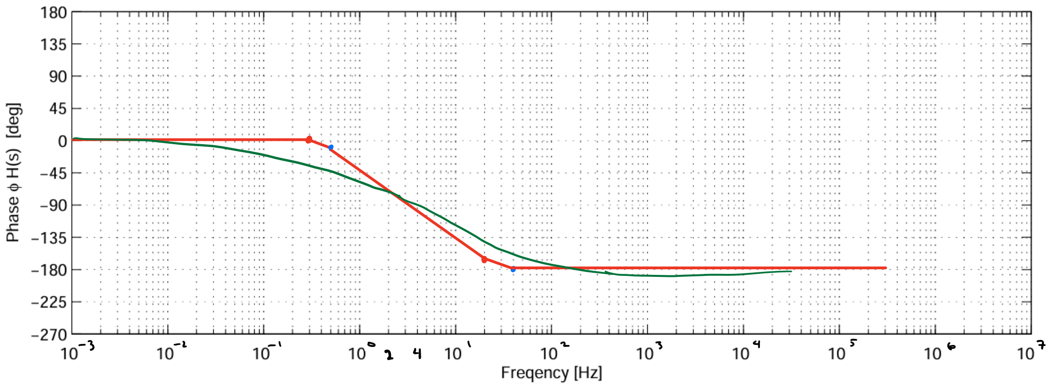
$$\Rightarrow 20\log_{10}|G(0)| = 20\log_{10}(0.125) = -18.06\,dB$$

② Break Points $(\omega=2,\ \omega=4)$

↳ both are pole contributions! so

at $\omega=2 \Rightarrow -20n$, $n=1$ ← only 1 pole atm

at $\omega=4 \Rightarrow -20n$, $n=2$ ← 2 poles!



atan2(b,a) if $a>0 \Rightarrow \arctan(b/a)$

$$\angle(G(j\omega)) = \sum_j \arg(j\omega - z_j) - \sum_i \arg(j\omega - p_i)$$

if −ve arg(const.) $= \pi$

$$= \arg(1) - \arg(s+2) - \arg(s+4)$$

$$= 0 - \arctan\left(\frac{\omega}{2}\right) - \arctan\left(\frac{\omega}{4}\right)$$

at $\omega=0 \Rightarrow \angle G(0) = 0$

For $\omega=2 \Rightarrow$ • $0.2 \rightarrow 20$ by $\omega_{mid}=2$

-45° by 20 -90°

$\omega_{mid} = \sqrt{\omega_{start}\,\omega_{end}}$

$\omega_{start} = 0.1\,\omega_p$

$\omega_{end} = 10\,\omega_p$

$\omega=4 \Rightarrow$ • $0.4 \rightarrow 40$ by $\omega_{mid}=4$

-45° by 40 -90°

sum them up! $\Rightarrow -90^\circ$ by

Performance Characteristics (FR)

T_s, T_p, T_r Time response Bandwidth Closed-loop

relationship between frequency response (FR) and time response! ⟹ Duality between speed of TR and BW of CL, FR

1. Closed-Loop Bandwidth $\omega_{BW} = \omega_n\sqrt{(1-2\zeta^2)+\sqrt{4\zeta^4-4\zeta^2+2}}$

↳ interpretation ①: frequency ω_{BW} where magnitude curve on Bode is 3 dB from its value at 0 frequency.
($M = 1/\sqrt{2}$)

↳ interpretation ②: waves slower than ω_{BW} pass through, waves faster than ω_{BW} get attenuated (Phone demo)

2. Gain Margin (USE PHASE CURVE)

↳ interpretation ①: locate ω_{GM} where $\angle G(j\omega) = 180°$. look at $|G(j\omega)| = M$ plot at ω_{GM}. The GM is then just how much gain is required to raise M curve to 0 dB

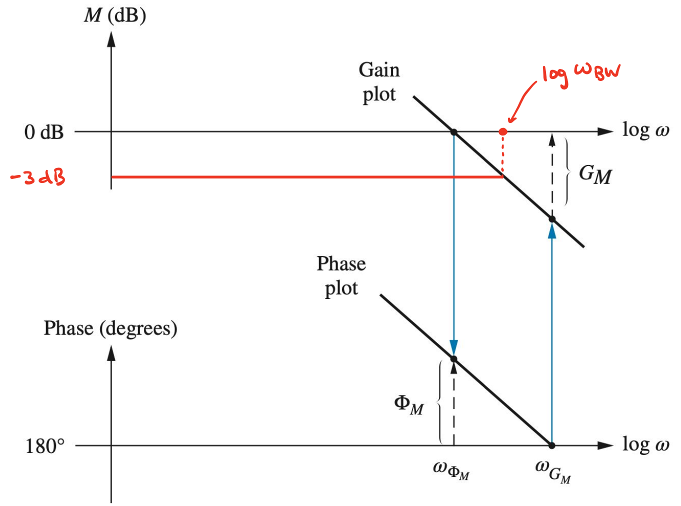
↳ interpretation ②: how much extra amplification before instability (0 dB crossing at 180°)
(sustained oscillations)

(crossover freq.)

3. Phase Margin (USE MAGNITUDE CURVE)

↳ interpretation ①: Find ω_{Φ_M} where M = 0 dB, Then at ω_{Φ_M}, look at phase curve. The distance from that pt. to 180° is the phase margin

↳ interpretation ②: how many degrees the phase has to decrease in order to reach 180°. If PM −ve ⟹ unstable (how much more delay you can tolerate before instability)



HIGHER MARGINS ⟹ MORE STABLE, Typically, PM > 45°. Also generally, higher 0 dB crossing ⟹ (not always) faster response

MATLAB tools

`bode(sys)`: generates Bode plot for a system

`rlocus(sys)`: generates root locus for a system

This has issues it seems...

`sgrid(zeta, omega)`: plots line of constant damping ratio on rlocus

`zpk(sys)`: allows to place zeros, poles, and gain of the system

`bandwidth(sys, drop)`: calculates the bandwidth frequency for a given decibel drop (default is -3dB)

`margin(sys)`: calculates the gain/phase margins + their frequencies office hours!