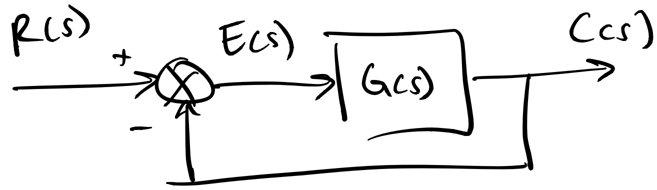
ME134 HW7 Trey Fortmuller

4/3/18

---

1) PI Controller

$$G(s) = \frac{k}{(s+2)(s+3)(s+12)}$$



$(s^2 + 5s + 6)(s+12)$

$s^3 + 12s^2 + 5s^2 + 60s + 6s + 72$

$s^3 + 17s^2 + 66s + 72$

the open loop transfer fn:

$$G(s) = \frac{1}{s^3 + 17s^2 + 66s + 72}$$

Now we use rlocus in MATLAB to plot the root locus of the closed loop transfer function, and determine the gain k for which $\zeta = 0.5$. See the appendix for the code.

→ we find
$$\begin{cases} s = -1.94 \pm 3.36i \; \} \text{ dominant poles} \\ k = 126 \\ \zeta = 0.5 \end{cases}$$

The steady state error of the uncompensated system is $e(\infty) = \dfrac{1}{1+k_p}$

$$k_p = \lim_{s \to 0} G(s) = \lim_{s \to 0} \frac{126}{(s+2)(s+3)(s+12)} = \frac{126}{72} = 1.75$$

so

$$e(\infty) = \frac{1}{1+1.75} = \boxed{0.36}$$

→ Now we add an ideal integral compensator to the forward path

$$G_c(s) = \underbrace{k_1}_{P} + \underbrace{\frac{k_2}{s}}_{I} = \frac{sk_1 + k_2}{s} = k_1 \frac{\left(s + \frac{k_2}{k_1}\right)}{s}$$

we want a zero close to 0 for pole-zero cancellation in order to affect the transient response as little as possible.

→ we pick $\dfrac{k_2}{k_1} = 0.1$

→ we simplify the compensator to a single block & a single gain.

$$G_c(s) = \frac{k(s+0.1)}{s}$$

Our new, compensated, open loop transfer function for

$$\begin{cases} G_c = \dfrac{(s+0.1)}{s} \\ G = \dfrac{1}{(s+2)(s+3)(s+12)} \end{cases}$$

is

$$G_c(s)\,G(s) = \frac{(s+0.1)}{s^4 + 17s^3 + 66s^2 + 72s}$$

- we use MATLAB's rlocus and step functions to obtain the root locus plot, our plot has not changed significantly so transient response will not be greatly affected.
- The step info function applied to the compensated and uncompensated systems allows us to compare their step response characteristics.

1 PI Controller

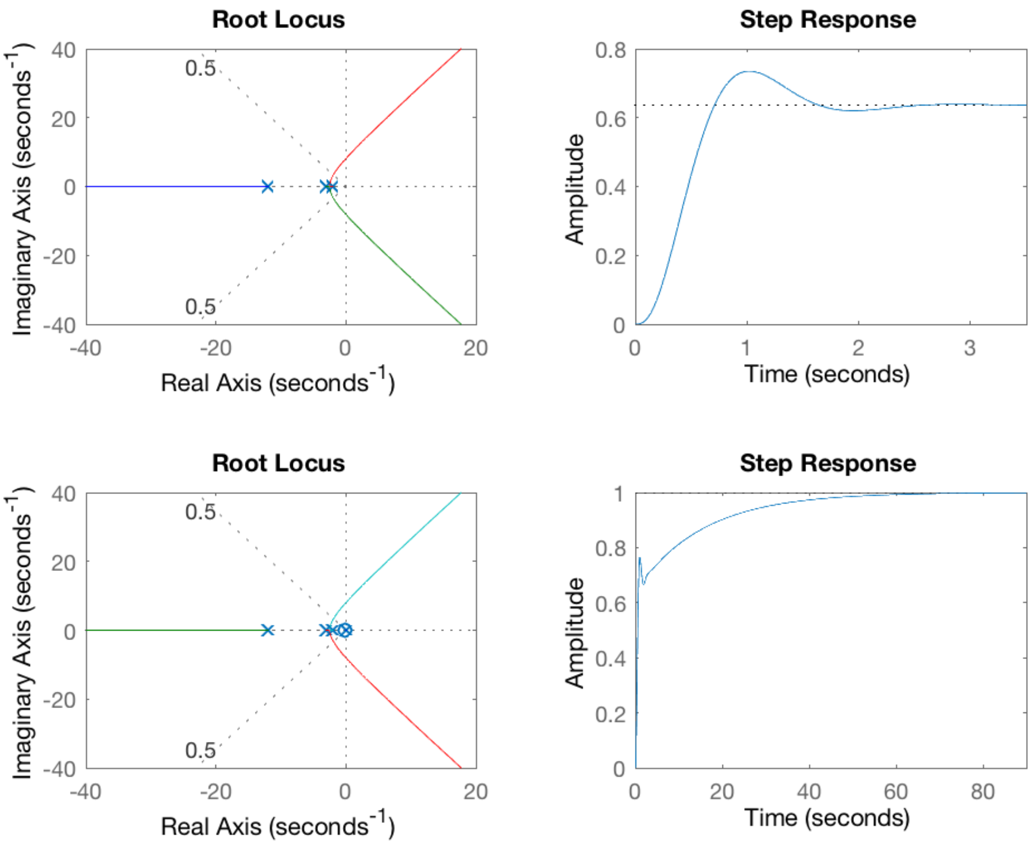


Figure 1: The root locus and step response of the uncompensated closed loop system on the top row, with the compensated system's root locus and step response on the bottom row

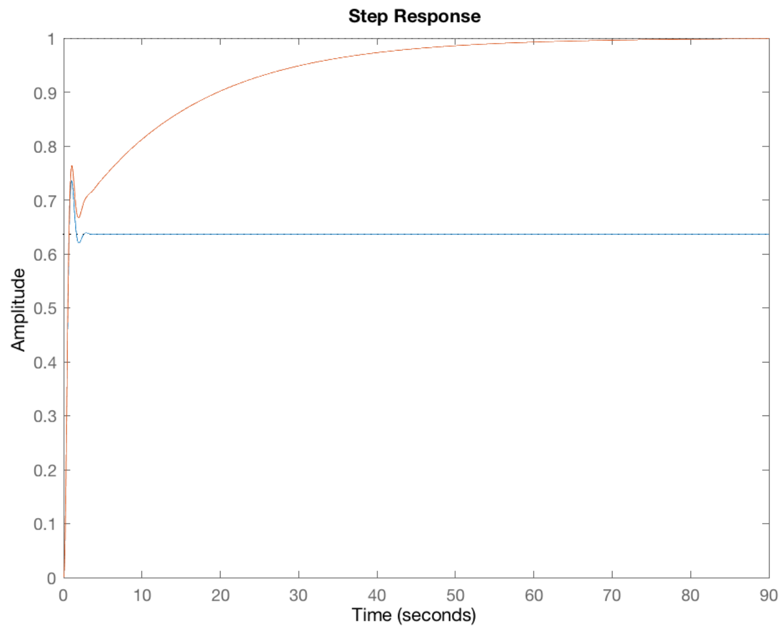


Figure 2: The uncompensated (in blue) and compensated (in red) step responses superimposed

```
ans =

  struct with fields:

        RiseTime: 0.4466
    SettlingTime: 2.1462
     SettlingMin: 0.5823
     SettlingMax: 0.7351
       Overshoot: 15.5222
      Undershoot: 0
            Peak: 0.7351
        PeakTime: 1.0207


ans =

  struct with fields:

        RiseTime: 19.5043
    SettlingTime: 44.5511
     SettlingMin: 0.9003
     SettlingMax: 0.9996
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9996
        PeakTime: 103.3629
```

Figure 3: The `stepinfo` step response characteristics of the uncompensated system (on top), and the compensated system (on bottom)

2) PD Controller

$$G(s) = \frac{k}{s(s+10)(s+20)}$$ is the open loop transfer function

$$= k \cdot \frac{1}{s^3 + 30s^2 + 200s}$$

using MATLAB's rlocus, we find the gain for $\zeta = 0.456$ is $k = 1160$.

→ using MATLAB's stepinfo, we find the settling time is 1.2291.

we design a PD controller to meet the design requirement $T_s = \frac{1.2291}{4} = 0.307s$

$$\sigma = \zeta \omega_n$$

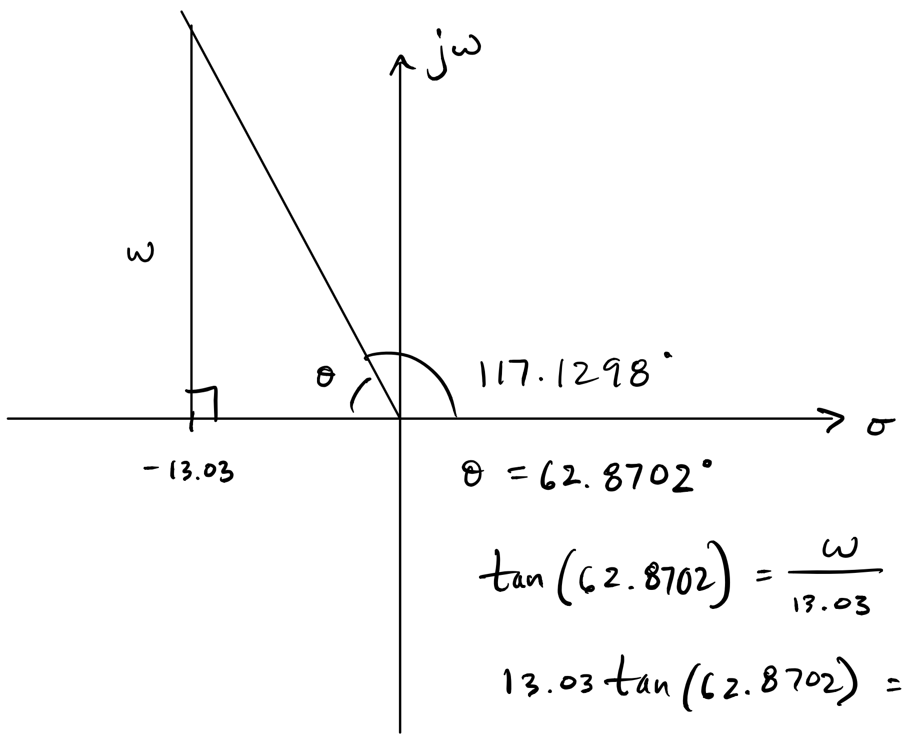
→ the third pole is ~−26 while the dominant poles are ~−3 on the real axis, so our 2nd order approximation of T_s is justified.

Now we design the PD controller

→ The real part of the compensated system's dominant poles will be $\sigma = \frac{4}{T_s}$

$$\sigma = \frac{4}{0.307s} = 13.03$$

we find the imaginary part w/ trig



$\theta = 62.8702°$

$$\tan(62.8702) = \frac{\omega}{13.03}$$

$$13.03 \tan(62.8702) = \omega$$

$$\omega = 25.4302$$

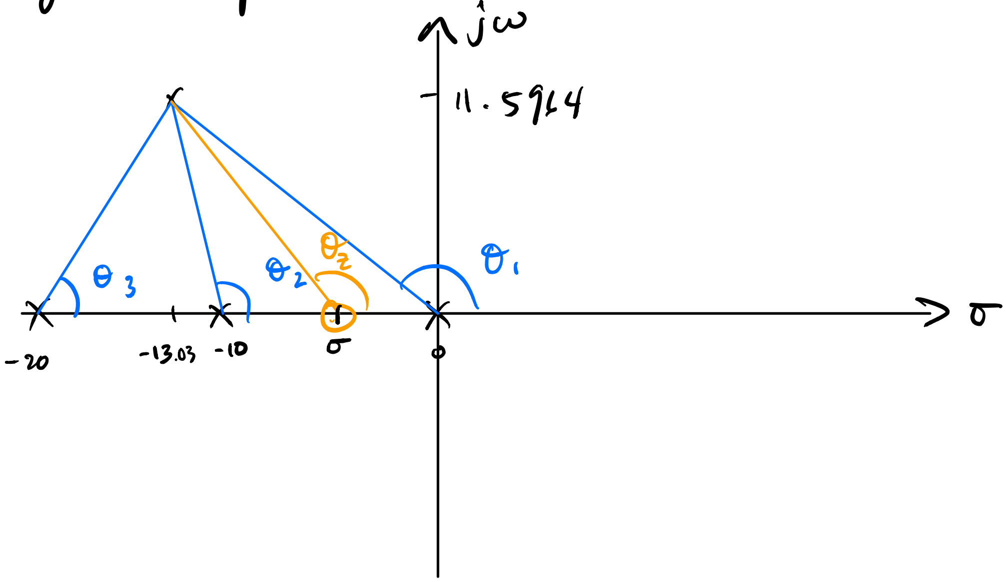
the dominant poles of the compensated system are thus $\boxed{-13.03 + 25.43j}$

we use MATLAB's damp function to find the uncompensated pole locations.

$$\begin{cases} \sigma_1 = 0 \\ \sigma_2 = -10 \\ \sigma_3 = -20 \end{cases}$$

Now we choose the location of the compensator's zero based on it's required angular contribution to place our poles.

$$\sum \text{angles of poles} - \sum \text{angles of zeros} = 180^\circ$$

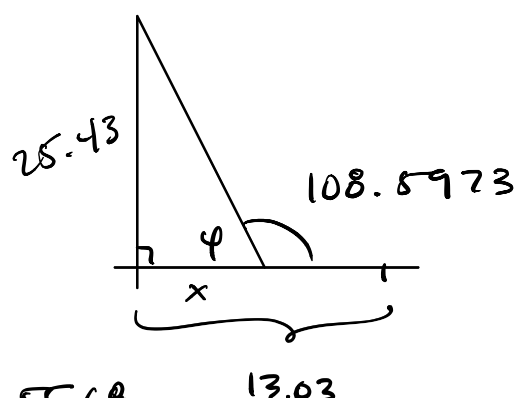


This trig is done in MATLAB, we get

$$\left.\begin{aligned} \theta_1 &= 117.13 \\ \theta_2 &= 96.7948 \\ \theta_3 &= 74.6725 \end{aligned}\right\} \Sigma = 288.5973$$

so

$$\theta_z = 108.5973$$



$$\tan(\varphi) = \frac{25.43}{x}$$

$$x = \frac{25.43}{\tan(71.4027)} = 8.5568$$

$$\boxed{\sigma = -4.4732}$$

then our compensator $G_c(s) = k(s - \sigma)$

is $G_c(s) = k(s + 4.4732)$

so the compensated open loop T.F. is

$$T = G_c(s)\,G(s) = \frac{(s + 4.4732)}{s^3 + 30s^2 + 200s}$$

to achieve $T_s < 0.307\,s$ we make a gain adjustment up to $k = 2150$.

2 PD Controller

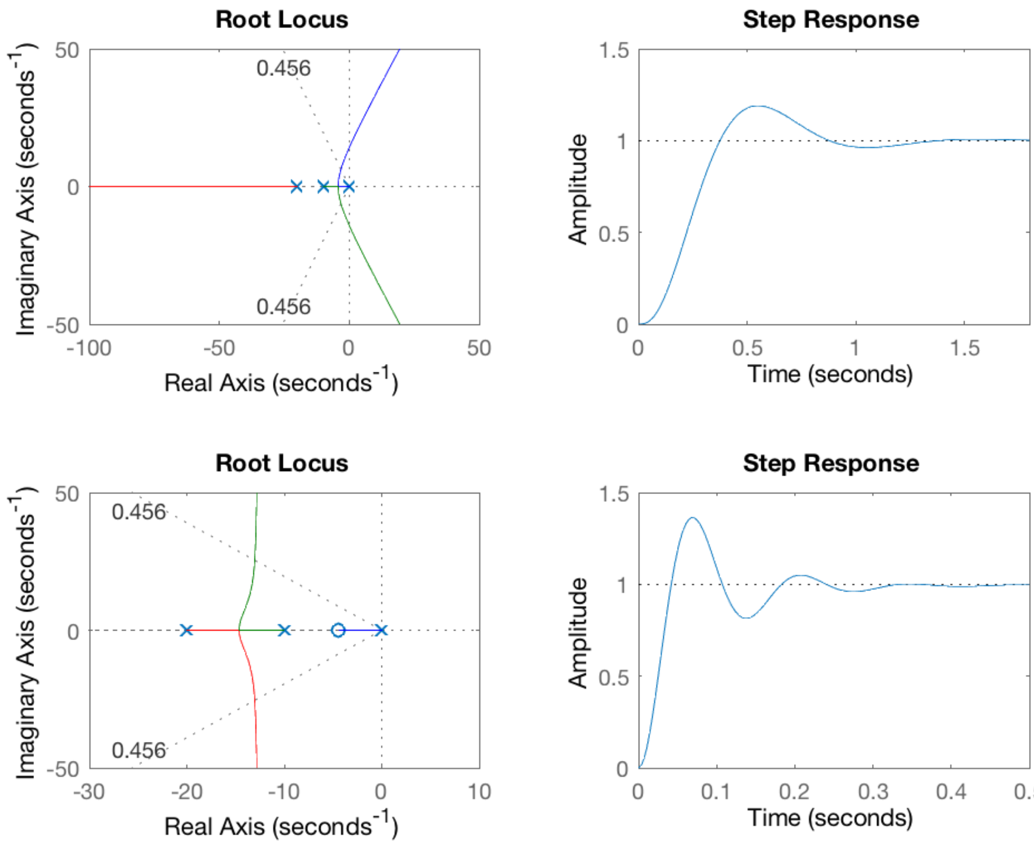


Figure 4: The root locus and step response of the uncompensated closed loop system on the top row, with the compensated system's root locus and step response on the bottom row

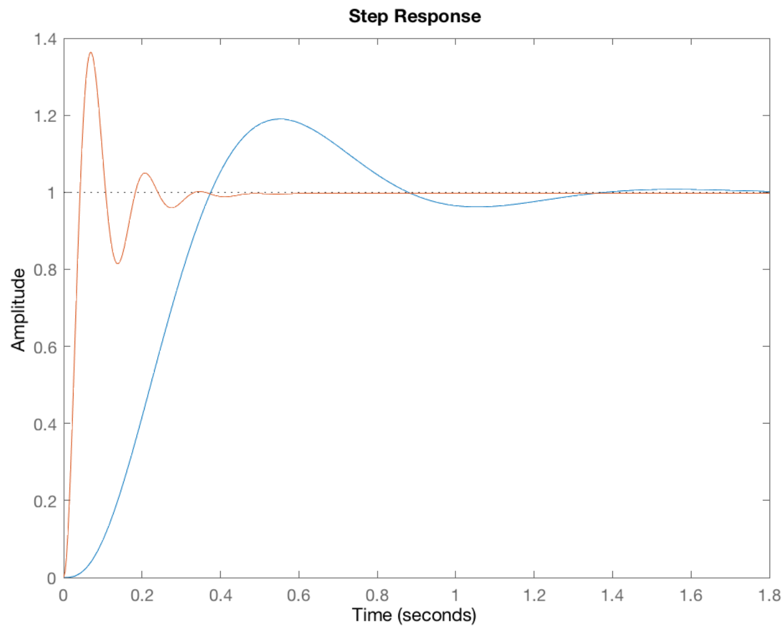


Figure 5: The uncompensated (in blue) and compensated (in red) step responses superimposed

```
ans =

  struct with fields:

        RiseTime: 0.2357
    SettlingTime: 1.2291
     SettlingMin: 0.9253
     SettlingMax: 1.1901
       Overshoot: 19.0090
      Undershoot: 0
            Peak: 1.1901
        PeakTime: 0.5477


ans =

  struct with fields:

        RiseTime: 0.0282
    SettlingTime: 0.3062
     SettlingMin: 0.8147
     SettlingMax: 1.3633
       Overshoot: 36.3304
      Undershoot: 0
            Peak: 1.3633
        PeakTime: 0.0681
```

Figure 6: The `stepinfo` step response characteristics of the uncompensated system (on top), and the compensated system (on bottom)

3) PID Controller

$$G(s) = \frac{k}{s(s+6)(s+10)}$$

we plot the unitary feedback system w/ MATLAB's step function and see that has $e(\infty) = 0$ already (this makes sense b/c the system is Type 1). So a PI compensator's gain would just be 0. We skip it and design the PD compensator.

→ using MATLAB's stepinfo function we find the unitary feedback system has a peak time of 0.9968 s which we seek to improve to 0.448 s

→ we find the real part of the compensated system's dominant poles via peak time

$$\sigma = \zeta \omega_n \; ; \quad T_p = \frac{\pi}{\omega_d} \; ; \quad \omega_d = \omega_n \sqrt{1-\zeta^2}$$

$$T_p = \pi \left(\omega_n \sqrt{1 - \zeta^2} \right)^{-1}$$

and

$$\omega_n = \frac{\sigma}{\zeta}$$

$$T_p = \frac{\pi}{\frac{\sigma}{\zeta} \sqrt{1 - \zeta^2}}$$

$$\frac{\sigma}{\zeta} T_p \sqrt{1 - \zeta^2} = \pi$$

then σ in terms of the peak time & the damping ratio is.

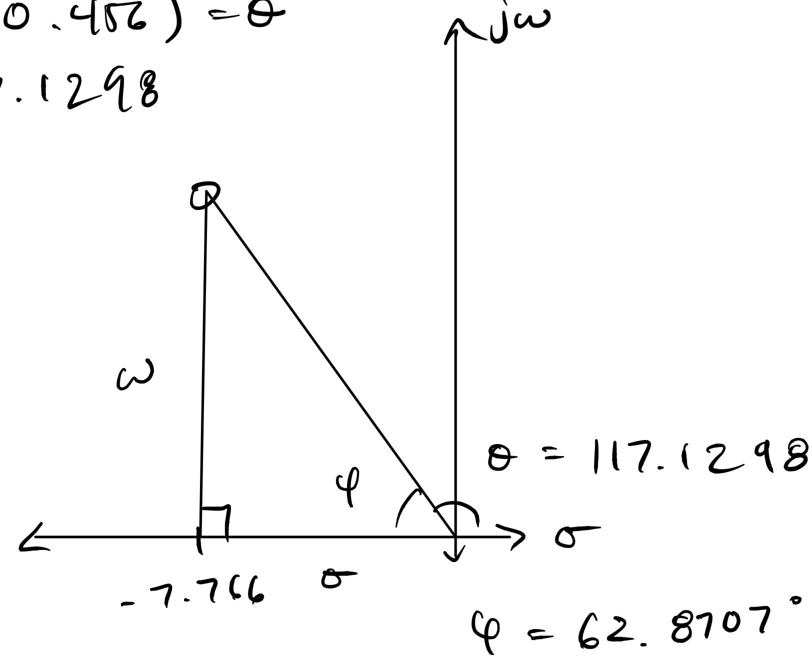
$$\sigma = \frac{\pi}{T_p \zeta \sqrt{1 - \zeta^2}}$$

we calculate this in MATLAB & find $\boxed{\sigma = 7.766}$

next, we find the imaginary part w/ trig on the imaginary plane using the damping ratio $\zeta = 0.456 = -\cos(\angle s)$

$\cos^{-1}(-0.456) = \theta$

$\theta = 117.1298$



$$\tan(\varphi) = \frac{\omega}{\sigma}$$

$$\omega = \sigma \tan(\varphi)$$

$$= 7.766 \tan(62.8707^\circ)$$

$$\omega = 15.1570$$

so the desired compensated zero is at

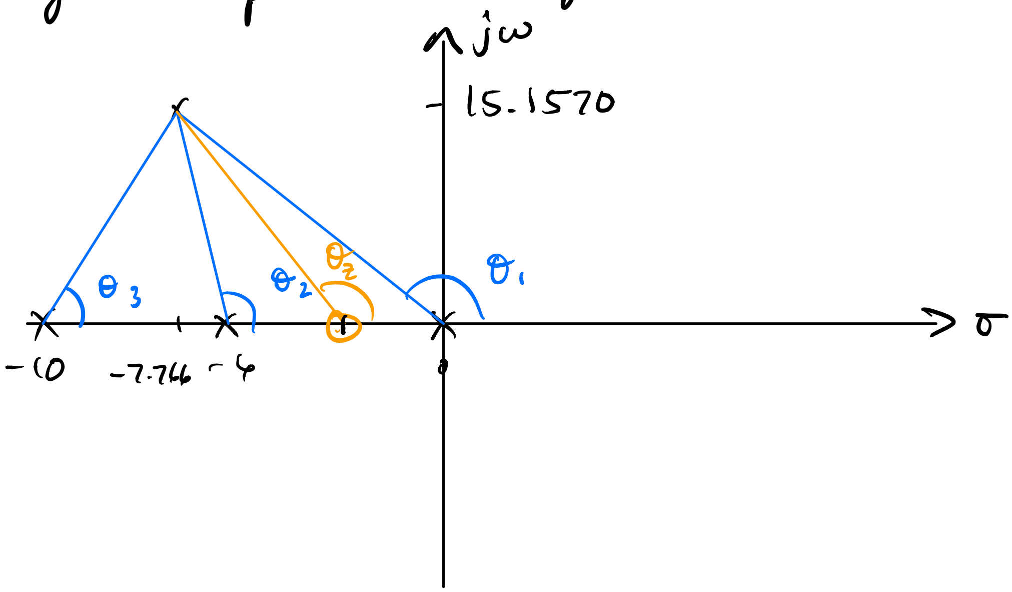
$$\boxed{-7.766 \pm 15.1570j}$$

we use MATLAB's damp function to find the uncompensated pole locations.

$$\begin{cases} \sigma_1 = 0 \\ \sigma_2 = -6 \\ \sigma_3 = -10 \end{cases}$$

Now we find the location of the PD compensator's zero, z_c, by calculating its angular contribution to our desired dominant poles.

$$\sum \text{angles of poles} - \sum \text{angles of zeros} = 180°$$

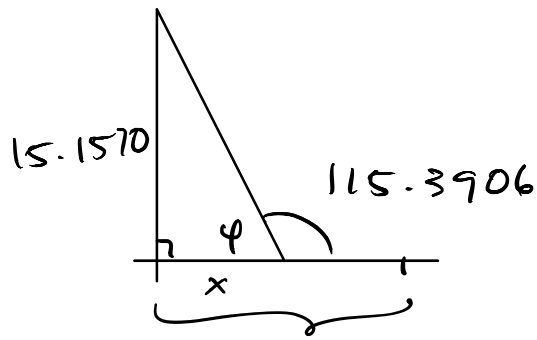


This trig is done in MATLAB, we get

$$\left.\begin{aligned} \theta_1 &= 117.1293 \\ \theta_2 &= 96.6458 \\ \theta_3 &= 81.6155 \end{aligned}\right\} \Sigma = 295.3906$$

So

$$\theta_z = 115.3906$$



$$\tan(\varphi) = \frac{15.1570}{x}$$

$$x = \frac{15.1570}{\tan(64.6094)} = 7.1940 \qquad 7.766$$

$$\boxed{\sigma = -0.5720}$$

compensator zero location.

then our compensator $G_c(s) = k(s-\sigma)$

is $G_c(s) = k(s+0.5720)$

so the compensated open loop T.F. is

$$T = G_c(s)\,G(s) = \frac{(s+0.5720)}{s^3+16s^2+60s}$$

in order to achieve a peak time of 0.448s we make a gain adjustment to $k = 245$ and achieve $T_p = 0.2017$, at the expense of settling time.

3 PID Controller

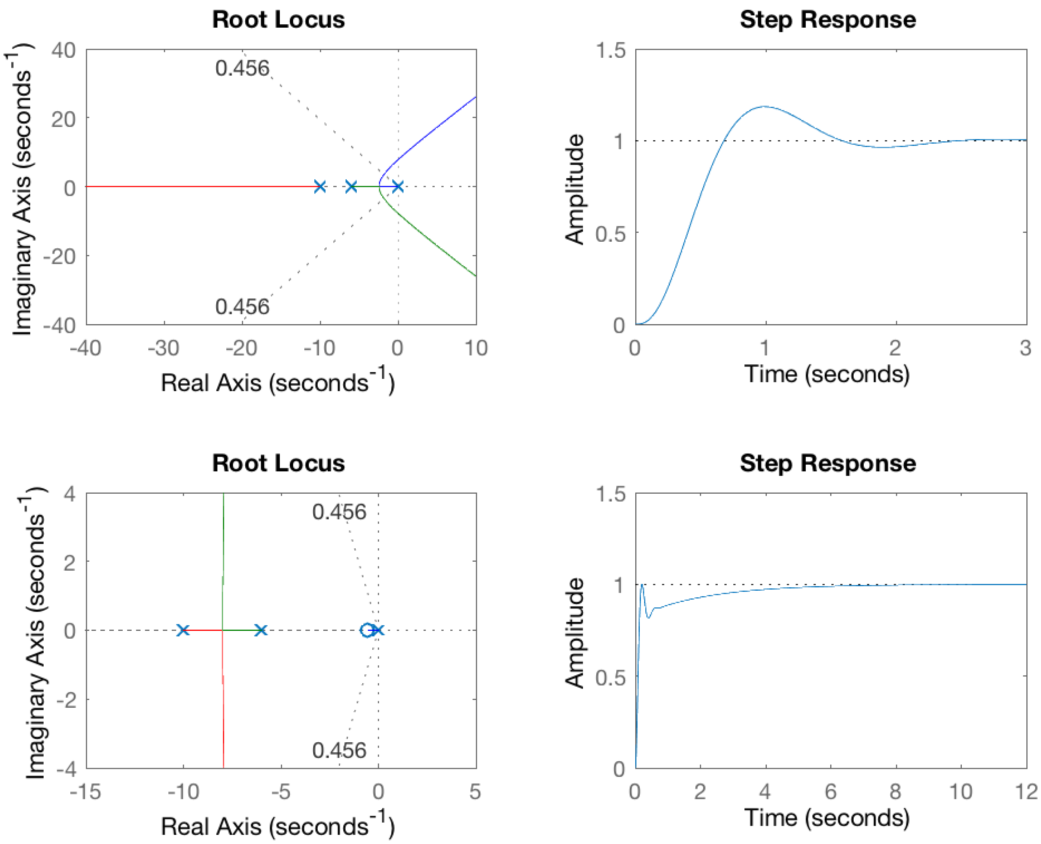


Figure 7: The root locus and step response of the uncompensated closed loop system on the top row, with the compensated system's root locus and step response on the bottom row

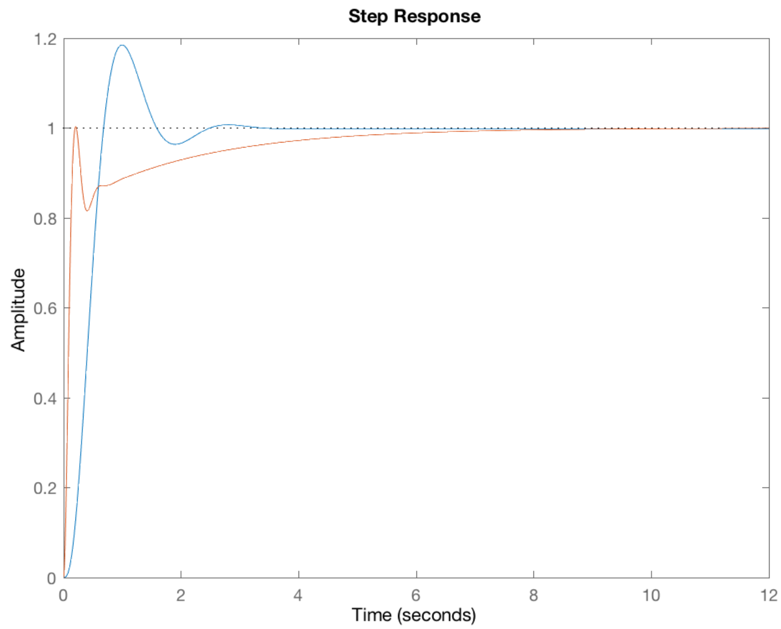


Figure 8: The uncompensated (in blue) and compensated (in red) step responses superimposed

```
ans =

  struct with fields:

        RiseTime: 0.4269
    SettlingTime: 2.2005
     SettlingMin: 0.9037
     SettlingMax: 1.1851
       Overshoot: 18.5062
      Undershoot: 0
            Peak: 1.1851
        PeakTime: 0.9968


ans =

  struct with fields:

        RiseTime: 0.1177
    SettlingTime: 4.6828
     SettlingMin: 0.8157
     SettlingMax: 1.0030
       Overshoot: 0.3035
      Undershoot: 0
            Peak: 1.0030
        PeakTime: 0.2017
```

Figure 9: The `stepinfo` step response characteristics of the uncompensated system (on top), and the compensated system (on bottom)

MATLAB Code Listing

```
% ME134 HW 7

%% PI Controller
close all % close all the previous figures

% the uncompensated system
G = tf([1], [1, 17, 66, 72]); % the open loop transfer function
f1 = figure;
subplot(2, 2, 1);
rlocus(G) % plot the root locus
sgrid([0.5], []) % plot the line of zeta = 0.5

G_cl = feedback(126*G, 1); % the unitary feedback, uncompensated system
subplot(2, 2, 2);
step(G_cl); % plot of the step response of the uncompensated system
stepinfo(G_cl) % step response characterisitcs of the uncompenstated system

% PI compensated system
G_comp = tf([1, 0.1], [1, 17, 66, 72, 0]); % open loop compensated tf
subplot(2, 2, 3);
rlocus(G_comp) % root locus of the compensated closed loop system
sgrid([0.5], []) % plot the line of zeta = 0.5

G_comp_cl = feedback(125*G_comp, 1); % unitary feedback, compensated system
subplot(2, 2, 4);
step(G_comp_cl);
stepinfo(G_comp_cl)

% plotting the step response of the uncompensated and compensated closed
% loop systems superimposed
f2 = figure;
hold on;
step(G_cl);
step(G_comp_cl);

%% PD Controller
close all;

G = tf(1, [1, 30, 200, 0]); % the open loop tf
f1 = figure;
subplot(2, 2, 1);
rlocus(G) % the root locus of the closed loop tf, to find he gain at zeta
sgrid(0.456, []);
damp(G) % get the uncompensated pole locations
% controlSystemDesigner(G)

K = 1160; % the gain for zeta = 0.456
G_cl = feedback(K*G, 1);
stepinfo(G_cl) % print the step info of the uncompensated system
subplot(2, 2, 2);
step(G_cl)

% get the angle between the ol poles and the desired cl pole
positive_real = [1, 0, 0]; % +real axis, with dim of 3 for cross product
des_pole1 = [-13.03, 25.43, 0]; % vector from the 1st ol pole to desired
des_pole2 = [-13.03 + 10, 25.43, 0];
des_pole3 = [-13.03 + 20, 25.43, 0];

theta1 = atan2d(norm(cross(positive_real,des_pole1)),dot(positive_real,des_pole1));
theta2 = atan2d(norm(cross(positive_real,des_pole2)),dot(positive_real,des_pole2));
theta3 = atan2d(norm(cross(positive_real,des_pole3)),dot(positive_real,des_pole3));
```

```
sum = theta1 + theta2 + theta3; % sum of the pole angles

theta_z = sum - 180; % the angular contribution required for the compensated pole

T = tf([1, 4.4732], [1, 30, 200, 0]); % the compensated open loop tf
subplot(2, 2, 3);
rlocus(T); % root locus of the compensated system
sgrid(0.456, []);

T_cl = feedback(2150*T, 1); % the compensated, closed loop transfer function
subplot(2, 2, 4); % plotting the step response of the compensated system
step(T_cl); % print the step info of the uncompensated system
stepinfo(T_cl)

% plotting the step response of the uncompensated and compensated closed
% loop systems superimposed
f2 = figure;
hold on;
step(G_cl);
step(T_cl);

%% PID Controller
close all;

G = tf(1, conv([1 0], conv([1 6], [1, 10]))); % the uncompensated ol tf
damp_ratio = 0.456; % the desired damping ratio of the system

f1 = figure;
subplot(2, 2, 1);
rlocus(G) % plot the root locus of the uncompensated system
sgrid(damp_ratio, []);
K = 190;

G_cl = feedback(K*G, 1);
stepinfo(G_cl)
subplot(2, 2, 2);
step(G_cl)

% the real part of the location of the compensated dominant poles
uncomp_tp = 0.9968; % the uncompensated system's peak time
sigma = pi / (uncomp_tp * damp_ratio * sqrt(1 - damp_ratio^2));

% get the uncompensated pole locations
% damp(G);

% the angular contribution of the compensator's zero to the poles
% get the angle between the ol poles and the desired cl pole
positive_real = [1, 0, 0]; % +real axis, with dim of 3 for cross product
des_pole1 = [-7.766, 15.1570, 0]; % vector from the 1st ol pole to desired
des_pole2 = [-7.766 + 6, 15.1570, 0];
des_pole3 = [-7.766 + 10, 15.1570, 0];

theta1 = atan2d(norm(cross(positive_real,des_pole1)),dot(positive_real,des_pole1));
theta2 = atan2d(norm(cross(positive_real,des_pole2)),dot(positive_real,des_pole2));
theta3 = atan2d(norm(cross(positive_real,des_pole3)),dot(positive_real,des_pole3));

sum = theta1 + theta2 + theta3; % sum of the pole angles

theta_z = sum - 180; % the angular contribution required for the compensated pole

T = tf([1, 0.5720], [1, 16, 60, 0]); % the compensated open loop tf
subplot(2, 2, 3);
rlocus(T); % root locus of the compensated system
sgrid(0.456, []);
```

```
K_comp = 245; % the PD compensator's tuning gain

T_cl = feedback(K_comp*T, 1); % the compensated, closed loop transfer function
subplot(2, 2, 4); % plotting the step response of the compensated system
step(T_cl); % print the step info of the uncompensated system
stepinfo(T_cl)

% plotting the step response of the uncompensated and compensated closed
% loop systems superimposed
f2 = figure;
hold on;
step(G_cl);
step(T_cl);
```