EXPERIMENT 1

TITLE: Determine the transfer function & find out step response of the given systems.
Objective: To obtain the transfer function & step response for the given systems.

Theory: A step signal is a signal whose value changes from one level to another level in zero time. Mathematically, the step signal is represented as given below.

r(t) = u(t) u(t)=1 ; t >= 0

=0 ; t < 0

In the Laplace transform form, R(s)= 1/s

The step response of the given transfer function is obtained as follows:

$G(s) = C(s)/R(s)$

Procedure:

Type the program in MATLAB editor preferably M-file.

Save and run the program.

Give the required inputs in the command window of MATLAB in matrix format.

Step' function calculates the unit step response of a linear system.

Note down the response of the transfer function obtained in MATLAB

QUESTION:

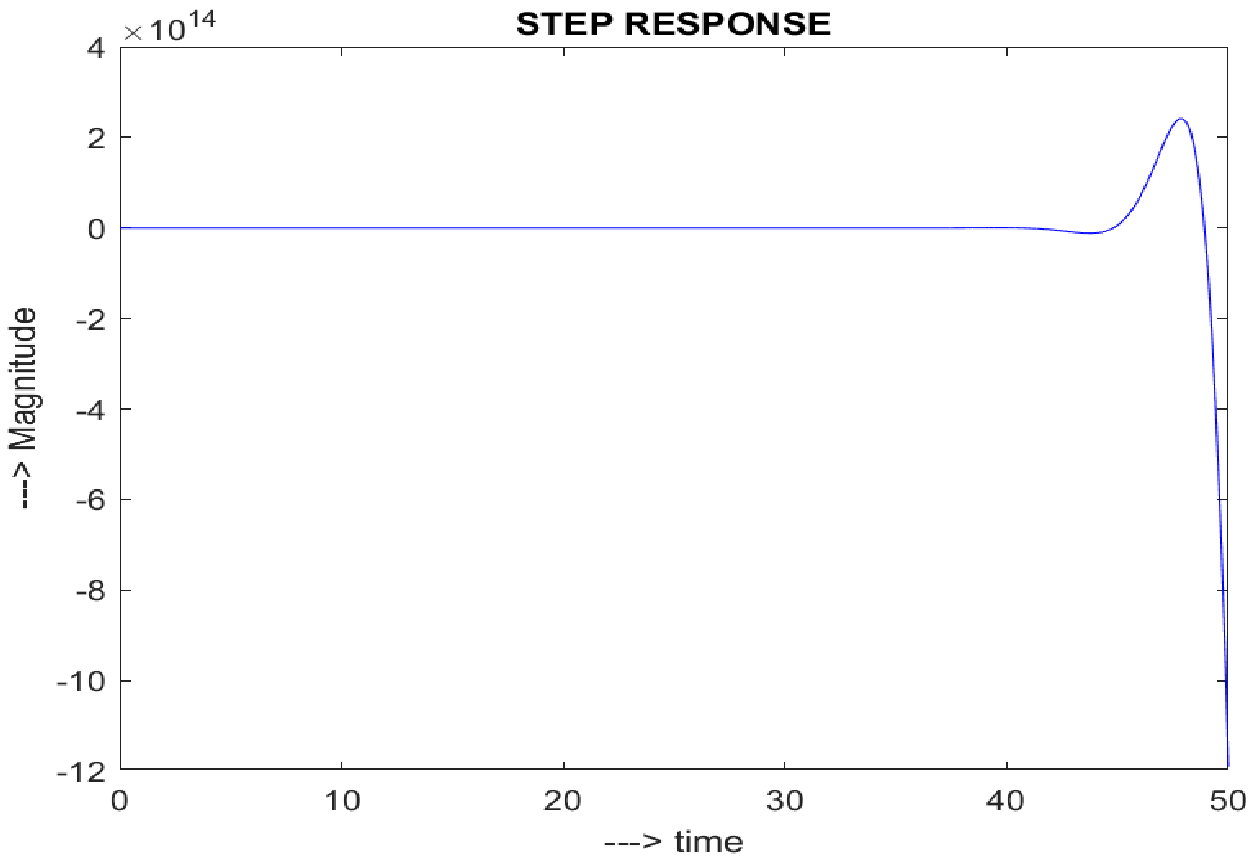
Find out the step response of the given transfer:

$G(s)=(s2+5)/(3s^7+4s^6+5s^5+2s^4+3s^3+6s^2+6s+1)$

PROGRAMME :

```
num = [1 0 5];
denum = [3 4 5 2 3 6 6 1];
g = tf(num,denum);
time_range = 50;
time = 0:0.01:time_range;
y = step(g,time);
plot(time,y,'blue')
hold on
title("STEP RESPONSE")
xlabel("---> time")
ylabel("---> Magnitude")
```

OUTPUT GRAPH :



QUESTION:

Determine the transfer function & find out the step response of the given transfer function g(s)=(s^2+5)/(3s^5+4s^4+5s^3+2s^2+1).

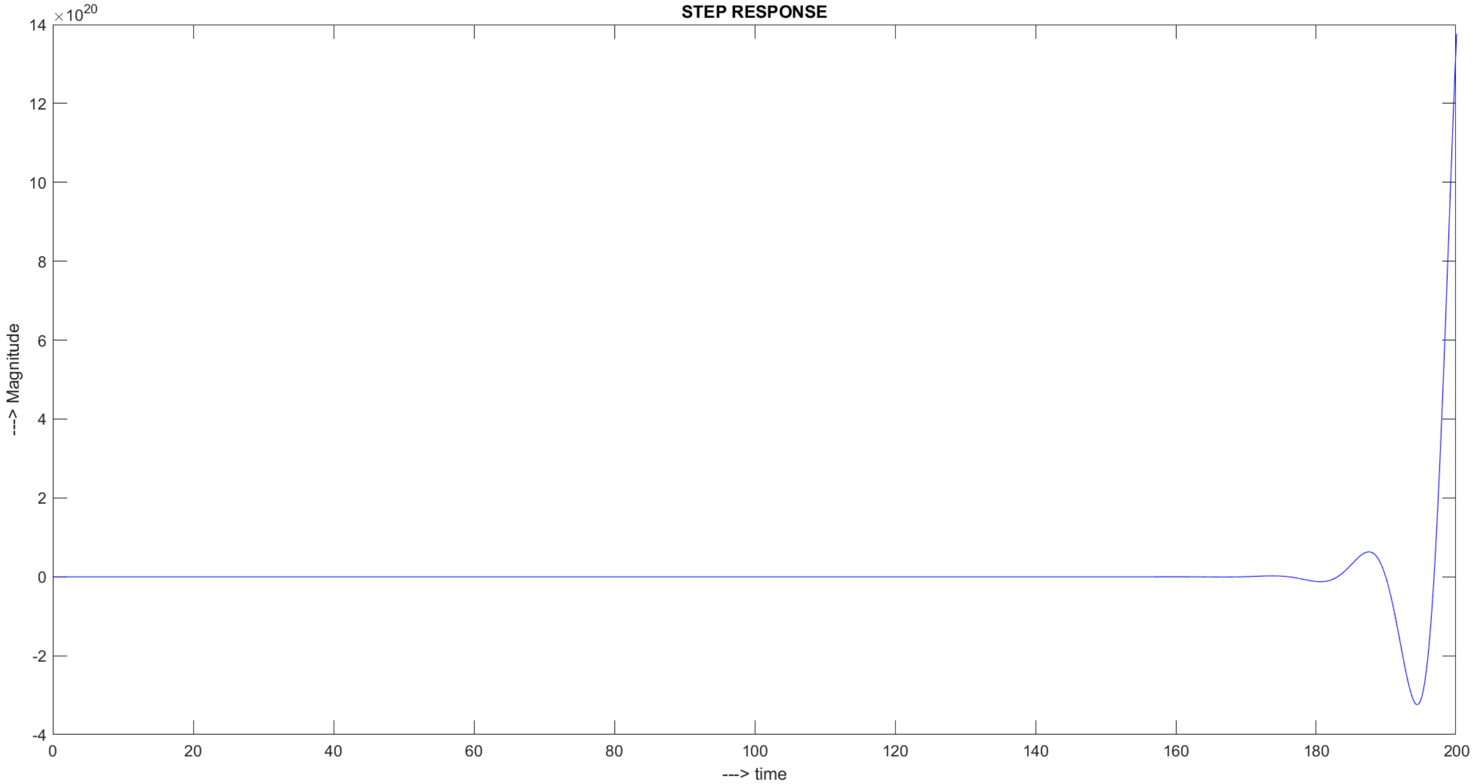
PROGRAMME :

```
num = [1 0 5];
denum = [3 4 5 2 0 1];
g = tf(num,denum)
time_range = 200;
time = 0:0.01:time_range;
y = step(g,time);
plot(time,y,'blue')
hold on
title("STEP RESPONSE")
xlabel("---> time")
ylabel("---> Magnitude")
```

TRANSFER FUNCTION

```
g =

              s^2 + 5
  ---------------------------------
  3 s^5 + 4 s^4 + 5 s^3 + 2 s^2 + 1
```

OUTPUT GRAPH :



EXPERIMENT 2

TITLE: Find & plot the Poles & Zeros from Transfer function and various connections of Transfer functions.

Objective: To study the poles-zeros and different connections of a system transfer function to understand the concept of stability of the system given below.

Theory: The transfer function provides a basis for determining important system response characteristics without solving the complete differential equation. As defined, the transfer function is a rational function in the complex variable Zeros are the value(s) for z where the numerator of the transfer function equals zero. The complex frequencies that make the overall gain of the filter transfer function zero. Poles are the value(s) for z where the denominator of the transfer function equals zero. The complex frequencies that make the overall gain of the filter transfer function infinite.

PROCEDURE:
Type the program in MATLAB editor preferably in M-file
Save and run the program.
Give the required inputs in the command window of MATLAB in matrix format.
tf2zp converts the transfer function filter parameters to pole-zero-gain form.
[z,p,k] = tf2zp(b,a) finds the matrix of zeros z, the vector of poles p, and the associated vector of gains k from the transfer function parameters b and a.

Problem: 01
Find out the pole zero of the given transfer function and locate it on s plane plot
G(*s*)=((*s*^**2**+5) / (3*s*^**7**+4*s*^**6**+5*s*^**5**+2*s*^**4**+3*s*^**3** +6*s*^**2**+6s+1))

PROGRAMME:

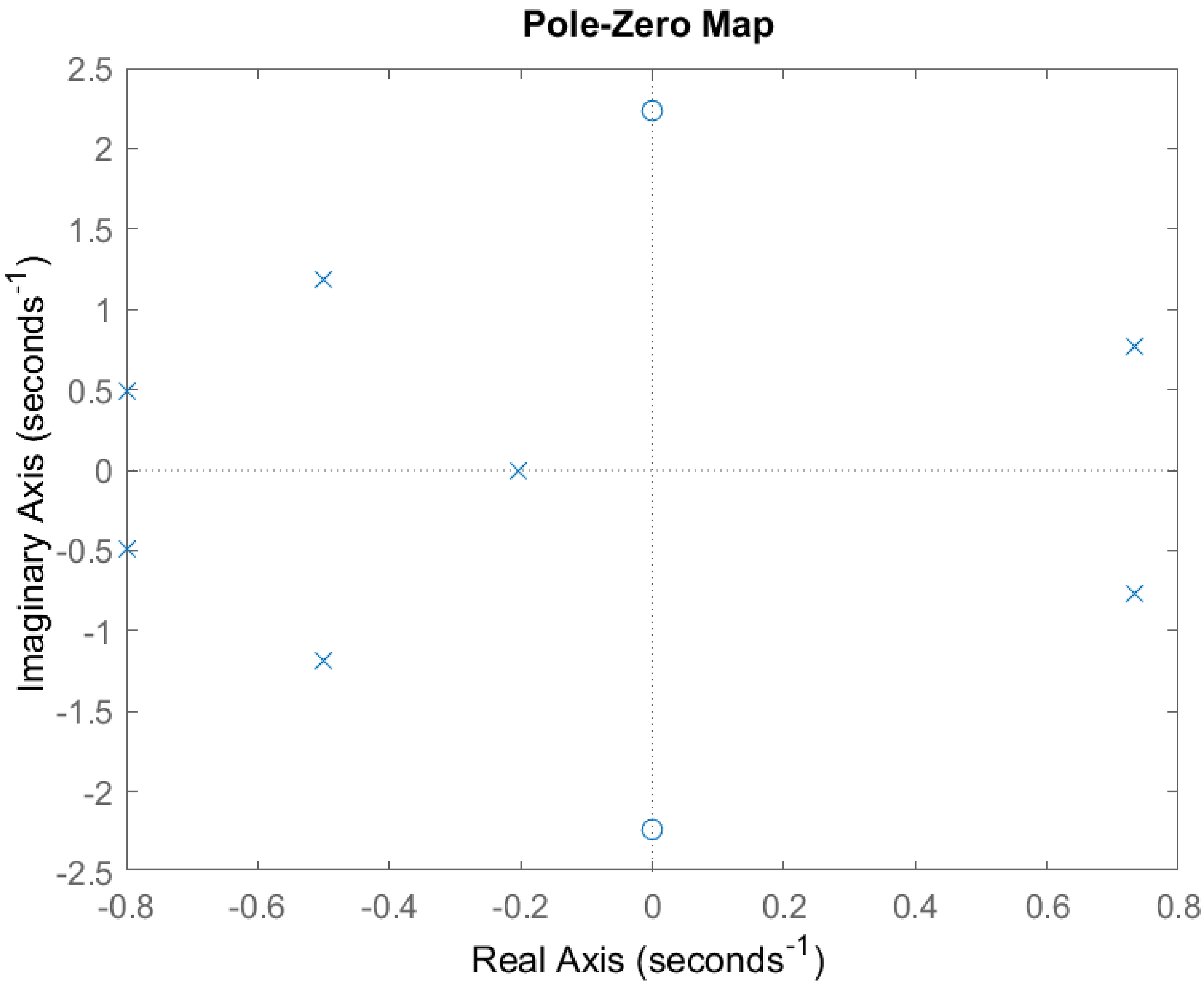
```
num = [1 0 5];
denum = [3 4 5 2 3 6 6 1];
[zero,pole, gain] = tf2zp(num,denum)
pzmap(num,denum)
```

ANS

```
zero =

   0.0000 + 2.2361i
   0.0000 - 2.2361i


pole =

   0.7339 + 0.7639i
   0.7339 - 0.7639i
  -0.4992 + 1.1827i
  -0.4992 - 1.1827i
  -0.7991 + 0.4926i
  -0.7991 - 0.4926i
  -0.2046 + 0.0000i
```

GRAPH:

```
gain =

    0.3333
```



Problem: 02(cascade or series)

(a) T.F= $\frac{2}{s+5}$ **(b) T.F=** $\frac{5}{s^2+3s+10}$

Program: (Overall Transfer function)

```
num1 = [2];
denum1 = [1 5];
num2 = [5];
denum2 = [1 3 10];
g1 = tf(num1,denum1)
g2 = tf(num2,denum2)
g3 = series(g1,g2)
```

OUTPUT:

```
g1 =

    2
  -----
  s + 5

Continuous-time transfer function.
Model Properties

g2 =

        5
  --------------
  s^2 + 3 s + 10

Continuous-time transfer function.
Model Properties

g3 =

            10
  -----------------------
  s^3 + 8 s^2 + 25 s + 50

Continuous-time transfer function.
Model Properties
```

Problem: 03(Parallel)

(a) T.F= $\frac{2}{s+5}$ **(b) T.F=** $\frac{5}{s^2+3s+10}$

Program: (Overall Transfer function)

```
num1 = [2];
denum1 = [1 5];
num2 = [5];
denum2 = [1 3 10];
g1 = tf(num1,denum1)
g2 = tf(num2,denum2)
g3 = parallel(g1,g2)
```

OUTPUT:

```
g1 =

    2
  -----
  s + 5

Continuous-time transfer function.
Model Properties

g2 =

        5
  --------------
  s^2 + 3 s + 10

Continuous-time transfer function.
Model Properties

g3 =

     2 s^2 + 11 s + 45
  -----------------------
  s^3 + 8 s^2 + 25 s + 50

Continuous-time transfer function.
Model Properties
```

Problem: 04 (Positive and negative Feedback)

(a) T.F= $\frac{2}{s+5}$ **(b) T.F=** $\frac{5}{s^2+3s+10}$

Program: (Overall Transfer function)

```
num1 = [2];
denum1 = [1 5];
num2 = [5];
denum2 = [1 3 10];
g1 = tf(num1,denum1)
g2 = tf(num2,denum2)
g3_possitive = feedback(g1,g2,1)
g3_negative = feedback(g1,g2,-1)
```

OUTPUT:

```
g1 =

    2
  -----
  s + 5

Continuous-time transfer function.
Model Properties

g2 =

        5
  --------------
  s^2 + 3 s + 10

Continuous-time transfer function.
Model Properties

g3_possitive =

     2 s^2 + 6 s + 20
  -----------------------
  s^3 + 8 s^2 + 25 s + 40

Continuous-time transfer function.
Model Properties

g3_negative =

     2 s^2 + 6 s + 20
  -----------------------
  s^3 + 8 s^2 + 25 s + 60

Continuous-time transfer function.
Model Properties
```

Problem: 05

Given poles are -3.2+j7.8,-3.2-j7.8,-4.1+j5.9,-4.1-j5.9,-8 and the zeroes are - 0.8+j0.43,-0.8-j0.43,-0.6 with a gain of 0.5

PROGRAMME:

```
poles = [-3.2+7.8j; -3.2-7.8j; -4.1+5.9j; -4.1-5.9j; -8];
zeros = [-0.8+0.43j; -0.8-0.43j; -0.6];
gain = 0.5;
[num,denum] = zp2tf(zeros,poles,gain);
g = tf(num,denum)
```

OUTPUT:

```
g =

              0.5 s^3 + 1.1 s^2 + 0.8924 s + 0.2475
  -----------------------------------------------------------
  s^5 + 22.6 s^4 + 292 s^3 + 2315 s^2 + 1.097e04 s + 2.935e04
```

EXPERIMENT 3

TITLE: Find out the Impulse response and ramp response of a system using MATLAB command and simulation.
Objective: To obtain Impulse response and Ramp response of a system.

THEORY:

System Type

The system type is determined by the form of the loop (or open-loop) transfer function GH(s). In general, the form of GH(s) can be written as -

$$G(s)\ H(s) = \frac{N(s)}{s^n D(s)}$$

Where N(s) and D(s) represent polynomials in s, and sn represents all powers of s that can be factored from the denominator. The number n determines the system type. It is said that the system is a —type n‖ system. As we will show in the following paragraphs, the value of n determines the "type" of steady-state error the system will have for various inputs.

PROCEDURE:

Type the program in the MATLAB editor that is in M-file.
Save and run the program.
Give the required inputs in the command window of MATLAB in matrix format.
Note down the response of the given transfer function

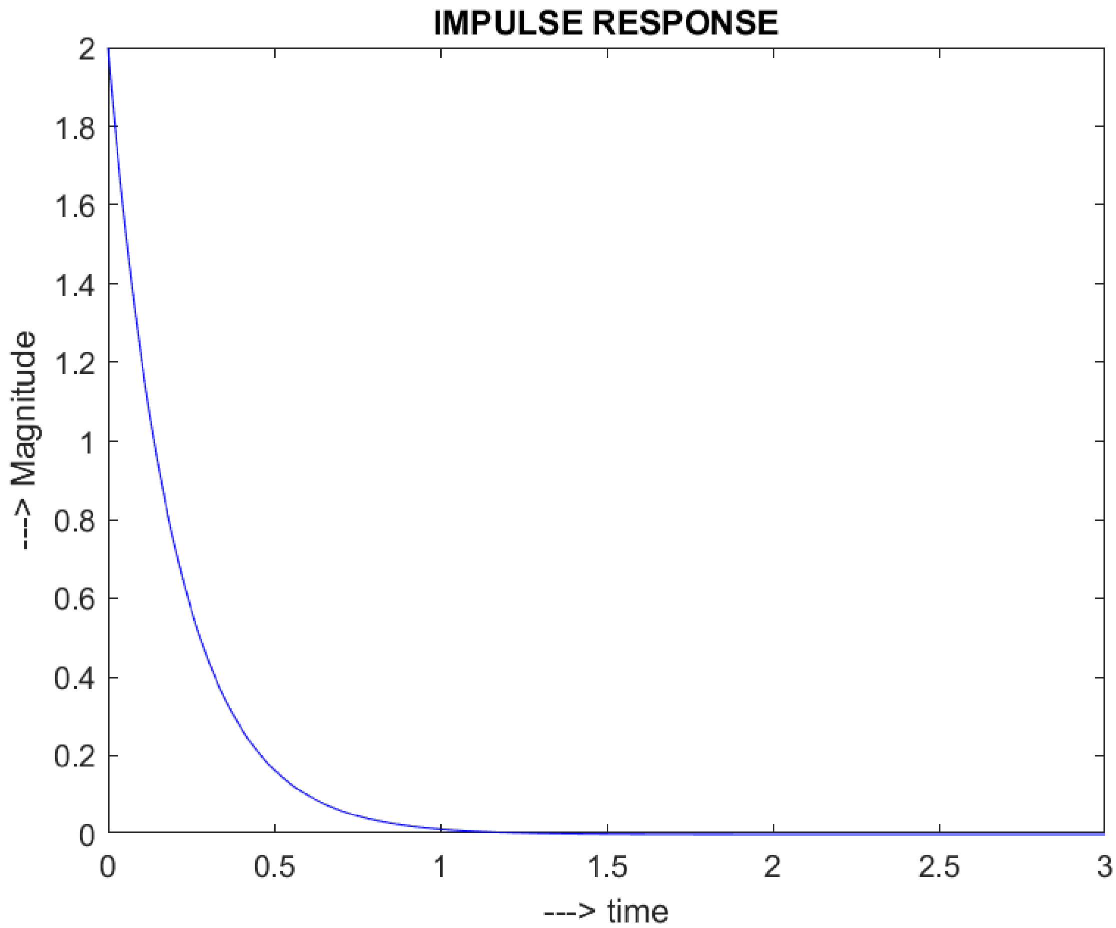
Problem: 01 Obtain the Impulse response of the following system by MATLAB command.

$$G(s) = \frac{s+2}{s^3+5s^2+8s+12}$$

PROGRAMME:

```
num = [1 2];
denum = [1 5 8 12];
g = tf(num,denum);
time_range = 3;
time = 0:0.01:time_range;
y = impulse(g1,time);
plot(time,y,'blue')
hold on
title("IMPULSE RESPONSE")
xlabel("---> time")
ylabel("---> Magnitude")
```

OUTPUT:



Problem: 02

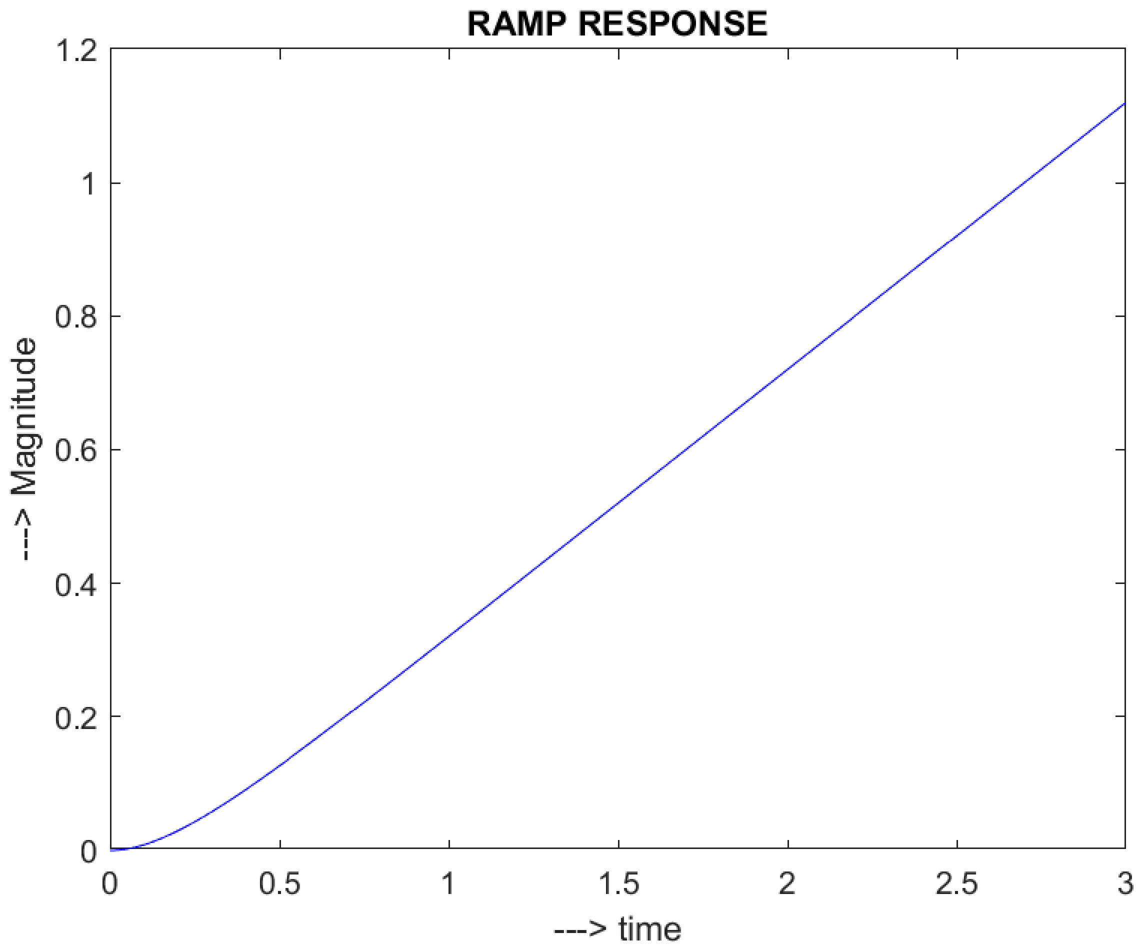
Obtain the Unit Ramp response of the following system by MATLAB command.

$$G(s) = \frac{s+2}{s^3+5s^2+8s+12}$$

Program:

```
num = [1 2];
denum = [1 5 8 12];
g = tf(num,denum);
time_range = 3;
time = 0:0.01:time_range;
ramp = time;
y = lsim(g1,ramp,time);
plot(time,y,'blue')
hold on
title("RAMP RESPONSE")
xlabel("---> time")
ylabel("---> Magnitude")
```

OUTPUT:


RAMP RESPONSE
1.2
1
0.8
0.6
0.4
0.2
0
---> Magnitude
0
0.5
1
1.5
2
2.5
3
---> time

EXPERIMENT 4

TITLE: Determination of closed loop unit step, unit ramp & unit parabolic response for the 1st order type 0, type 1 and type 2 system.
Objective: To obtain the closed loop response for the 1st order type 0, type 1 and type 2 system for different inputs like unit step, unit ramp & unit parabolic and observe whether the response is bounded or unbounded.
Theory:
PROCEDURE:
Type the program in the MATLAB editor that is in M-file.
Save and run the program.
Give the required inputs in the command window of MATLAB in matrix format.
Note down the response of the given transfer function.

Problem : 01(for type 0)

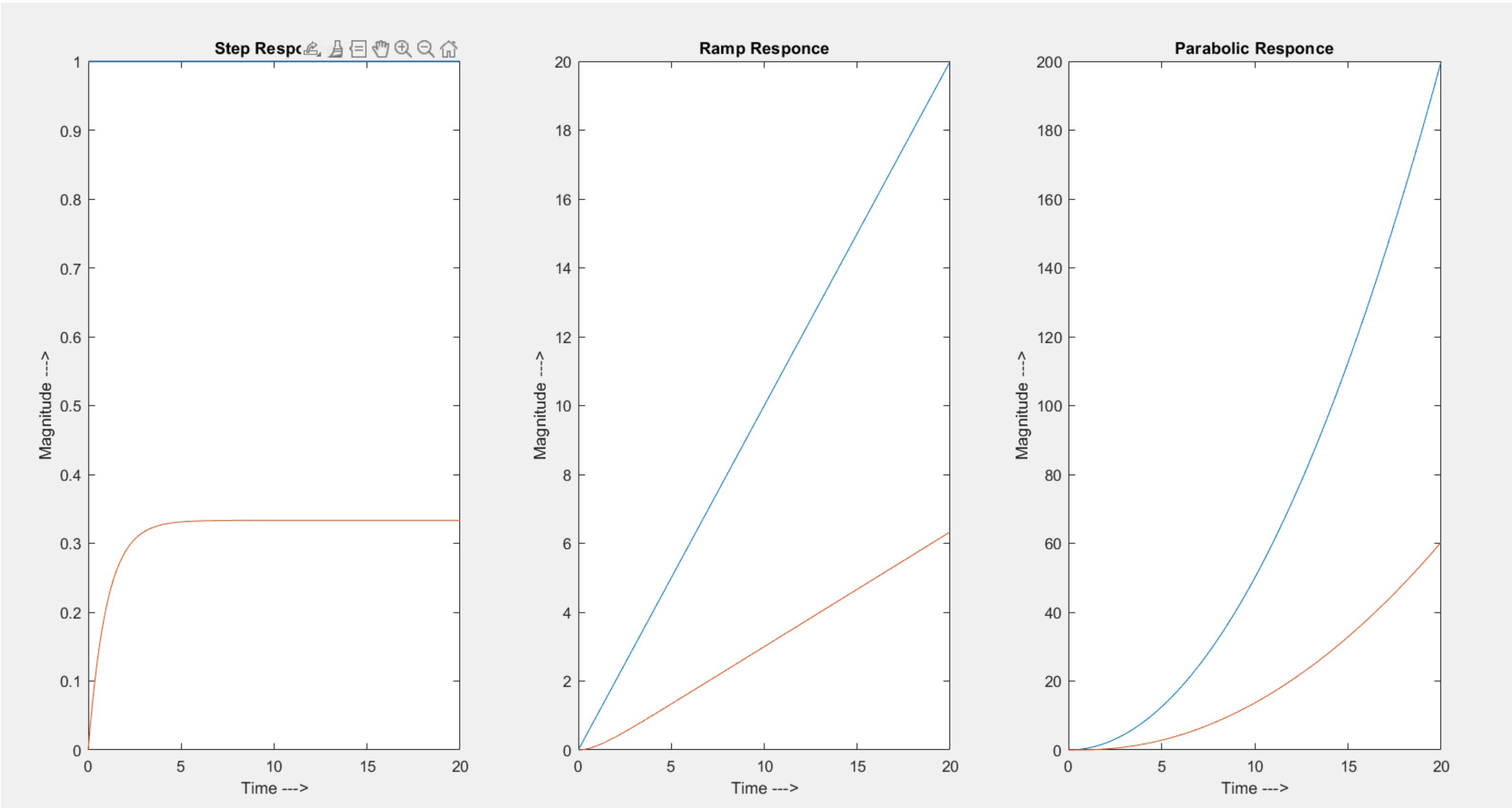
Open loop transfer function of a type 0 system with negative feedback. Find the unit step, unit ramp & unit parabolic response.

$$\frac{c(s)}{r(s)} = \frac{1}{(3s+2)}$$

PROGRAMME:

```
num1 = [1];
denum1 = [3 2];
g1 = tf(num1,denum1)
g2_feedback = feedback(g1,1,-1)
t = 0:0.01:20;
u_step = t./t;
u_ramp = t;
u_parabolic = (t.^2)/2;
y_step = step(g2_feedback,t);
y_ramp = lsim(g2_feedback,u_ramp,t);
y_parabolic = lsim(g2_feedback,u_parabolic,t);
subplot(1,3,1), plot(t,u_step, t,y_step);
title("Step Responce")
xlabel("Time --->")
ylabel("Magnitude --->")
subplot(1,3,2), plot(t,u_ramp,t,y_ramp);
title("Ramp Responce")
xlabel("Time --->")
ylabel("Magnitude --->")
subplot(1,3,3), plot(t,u_parabolic, t,y_parabolic);
title("Parabolic Responce")
xlabel("Time --->")
ylabel("Magnitude --->")
```

OUTPUT:



Problem : 02(for type 1)

Open loop transfer function of a type 1 system with negative feedback. Find the unit step, unit ramp & unit parabolic response.

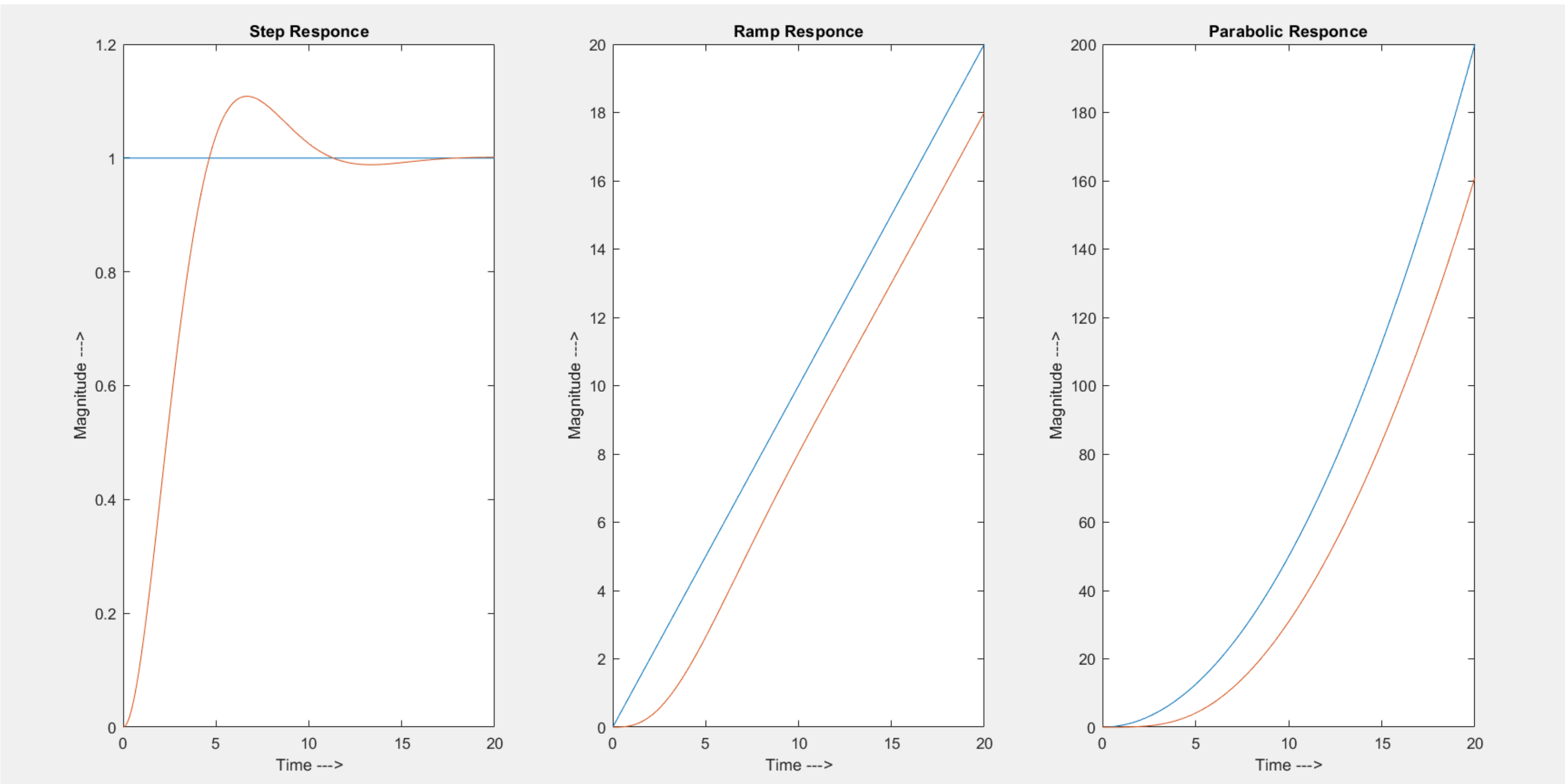
$$\frac{c(s)}{r(s)} = \frac{1}{s\,(3s+2)}$$

Program:

```
num1 = [1];
denum1 = [3 2 0];
g1 = tf(num1,denum1)
g2_feedback = feedback(g1,1,-1)
t = 0:0.01:20;
u_step = t./t;
u_ramp = t;
u_parabolic = (t.^2)/2;
y_step = step(g2_feedback,t);
y_ramp = lsim(g2_feedback,u_ramp,t);
y_parabolic = lsim(g2_feedback,u_parabolic,t);
subplot(1,3,1), plot(t,u_step, t,y_step);
title("Step Responce")
```

```
xlabel("Time --->")
ylabel("Magnitude --->")
subplot(1,3,2), plot(t,u_ramp,t,y_ramp);
title("Ramp Responce")
xlabel("Time --->")
ylabel("Magnitude --->")
subplot(1,3,3), plot(t,u_parabolic, t,y_parabolic);
title("Parabolic Responce")
xlabel("Time --->")
ylabel("Magnitude --->")
```

OUTPUT:



Example: 03(for type 2)

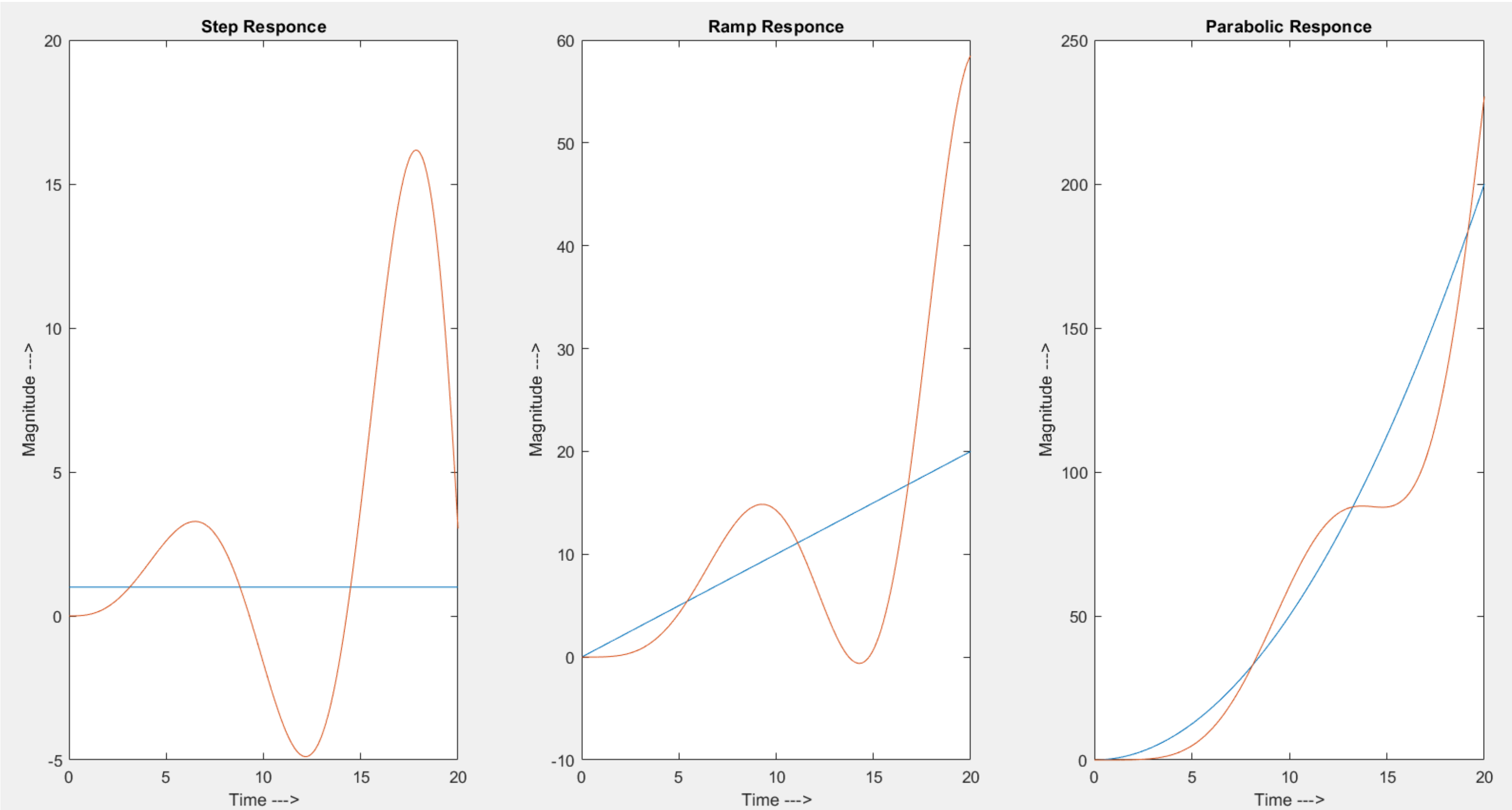
Open loop transfer function of a type 2 system with negative feedback. Find the unit step, unit ramp & unit parabolic response.

$$\frac{c(s)}{r(s)} = \frac{1}{s^2\,(3s+2)}$$

Program:

```
num1 = [1];
denum1 = [3 2 0 0];
g1 = tf(num1,denum1)
g2_feedback = feedback(g1,1,-1)
t = 0:0.01:20;
u_step = t./t;
u_ramp = t;
u_parabolic = (t.^2)/2;
y_step = step(g2_feedback,t);
y_ramp = lsim(g2_feedback,u_ramp,t);
y_parabolic = lsim(g2_feedback,u_parabolic,t);
subplot(1,3,1), plot(t,u_step, t,y_step);
title("Step Responce")
xlabel("Time --->")
ylabel("Magnitude --->")
subplot(1,3,2), plot(t,u_ramp,t,y_ramp);
title("Ramp Responce")
xlabel("Time --->")
ylabel("Magnitude --->")
subplot(1,3,3), plot(t,u_parabolic, t,y_parabolic);
title("Parabolic Responce")
xlabel("Time --->")
ylabel("Magnitude --->")
```

OUTPUT:


Step Responce
Magnitude --->
Time --->
20
15
10
5
0
-5
0
5
10
15
20
Ramp Responce
Magnitude --->
Time --->
60
50
40
30
20
10
0
-10
0
5
10
15
20
Parabolic Responce
Magnitude --->
Time --->
250
200
150
100
50
0
0
5
10
15
20

EXPERIMENT 5

TITLE: Step response of second order system for different damping ratio.
Objective: To obtain poles and zeros of a second order system and also find out the system response for different damping ratio.

Theory:

System Order

The order of a dynamic system is the order of the highest derivative of its governing differential equation. Equivalently, it is the highest power of s in the denominator of its transfer function. The important properties of first, second, and higher order systems will be reviewed in this section.

Second Order Systems

Second order systems are commonly encountered in practice, and are the simplest type of dynamic system to exhibit oscillations. In fact many real higher order systems are modeled as second order to facilitate analysis. Examples include mass-spring-damper systems and RLC circuits.

The general form of the first order differential equation is as follows

$$m\ddot{y}+b\dot{y}+ky=(t) \text{ or } \ddot{y}+2\xi m_n\dot{y}+m_n^2y=k_{dc}m_n^2u$$

The Second order transfer function is-

$$G(s)=\frac{1}{ms^2+bs+k}=\frac{k_{dc}m_n^2}{s^2+2\xi m_n s+m_n^2}$$

DC Gain

The DC gain, k_{dc} again is the ratio of the magnitude of the steady-state step response to the magnitude of the step input, and for stable systems it is the value of the transfer function when $s=0$. For second order systems,

$$k_{dc}=\frac{1}{k}$$

Damping Ratio

The damping ratio is a dimensionless quantity characterizing the energy losses in the system due to such effects as viscous friction or electrical resistance. From the above definitions,

$$\xi=2\sqrt{\frac{k}{m}}$$

Natural Frequency

The natural frequency is the frequency (in rad/s) that the system will oscillate at when there is no damping, $\xi=0$

$$m_n=\sqrt{\frac{k}{m}}$$

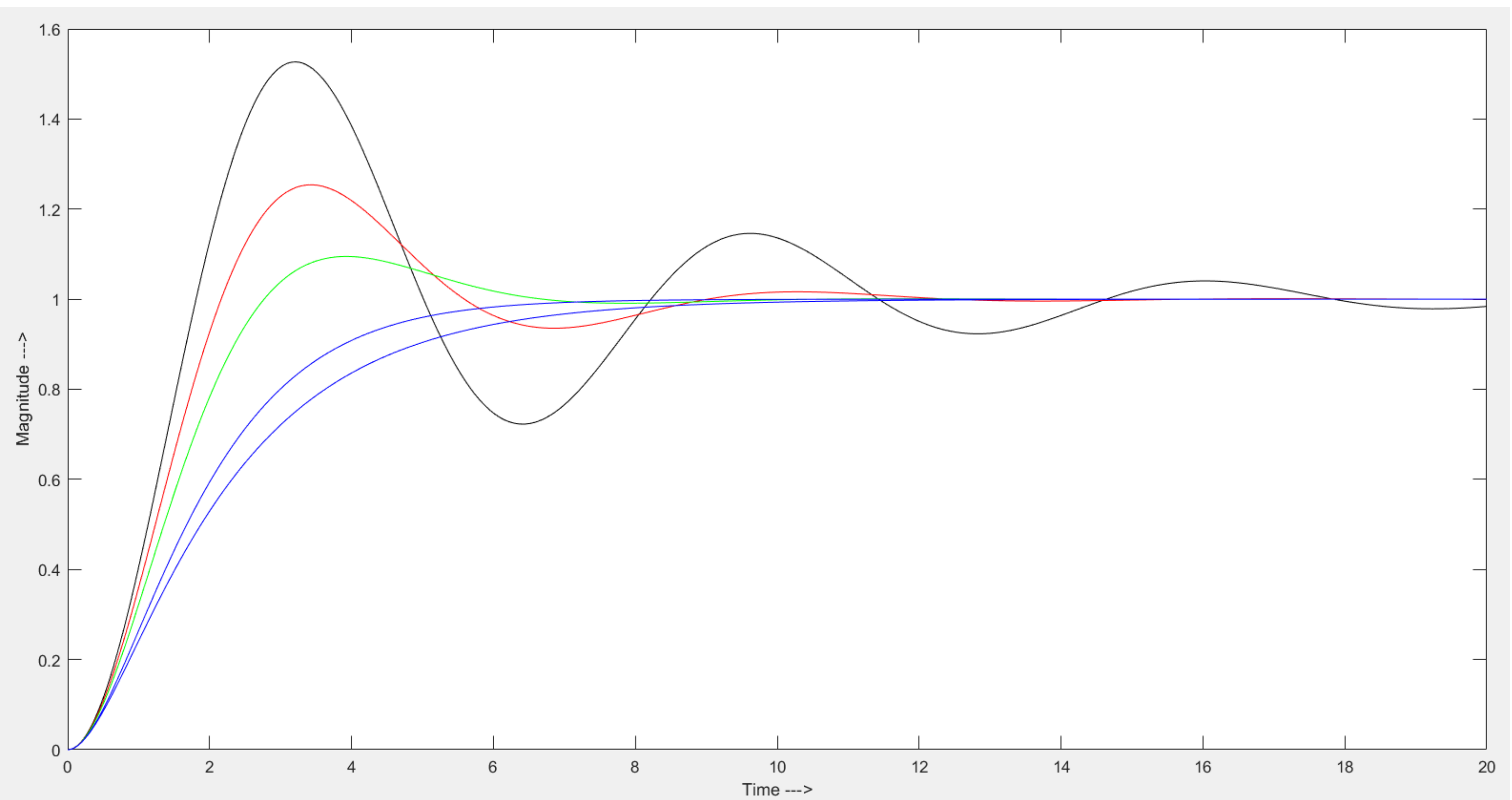
Problem:

Obtain second order step response for different damping coefficient, (ξ=0.2, 0.4, 0.6, 1, 1.2).Here natural frequency of oscillation m_n & Gain(k) of the system is 1.

PROGRAM:

```
t=0:0.01:20;
Omega_n = 1;
gain = 1;
zeta = 0.2;
[n1,d1] = ord2(Omega_n,zeta);
g1 = tf(n1,d1)
y_step1 = step(g1,t);
zeta = 0.4;
[n2,d2] = ord2(Omega_n,zeta);
g2 = tf(n2,d2)
y_step2 = step(g2,t);
zeta = 0.6;
[n3,d3] = ord2(Omega_n,zeta);
g3 = tf(n3,d3)
y_step3 = step(g3,t);
zeta = 1;
[n4,d4] = ord2(Omega_n,zeta);
g4 = tf(n4,d4)
y_step4 = step(g4,t);
zeta = 1.2;
[n5,d5] = ord2(Omega_n,zeta);
g5 = tf(n5,d5)
y_step5 = step(g5,t);
plot(t,y_step1,"k",t,y_step2,"red",t,y_step3,"green",t,y_step4,"blue",t,y_step5,"blue")
```

OUTPUT:



EXPERIMENT 6

TITLE: Determination of stability analysis of linear control system by Root Locus.

Objective: To find out the stability of a system by Root Locus method.

Theory: The root locus technique is a powerful tool for adjusting the location of closed loop poles to achieve the desired system performance by varying one or more system parameters. The path taken by the roots of the characteristics equation when open loop gain K is varied from 0 to ∞ are called root loci (or the path taken by a root of characteristic equation when open loop gain K is varied
From 0 to ∞ is called root locus.).

PROCEDURE:

- Write MATLAB program in the MATLAB specified documents. Then save the program to run it.
- The input is to be mentioned. The syntax "h=tf (num,den)" gives the transfer function and is represented as h. The syntax "rlocus(h)" plots the root locus of the transfer function h.

Problem: 01

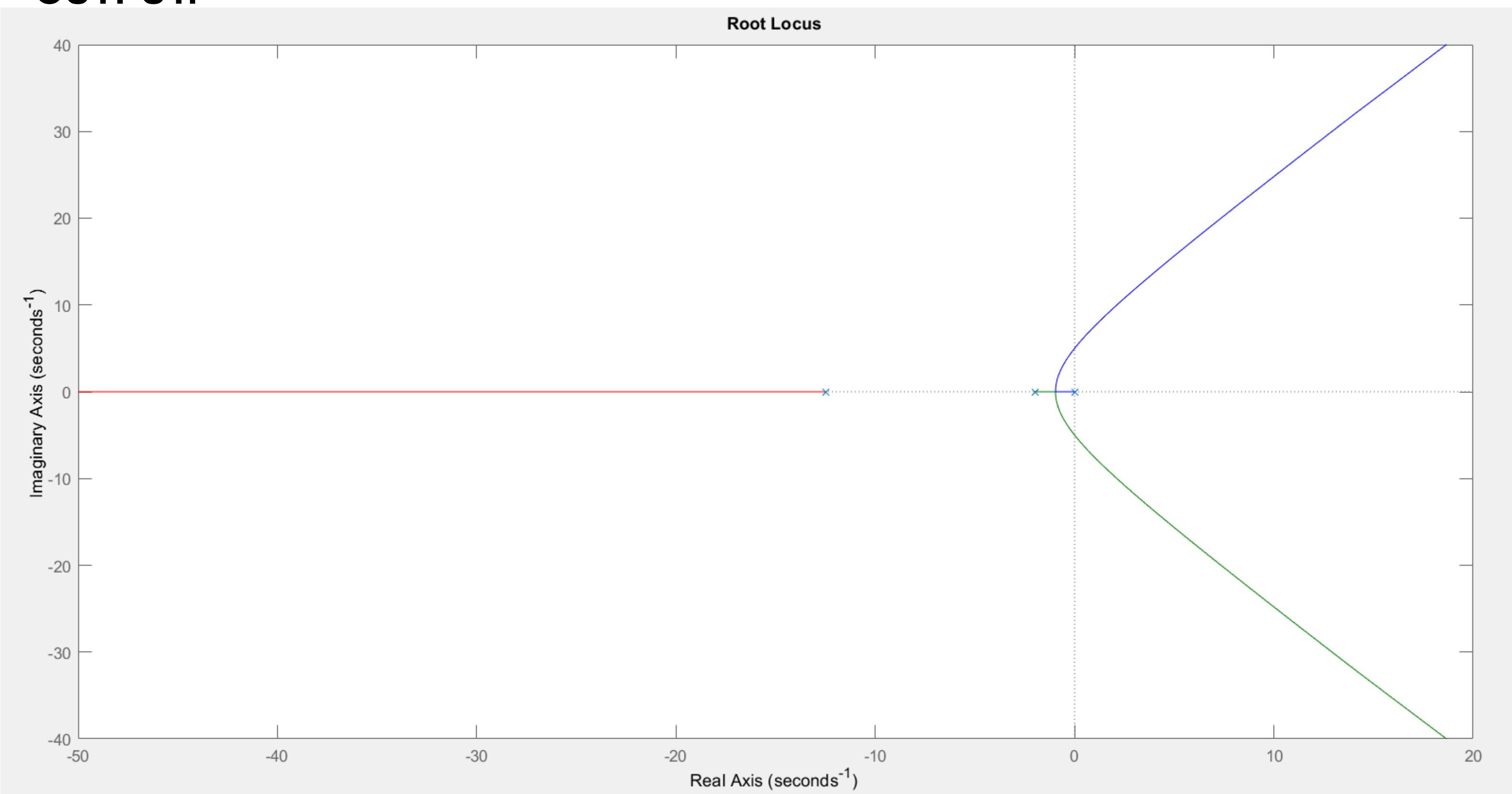
Root Locus Plot for the given system transfer function given below.

$$G(s) = \frac{4}{s(1+0.5s)(1+0.08s)}$$

Program:

```
num = [4];
denum = [0.04 0.58 1 0];
g = tf(num,denum);
rlocus(g)
```

OUTPUT:


Root Locus
40
30
20
10
0
-10
-20
-30
-40
Imaginary Axis (seconds^{-1})
-50
-40
-30
-20
-10
0
10
20
Real Axis (seconds^{-1})

EXPERIMENT 7

TITLE: Determination of stability analysis of linear control system by Bode plot.

Objective: To find out the stability of a linear control system by Bode plot.

Theory:
The performance and characteristics of a system in frequency domain are measured in term of frequency domain specifications. The requirements of a system to be designed are usually specified in terms of these specifications. Bode computes the magnitude and phase of the frequency response of LTI models. When invoked without left-side arguments, bode produces a Bode plot on the screen. The magnitude is plotted in decibels (dB), and the phase in degrees. The decibel calculation for magnitude is computed as 20log10 (|H (jw)|), where H(jw) is the system's frequency response. Bode plots are used to analyze system properties such as the gain margin, phase margin, DC gain, bandwidth, disturbance rejection, and stability

The frequency domain specifications are

- Gain margin
- Phase margin
- Phase cross over frequency
- Gain cross over frequency

GAIN MARGIN
The gain margin K is defined as the reciprocal of the magnitude of open loop transfer function at phase cross over frequency. The frequency at which the phase of open loop transfer function is 180 is called the phase cross over frequency wpc.

PHASE MARGIN
The phase margin is that amount of additional phase lag at the gain cross over frequency required bringing the system to the verge of instability; the gain cross over frequency wgc is the frequency at which the magnitude of open loop transfer function is unity (or it is the frequency at which the db magnitude is zero).

PROCEDURE
• Enter the command window of the MATLAB.
• Create a new M – file by selecting File – New – M – File.
• Type and save the program.
• View the results.
• Analysis the stability of the system for various values of gain

Example: 01

Find out the Bode plot and gain margin, phase margin, gain and phase cross over frequency of the system given below:

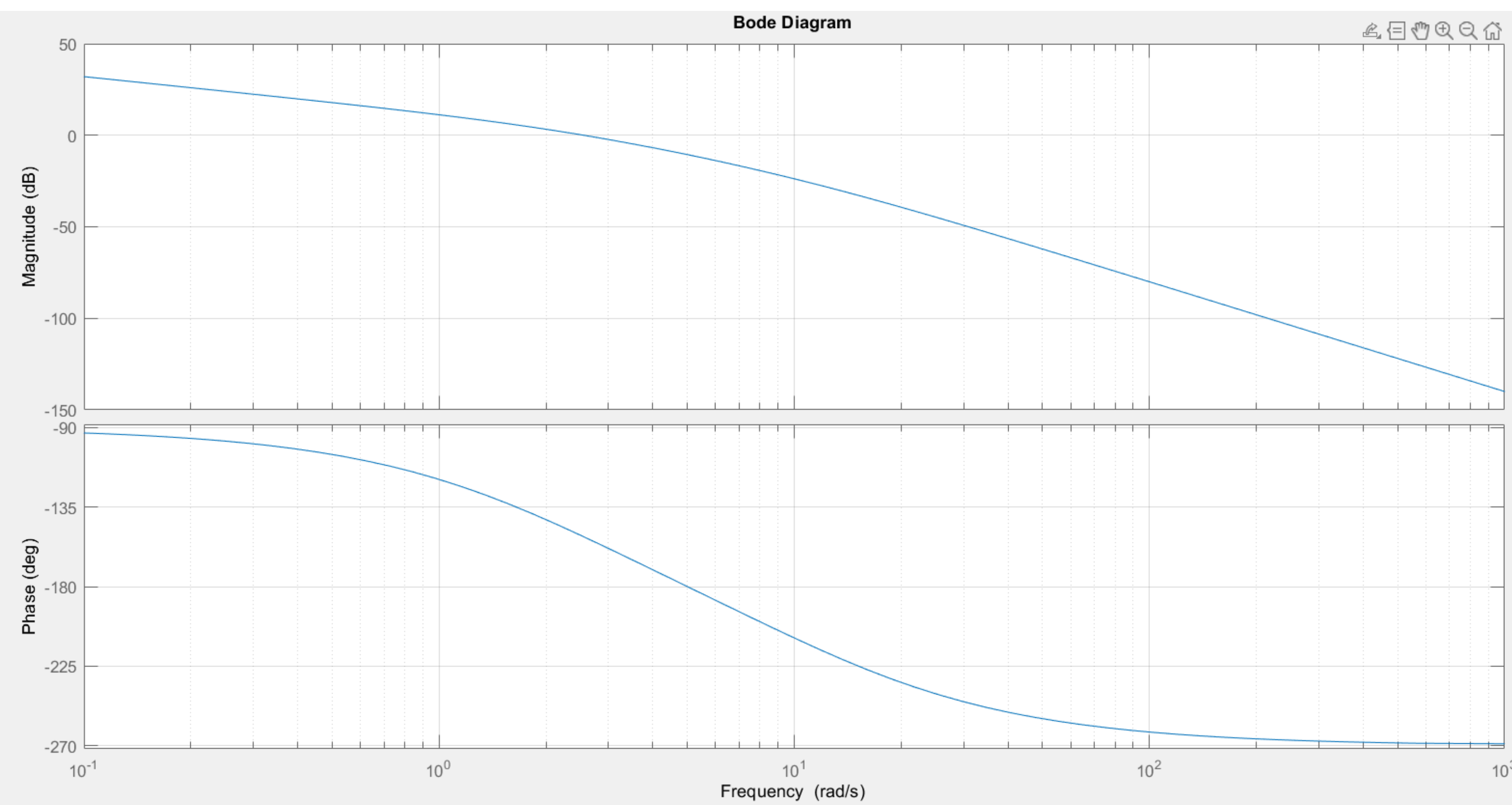
$$G(s) = \frac{4}{s(1+0.5s)(1+0.08s)}$$

Program:

```
numerator = 4;
denominator = [0.04, 0.54, 1, 0];
G = tf(numerator, denominator);
[gain_margin, phase_margin, w_gc, w_pc] = margin(G);
bode(G)
grid on
fprintf('Gain Margin (dB): %.2f\n', 20*log10(gain_margin));
fprintf('Phase Margin (degrees): %.2f\n', phase_margin);
fprintf('Gain Crossover Frequency (rad/s): %.2f\n', w_gc);
fprintf('Phase Crossover Frequency (rad/s): %.2f\n', w_pc);
```

OUTPUT:

```
Gain Margin (dB): 10.57
Phase Margin (degrees): 28.16
Gain Crossover Frequency (rad/s): 5.00
Phase Crossover Frequency (rad/s): 2.56
```


Bode Diagram
50
0
-50
-100
-150
Magnitude (dB)
-90
-135
-180
-225
-270
Phase (deg)
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)

EXPERIMENT 8

TITLE: Determination of stability analysis of linear control system by Nyquist plot.

Objective: To find out the stability of a linear control system by Nyquist plot.

Theory: A nyquist plot is used in automatic control and signal processing for assessing the stability of a system with feedback. It is represented by a graph in polar coordinates in which the gain and phase of a frequency response are plotted. The plot of these phasor quantities shows the phase as the angle and the magnitude as the distance from the origin. This plot combines the two types of Bode plot — magnitude and phase — on a single graph with frequencry as a parameter along the curve. Nyquist calculates the Nyquist frequency response of LTI models. When invoked without left-hand arguments, nyquist produces a Nyquist plot on the screen. Nyquist plots are used to analyze system properties including gain margin, phase margin, and stability.

The nyquist stability criterion , provides a simple test for stability of a closed-loop control system by examining the open-loop system's Nyquist plot. Stability of the closed-loop control system may be determined directly by computing the poles of the closed-loop transfer function. The Nyquist Criteria can tell us things about the *frequency characteristics* of the system.

The Nyquist Contour

The nyquist contour, the contour that makes the entire nyquist criterion work, must encircle the entire right half of the complex s plane. Remember that if a pole to the closed-loop transfer function (or equivalently a zero of the characteristic equation) lies in the right-half of the s plane, the system is an unstable system.To satisfy this requirement, the nyquist contour takes the shape of an infinite semi-circle that encircles the entire right-half of the s plane.

Nyquist Criteria

Let us first introduce the most important equation when dealing with the Nyquist criterion:

$$\mathbf{N=Z-P}$$

Where:

N is the number of encirclements of the (-1, 0) point. Z is the number of zeros of the characteristic equation.

P is the number of poles of the characteristic equation. With this equation stated, we can now state the -

Nyquist Stability Criterion: Nyquist Stability Criterion a feedback control system is stable, if and only if the contour ΓF(s) in the F(s) plane does not encircle the (-1, 0) point when P is 0.

A feedback control system is stable, if and only if the contour ΓF(s) in the F(s) plane encircles the (-1, 0) point a number of times equal to the number of poles of F(s) enclosed by Γ. In other words, if P is zero then N must equal zero. Otherwise, N must equal P. Essentially, we are saying that Z must always equal zero, because Z is the number of zeros of the characteristic equation (and therefore the number of poles of the closed-loop transfer function) that are in the right-half of the s plane.

PROCEDUR:

- Write MATLAB program in the MATLAB editor document.
- Then save and run the program. Give the required input.
- The syntax "tf(num ,den)" solves the given transfer function.
- "nyquist(sys)", nyquist calculates the Nyquist frequency response of LTI models. When invoked without left-hand arguments, nyquist produces a Nyquist plot on the screen. Nyquist plots are used to analyze system properties including gain margin, phase margin, and stability. "nyquist(sys)" plots the Nyquist response of an arbitrary LTI model sys.
- "[Gm, Pm,Wcg,Wcp] = margin(sys)", margin calculates the minimum gain margin, phase margin, and associated crossover frequencies of SISO open-loop models. The gain and phase margins indicate the relative stability of the control system when the loop is closed.

Problem: 01

$$G(s) = \frac{4}{s(1+0.5s)(1+0.08s)}$$

PROGRAMME:

```
numerator = 4;
denominator = [0.04, 0.58, 1, 0];
G = tf(numerator, denominator);
[gain_margin, phase_margin, w_gc, w_pc] = margin(G);
nyquist(G)
fprintf('Gain Margin (dB): %.2f\n', 20*log10(gain_margin));
fprintf('Phase Margin (degrees): %.2f\n', phase_margin);
fprintf('Gain Crossover Frequency (rad/s): %.2f\n', w_gc);
fprintf('Phase Crossover Frequency (rad/s): %.2f\n', w_pc);
```

OUTPUT:

```
Gain Margin (dB): 11.19
Phase Margin (degrees): 27.83
Gain Crossover Frequency (rad/s): 5.00
Phase Crossover Frequency (rad/s): 2.47
```