**EE324 Control Systems Lab**

Problem sheet 6

Tarun Sai Goddu | 190070024

**Question 1**

**1.a** We are given G(s), as the open loop transfer function of a system with unity negative feedback. And a proportional controller K is applied to the given system.

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

* 1. For designing a P controller to obtain a steady state error of 0.489 on applying step input, we obtain the gain value(K) as Follows:

e(∞) = lim sE(s) → 0 ( )

sE(s) = sR(s) / (1 + KG(s)); R(s) = 1 /s

Therefore, we obtain:

e(∞) = 1 / (1 + KG(0)) = 0.489

which upon solving leads to K = (144\*511) / 489.

We observe that the steady state value is 0.511, which hence has an steady state error

= 0.489 (1-0.511) as required.

Scilab Code for the same:

clear;

clc;

s=poly(0,'s');

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

sysG = syslin('c',G);

fig=scf();

evans(sysG);

k = ((144\*511)/489);

t=0:0.01:10;

sysT = syslin('c',k\*G/(1+k\*G));

step = csim('step',t,sysT);

fig=scf();

plot(t,step);

xtitle ('Unit step response for steady state error = 0.489',"time instants (s)" , "unit step response" );

* 1. b

To obtain a damping ratio 𝜌 = 0.35, we locate the intersection point of the root locus in Figure 1 From the root locus directly, we observe that at intersection at gain = 371.9 as shown in Figure 1 with one of the pole values also visible:

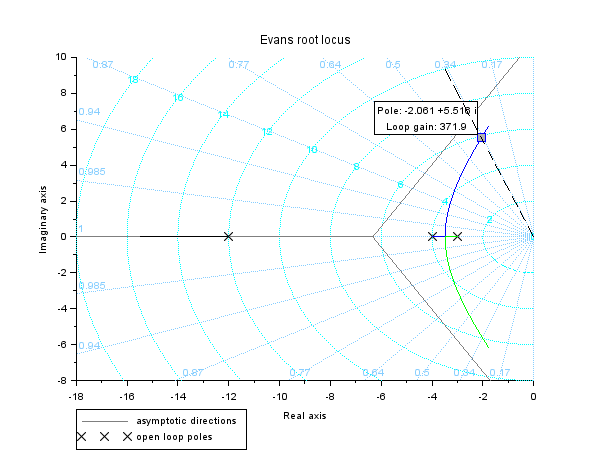
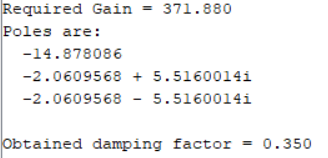
**

Figure 1: Intersection of RL and 𝜁 = 0.35 line

Iterating over values of K with maximum error 10 as to get a better estimate, we get . K = 371.88



We see that at this gain, we can approximate the system as a second order system {dominant pole approximation : as |Re(Far - away pole)| > |5\*Re(dominant pole)|} and hence we can define the damping ratio for the system, as equal to 0.35. The unit step respnse for the system is plotted below:

We observe there are 3 poles, and the system's poles lie in the LHP till about Critical gain = 2. Beyond that, the system becomes unstable.

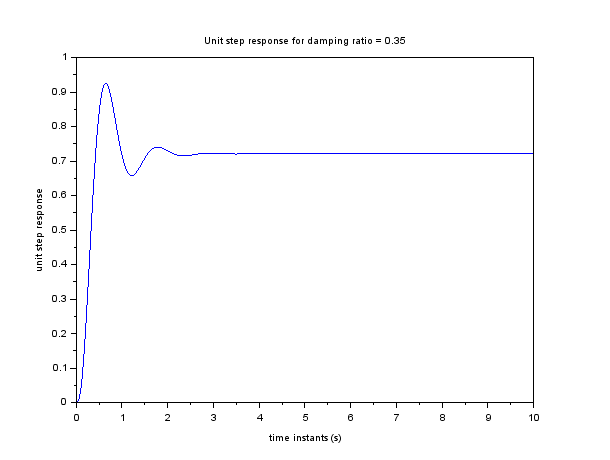


Figure2: Unit step response of system with damping ratio 𝜌 = 0.35

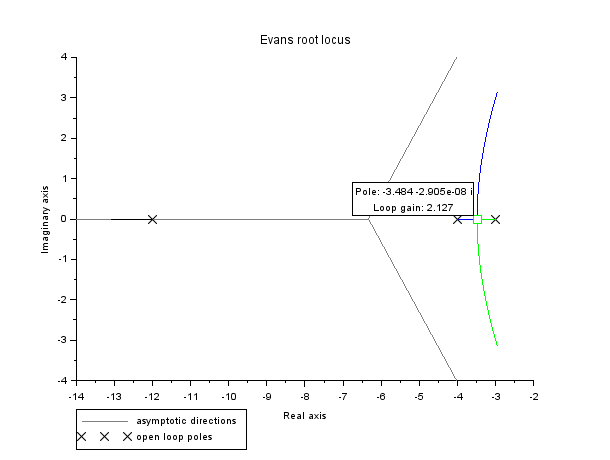
1.c The gain value at the break away point was obtained as K = 2.127. It was obtained to be same from numerical calculations. The RL plot with the break away point marked is shown below: 

Figure 3: : Root Locus for 1c

clear;

clc;

s=poly(0,'s');

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

sysG = syslin('c',G);

fig=scf();

evans(sysG);

**1.d** Given the open loop system, upon increasing the controller gain K in a small range of 0.1 to 1 in steps of 0.15. The following step responses were obtained for the above values of K:

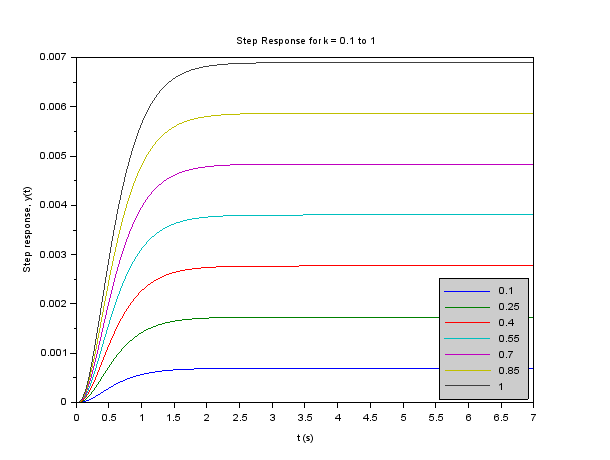


Figure 4

Figure 4: Unit step responses of Closed loop system with K varying between 0 and 1

1. Closed loop poles: All of them lie on the open left half plane as the system reponses observed are stable. Due to the no oscillations in responses for ∀K < 2.127 , which is the break-away point as observed in part (c), we conclude a response similar to overdamped type, which implies all poles are distinct and lie on negative real axis, for values of K between 0 and 1.
2. Steady state errors: Upon increase in gain value (0 ⩽ K ⩽ 1), steady state value increases ( which is < 1 ∀0 ⩽ K ⩽ 1) as depicted in the above plot. and hence the steady state error with unit step input decreases as it's equal to difference of 1 and steady state value.

1e Given the open loop system, upon increasing the controller gain K in a larger range from 1 to 997 in steps of 166. The following step responses were obtained for the above values of K:

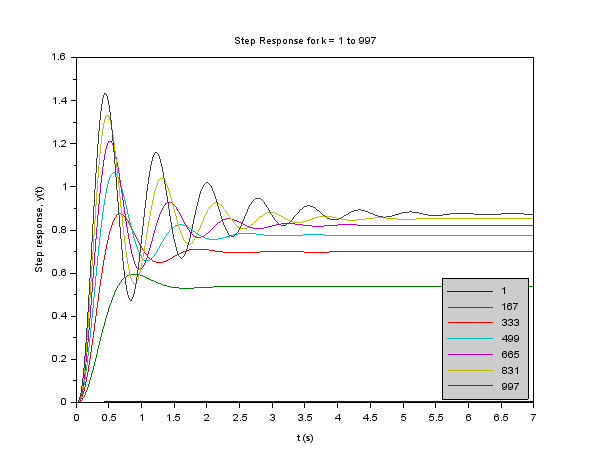


Figure 5

1. Closed loop poles: All of them lie on the open left half plane as the system reponses are stable. Due to the oscillations in responses for ∀K > 2.127, as K = 2.127 is the break-away point as observed in part (c), we conclude an underdamped type response implying 2 complex closed loop poles ∀K > 2.127, in the above plot and the third pole lies on real axis ∀K. And∀K ⩽ 2.127, all the poles lie on negative real axis.
2. Steady state errors: Upon increase in gain value (1 ⩽ K ⩽ 1000), these decrease as is depicted in the above plot by increase of the steady state value towards 1.
3. 5 % settling times: We see that as K becomes larger from 1 to the gain at the breakaway point, and the settling times decrease. However, as K increases till about 200, the settling time continues to decrease. This shows the effects of the third pole. After K = 300, we see that as K increases, the settling time increases, with a minima for settling time observed near K = 600, and then settling time increases as K increases till 1000. Though upon reducing the steps, I observed the variation of 5% settling time vs open loop gain K as plotted
4. Stability of the system: As the steady state is reached for all K in the range taken. Therefore, stability of system is guaranteed for 1 ⩽ K ⩽ 1000 .

**Question 2**

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

Gopen(s) = 1/(s + 4)(s + 3)(s + 12)

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

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

2a

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.

Therefore, PI controller to be used is:

C(s) = 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--');

2b

To obtain undamped natural frequency (𝜔n) 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 𝜔n = 8 rad /s, we obtain open loop gain K = 953.3 as shown below:

Therefore, PI controller to be used is: C(s) = 953.3(s + 0.01)/ s For 𝜔 = 9 rad /s, we obtain open loop gain K = 1329

Therefore, PI controller to be used is: C(s) = 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--');

2c

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.

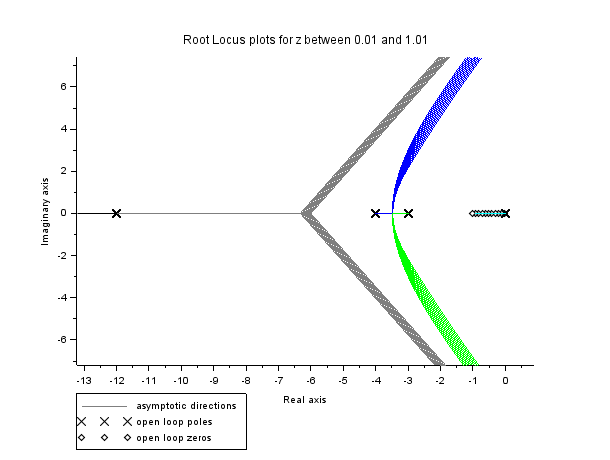
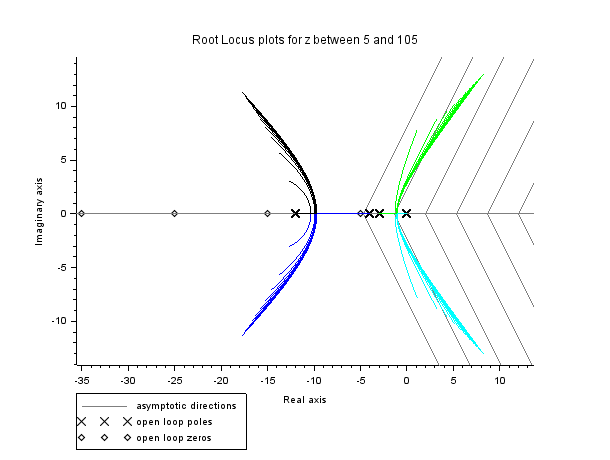


Figure 4: Root Locii of system for z between 0.01 to 1.01

Upon varying z between 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.



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');

2d

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).

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

**Question 3**

Given transfer function:

G(s) = 1 /s^2+ 5s + 6

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

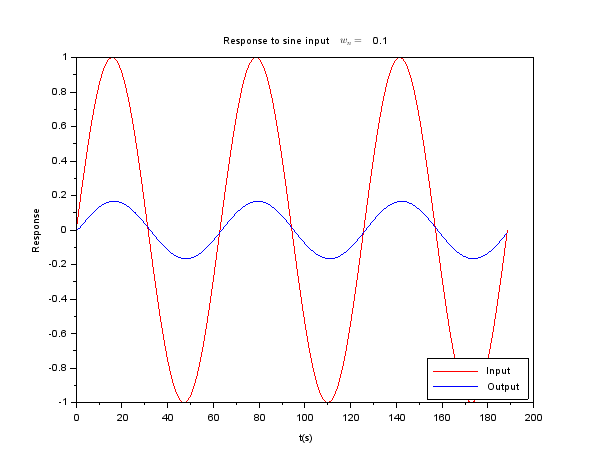


Figure : Response to sin(0.1\*t)

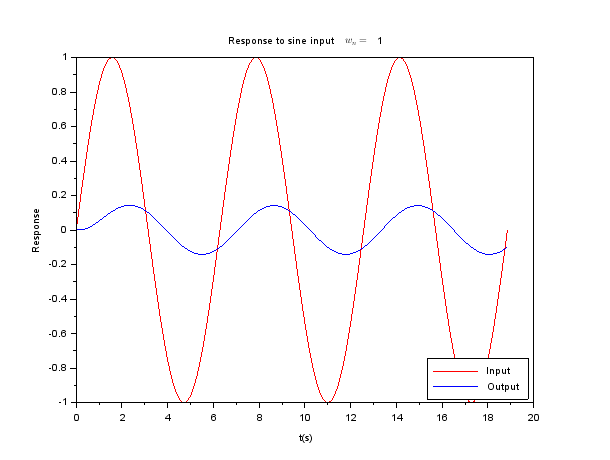


Figure : Response to sin(1\*t)

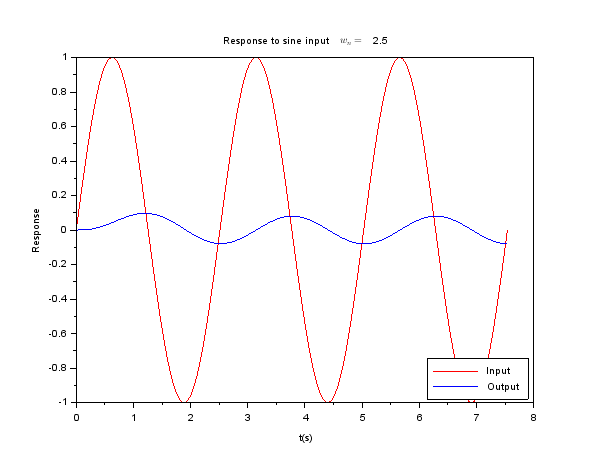


Figure : Response to sin(t)

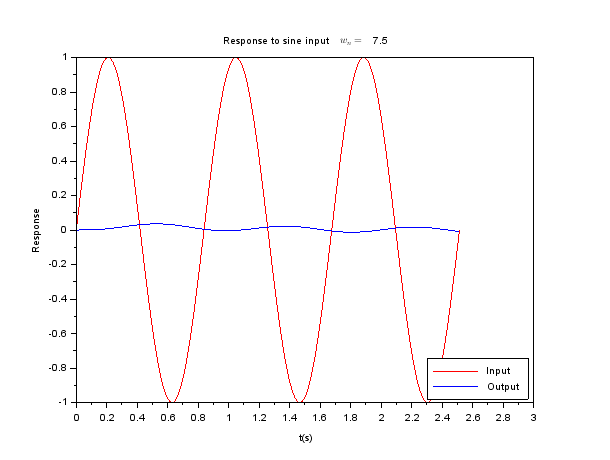


Figure : Response to sin(7.5\*t)

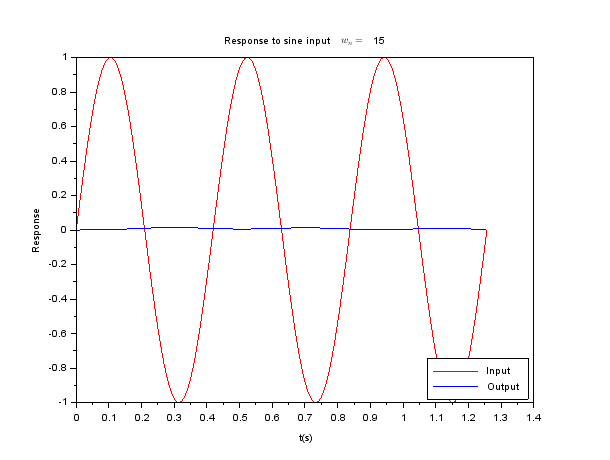
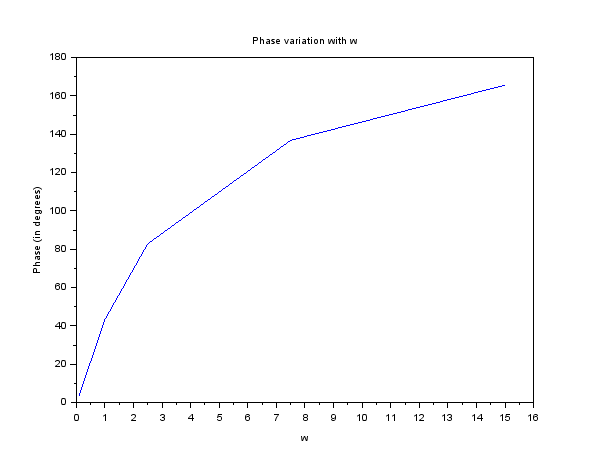
****

Figure : Response to sin(15\*t)

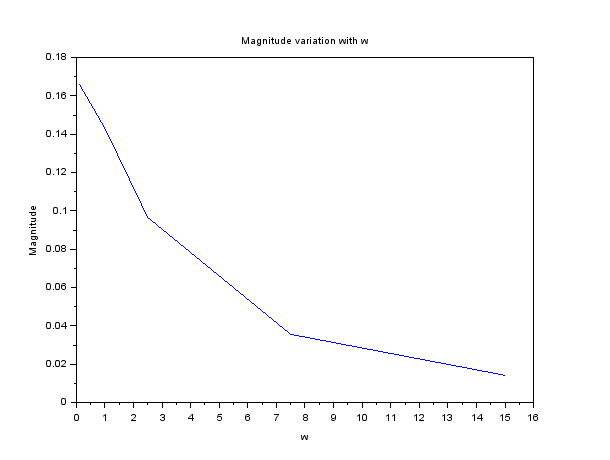
The variation of phase with 𝜔 is shown below:



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

G(j𝜔) = 1 /-𝜔^2 + 5j𝜔 + 6

The variation of magnitude with 𝜔 is shown below

****

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

|G(j𝜔)| = 1/(𝜔^4 + 13𝜔^2 + 36)^1/2

3b The desired relation (between phase difference and angle of G(j⍵) is for frequency measured in rad/s and not Hz.

3c

Given transfer function: G(s) = 1/s^3 + 6s^2 + 11s + 6

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

Here, as well we observe that ratio of magnitudes matches the |G(j𝜔)| and phase difference matches ∠G(j𝜔). The 180° phase difference happens at 𝜔 = 11 = 3.316 rad /s as the phase

difference theoretically is given by tan^-1(11w-w3/6-6w^2). 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):10\*%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');