FINAL PRJECT

Example 9.5, 9.9, Skills Assessment 9.3

APRIL 24, 2024

UNIVERSITY OF SOUTH FLORIDA
LINEAR CONTROL

Table of Contents

Example 9.9

Handwritten Calculation

The angle to draw the percent overshoot line is first calculated:

$$G(s) = \frac{k(s+8)}{(s+3)(s+6)(s+10)}$$

$$T_{p,new} = 0.6\, T_{p,uncompensated}$$

$$\%O.S_{new} = 30\%$$

1) Find poles for 30% O.S

$$O.S = e^{-\zeta\pi/\sqrt{1-\zeta^2}} = 0.3$$

$$\Rightarrow \frac{\zeta\pi}{\sqrt{1-\zeta^2}} = -\ln(0.3) = \ln(10/3)$$

$$\Rightarrow \pi^2\zeta^2 = \ln^2(10/3)(1-\zeta^2)$$

$$\Rightarrow \zeta^2\left[\pi^2 + \ln^2(10/3)\right] = \ln^2(10/3)$$

$$\Rightarrow \zeta = \sqrt{\frac{\ln^2(10/3)}{[\pi^2 + \ln^2(10/3)]}} = 0.3579$$

$$\Rightarrow \theta_\zeta = \cos^{-1}(0.3579) = 69.031$$

The root locus parameters (centroid, asymptotes, break-in, ...) is then calculated, and the hand-drawn root locus is provided:

* Draw root locus

$\begin{cases} \text{poles}: & -3, -6, -10 \\ \text{zeros}: & -8 \end{cases}$

$\Rightarrow n - m = 3 - 1 = 2$

$\Rightarrow 2$ loci $\rightarrow \infty$

Centroid:

$$\sigma_c = \frac{(-3-6-10)-(-8)}{n-m} = \frac{-11}{2}$$

$$= -5.5$$

Asymtote:

$$\theta_c = \frac{(2h+1)180^\circ}{n-m}$$

. $h = 0 \Rightarrow \theta_c = 90^\circ$

. $h = -1 \Rightarrow \theta_c = -90^\circ$

. Break out point s_B:

$-6 < s_B < -3$

$$K = \frac{-(s+3)(s+6)(s+10)}{s+8}$$

$$K = \frac{-(s+3)(s+6)(s+10)}{s+8}$$

$$= \frac{-(s^2+9s+18)(s+10)}{s+8}$$

$$= \frac{-(s^3+19s^2+108s+180)}{s+8}$$

$$\Rightarrow \frac{\sigma K}{\sigma s} = \frac{-(2s^3+43s^2+304s+684)}{(s+8)^2}$$

$$\frac{\sigma K}{\sigma s} = 0 \Rightarrow 2s^3+43s^2+304s+684 = 0$$

$$\Rightarrow \begin{cases} s = -4.623 \\ s = -8 \pm 1.7 \end{cases}$$

$$\Rightarrow s_B = -4.623$$

The dominant poles is calculated, and second order approximation is used to calculate peak Time.

The intersection of 69.031° line and root locus :

$s = -5.45 \pm 14j$

$K = \left| \dfrac{(s+3)(s+6)(s+10)}{s+8} \right|$

$\Rightarrow K = 206$

* Find another pole $-10 < s < -8$ with $K = 206$

	x	f(x)
1	-8.1	203.49
2	-8.085	238.86
3	-8.07	289.35
4	-8.055	367.35

203.49

Use Table, estimate $p_3 \simeq -8.09$

* Use 2nd order approximation :

$\Delta_{2nd}(s) = (s + 5.45 + 14j)(s + 5.45 - 14j)$

$\Rightarrow \begin{cases} \zeta\omega_n = -5.45 \\ \omega_n\sqrt{1-\zeta^2} = 14 \end{cases}$

$\Rightarrow T_{p,uncompensated} = \dfrac{\pi}{\omega_n\sqrt{1-\zeta^2}} = \dfrac{\pi}{14} = 0.224 \text{ s}$

The dominant new poles is found to meet the new peak Time

$$\Rightarrow T_{p,new} = 0.6(0.224) = 0.1344$$

2) Find $\Delta_{new}(s)$ for $T_{p,new}$

$$\% P.O = 30\% \Rightarrow \zeta = 0.3579$$

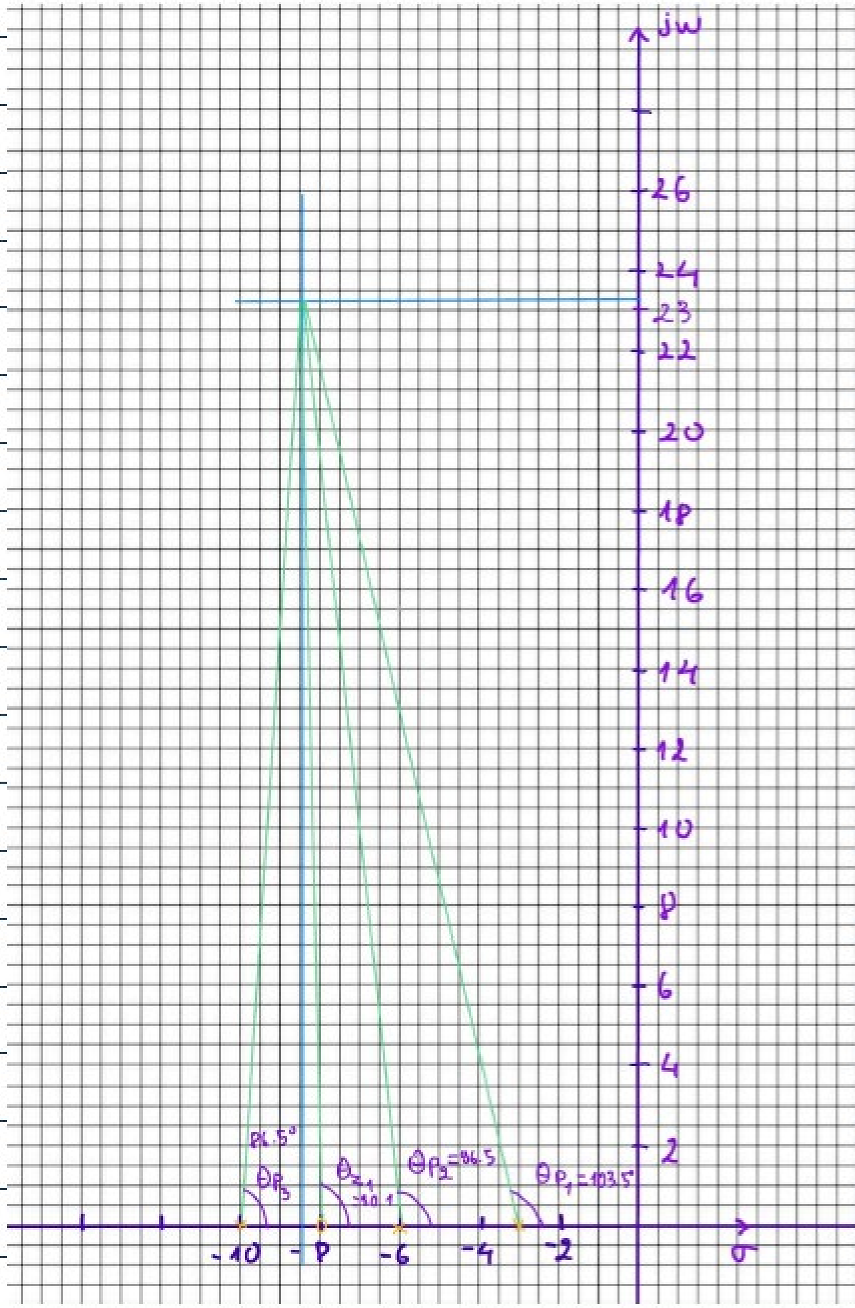
$$w_d = \frac{\pi}{T_{p,new}} = \frac{\pi}{0.13414} = 23.37$$

$$\Rightarrow w_n = \frac{23.37}{\sqrt{1-\zeta^2}} = 25.03$$

$$\Rightarrow \sigma = -\zeta w_n = -8.958$$

$$\Rightarrow s_{Desired} = -8.958 \pm 23.37$$

The angle contribution of each pole and zero is then calculated with respect to the desired dominant poles, and the position of the zero is determined:



$$\theta_{z_1} - (\theta_{p_3} + \theta_{p_2} + \theta_{p_1}) = 90.1° - (86.5° + 96.5° + 103.5°)$$
$$= -196.4°$$

$\Rightarrow$ need a new zero such that $\theta_{z_2} = -180° + 196.4°$
$$= 16.4°$$

The location of the new zero is calculated:

* Find location of new zero

New zero at $-z_c$

$$\Rightarrow \tan 16.4^\circ = \frac{23.37}{z_c - 8.958} \Rightarrow z_c = \frac{23.37}{\tan 16.4^\circ} + 8.958$$

$\Rightarrow z_c = 88$ (PD controller)

$$\Rightarrow \frac{K(s+8)(s+88)}{(s+3)(s+6)(s+10)} = -1$$

evaluate K at $s = -8.958 + 23.37j$

$\Rightarrow K = 6.89$

Arbitrarily choose the PI :

3) Add ideal PI to drive $e_{ss} \rightarrow 0$

Choose the PI to be $\frac{s+0.2}{s}$

Root Locus Parameters are calculated:

Root locus :

$n - m = 4 - 3 = 1 \Rightarrow$ only 1 loci go to ∞

Asymtote :

$$\theta_c = 180^\circ$$

Break out point $s_{b.o}$:

$$-6 < s_{b.o} < -3$$

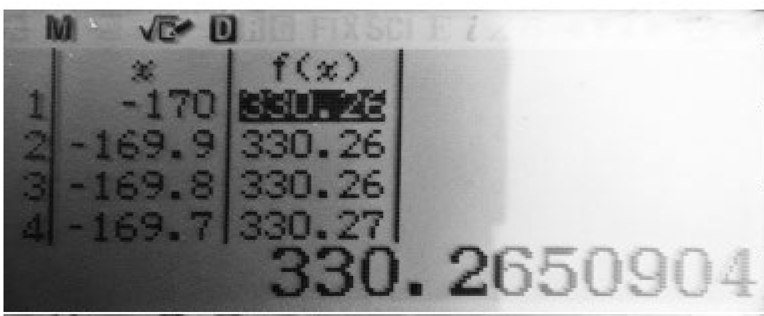
Break in point $s_{b.i}$:

$$s_{b.i} < -90$$

Use Table method , estimate $\min|K|\Big|_{s<-90}$ and $\max|K|\Big|_{-6<s<-3}$

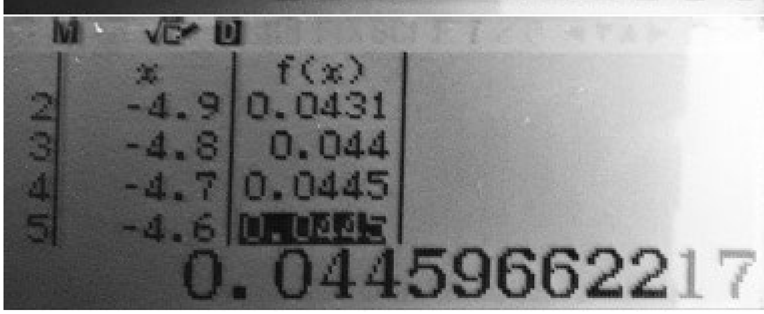
$$s_{b.i} \approx -169.9$$

$$s_{b.o} \approx -4.6$$



	x	f(x)
1	-170	330.26
2	-169.9	330.26
3	-169.8	330.26
4	-169.7	330.27

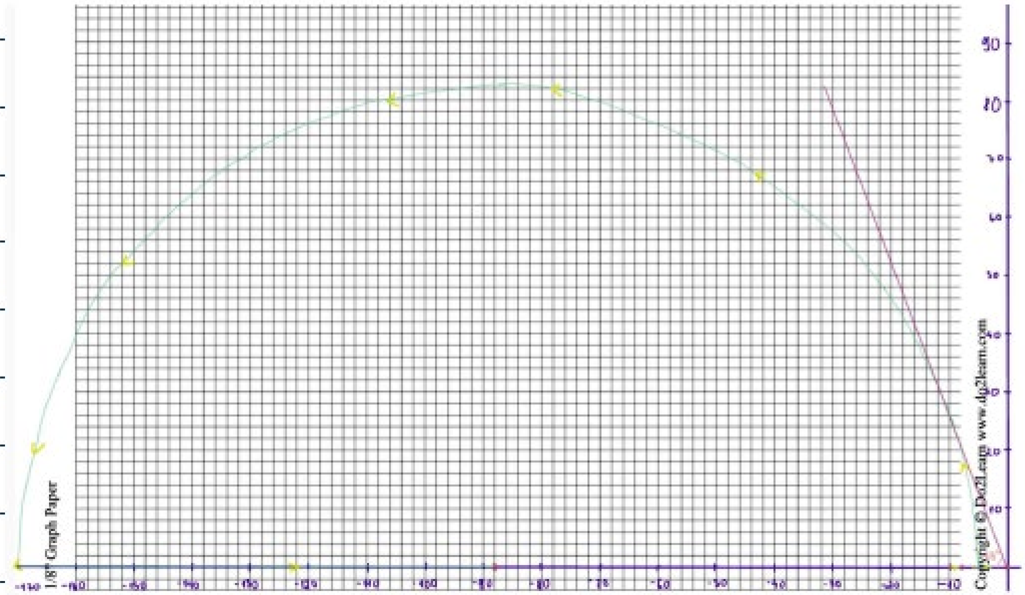
330.2650904



	x	f(x)
2	-4.9	0.0431
3	-4.8	0.044
4	-4.7	0.0445
5	-4.6	0.0445

0.04459662217

The Root Locus is sketched, and the gain the desired dominant poles is calculated:



Calculate gain at $s = -8.958 + 23.37j$

$$K = \left| \frac{s(s+3)(s+6)(s+10)}{(s+0.2)(s+8)(s+88)} \right|_{s=-8.958+23.37j}$$

$\Rightarrow K = 6.9$

4) Determine k_P, k_i, k_D

$$k_P + \frac{k_i}{s} + k_D s = \frac{k_D\left(s^2 + \frac{k_P}{k_D}s + \frac{k_i}{k_D}\right)}{s}$$

$$= \frac{k(s+88)(s+0.2)}{s} \Big|_{k=6.9}$$

$$= \frac{6.9(s^2 + 88.2s + 17.6)}{s}$$

$$\Rightarrow \begin{cases} k_D = 6.9 \\ \frac{k_P}{k_D} = 88.2 \\ \frac{k_i}{k_D} = 17.6 \end{cases}$$

$$\Rightarrow \begin{cases} k_D = 6.9 \\ k_P = 608.58 \\ k_i = 121.44 \end{cases}$$

MATLAB/ SIMULINK verification and Code

Matlab Code:

```
syms s

sys1 = tf([1 8],[1 3]);

sys2 = tf(1,[1 6]);

sys3 = tf(1, [1 10]);
```

```
sys = sys1*sys2*sys3;
rlocus(sys)

eta = sqrt(log(10/3)^2/(pi^2+log(10/3)^2));
% Define the angle in radians
angle_rad = deg2rad(180-acosd(eta));

% Define the length of the line
length = 50; % Adjust this value as needed

% Calculate the coordinates of the end point
x_end = length * cos(angle_rad);
y_end = length * sin(angle_rad);

% Plot the line
hold on
plot([0, x_end], [0, y_end], '-');
axis equal; % Ensure equal scaling on x and y axes
```

The Root Locus of the system is first drawn in **Figure 1**. The intersection between the 30% O.S line (light blue line) and the root locus is found as in **Figure 2**.

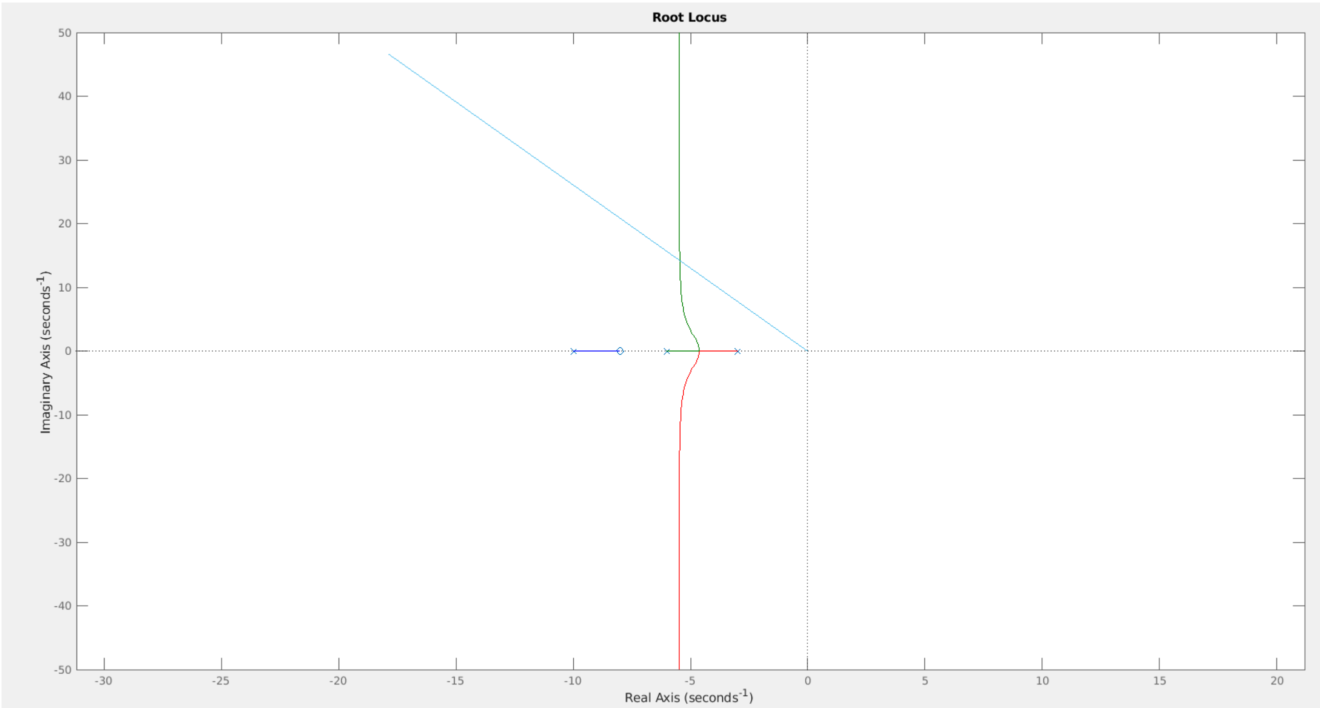


Figure 1. Root Locus of the Original System

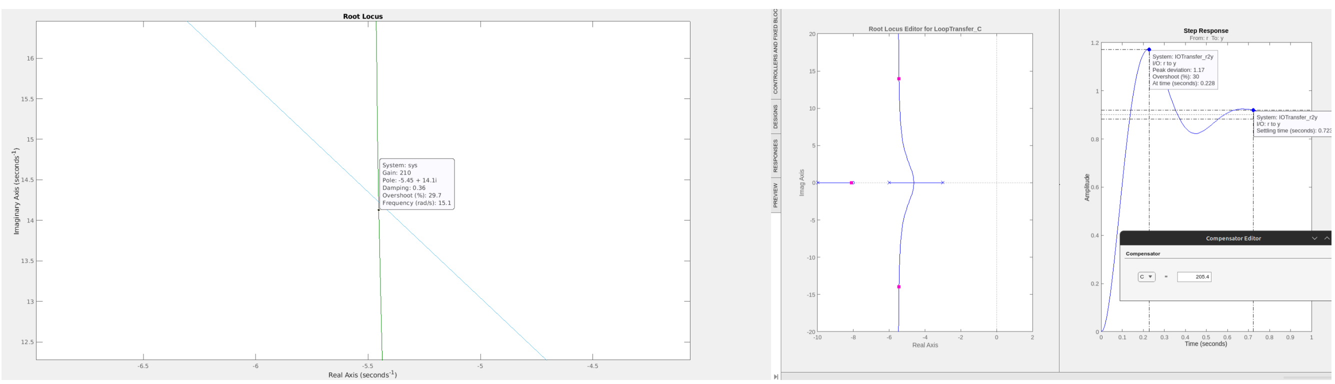


Figure 2. Intersection of the 30%O.S line and the Root Locus

The Dominant Poles found at the intersection is: $-5.45 \pm 14j$. The Gain at these dominant poles are also found using the rltool command as in the left of the **Figure 2**. The Gain is 205.4. Using this Control Window, the Third Poles is also found at -8.1 on the real axis (also shown in the left of **Figure 2**).

MATLAB Code:

```
Tp = 0.224;

Tpnew = 0.6*Tp;

wd = pi/(Tpnew);

wn = wd/sqrt(1-eta^2);

eta = sqrt(log(10/3)^2/(pi^2+log(10/3)^2));
```

```
syms s
sDesired = -eta*wn + wd*1i;
H = (s+8)/((s+10)*(s+6)*(s+3));
theta_zero = 180-rad2deg(double(angle(subs(H, s, sDesired))));
zc = imag(sDesired)/tand(theta_zero)-real(sDesired)

H = H*(s+zc);
K = -1/H;
Gain = double(abs(subs(K, s, sDesired)))

s = tf('s');
sys = (s+8)*(s+zc)/((s+10)*(s+6)*(s+3));
rltool(sys, Gain)
```

```
Tp = 0.224;
Tpnew = 0.6*Tp;
wd = pi/(Tpnew);
wn = wd/sqrt(1-eta^2);
eta = sqrt(log(10/3)^2/(pi^2+log(10/3)^2));

syms s
sDesired = -eta*wn + wd*1i
H = (s+8)/((s+10)*(s+6)*(s+3));
theta_zero = 180-rad2deg(double(angle(subs(H, s, sDesi
zc = imag(sDesired)/tand(theta_zero)-real(sDesired)

H = H*(s+zc);
K = -1/H;
Gain = double(abs(subs(K, s, sDesired)))
```

Command Window

```
>> Project95_1

sDesired =

  -8.9581 +23.3749i


zc =

   87.3027


Gain =

    6.9528
```

Figure 3. Calculating position for new zero for PD compensator

Using the second order approximation for the dominant poles, I found that the desired dominant poles are -8.958 ± 23.37, as shown in **Figure 3**. The root locus and the response is shown in **Figure 4**.

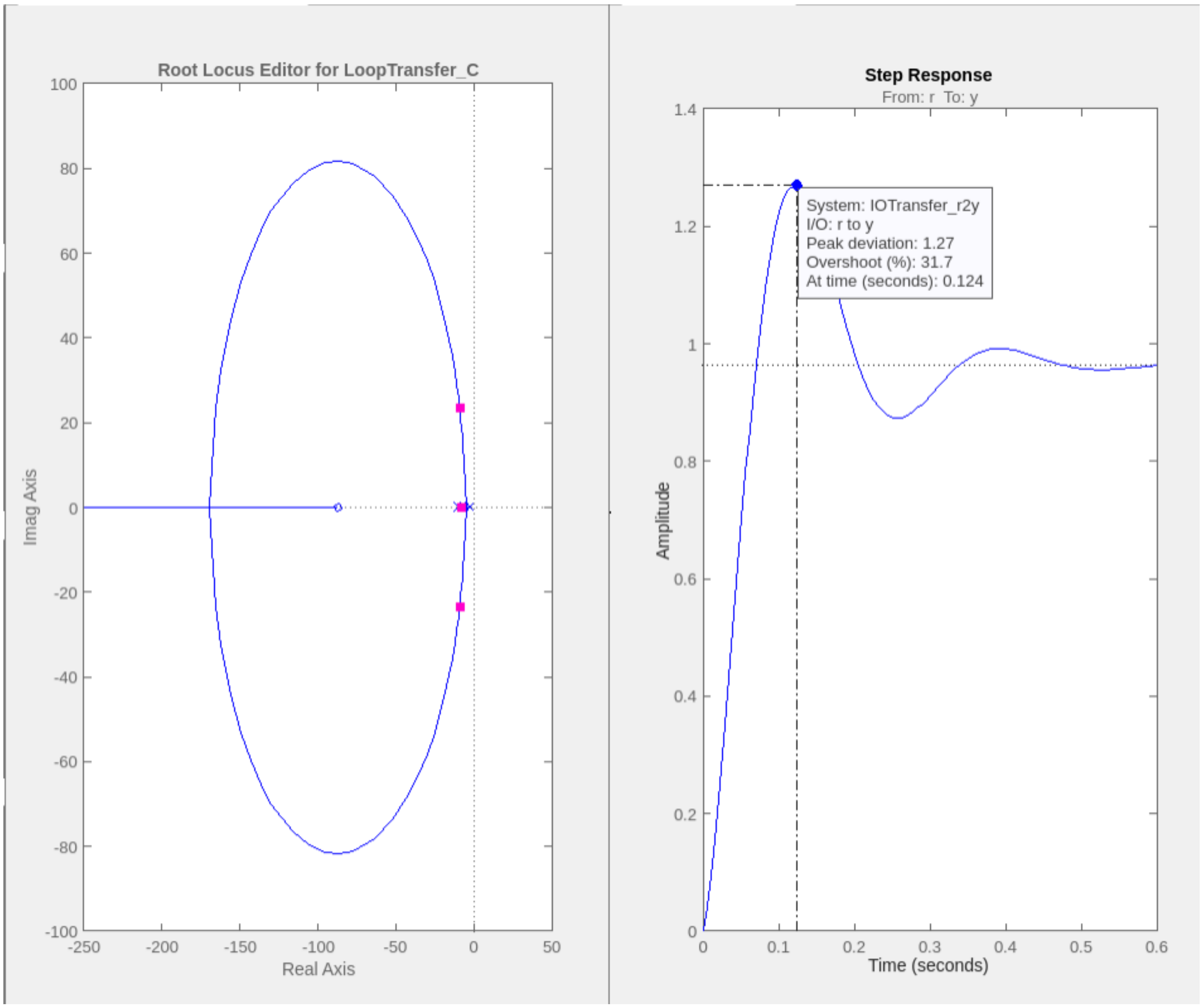


Figure 4. Resulted Root Locus after adding zero at -87.3

Figure 5 onward shows the results of the hand-calculated value. The Control System tool in MATLAB allows user to add poles and zeros directly to the original system in the compensator window as shown in **Figure 5** and **Figure 7**.

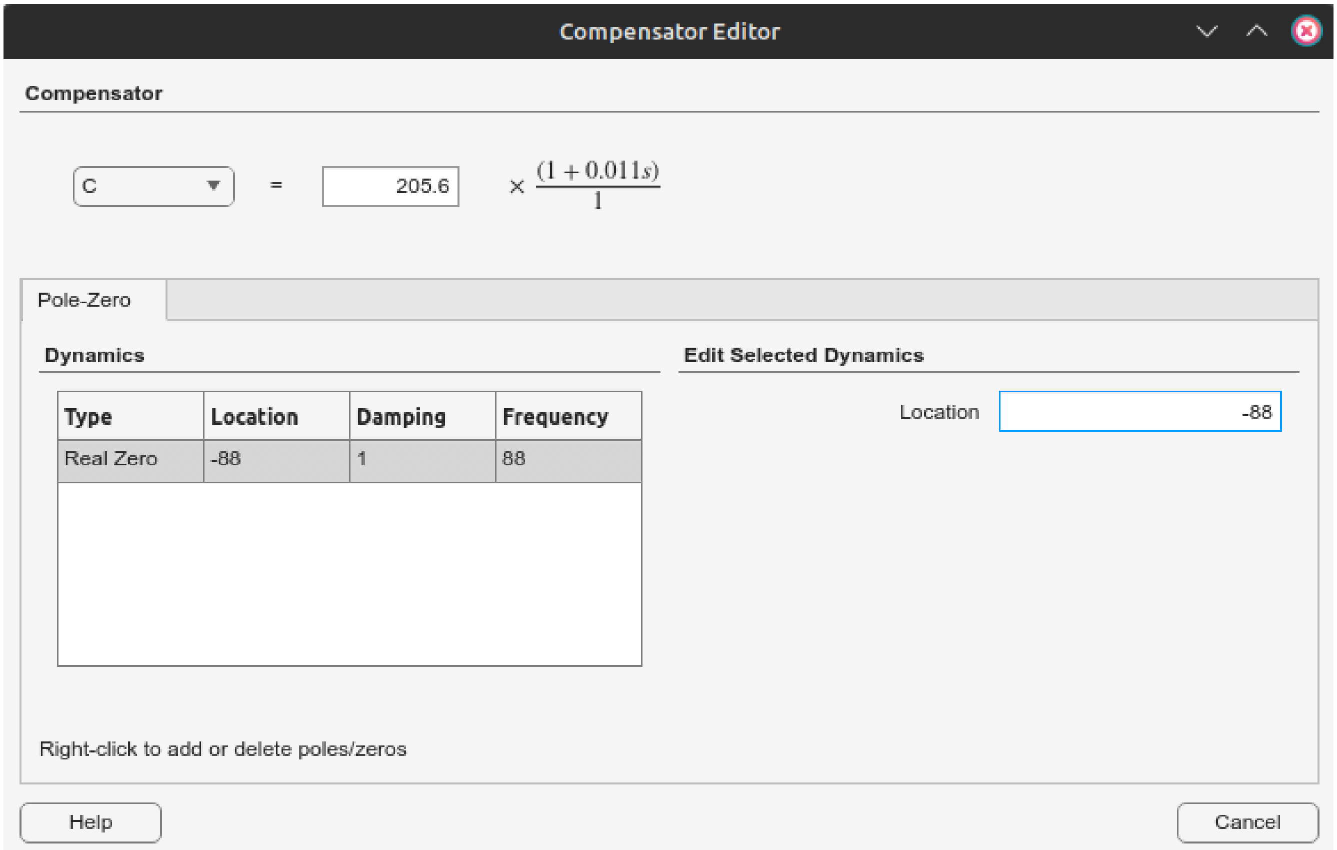


Figure 5. Add real zero at -88

The result after this step is shown in **Figure 6**.

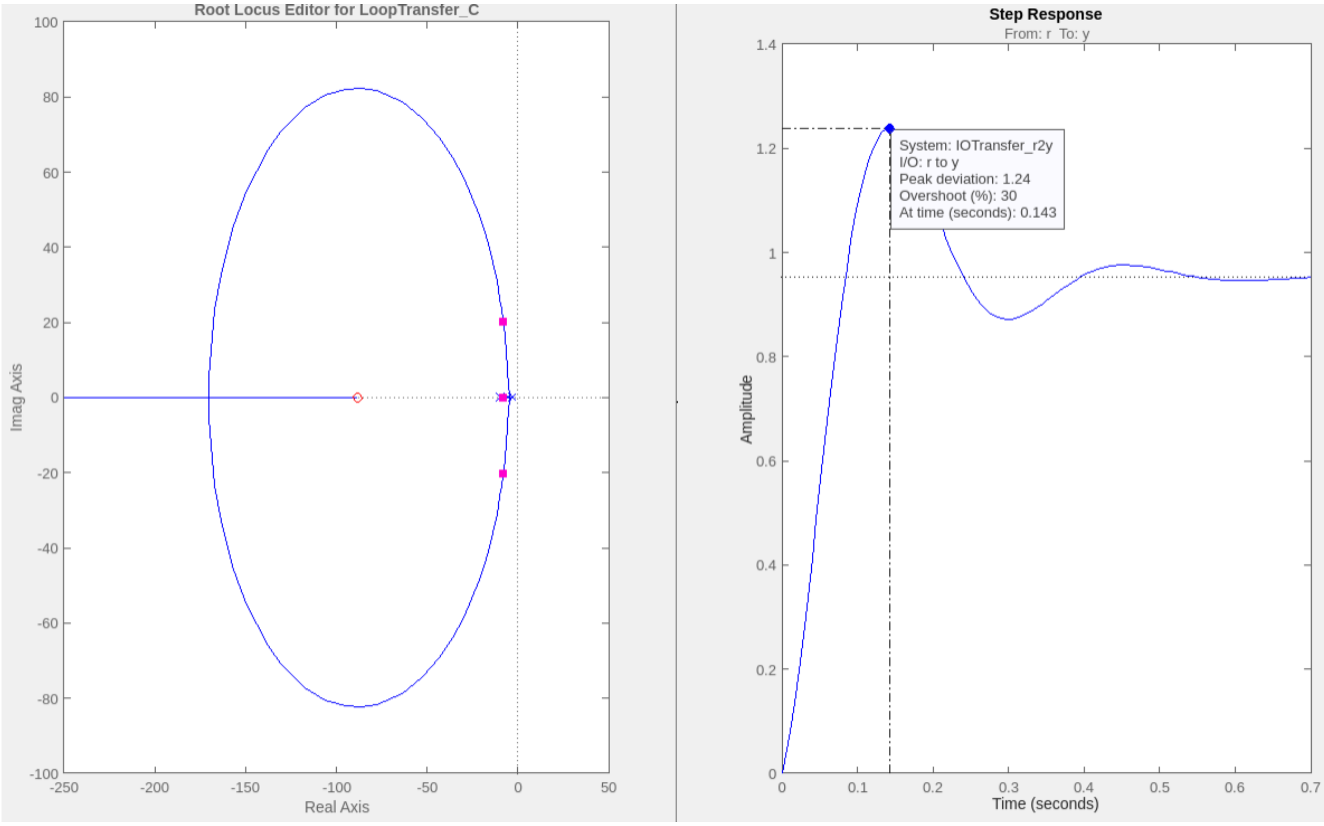


Figure 6. Result of Adding Real Zero at -88

Pole-Zero

Dynamics

Type	Location	Damping	Frequency
Real Zero	-88	1	88
Integrator	0	-1	0
Real Zero	-0.2	1	0.2

Figure 7. Add a real pole at 0 and real zero at -0.2

The result after this step is shown in **Figure 8**

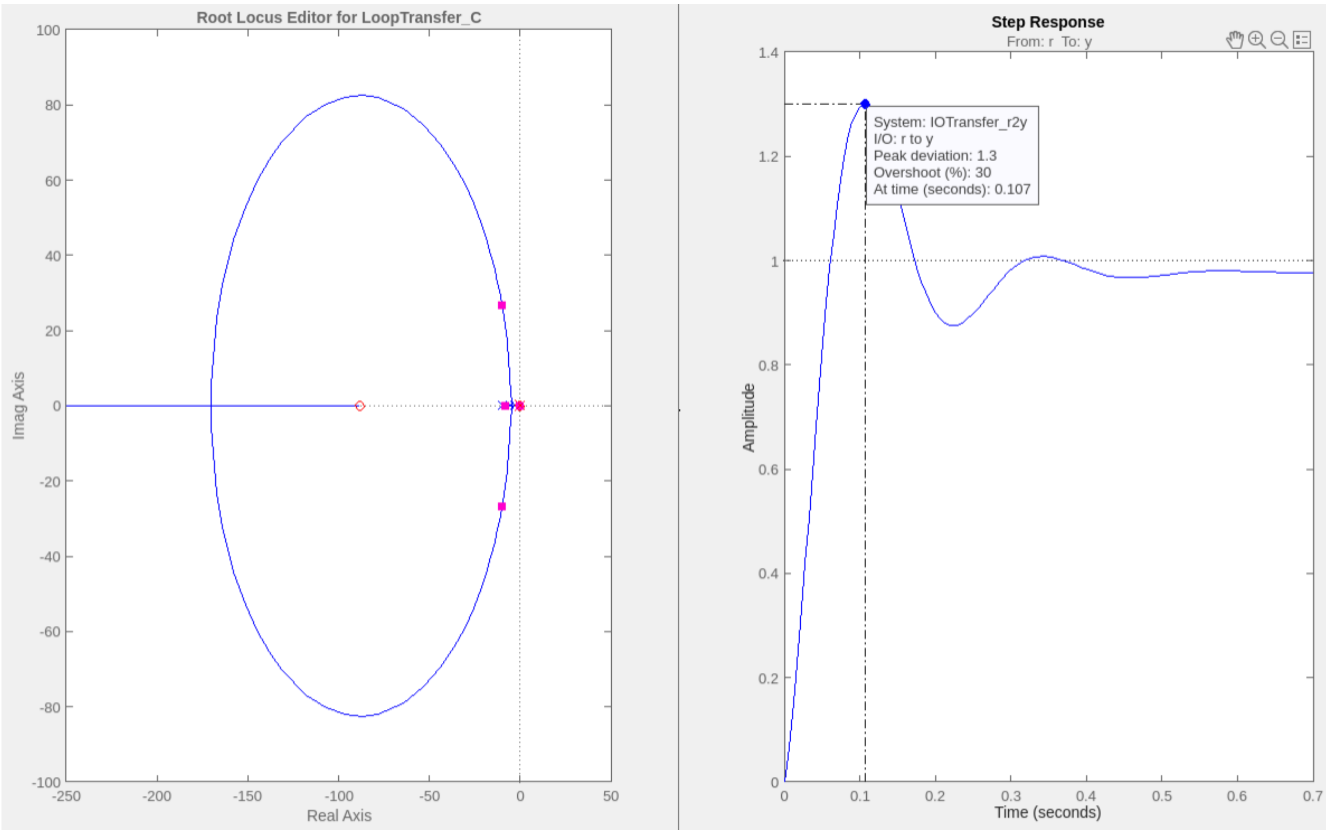


Figure 8. Resulted Root Locus after adding PID compensator

Since adding zero and pole directly in the Control System Tool does not give a precise gain, I proceeded to use the below MATALB code to redraw the root locus in order to determine the accurate gain K for the PID compensator.

MATLAB code:

```
syms s

sys1 = tf([1 8],[1 3]);

sys2 = tf([1 88],[1 6]);

sys3 = tf(1, [1 10]);

sys4 = tf([1 0.2], [1 0]);

sys = sys1*sys2*sys3*sys4;

rlocus(sys)

rltool(sys)
```

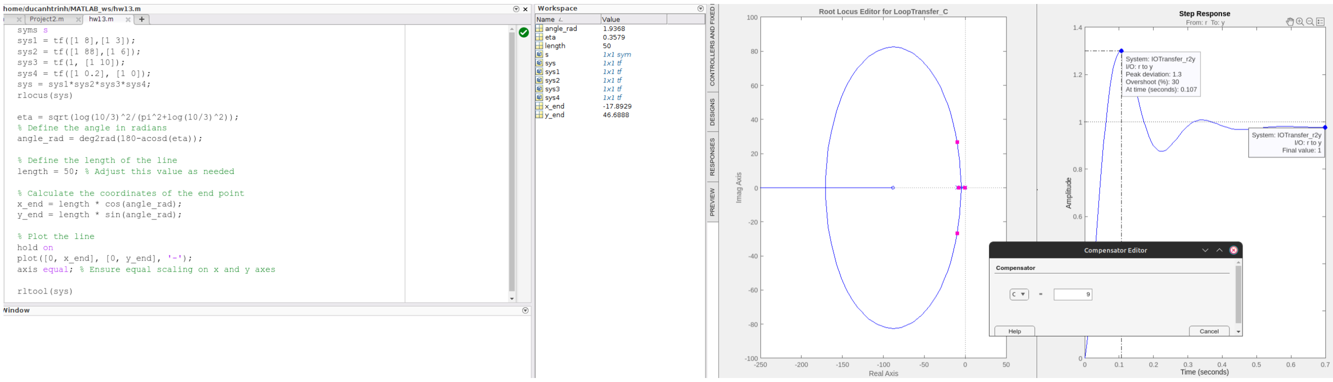


Figure 7. Redraw the Root Locus and Calculate Gain

After the PID (extra poles and zeros) is determined, I calculate the values of Kp, Kd, Ki and use those value to create the PID compensator block in SIMULINK. **Figure 8** shows the block diagram and the response of the compensated system's response, which confirms that the hand calculated values are correct by comparing to the response generated by the MATLAB's control tool.

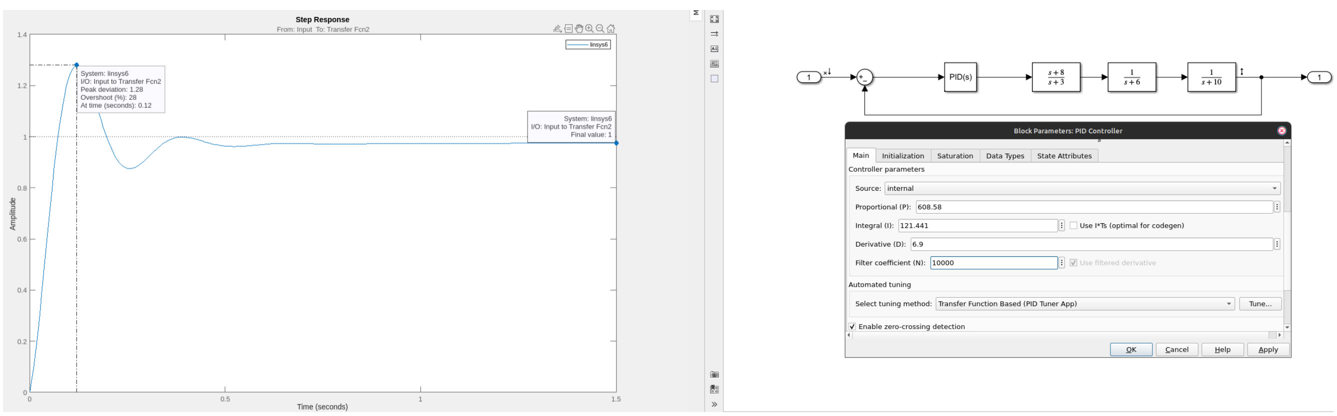


Figure 8. Using Hand-Calculated Value for PID Compensator in SIMULINK

To compare the performance of the compensated hand-calculated values with the MATLAB-generated values, a new PID block diagram is created, and the system's response is shown in **Figure 9**.

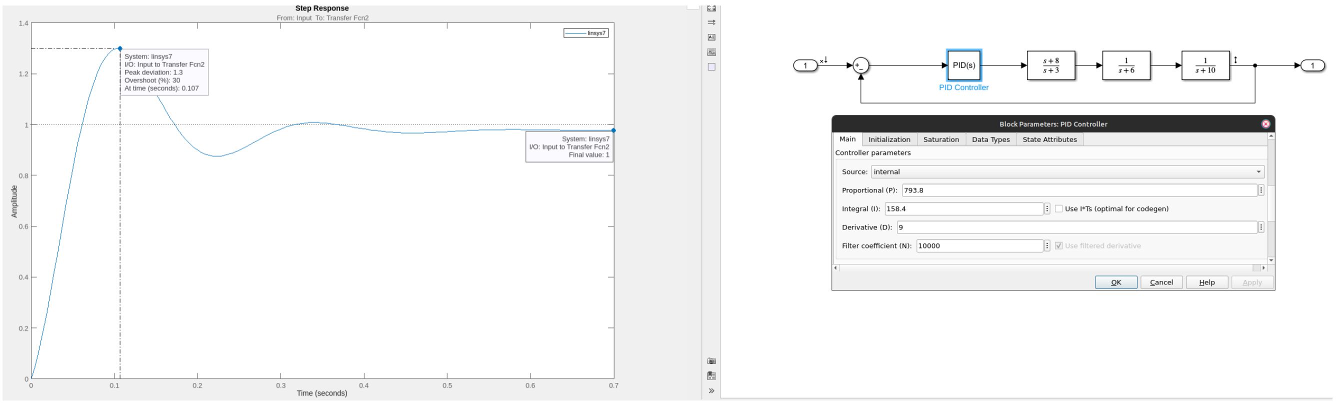


Figure 9. Using Simulated Value for PID Compensator in SIMULINK

Error Analysis

Table 1. Error Analysis for the system

Parameter	Hand Calculated	Simulation	Desired Value	%error of Hand vs Desired Value	%error of MATLAB vs Desired Value
Intersection of the O.S line	$-5.45 \pm 14j$	$-5.45 \pm 14j$	NaN		
Uncompensated peak Time	0.224 s	0.228 s	NaN	1.76	
Compensated peak Time	0.12 s	0.107	0.1344 s	10.71	20.387
Percent Overshoot	28 %	30%	30%	6.67	0

Conclusion

The hand calculated gain is: Kd = 6.9, Kp = 608.58, and Ki = 121.44. After the error analysis process, the conclusion is that the PID value obtained the desired KPI value with acceptable error. The analysis process is also done with the value obtained from the MATLAB program. **Table 1** shows that even though the system of MATLAB gives perfect desired percent overshoot, the system peak time significantly deteriorates.

Example 9.9

Hand Calculation

The value of the PID compensator is obtained from the last part:

Example 9.9

Sunday, April 14, 2024 5:13 AM

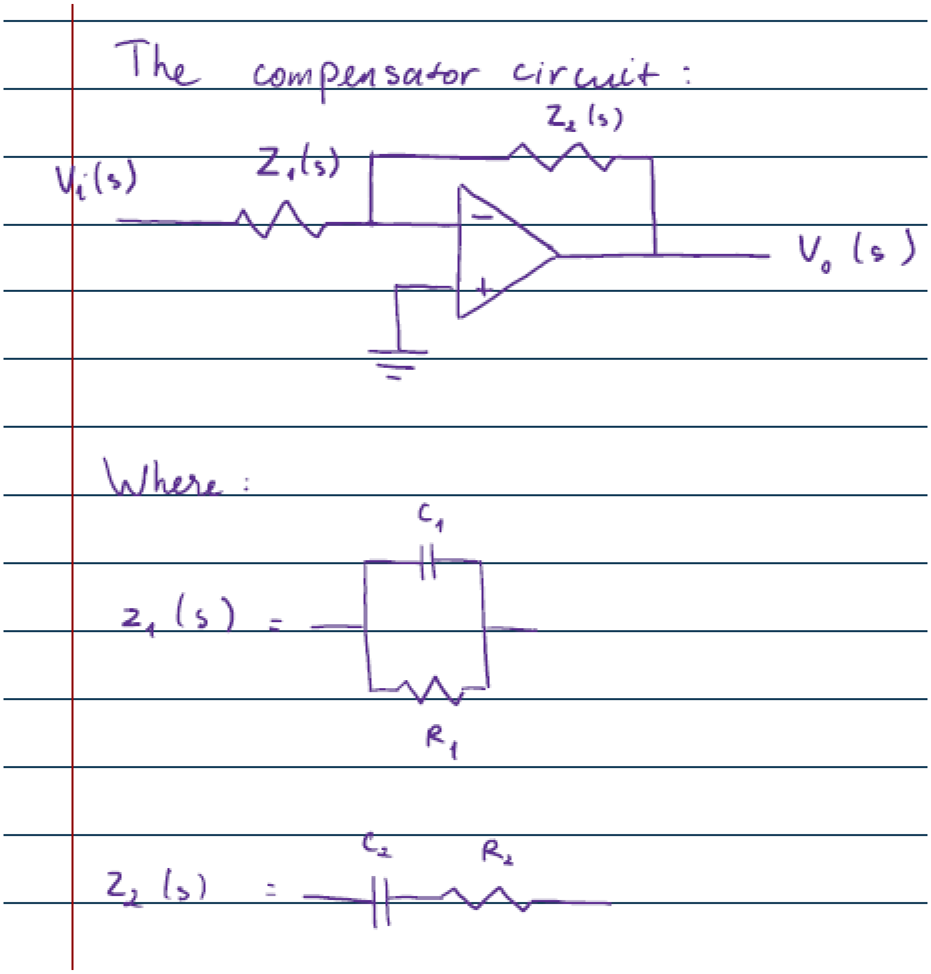
Design PID circuit

$k_p + \frac{k_i}{s} + k_D s$

Using the values $\begin{cases} k_D = 6.9 \\ k_p = 608.58 \\ k_i = 121.44 \end{cases}$

$\Rightarrow G_c(s) = 608.58 + \frac{121.44}{s} + 6.9s$

The PID circuit is taken from the textbook:



The transfer function of the PID circuit is obtained by using the ideal op-amp theory:

$$\Rightarrow G_c(s) = -\frac{Z_2(s)}{Z_1(s)}$$

$$= -\left[\left(\frac{R_2}{\ } + \frac{C_1}{\ }\right) + R_2 C_1 s + \left(\frac{1}{R_1 C_2}\right)\right.$$

$$= -\left[\left(\frac{R_2}{R_1} + \frac{C_1}{C_2}\right) + R_2 C_1 s + \left(\frac{\frac{1}{R_1 C_2}}{s}\right)\right]$$

$$= 608.58 + \frac{121.44}{s} + 6.9s$$

$$\Rightarrow \begin{cases} \dfrac{R_2}{R_1} + \dfrac{C_1}{C_2} = 608.58 \\ R_2 C_1 = 6.9 \\ \dfrac{1}{R_1 C_2} = 121.44 \end{cases}$$

The value of the elements is solved:

Arbitrarily choose $C_2 = 0.1\,\mu F$

$$\Rightarrow R_1 = \frac{1}{121.44\,C_2} = 82.345\,k\Omega$$

$$\Rightarrow \begin{cases} \dfrac{R_2}{82.345\,k\Omega} + \dfrac{C_1}{0.1\,\mu F} = 608.58 \\ R_2 = \dfrac{6.9}{C_1} \end{cases}$$

$$\Rightarrow \begin{cases} C_1 = 60.72\,\mu F \\ R_2 = 113.636\,k\Omega \end{cases}$$

SIMULINK Verification

The object of this part is to design an electrical circuit that behaves as a PID block with the PID values determined in the last step. After calculating the values of the resistors and capacitors required for the design, the diagram is drawn in SIMULINK. **Figure 10** shows the SIMSCAPE Electrical components in SIMULINK model, with the STEP input and scope output to observe the behavior of the circuit.

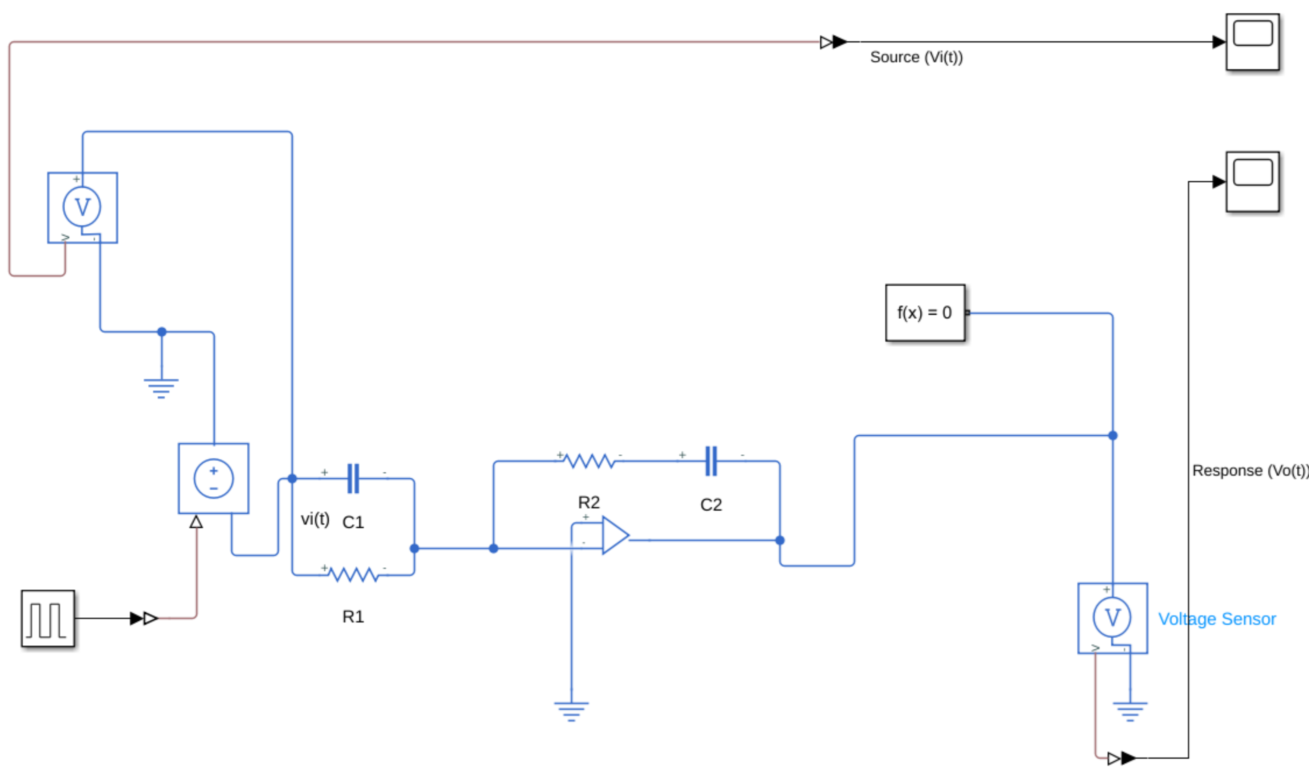


Figure 10. Design of the PID Electrical Circuit

The response of this circuit against Pulse input is recorded in **Figure 11** by changing the input block to PULSE. When compared to the Pulse input response of the PID block in SIMULINK, the response bares high similarity, indicating that the PID block can be replaced by the electrical circuit representation.

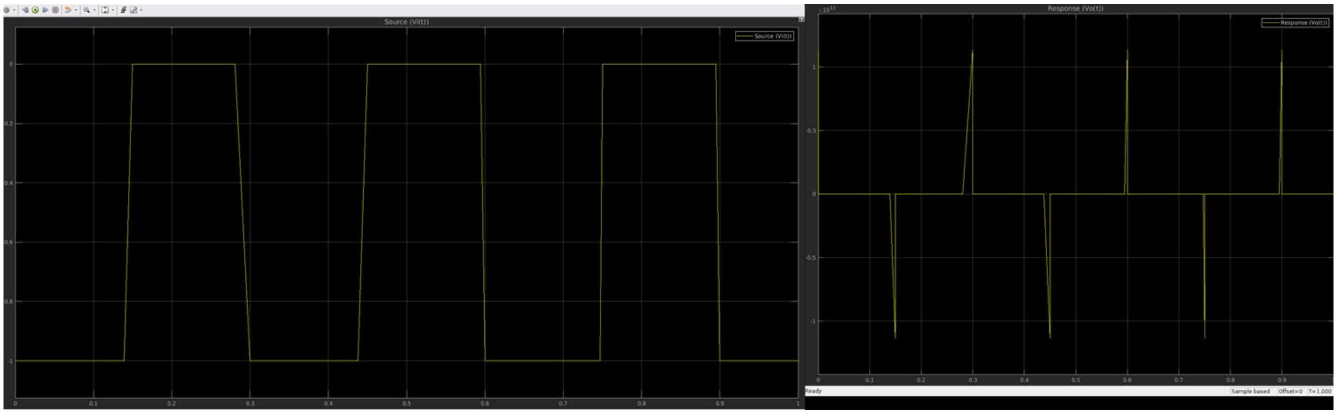


Figure 11. Pulse Input and the Output of the Circuit

After converting the electrical circuit to a subsystem, the electrical model is ready to be used as a block in the block diagram of the interested system to be compensated. The converting process is shown in **Figure 12**, whereas the block diagram is shown in **Figure 13**.

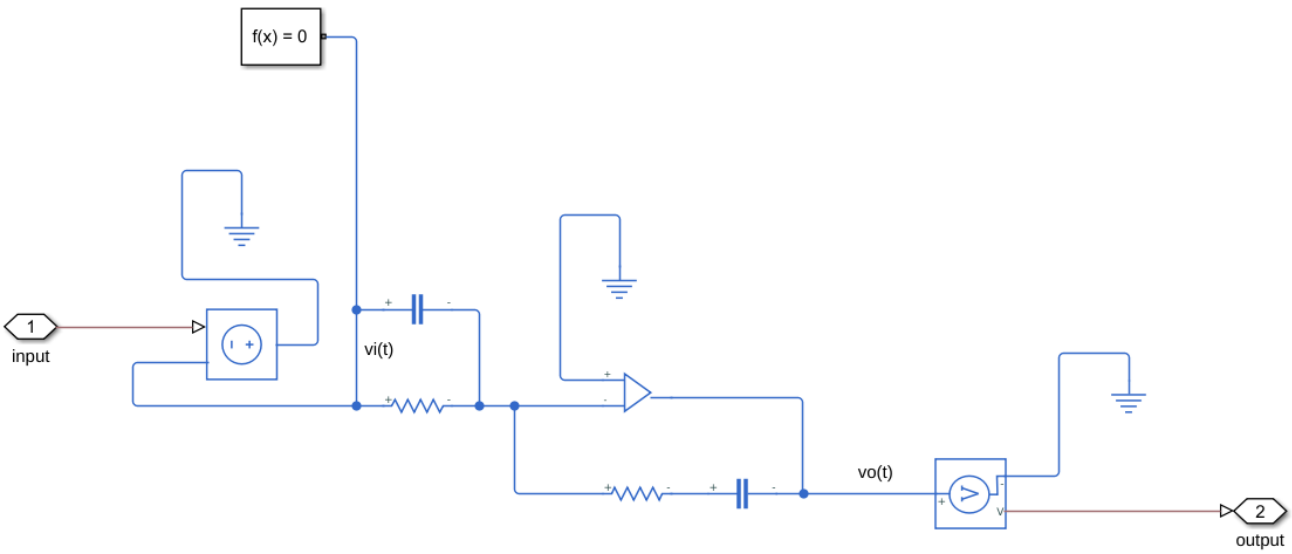


Figure 12. Redesign and Make the Circuit Subsystem

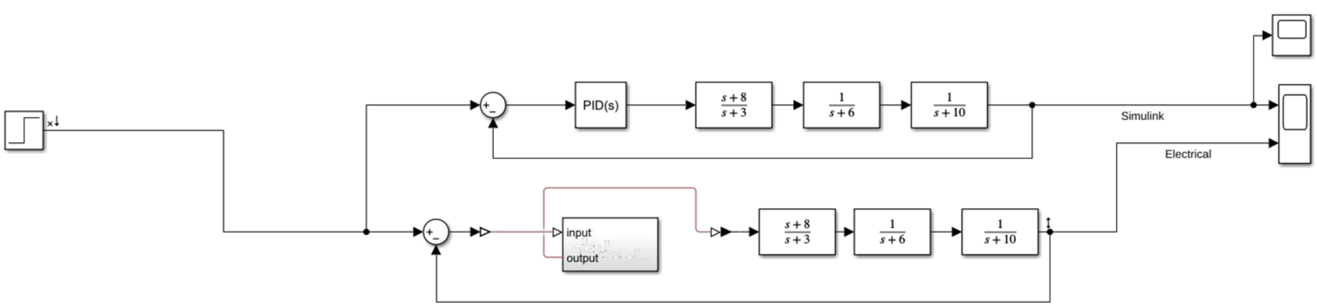


Figure 13. Use the Electrical Circuit (Subsystem) as the Electrical Circuit

The output of **Figure 13** will show the response of the compensated system by PID block in the upper scope and by Electrical Circuit in the bottom scope in **Figure 15**.

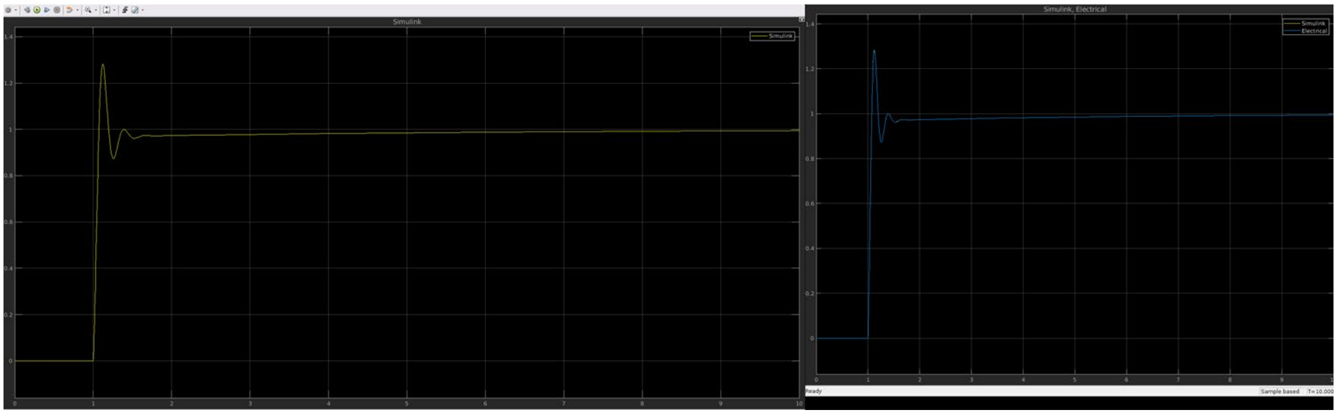


Figure 15. Response of the system Using PID Block (left) and Using Electrical PID (right)

Note: The right of Figure 15 only have one plot visible because both responses plot from PID Simulink Block and from Electrical Circuit Subsystem are identical.

By comparing the response of the compensated systems using the SIMULINK's PID block and the Electrical Circuit PID Subsystem, it is confident that both the system's KPI (shown in **Figure 16**) and waveform are identical. In conclusion, the electrical circuit subsystem is successful in re-implementing the behavior of a PID block, or adding new poles and zeros to the compensated system.

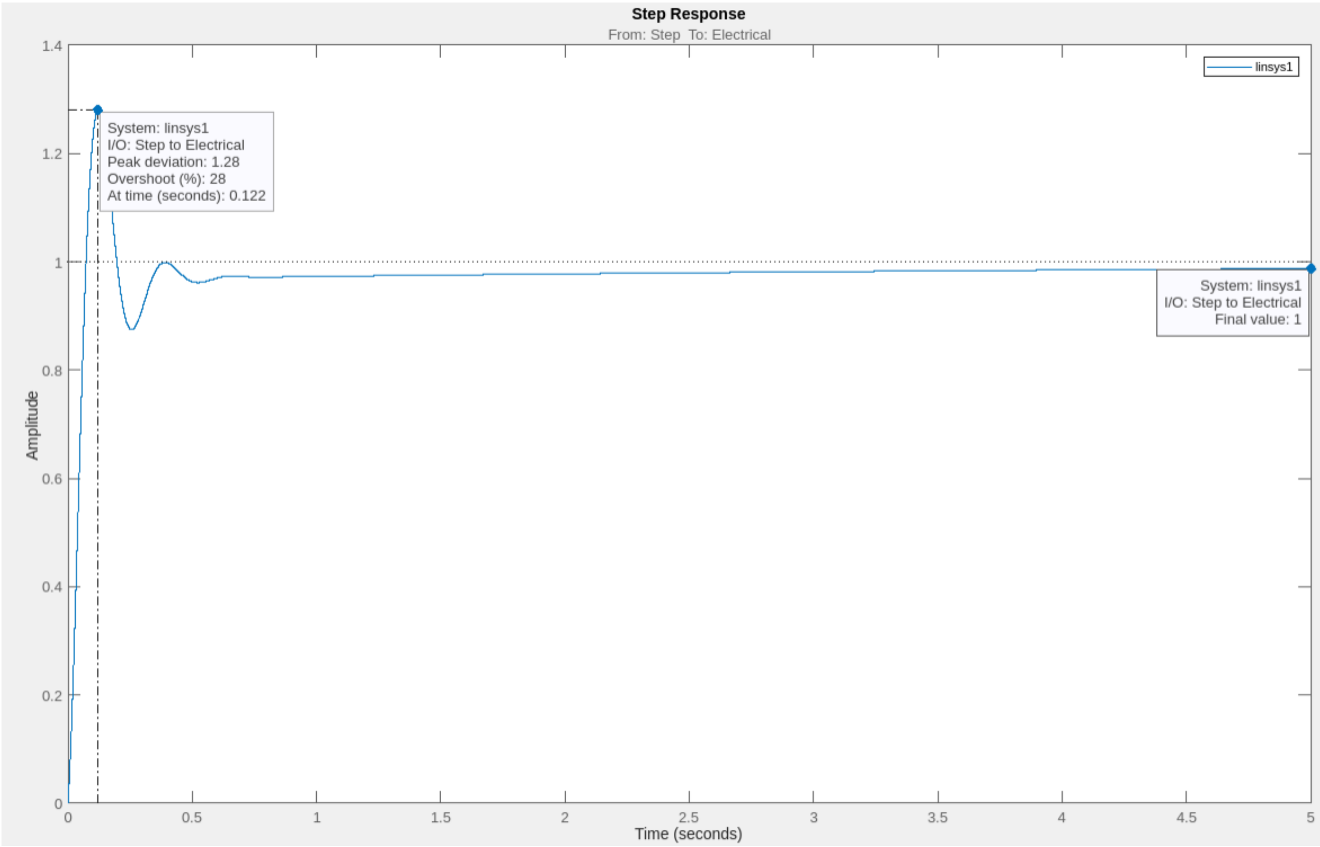


Figure 16. KPI of the Step Response of the system using Electrical PID by Model Linearizer (SIMULINK)
Note: it is necessary to specify time for the Model Linearizer Step Plot if the time is too short

Figure 16 shows the overshoot and the Peak time of the system, which shows identical value for the response using the PID block.

Error Analysis

The peak time and the percent overshoot of the system using the electric circuit subsystem is identical to that of using the PID block in SIMULINK. As long as the accuracy of the electrical circuit behavior on PID, there is zero percent error. In real-world situations, error can occur due to the unideal behavior of the electrical part of the circuit, which will always have some number of percent tolerance.

Conclusion

In this section, an electrical circuit is designed to implement the PID compensator. Both the calculation and the verification are shown, and the design is proved to be both accurate and precise in compensating the system.

Skills Assessment 9.3

Hand calculation

Use second order approximation to calculate Settling Time:

Skills Assessment 93

Monday, April 15, 2024 4:07 AM

$$G(s) = \frac{k}{s(s+7)}$$

a. T_s ?

$$\Delta(s) = 1 + \frac{k}{s(s+7)}$$

$$= \frac{s^2 + 7s + k}{s(s+7)}$$

$$\Rightarrow s^2 + 7s + k = 0$$

Use second order form:

$$\Delta(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$$

$$\Rightarrow \begin{cases} 2\zeta\omega_n = 7 & (1) \\ \omega_n = \sqrt{k} & (k > 0) \end{cases}$$

* The system is operating at 20% O.S

$$O.S = e^{-\zeta\pi/\sqrt{1-\zeta^2}} = 0.2$$

$$\Rightarrow \zeta^2\pi^2 = (1-\zeta^2)\,2.5903$$

$$\Rightarrow \zeta = 0.4559 \quad (2)$$

From (1) and (2) $\Rightarrow \omega_n = \frac{7}{2\zeta} = 7.676$

$$\Rightarrow k = \omega_n^2 = 58.925$$

$$T_s = \frac{4}{\zeta\omega_n} = \frac{4}{(7/2)} = 1.1429$$

Calculate ramp steady state error using limit as s approach zero:

b) $e_{ramp,ss}$?

$$E'(s) = R(s) - Y(s)$$

$$= R(s) - G'(s)R(s)$$

$$= R(s)\left[1 - \frac{G(s)}{1+G(s)}\right]$$

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

Ramp input $\Rightarrow R(s) = \frac{1}{s^2}$

$$e_{ramp,ss} = \lim_{s\to 0} sE'(s)$$

$$= \lim_{s\to 0} s.\frac{1}{s^2}\frac{1}{\left(1+\frac{K}{s(s+7)}\right)}$$

$$= \lim_{s\to 0} \frac{1}{\left(s+\frac{K}{s+7}\right)}$$

$$= \frac{7}{K} = \frac{7}{58.925} = 0.1188$$

Calculate the desire settling time:

c) Lead - Lag compensator for:

$$\begin{cases} T_{s,new} = \frac{1}{2} T_s \\ e_{ramp,ss,new} = \frac{1}{10} e_{ramp,ss} \\ \text{Lead zero at } -3 \end{cases}$$

$$\frac{4}{\zeta \omega_{n,new}} = T_{s,new} = \frac{1}{2} T_s = \frac{1}{2} \frac{4}{\zeta \omega_n}$$

Find the location of the desired dominant poles:

$$\Rightarrow \zeta w_{n,new} = 2\zeta w_n = 2\left(\frac{7}{2}\right) = 7$$

$$\Rightarrow w_{n,new} = \frac{7}{\zeta} = 15.3542 \Big|_{\zeta = 0.455}$$

$$\Rightarrow w_{n,new}\sqrt{(1-\zeta^2)} = 13.6728$$

$$\Rightarrow \text{Dominant poles} : -\zeta w_{n,new} \pm w_{n,new}\sqrt{(1-\zeta^2)}$$

$$= -7 \pm 13.6728j = s_1$$

Find the location of the zero using the angle criterion and trigonometry:

1) Find location of a zero and pole for lead compensator :

$z_{lead} = -3$ (arbitrarily chosen)

$\angle s_1 = 117.111°$

$\angle s_1+7 = 90°$

$\angle s_1+3 = 106.310°$

. angle criterion :

$$106.31 - (117.111 + 90 + \theta) = -180°$$

$$\Rightarrow \quad -100.801 - \theta = -180°$$

$$\Rightarrow \quad \theta = 79.199$$

$$\tan 79.199° = \frac{13.6278}{p_{lead} - 7}$$

$$\Rightarrow p_{lead} = \frac{13.6278}{\tan 79.199} + 7$$

$$= 9.60$$

Calculate the Root Locus Parameter:

* Root locus :

poles : 0 , -7, -9.6

zeros : -3

$n - m = 2$

$$\sigma_c = \frac{-7-9.6+3}{2} = -6.8$$

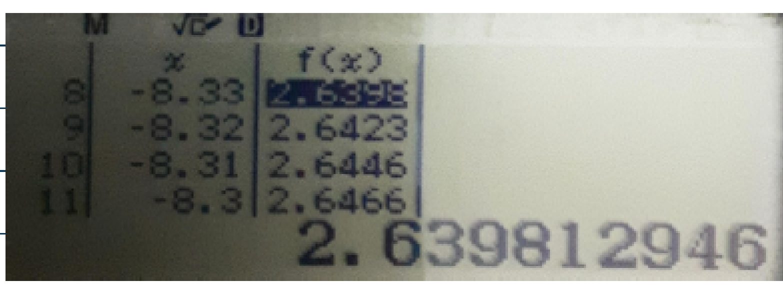
Asymptotes:

$$\varphi = \frac{(2h+1)180}{2}$$

$$\begin{cases} h = 0, & \varphi = 90^\circ \\ h = -1, & \varphi = -90^\circ \end{cases}$$

. Find $\max\left(K = \left|\frac{s(s+7)(s+9.6)}{s+3}\right|\right)$

for $-9.6 < s < -7$ (either use table or $\frac{\partial K}{\partial s}$)

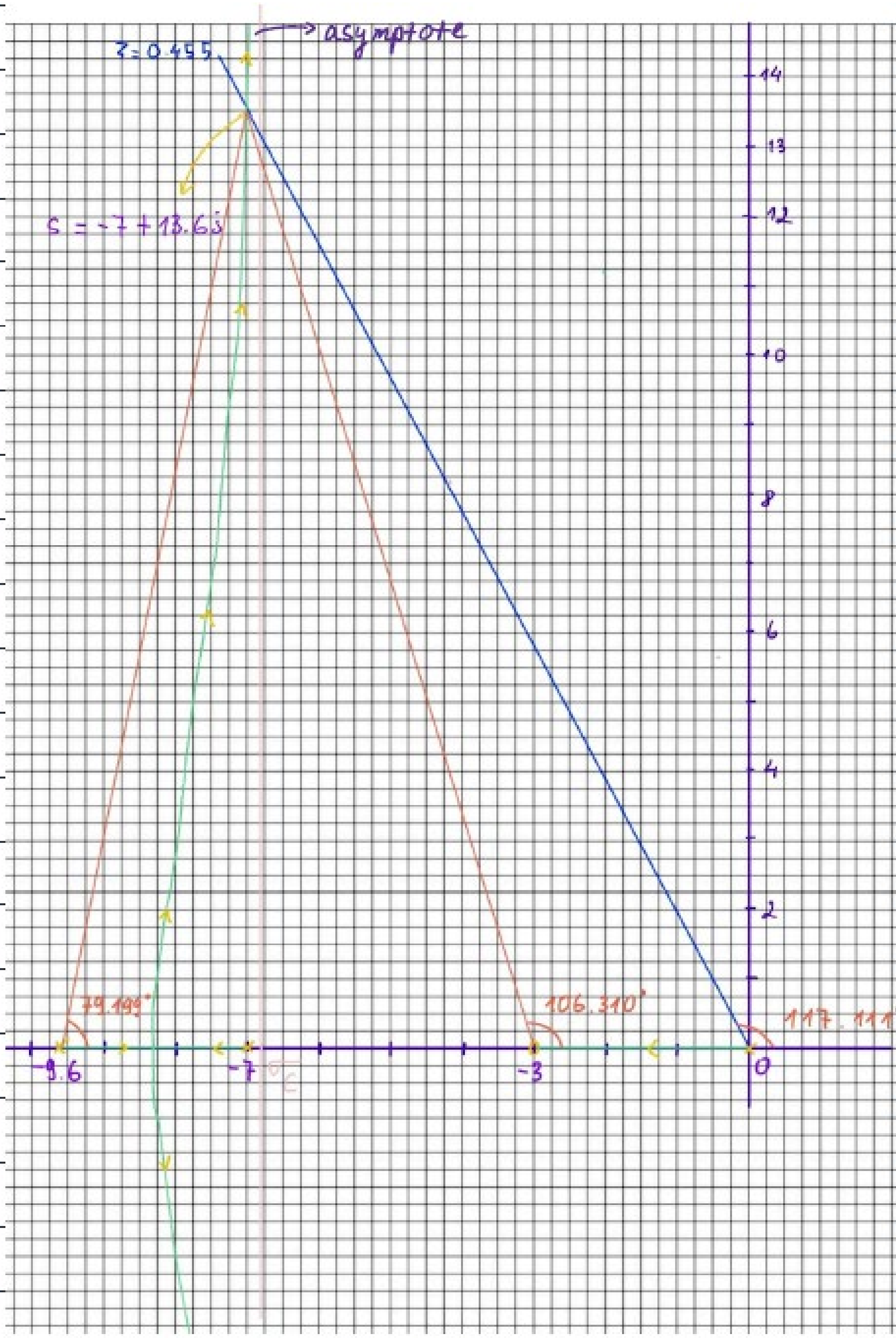


	x	f(x)
8	-8.33	2.6398
9	-8.32	2.6423
10	-8.31	2.6446
11	-8.3	2.6466

2.639812946

→ break out point $s_{br} = -8.3$

Sketch the Root Locus and calculate the Gain at the dominant poles:



$$K = \left| \frac{s(s+7)(s+9.6)}{s+3} \right| \text{ for } s = -7 + 13.6728j$$

$\Rightarrow K = 205.184$

Find the ramp steady state error due to adding the leading compensator:

2) Lag Compensator :

$$G_{Lead}(s) = 205.184 \frac{s+3}{s(s+7)(s+9.6)}$$

$$\Rightarrow e_{ramp,ss} = \lim_{s \to 0} s \cdot \frac{1}{s^2} \frac{1}{\left[1 + \frac{205.184(s+3)}{s(s+7)(s+9.6)}\right]}$$

$$= \lim_{s \to 0} \frac{1}{\left[s + \frac{205.184(s+3)}{(s+7)(s+9.6)}\right]}$$

$$= 0.10917$$

Determine how much error the lag compensator need to reduce:

By using lead compensator, $e_{ss,new} = \frac{0.10917}{0.1188} e_{uncompensated}$

$= 0.9189\, e_{uncompensated}$

$\Rightarrow$ lag compensator need to decrease e_{ss} $\frac{0.10917}{0.01188}$

$= 9.1894$ times

Design the lag compensator based on the difference value:

Arbitrarily choose a pole $p_{lag} = 0.01$

$\Rightarrow z_{lag} = 9.1894(0.01) = 0.0919$

$$\Rightarrow e_{ss,lead\text{-}lag} = \lim_{s \to 0} \frac{1}{s + \frac{205.184(s+3)(s+0.0919)}{(s+7)(s+9.6)(s+0.01)}}$$

$= 0.01188$

MATLAB verification and code

MATLAB code:

```
s = tf('s');
sys1 = 1/(s*(s+7));
rltool(sys1) %Control System Tool
rlocus(sys) %Draw Root Locus
K = 58.81;
G1 = feedback(K*sys1, 1);
step(G1) % Step Response
hold on

p = -7+13.6728i;
K = abs(p*(p+7)*(p+9.6)/p+3);
sys2 = (s+3)/(s+9.6) * sys1;
rltool(sys2)
G2 = feedback(K*sys2, 1);
step(G2)
sys3 = (s+0.0919)/(s+0.01)*sys2;
G3 = feedback(K*sys3, 1);
```

The first step is to determine the settling time and percent overshoot of the system. In the handwritten calculation, it is not necessary to draw a root locus for this system since it is a second order system. Therefore, the formula for settling time and percent overshoot can be applied directly without the complex of the approximation process. **Figure 17** shows the root locus and the KPI of the system to prove this statement.

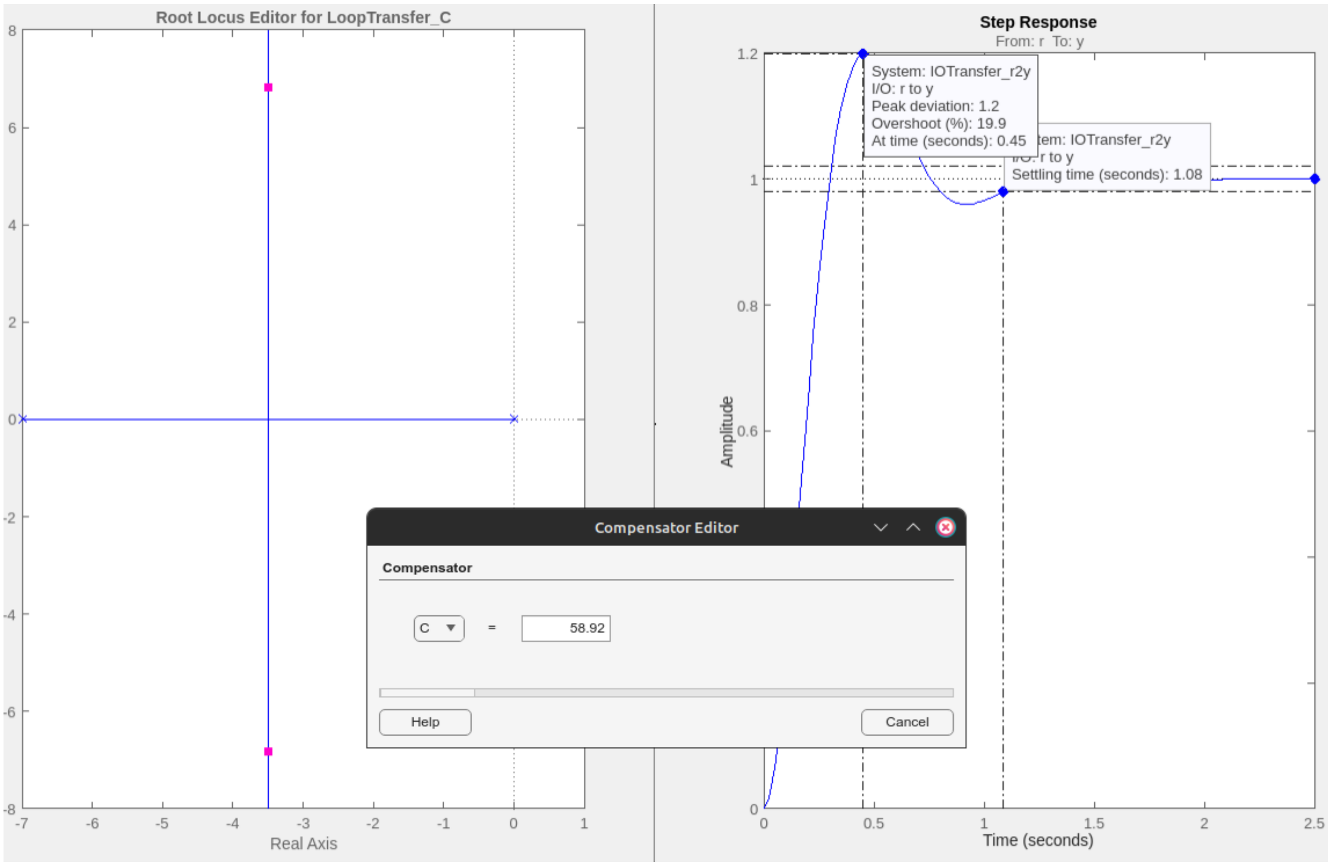


Figure 17. Root Locus and Step Response

After obtaining the KPI values, the ramp steady state error is calculated. The MATLAB code and the results is shown **Figure 18**.

```
tfinal = 10000;
[y, t] = step(G1/s, tfinal);
steady_state = y(end) - t(end)

t = 0:0.01:tfinal;
ramp = t;
[y,t] = lsim(G1,ramp,t);
steady_state = y(end) - t(end)
```

Command Window

```
>> project93

steady_state =

   -0.1190


steady_state =

   -0.1190
```

Figure 18. Ramp Steady State Error

With the uncompensated system's KPI value, the desired KPI value is calculated, which leads to the desired dominant poles that we wish the root locus to pass through.
The Lead Compensator is designed so that the desired dominant poles lie on the compensated system's Root Locus.
Finally, the Root Locus of the the Lead Compensated System can be drawn. **Figure 19** shows the Root Locus, the KPI, and the Gain at the desired dominant poles.

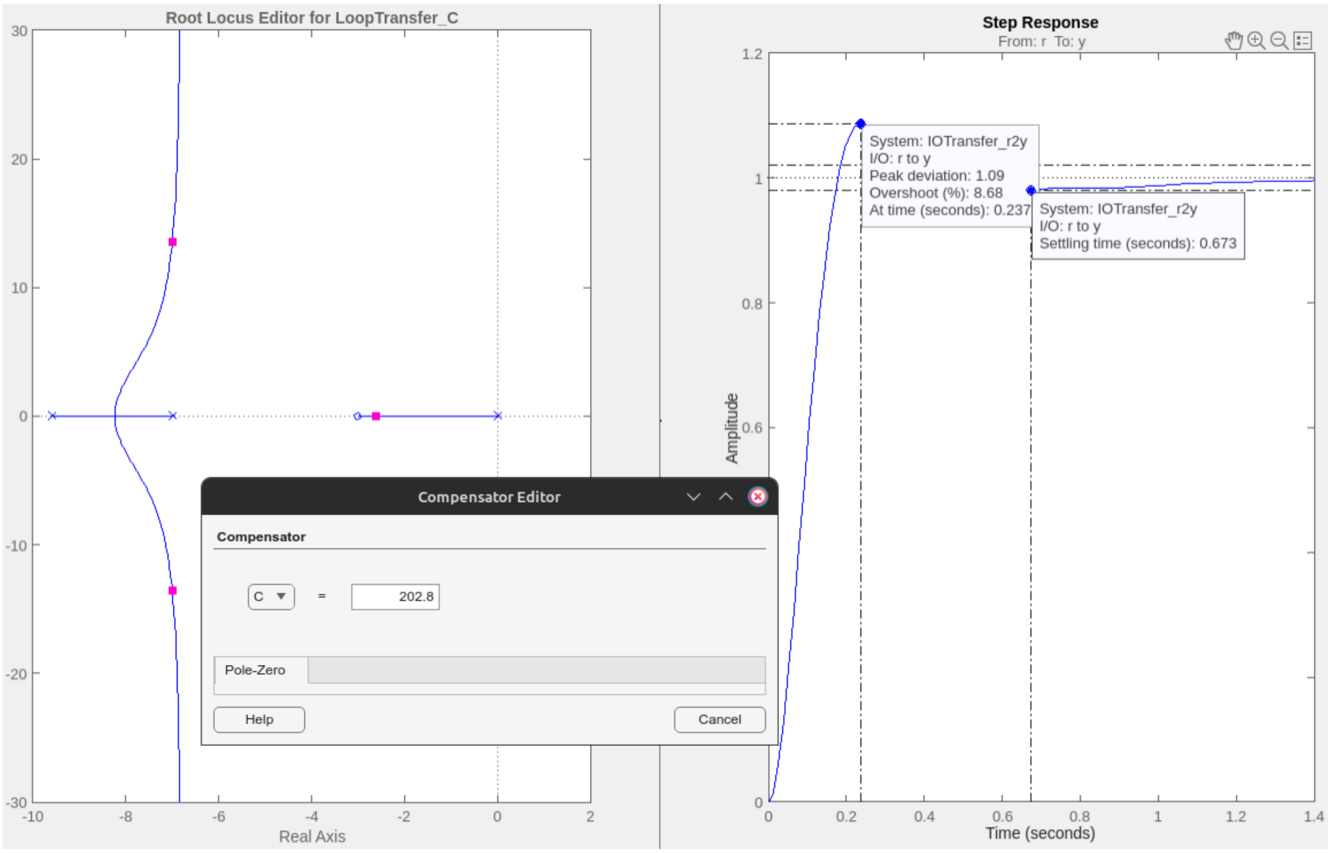


Figure 19. Response for the calculated Dominant Poles (-7+-13.6j)

Since the problem requires to place the lead zero at -3, the second order approximation will largely deviate from the actual system response since the number of poles and zeros increase. Normally, it is recommended to add the arbitrarily lead zero at the system's pole so that the pole is canceled, which will retain the number of poles and zeros of the system after the lead compensator is designed. Therefore, **Figure 20** and **Figure 21** show the responses of the system for the desired overshoot and settling time respectively in order to show the effect of not adding the lead zero at the recommended location.

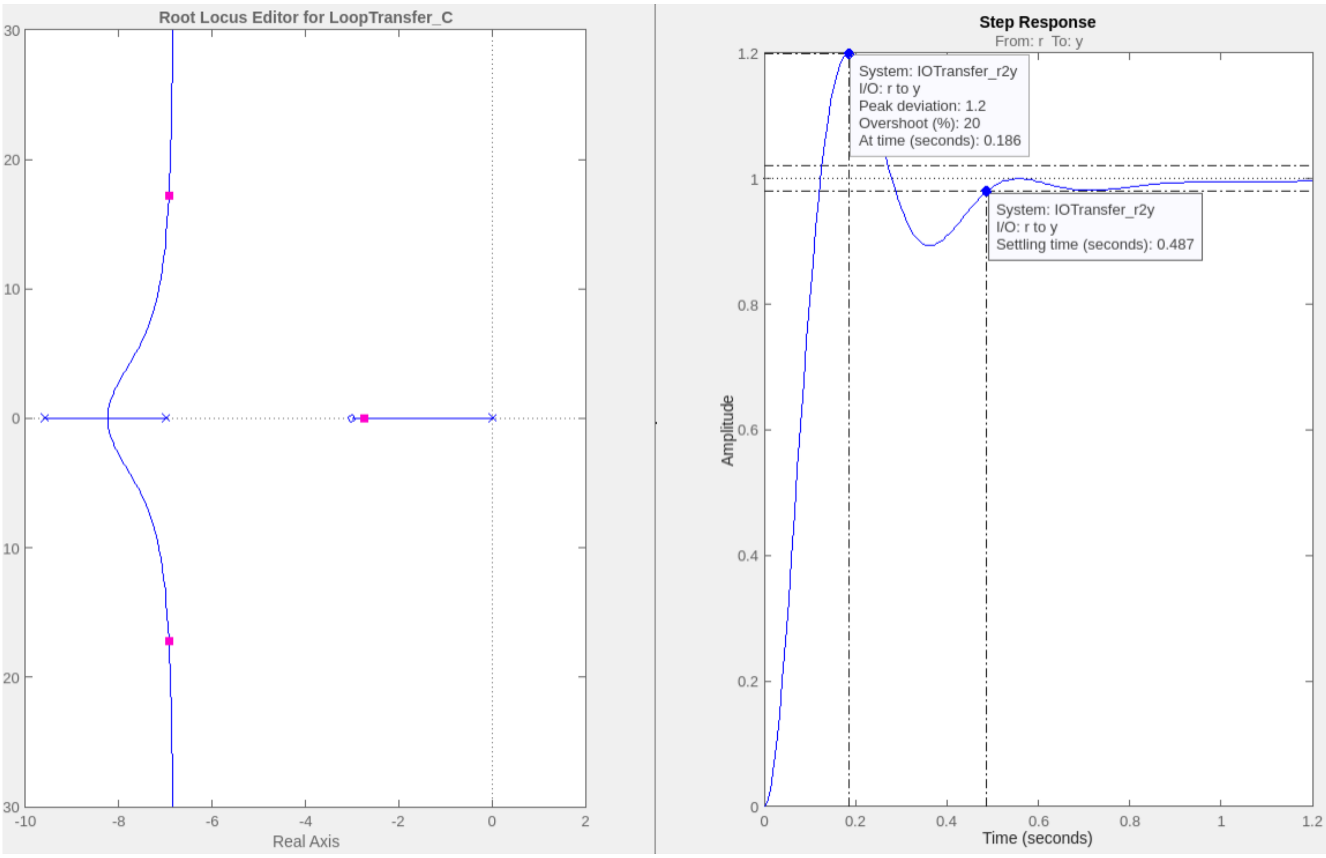


Figure 20. Response for 20% O.S (-6.93+-17.2j)

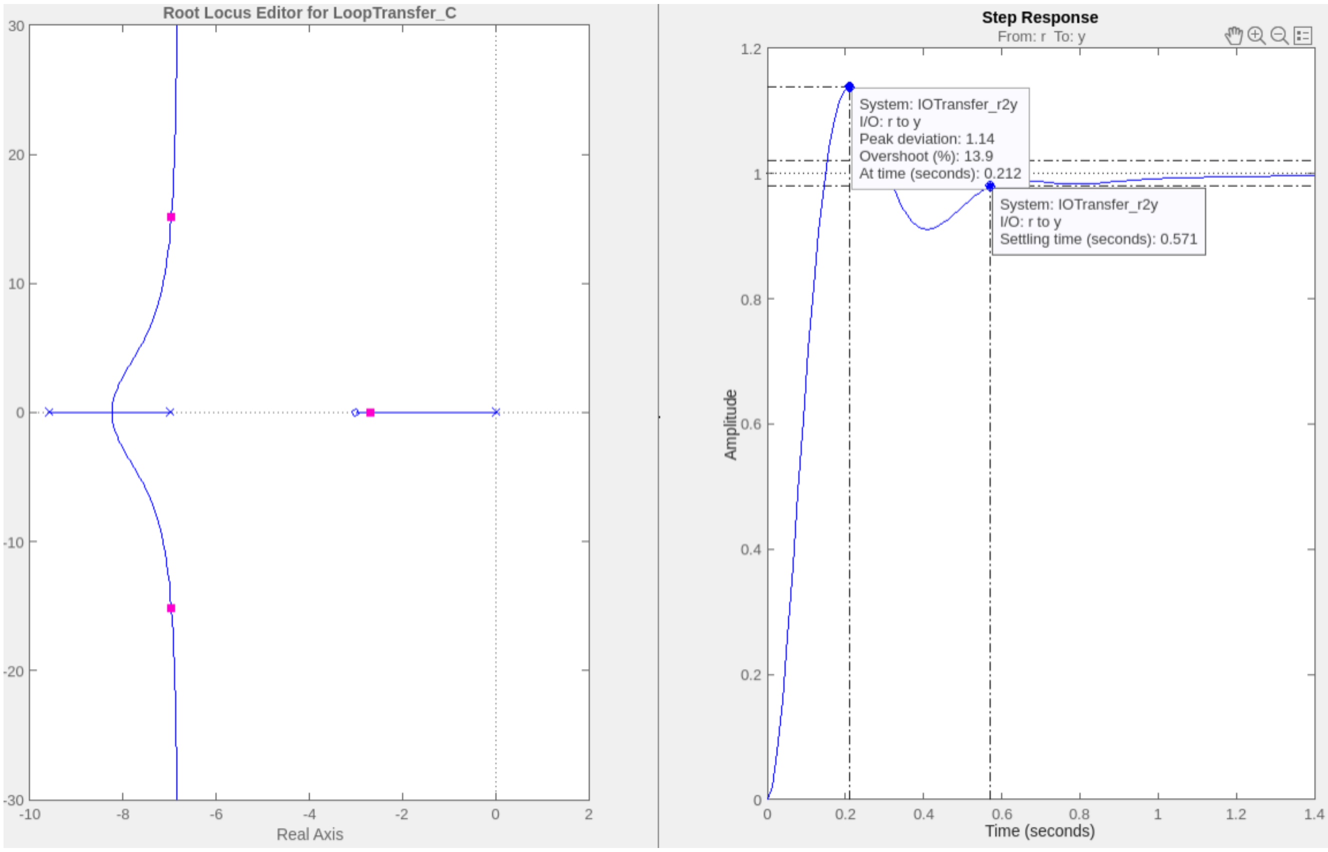


Figure 21. Response for 0.571 Settling Time

After designing the Lead Compensator, the ramp steady-state error is changed, which need to be calculated. Finally, the Lag Compensator can be designed to account for the difference between the ramp steady-state error and the desired steady-state error.

Figure 22 shows the MATLAB code and two methods to measure the ramp steady-state error, which return the same value.

Matlab Code:

```
tfinal = 10000;
[y, t] = step(G3/s, tfinal);
steady_state = y(end) - t(end)

t = 0:0.01:tfinal;
ramp = t;
[y,t] = lsim(G3,ramp,t);
steady_state = y(end) - t(end)
```

```
tfinal = 10000;
[y, t] = step(G3/s, tfinal);
steady_state = y(end) - t(end)

t = 0:0.01:tfinal;
ramp = t;
[y,t] = lsim(G3,ramp,t);
steady_state = y(end) - t(end)
```

Command Window

```
>> project93

steady_state =

   -0.0119


steady_state =

   -0.0119
```

Figure 22. Ramp Steady State Error for after Lead-Lag Compensated

Error Analysis

Table 2. Error Analysis for lead-lag system

Parameter	Calculated/ Desired Value	Simulation Value	%Error
Old Settling Time	1.1429 s	1.08	5.82
New Settling Time	0.571 s	0.673	17.863
Old O.S	20	19.9	0.5
New O.S	20	8.68	56.6
Final ramp steady state error	0.01188	0.0119	0.17

Conclusion

In conclusion, the process of designing a lead compensator for a second-order system involves determining the settling time and percent overshoot initially. Root locus analysis is skipped for second-order systems due to their simplicity, allowing direct application of formulas for settling time and percent overshoot. The resulting Key Performance Indicator (KPI) values guide the design process.

Following this, the ramp steady-state error is calculated. Using the uncompensated system's KPI value, desired KPI values are derived, leading to the identification of desired dominant poles. The lead compensator is then crafted to ensure these poles align with the system's root locus.

However, if the lead zero placement is placed at different location from a poles, as mandated by the problem, the second-order approximation may significantly differ from the actual system response due to alterations in poles and zeros number, which is clearly shown in the larger percent error in **Table 1**.

Figures 20 and **21** illustrate the consequences of not adhering to the recommended lead zero position, illustrate that to obtain one specific KPI, the other KPI deteriorates significantly.

Finally, after designing the lead compensator, adjustments to the ramp steady-state error necessitate recalculation. Finally, the lag compensator is introduced to address any disparities between the actual and desired steady-state errors. Because the steady-state errors can be closely related to the position of the lead compensator, the desired steady-state error is achieved with very small error, as illustrated in **Table 1**.