## Problem 1:

A unity feedback system is characterized by the open loop transfer function

$G(s) = 1 / s\,(0.5 s + 1)(0.2 s + 1)$

A)Determine damping ratio & natural frequency of dominant closed loop poles.
B)Determine the error constants Kp, Kv, Ka .
C)Determine peak overshoot , time to peak, Tr, Ts, Td of the step response of the feedback system.

***Program Coding***

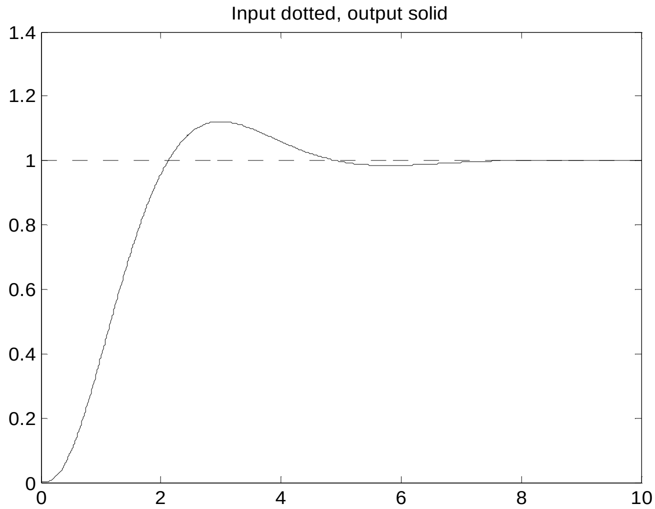
```
nnum=1;
den=conv([conv([1 0],[0.5 1])],[0.2 1]);
G=tf(num,den);
sys=feedback(G,1);
disp('A)The values of damping ratio (Wn) , Natural Frequency (Zeta) & Dominant
closed loop poles ')
disp('are as following :')
[Wn,Z,P]=damp(sys);
Wn=Wn(1)
Zeta=Z(1)
Dominant_Pole=P
disp('B)The error constants Kp, Kv, Ka are as following :')
Kp=dcgain(G)
G1=tf([1 0],1)*G;
Kv=dcgain(G1)
G2=tf([1 0],1)*G1;
Ka=dcgain(G2)
disp('Ess for Ramp Input')
num1=tf(1,[1 0]);
den1=1+G;
sys1=num1/den1;
dcgain(sys1)
disp('Ess for Step Input')
num2=1;
den1=1+G;
sys2=num2/den1;
dcgain(sys2)
disp('C)Peak overshoot, time to peak, Tr, Ts, Td For the step response are as
following :')
t=0:0.01:10;
u=ones(size(t));
[y,t]=step(sys,t);
plot(t,y,'k',t,u,':k')
title('Input dotted, output solid')
[mp,i]=max(y);
Maximum_Overshoot=(mp-1)*100
Tp=t(i)
x1=max(find((y>=0.09)&(y<=0.11)));
x2=max(find((y>=0.89)&(y<=0.91)));
Tr=(x2-x1)*0.0001
j=max(find((y<=0.98)|(y>=1.02)));
Ts=t(j)
d=max(find((y<=0.51)&(y>=0.49)));
Td=t(d)
```

***Output:***



..................................
**Teacher’s Signature**

```
A)The values of damping ratio (Wn) , Natural Frequency (Zeta) & Dominant closed
loop poles are as following :

Wn =    1.3465

Zeta =    0.5512

Dominant_Pole =
  -0.7422 + 1.1235i
  -0.7422 - 1.1235i
  -5.5157

B)The error constants Kp, Kv, Ka are as following :

Kp =   Inf
Kv =     1
Ka =     0
Ess for Ramp Input     1
Ess for Step Input     0

C)Peak overshoot, time to peak, Tr, Ts, Td For the step response are as following
:

Maximum_Overshoot(%) =   12.1186
Tp =     3
Tr =    0.0135
Ts =    4.5200
Td =    1.1700
```




***Discussions:***

1.The systemis a second order type & its natural frequency is found out to be $\omega_n$=910.1923.
2.The ζ (zeta) for the system is equal to 0.5512 which is very close to 0. Therefore the system can be termed as an underdamped system which is reflected by its step respons.The system proceeds towards the steady state value only often a number of oscillations & the peak overshoot ($M_p$) is also large (12.1186).
3. The rise time of the system $t_r$ = 0.0135 sec is very low as is the case with all the underdamped systems.
4. The damping ratio ζ = 0.5512 The low value of the damping ratio gives rise to a large overshoot & oscillatory response.
5. As seen from the step response, the system reaches steady state after a number of oscillations.

## Problem 2:

A unity feedback position control system has open loop transfer function

$$G(s) = 700 / s ( s + 251.2 )$$

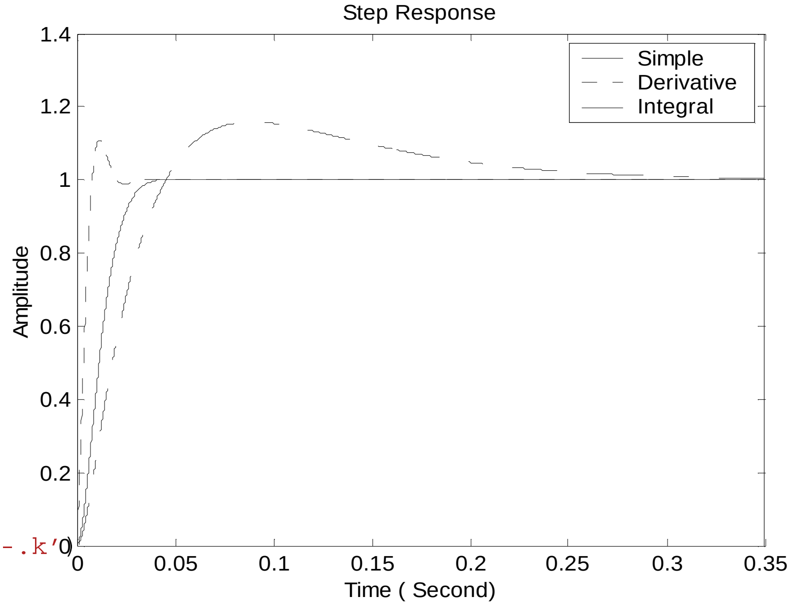
Plot step response of the system with cascade controller T(s) with

a) $T(s) = 28.2$
b) $T(s) = 128.2 + 0.144s$
c) $T(s) = 14.728 + 147 / s$

Compare the response of (a) , (b), (c). Comment upon the effects of derivative & integral control actions.

### *Program Coding*

```
num=700;
den=conv([1 0],[1 251.2]);
g=tf(num,den)
g_a=28.2;
g_b=tf([0.144 128.2],1);
g_c=tf([14.278 147],[1 0]);
g_ol_a=series(g,g_a);
g_cl_a=feedback(g_ol_a,1);
g_ol_b=series(g,g_b);
g_cl_b=feedback(g_ol_b,1);
g_ol_c=series(g,g_c);
g_cl_c=feedback(g_ol_c,1);
t=0:.0001:.35;
[y1 t]=step(g_cl_a,t);
[y2 t]=step(g_cl_b,t);
[y3 t]=step(g_cl_c,t);
plot(t,y1,'k',t,y2,':k',t,y3,'-.k')
```




**Discussions:**

1. The system has a peak amplitude of 1.1 for D control & 1.2 for I control.
2. In the I control the system is nullified by a factor 1/s, which adds a pole at the origin due to the addition of this pole the maximum overshoot of the system increases than P control & also rise time increases.
3. In P(propertional) control $D(s) = K_A$ ( $K_A$=28.2) which signifies the signal output is proportional to the input by a constant factor.
4. In PI control $D(s) = K_C + K_I/s$ where $K_C$ =128.2 & $K_I$ = 0.144. It increases the type of the system by one; which makes it finally a Type 2 system.
5. The system is considered to be unity feedback, closed loop.
6. Since PI controller is essentially a low pass filter, the compensated system now has a larger rise time & settling time . It thus reduces bandwidth of the system . The gain margin, phase margin & resonant peak margin has increased.



..................................
**Teacher's Signature**

## Problem 3:

A unity feedback system has open loop transfer function

$$G(s) = 20 / s\,(1 + 0.8\,s)\,(1 + 0.9\,s)$$

Determine the following for parabolic input

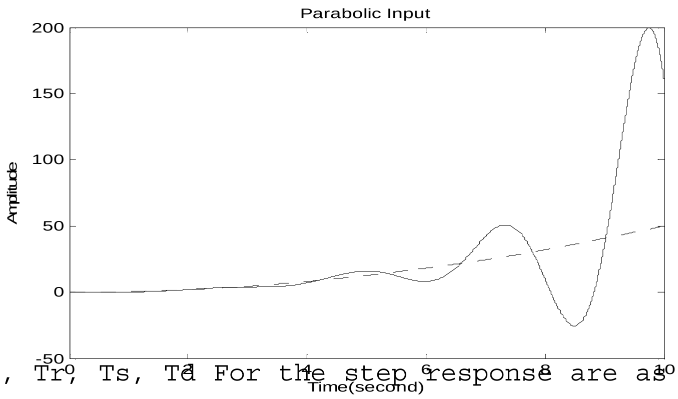
A) Determine damping ratio & natural frequency of dominant closed loop poles.

B) Determine the error constants Kp, Kv, Ka .

C) Determine peak overshoot , time to peak, Tr, Ts, Td of the step response of the feedback system.

***Program Coding***

```
num=20;
den=conv([conv([1 0],[0.8 1])],[0.9 1]);
G=tf(num,den);
sys=feedback(G,1);
disp('A)The values of damping ratio (Wn) , Natural Frequency (Zeta) & Dominant
closed loop poles ')
disp('are as following :')
[Wn,Z,P]=damp(sys);
Wn=Wn(1)
Zeta=Z(1)
Dominant_Pole=P
disp('B)The error constants Kp, Kv, Ka & Ess are as following :')
Kp=dcgain(G)
G1=tf([1 0],1)*G;
Kv=dcgain(G1)
G2=tf([1 0],1)*G1;
Ka=dcgain(G2)
disp('Ess for Parabolic Input')
num1=tf(1,[1 0 0]);
den1=1+G;
sys1=num1/den1;
dcgain(sys1)
disp('Ess for Step Input')
num2=1;
den1=1+G;
sys2=num2/den1;
dcgain(sys2)
disp('C)Peak overshoot, time to peak, Tr, Ts, Td For the step response are as
following :')
t=0:0.001:10;
p=0.5.*t.*t;
[y,t,x]=lsim(sys,p,t);
plot(t,y,'k',t,p,':k')
[mp,i]=max(y);
Maximum_Overshoot=(mp-1)*100
Tp=t(i)
x1=max(find((y>=0.09)&(y<=0.11)));
x2=max(find((y>=0.89)&(y<=0.91)));
Tr=(x2-x1)*0.0001
j=max(find((y<=0.98)|(y>=1.02)));
Ts=t(j)
d=max(find((y<=0.51)&(y>=0.49)));
Td=t(d)
```



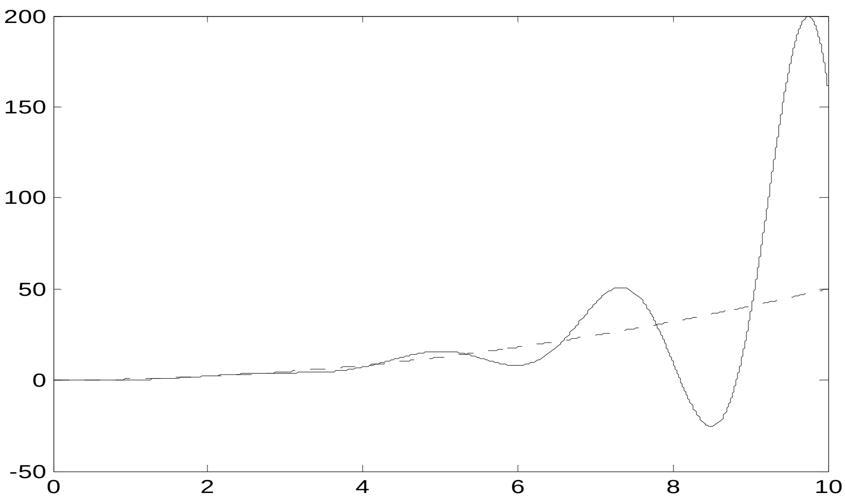

**Output:**



..................................

**Teacher's Signature**

```
A)The values of damping ratio (Wn) , Natural Frequency (Zeta) & Dominant closed
loop poles are as following :

Wn =    2.6816
Zeta =   -0.2800
Dominant_Pole =
   0.7509 + 2.5743i
   0.7509 - 2.5743i
  -3.8630

B)The error constants Kp, Kv, Ka are as following :

Kp =   Inf
Kv =    20
Ka =     0
Ess for Parabolic Input    Inf
Ess for Step Input     0

C)Peak overshoot, time to peak, Tr, Ts, Td For the step response are as following
:

Maximum_Overshoot =  1.9861e+004
Tp =    9.7490
Tr =    0.7155
Ts =    10
Td =    8.0940
```



## Problem 4:

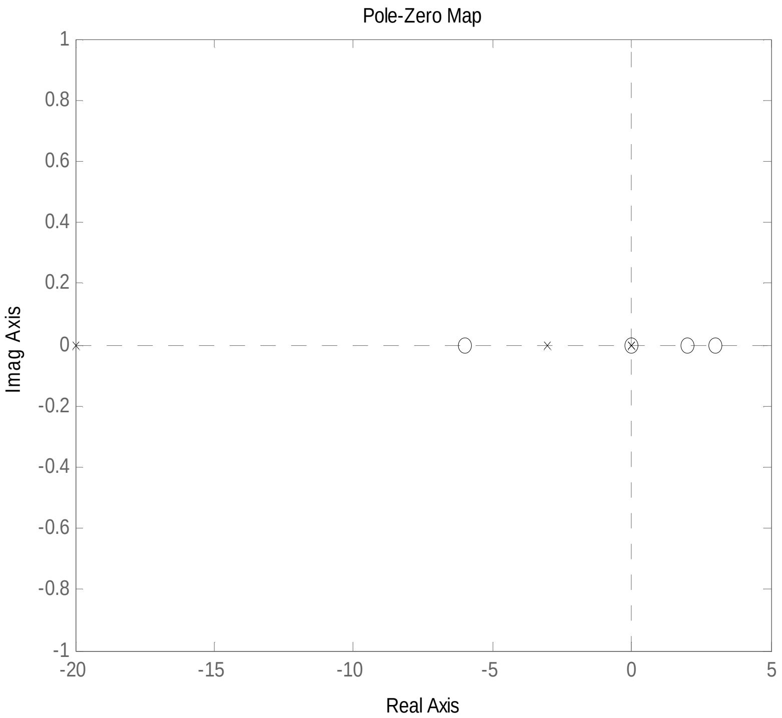
G(s) = s ( s –2 ) ( s + 6) (s – 3 ) / $s^2$ ( s + 20 ) ( s + 3 )

Determine the poles & zeros & plot the outputs.

### *Program Coding*

```
num=poly([0;2;-6;3]);
den=poly([0;0;-20;-3]);
G=tf(num,den);
[z p]=tf2zp(num,den)
pzmap(G,'k')
```

```
Outputs:
z =
         0
   -6.0000
    3.0000
    2.0000

p =
     0
     0
   -20
    -3
```




**Discussions:**

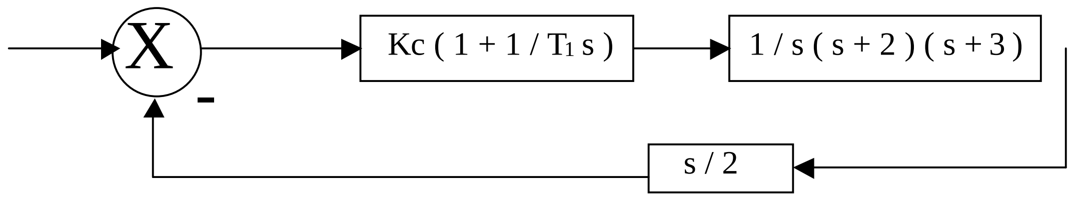
1. Since there are no poles on the right hand of immaginary axis as shown by the pole-zero map the system is stable.
2. There are four poles of the system, all are real.
3.2 poles are at origin.



..................................

**Teacher's Signature**

## Problem 5:

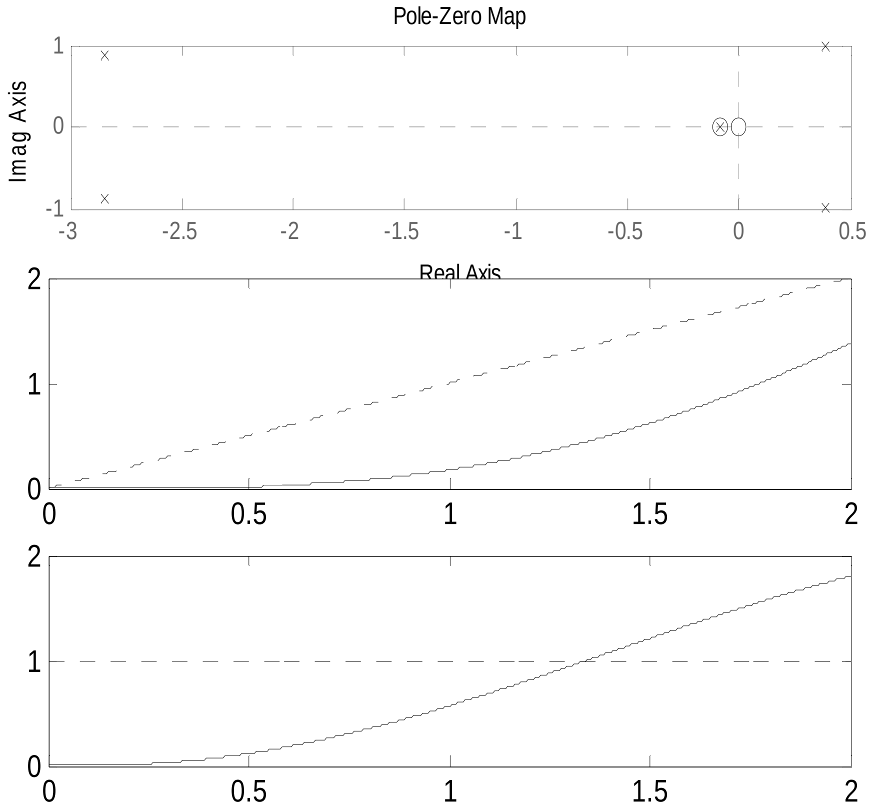



Kc = 10 ; $T_1$ = 12

Determine the system response , closed loop poles and also check the stability of the system for .

***Program Coding***

```
Kc=10;
T1=12;
num1=conv([T1 1],Kc);
den1=[T1 0];
G1=tf(num1,den1);
num2=1;
den2=poly([0;-2;-3]);
G2=tf(num2,den2);
G=series(G1,G2);
H=tf([1],[1 0]);
sys_cl=feedback(G,H);
disp('Close loop poles')
P=pole(sys_cl)
subplot(3,1,1);
pzmap(sys_cl,'k')
[y,t]=step(sys_cl);
t=0:0.001:2;
r=t;
[y,t,x]=lsim(sys_cl,r,t);
subplot(3,1,2);
plot(t,y,'k',t,r,':k')
u=ones(size(t));
[y,t]=step(sys_cl,t);
subplot(3,1,3);
plot(t,y,'k',t,u,':k')
```




**Outputs:**

```
Close loop poles
P =
  -2.8468 + 0.8826i
  -2.8468 - 0.8826i
   0.3883 + 0.9896i
   0.3883 - 0.9896i
  -0.0830
```

**Discussions:**

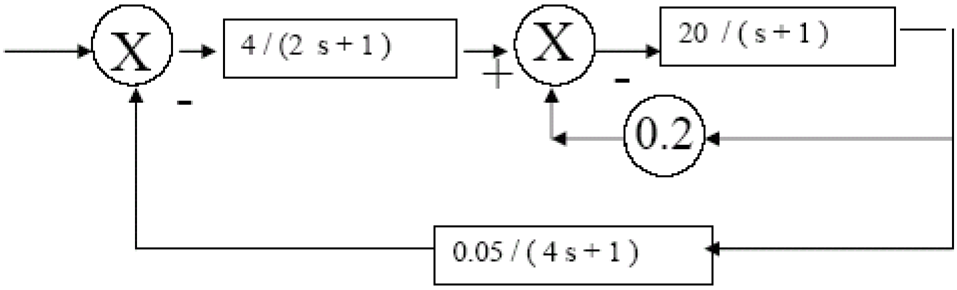
1.Since 2 poles are situated at the right hand sight of the imaginary axis the system is unstable.



..................................

**Teacher's Signature**

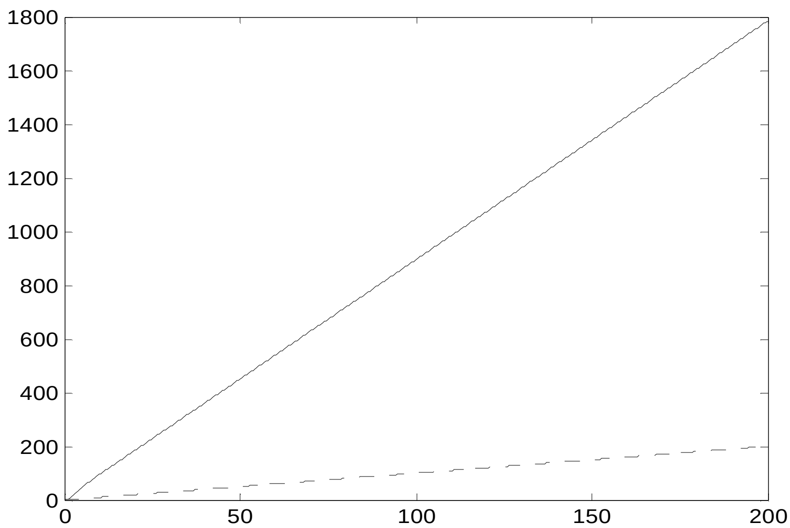
## Problem 6:




Find closed loop poles and system response for ramp input.

***Program Coding***

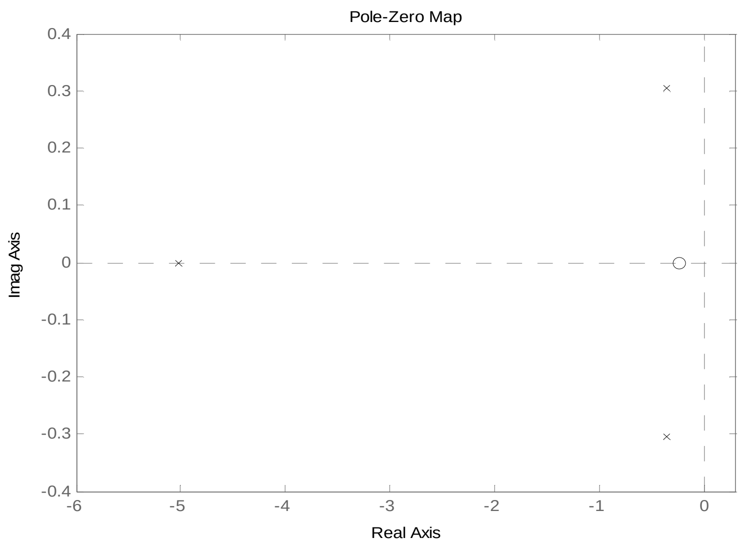
```
Gs1=tf(4,[2 1]);
Gs2=tf(20,[1 1]);
Hs1=0.2;
Gs3=feedback(Gs2,Hs1);
Gs=series(Gs1,Gs3);
Hs=tf(0.05,[4 1]);
sys=feedback(Gs,Hs);
p=esort(pole(sys))
t=[0:0.1:200];
r=t;
[y,t,x]=lsim(sys,r,t);
plot(t,y,'k',t,r,':k')
pzmap(sys,'k')
```



**Outputs:**

p =

-0.3634 + 0.3031i
-0.3634 - 0.3031i
-5.0232



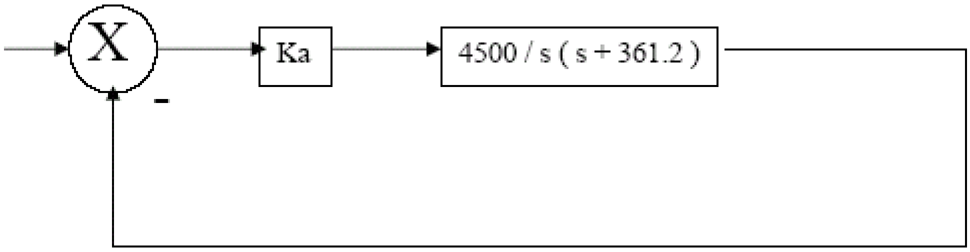

**Discussions:**

1. Since there are no poles on the right hand of immaginary axis as shown by the pole-zero map the system is stable.
2. There are 3 poles of the system, one real & two immaginary.



**...................................**
**Teacher’s Signature**

## Problem 7:

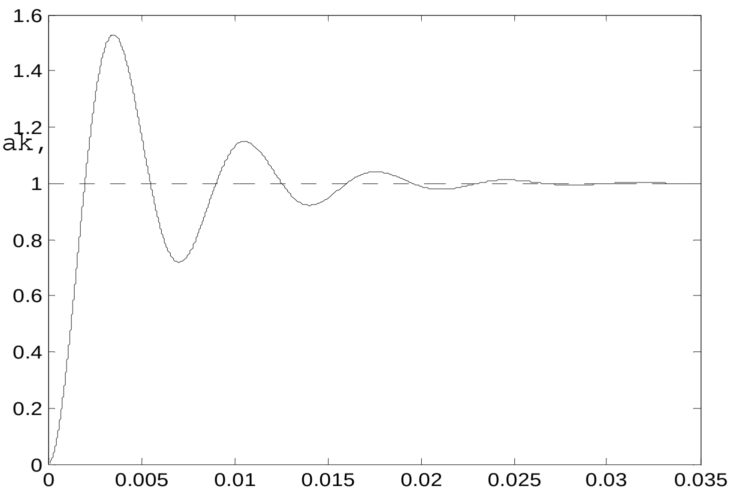



$K_a = 184.1$

Determine Wn, zeta, Mp, poles, Tr, Tp, Td, Ts for step input.

***Program Coding***

```
G1=184.1;
G2=tf(4500,[1 361.2 0]);
G=series(G1,G2);
sys=feedback(G,1);
disp('A)The values of damping ratio (Wn) , Natural Frequency (Zeta) & Dominant
closed loop poles ')
disp('are as following :')
[W,Z,P]=damp(sys);
Wn=W(2)
Zeta=Z(2)
Dominant_Pole=P
disp('C)Peak overshoot, time to peak,
following :')
t=0:0.000001:.035;
u=ones(size(t));
[y,t]=step(sys,t);
plot(t,y,'k',t,u,':k')
[mp,i]=max(y);
Maximum_Overshoot=(mp-1)*100
Tp=t(i)
x1=max(find((y>=0.09)&(y<=0.11)));
x2=max(find((y>=0.89)&(y<=0.91)));
Tr=(x2-x1)*0.0001
j=max(find((y<=0.98)|(y>=1.02)));
Ts=t(j)
d=max(find((y<=0.51)&(y>=0.49)));
Td=t(d)
```




**Outputs:**

The values of damping ratio (Wn) , Natural Frequency (Zeta) & Dominant closed loop poles are as following :

Wn = 910.1923

Zeta = 0.1984



...................................
**Teacher's Signature**

Dominant_Pole =
1.0e+002 *
-1.8060 + 8.9210i
-1.8060 - 8.9210i

Peak overshoot, time to peak, Tr, Ts, Td For the step response are as following :

Maximum_Overshoot = 52.9406
Tp = 0.0035

Tr = 0.8013

Ts = 0.0216

Td = 0.0013



..................................
**Teacher's Signature**

## Problem 8:

$D(s)=k_c + k_i/S$

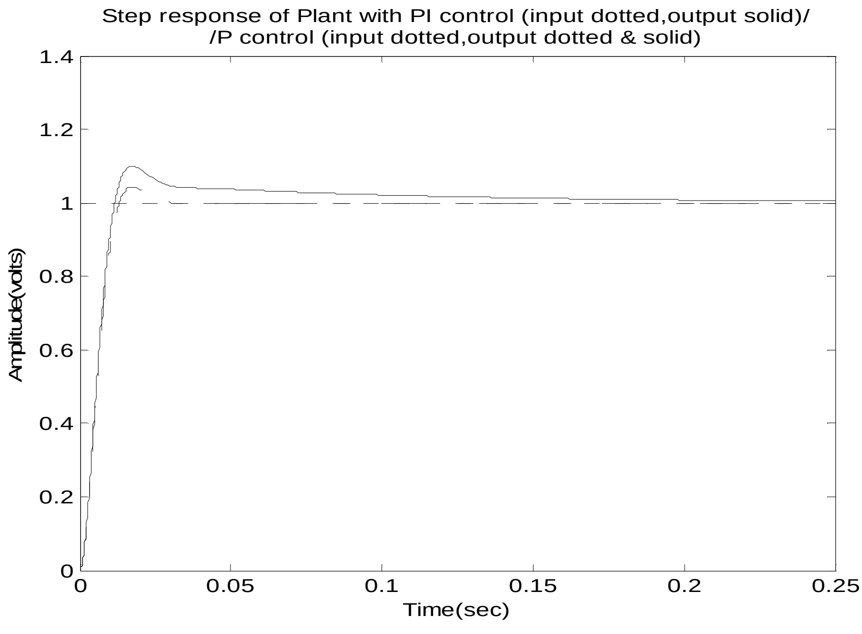
$D(s)G(s)= 4500(k_c s+k_i)/(s^2(s+361.2))$ // with PI control open loop tf

$D(s)G(s)= 4500\,k_A/(s(s+361.2))$ // for P control

For $k_C$ =14.728, $k_I$ =147.28 , $k_A$ =14.5, determine the step response of P control & PI control .Comment on your results.

***Program Coding***

```
Kc=14.728;
Ki=147.28;
Ka=14.5;
den=conv([1 0 0],[1 361.2]);
DsGs_PI_ol=4500*Kc*tf([1 Ki/Kc],den);
DsGs_P_ol=4500*tf(Ka,conv([1 0],[1 361.2]));
plant_PI=feedback(DsGs_PI_ol,1);
plant_P=feedback(DsGs_P_ol,1);
t=[0:0.0001:0.25];
u=ones(size(t));
y_PI=step(plant_PI,t);
y_P=step(plant_P,t);
plot(t,y_PI,'k',t,y_P,'-.k',t,u,':k');
xlabel('Time(sec)');
ylabel('Amplitude(volts)');
title('Step response of Plant with PI control (input dotted,output solid)//P
control (input dotted,output dotted & solid)');
```




**Discussions:**

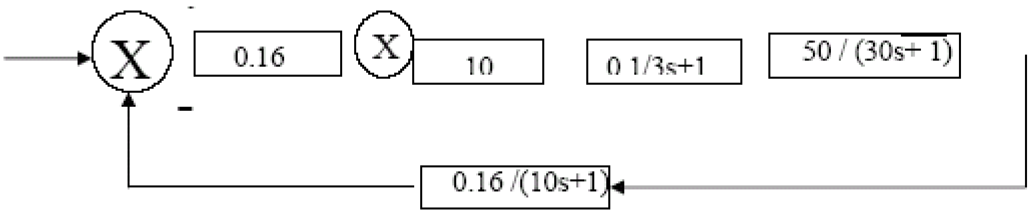
1. The systemhas a peak amplitude of 1.7 for PI control & 1.04 for P control.
2. In the PI control the system is nullified by a factor 1/s, which adds a pole at the origin due to the addition of this pole the maximum overshoot of the system increases than P control & also rise time increases.
3. In P(propertional) control $D(s) = K_A$ ( $K_A$=14.5) which signifies the signal output is proportional to the input by a constant factor.
4. In PI control $D(s) = K_C + K_I/s$ where $K_C$ =14.728 & $K_I$ = 147.28. It increases the type of the system by one; which makes it finally a Type 2 system.
5. The system is considered to be unity feedback, closed loop.
7. As G(s) has pole in the origin, the system is type one. Therefore the constant $k_p$ for step response is infinite. So, the steady state errorfor step input is zero.



..................................

**Teacher's Signature**

## Problem 9:

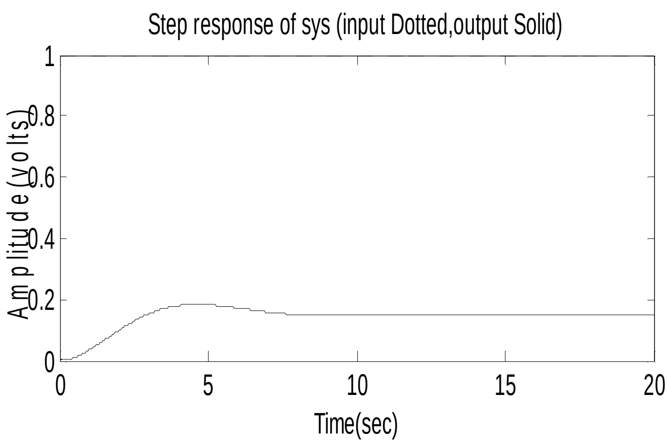


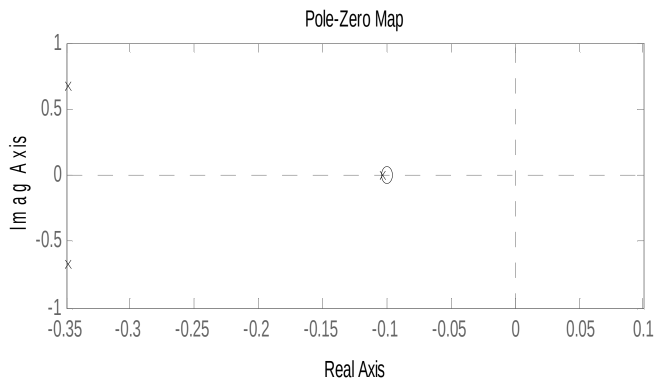

Determine the transfer function for step input & observe the closed loop poles.

### *Program Coding*

```
Gs1=10*tf(0.1,[3 1]);
Gs2=feedback(Gs1,1);
Gs3=tf(50,[30 1]);
Gs4=series(Gs2,Gs3);
Gs=0.16*feedback(Gs4,1);
Hs=tf(0.16,[10 1]);
sys=feedback(Gs,Hs)
t=[0:0.001:20];
y=step(sys,t);
u=ones(size(t));
subplot(2,1,1);
plot(t,y,'k',t,u,':k')
xlabel('Time(sec)');
ylabel('Amplitude(volts)');
title('Step response of sys (input Dotted,output Solid)');
p=esort(pole(sys))
subplot(2,1,2);
pzmap(sys)
```

```
Output:
Transfer function:
            80 s + 8
---------------------------------
900 s^3 + 720 s^2 + 583 s + 53.28


p =

  -0.1028
  -0.3486 + 0.6742i
  -0.3486 - 0.6742i
```






...................................
**Teacher's Signature**

## Problem 10:

G(s)= (ks+1.5)/ (s(s+1)(s+5)(s+15))

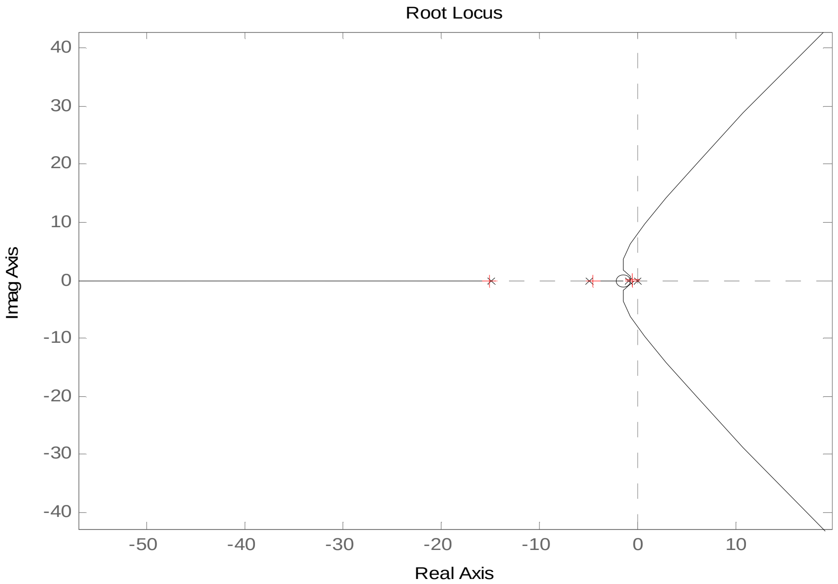
Determine the value of k for which the system will be stable.

### *Program Coding*

```
r1=[-1.5];
r2=[0;-1;-5;-15];
num=poly(r1);
den=poly(r2);
Gs=tf(num,den);
rlocus(Gs);
[K poles]=rlocfind(Gs)
```

**Outputs:**

```
Select a point in the graphics window
selected_point =
  -0.2276 - 0.3990i
K =  21.1962

poles =

 -15.1333
  -4.6190
  -0.6238 + 0.2563i
  -0.6238 - 0.2563i
```




### Discussions:

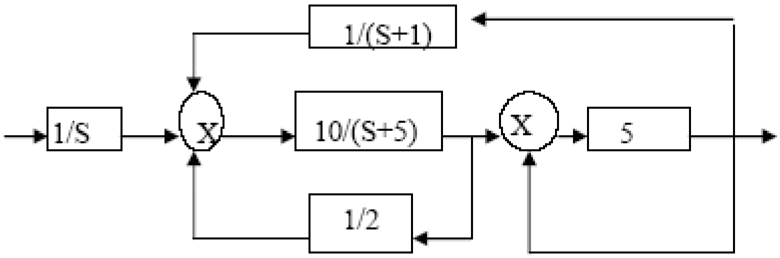
1. As has been obtained by the use of functions rlocus(sys) & [k,poles]=rlocus(sys), the value of the k for which the system is just stable is around 21 & the value of k for which it is unstable is 22.Thus approximately, the system performs sustained oscillations for a value of k around 22.Hence the range of k fro a stable system is 0<k<22.
2. The value of k for marginal stability obtained from the point of insertion of the root loci branches with the imaginary axis.
5. From the answer sheet the value of k for which the system is stable is found 0<k<=`21.1962`.
6. The function rlockfind is used to determine the value of k . This function gives us the option of selecting any point on the root locus plot after the execution of the program.
7. K is the point at which the system is still stable. This value was taken since a better approximation could not be obtained.



..................................

Teacher's Signature

## Problem 11:

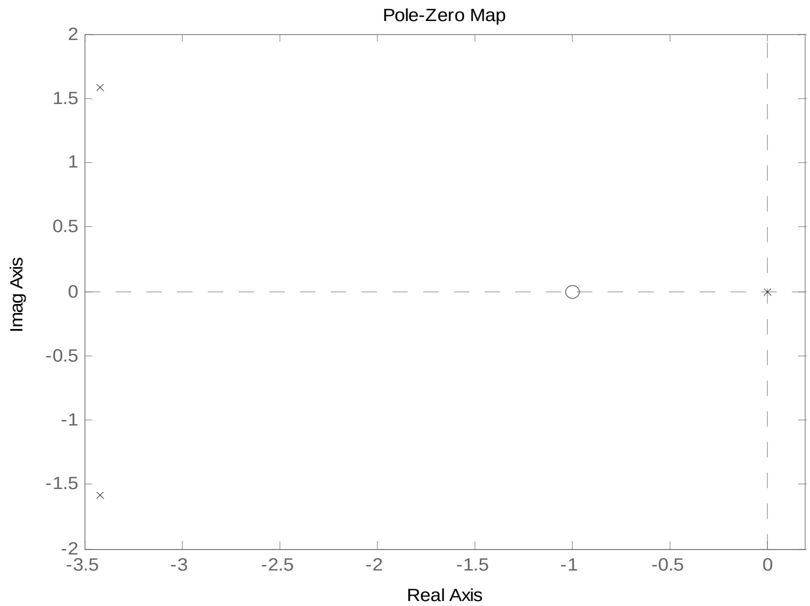



Determine C(s)/R(s) & plot a PZ map evaluating poles & zeros.

### *Program Coding*

```
g1=tf(5,1);
sys_1=feedback(g1,1);
g4=tf(0.5,1);
sys_10=tf(1,5);
sys_2=series(g4,sys_10);
num=10;
den=[1 5];
g2=tf(num,den);
sys_3=series(g2,sys_1);
g3=tf(1,[1 1]);
sys_4=parallel(g3,sys_2);
sys_5=feedback(sys_3,sys_4);
g5=tf(1,[1 0]);
sys=series(g5,sys_5)
poles=pole(sys)
zeros=zero(sys)
pzmap(sys,'k')
```

```
Outputs:

Transfer function:
      250 s + 250
------------------------
30 s^3 + 205 s^2 + 425 s

poles =
        0
  -3.4167 + 1.5789i
  -3.4167 - 1.5789i

zeros =
    -1
```

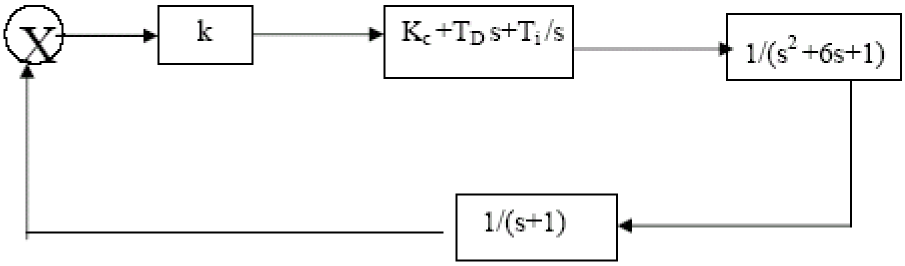



Discussions:
1.Since there are no poles on the right hand of immaginary axis as shown by the pole-zero map the system is stable.
2. There are 3 poles of the system, one is at origin & two immaginary.



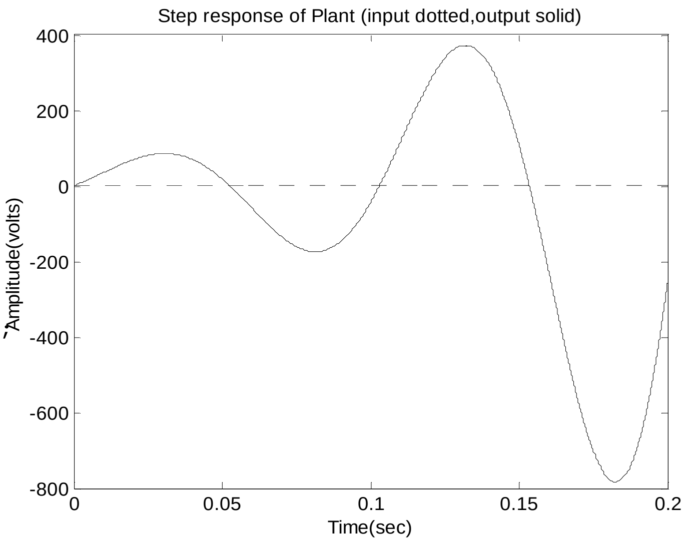
..................................
**Teacher's Signature**

## Problem 12:




For k= 500 , $k_C$ =300 , $T_d$ =6, $T_i$ =5, plot the step response of the PID controller.

### *Program Coding*




```
K=500;
Kc=300;
Td=6;
Ti=5;
Gs1=tf(1,[1 6 1]);
Gs2=tf([Td Kc Ti],[1 0]);
Gs=K*series(Gs1,Gs2);
Hs=tf(1,[1 1]);
disp('The System Transfer function');
sys=feedback(Gs,Hs)
t=[0:0.00001:.2];
y=step(sys,t);
u=ones(size(t));
plot(t,y,'k',t,u,':k');
xlabel('Time(sec)');
ylabel('Amplitude(volts)');
title('Step response of Plant (input dotted,output solid)');
Poles_of_the_system=pole(sys)
```

**Outputs:**

```
The System Transfer function

3000 s^3 + 153000 s^2 + 152500 s + 2500
---------------------------------------
s^4 + 7 s^3 + 3007 s^2 + 150001 s + 2500

Poles_of_the_system =
  14.8256 +62.2362i
  14.8256 -62.2362i
 -36.6346
  -0.0167
```

## Problem 13:

$G(s)=(6.3s^2+18s+12.811)/(s^4+6s^3+11.3s^2+18s+12.811)$

Plot the step response & find rise time, peak time, peak overshoot & settling time.

### *Program Coding*

```
sys=tf([6.3 18 12.811],[1 6 11.3 18 12.811]);
t=[0:0.001:15];
y=step(sys,t);
u=ones(size(t));
plot(t,y,'k',t,u,':k');
xlabel('Time(sec)');
ylabel('Amplitude(volts)');
title('Step response of sys (input Dotted,output Solid)');
[mp,i]=max(y);
disp('% peak overshoot:');
mp=(mp-1)*100
disp('The value of peak time');
Tp=t(i)
j=max(find((y<=0.98)|(y>=1.02)));
disp('Settling time');
Ts=t(j)
k=max(find((y<=0.51)&(y>=0.49)));
disp('Delay time');
Td=t(k)
t10=max(find((y<=0.11)&(y>=0.09)));
t90=max(find((y<=0.91)&(y>=0.89)));
t1=t(t10);
t2=t(t90);
t3=t2-t1;
disp('Rise time');
Tr=t3
```

**Outputs:**

```
% peak overshoot:
mp =   61.9877
The value of peak time
Tp =    1.6690
Settling time
Ts =   10.0400
Delay time
Td =    0.5160
Rise time
Tr =    4.2510
```



..................................

**Teacher's Signature**

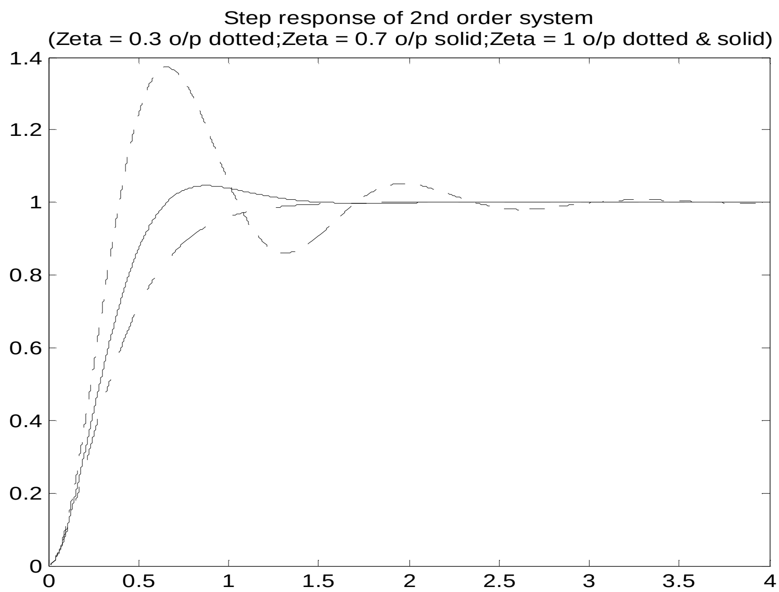
## Problem 13:

Plot the step response for a second order closed loop poles with an underdamped frequency $w_n$ =5 & damping ratios 0.3,0.7 & 1. Explain the variations in the curves.

### *Program Coading*

```
w=5;
num=w*w;
den=[1 2*.3*w w*w];
R1=tf(num,den);
den2=[1 2*.7*w w*w];
R2=tf(num,den2);
den3=[1 2*1*w w*w];
R3=tf(num,den3);
t=0:0.001:4;
u=ones(size(t));
[x,t]=step(R1,t);
[y,t]=step(R2,t);
[z,t]=step(R3,t);
plot(t,x,':k',t,y,'k',t,z,'-.k')
title('Step response of 2nd order system (Zeta = 0.3 o/p dotted;Zeta = 0.7 o/p solid;Zeta = 1 o/p dotted & solid)');
```

**Outputs:**




Discussions:

1.As damping ratio increases the peak overshoot decreases.

2.As the damping ratio decreases, system become more oscillatory.

3.When damping ratio is 1, the system is critically damped. If damping ratio is less than 1,then the system become oscillatory rather underdamped.

4.The system will be overdamped, if the value of damping ratio is more than 1.



..................................

**Teacher's Signature**

## Problem 15:

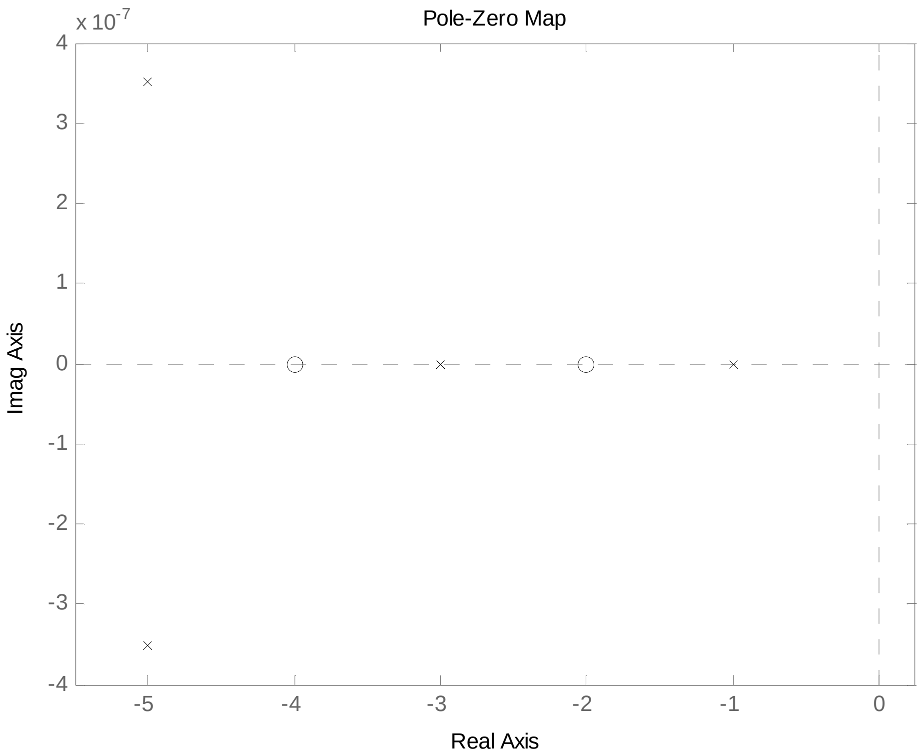
Obtain poles & zeros. G(s)= (10(s+2)(s+4)/ (s+1)(s+3)(s+5)$^2$)

### *Program Coding*

```
num=10*poly([-2 -4]);
den=poly([-1 -3 -5 -5]);
g=tf(num,den);
poles=pole(g)
zeros=zero(g)
pzmap(g,'k')
```

Output:

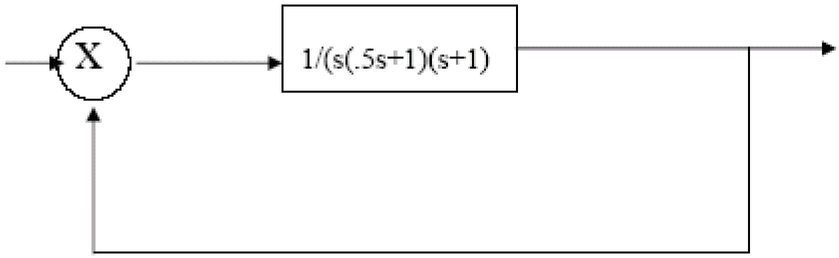
```
poles =

  -5.0000 + 0.0000i
  -5.0000 - 0.0000i
  -3.0000
  -1.0000


zeros =

    -4
    -2
```






..................................
**Teacher's Signature**

## Problem 16:



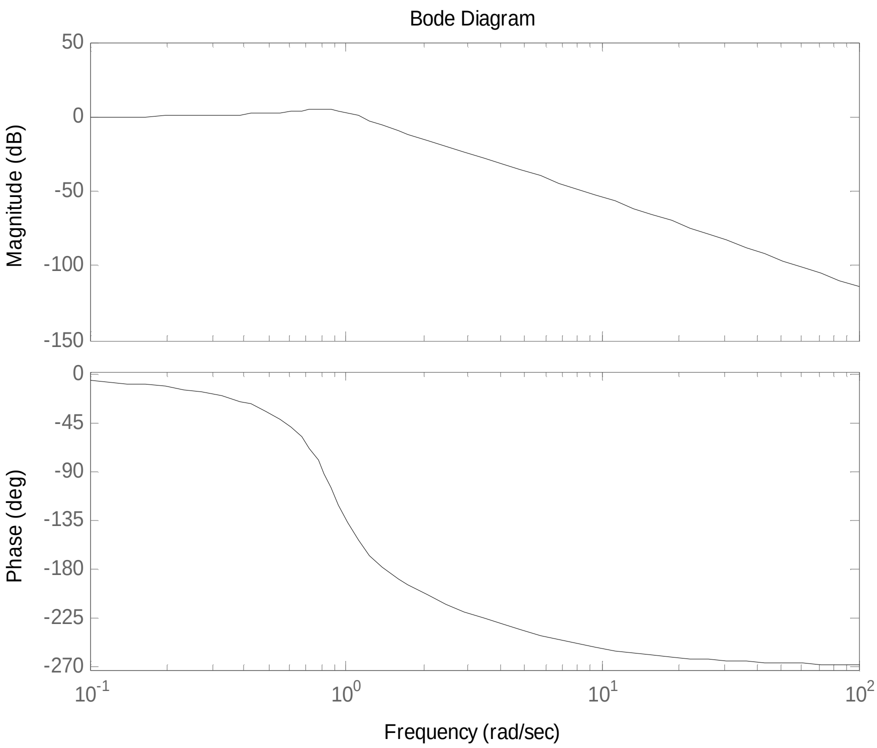

Obtain the Bode plot of the system.

### *Program Coding*

```
num=1;
den=conv([1 0],[conv([0.5 1],[1 1])]);
Gs=tf(num,den);
sys=feedback(Gs,1);
[GM,PM,GCF,PCF]=margin(sys)
bode(sys,'k')
```

Output:

```
GM =

    2.0018


PM =

   24.1643


GCF =

    1.4146


PCF =

    1.1291
```






……………………………
**Teacher's Signature**

## Problem 17:

Consider the open loop tf G(s)= $1/(s^2+0.8s+1)$ & draw the Nyquist plot.

### *Program Coding*

```
num=1;
den=[1 0.8 1];
g=tf(num,den);
nyquist(g,'k')
```

### *Output*