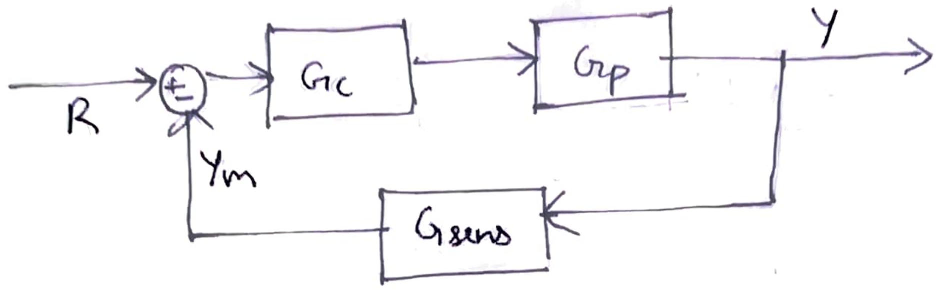
BY : S. VISHAL
CH18B020

CH3050 ASSIGNMENT- 5

(1) a)



$$(R - G_{sens} Y)(G_c G_p) = Y$$

$$\Rightarrow \frac{Y}{R} = \frac{G_c G_p}{1 + G_c (G_{sens} G_p)}$$

where $G_c = K_c$.

$\therefore$ C.E. : $1 + K_c G_{sens} G_p = 0$

$$\Rightarrow 1 + K_c \left[\frac{s^2 - 4s + 8}{(s)(s+1)(s+3)(s+10)} \right] = 0$$

poles : $\boxed{0, -1, -3, -10}$

zeroes : $\frac{4 \pm \sqrt{16 - 32}}{2} = \boxed{2 \pm 2j}$

i) Asymptotes angles.

totally $p - z = 4 - 2 = 2$ asymptotes

$$\theta_1 = \frac{2-1}{2} \pi = \boxed{\frac{\pi}{2}}$$

$$\theta_2 = \frac{2(2) - 1}{2} \pi = \boxed{\frac{3\pi}{2}}$$

ii) Centroid

$$\sigma = \frac{\sum P_i - \sum z_i}{P - Z}$$

$$= \frac{-14 - 4}{2} = -9$$

$\therefore$ centroid, $\boxed{\sigma = -9 + 0j}$

iii) Angles of arrival. $\left[180 + \sum_{\forall i} \angle z_j - P_i - \sum_{j \neq k} \angle (z_j - z_k)\right]$

The root locus 'arrives' at zeroes.

$$\theta_{z_1} : 180 + \tan^{-1}\left(\frac{2}{2}\right) + \tan^{-1}\left(\frac{2}{2-(-1)}\right)$$

$$+ \tan^{-1}\left(\frac{2}{2-(-3)}\right) + \tan^{-1}\left(\frac{2}{2-(-10)}\right)$$

$$- \tan^{-1}\left(\frac{2-(-2)}{2-2}\right)$$

$\Rightarrow \boxed{\theta_{z_1} = 199.954^\circ}$

$$\theta_{z_2} = 180^\circ + \tan^{-1}\left(\frac{2}{-2}\right) + \tan^{-1}\left(\frac{2}{-3}\right)$$

$$+ \tan\left(\frac{2}{-5}\right) + \tan^{-1}\left(\frac{-2}{12}\right)$$

$$- \tan^{-1}\left(\frac{-2 - 2}{2 - 2}\right)$$

$\Rightarrow \boxed{\theta_{z_2} = 160.046^\circ}$

iv) Break-in points / Break-away points.

$$\beta = \frac{-(s)(s+1)(s+3)(s+10)}{s^2 - 4s + 8}$$

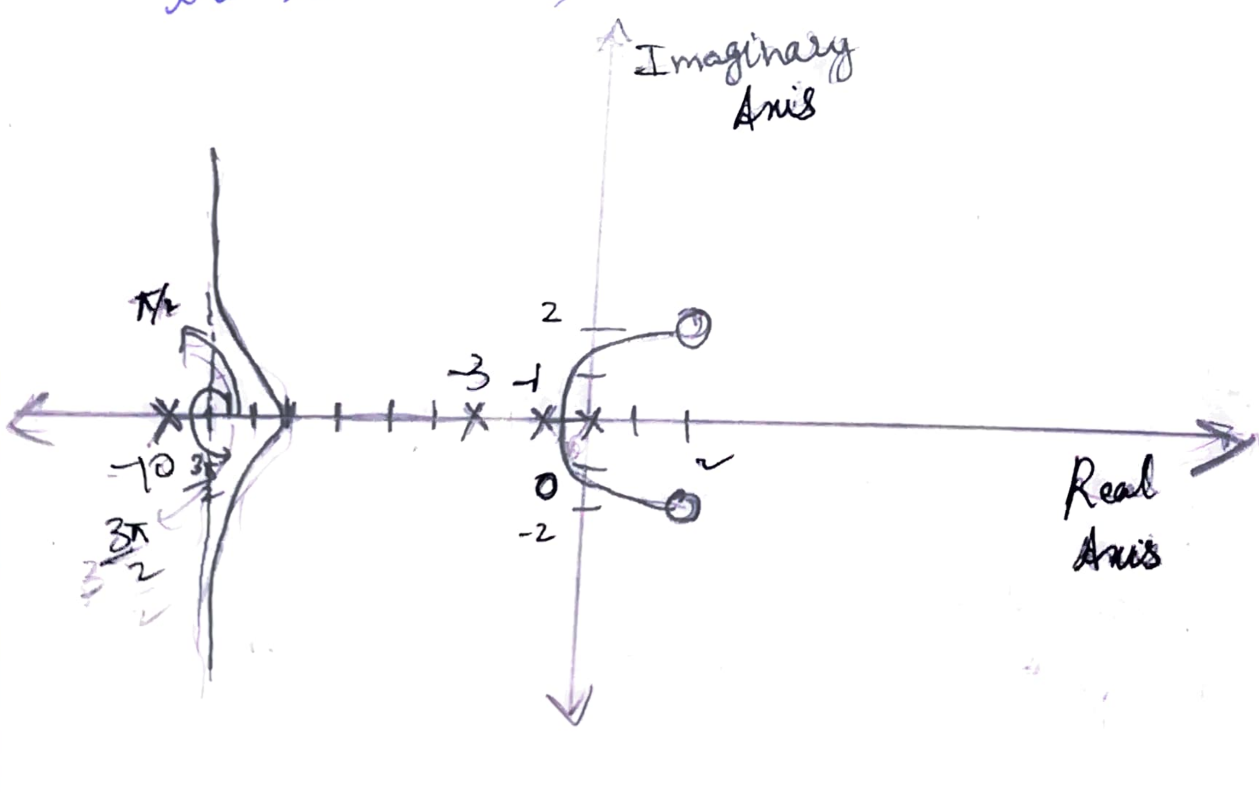
$$= \frac{-(s^4 + 14s^3 + 43s^2 + 30s)}{s^2 - 4s + 8}$$

$$\frac{d\beta}{ds} = 0 \Rightarrow (4s^3 + 42s^2 + 86s + 30)(s^2 - 4s + 8) - (2s-4)(s^4 + 4s^3 + 43s^2 + 30s) = 0$$

Solving for s,

$s = -0.384, -7.07$ are the relevant maximas.

(others are imaginary roots – don't lie on the real axis)



c) ~~Sol~~. Recall, the C.E. cast in RL form:

$$1 + K_c \left[\frac{s^2 - 4s + 8}{(s)(s+1)(s+3)(s+10)} \right]$$

Call the parameter K_c as β

Now, we have the ultimate gain when the root locus intersects the imaginary axis (and goes to RHP)

Let the intersection be at some $s = j\omega$ (where ω is a real value)

$$\Rightarrow 1 + \beta \cdot \frac{((j\omega)^2 - 4(j\omega) + 8)}{((j\omega)^4 + 14(j\omega)^3 + 43(j\omega)^2 + 30 j\omega)} = 0$$

$$\Rightarrow (\omega^4 - 14 j\omega^3 - 43\omega^2 + 30 j\omega) + \beta(-\omega^2 - 4j\omega + 8) = 0$$

$$\Rightarrow (\omega^4 - \beta\omega^2 - 43\omega^2 + 8\beta) + j(-14\omega^3 + 30\omega - 4\beta\omega) = 0$$

Equate the real and imaginary parts to 0

$$\omega^4 - \beta\omega^2 - 43\omega^2 + 8\beta = 0 \quad \text{——— (1)}$$

$$-14\omega^3 + 30\omega - 4\beta\omega = 0 \quad \text{——— (2)}$$

② $\Rightarrow$ $\omega^2 = \dfrac{30 - 4\beta}{14}$ ——— ③

Substitute the above equation in ①,

$$\left(\frac{30-4\beta}{14}\right)^2 - \beta\left(\frac{30-4\beta}{14}\right) - 43\left(\frac{30-4\beta}{14}\right) + 8\beta = 0$$

$$\Rightarrow 900 + 16\beta^2 - 240\beta - 420\beta + 56\beta^2 - 18060 + 2408\beta + 8\times196\beta = 0.$$

$$\Rightarrow 72\beta^2 + 3316\beta - 17160 = 0$$

$$\Rightarrow \beta = \frac{-3316 \pm \sqrt{(3316)^2 + 4\times17160\times72}}{144}$$

$$= 4.696 \quad \text{or} \quad -50.252$$

Since $K_c > 0$ for a P-controller,

$\beta = 4.696 \Rightarrow \omega = \sqrt{\dfrac{30-4\beta}{14}}$

$= 0.895$

$\therefore K_{c,\,\text{ultimate}} = 4.696$

$\therefore$ Ultimate Gain $K_{cu} = \boxed{4.696}$

1e) Characterstic Equation : $1 + G_c G_p = 0$

$$1 + \left(K_c + \frac{K_I}{s}\right)\left(\frac{s^2 - 4s + 8}{s(s+1)(s+3)(s+10)}\right) = 0$$

$$\Rightarrow \left[1 + K_c\left(\frac{s^2 - 4s + 8}{s(s+1)(s+3)(s+10)}\right)\right] + \frac{K_I}{s}\,\frac{s^2 - 4s + 8}{s(s+1)(s+3)(s+10)} = 0$$

$$\Rightarrow 1 + \frac{K_I \frac{1}{s}\left(\frac{s^2 - 4s + 8}{s(s+1)(s+3)(s+10)}\right)}{1 + K_c\left(\frac{s^2 - 4s + 8}{s(s+1)(s+3)(s+10)}\right)} = 0$$

Substituting $K_c = 0.85$ & simplifying

$$\Rightarrow 1 + \frac{K_I (s^2 - 4s + 8)}{s^5 + 14s^4 + 43.85s^3 + 26.6s^2 + 6.8s} = 0$$

To get the K_I, ultimate we need to find the gain at cross-over point.

For that purpose we can use the root locus plot.

Question 1)

Question 1) b) RL plot

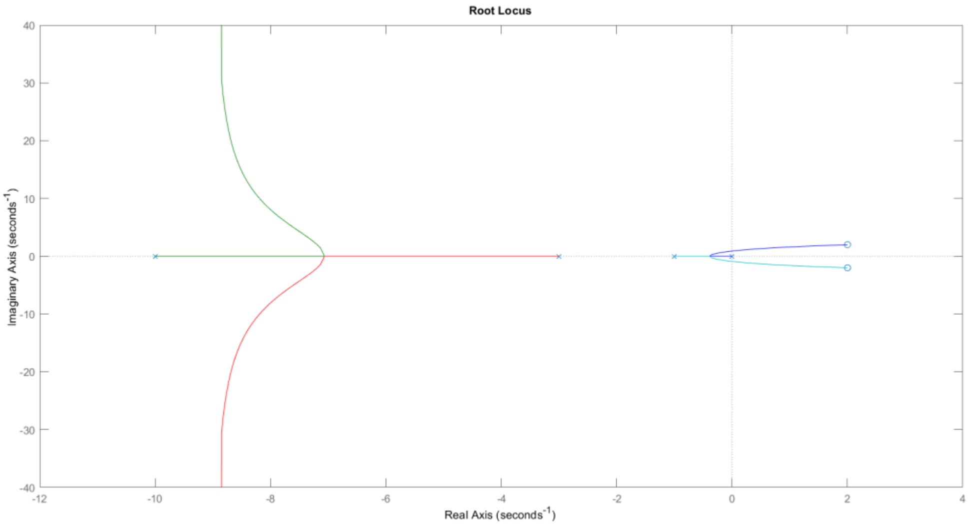


Figure 1.1: Root Locus Plot generated using rlocusplot in MATLAB

We can also see that the asymptotes meet at -9 (centroid) and at angles of 90 and 270 degrees respectively

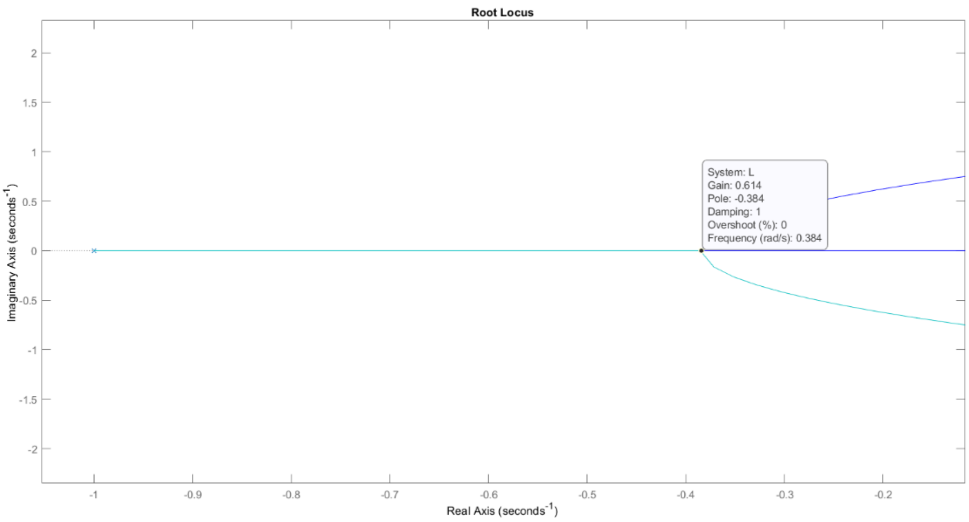


Figure 1.2 a): Verifying break away point-1

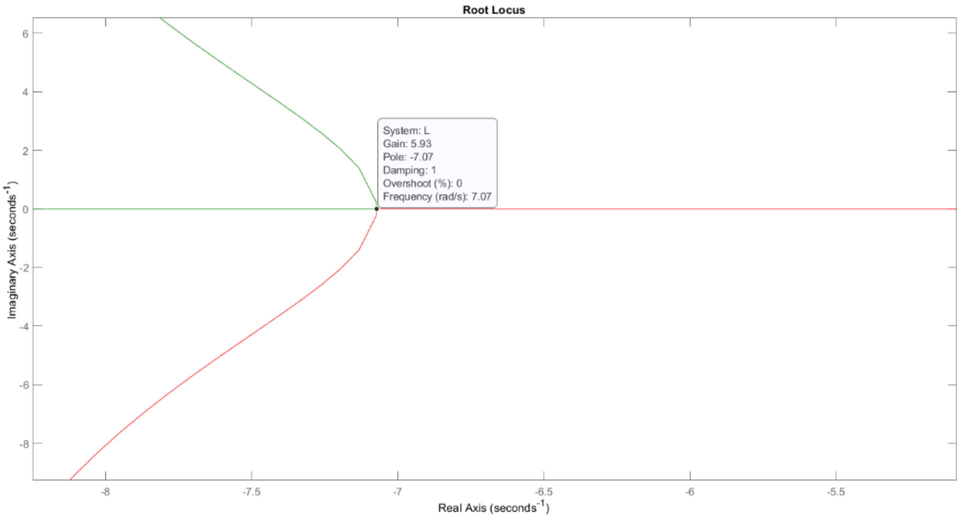


Figure 1.2 b): Verifying break away point-2

Question 1) c) Ultimate Gain verification

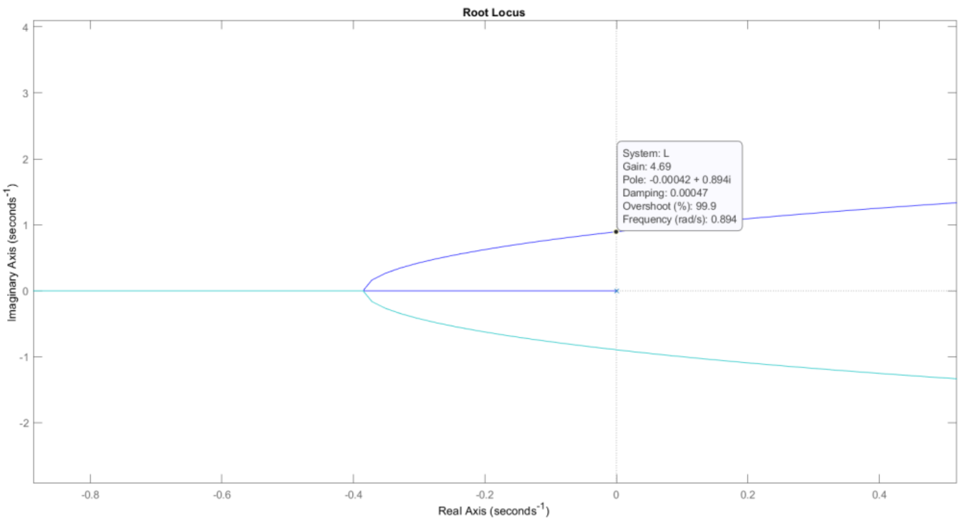


Figure 1.3: Finding the cross-over point.

From the above plot we can see that at approximately the cross-over point, the gain is approximately 4.69. This is very close to the analytically computed value of 4.696 hence verifying it.

Question 1) d) Minimising settling time

We know that $K_C > 0$, and $K_C < K_{C,U} = 4.696$.

So the K_C value which gives the minimum settling time must be between these 2 values. Employing the function '**stepinfo**' which gives the settling time (apart from an assortment of other information about the step response), we evaluate settling time for all K_C values between 0 and 4.69 (with 2 decimal place accuracy) and find the K_C which gives the minimum value.

The required **K_C = 0.85** and it has a settling time of 9.0798 units.

This is further verified using rltool. We need to get the dominant pole away from origin as far as possible to quicken the dynamics of the system. (And that the same time all poles should be in LHP)

This is achieved in this configuration:

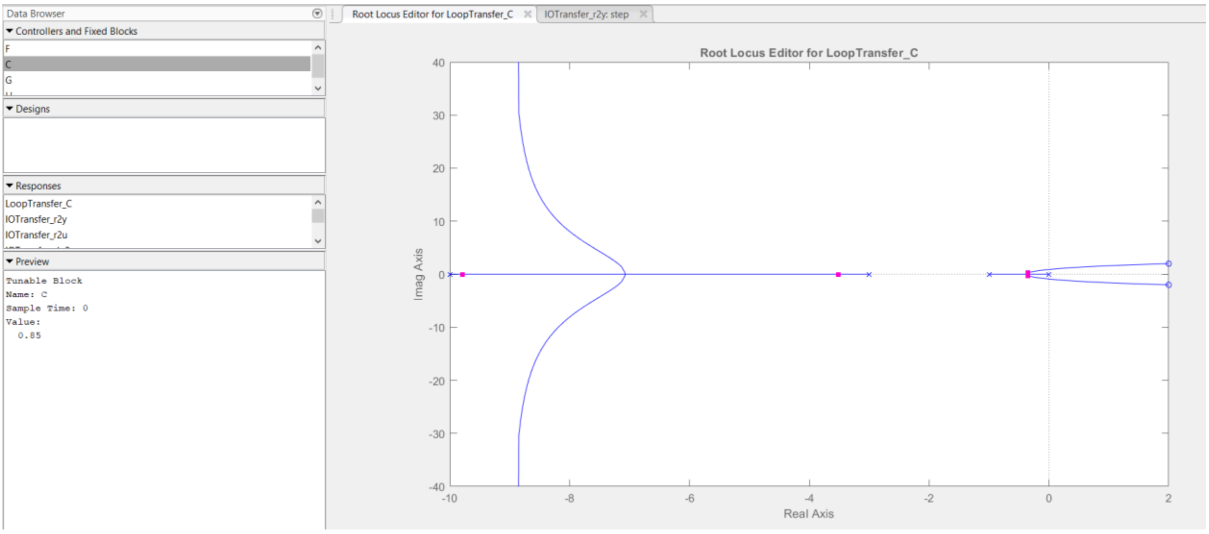


Figure 1.4: RL plot using rltool at minimum settling time

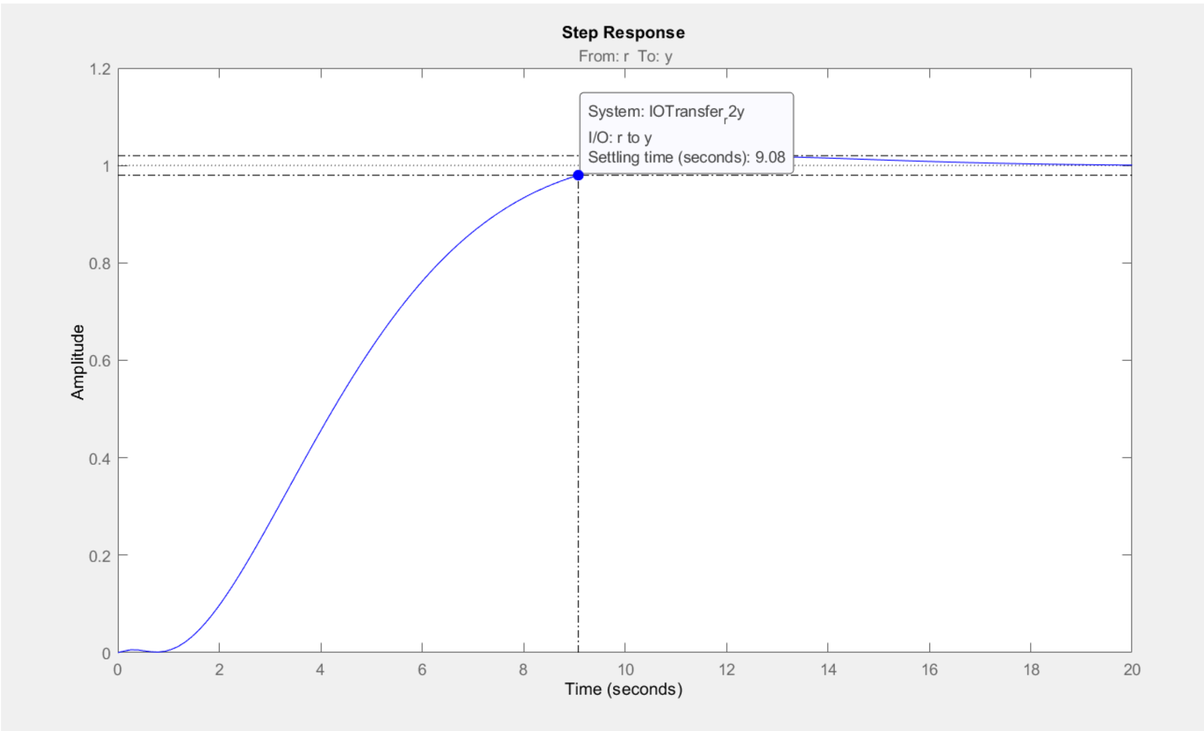


Figure 1.5: Settling time for the optimal Kc.

Question 1) e) Ultimate gain for K_I

Given that the equation for solving s = jw is complicated, it is much easier to use the root-locus plot generated by MATLAB.

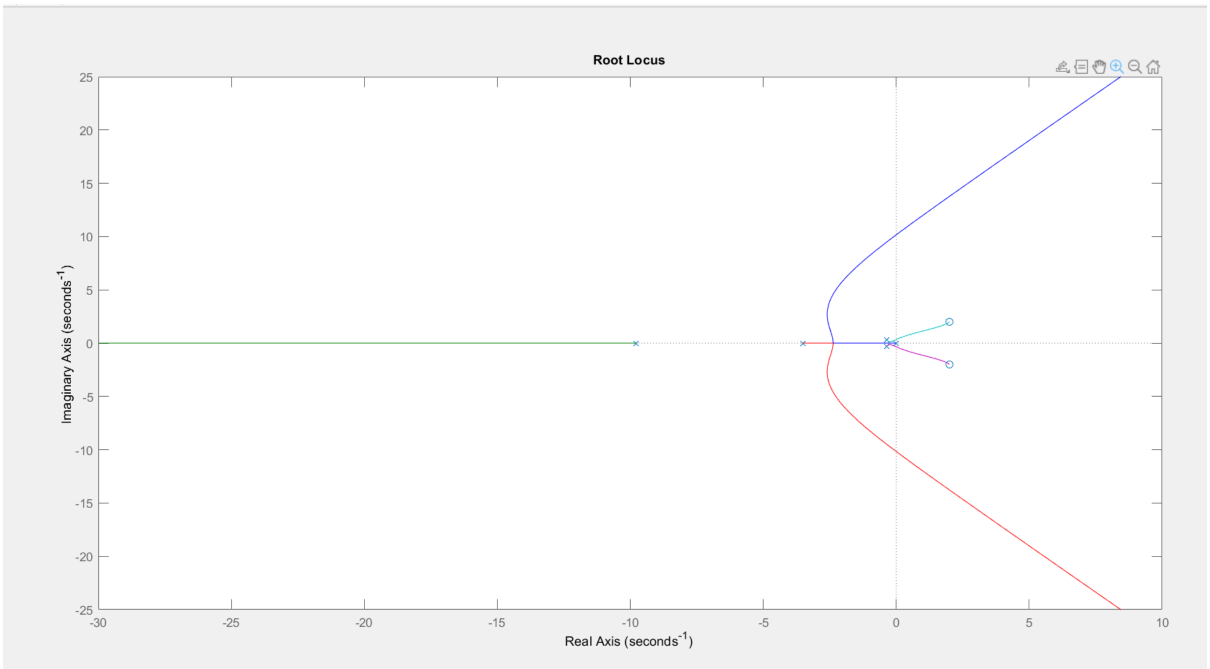


Figure 1.6 a): The required root locus plot

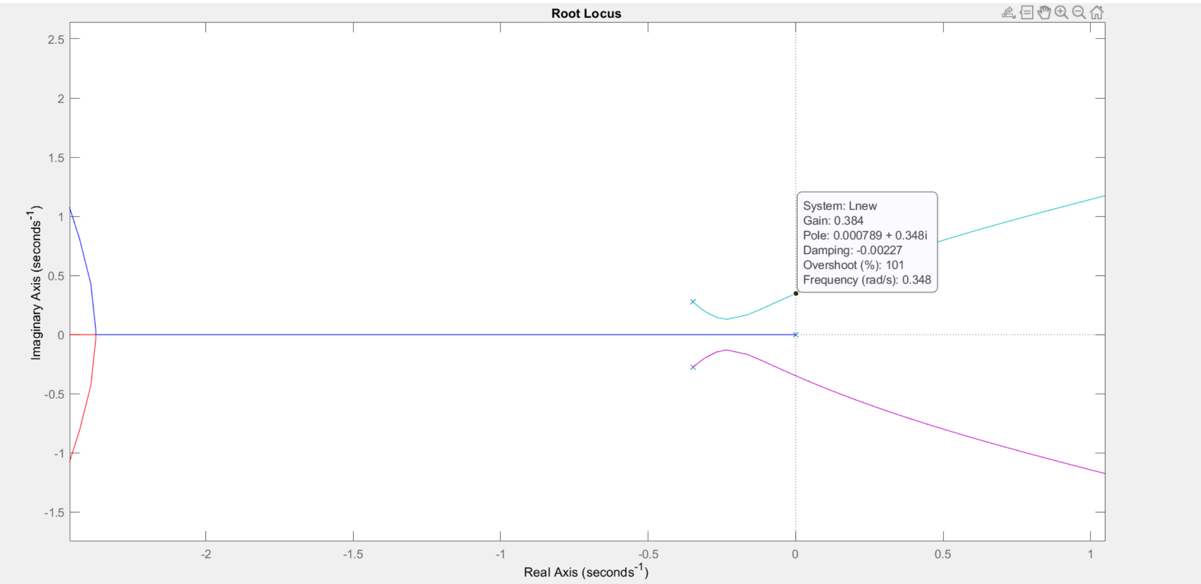


Figure 1.6 b): Finding the cross-over point

$K_{I,ultimate}$ = 0.384.

Note: there is a another cross-over point at 1.540*10^3. But if K_I>0.38 there will be RHP poles (although there will be some LHP poles also). So the ultimate gain is the gain at the first cross-over point.

Question 2)

Question 2) a) P-Control design for given GM

We can use the Bode stability criteria and obtain the Gain margin for K_C = 1. So L is simply equal to G_P as given below.

```
L =

                 2 s + 8
  exp(-2*s) * ----------------
              10 s^2 + 7 s + 1

Continuous-time transfer function.
```

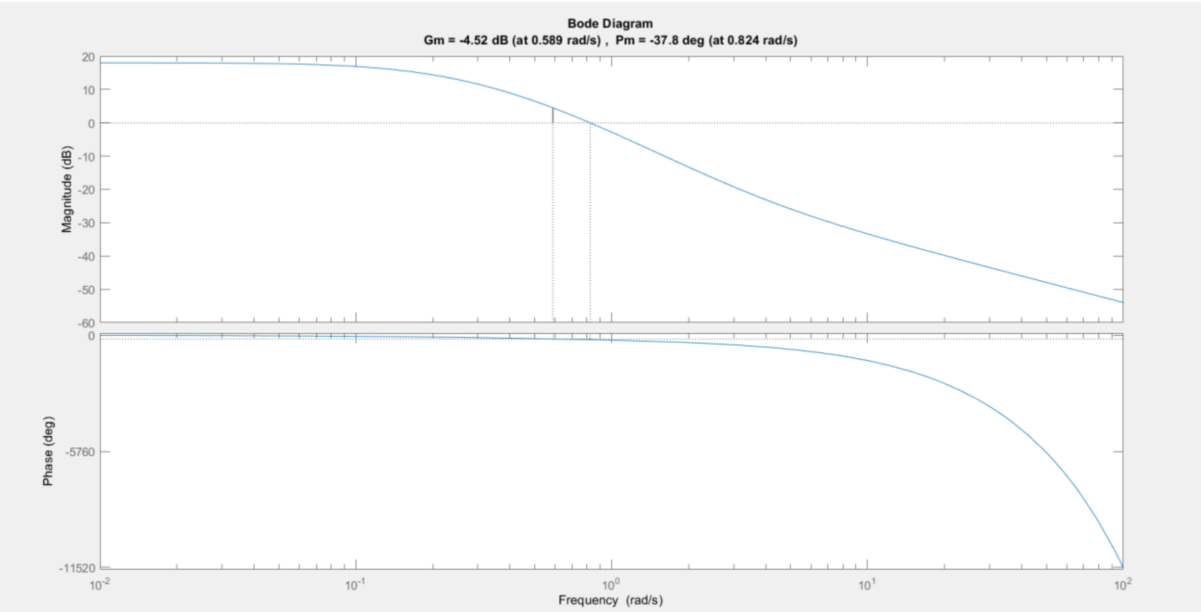


Figure 2.1: Bode Diagram (using 'margin' function).

G_M = -4.52 dB

So using this gain margin and the definition of gain margin,

$G_M = \log(\frac{1}{|L_G|})$ and $L_G = G_C G_P = K_C * G_P$

We can get the required K_C as

$$K_C = 10^{\frac{(G_M - G_{M,reqd})}{20}}$$

where $G_{M,required}$ = 8.2 dB

K_C was found to be **0.2311**. The corresponding phase margin = **73.815 degrees**

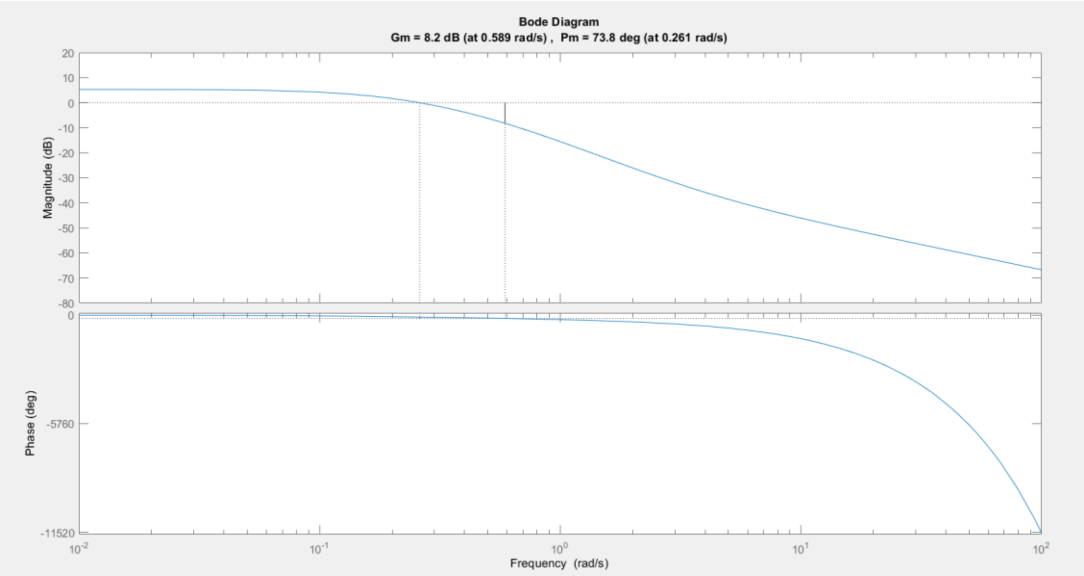


Figure 2.2: Margins of the required P-Controller

Question 2) b) Delay uncertainty

Value of delay uncertainty = Phase Margin/gain cross-over frequency

(since, Phase is directly proportional to –w*Delay)

Gain cross-over frequency = 0.2607 rad/s

=> **Delay Uncertainity = 4.942 units**

Question 2) c) PI Controller

Unlike part a), changing τ_I changes both magnitude and phase responses of the system.

So one option is to solve for gain cross over frequency and the controller parameter τ_I simultaneously by imposing

$$|L(j\omega)| = 1 \tag{1}$$

$$phase\big(L(j\omega)\big) = 60 \tag{2}$$

Instead, I used the function 'margin' to compute the phase margin for the closed loop system given a τ_I. This results in simply solving for tau such that phase is 60.

```
%% function that gives 60-PM for a given tau
function P  = func(tauI,L,Kc)
    s = tf('s');
    [~,PM,~,~] = margin(L*Kc*(1+1/tauI/s));
    P = 60 - PM;
end
```

PI controller parameter τ_I was found to be **17.8327** units

The Gain margin was found be **7.46 dB**

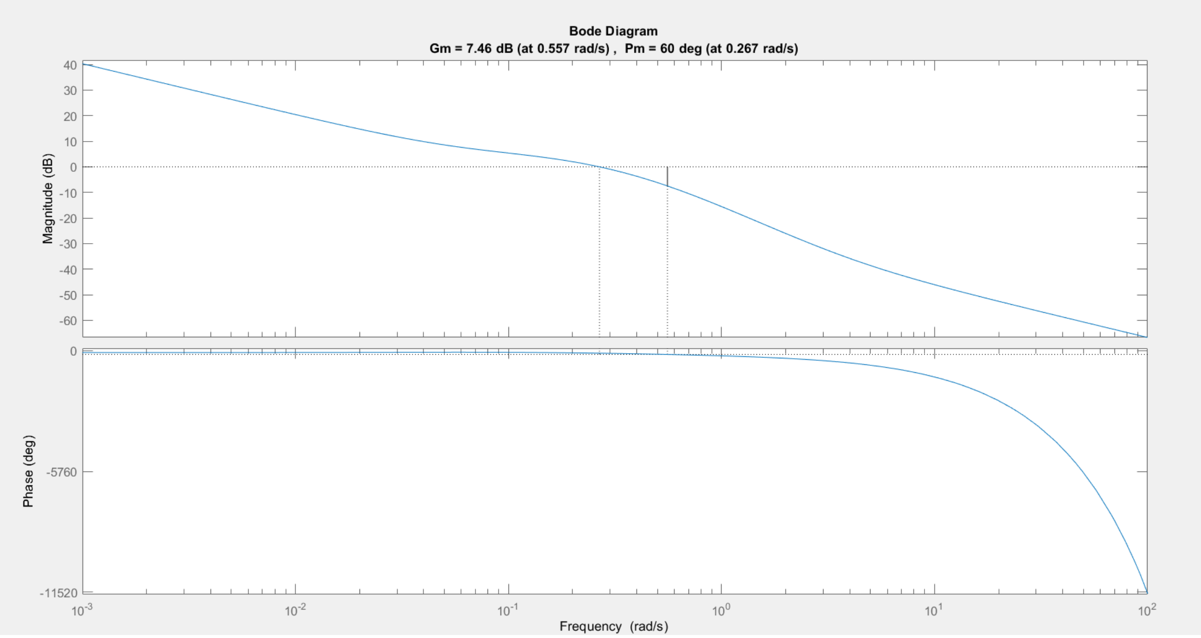


Figure 2.3: Bode diagram for the closed loop system with the PI controller

Question 2) d) Verifying Bode's Sensitivity Integral

$$\int_0^\infty \log|S_O(j\omega)|d\omega = 0$$

where $S_O(j\omega) = \frac{1}{1+Gp(j\omega)Gc(j\omega)}$

The above integral is verified using the following script:

1. PI Controller

```
%% Sensitivity function
function So  = Q2_So(tauI,Kc,w)
    Gp = 2*(1j*w + 4)./(-10*w.^2+1+7*1j*w).*exp(-2j*w);
    Gc = Kc*(1+1/tauI./(1j*w));
    So = 1./(1+Gp.*Gc);
end
```

```
%% Evaluating the sensitivity integral
logmod = @(tauI,Kc,w)  (log(abs(Q2_So(tauI,Kc,w))));
int_val = integral(@(w)logmod(tauI,Kc,w), 0, 10^4);
```

(Since it is numerical integration, I replaced infinity with a high value instead. Higher the value, closer it gets to zero)

The integral turns out to be, int_val = -1.8790e-06 which is sufficiently close to 0!

Thus, the Bode's sensitivity integral holds for the PI controller!

2. P Controller

```
% Sensitivity function P controller
function So  = Q2_So2(Kc,w)
    Gp = 2*(1j*w + 4)./(-10*w.^2+1+7*1j*w).*exp(-2j*w);
    Gc = Kc;
    So = 1./(1+Gp.*Gc);
end
```

```
% P Controller
logmod2 = @(Kc,w)  (log(abs(Q2_So2(Kc,w))));
int_val2 = integral(@(w)logmod2(Kc,w), 0, 10^5);
```

The integral turns out to be, int_val = -2.3075e-07which is sufficiently close to 0!

Thus, the Bode's sensitivity integral holds for P-Controller also!

② d) $$S_o = \frac{1}{1 + G_p G_c}$$

$$G_p = \frac{2(s+4)}{10s^2+7s+1} \times e^{-2s}$$

$$G_c = K_c \left(1 + \frac{1}{\tau_I s}\right)$$

$$\Rightarrow S_o = \frac{1}{1 + \frac{2(s+4)}{10s^2+2s+1} K_c \left(1+\frac{1}{\tau_I s}\right) e^{-2s}}$$

$$\Rightarrow S_o(j\omega) = \frac{1}{1 + \frac{2(j\omega+4)}{(1-10\omega^2+7j\omega)} K_c \left(1+\frac{1}{\tau_I j\omega}\right) e^{-2j\omega}}$$

③ a) Let ~~us~~ the controller gain be β.

$$G_{approx}(s) = \frac{2(s+2)}{s^2+2s-3} \frac{\left(1-\frac{s}{2}\right)}{\left(1+\frac{s}{2}\right)} = \frac{2(s+2)(2-s)}{(s^2+2s-3)(2+s)}$$

C.E. :

$$1 + \beta G_p = 0$$

Case ①: Cancel off $(s+2)$ in Nr & Dr

$$\Rightarrow \frac{2\beta\left(\frac{2-s}{\cancel{s+2}}\right)}{(s^2+2s-3)} + 1 = 0$$

$$\Rightarrow s^2 + 2(1-\beta)s + 4\beta - 3 = 0$$

Conditions $P_1 = -2$ & $Re(P_2) < 0$ (& preferably $Re(P_2) < -2$)

condition ① $\Rightarrow 4 - 4(1-\beta) + 4\beta - 3 = 0$

$$\Rightarrow \beta = 3/8$$

condition ② $\Rightarrow$ product of roots > 0
sum of roots < 0

$$\Rightarrow 4\beta - 3 > 0 \text{ \& } -2(1-\beta) < 0$$

$$\Rightarrow \beta > 3/4 \text{ \& } \beta > 1$$

But $\beta = 3/8$ for pole $= -2$.

$\therefore$ We can't satisfy the condition —— ①

case ②: dont cancel $(s+2)$ term.

$C.E \Rightarrow (s+2)(s^2 + 2(1-\beta)s + 4\beta - 3) = 0$

(same as previous except with $as+2$ facts)

This time we already have $p = -2$.

So we just need to impose

$Re(p) < 0$ & preferably $Re(p) < -2$

Roots : $-(-1+\beta) \pm \sqrt{\beta^2 - 6\beta + 4}$ —— ②

Now the term under squareroot has roots

$\beta = 3 \pm \sqrt{5}$

Subcase ①:

$\sqrt{\beta^2 - 6\beta + 4}$ is imaginary $\left(\begin{array}{l} \beta > 3 - \sqrt{5} \\ \beta < 3 + \sqrt{5} \end{array} \right)$

$\Rightarrow +(\beta - 1) < -2$

$\Rightarrow \beta < -1$

But $\beta > 3 - \sqrt{5}$

$\therefore$ No soln exist in subcase ①

—— ③

Subcase ②:

$\sqrt{\beta^2-6\beta+4}$ is real $\mid$ $\beta > 3+\sqrt{5}$ or $\beta < 3-\sqrt{5}$

$\Rightarrow \beta - 1 \pm \sqrt{\beta^2-6\beta+4} < -2$

$\Rightarrow \cancel{(\beta-3)} \pm \sqrt{\beta^2-6\beta+4} \; \cancel{<} > \; \cancel{3-\beta} \;\; \beta+1$

$\Rightarrow \beta^2-6\beta+4 \; \cancel{<} > \; \cancel{9+\beta^2-6\beta} \;\; \beta^2+2\beta+1$

$\Rightarrow \beta < 3/8$

But $\beta > 3+\sqrt{5}$

not satisfied

$\therefore$ No solution exists in subcase ② ④

From ①, ⑤, ④

We can't solve for K_c such that $p=-2$ is the dominant pole.

$\rightarrow$ This was also verified using RLTOOL in MATLAB

b) $G(s) = \dfrac{2(s+2)}{s^2+2s-3} e^{-s}$; $\Rightarrow G_p G_c = L$

$\Rightarrow L(j\omega) = K_c \dfrac{2(j\omega)+2}{(j\omega)^2+2(j\omega)-3} e^{-j\omega}$

Solve for intersection with real axis (to get gain margin)

$\phi = k\pi$

$\Rightarrow$

$\Rightarrow -\omega + \tan^{-1}\left(\dfrac{\omega}{2}\right) - \tan^{-1}\left(\dfrac{2\omega}{\omega^2+3}\right) = k\pi$

$\Rightarrow \omega = \tan^{-1}\left(\dfrac{\frac{\omega}{2} + \frac{2\omega}{\omega^2+3}}{1 - \frac{2\omega^2}{\omega^2+3}}\right)$ (if $k=0$)

$\Rightarrow \tan\omega = \dfrac{4\omega + \omega^3 + 3\omega}{6}$

$\Rightarrow \tan\omega = 7\omega + \dfrac{\omega^3}{6}$

Sols $\omega = 0$, $\omega = 0.78$

$|L(j\omega)|_{\omega=0} = \dfrac{2K_c\sqrt{4}}{\sqrt{9}}$

$\Rightarrow GM = 1.092\,K_c$

In dB, $GM = -20\log\dfrac{1}{1.092K_c}$

$\Rightarrow 4.5 = -20\log\dfrac{1}{1.092K_c} \Rightarrow K_c = 0.224$

Similarly one can get.

$K_c = 0.2733$ by subst $\omega = 0.78$

We can use FVT to get offset provided the system is closedloop stable.

Stability is checked using the Nyquist plot

poles of G: $\frac{-2 \pm \sqrt{16}}{2} = 1, -3$

1 RHP pole.

$\therefore Z = N + P$ But $N = 0$ (no encirclements around -1)

$\Rightarrow Z = 1$

We have one RHP ~~zero~~ zero for $L + 1 = 0$

$\Rightarrow$ we have an RHP pole for the CL system

$\therefore$ the CL system is unstable i.e. mathematically offset $\rightarrow \infty$.

Question 3)

Question 3) a)

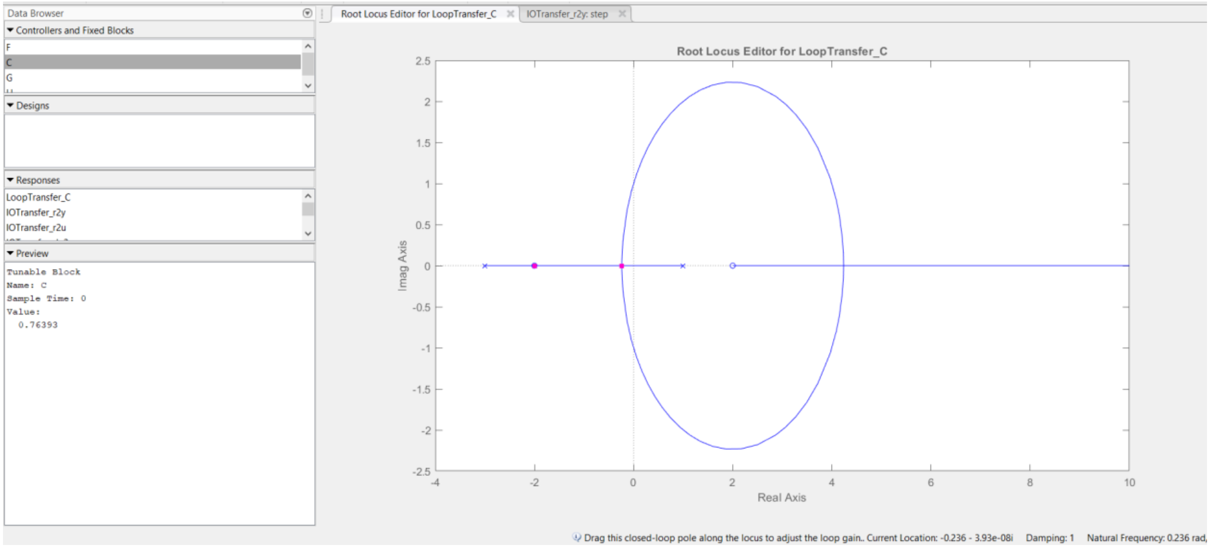


Figure 3.1: Root locus diagram with Pade's first order approximation

Question 3) b)

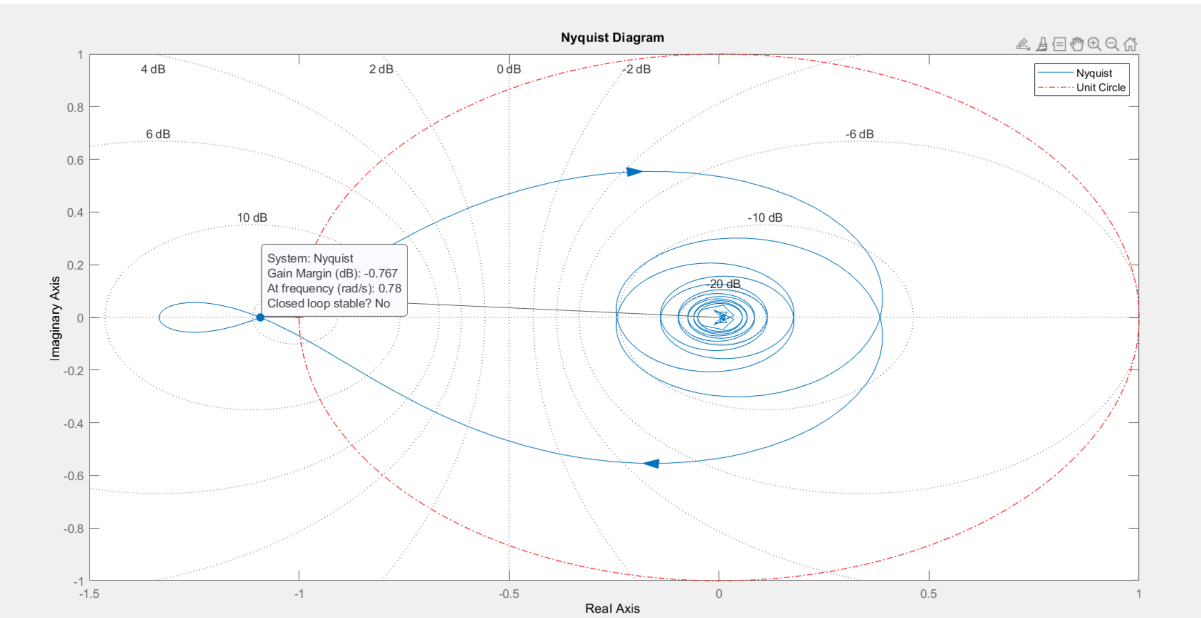


Figure 3.2: Nyquist diagram. K_C = 1. The red dotted curve is the unit circle.

Using the above diagram, we see that GM = -0.767 dB for K_C = 1.

So the required K_C for which gain margin is 10.5 dB is given by,

$$K_C = 10^{\frac{-0.767-10.5}{20}}$$

This gives K_C = 0.2733. However this system is unstable as depicted below in the Nyquist plot.

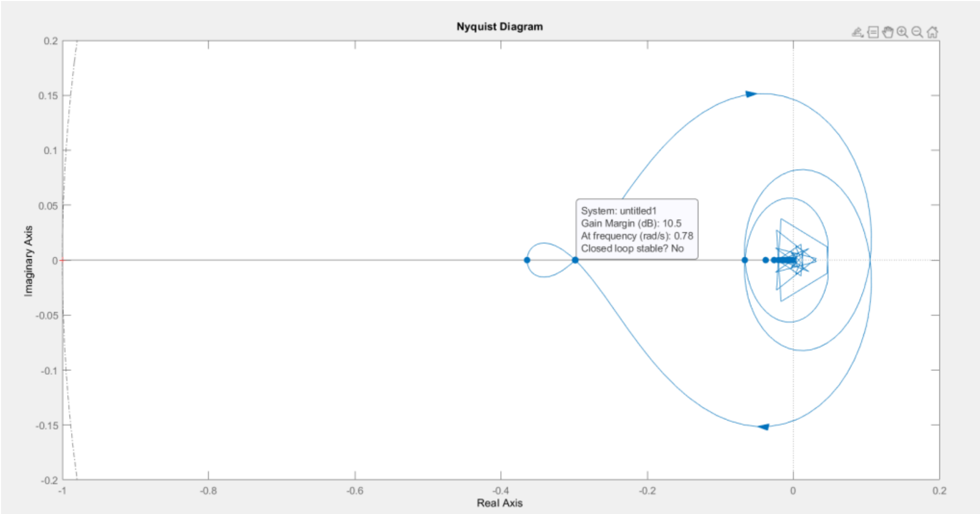


Figure 3.3: Nyquist Diagram for the system with gain margin 10.5 dB (K_C = 0.2733.)

As shown in handwritten part we can also get GM = 10.5 dB at w = 0 rad/s if K_C = 0.224.

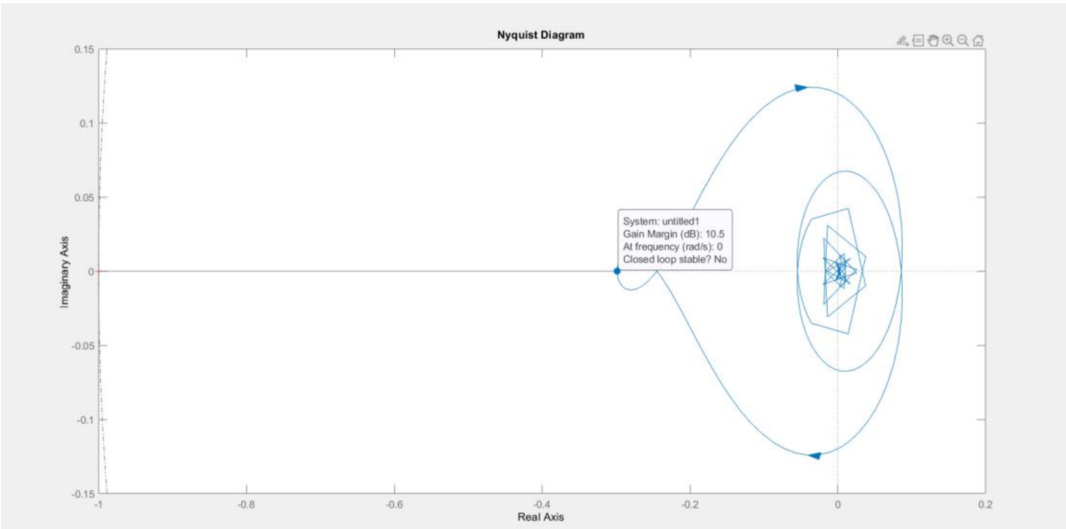


Figure 3.4: Nyquist Diagram for the system with gain margin 10.5 dB (K_C = 0.224)

Since the system is closed loop unstable, the step response of the system blows up. (So we can't define offset for this situation; offset -> infinity)

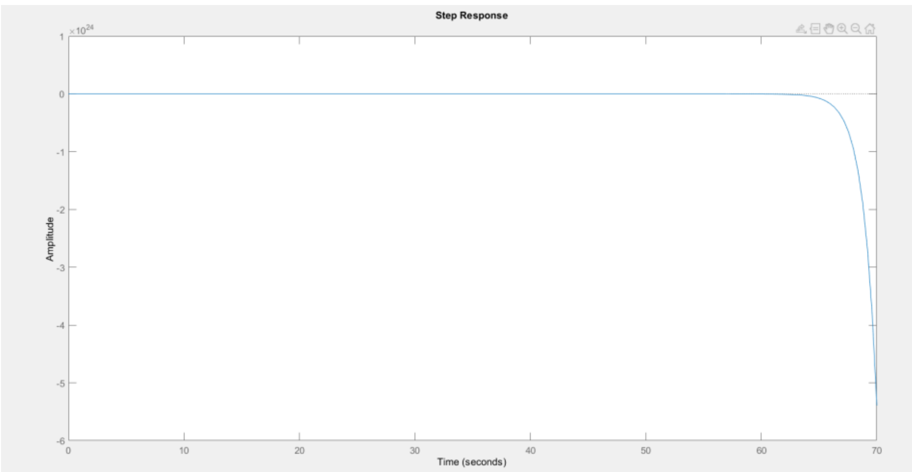


Figure 3.3: Step response of the system.

Question 3) c)

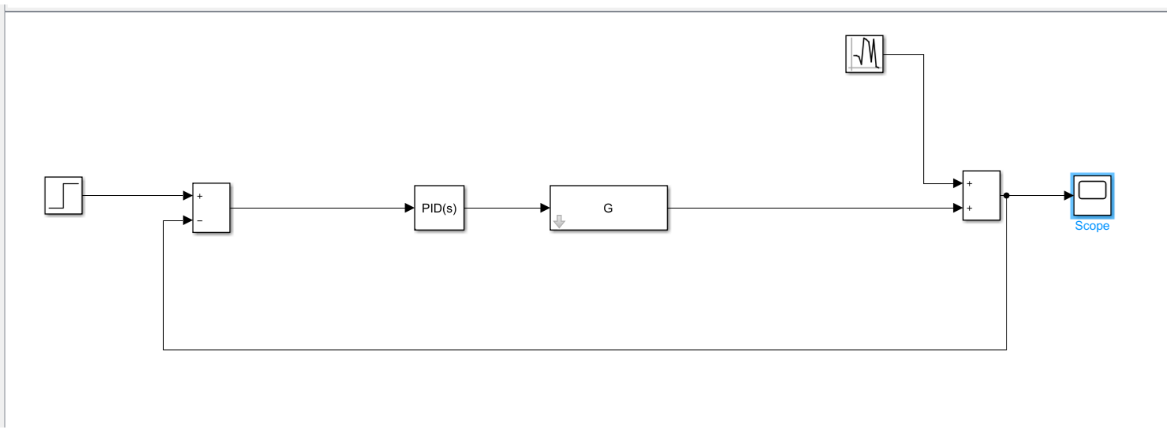
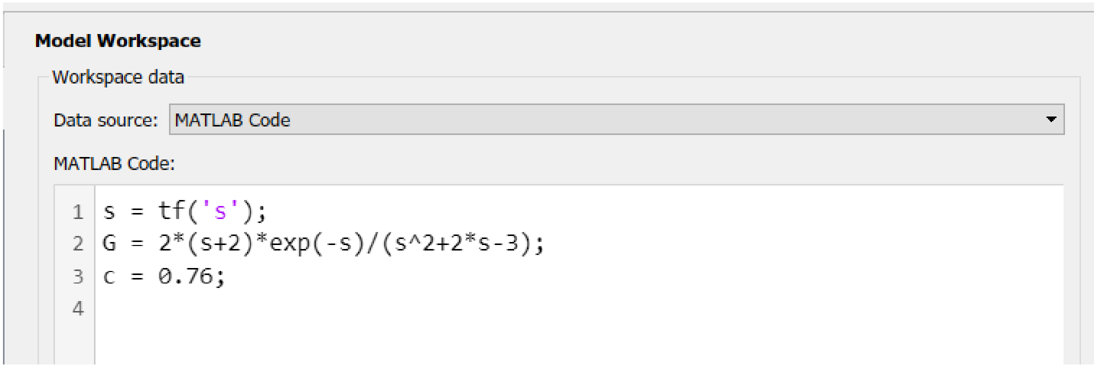


Figure 3.5 Simulink diagram

Variance of the disturbance is set to be 0.1.



Using Pade's second order approximation

Since we can't impose p = -2 is dominant condition I simply just used a K_C which gives CL stability when approximating the function using Pade's.

L for the case of **second order Pade's** approximation:

$$L = 2 * \frac{s+2}{s^2 + 2 * s - 3} * \frac{1 - \frac{s}{2} + \frac{s^2}{8}}{1 + \frac{s}{2} + \frac{s^2}{8}}$$

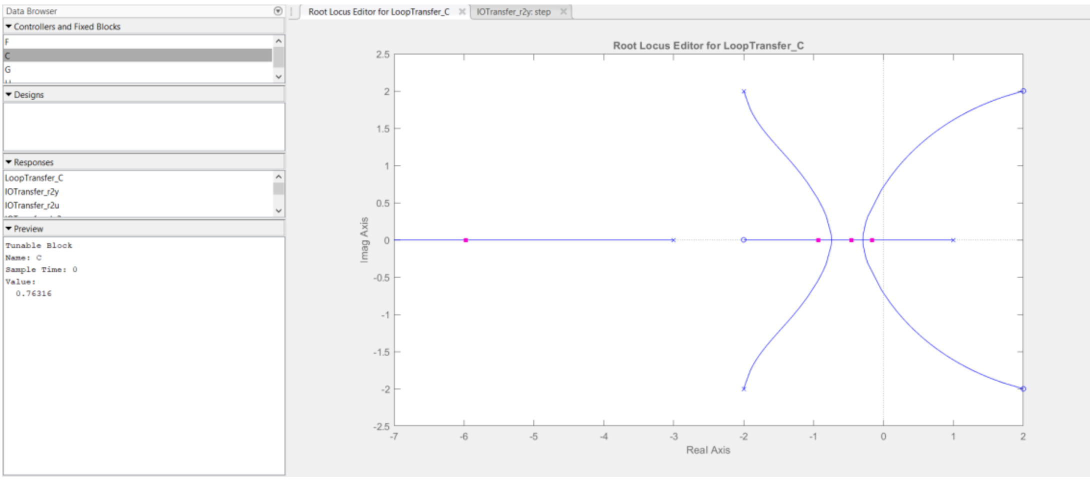


Figure 3.6: Rlocus plot for second order Pade's approximation.

A stable value of K_C came out to be **0.76** as shown in the above figure.

The same was used in the SIMULINK file and the following response was obtained.

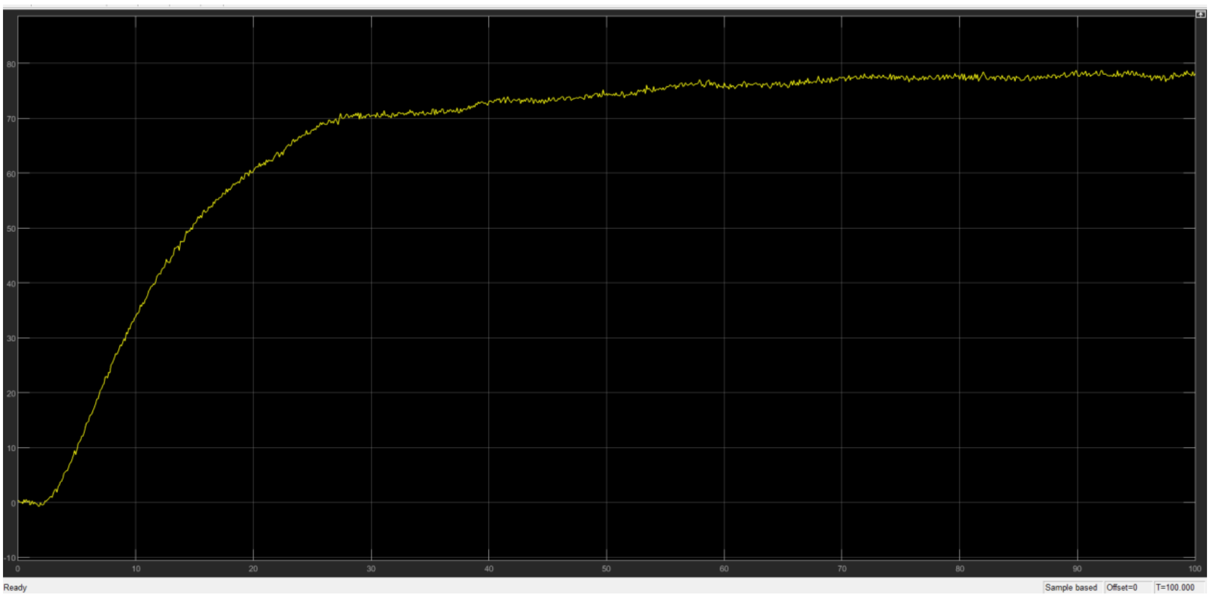


Figure 3.7: Step response controller obtained using second order Pade's approximation.

As we can see, the system is stable but there is a huge offset (~78 units)

Using the controller designed from Nyquist diagram

As shown in the figure below, step response is unbounded for K_C obtained using the Nyquist criterion. (K_C = 0.2733)

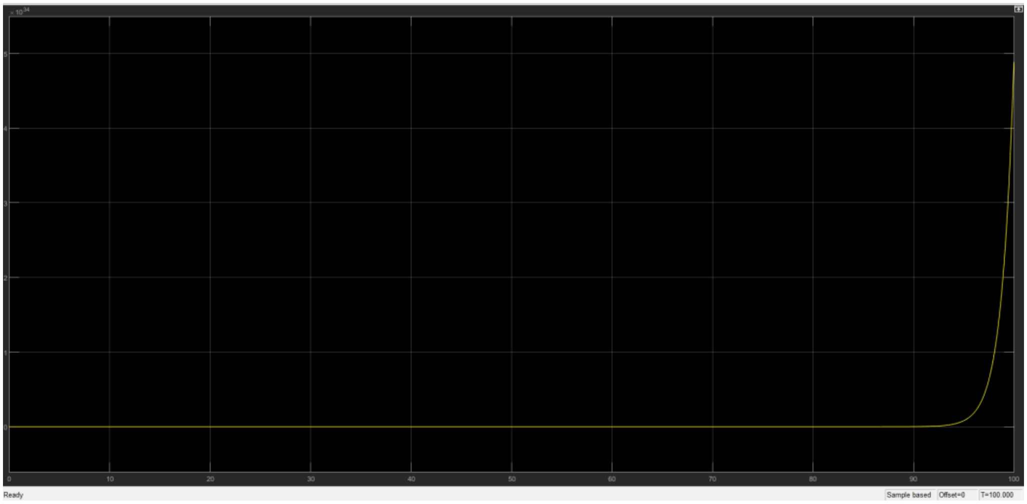


Figure 3.8: Step response for Kc = 0.2733

Summary of question 3:

1. With first order Pade's approximation, we were not able to find any controller that satisfies the given condition of p = -2 being dominant.
2. Using the given condition on Gain Margin and Nyquist plot, we obtained a K_C but that again gave unbounded response as shown by the above SIMULINK simulation.
3. Also, Nyquist plot has multiple cuts on real axis, so it is not easy to find out a gain margin. (Need the notion of a 'minimum stability' gain margin)
4. Controller gain obtained from checking RL plot of a second order Pade's approximated G_P seems to be stable, but the final value has a huge offset of about 79 units.
5. Pade's second order approximation improved the prediction but again, the choice of c was random, we weren't able to impose all the required conditions. So nothing special yielded from this.(could just be luck that I somehow got a stable K_C)
6. **Conclusion**: Delay seems to be badly affect the design of a P-Controller. It imposes severe restrictions on K_C and the responses are also not easily predictable.

Codes

Q1

```
clear; close all;
%% Setup the system
s = tf('s');
Gp = (s^2-4*s+8)/(s*(s+1)*(s+3));
G_sens = 1/(s+10);
L = Gp*G_sens;
%% Rootlocus plot
%rltool(L);
rlocus(L);
%% Solve for break in point
p = conv([4 42 86 30],[1 -4 8]) - conv([2 -4],[1 14 43 30 0]);
r = roots(p);
%% Part d)
Kcu = 4.69;
k = 0.01:0.01:Kcu;
r1 = zeros(length(k),1);
for i = 1:length(k)
  G = Gp*G_sens;
  sys = k(i)*G/(1+k(i)*G);
  S = stepinfo(sys);
  r1(i) = S.SettlingTime;
end
[val,loc] = min(r1);
Kc = k(loc);
%% Part e)
Lnew = tf([1 -4 8],([1 14 43 30 0 0]+Kc*[0 0 1 -4 8 0]));
figure;
rlocusplot(Lnew);
```

Q2

```
clear; close all;
%% Setup a P controller
Gp = tf([2 8],[10 7 1],'iodelay',2);
L = Gp;
[Gm,Pm,Wcg,Wcp] = margin(L);
margin(Gp);
Gm= 20*log10(Gm);
Gm_reqd = 8.2;
K_cu = 10^(Gm/20);
Kc = K_cu*10^((-Gm_reqd)/20);
[Gm2,Pm2,Wcg2,Wcp2] = margin(L*Kc);
%% Delay Uncertainity
w = Wcp2; % Here wcp and wcg correspond to Wgc, and Wpc respectively.
pm_verify = 180 + (atan(w/4)-atan(7*w/(1-10*w^2)) - 2*w)*180/pi;
```

```
figure;
margin(L*Kc);
DM = pm_verify/(w*180)*pi;
%% Designing a PI controller
s = tf('s');
[tauI,fval,exitflag]= fsolve(@(tauI) func(tauI,L,Kc),0.5);
figure;
margin(L*Kc*(1+1/tauI/s));
%% Evaluating the sensitivity integral
% P Controller
logmod2 = @(Kc,w) (log(abs(Q2_So2(Kc,w))));
int_val2 = integral(@(w)logmod2(Kc,w), 0, 10^5);
% PI controller
logmod = @(tauI,Kc,w) (log(abs(Q2_So(tauI,Kc,w))));
int_val = integral(@(w)logmod(tauI,Kc,w), 0, 10^4);
%% function that gives 60-PM for a given tau
function P  = func(tauI,L,Kc)
   s = tf('s');
   [~,PM,~,~] = margin(L*Kc*(1+1/tauI/s));
   P = 60 - PM;
end

%% Sensitivity function PI Controller
function So  = Q2_So(tauI,Kc,w)
   Gp = 2*(1j*w + 4)./(-10*w.^2+1+7*1j*w).*exp(-2j*w);
   Gc = Kc*(1+1/tauI./(1j*w));
   So = 1./(1+Gp.*Gc);
end


%% Sensitivity function P controller
function So  = Q2_So2(Kc,w)
   Gp = 2*(1j*w + 4)./(-10*w.^2+1+7*1j*w).*exp(-2j*w);
   Gc = Kc;
   So = 1./(1+Gp.*Gc);
end
```

Q3

```
clear; close all;
%% Setup the system
s = tf('s');
Gp = 2*(s+2)/(s^2+2*s-3)*exp(-s);
Gp_pade = 2*(2-s)*(s+2)/(s^2+2*s-3)/(s+2);
%% rl plot for fun
%rlocusplot(sys)
% Kc = 0.76393 gives stable roots (from rltool
%% Plot the nyquist diagram
figure();
nyquist(Gp);
grid on;
```

```
hold on;
% Plot the unit circle
n = 500;
theta = linspace(0,2*pi,n);
x = cos(theta); y = sin(theta);
plot(x,y,'r-.')
hold off;
legend('Nyquist','Unit Circle');
w = 0.78; % From nyquist plot
GM = -0.767; % From Nyquist Plot
Kc1 = 10^(-10.5/20)/4*3; % Derived by hand
figure;nyquist(Kc1*Gp)
Kc2 = 10^((-0.767-10.5)/20);
figure;nyquist(Kc2*Gp);
L = 2*(s+2)*(1-s/2+s^2/8)/(s^2+2*s-3)/(1+s/2+s^2/8);
c = 0.76566;
```

NEW QUESTION (updated)

③ a) $G(s) = \frac{2(s+2)}{s^2+2s-3} e^{-s}$.

$$\Rightarrow G_{approx}(s) = \frac{2(s+2)}{s^2+2s-3} \cdot \frac{\left(1-\frac{s}{2}\right)}{\left(1+\frac{s}{2}\right)}$$

$$= \frac{2(2-s)}{s^2+2s-3}.$$

$$C.E = 1 + G_c G_p = 0$$

$$\Rightarrow 1 + K_c \frac{2(2-s)}{(s^2+2s-3)} = 0$$

-0.2 is a root!

$$\Rightarrow K_c = \frac{-\left[(-0.2)^2 + 2(-0.2) - 3\right]}{2(2+0.2)}$$

$$\Rightarrow G_{c1} = 0.76364$$

③ c) 2nd order approx: $L = \frac{2(s+2)\left(1-\frac{s}{2}+\frac{s^2}{8}\right)}{(s^2+2s-3)\left(1+\frac{s}{2}+\frac{s^2}{8}\right)}$

$$\Rightarrow L = \frac{64s^3 - 128s^2 + 1024}{32s^4 + 192s^3 + 416s^2 + 128s - 268}$$

Using rltool, the value of K_c for which the dominant poles has real part -0.2 is 0.7878.

Dominant poles : $-0.2 \pm 0.37j$

Question 3 c) updated

For all simulations, the variance of disturbance was set as 0.1

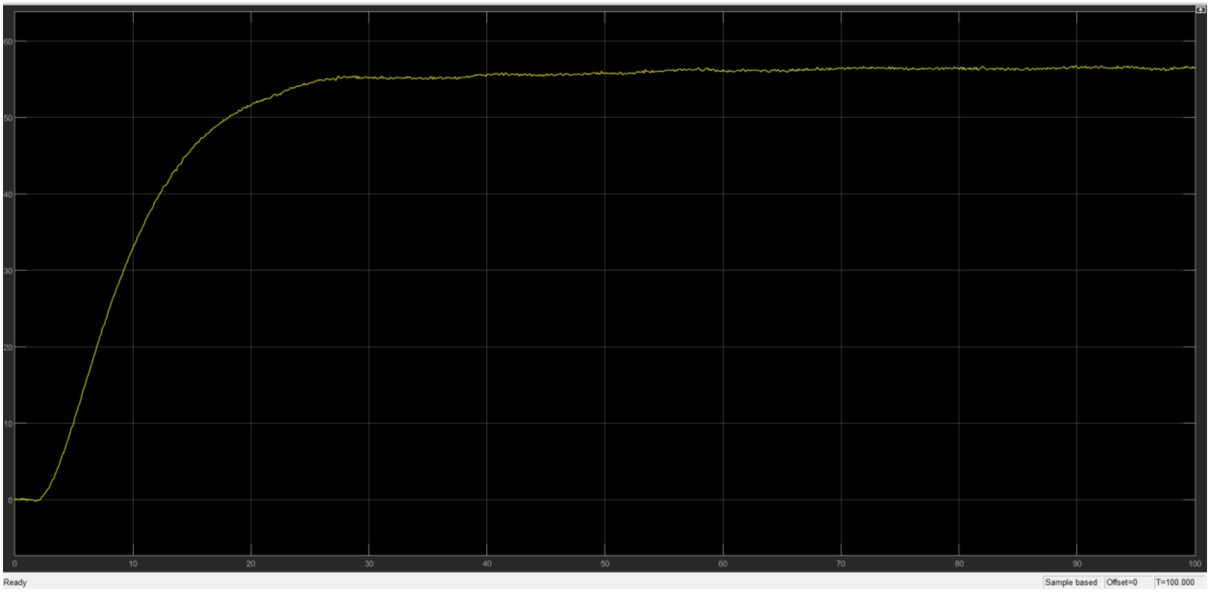


Figure: Plot of step response of the closed loop system for Kc = 0.7636 (part-a)

We see that G_{C1} has an offset of about 55.

For controller design using Pade's second order approximation, we use the rltool on L to get -0.2 as real part of the dominant pole as shown in the below figure

```
Gp_pade_second =

          64 s^3 - 128 s^2 + 1024
  ----------------------------------------
  32 s^4 + 192 s^3 + 416 s^2 + 128 s - 768

Continuous-time transfer function.
```

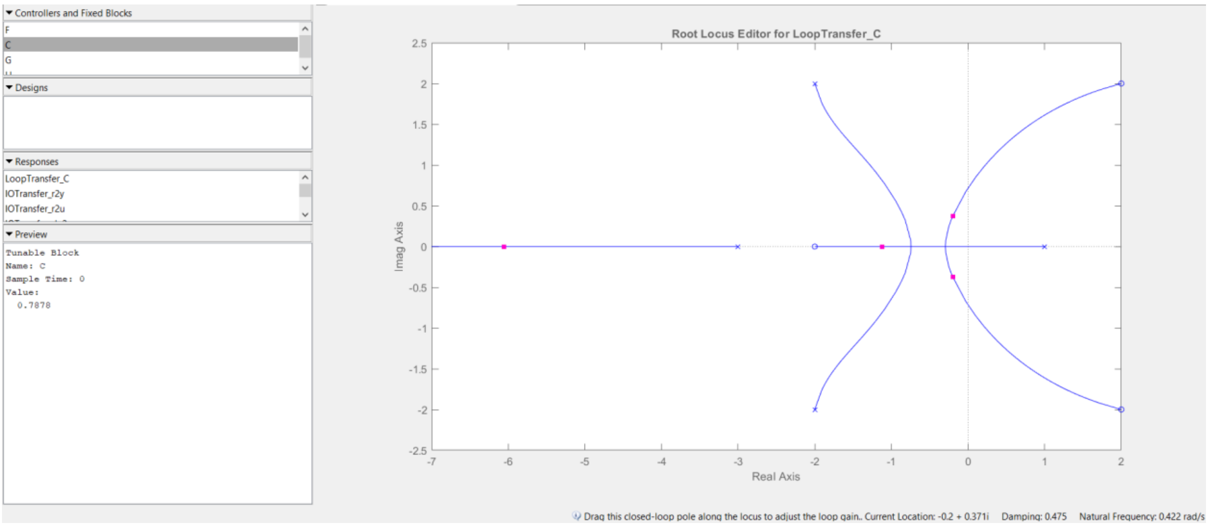


Figure: RL plot with the required roots marked. (along with the K_C value)

K_C = 0.7878

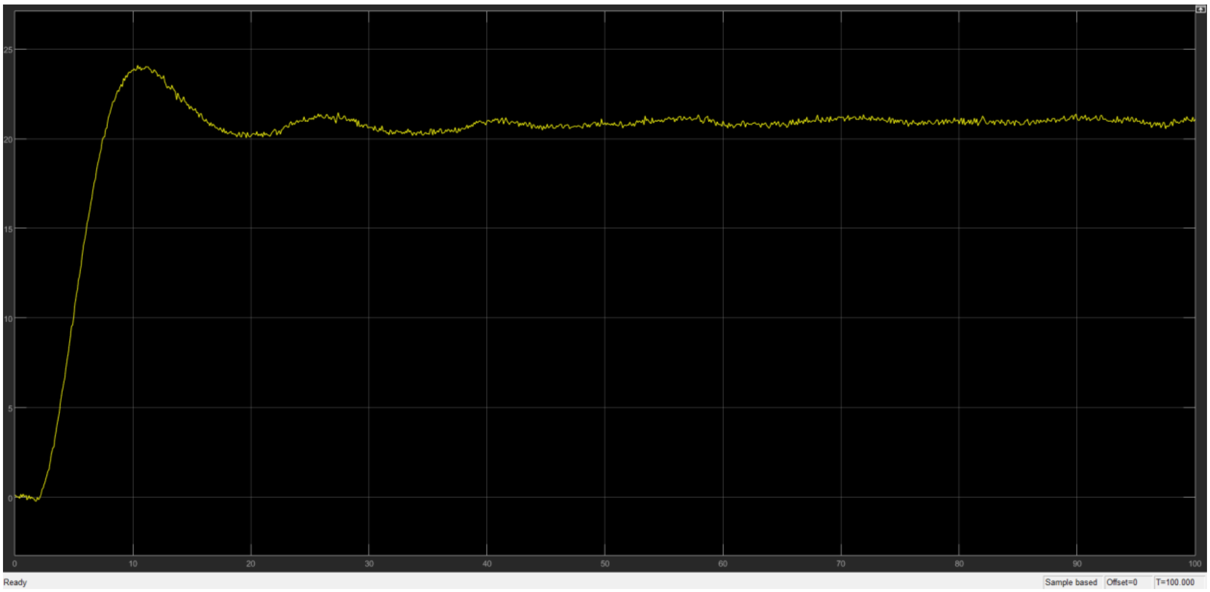


Figure: step response of the closed loop system with $K_C = 0.7878$

Conclusion: Closed loop system with controller from part b) is unstable as shown earlier. We see that the offset with controller from part a) (~55) is much greater than controller from part c) (~21). So we conclude that controller proposed by utilizing Pade's second order approximation has proven to be more effective in dealing with the actual system.

<u>Code:</u>

```
clear; close all;
%% Setup the system
s = tf('s');
Gp = 2*(s+2)/(s^2+2*s-3)*exp(-s);
%% 3a
Gp_pade = 2*(2-s)/(s^2+2*s-3);
f = @(s)(2*(2-s)/(s^2+2*s-3));
Kc_a = -1/(f(-0.2));
poles_parta = pole(1/(1+Kc_a*Gp_pade))
%% 3c
Gp_pade_second = 2*(s+2)*(1-s/2+s^2/8)/((s^2+2*s-3)*((1+s/2+s^2/8)));
rltool(Gp_pade_second)
%0.7878
```