**CONTROL SYSYTEM LAB MANUAL**

**Submitted by**

**VASU SINGH (2100290310166)**

## Under the Supervision of

**Dr.Himanshu Chaudhary**

**Dr.Niraj Singh Mehta**



**Department of Electronics and Communication Engineering**

**KIET Group of Institutions, Delhi-NCR, Meerut Road (NH-58) Ghaziabad - 201206**

**(2023-24)**

**TABLE OF CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Title** | **Pg.No.** | **Sign.** |
| **1.** | **Introduction to MatLab control system tool box** | **3-5** |  |
| **2.** | **Determine transpose, inverse of given matrix** | **6-7** |  |
| **3.** | **Plot Pole zero config. in s-plane for the given transfer func. using MatLab** | **8-9** |  |
| **4.** | **Determine the transfer func. for the closed loop system in the block diag. representation** | **10-11** |  |
| **5.** | **Determine the time response of the given system subjected to any arbitrary input** | **12-13** |  |
|  | **Plot unit step response of the given transfer func. and find delayed time , peak time ,peak overshoot and settling time.** | **14-15** |  |
| **7.** | **Determine the steady state error of given transfer func.** | **16-17** |  |
| **8.** | **Plot root locus of a given transfer func.,locate closed loop pole for the diff. value of k.** | **18-19** |  |
| **9.** | **Plot root locus of a given transfer func.,locate open loop pole for the diff. value of k.** | **20-21** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**EXPERIMENT – 1**

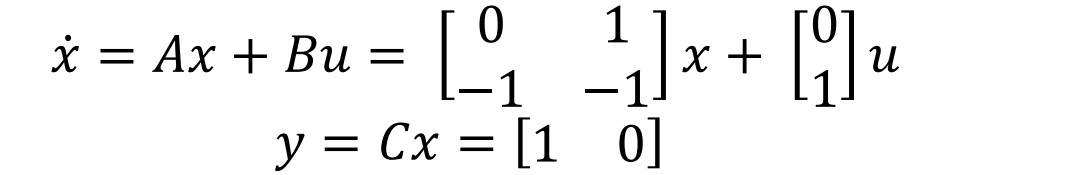
**OBJECTIVE:** Introduction to MATLAB Control System Toolbox.

**SOFTWARE REQUIRED:** MATLAB

**THEORY:**

**1.1 Introduction to Control System**

Control System Toolbox is a package for Matlab consisting of tools specifically developed forcontrol applications. The package offers data structures to describe common system representations such as state space models and transfer functions, as well as tools for analysis and design of control systems. There are also tools for simulation of systems.Here we will get to know the basic commands of Control System Toolbox. After completion of this experiment you should be able to understand and use Control Systems Toolbox to create and analyze linear systems. Extensive use of the Matlab help command is recommended. Student is recommended to create a script file (e.g. myscript.m) in which commands can be written and by running a script file instead of typing the commands directly at the Matlab prompt, it is quite easier to correct mistakes, and also, the work will be saved for lateruse.The system we will use, of the form



To enter a matrix in MATLAB, for example matrix A, do

𝑦 = 𝐶𝑥 = [1 0]

To enter a matrix in MATLAB, for example matrix A, do

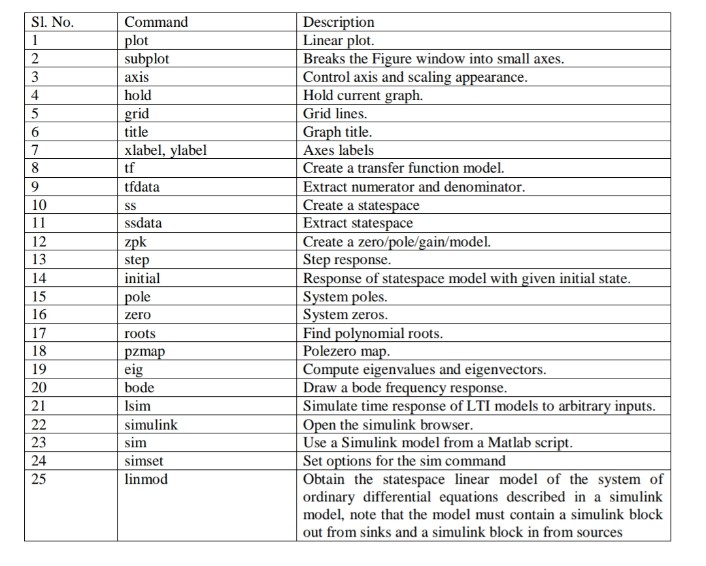
𝐴 = [0 1; −1 −1]

***Creation and Conversion of systems***

Control System Toolbox supports several system representations of linear time invariant systems. In this exercise, we will use two of the most common representations; state space models and transfer functions.Create a state space description of the system using ss, and name it systems. Find out how to use ss by using the help function (help ss). At this stage, you should have obtained a state space description of the system.Let us now create an equivalent transfer function model of the system above. This could, as you know, be done by using the formula 𝐺(𝑠) = 𝐶(𝑠𝐼 − 𝐴)−1 𝐵 + 𝐷. However, Matlab may also be used for the task. Use the command tf to convert the state space model to a transfer function and name it.

**1.2 Introduction to Simulink**

Simulink is a simulation program based upon Matlab. There are several ways to define a model. One can work graphically and connect block diagrams with predefined blocks. Alternatively one can give the mathematical description in forms of differential equations in an mfile (the format for programs written in the Matlab programming language). Matlab/Simulink supports both these representations as well as combinations. Furthermore one can use descriptions that include a hierarchy of connected subsystems.

**Some useful Matlab commands:**

**Using Simulink Models in Matlab Scripts**

Often, it is convenient to work with Matlab scripts (mfiles), in order to save a sequence of Commands. It is possible to use Simulink models from within a Matlab script, using the Command sim. By using the command simset options for the sim command may be specified.Use the model from the previous example. Save the model, and name it “mymodel.mdl”. CreateA Matlab script named “mysim.m”, and enter the following commands:Tfinal = 300;Options = simset(’reltol’,1e-5,’refine’,10,’solver’,’ode45’);

Sim(’mymodel’,tfinal,options)

;%plot resultsFigure(1)

Clf

Subplot(211)

Plot(t,u);

Ylabel(’u’)

Subplot(212)

Plot(t,y)

Ylabel(’y’)

When you run the script, you should see a plot showing the input and the output of the transfer Function. Use the help command to learn more about how to use the simset and sim commands.

**Experiment** **– 2**

**OBJECTIVE: Determine transpose, inverse value of given matrix.**

**SOFTWARE REQUIRED:** MATLAB

**PROGRAM:**

TRANSPOSE OF MATRIX

clc;

clear all;

close all;

A = [2 3 1; 4 5 1; 2 3 1]

B = transpose(A)

INVERSE OF MATRIX

clc;

clear all;

A = [1 2; 3 4]

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))

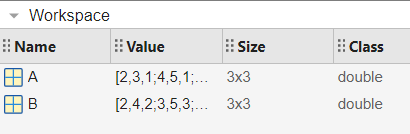
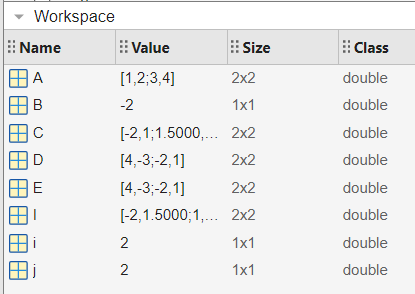
I (i, j) = (E (i, j)/B);

C = I'

end

end

**WORKSPACE:**

****

**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);

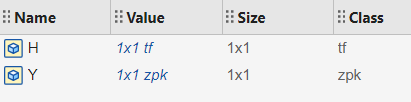
pzmap (H);

grid on;

subplot (2,1,2);

pzmap(Y);

**WORKSPACE:**



**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

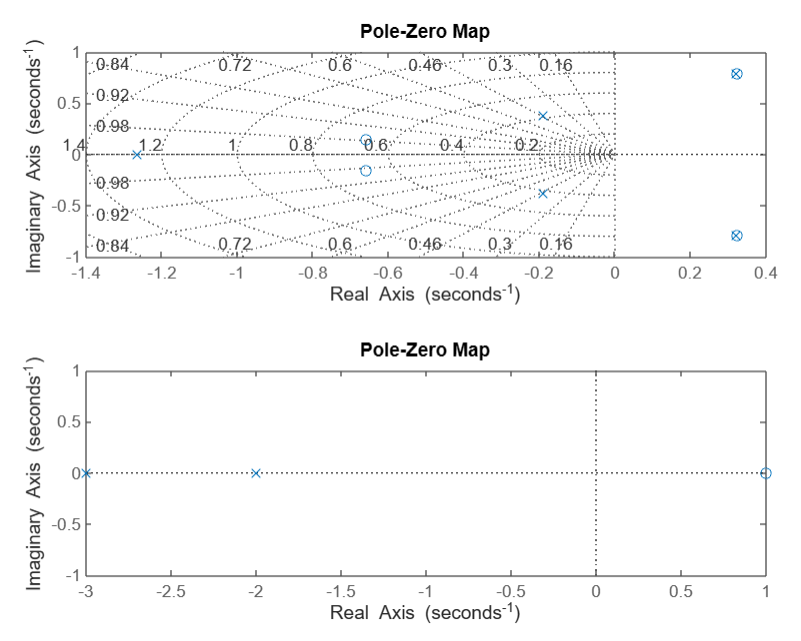
Continuous-time transfer function.

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)

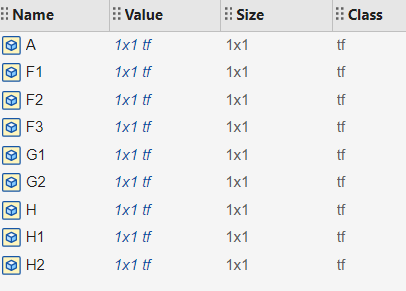
A = series (F1, F2)

H = tf ([1 0], [1 3])

F3 = feedback (A, H)

f =F3(step);

**WORKSPACE:**

****

**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

-----------------------------------------------

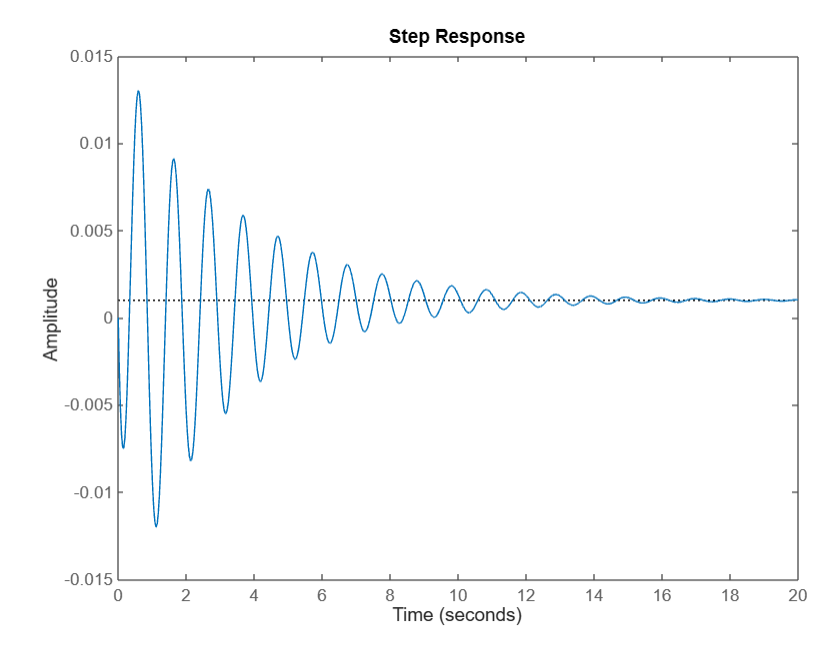
s^6 + 12 s^5 + 53 s^4 + 101 s^3 + 76 s^2 + 12 s

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);

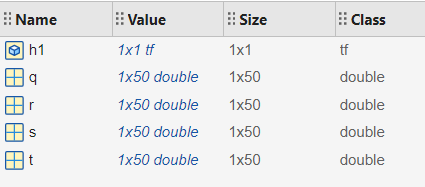
subplot (3,2,4);

lsim (h1, s, t);

subplot (3,2,5);

lsim (h1, q, t);

**WINDOWSPACE:**

****

**COMMAND WINDOW:**

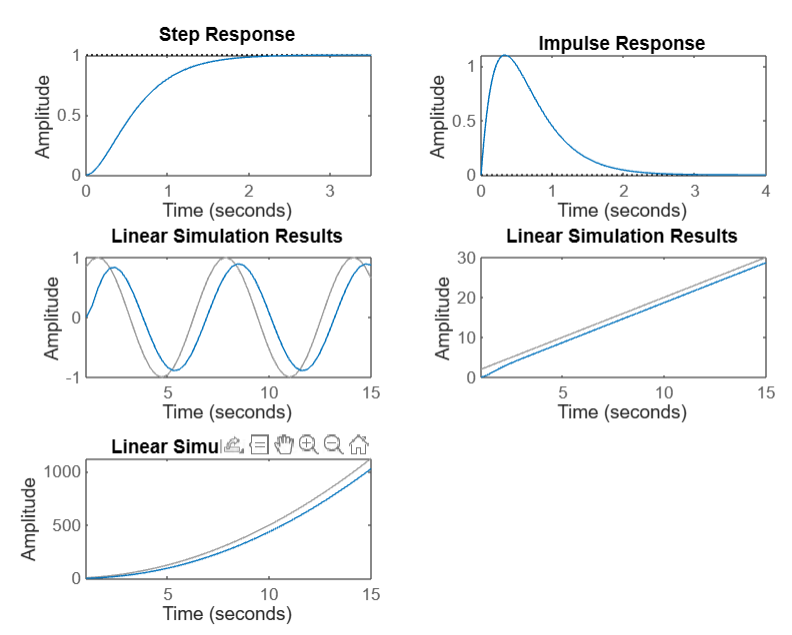
h1 = 9

-------------

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);

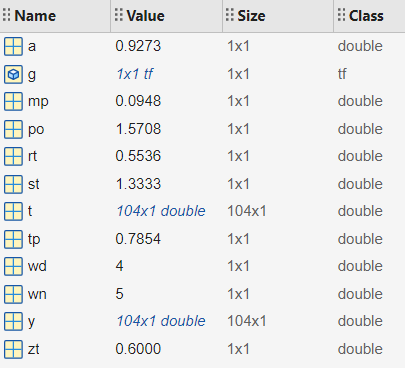
rt = (pi-a)/wd

t = 0:0.01:10;

[y, t] = step(g);

Plot (t, y)

**WINDOWSPACE:**

****

**COMMAND WINDOW:**

g = 25

--------------

s^2 + 6 s + 25

Continuous-time transfer function.

wn = 5

zt = 0.6000

wd = 4

tp = 0.7854

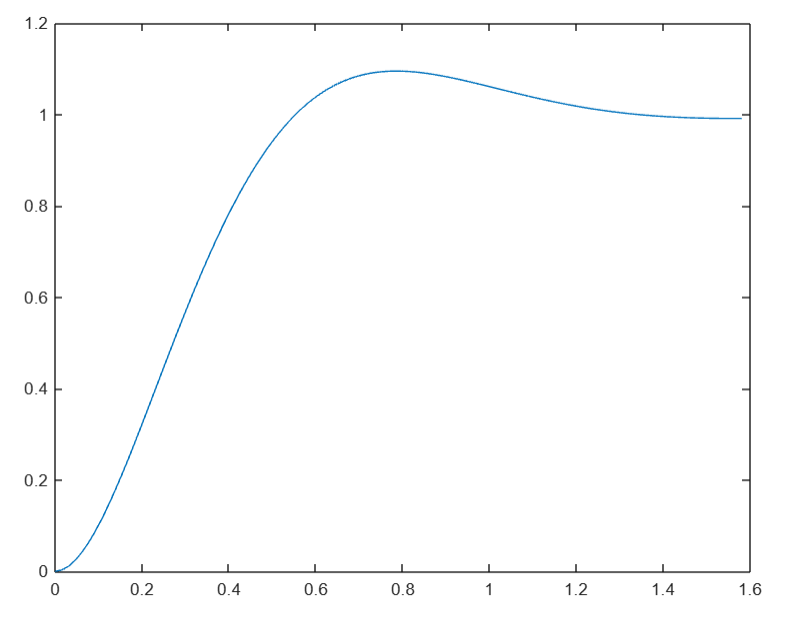
mp = 0.0948

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);

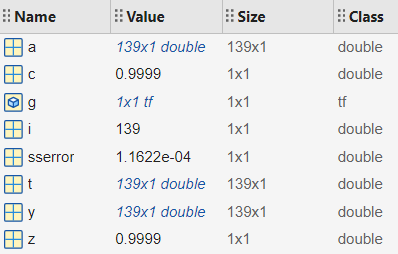
i=length(y)

z = y(i)

c = y(end)

sserror = abs(1-c)

**WINDOWSPACE:**

****

**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:**

%root locus plot

clc

clear all

close all

num = 1;

p = conv ([1 0], [1 4]);

den = conv (p, [1 4 13]);

G = tf (num, den);

k = linspace (1,1000,1000);

for i =1:1000:1000

h = k(i)\*G;

m = feedback(h,1);

pzmap(m)

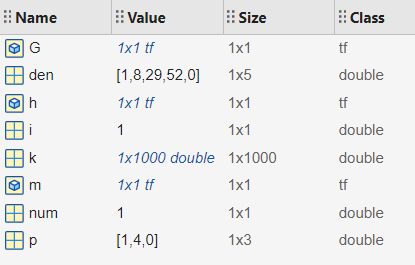
hold on

end

m = feedback(G,1)

rlocus (m)

**WINDOWSPACE:**

****

**COMMAND WINDOW:**

num = 1

p = 1 4 0

den = 1 8 29 52 0

k =

Columns 1 through 11

1 2 3 4 5 6 7 8 9 10 11

Columns 12 through 22

12 13 14 15 16 17 18 19 20 21 22

Columns 23 through 33

23 24 25 26 27 28 29 30 31 32 33

…

Columns 991 through 1,000

991 992 993 994 995 996 997 998 999 1000

m = 1

-------------------------------

s^4 + 8 s^3 + 29 s^2 + 52 s + 1

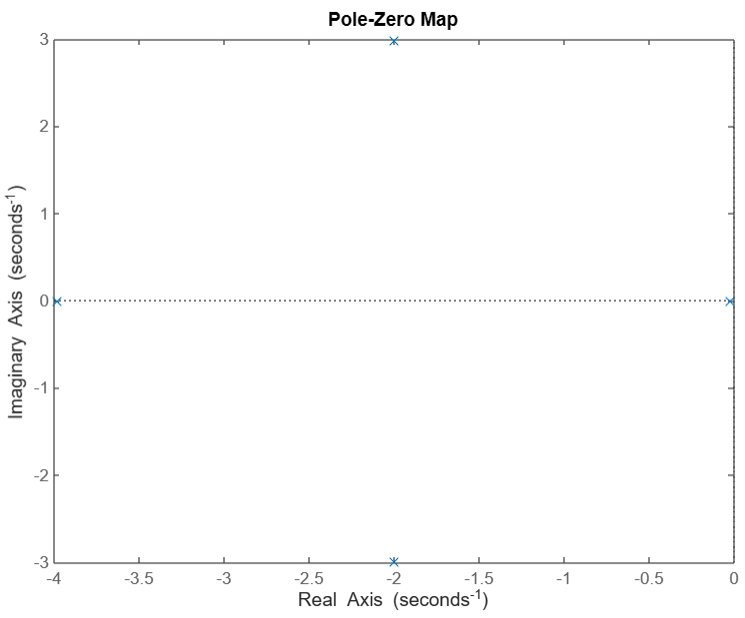
Continuous-time transfer function.

m = 1

-------------------------------

s^4 + 8 s^3 + 29 s^2 + 52 s + 1

Continuous-time transfer function.

**FIGURE: **

**Experiment** **– 9**

**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)

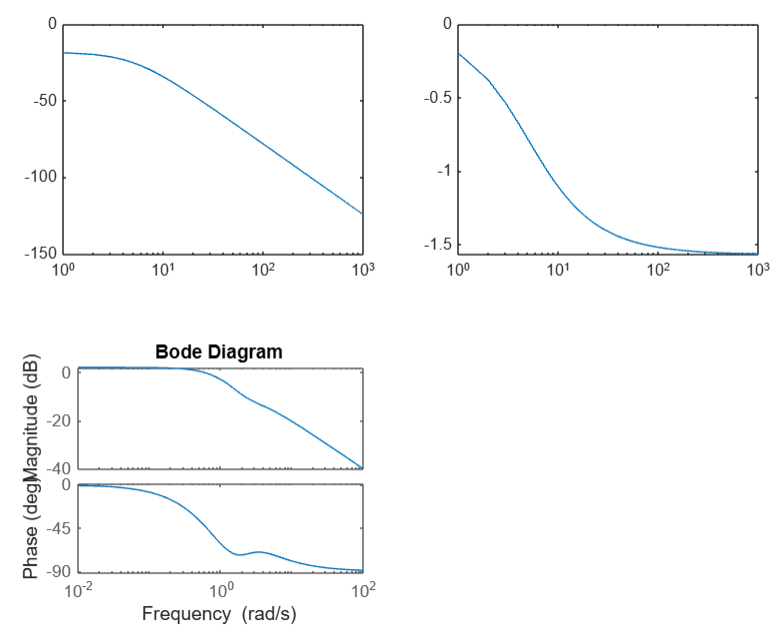
subplot(2,2,3)

bodeplot(g)

**COMMAND WINDOW:**

g =  
   
 s^2 + 3 s + 5  
 ---------------------  
 s^3 + 5 s^2 + 8 s + 4  
   
Continuous-time transfer function.

**FIGURE:**

****