EE324 Control Systems Lab

Problem Sheet 7

Prakhar Diwan, Roll no. 180100083

1. Problem 1

We are given a closed loop system with negative unity feedback with open loop transfer function as follows:

$$G_{open}(s) = \frac{1}{(s+4)(s+3)(s+12)}$$

For obtaining the desired specifications, a PI controller is used with a transfer function:

$$C(s) = \frac{K(s+z)}{s}$$

1.1. 1a

To reach a damping ratio of 0.2 for an initial value of z = 0.01, we obtain the intersection of the locus of constant damping ratio (= 0.2) and root locus plot of the system shown below at open loop gain K = 666.3.

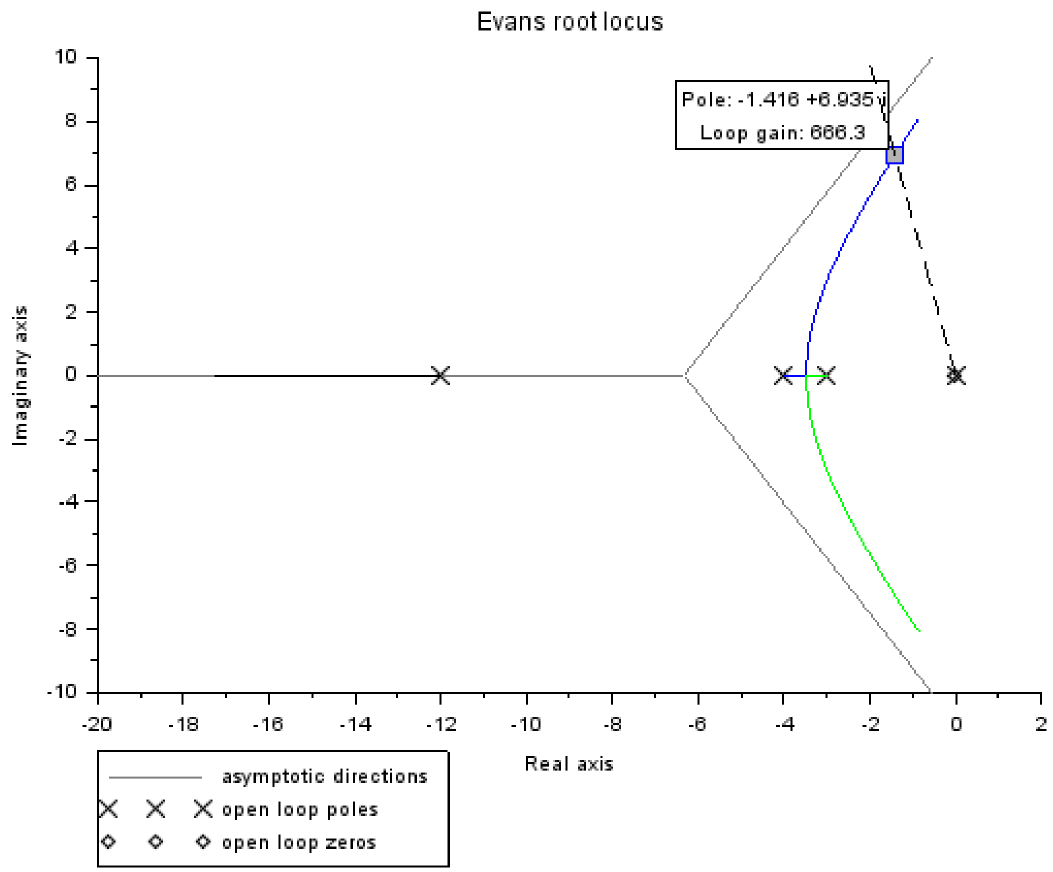

Figure 1: Root-Locus of system and locus of constant damping ratio = 0.2

Therefore, PI controller to be used is:

$$C(s) = \frac{666.3(s+0.01)}{s}$$

```
clear;
clc;
s=poly(0,'s');
z = 0.01;
G = 1/((s+3)*(s+4)*(s+12));
C = (s+z)/s;

H = C*G;
sysH = syslin('c',H);

rho=0.2; // Damping ratio reqd
theta=atan(sqrt(1-rho^2)/rho); //Angle made for given rho
a=[0:0.01:10];
fig=scf();
evans(sysH, 1000);
x=-cos(theta)*a;
y=sin(theta)*a;
plot(x, y, 'k--');
```

1.2. 1b

To obtain undamped natural frequency (ω_n) of 8 rad/s and 9 rad/s for an initial value of z = 0.01, we obtain the intersection of the locii of constant undamped natural frequency and root locus plot of the system.

For $\omega_n = 8\ rad/s$, we obtain open loop gain K = 953.3 as shown below:

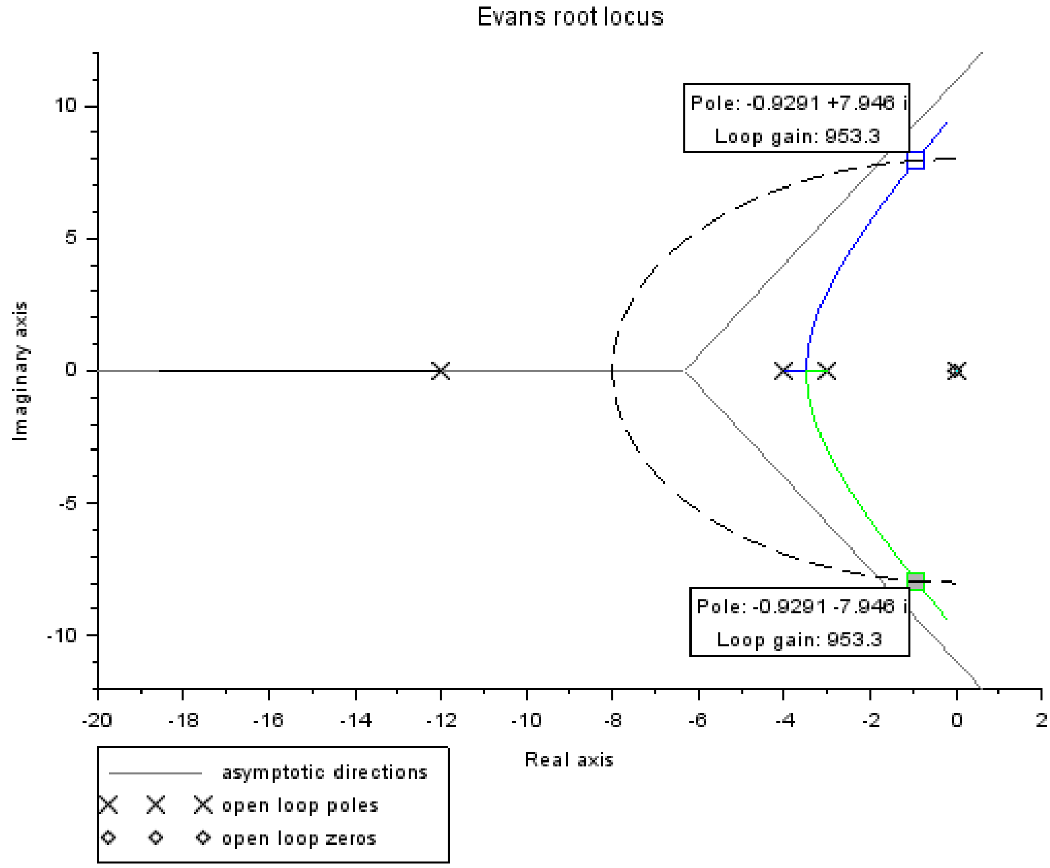

Figure 2: Root-Locus of system and locus of constant $\omega_n = 8\ rad/s$

Therefore, PI controller to be used is:

$$C(s) = \frac{953.3(s+0.01)}{s}$$

For $\omega_n = 9\ rad/s$, we obtain open loop gain K = 1329 as shown below:

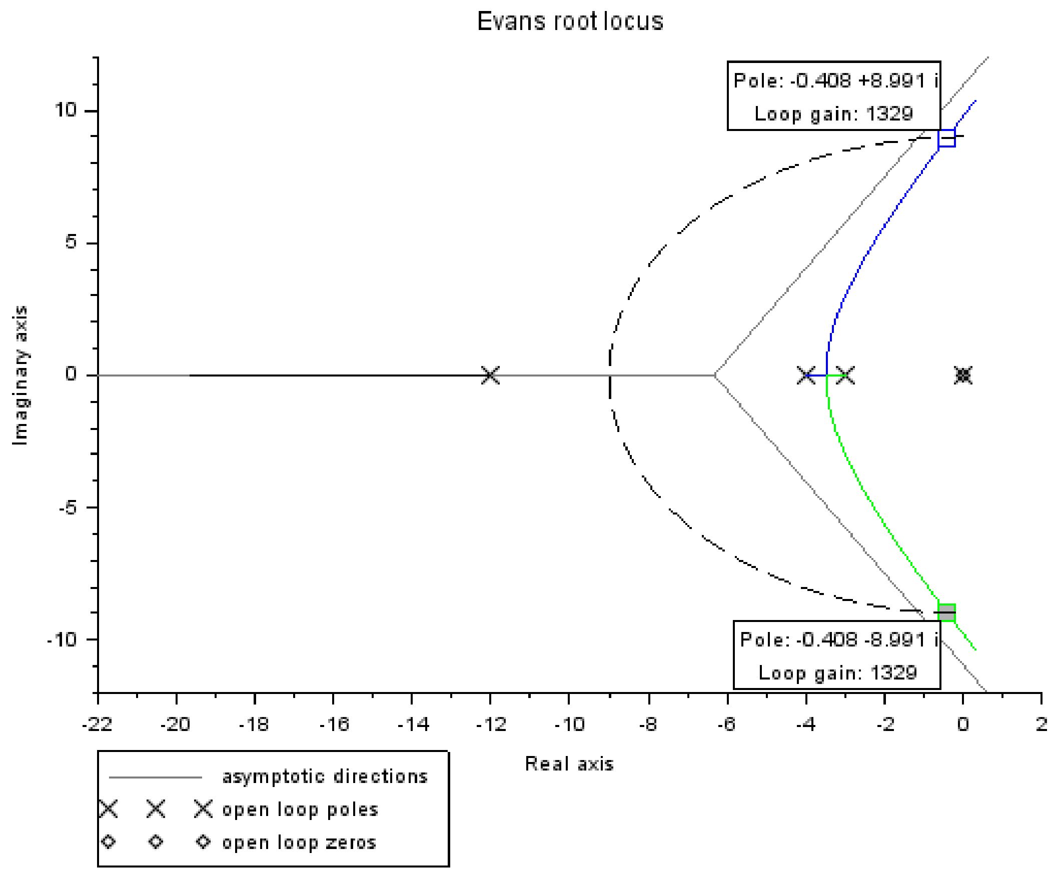

Figure 3: Root-Locus of system and locus of constant $\omega_n = 9\ rad/s$

Therefore, PI controller to be used is:

$$C(s) = \frac{1329(s+0.01)}{s}$$

```
clear;
clc;
s=poly(0,'s');
z = 0.01;
G = 1/((s+3)*(s+4)*(s+12));
C = (s+z)/s;

H = C*G;
sysH = syslin('c',H);

theta=[1.57:0.01:3*1.57];
a = 9; // a represents omega_n
fig=scf();
evans(sysH, 2000);
```

```
x=cos(theta)*a;
y=sin(theta)*a;
plot(x, y, 'k--');
```

1.3. 1c : Variation of Root locus plot with respect to z (zero of PI)

Upon varying z between 0.01 and 1.01 in steps of 0.1, we observe that as z increases the 2 branches (in blue and green) bend towards inside and the open loop zero location moves towards left on negative real axis.

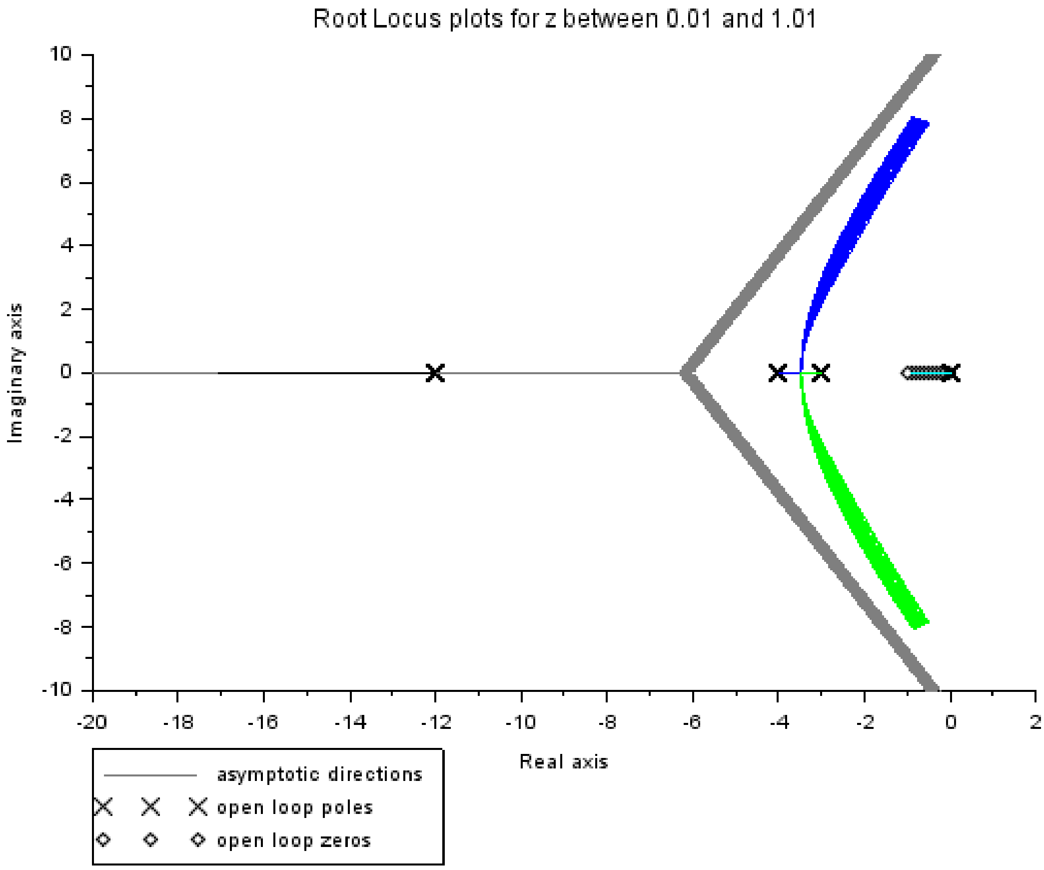

Figure 4: Root Locii of system for z between 0.01 to 1.01

Upon varying z between 1 and 5 in steps of 0.5, we observe that as z increases the 2 branches (in light blue and green) , shifts towards left untill zero crossed the pole in the denominator, upon which we obtain 2 branches (in green and blue).

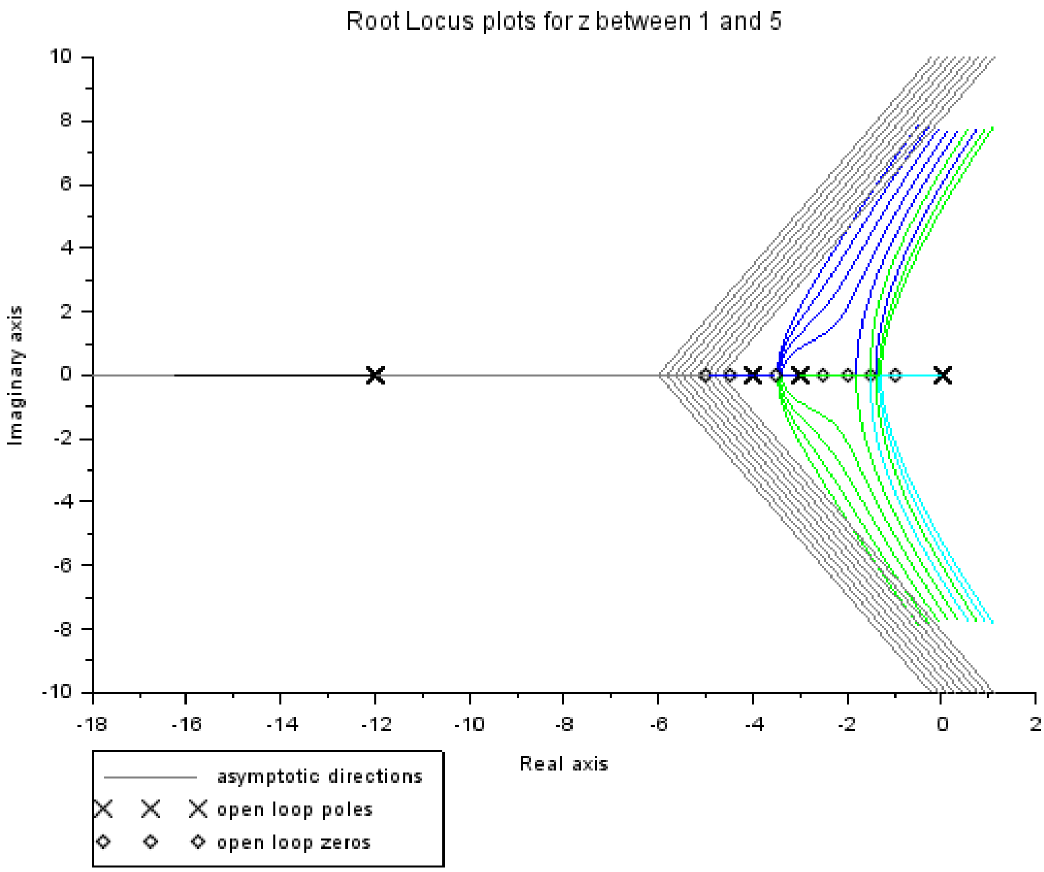

Figure 5: Root Locii of system for z between 1 to 5

Upon varying z between 5 and 105 in steps of 10, we observe that as z increases the 2 branches (in light blue and green) bend towards inside and the break away point on the right has shifted significantly as per the transition z value (due to zero crossing pole value) shown in above and the open loop zero location moves towards left on negative real axis. Also, 2 another branches (in black and blue) are seen here emerging the complex plane and shifting towards the left, upon increase in z value.

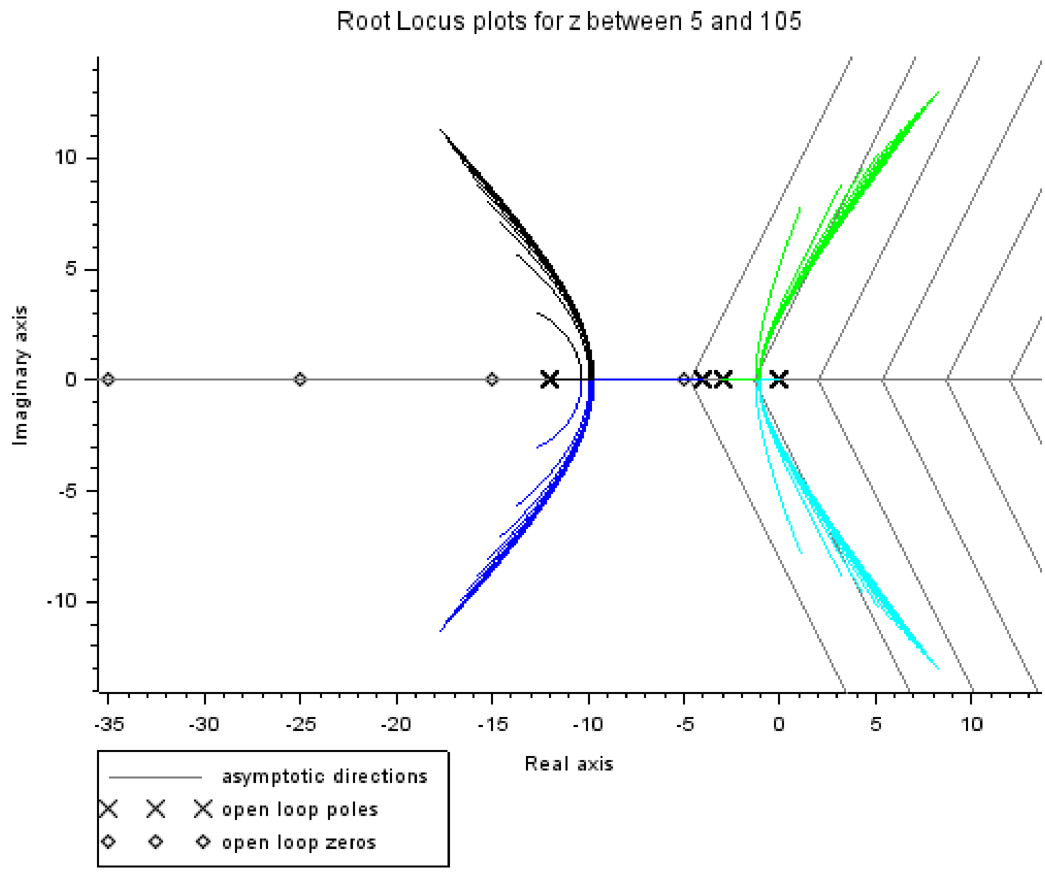

Figure 6: Root Locii of system for z between 5 to 105

```
clear;
clc;
s=poly(0,'s');

G = 1/((s+3)*(s+4)*(s+12));

fig=scf();

for z = 5:10:105
    C = (s+z)/s;
    H = C*G;
    sysH = syslin('c',H);
    evans(sysH, 1000);
end
xtitle ('Root Locus plots for z between 5 and 105');
```

1.4. 1d

Yes, it is possible to alter the pole locations of a system using a PI controller without changing
the damping ratio. The plot below shows two Root locii for z = 0.01 and 1.01; and a straight line, which is the locus of constant damping ratio = 0.8. At the intersection points of the RL plots with the straight line, we see that closed pole locations of system are different but damping ratio at both points is the same (=0.8).

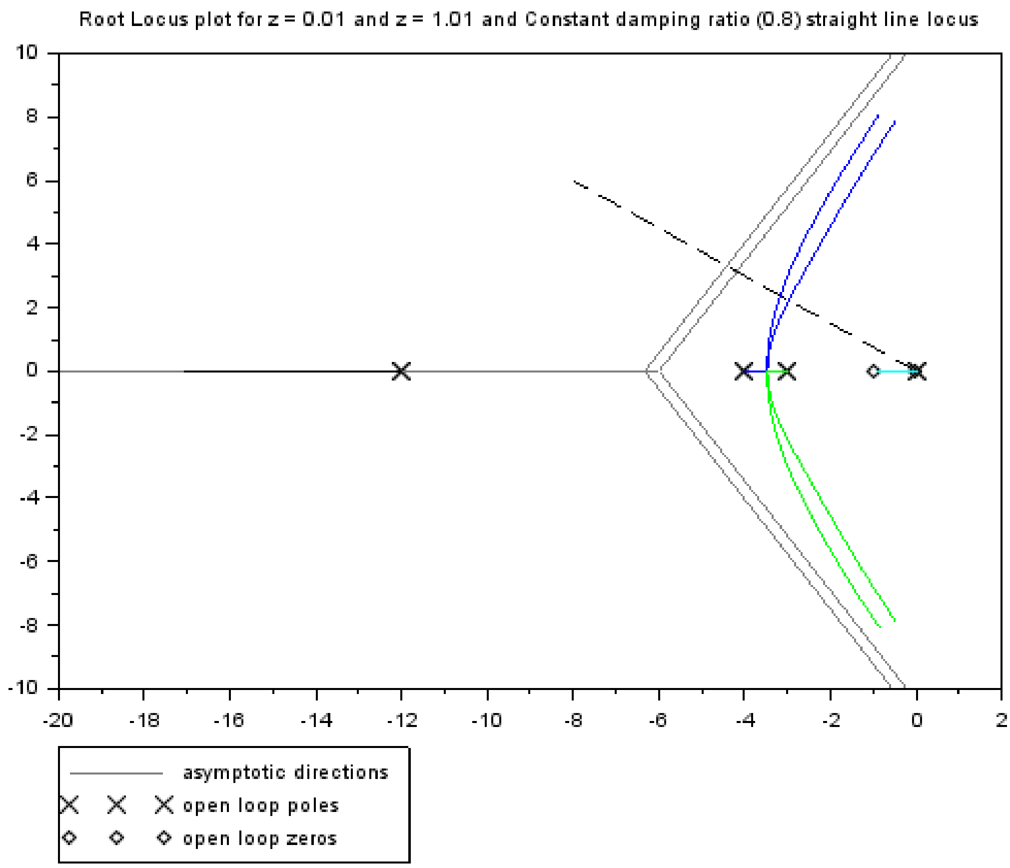

Figure 7: Root Locus for z = 0.01 and 1.01 and damping ratio = 0.8

```
clear;
clc;
s=poly(0,'s');

G = 1/((s+3)*(s+4)*(s+12));
rho=0.8; // Damping ratio reqd
theta=atan(sqrt(1-rho^2)/rho); //Angle made for given rho
a=[0:0.01:10];
fig=scf();
x=-cos(theta)*a;
```

```
y=sin(theta)*a;
plot(x, y, 'k--');
for z = 0.01:1:1.01
    C = (s+z)/s;
    H = C*G;
    sysH = syslin('c',H);
    evans(sysH, 1000);
end
xtitle ('Root Locus plot for z = 0.01 and z = 1.01 and Constant
damping ratio (0.8) straight line locus');
```

2. Problem 2

We are given a closed loop system with negative unity feedback with open loop transfer function as follows:

$$G_{open}(s) \;=\; \frac{1}{(s+2)(s+1)}$$

For obtaining the desired specifications, a lag compensator is used with a transfer function:

$$C(s) \;=\; \frac{K(s+z)}{(s+p)}\; with\; |z/p| \;=20$$

2.1. 2a

Constant gain K to achieve 10% OS in the closed-loop. We obtain the locus of 10% OS as follows:

It is a straight line with angle of $tan^{-1}\left(\frac{\sqrt{1-\rho^2}}{\rho}\right)$ from the imaginary axis. Here we see:

$$\frac{\sqrt{1-\rho^2}}{\rho} \;=\; \frac{\pi}{ln(10)}$$

Therefore we obtain the open loop contant gain K = 4.44 as is shown by the intersection in plot below:

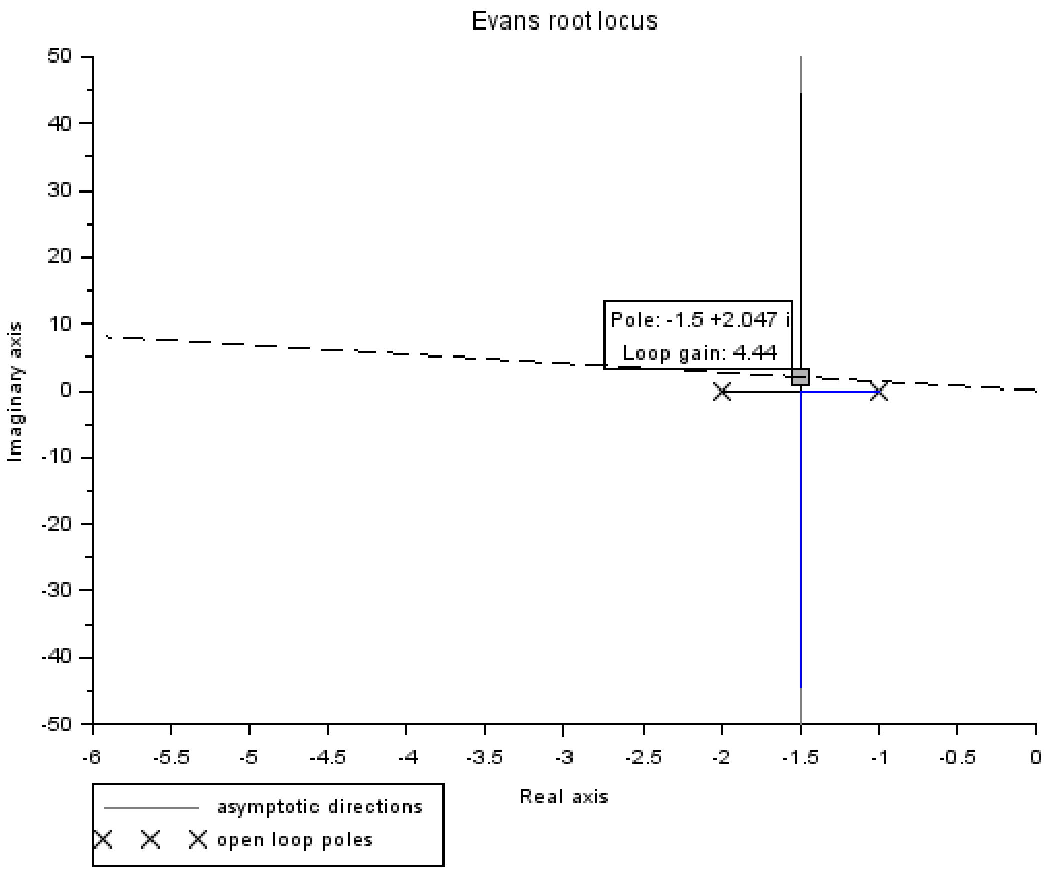

Figure 8: Root Locus for 2a

```
clear;
clc;
s=poly(0,'s');
G = 1/((s+1)*(s+2));
sysG = syslin('c',G);
pi = 3.1415;
// 10 % OS
theta=atan((pi/log(10))); //Angle made for given rho
a=[0:0.01:10];
fig=scf();
evans(sysG, 2000);
x=-cos(theta)*a;
y=sin(theta)*a;
plot(x, y, 'k--');
```

2.2. 2b

The steady state error of the above system upon step input comes out be = 1-0.689 = 0.311

The unit step response is shown below:

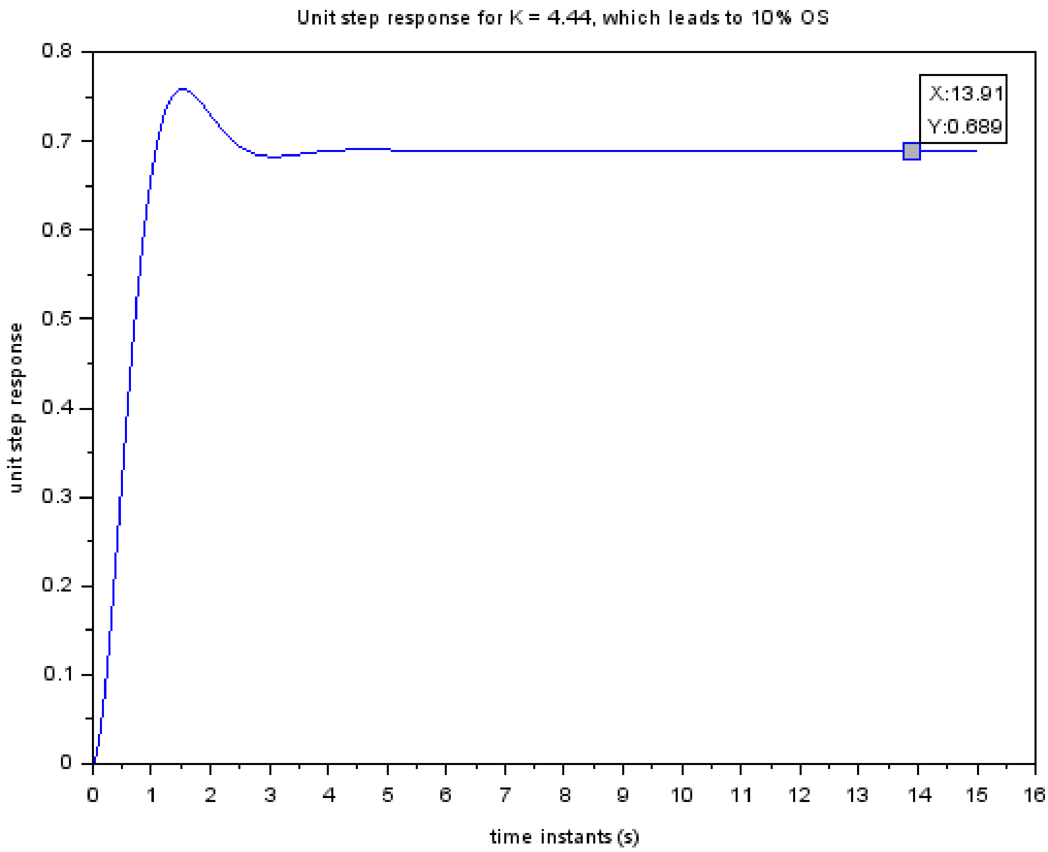

Figure 9: Step response for 2b without Lag compensator

After placing the lag compensator, we obtain the following step response, which gives steady state error = 1- 0.978 = 0.022

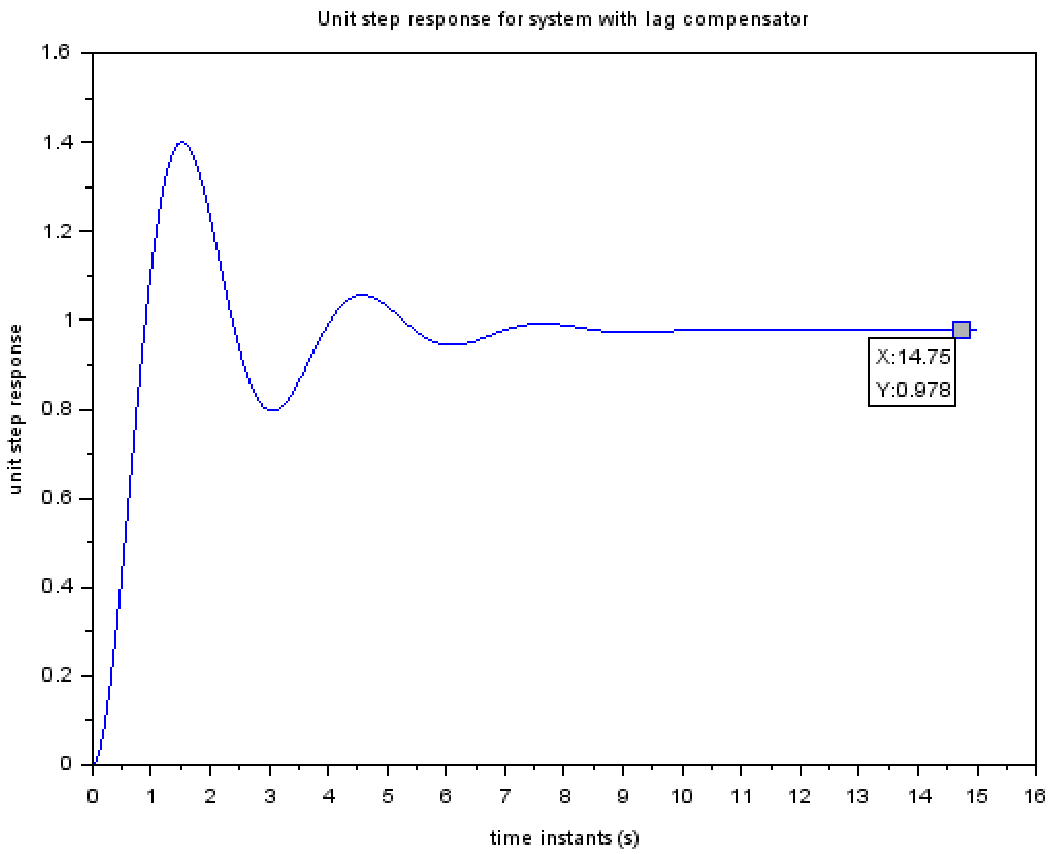

Figure 10: Step response for 2b with Lag compensator

Lag compensator used had the following trransfer function:

$$C(s) = 4.44\left(\frac{s+2}{s+0.1}\right)$$

```
clear;
clc;
s=poly(0,'s');
z = 0.01;
G = 1/((s+1)*(s+2));
sysG = syslin('c',G);
pi = 3.1415;
//from  10 % OS we get k = 4.44
k = 4.44;
t=0:0.01:15;
sysT = syslin('c',k*G/(1+k*G));
step_r = csim('step',t,sysT);
fig=scf();
```

```
plot(t,step_r);
xtitle ('Unit step response for K = 4.44, which leads to 10%
OS',"time instants (s)" , "unit step response" );
```

2.3. 2c

Upon changing the pole-zero location while maintaining the ratio, we observe the following plot. When z is increased from 0.1 to 4.1, the damping ratio decreases and %OS increase as well.
And the lag compensator takes lesser time to come into action as z values increases. All the systems are stable though, in this region of z.

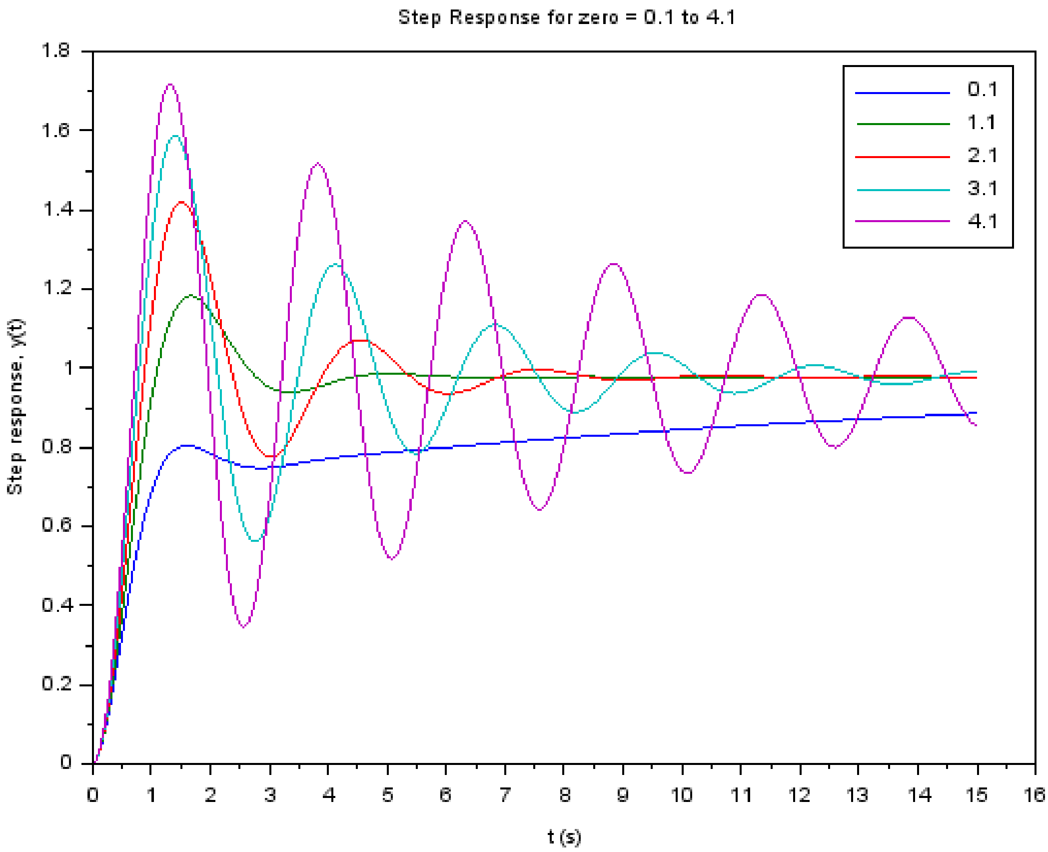

Figure 11: Step response for $0.1 \leqslant z \leqslant 4.1$:

When zero is increased from 1 to 51 we observe that system becomes unstable as z increases beyond 5.1. When z is increased from 1 to 51, the %OS increase and system becomes unstable after z crosses 5.1. And the lag compensator takes lesser time to come into action as z values increases. The plot is shown below:

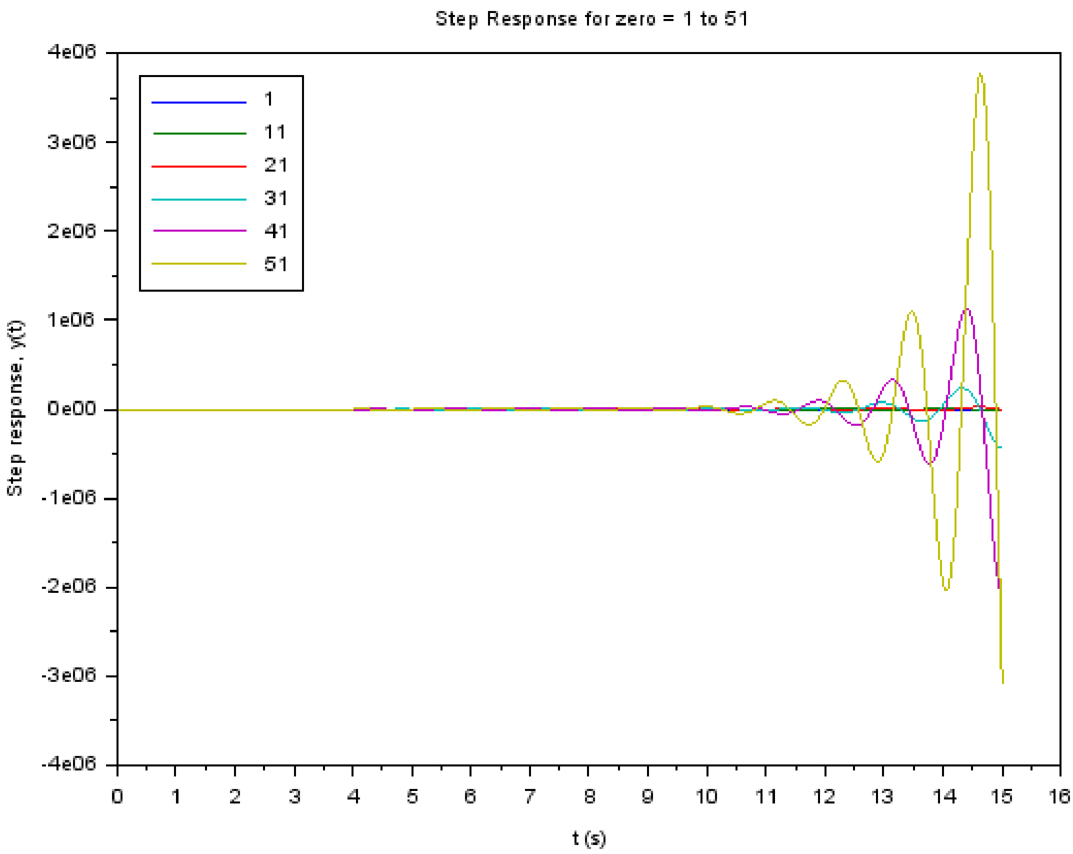

Figure 12: Step response for $1 \leqslant z \leqslant 51$

Therefore, we observe that system becomes unstable or %OS increases (in case of stable systems), as the z value is increases, and as a tradeoff we get a better response time from lag compensator.

```
clear;
clc;
s=poly(0,'s');
z = 0.01;
G = 1/((s+1)*(s+2));
sysG = syslin('c',G);
pi = 3.1415;
//from  10 % OS we get k = 4.44
k = 4.44;
t=0:0.01:15;
fig=scf();
z = 0.1 : 1 : 4.1;
y = zeros(length(t),length(z));
i=1;
```

```
for z1 = z
    p = z1/20;
    C = (s+z1)/(s+p);
    H = C*G;
    sysT = syslin('c',k*H/(1+k*H));
    y(:,i) = csim('step', t, sysT);
    i=i+1;
end
plot(t, y);
h1=legend(string(z),5);
xlabel("t (s)")
ylabel("Step response, y(t)")
title("Step Response for zero = 0.1 to 4.1");
f = gcf();
f.background = 8;
```

3. Problem 3

We are given open loop transfer function G(s) as follows:

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

3.1. 3a

Need to design a lead compensator for G(s) to obtain half 2% settling time of that in Q2-a. For 2-a, the settiling time was 2.33 s. Therefore required settling time = halving the settling time we can double the magnitude of Re(pole) as:
For 10% OS we obtain damping ratio (ρ) as:

$$\rho = \sqrt{\frac{ln(10)}{ln(10) + \pi^2}} = 0.435$$

$$2\%\ settling\ time = \frac{-ln\left(0.02\sqrt{1-\rho^2}\right)}{|Re(pole)|}$$

$$Re(pole) = -3.448$$

Therefore we see that Re(pole) reqired is -3.448 (as assumed stable system, while applying the formula). Intersection with required %OS = 10 locus leads to the following plot:

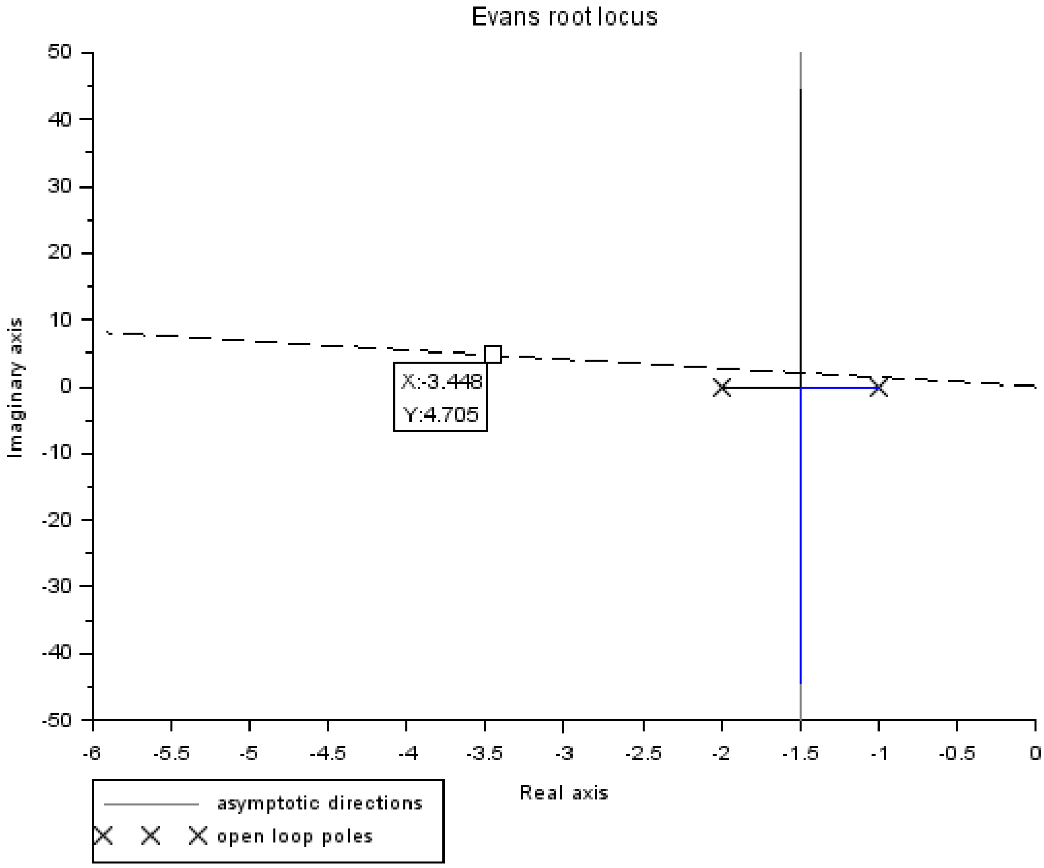

Figure 13: RL plot of 3a and intersection of locci of constant t_s and 10% OS

Therefore we get 2 pole values of the desired system as: $-3.448 \pm 4.705j$

Now upon changing the varying the z and p values in the lead compesator transfer function shown below:

$$C(s) = K\left(\frac{s+z}{s+p}\right)$$

We obtain p = 9.318, z = 4 and K = 41.47 for having pole values as $-3.448 \pm 4.705j$. The plot is shown below:

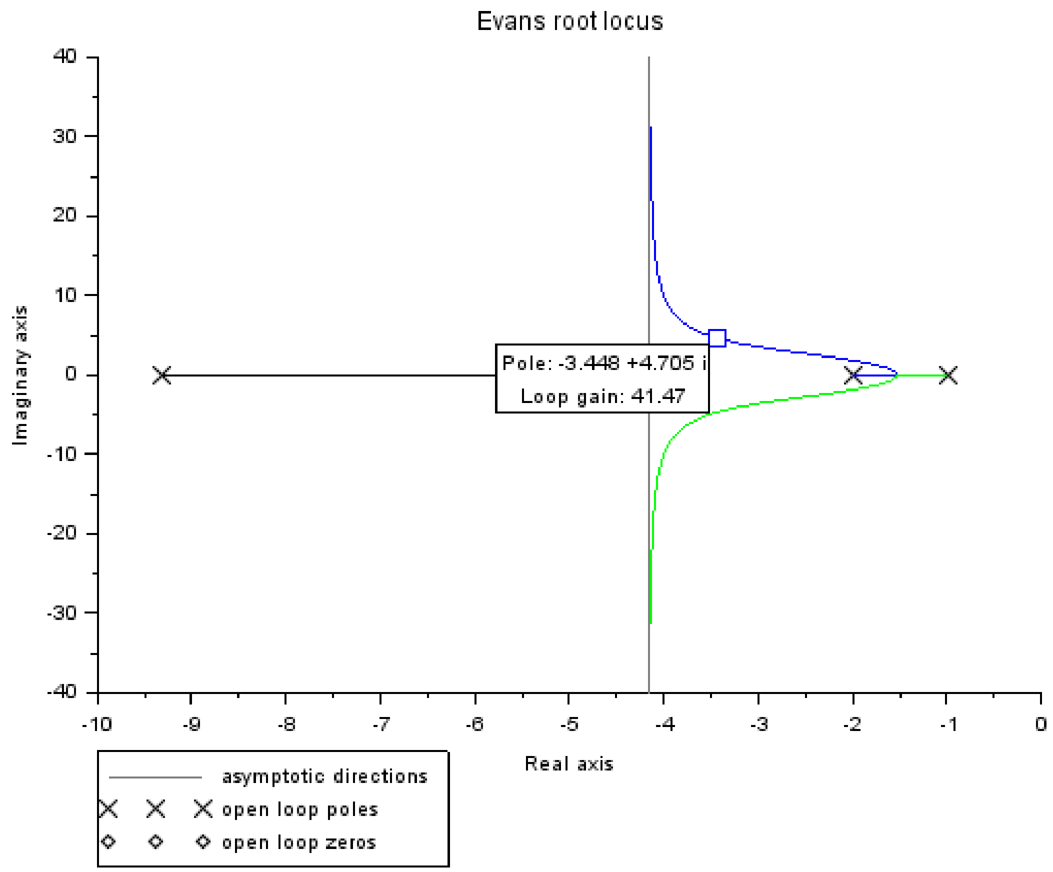

Figure 14: RL plot of the obtained system

Therefore, required lead compensator is as follows:

$$C(s) = 41.47\left(\frac{s+4}{s+9.318}\right)$$

```
clear;
clc;
s=poly(0,'s');
z = 0.01;
G = 1/((s+1)*(s+2));
sysG = syslin('c',G);
pi = 3.1415;
// 10 % OS
theta=atan((pi/log(10))); //Angle made for given rho
a=[0:0.01:10];
fig=scf();
evans(sysG, 2000);
x=-cos(theta)*a;
```

```
y=sin(theta)*a;
plot(x, y, 'k--');
// half t_s as 2a implies Re(Pole) = -3.4483 and upon intersection of
locii we get point as -3.4483+4.705j

z = 4;
p = 9.318;
C = (s+z)/(s+p);
H = C*G;
sysH = syslin('c',H);
fig=scf();
evans(sysH,1000);
```

3.2. 3b

Same specifications as above with a PD controller C(s) specified below:

$$C(s) = K(s+z)$$

Again 2 pole values of the desired system as: $-3.448 \pm 4.705j$, hence upon varying values of z and K for having the above 2 points on RL plot, we obtain z = 8.223 and K = 3.897.

Therefore, required PD controller has following transfer function:

$$C(s) = 3.897(s+8.223)$$

The plot with required pole labeled is shown below:

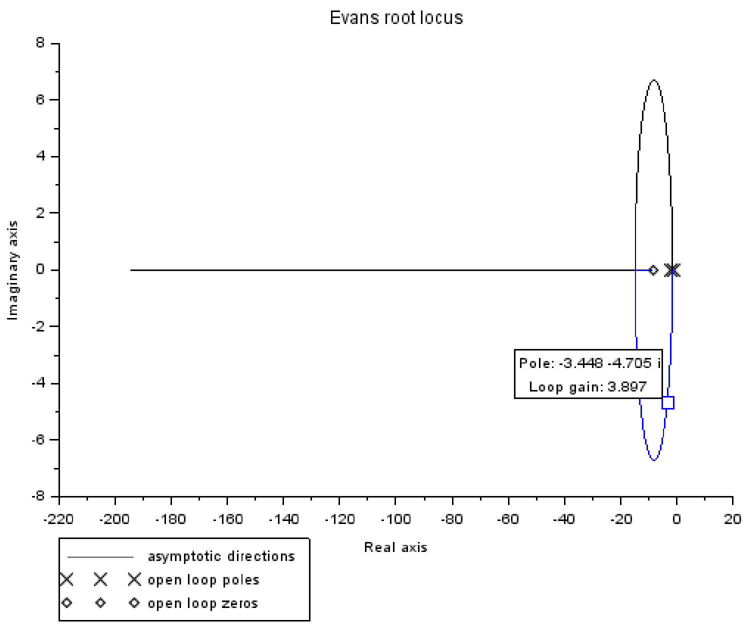

Figure 15: Root locus for Obtained system

```
clear;
```

```
clc;
s=poly(0,'s');
z = 0.01;
G = 1/((s+1)*(s+2));
sysG = syslin('c',G);
pi = 3.1415;
// 10 % OS
theta=atan((pi/log(10))); //Angle made for given rho
a=[0:0.01:10];
fig=scf();
evans(sysG, 2000);
x=-cos(theta)*a;
y=sin(theta)*a;
plot(x, y, 'k--');
// half t_s as 2a implies Re(Pole) = -3 and upon intersection of
locii we get point as -3+4.094j
z = 8.223;
C = (s+z);
H = C*G;
sysH = syslin('c',H);
fig=scf();
evans(sysH, 200);
```

4. Problem 4

4.1. 4a

Given transfer function:

$$G(s) = \frac{1}{s^2 + 5s + 6}$$

Upon giving the input of sine function ($sin(\omega t)$) to the above system, the following input-output plots are obtained for $\omega = 0.1,\ 1,\ 2.5,\ 7.5\ and\ 15\ rad/s$.

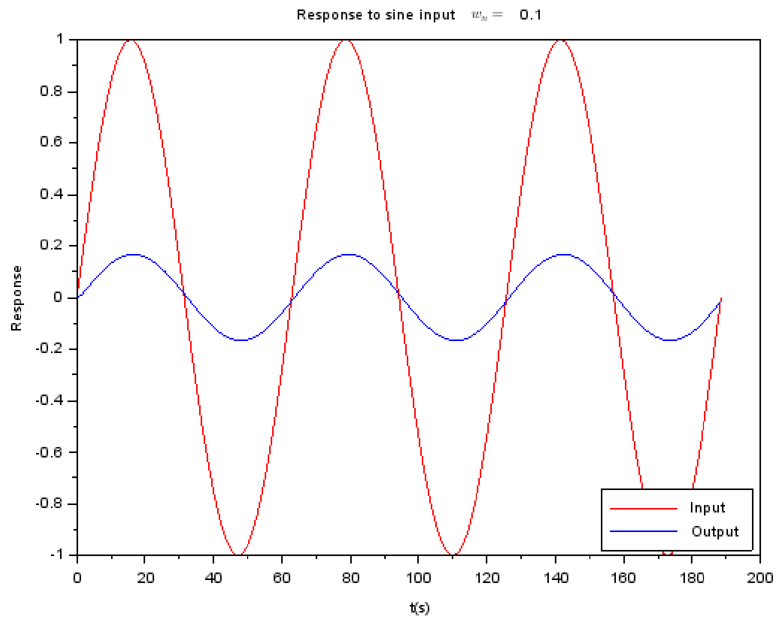

Figure 16: Response to sin(0.1*t)

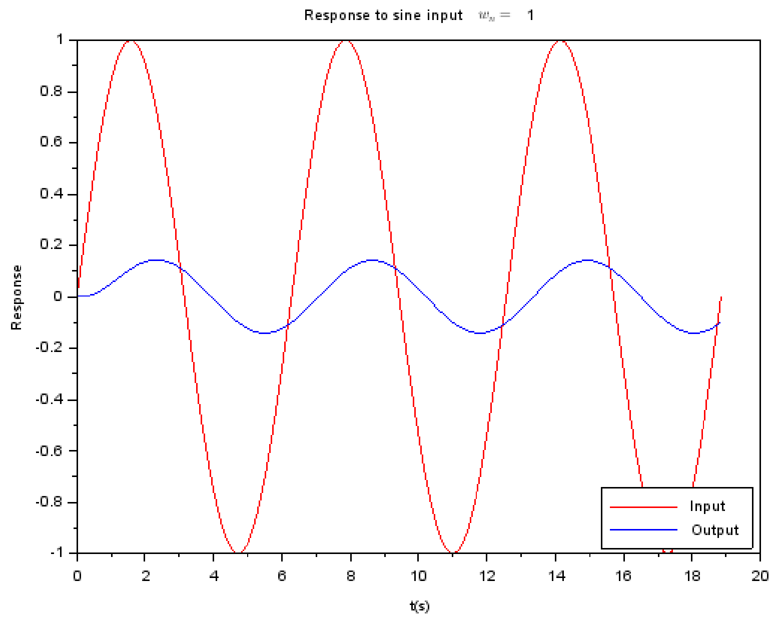

Figure 17: Response to sin(t)

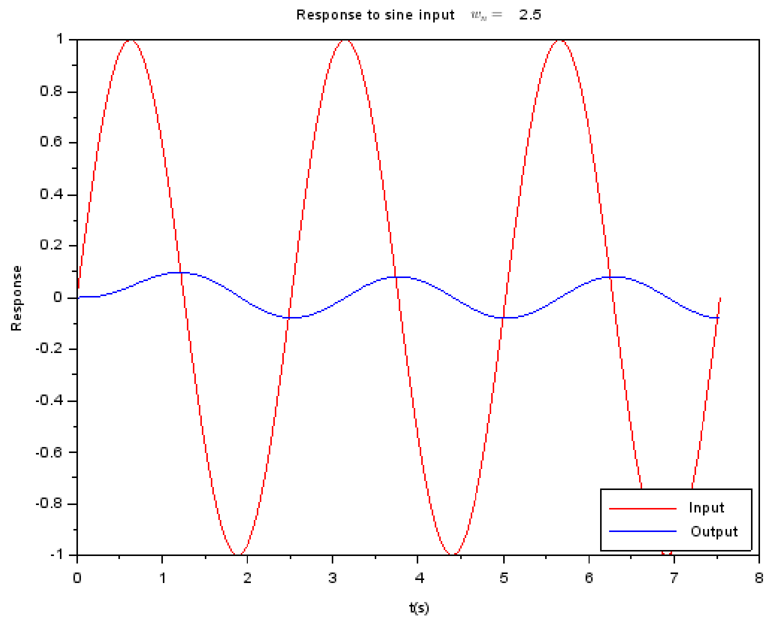

Figure 18: Response to sin(2.5*t)

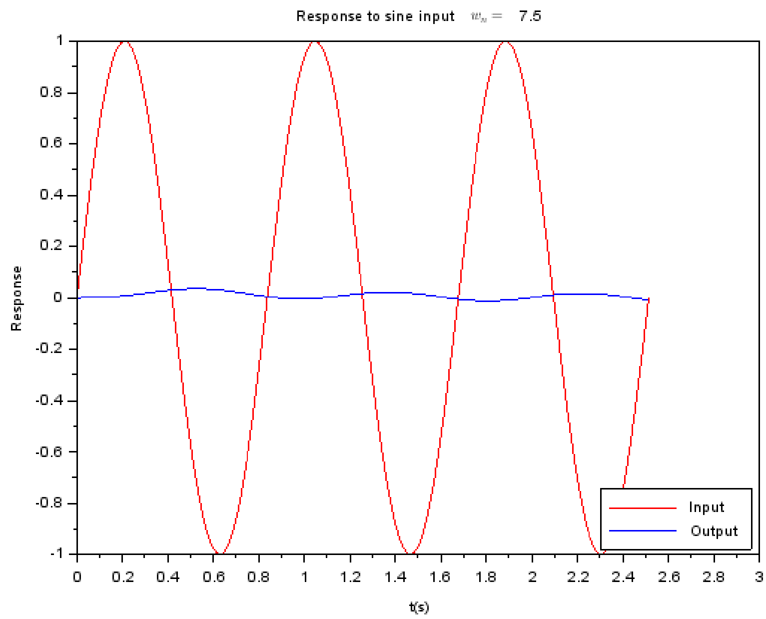

Figure 19: Response to sin(7.5*t)

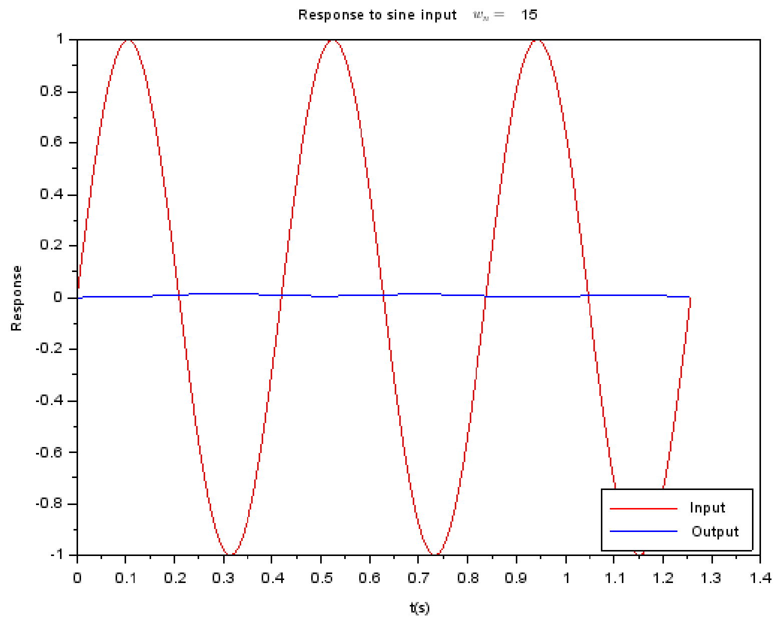

Figure 20: Response to sin(15*t)

The variation of phase with ω is shown below:

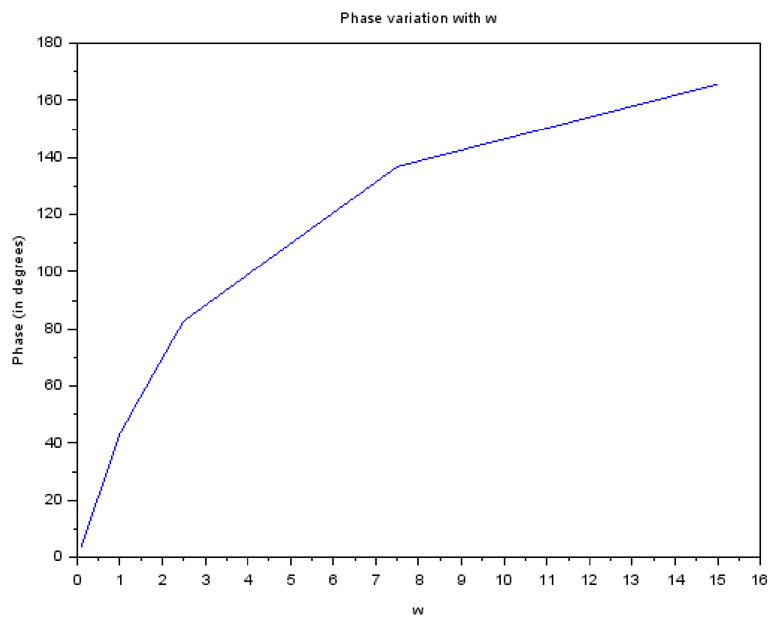

Figure 21: Variation of phase difference (in $^\circ$) with ω

Here, we observe that phase difference at different ω matches the value of angle of $G(j\omega)$ given below:

$$G(j\omega) = \frac{1}{-\omega^2 + 5j\omega + 6}$$

The variation of magnitude with ω is shown below:

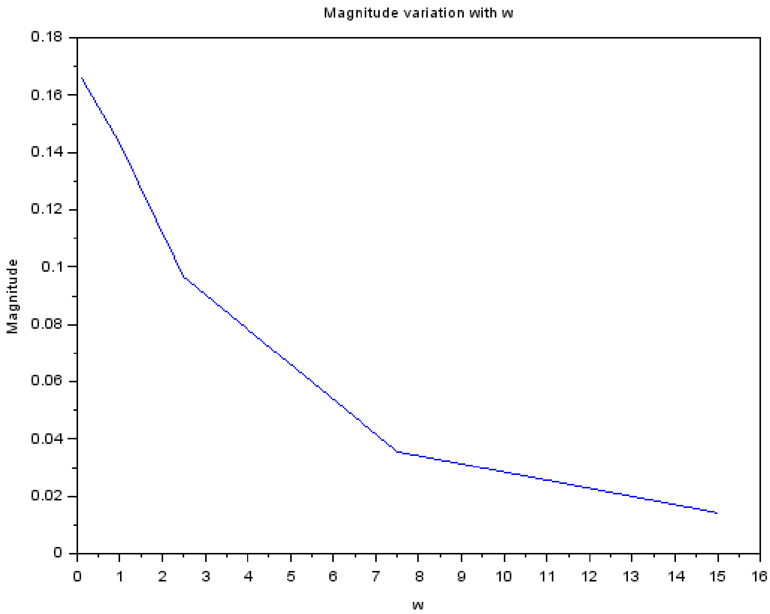

Figure 22: Variation of magnitude with ω

Here, we observe that ratio of magnitude of output to input at different ω, varies as $|G(j\omega)|$, which is calculated below:

$$G(j\omega) = \frac{1}{-\omega^2 + 5j\omega + 6}$$
$$|G(j\omega)| = \frac{1}{\sqrt{\omega^4 + 13\omega^2 + 36}}$$

```
clear;
clc;
s=poly(0, 's')

G = 1/(s^2+5*s+6);
G_sys = syslin('c', G);

w_list=[0.1,1,2.5 ,7.5, 15];
phases=zeros(w_list);
mags=zeros(w_list);
i=1;
for w=w_list
    t=0:2*%pi/(w*100):6*%pi/w;
    x=sin(w*t);
```

```
    y=csim(x, t, G_sys);
    ty=t(find(abs(y-max(y))<0.00000001)(1));
    tx=t(find(abs(x-max(x))<0.00000001)(1));
    phases(i)=(ty-tx)*w*180.0/%pi; // degrees
    mags(i)=max(y);
    fig=scf(i);
    plot(t, x, 'r');
    plot(t, y, 'b');
    hl=legend(['Input', 'Output'], [4]);
    xtitle(['Response to sine input', '$w_n=$', string(w)], 't(s)',
'Response');
    i=i+1;
end
fig=scf(7);
plot('ln', w_list, phases);
xtitle('Phase variation with w', 'w', 'Phase (in degrees)');
fig=scf(8);
plot('ln', w_list, mags);
xtitle('Magnitude variation with w', 'w', 'Magnitude');
```

4.2. 4b

The desired relation (between phase difference and angle of G(jω) is for frequency measured in rad/s and not Hz.

4.3. 4c

Given transfer function:

$$G(s) = \frac{1}{s^3 + 6s^2 + 11s + 6}$$

Upon giving the input of sine function ($sin(\omega t)$) to the above system, the following input-output plots are obtained for $\omega = 0.1,\ 1,\ 2.5,\ 7.5\ and\ 15\ rad/s$.

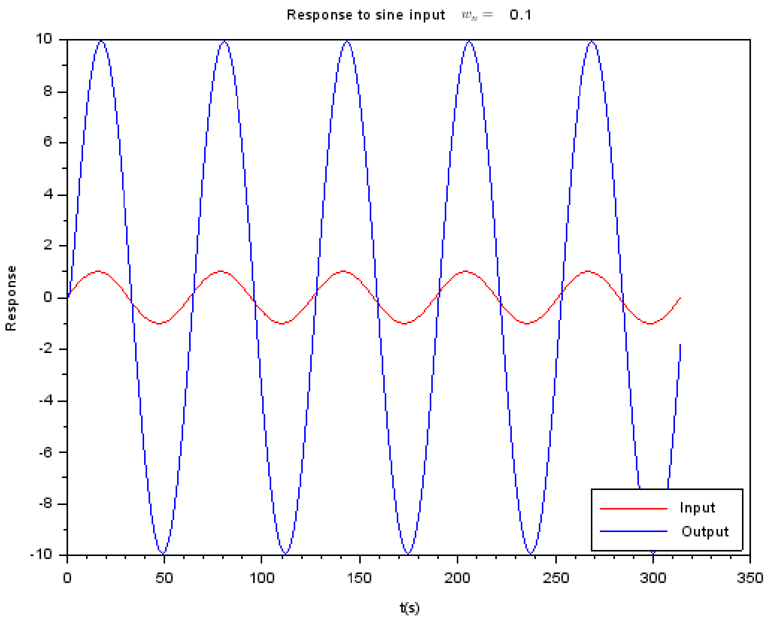

Figure 23: Response to sin(0.1*t)

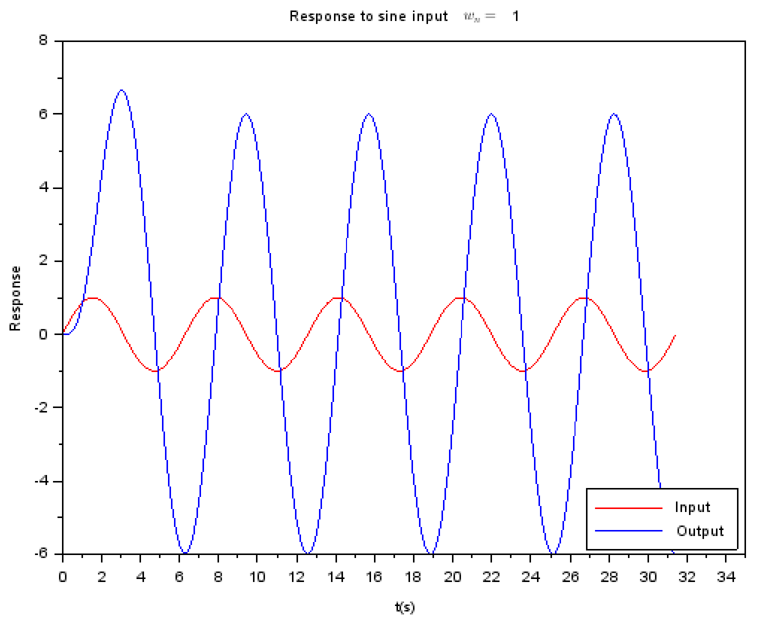

Figure 24: Response to sin(t)

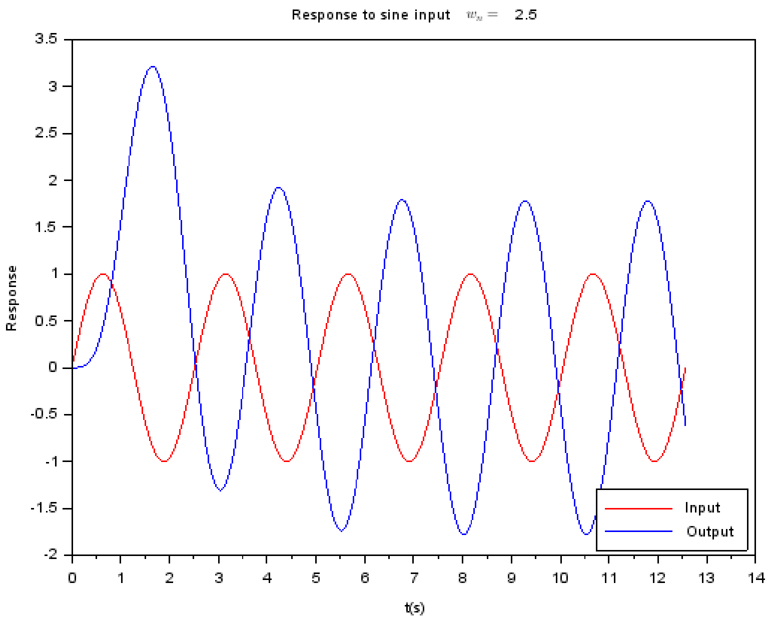

Figure 25: Response to sin(2.5*t)

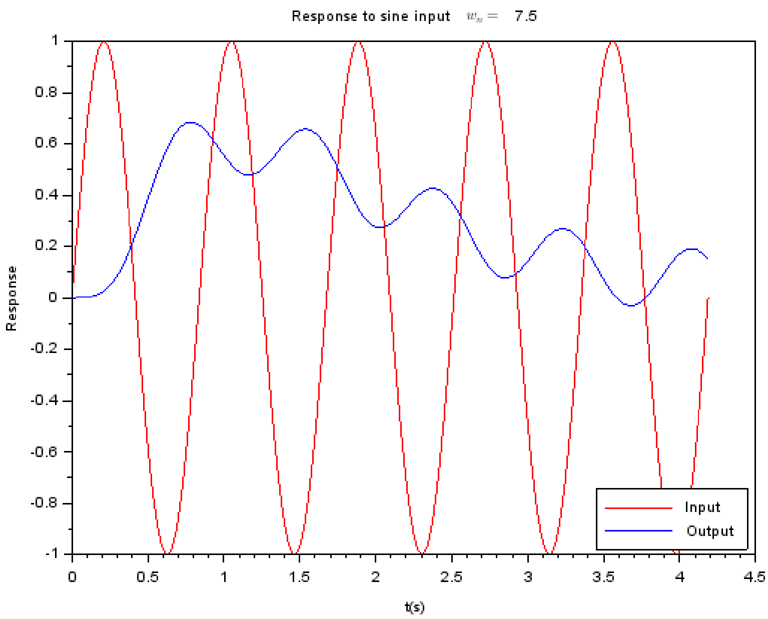

Figure 26: Response to sin(7.5*t)

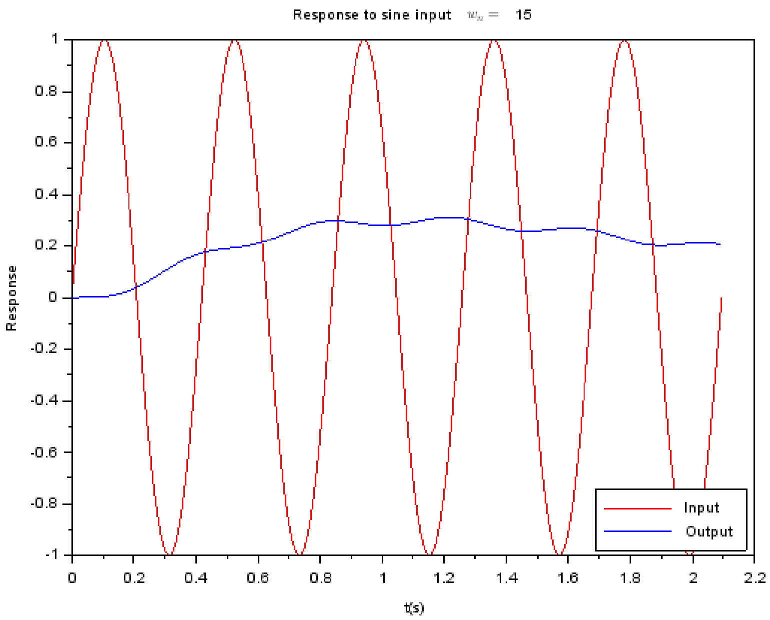

Figure 27: Response to sin(15*t)

Here, as well we observe that ratio of magnitudes matches the $|G(j\omega)|$ and phase difference matches $\angle G(j\omega)$.

The $180°$ phase difference happens at $\omega = \sqrt{11} = 3.316\ rad/s$ as the phase difference theoretically is given by $tan^{-1}\left(\dfrac{11\omega - \omega 3}{6 - 6\omega^2}\right)$. We also see using hit and trial that near f = 0.52 Hz. the phase difference is 180 degrees.

The numerator 60 does not play a role in finding this argument (of finding the frequency for which
we have 180 degrees phase difference between input and output).

```
clc;
clear;
s=poly(0, 's')

G = 6/(s^3 + 6*s^2+11*s+6);
G_sys = syslin('c', G);

w_list=[0.1,1,2.5 ,7.5, 15];
phases=zeros(w_list);
mags=zeros(w_list);
i=1;
for w=w_list
    t=0:2*%pi/(w*100):6*%pi/w;
    x=sin(w*t);
```

```
    y=csim(x, t, G_sys);
    ty=t(find(abs(y-max(y))<0.00000001)(1));
    tx=t(find(abs(x-max(x))<0.00000001)(1));
    phases(i)=(ty-tx)*w*180.0/%pi; // degrees
    mags(i)=max(y);
    fig=scf(i);
    plot(t, x, 'r');
    plot(t, y, 'b');
    hl=legend(['Input', 'Output'], [4]);
    xtitle(['Response to sine input', '$w_n=$', string(w)], 't(s)',
'Response');
    i=i+1;
end
```