KIET Group of Institutions, Ghaziabad
Electronics and Communication Engineering

Control System Lab Report

Submitted to: Mr Vipin Verma and Dr. Himanshu Chaudhary

(2023-24)

Submitted By: Ira Nafees
Course: Btech
Branch: ECE
Semester: 6th
Sec: B
Roll No: 2100290310072

INDEX

Experiment – 2

OBJECTIVE: Determine transpose, inverse value of given matrix.

SOFTWARE REQUIRED: MATLAB

PROGRAM:

TRANSPOSE OF MATRIX

```
clc;
clear all;
close all;
A=[231;451;231]
B = transpose(A)
```

INVERSE OF MATRIX

```
clc;
clear all;
A=[12;34]
B = det(A)
for i = 1:2
   for j = 1:2
      D (i, j) = ((-1) ^ (i+ j)) *(A(3-i,3-j));
 E (i, j) = transpose(D(i,j))
      I (i, j) = (E (i,
      j)/B); C=I'
   end
end
```

WORKSPACE:

Workspace

Name	Value	Size	Class
A	[1,2;3,4]	2x2	double
B	-2	1x1	double
C	[-2,1;1.5000,...	2x2	double
D	[4,-3;-2,1]	2x2	double
E	[4,-3;-2,1]	2x2	double
I	[-2,1.5000;1,...	2x2	double
i	2	1x1	double
j	2	1x1	double

Workspace

Name	Value	Size	Class
A	[2,3,1;4,5,1;...	3x3	double
B	[2,4,2;3,5,3;...	3x3	double

COMMAND WINDOW:

```
A=1 2
   3  4
B=-2
E = 4
C=-2
E = 4    -3
C = -2.0000
   1.5000
E = 4    -3
  -2     0
C = -2.0000   1.0000
    1.5000       0
E = 4    -3
  -2     1
C = -2.0000   1.0000
    1.5000   -0.5000

A = 2     3    1
   4     5    1
   2     3    1

B = 2     4    2
   3     5    3
   1     1    1
```

Experiment – 3

OBJECTIVE: PLOT POLE-ZERO CONFIGURATION IN S-PLANE FOR THE GIVEN TRANSFER FUNCTION USING MATLAB

SOFTWARE REQUIRED: MATLAB

PROGRAM:

```
clc;
clear all;
H = tf ([3 2 1 2 1], [6 6 2 6 2 1])
Y = zpk (1, [-2, -3], 6)
subplot (2,1,1);
pzmap (H);
grid on;
subplot (2,1,2);
pzmap(Y);
```

WORKSPACE:

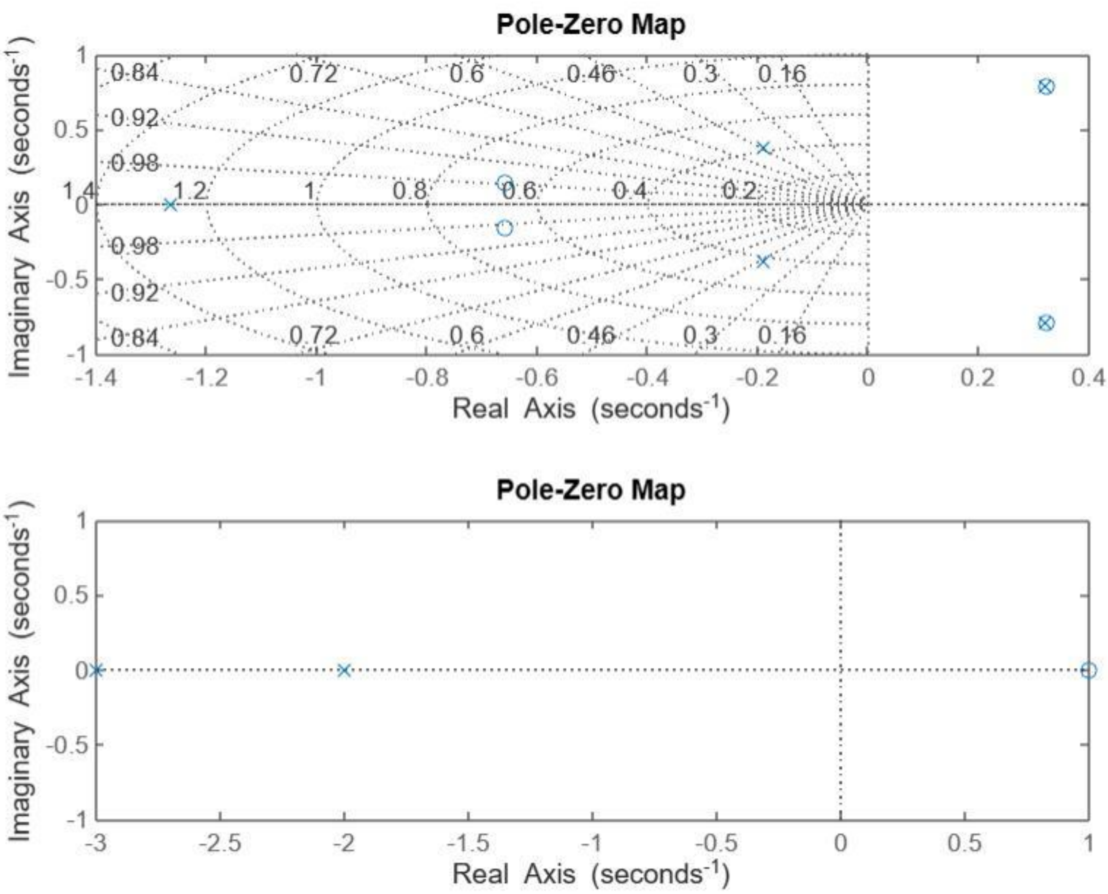
Name	Value	Size	Class
H	*1x1 tf*	1x1	tf
Y	*1x1 zpk*	1x1	zpk

COMMAND WINDOW:

```
H = 3 s^4 + 2 s^3 + s^2 + 2 s + 1
    ---------------------------------------
    6 s^5 + 6 s^4 + 2 s^3 + 6 s^2 + 2 s + 1

Continuous-time transfer function.


Y = 6 (s-1)
    -----------
   (s+2) (s+3)
```

Continuous-time zero/pole/gain model.

FIGURE:



Experiment – 4

OBJECTIVE: Determine the transfer function for the closed loop system in the block diagram representation.

SOFTWARE REQUIRED: MATLAB

PROGRAM:

```
clc;
close all;
clear all;
%for system1
G1 = tf ([1 0], [1 1])
H1 = tf ([1], [1 0])
G2 = tf ([1 4], [1 3 1])
H2 = tf ([1 0], [1 2])
F1 = feedback (G1, H1)
F2 = feedback (G2, H2)
A = series (F1, F2)
H = tf ([1 0], [1 3])
F3 = feedback (A, H)
f =F3(step);
```

WORKSPACE:

Name	Value	Size	Class
A	1x1 tf	1x1	tf
F1	1x1 tf	1x1	tf
F2	1x1 tf	1x1	tf
F3	1x1 tf	1x1	tf
G1	1x1 tf	1x1	tf
G2	1x1 tf	1x1	tf
H	1x1 tf	1x1	tf
H1	1x1 tf	1x1	tf
H2	1x1 tf	1x1	tf

COMMAND WINDOW:

```
G1 =  s
        -----
      s + 1
 Continuous-time transfer function.
H1=1
         -
         s
 Continuous-time transfer function.
G2 = s + 4
        -------------
      s^2 + 3 s + 1
 Continuous-time transfer function.
H2 = s
       -----
      s + 2
 Continuous-time transfer function.
F1 = s^2
       ---------
      s^2 + 2 s
 Continuous-time transfer function.
F2 = s^2 + 6 s + 8
       ----------------------
      s^3 + 6 s^2 + 11 s + 2
 Continuous-time transfer function.
A = s^4 + 6 s^3 + 8 s^2
      -----------------------------------
    s^5 + 8 s^4 + 23 s^3 + 24 s^2 + 4 s
 Continuous-time transfer function.
```

H = s

-----

s + 3

Continuous-time transfer function.

F3 = s^5 + 9 s^4 + 26 s^3 + 24 s^2

----

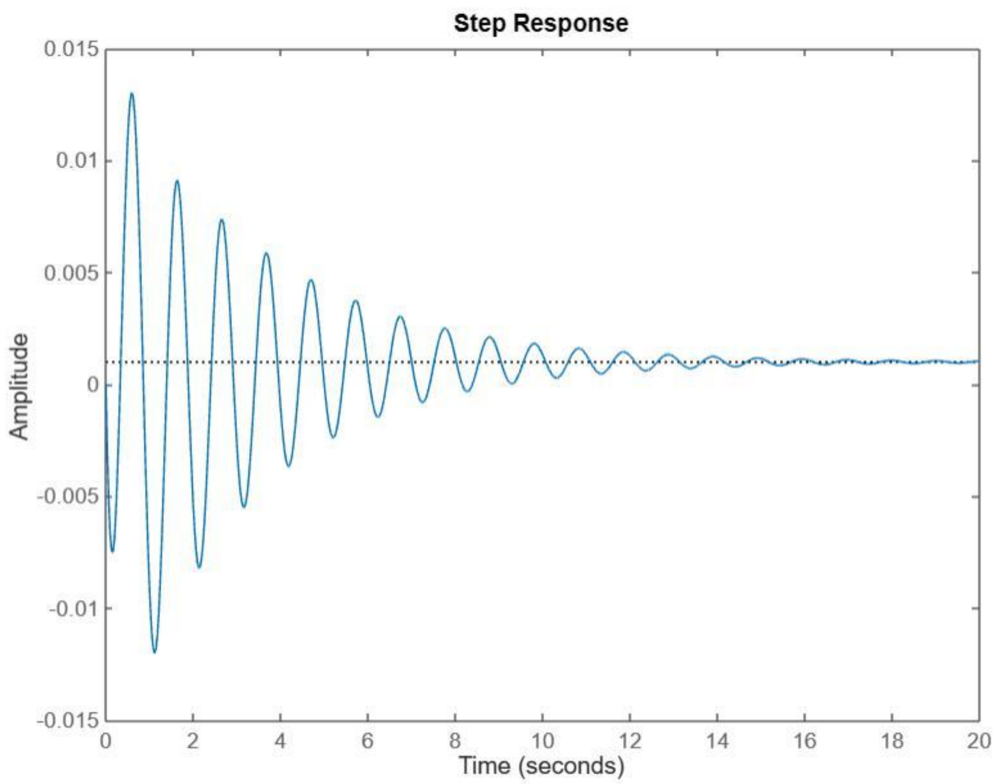
s^6 + 12 s^5 + 53 s^4 + 101 s^3 + 76 s^2 + 12

s Continuous-time transfer function.

num = 0 -0.1022 0.0316 0.1934 -0.1795 0.1620

den = 1.0000 6.2190 50.6538 222.7866 359.5180 162.7478

FIGURE:



Experiment – 5

OBJECTIVE: Determine the time response of the given system subjected to any arbitrary input.

SOFTWARE REQUIRED: MATLAB

PROGRAM:

```
clc;
close all;
clear all;
h1 = tf ([9], [1 6 9])
t = linspace (1,15,50);
%step response
Subplot (3,2,1)
step(h1)
%impulse response
Subplot (3,2,2)
impulse(h1)
%sinusoidal
r =sin(t);
%ramp
s=2*t;
%parabolic q =
5*(t.^2);
subplot (3,2,3);
lsim (h1, r, t);
subplot (3,2,4);
lsim (h1, s, t);
subplot (3,2,5);
lsim (h1, q, t);
```

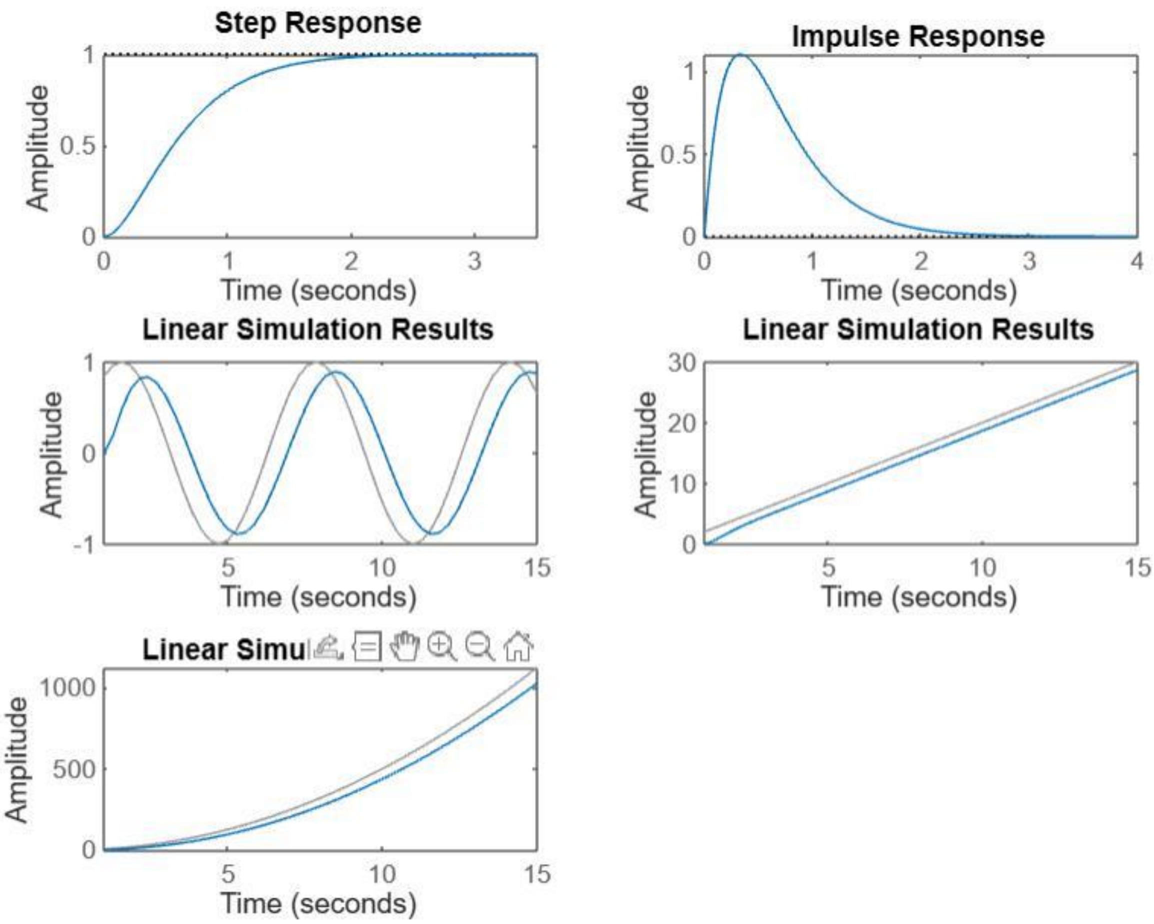
WINDOWSPACE:

Name	Value	Size	Class
h1	*1x1 tf*	1x1	tf
q	*1x50 double*	1x50	double
r	*1x50 double*	1x50	double
s	*1x50 double*	1x50	double
t	*1x50 double*	1x50	double

COMMAND WINDOW:

```
h1 =        9
     -------------
     s^2 + 6 s + 9
  Continuous-time transfer function.
```

FIGURE:



Experiment – 6

OBJECTIVE: Plot unit step response of the given transfer function and find delay time, rise time, peak time, peak time, peak overshoot and settling time.

SOFTWARE REQUIRED: MATLAB

PROGRAM:

```
clc
clear all
close all
g = tf ([25], [1 6 25])
wn = 5
zt = 6/(wn*2)
wd = wn*sqrt(1-(zt*zt))
tp =pi/wd
mp =exp((-1*zt*pi)/sqrt(1-(zt*zt)))
po = (2*pi)/wd
st = 4/(zt*wn)
a = acos(zt);
rt = (pi-a)/wd
t = 0:0.01:10;
[y, t] = step(g);
Plot (t, y)
```

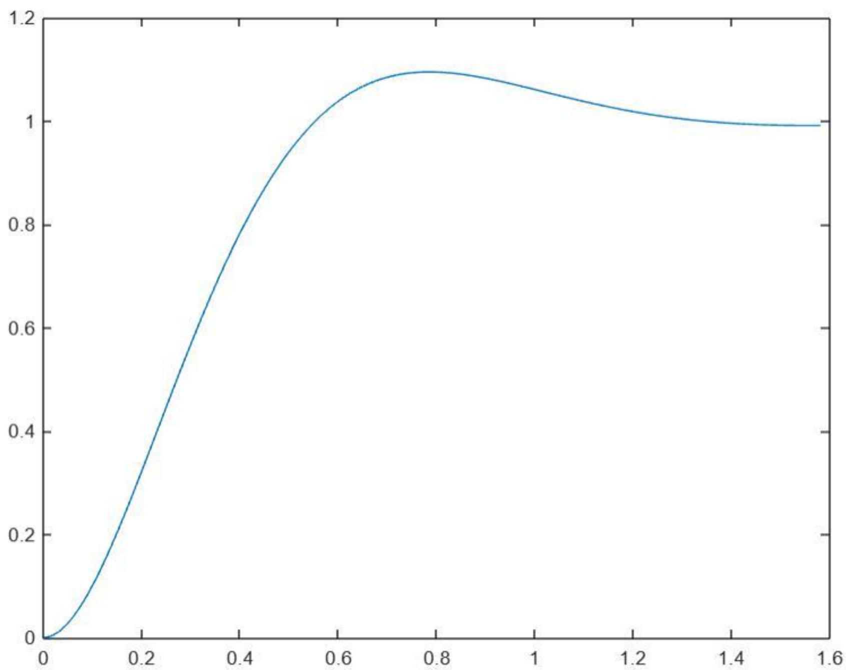
WINDOWSPACE:

Name	Value	Size	Class
a	0.9273	1x1	double
g	*1x1 tf*	1x1	tf
mp	0.0948	1x1	double
po	1.5708	1x1	double
rt	0.5536	1x1	double
st	1.3333	1x1	double
t	*104x1 double*	104x1	double
tp	0.7854	1x1	double
wd	4	1x1	double
wn	5	1x1	double
y	*104x1 double*	104x1	double
zt	0.6000	1x1	double

COMMAND WINDOW:

```
g =      25
       --------------
     s^2 + 6 s + 25

 Continuous-time transfer function.

wn = 5

zt = 0.6000

wd = 4

tp = 0.7854

mp = 0.0948

po = 1.5708

st = 1.3333

rt = 0.5536
```

FIGURE:



Experiment – 7

OBJECTIVE: **Determine the steady state error of a given transfer function.**

SOFTWARE REQUIRED: MATLAB

PROGRAM:

```
clc
close all
clear all
g = tf ([4], [1 3 4])
a = step(g);
[y, t] = step(g);
i=length(y)
z = y(i)
c = y(end)
sserror = abs(1-c)
```

WINDOWSPACE:

Name	Value	Size	Class
a	*139x1 double*	139x1	double
c	0.9999	1x1	double
g	*1x1 tf*	1x1	tf
i	139	1x1	double
sserror	1.1622e-04	1x1	double
t	*139x1 double*	139x1	double
y	*139x1 double*	139x1	double
z	0.9999	1x1	double

COMMAND WINDOW:

```
g =      4
    -------------
   s^2 + 3 s + 4
```

Continuous-time transfer function.

i = 139

z = 0.9999

c = 0.9999

sserror = 1.1622e-04

Experiment – 8

OBJECTIVE: **Plot root locus of a given transfer function, locate closed loop pole for the different value of K**

SOFTWARE REQUIRED: MATLAB

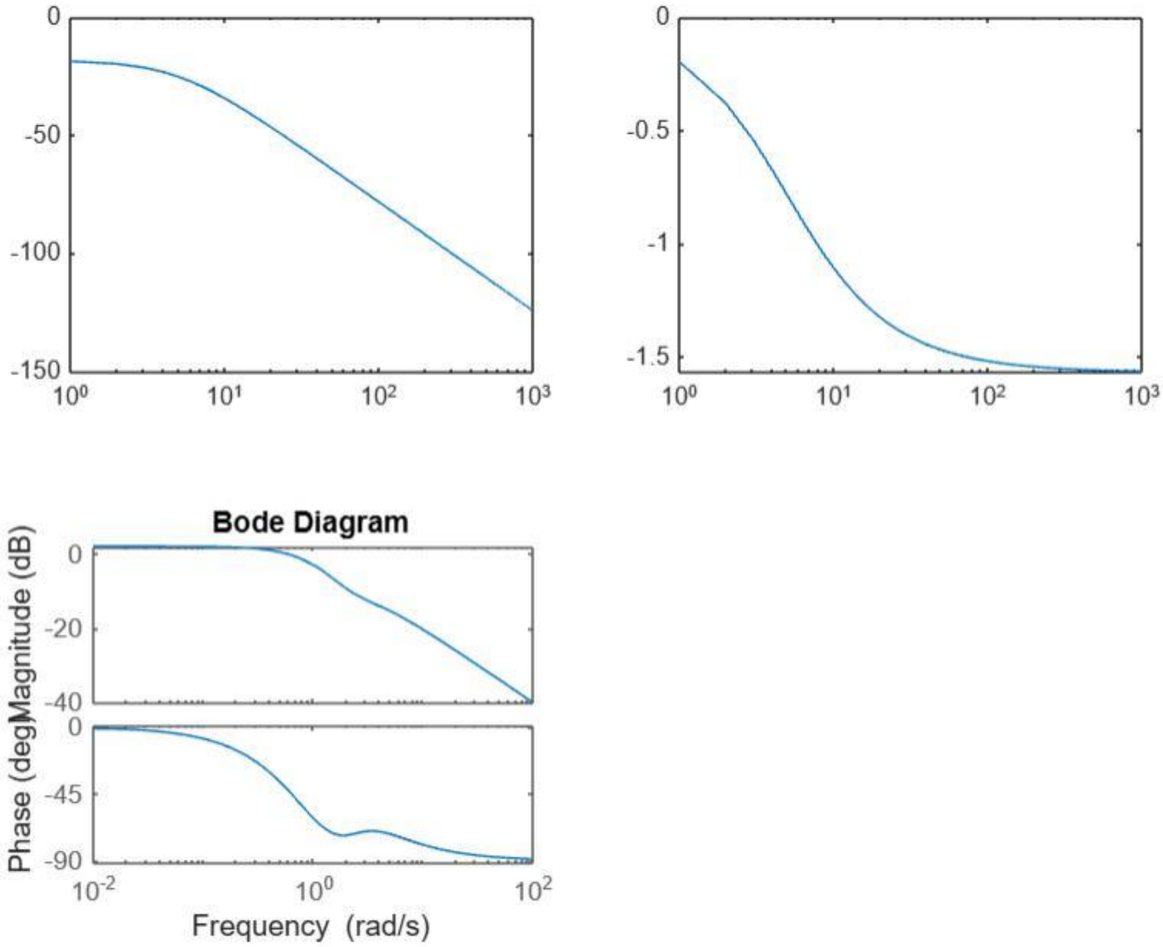
PROGRAM:

```
clc
 close all
 g = tf ([1 3 5], [1 5 8 4]);
 w = linspace(1,1000,1000);
 for i = 1:1:1000
    mag(i) = 2/sqrt(25+w(i).^2);
    magdb(i) = 20*log(mag(i));
    phase(i) = -atan(w(i)/5);
 end
 subplot(2,2,1)
 semilogx(w,magdb)
 subplot(2,2,2)
 semilogx(w,phase)
 subplot(2,2,3)
 bodeplot(g)
```

COMMAND WINDOW:

```
g =

      s^2 + 3 s + 5
  ---------------------
  s^3 + 5 s^2 + 8 s + 4

Continuous-time transfer function.
```

FIGURE:


0
-50
-100
-150
10^0
10^1
10^2
10^3
0
-0.5
-1
-1.5
10^0
10^1
10^2
10^3
Bode Diagram
Magnitude (dB)
0
-20
-40
Phase (deg)
0
-45
-90
10^{-2}
10^0
10^2
Frequency (rad/s)

Experiment – 9

OBJECTIVE: Plot Bode plot of given transfer function. Also determine gain and phase margins

SOFTWARE REQUIRED: MATLAB

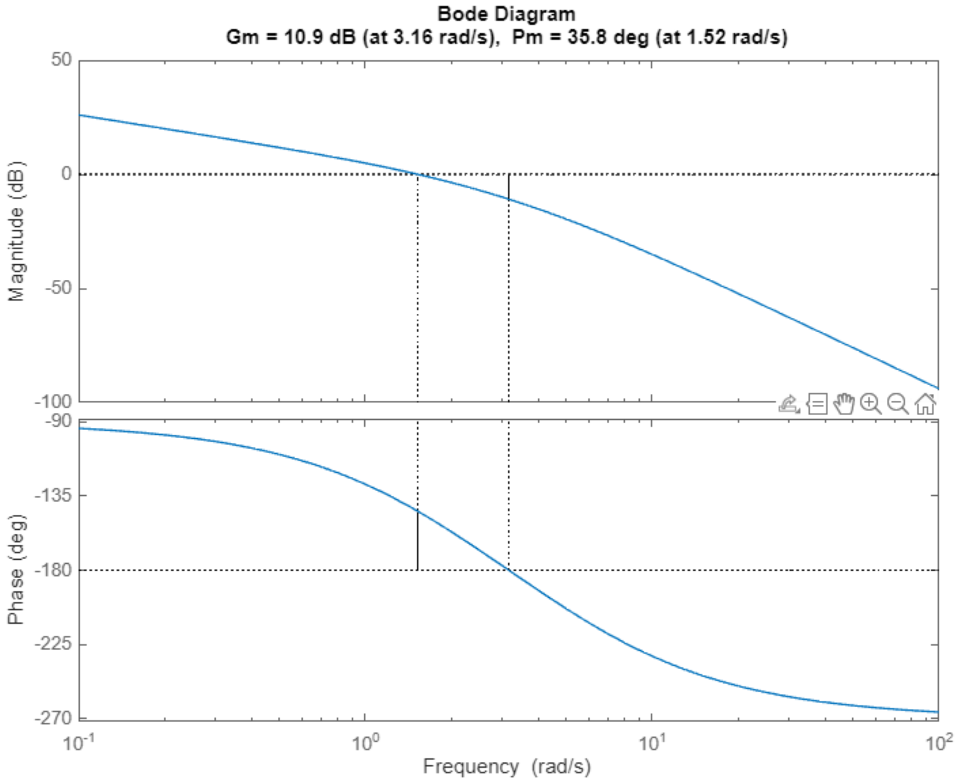
PROGRAM:

```
clc;
clear all;
close all;
w = linspace(0,1000,1000);
for i=1:1:1000
    mag(i) = 20/ (w(i)* (sqrt(4+(w(i)*w(i)))) *(sqrt(25+(w(i)*w(i)))));
    magdb(i) = 20*log(mag(i));
    phase(i) = ((-atan(w(i)/5)-atan(w(i)/2))*180/pi) -90 ;

end
gh = tf(20,[1 7 10 0])

[pm,gm,wcg,wcp] = margin(gh)
margin(gh)

subplot(2,1,1)
semilogx(w,magdb)
subplot(2,1,2)
semilogx(w,phase)
```

COMMAND WINDOW:

```
gh =

           20
  ------------------
  s^3 + 7 s^2 + 10 s

Continuous-time transfer function.
Model Properties

pm =

    3.5000


gm =

   35.7873


wcg =
    3.1623
wcp =
    1.5224
```

FIGURE:


Bode Diagram
Gm = 10.9 dB (at 3.16 rad/s), Pm = 35.8 deg (at 1.52 rad/s)
Magnitude (dB)
50
0
-50
-100
Phase (deg)
-90
-135
-180
-225
-270
10-1
100
101
102
Frequency (rad/s)

Experiment – 10

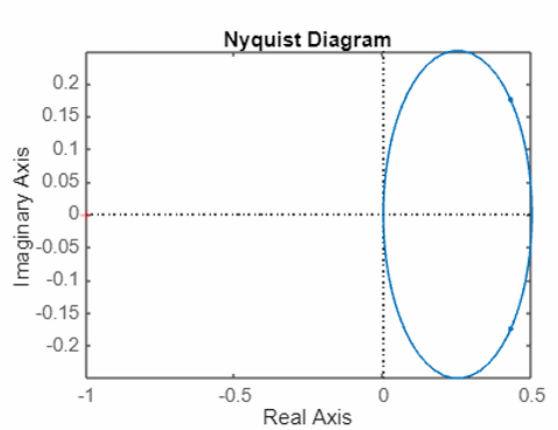
OBJECTIVE: Plot Nyquist plot for given transfer function. Also determine the relative stability by measuring gain and phase margin.

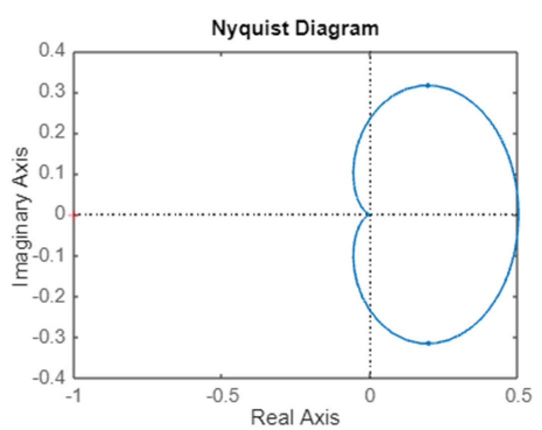
SOFTWARE REQUIRED: MATLAB

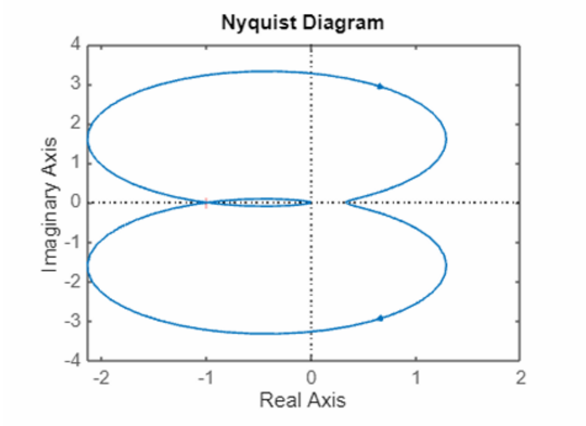
PROGRAM:

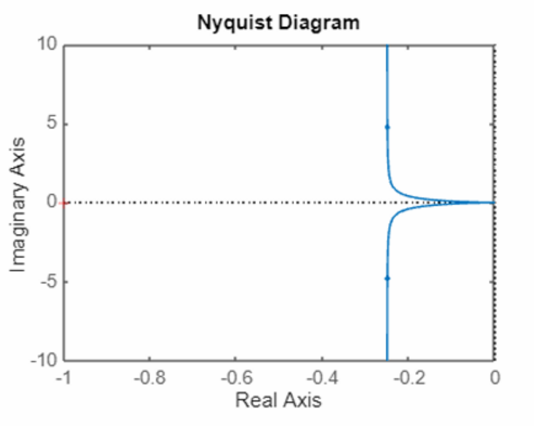
```
gshs1 = tf([2],[1 4]) % order 1
figure
subplot(2,2,1)
nyquist(gshs1);
gshs2 = tf([4], [1 6 8]) %order 2
subplot(2,2,2)
nyquist(gshs2);
gshs3 = tf([2],[1 4 2 6]) % order 3
subplot(2,2,3)
nyquist(gshs3);
 gshs4 = tf([4],[1 4 0]) % order 2 type 1
[R,I,w] = nyquist(gshs4);
Mag = abs(R+I*i);
Phi = angle(R+I*i);
for j=1:1:length(R)
        mag1(j) = Mag(j);
        phi1(j) = Phi(j);
end
subplot(2,2,4)
nyquist(gshs4)
figure
polar(phi1,mag1)
```

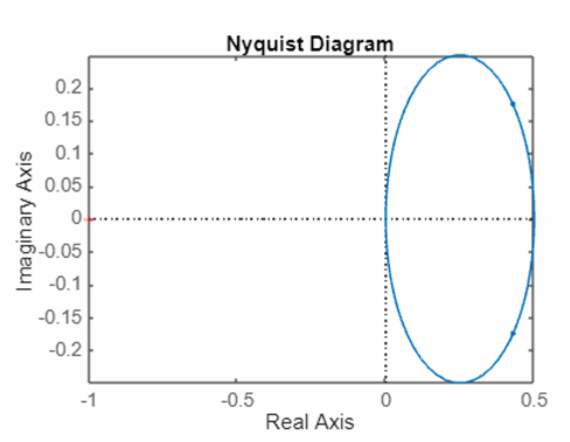
FIGURE:

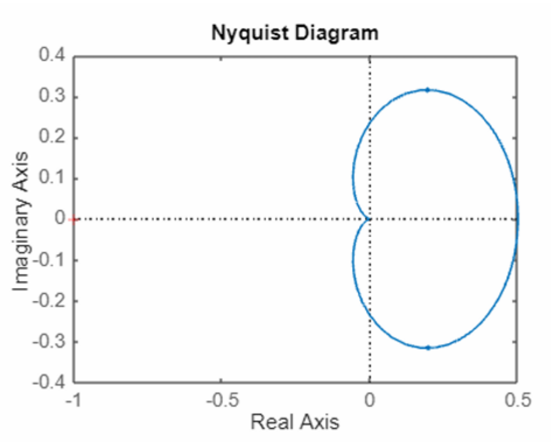





Nyquist Diagram
4
3
2
1
0
-1
-2
-3
-4
Imaginary Axis
-2
-1
0
1
2
Real Axis


Nyquist Diagram
10
5
0
-5
-10
Imaginary Axis
-1
-0.8
-0.6
-0.4
-0.2
0
Real Axis


Nyquist Diagram
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
Imaginary Axis
-1
-0.5
0
0.5
Real Axis


Nyquist Diagram
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
Imaginary Axis
-1
-0.5
0
0.5
Real Axis