EE 593

EXPERIMENT NO : CS I/6

TILLE : **DETERMINATION OF ROOT LOCUS PLOT USING MATLAB CONTROL SYSTEM TOOLBOX FOR 2^{ND} ORDER SYSTEM & OBTAIN CONTROLLER SPECIFICATION PARAMETERS.**

OBJECTIVE : To determine Root Locus plot of a 2^{nd} order system

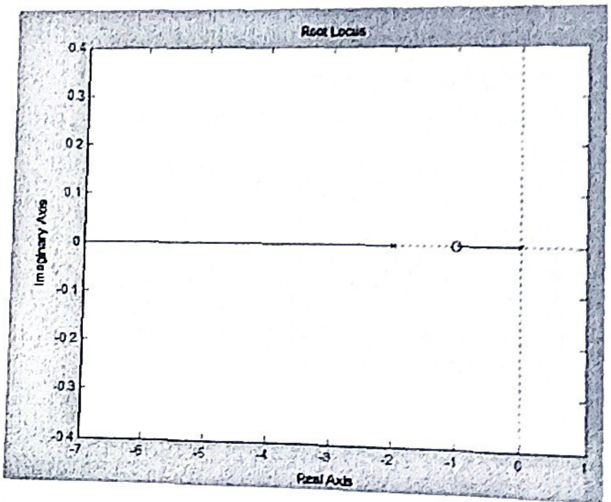
THEORY: rlocus computes the Evans root locus of a SISO open-loop model. The root locus gives the closed-loop pole trajectories as a function of the feedback gain k (assuming negative feedback). Root loci are used to study the effects of varying feedback gains on closed-loop pole locations. In turn, these locations provide indirect information on the time and frequency responses. rlocus(sys) calculates and plots the rootlocus of the open-loop SISO model sys. This function can be applied to any of the following feedback loops by setting sys appropriately. If sys has transfer function $G(s) = \frac{N(S)}{D(S)}$, The closed-loop poles are the roots of d(s) + k*n(s)=0

Example 1: Obtain Root Locus Plot of a system having forward path transfer function of $G(s) = \frac{1+s}{s(1+0.5s)}$

Matlab Code:

```
num = [1 1]
den = conv([1 0],[.5 1])
g = tf(num,den);
rlocus(g)
```

Output :



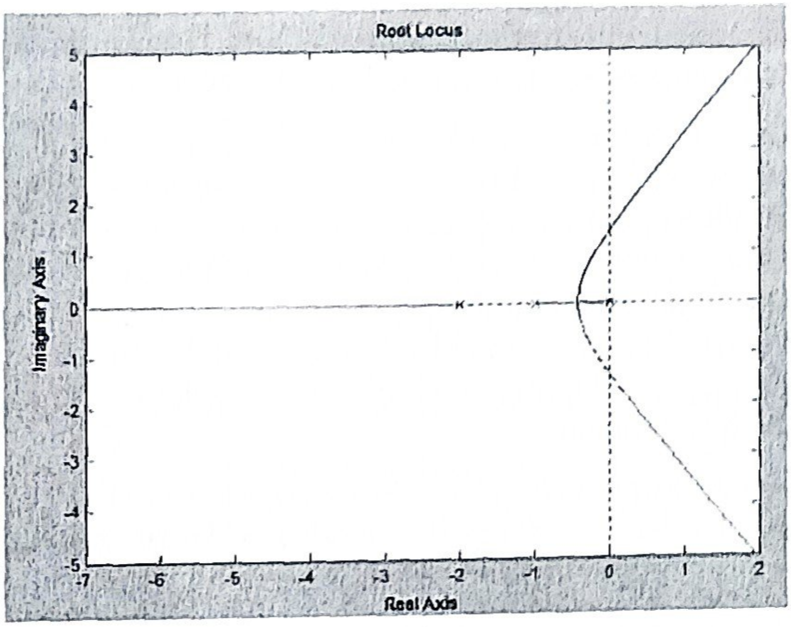
Example 2: Obtain Root Locus Plot of a system having forward path transfer function of $G(s) = \frac{1}{s(1+s)(s+2)}$

Matlab Code :

```
num = [1]
den = poly([0 -1 -2])
g = tf(num,den);
```

```
rlocus(g)
```

Output :



Assignment:

Obtain Root Locus Plot of the following transfer function. Also verify your result theoretically.

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

DISCUSSION:

1. **What is relative stability? How can you measure relative stability using Root locus?**
2. **What do you mean by break away point?**
3. **What is asymptotic line?**

EXPERIMENT NO : CS I/7

TILLE : DETERMINATION OF NYQUIST PLOT USING MATLAB CONTROL SYSTEM TOOLBOX.

OBJECTIVE : To determine nyquist plot of a 2^{nd} order system

THEORY: A stability test for time invariant linear systems can also be derived in the frequency domain. It is known as Nyquist stability criterion. It is based on the complex analysis result known as Cauchy's principle of argument. Note that the system transfer function is a complex function. By applying Cauchy's principle of argument to the open-loop system transfer function, we will get information about stability of the closed-loop system transfer function and arrive at the Nyquist stability criterion (Nyquist, 1932). The importance of Nyquist stability lies in the fact that it can also be used to determine the relative degree of system stability by producing the so-called phase and gain stability margins. These stability margins are needed for frequency domain controller design techniques.

We present only the essence of the Nyquist stability criterion and define the phase and gain stability margins. The Nyquist method is used for studying the stability of linear systems with pure time delay. For a SISO feedback system the closed-loop transfer function is given by:

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

Since the system poles are determined as those values at which its transfer function becomes infinity, it follows that the closed-loop system poles are obtained by solving the following equation.

In the following we consider the complex function

$$D(s) = 1 + G(s)H(s)$$

whose zeros are the closed-loop poles of the transfer function. In addition, it is easy to see that the poles of D(s) are the zeros of T(s) . At the same time the poles of are the open-loop control system poles since they are contributed by the poles of , which can be considered as the open-loop control system transfer function— obtained when the feedback loop is open at some point. The Nyquist stability test is obtained by applying the Cauchy principle of argument to the complex function . First, we state Cauchy's principle of argument.

Let F(s) be an analytic function in a closed region of the complex plane except at a finite number of points. It is also assumed that F(s) is analytic at every point on the contour. Then, as travels around the contour in the - plane in the clockwise direction, the function F(s) encircles the origin in the plane in the same direction N times, given by N = Z-P

Where Z and P stand for the number of zeros and poles (including their multiplicities) of the function inside the F(s) contour.

The above result can be also written as

$$\operatorname{Arg}(F(s)) = 2\pi N$$

Nyquist Criterion:

It states that the number of unstable closed-loop poles is equal to the number of unstable open-loop poles plus the number of encirclements of the origin of the Nyquist plot of the complex function D(s). This can be easily justified by applying Cauchy's principle of argument to the function D(s) with the -plane contour. Note that and represent the numbers of zeros and poles, respectively, of in the unstable part of the complex plane. At the same time, the zeros of D(s) are the closed-loop system poles, and the poles of D(s) are the open-loop system poles (closed-loop zeros).

The above criterion can be slightly simplified if instead of plotting the function $D(s) = 1 + G(s)H(s)$, we plot only the function $G(s)H(s)$ and count encirclement of the Nyquist plot of around the point (-1+j0), so that the modified Nyquist criterion has the following form. The number of unstable closed-loop poles (Z) is equal to the number of unstable open-loop poles (P) plus the number of encirclements (N) of the point (=1+j0),

$$Z=N+P$$

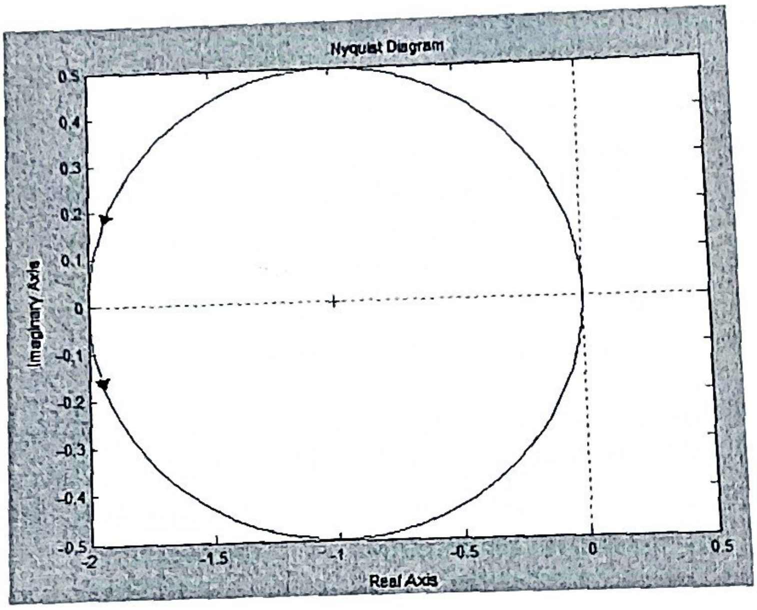
Example 1: Obtain Nyquist Plot of a system having forward path transfer function of

$$G(s) = \frac{2+s}{(s+1)(s-1)}$$

Matlab Code:

```
num = [1 2]
den = conv([1 1], [1 -1])
g = tf(num,den);
nyquist(g)
```

Output :



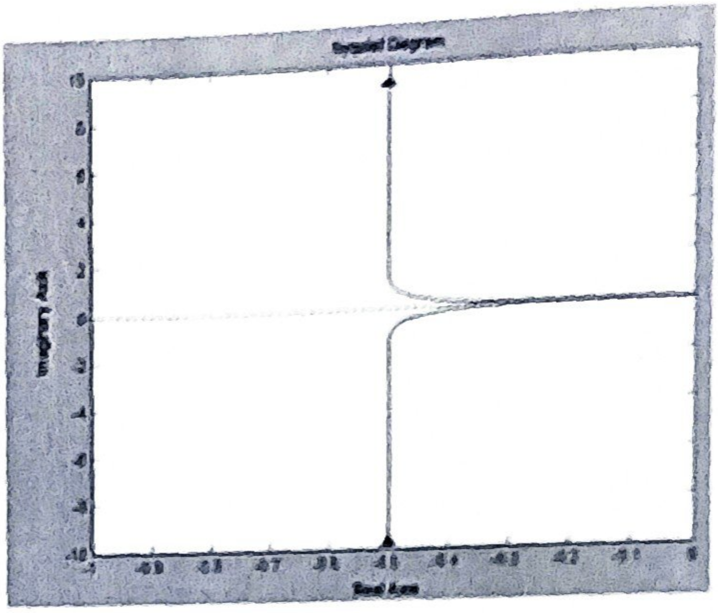
Example 2: Obtain Nyquist Plot of a ufb system having forward path transfer function of

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

Matlab Code :

```
num = [1]
a=[1 0]
b= [1 2 2]
den =conv(a,b)
g = tf(num,den)
nyquist(g)
```

Output :



Assignment:

Obtain Nyquist Plot of the following transfer function. Also verify your result theoretically.

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

DISCUSSION:

1. **What is nyquist contour?**
2. **What is polar plot? How it differs from Nyquist plot?**

EXPERIMENT NO : CS I/8

TILLE : STUDY THE EFFECT OF PI & PD CONTROLLER ON SYSTEM PERFORMANCE

OBJECTIVE : To determine

I. Effect of PI controller on system performance
II. Effect of PD controller on system performance

THEORY :

PID controllers use a 3 basic behavior types or modes: P - proportional, I - integrative and D - derivative. While proportional and integrative modes are also used as single control modes, a derivative mode is rarely used on it' s own in control systems. Combinations such as PI and PD control are very often in practical systems.

P Controller: In general it can be said that P controller cannot stabilize higher order processes. For the 1st order processes, meaning the processes with one energy storage, a large increase in gain can be tolerated. Proportional controller can stabilize only 1st order unstable process. Changing controller gain K can change closed loop dynamics. A large controller gain will result in control system with: a) smaller steady state error, i.e. better reference following b) faster dynamics, i.e. broader signal frequency band of the closed loop system and larger sensitivity with respect to measuring noise c) smaller amplitude and phase margin When P controller is used, large gain is needed to improve steady state error. Stable systems do not have problems when large gain is used. Such systems are systems with one energy storage (1st order capacitive systems). If constant steady state error can be accepted with such processes, than P controller can be used. Small steady state errors can be accepted if sensor will give measured value with error or if importance of measured value is not too great anyway.

PD Controller: D mode is used when prediction of the error can improve control or when it necessary to stabilize the system. From the frequency characteristic of D element it can be seen that it has phase lead of 90°.

Often derivative is not taken from the error signal but from the system output variable. This is done to avoid effects of the sudden change of the reference input that will cause sudden change in the value of error signal. Sudden change in error signal will cause sudden change in control output. To avoid that it is suitable to design D mode to be proportional to the change of the output variable. PD controller is often used in control of moving objects such are flying and underwater vehicles, ships, rockets etc. One of the reason is in stabilizing effect of PD controller on sudden changes in heading variable y(t). Often a "rate gyro" for velocity measurement is used as sensor of heading change of moving object.

PI Controller: PI controller will eliminate forced oscillations and steady state error resulting in operation of on-off controller and P controller respectively. However, introducing integral mode has a negative effect on speed of the response and overall stability of the system. Thus, PI controller will not increase the speed of response. It can be expected since PI controller does not have means to predict what will happen with the error in near future. This problem can be solved by introducing derivative mode which has ability to predict what will happen with the error in near future and thus to decrease a reaction time of the controller. PI controllers are very often used in industry, especially when speed of the response is not an issue. A control without D mode is used when: a) fast response of the system is not required b) large disturbances and noise are present during operation of the process c) there is only one energy storage in process (capacitive or inductive) d) there are large transport delays in the system.

PID Controller: PID controller has all the necessary dynamics: fast reaction on change of the controller input (D mode), increase in control signal to lead error towards zero (I mode) and suitable action inside control error area to eliminate oscillations (P mode). Derivative mode improves stability of the system and enables increase in gain K and decrease in integral time constant Ti, which increases speed of the controller response. PID controller is used when dealing with higher order capacitive processes (processes with more than one energy storage) when their dynamic is not similar to the dynamics of an integrator (like in many thermal processes). PID controller is often used in industry, but also in the control of mobile objects (course and trajectory following included) when stability and precise reference following are required. Conventional autopilot is for the most part PID type controllers.

Effects of Coefficients:

1. **Example 1:** Consider a unity feedback system with forward path transfer function $G(s) = \frac{1}{s^2+10s+20}$. Show the effect of addition of a PID controller on the system performance.

Matlab Code:

```
num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
step(t1,'g')
hold on
num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
step(t2,'m')
hold on
Kp=500;
Kd=10;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t3=feedback(g3,1)
step(t3,'b')
hold on
Kp=500;
Kd=5;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t4=feedback(g3,1)
step(t4,'y')
hold on
Kp=500;
Kd=.01;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
```

PID Controller: PID controller has all the necessary dynamics: fast reaction on change of the controll input (D mode), increase in control signal to lead error towards zero (I mode) and suitable action insi control error area to eliminate oscillations (P mode). Derivative mode improves stability of the syste and enables increase in gain K and decrease in integral time constant Ti, which increases speed of controller response. PID controller is used when dealing with higher order capacitive proces (processes with more than one energy storage) when their dynamic is not similar to the dynamics of integrator (like in many thermal processes). PID controller is often used in industry, but also in control of mobile objects (course and trajectory following included) when stability and precise refere following are required. Conventional autopilot is for the most part PID type controllers.

Effects of Coefficients:

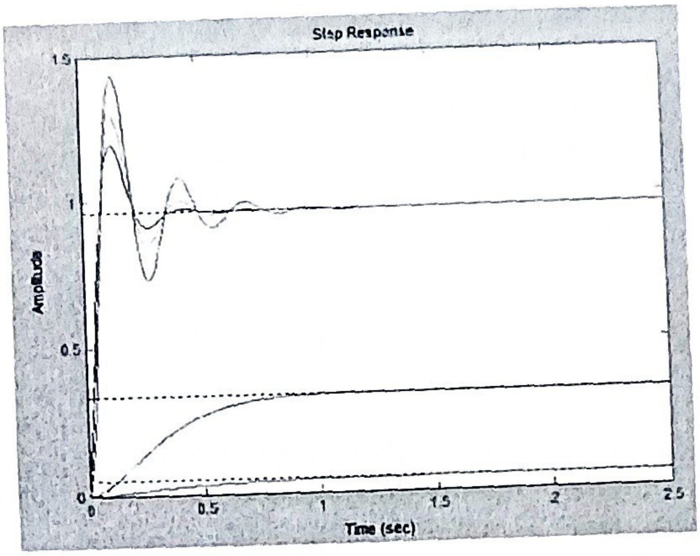
1. **Example 1:** Consider a unity feedback system with forward path transfer function $G(s) = \frac{1}{s^2+10}$
   Show the effect of addition of a PD controller on the system performance.

Matlab Code:

```
num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
step(t1,'g')
hold on
num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
step(t2,'m')
hold on
Kp=500;
Kd=10;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t3=feedback(g3,1)
step(t3,'b')
hold on
Kp=500;
Kd=5;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t4=feedback(g3,1)
step(t4,'y')
hold on
Kp=500;
Kd=.01;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
```

```
t5=feedback(g3,1)
step(t5,'r')
hold on
```

Output:



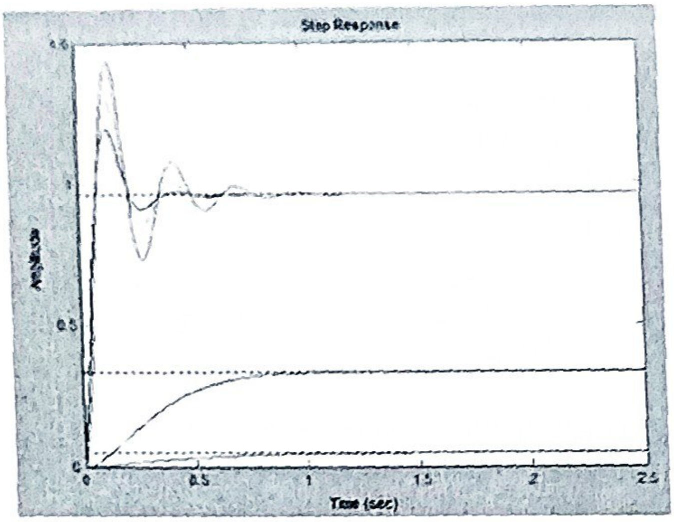
2. Consider a unity feedback system with forward path transfer function $G(s) = \frac{1}{s^2+10s+20}$. Show the ef of addition of a PI controller on the system performance.

Matlab Code:

```
num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
step(t1,'g')
hold on
 num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
step(t2,'m')
hold on
Kp=500;
Ki = 1
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t3=feedback(g3,1)
step(t3,'b')
hold on
Kp=500;
Ki = 100
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
```

```
t5=feedback(g3,1)
step(t5,'r')
hold on
```

Output:



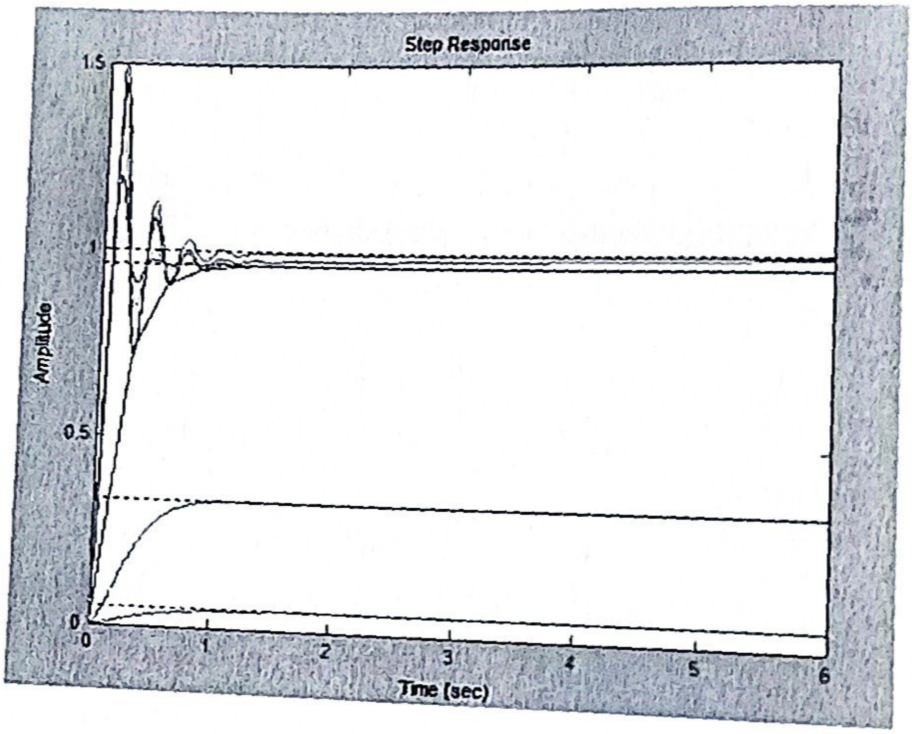
2. Consider a unity feedback system with forward path transfer function $G(s) = \frac{1}{s^2+10s+20}$. Show the effect of addition of a PI controller on the system performance.

Matlab Code:

```
num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
step(t1,'g')
hold on
 num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
step(t2,'m')
hold on
Kp=500;
Ki = 1
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t3=feedback(g3,1)
step(t3,'b')
hold on
Kp=500;
Ki = 100
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
```

```
deno=conv(den,denc)
g3=tf(numo,deno)
t4=feedback(g3,1)
step(t4,'r')
hold on
Kp=500;
Ki = 500
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t5=feedback(g3,1)
step(t5,'g')
hold on
```

Output:



Assignments:

1. Consider a unity feedback system with forward path transfer function $G(s) = \frac{1}{s(s+1.6)}$. Show the effect of addition of a PD controller on the system performance.
2. Consider a unity feedback system with forward path transfer function $G(s) = \frac{1}{s(s+1.6)}$. Show the effect of addition of a PI controller on the system performance.
3. Consider a unity feedback system with forward path transfer function $G(s) = \frac{1}{s(s+1.6)}$. Show the effect of addition of a PID controller on the system performance.

EXPERIMENT NO : CS I/9

TILLE : **STUDY THE EFFECT OF ADDITION OF ZEROS TO THE FORWARD PATH TRANSFER FUNCTION OF A CLOSED LOOP SYSTEM**

OBJECTIVE : To study

I. Effect of addition of zeros to forward path of a open loop system.
II. Effect of addition of zeros to forward path of a closed loop system.

THEORY: The forward path transfer function of general second order system is given by,

$$FPTF \quad G(s) = \frac{W_n^2}{s(s + 2\zeta W_n)}$$

Addition of zero to forward path transfer function:

When we add a zero the forward path transfer function becomes,

$$G'(s) = \frac{W_n^2(T_z s + 1)}{s(s + 2\zeta W_n)}$$

$$G'(s) = \frac{W_n^2(s + z)}{s(s + 2\zeta W_n)z}$$

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

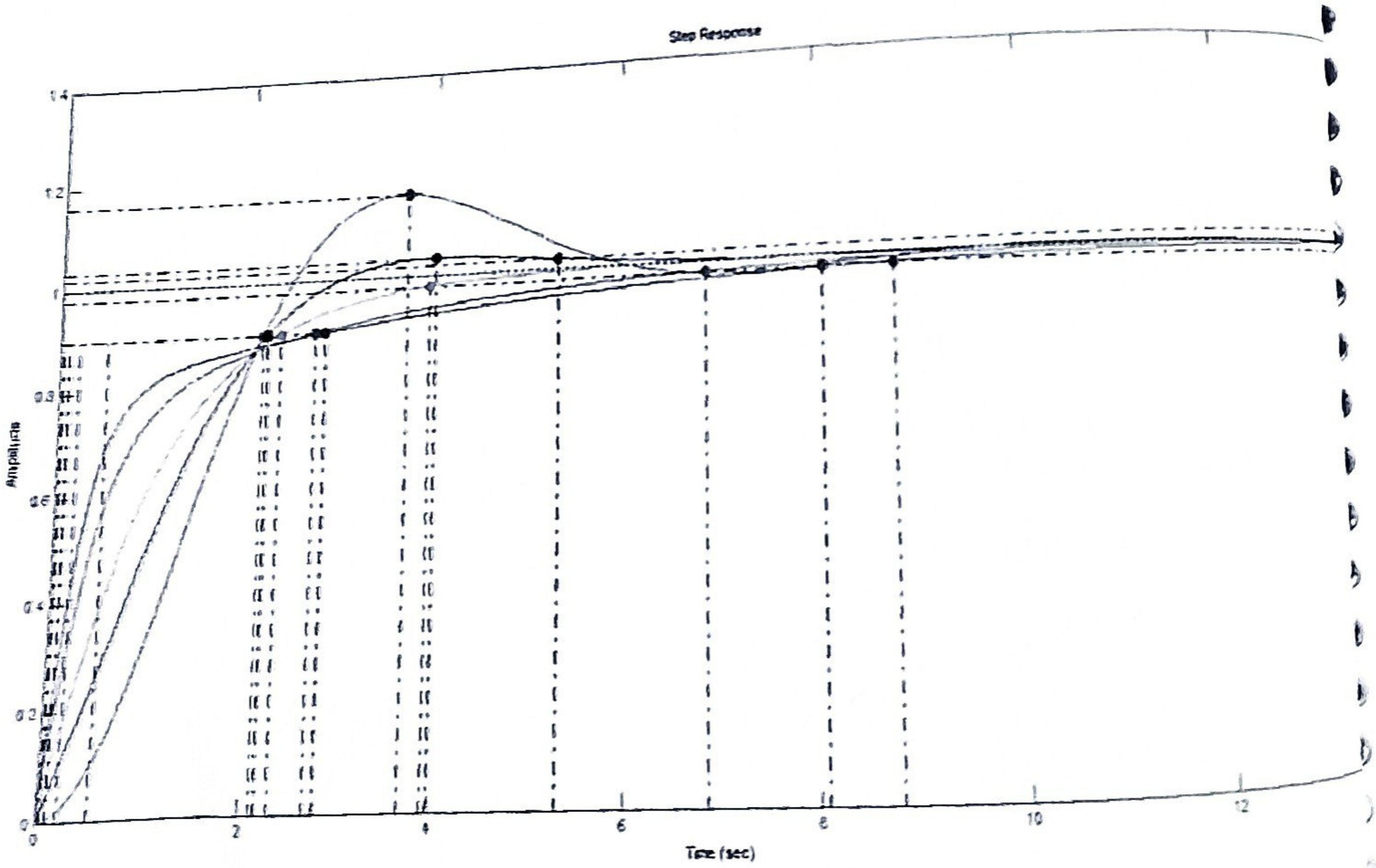
$$\frac{C(s)}{R(s)} = \frac{W_n^2(s + z)}{zs^2 + (W_n^2 + 2\zeta W_n z)s + W_n^2 z}$$

Example 1. Consider a open loop system having forward path transfer function of G(S)=1/S*(S+1). Write a MATLAB code to show the effect of addition of zeros at -2, -1, -0.5.

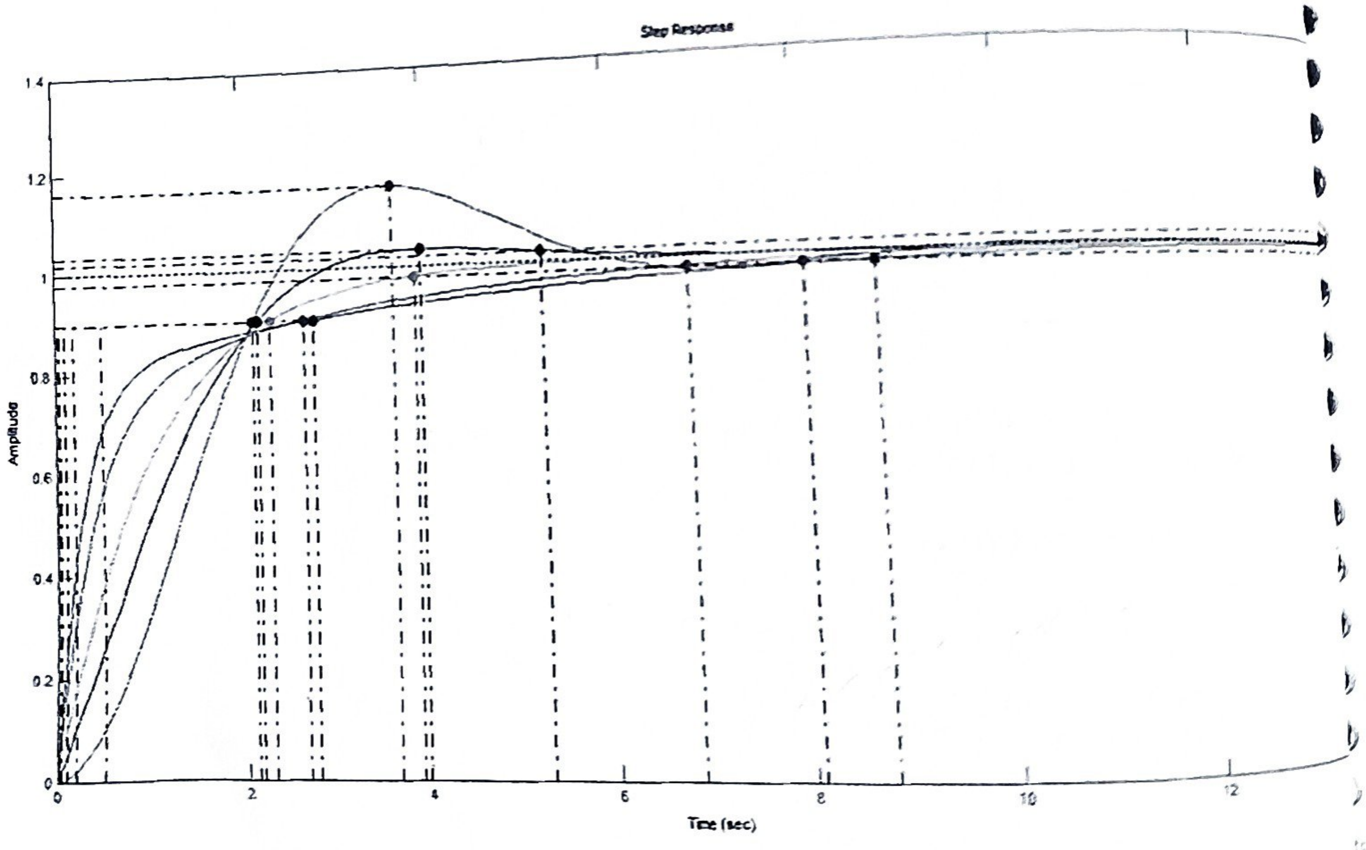
MATLAB CODE:

```
n1=1
d1=[1 1 0]
g1=tf(n1,d1)
t1=feedback(g1,1)
step(t1,'r')
hold on
Tz=0.5
Z1=[Tz 1]
```

```
n2=conv(n1,Z1)
g2=tf(n2,d1)
t2=feedback(g2,1)
step(t2,'b')
hold on
Tz=1
Z2=[Tz 1]
n3=conv(n1,Z2)
g3=tf(n3,d1)
t3=feedback(g3,1)
step(t3,'y')
hold on
Tz=2
Z3=[Tz 1]
n4=conv(n1,Z3)
g4=tf(n4,d1)
t4=feedback(g4,1)
step(t4,'g')
hold on
Tz=3
Z4=[Tz 1]
n5=conv(n1,Z4)
g5=tf(n5,d1)
t5=feedback(g5,1)
step(t5,'m')
```



```
n2=conv(n1,Z1)
g2=tf(n2,d1)
t2=feedback(g2,1)
step(t2,'b')
hold on
Tz=1
Z2=[Tz 1]
n3=conv(n1,Z2)
g3=tf(n3,d1)
t3=feedback(g3,1)
step(t3,'y')
hold on
Tz=2
Z3=[Tz 1]
n4=conv(n1,Z3)
g4=tf(n4,d1)
t4=feedback(g4,1)
step(t4,'g')
hold on
Tz=3
Z4=[Tz 1]
n5=conv(n1,Z4)
g5=tf(n5,d1)
t5=feedback(g5,1)
step(t5,'m')
```



Observation Table:

Tz	Tr	Tp	Ts	%Mp
0.5				
1				
2				
3				

Example 2. Consider a closed loop system having overall transfer function of $T(S)=1/(S^2+S+1)$. Write a MATLAB code to show the effect of addition of zeros at -2, -1, -0.5.

MATLAB CODE:

```
n1=[1]
d1=[1 1 1]
t1=tf(n1,d1)
step(t1,'r')
hold on
Tz=0.5
Z1=[Tz 1]
n2=conv(n1,Z1)
t2=tf(n2,d1)
step(t2,'b')
hold on
Tz=1
Z2=[Tz 1]
n3=conv(n1,Z2)
t3=tf(n3,d1)
step(t3,'y')
hold on
Tz=2
Z3=[Tz 1]
n4=conv(n1,Z3)
t4=tf(n4,d1)
step(t4,'g')
hold on
Tz=3
Z4=[Tz 1]
n5=conv(n1,Z4)
t5=tf(n5,d1)
step(t5,'m')
```

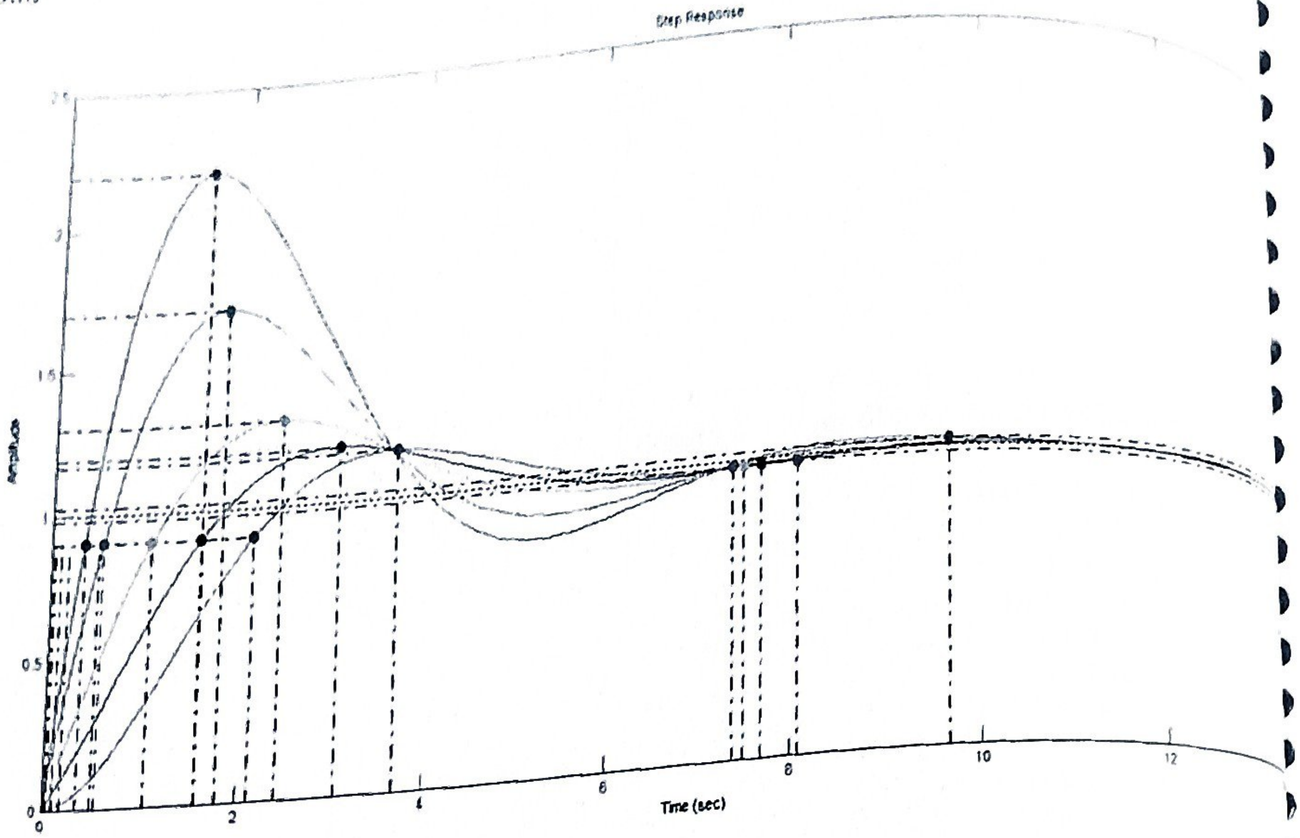
Observation Table:

Tz	Tr	Tp	Ts	%Mp
0.5				
1				
2				
3				

Example 2. Consider a closed loop system having overall transfer function of $T(S)=1/(S^2+S+1)$. Write a MATLAB code to show the effect of addition of zeros at -2, -1, -0.5.

MATLAB CODE:

```
n1=[1]
d1=[1 1 1]
t1=tf(n1,d1)
step(t1,'r')
hold on
Tz=0.5
Z1=[Tz 1]
n2=conv(n1,Z1)
t2=tf(n2,d1)
step(t2,'b')
hold on
Tz=1
Z2=[Tz 1]
n3=conv(n1,Z2)
t3=tf(n3,d1)
step(t3,'y')
hold on
Tz=2
Z3=[Tz 1]
n4=conv(n1,Z3)
t4=tf(n4,d1)
step(t4,'g')
hold on
Tz=3
Z4=[Tz 1]
n5=conv(n1,Z4)
t5=tf(n5,d1)
step(t5,'m')
```



Observation Table:

Tz	tr	Tp	Ts	%Mp
0.5				
1				
2				
3				

DISCUSSION:

1. Discuss the effect of adding zeros to overall transfer function on system response curve.

EXPERIMENT NO : CS I/10

TILLE : **STUDY THE EFFECT OF ADDITION OF POLES TO THE FORWARD PATH TRANSFER FUNCTION OF A CLOSED LOOP SYSTEM**

OBJECTIVE : To study

I. Effect of addition of poles to forward path of a open loop system.
II. Effect of addition of poles to forward path of a closed loop system.

The forward path transfer function of general second order system is given by,

$$FPTF \quad G(s) = \frac{W_n^2}{s(s + 2\zeta W_n)}$$

Addition of pole to forward path transfer function:

When we add a pole, the transfer function becomes,

$$G'(s) = \frac{W_n^2}{s(s + 2\zeta W_n)(T_p s + 1)}$$

$$G'(s) = \frac{W_n^2 P}{s(s + 2\zeta W_n)(s + P)}$$

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

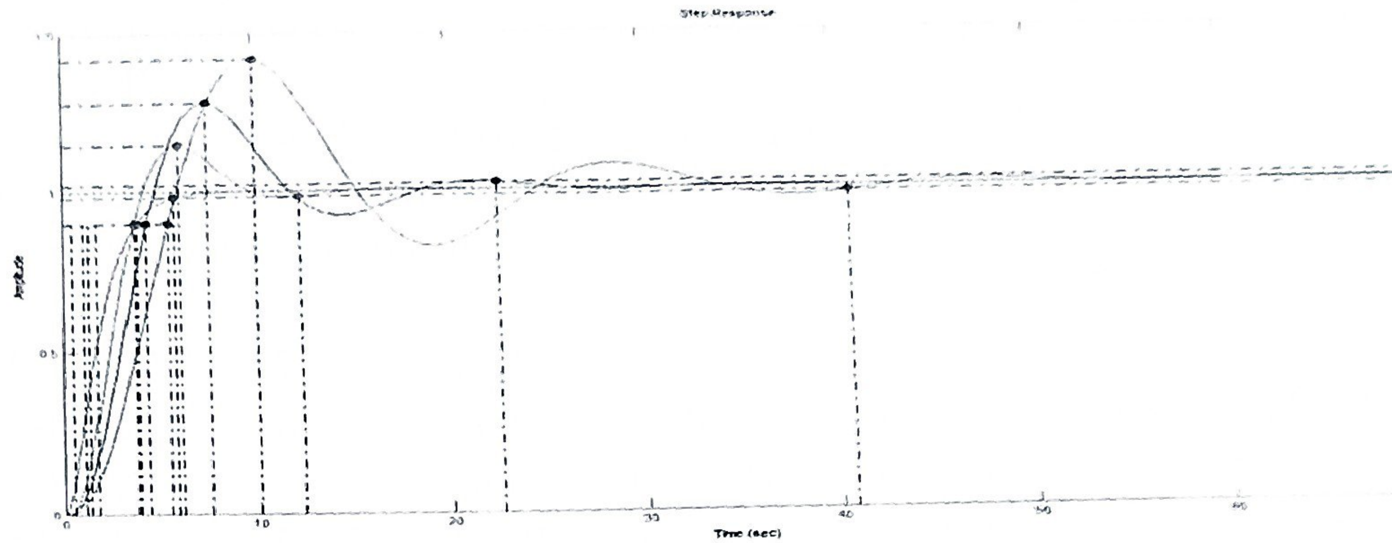
$$\frac{C(s)}{R(s)} = \frac{W_n^2 P}{s^3 + s^2(P + 2\zeta W_n) + 2\zeta W_n P s + W_n^2 P}$$

Example 1. Consider a open loop system having forward path transfer function of $G(S)=1/S*(S+2)$. Write a MATLAB code to show the effect of addition of poles at -1, -0.5, -0.25.

$$G_{new}(S)=[1/S*(S+2)]*[1/(TpS+1)]$$

MATLAB CODE:

```
n1=1
d1=[1 2 0]
g1=tf(n1,d1)
t1=feedback(g1,1)
step(t1,'r')
hold on
Tp=1
P1=[Tp 1]
d2=conv(d1,P1)
g2=tf(n1,d2)
t2=feedback(g2,1)
step(t2,'g')
hold on
Tp=2
P2=[Tp 1]
d3=conv(d1,P2)
g3=tf(n1,d3)
t3=feedback(g3,1)
step(t3,'b')
hold on
Tp=4
P3=[Tp 1]
d4=conv(d1,P3)
g4=tf(n1,d4)
t4=feedback(g4,1)
step(t4,'m')
```



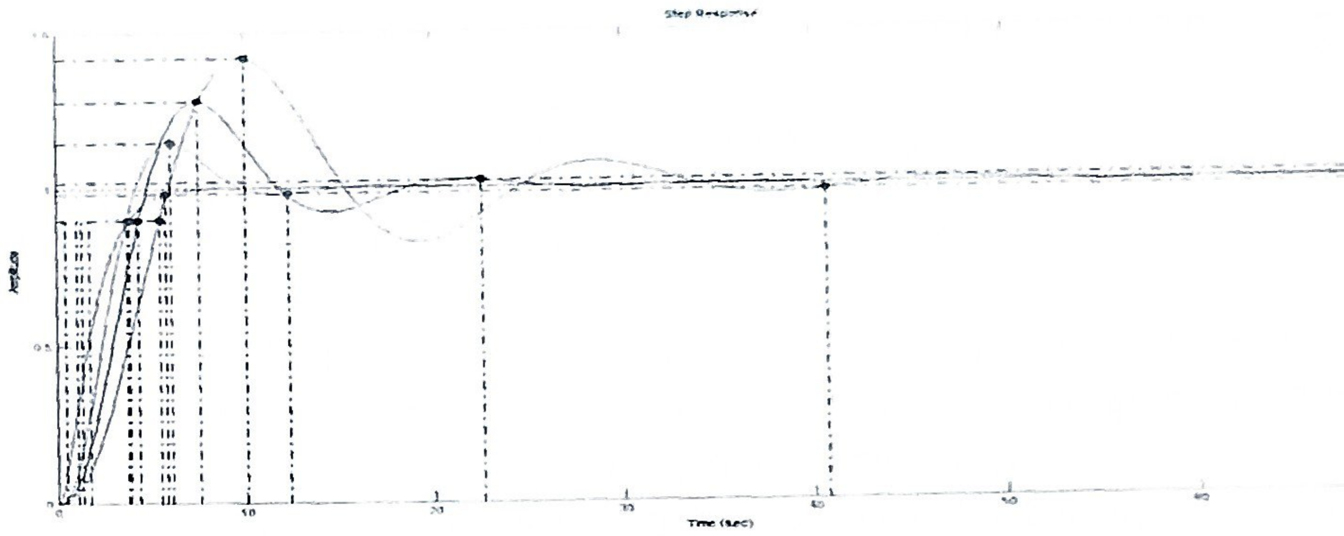
Observation Table:

Tp	tr	Tp	Ts	%Mp
1				
2				
4				

Example 2. Consider a closed loop system having overall transfer function of $T(S)=1/(S^2+0.6S+1)$. Write a MATLAB code to show the effect of addition of zeros at -2, -1, -0.5.

MATLAB CODE:

```
n1=1
d1=[1 0.6 1]
t1=tf(n1,d1)
step(t1,'r')
hold on
Tp=1
P1=[Tp 1]
d2=conv(d1,P1)
t2=tf(n1,d2)
step(t2,'g')
hold on
Tp=2
P2=[Tp 1]
d3=conv(d1,P2)
t3=tf(n1,d3)
step(t3,'b')
hold on
Tp=4
```



Observation Table:

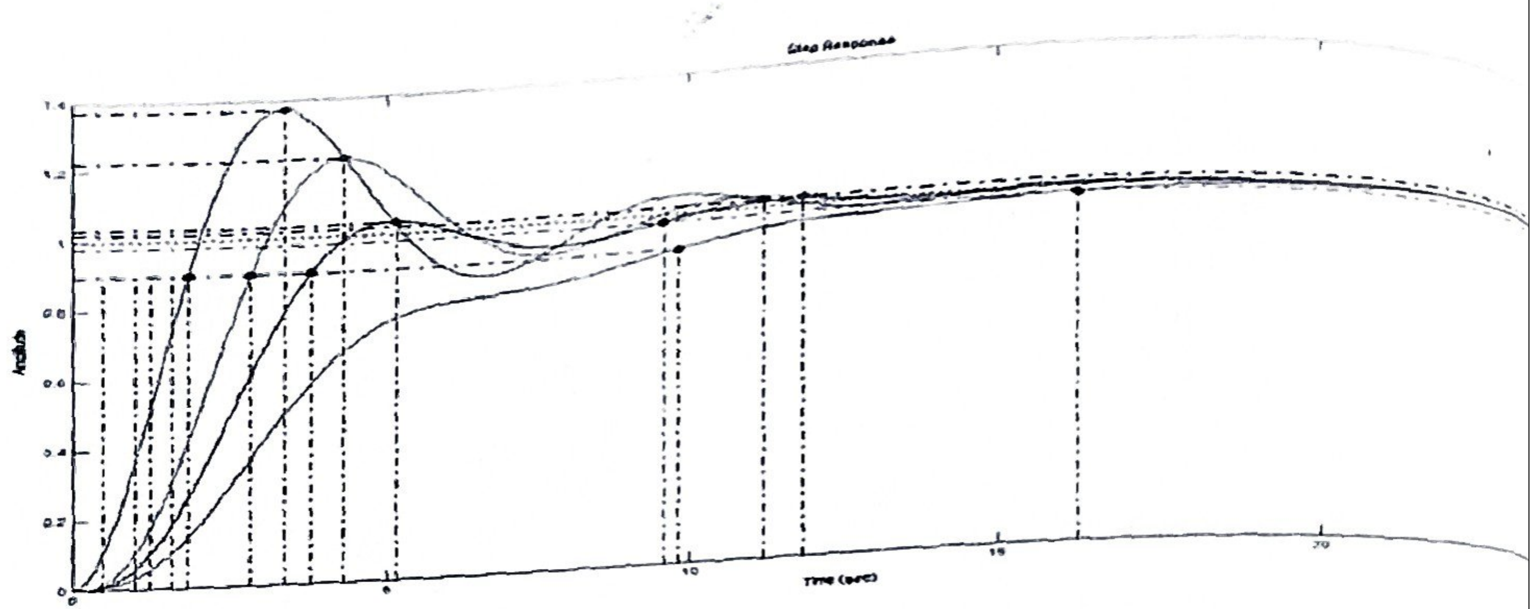
Tp	tr	Tp	Ts	%Mp
1				
2				
4				

Example 2. Consider a closed loop system having overall transfer function of $T(S)=1/(S^2+0.6S+1)$. Write a MATLAB code to show the effect of addition of zeros at -2, -1, -0.5.

MATLAB CODE:

```
n1=1
d1=[1 0.6 1]
t1=tf(n1,d1)
step(t1,'r')
hold on
Tp=1
P1=[Tp 1]
d2=conv(d1,P1)
t2=tf(n1,d2)
step(t2,'g')
hold on
Tp=2
P2=[Tp 1]
d3=conv(d1,P2)
t3=tf(n1,d3)
step(t3,'b')
hold on
Tp=4
```

```
P3=[Tp 1]
d4=conv(d1,P3)
t4=tf(n1,d4)
step(t4,'m')
```



Observation Table:

Tp	tr	Tp	ts	%Mp
1				
2				
4				

DISCUSSION:

1. Discuss the effect of adding poles to overall transfer function on system response curve.