DEPARTMENT OF ECE / EEE

SAVEETHA SCHOOL OF ENGINEERING

Engineer to Excel

CONTROL SYSTEM

LABORATORY RECORD

submitted by

NAME OF THE STUDENT (Reg No)

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

SAVEETHA

INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

(Declared as Deemed to be University under Section 3 of UGC Act 1956)

CHENNAI - 602105

2024

<Attach Bonafide Page>

CONTROL SYSTEMS LABORATORY

List of experiments

1. TRANSFER FUNCTION OF PARALLEL SYSTEM, CASCADE SYSTEM AND FEEDBACK SYSTEM
2. TRANSFER FUNCTION FROM POLES AND ZEROS & ZEROS AND POLES FROM TRANSFER FUNCTION
3. STEP AND IMPULSE RESPONSE OF A FIRST ORDER SYSTEM
4. TRANSIENT RESPONSE OF A SECOND ORDER SYSTEM
5. TIME RESPONSE PARAMETERS FOR A GIVEN SYSTEM
6. STEP AND RAMP RESPONSE OF DIFFERENT CONTROLLERS
7. P, PI & PID CONTROLLER USING PROCESS CONTROL SIMULATOR
8. BODE PLOT FOR A TRANSFER FUNCTION
9. POLAR PLOT FOR A TRANSFER FUNCTION
10. ROOT LOCUS FOR A TRANSFER FUNCTION
11. NYQUIST PLOT FOR A TRANSFER FUNCTION
12. FIND STABILITY OF A SYSTEM USING ROUTH HURWITZ CRITERION
13. TRANSFER FUNCTION FROM STATE MODEL AND STATE MODEL FROM TRANSFER FUNCTION
14. IMPLEMENTATION OF THE BLOCK DIAGRAM AND STATE MODEL REDUCTION TECHNIQUE
15. TRANSFER FUNCTION TO CONTROLLABLE, OBSERVABLE AND JORDON CANONICAL FORMS
16. STATE AND OUPUT CONTROLLABILITY AND OBSERVABILITY (USING KALMANS TEST)

******

INDEX

EXP. NO.	***DATE OF EXP.***	***EXPERIMENT NAME***	***PAGE NO.***	***MARKS OUT OF 100***	***FACULTY INITIAL***
14		IMPLEMENTATION OF THE BLOCK DIAGRAM AND STATE MODEL REDUCTION TECHNIQUE			
15		TRANSFER FUNCTION TO CONTROLLABLE, OBSERVABLE AND JORDON CANONICAL FORMS			
16		STATE AND OUPUT CONTROLLABILITY AND OBSERVABILITY (USING KALMANS TEST)			

TOTAL MARKS SCORED (OUT OF 1600)	
AVERAGE MARKS (OUT OF 100)	
CONVERSION (TO 20 MARKS)	

FACULTY SIGNATURE

Ex. No. 1	TRANSFER FUNCTION OF THE SERIES PARALLEL AND FEEDBACK CONTROL SYSTEM
Exp. Date:	

Aim:

To obtain the transfer function of the series, parallel and feedback control system for the given open loop transfer functions

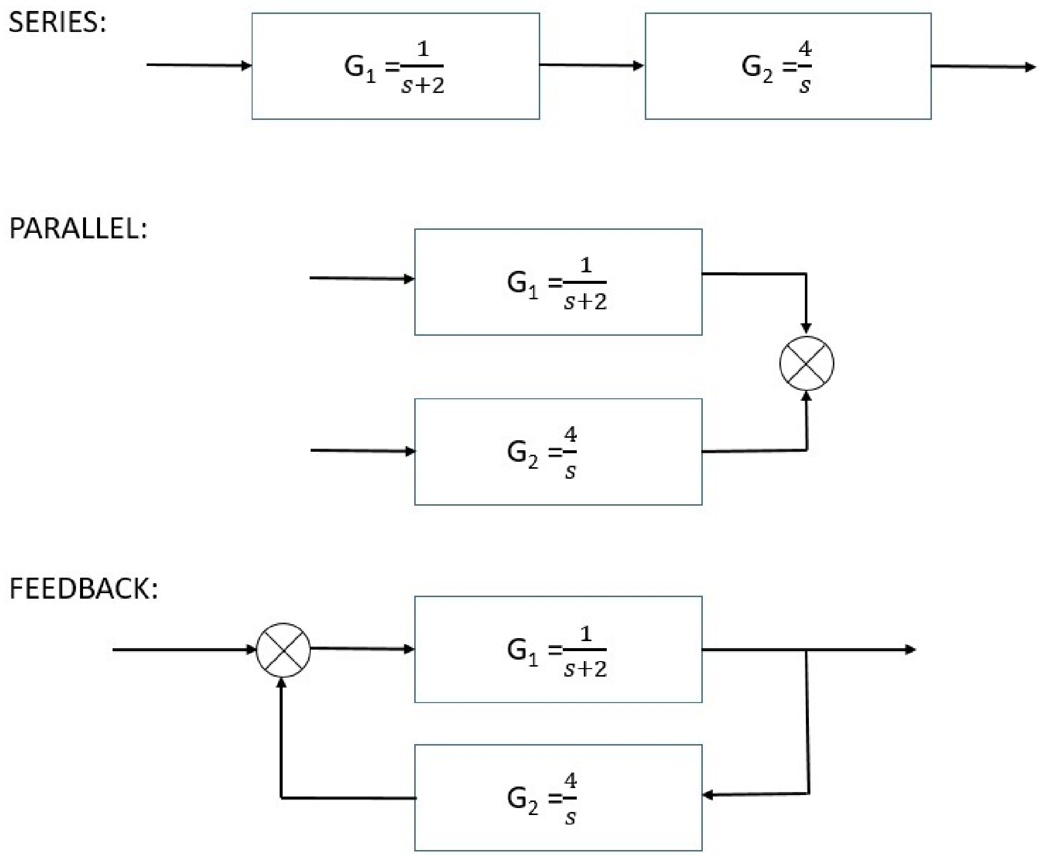
$$G_1 = \frac{1}{s+2} \text{ and } G_2 = \frac{4}{s}$$

Tools Used:

(i) SCILAB software

(ii) PC

Block diagrams:



Procedure:

Step 1: Get the numerator and denominator of the transfer function G_1.

Step 2: Get the numerator and denominator of the transfer function G_2.

Step 3: Multiply G_1 and G_2 to get the series transfer function.

$$T(s) = G_1(s)G_2(s)$$

Step 4: Add G_1 and G_2 to get the parallel transfer function.

$$T(s) = G_1(s) + G_2(s)$$

Step 5: Using the formula, $T(s) = \frac{G_1}{1 \pm G_1 G_2}$, the transfer function of the feedback control system is obtained.

Step 6: Display all the output.

Program:

```
clc;
clear;
s=%s;
G1=syslin('c',1,s+2);//Get the numerator and denominator of the transfer function G1
G2=syslin('c',4,s);//Get the numerator and denominator of the transfer function G2

T1=G1*G2;//series or cascade system
disp (T1, "The series transfer function of T1 = ");

T2=G1+G2;//parallel system
disp (T2, "The parallel transfer function of T2 = ");

T3=G1/(1+G1*G2);//positive feedback system
disp (T3, "The positive feedback transfer function of T3 = ");

T4=G1/(1-G1*G2);//negative feedback system
disp (T4, "The negative feedback transfer function of T4 = ");
```

Executed output:

Scilab 6.0.1 Console

```
 The series transfer function of T1 =


     4
   -------
         2
   2s + s

 The parallel transfer function of T2 =


   8 + 5s
   -------
         2
   2s + s

 The positive feedback transfer function of T3 =


       s
   -----------
             2
   4 + 2s + s

 The negative feedback transfer function of T4 =


        s
   ------------
              2
   -4 + 2s + s

-->
```

Exp. No. 1 practice examples:

Obtain the transfer function of the series, parallel, positive and negative feedback control system for the following open loop transfer functions

(i) $G_1 = \frac{1}{s^2+6s+9}$ and $G_2 = \frac{s+2}{s-3}$

(ii) $G_1 = \frac{1}{s^3+2s^2+6s+9}$ and $G_2 = \frac{s}{s+3}$

(iii) $G_1 = \frac{1}{s^3+2s^2+6s+9}$; $G_2 = \frac{s+2}{s-3}$ (consider G_1 and G_2 are in series) and $G_3 = \frac{s}{s+3}$ (G_3 acts as a feedback system function)

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Results:

Ex. No. 2	TRANSFER FUNCTION FROM POLES AND ZEROS & ZEROS AND POLES FROM TRANSFER FUNCTION
Exp. Date:	

Aim:

To obtain (i) the transfer function from the given poles and zeros, (ii) the zeros and poles from the given transfer functions and (iii) to plot the zeros and poles for the given transfer function.

Tools Used:

(i) SCILAB software

(ii) PC

Program code:

(i) Transfer function from the given poles and zeros

```
clc; clear;
Z=[2;1];P=[0;-3-%i;-3+%i];K=2.5;
disp(Z,'Enter the zeros Z = ')
disp(P,'Enter the poles P = ')
disp(K,'Enter the system gain K = ')
S=zp2tf(Z,P,K,"c")
roots(S.num)
disp(S, 'The obtained transfer function S =')
```

Simulation results:

(i) Transfer function from the given poles and zeros

Scilab 6.0.2 Console

```
 Enter the zeros Z =


   2.
   1.

 Enter the poles P =


   0.
  -3. - i
  -3. + i

 Enter the system gain K =


   2.5

 The obtained transfer function S =


                   2
   5 - 7.5s + 2.5s
   ----------------
             2   3
    10s + 6s + s

-->
```

(ii) The zeros and poles from the given transfer functions.

```
clc
clear
H=syslin("d",[45.76+6*%s+2*%s^2],[23.28+3.1*%s+%s^2+3.796*%s^3])
disp(H, 'The given transfer function H =' )
[z,p,k]=tf2zp(H)
disp(z, 'The obtained zeros for the given transfer function z = ');
disp(p, 'The obtained poles for the given transfer function p = ');
disp(k, 'The obtained system gain for the given transfer function k = ');
```

Simulation result:

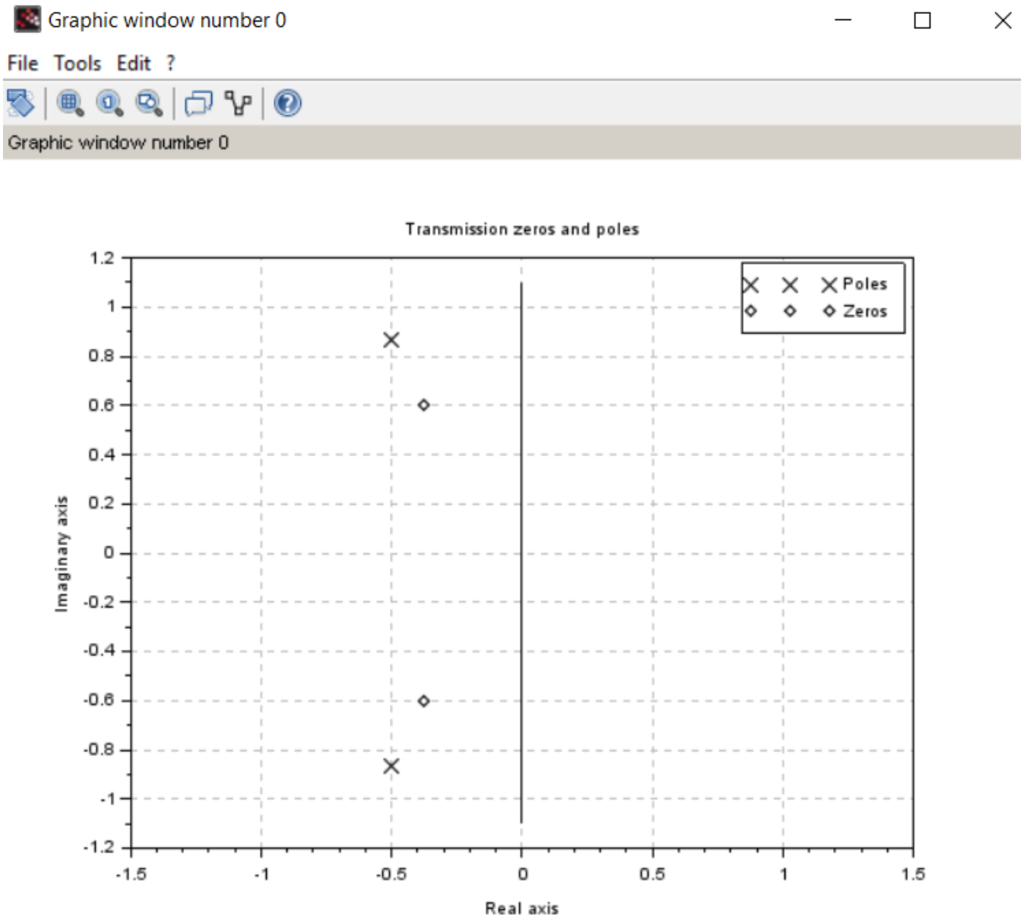
Scilab 6.0.2 Console

```
 The given transfer function H =

                       2
        45.76 + 6s + 2s
   --------------------------
                   2        3
   23.28 + 3.1s + s + 3.796s

 The obtained zeros for the given transfer function z =

  -1.5 + 4.542026i
  -1.5 - 4.542026i

 The obtained poles for the given transfer function p =

  -1.7664798
   0.7515223 + 1.7049813i
   0.7515223 - 1.7049813i

 The obtained system gain for the given transfer function k =

   0.5268704

-->
```

(iii) Plot zeros and poles in the s plane for the given transfer function

```
clc
clear
s=%s;
n=[2+3*s+4*s^2]; //input numerator of transfer function
d=[1+s+s^2]; //input denominator of transfer function
h=syslin("c",n,d);
plzr(h);
```

Simulation results:



Exp. No. 2 practice examples:

(i) Obtain the transfer function from the following given poles, zeros and system function

(a) Z=[-1;1]; P=[0;-1+i;-1-i]; K=1.5
(b) Z=[1;3]; P=[0;2;-0.5+i;-0.5-i]; K=1
(c) Z=[0;1]; P=[0;1;10;5+10i;5-10i]; K=0.5

(ii) Obtain the zeros and poles from the given transfer functions

(a) num=$45+6s+s^2$, den=$20+3s+s^2+4s^3$
(b) num=$1+s$, den=$s+3s^2+3s^3+s^4$

(iii) To plot the zeros and poles for the given transfer function.

(a) num=$45+6s+s^2$, den=$20+3s+s^2+4s^3$
(b) num=$1+s$, den=$s+3s^2+3s^3+s^4$

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result:

Ex. No. 3	STEP AND IMPULSE RESPONSE OF A FIRST ORDER SYSTEM
Exp. Date:	

Aim:

To obtain the step and the impulse response for the given first order system

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

Tools Used:

(i) SCILAB 6.0.2. software

(ii) PC

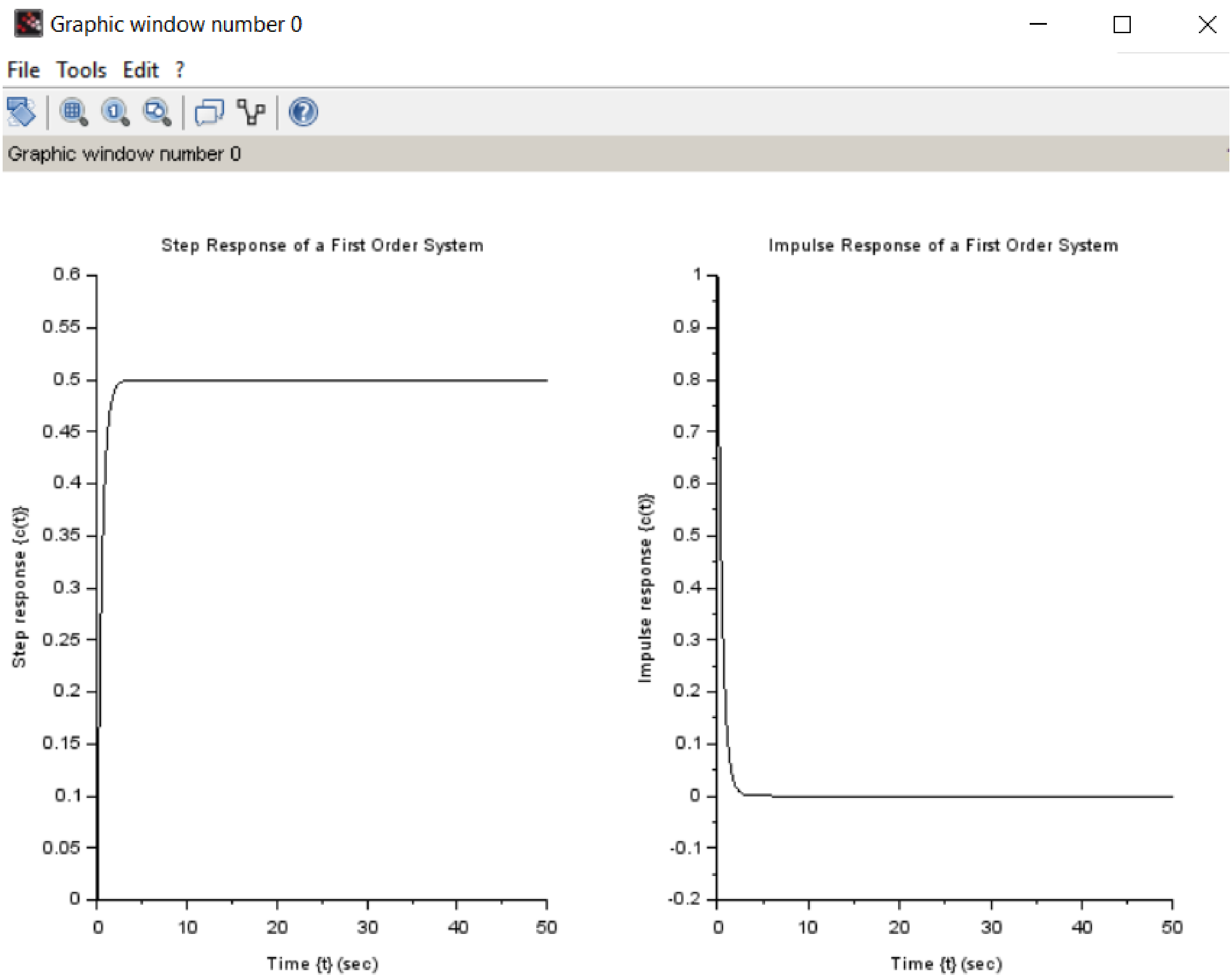
Program:

```
clc;
num=poly([1],'s','coeff');
den=poly([2 1],'s','coeff');
g=syslin('c',num/den);
disp(g, 'The given first order system function G =');

t=0:0.05:50;

gs=csim('step',t,g);
subplot(121)
plot2d(t,gs)
xlabel('Time {t} (sec)')
ylabel('Step response {c(t)}')
title('Step Response of a First Order System')

gi=csim('impulse',t,g);
subplot(122)
plot2d(t,gi)
xlabel('Time {t} (sec)')
ylabel('Impulse response {c(t)}')
title('Impulse Response of a First Order System')
```

Console display after execution of the program:

```
Scilab 6.0.2 Console

 The given first order system function G =


    1
   ------


   2 + s

-->
```

Simulation result:



Exp. No. 3 practice examples:

Obtain the step and the impulse response of a first order system

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

(b) $G = \frac{10}{s+10}$

(c) $G = \frac{0.25}{0.5\,s+0.75}$

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

(e) $G = \frac{s+1}{s+10}$

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 4	TRANSIENT RESPONSE OF A SECOND ORDER SYSTEM
Exp. Date:	

Aim:

To obtain the transient response of a second order system whose transfer function is

$$T(s) = \frac{36}{s^2 + 2\zeta s + 36}$$

when the damping ratio (i) ζ = 0, (ii) ζ = 0.5, (iii) ζ= 1 and (iv) ζ = 10

Tools Used:

(i) SCILAB 6.0.2. software

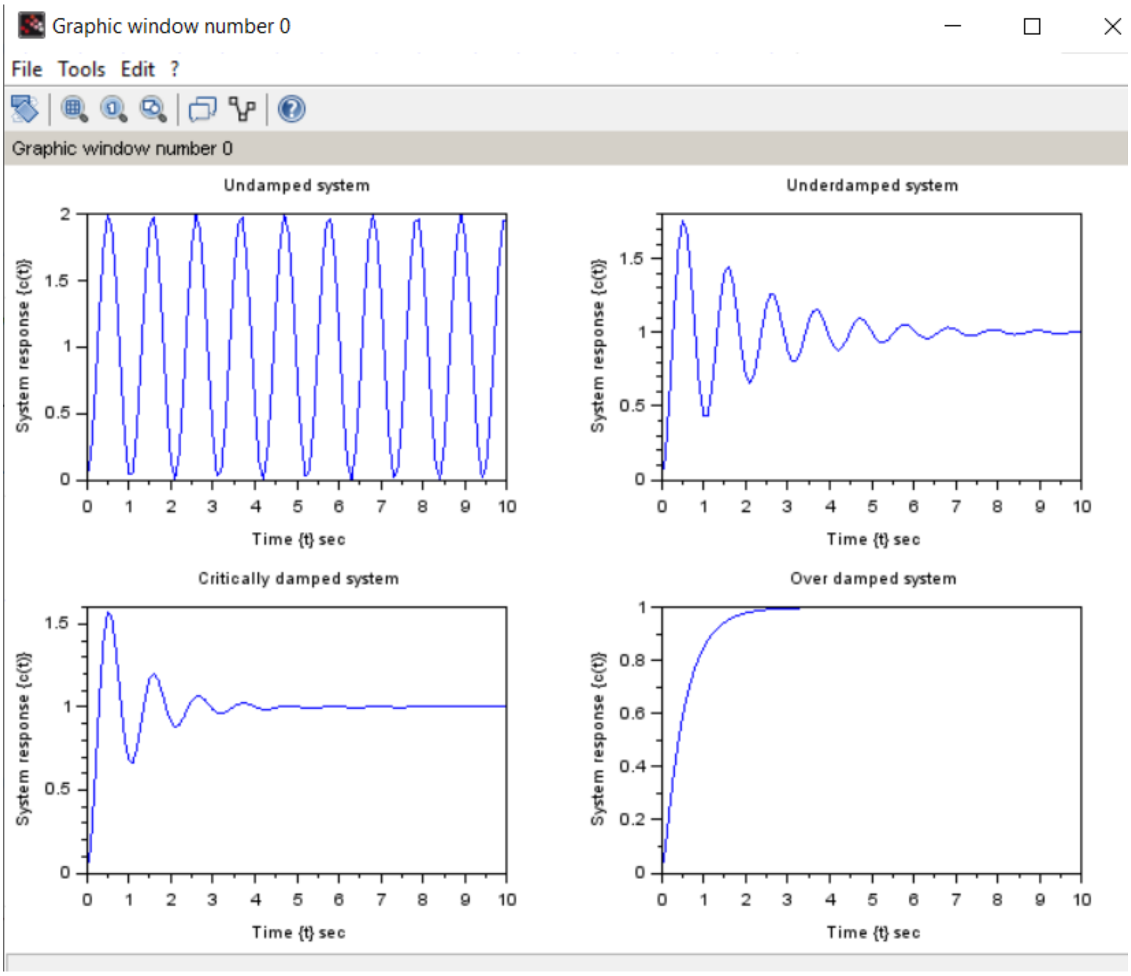
(ii) PC

Program:

```
clc;
s=%s;
z=0;//Undamped system
num=36; den=36+2*z*s+s^2;
TF=syslin('c',num,den);
subplot(221)
t=0:0.1:10;
y1=csim('step',t,TF);
plot(t,y1)
xlabel('Time {t} sec')
ylabel('System response {c(t)}')
title('Undamped system');

z=0.5;//Underdamped system
num=36; den=36+2*z*s+s^2;
TF=syslin('c',num,den);
subplot(222)
t=0:0.1:10;
y2=csim('step',t,TF);
plot(t,y2)
xlabel('Time {t} sec')
ylabel('System response {c(t)}')
title('Underdamped system');
```

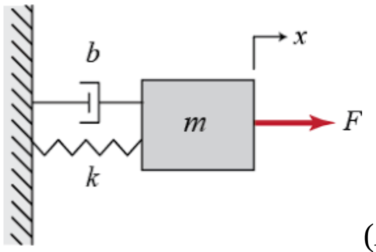
```
z=1;//Critically damped system
num=36; den=36+2*z*s+s^2;
TF=syslin('c',num,den);
subplot(223)
t=0:0.1:10;
y3=csim('step',t,TF);
xlabel('Time {t} sec')
ylabel('System response {c(t)}')
plot(t,y1)
title('Critically damped system');

z=10;//Over damped system
num=36; den=36+2*z*s+s^2;
TF=syslin('c',num,den);
subplot(224)
t=0:0.1:10;
y4=csim('step',t,TF);
plot(t,y4)
xlabel('Time {t} sec')
ylabel('System response {c(t)}')
title('Over damped system');
```

Simulation result:



Exp. No. 4 practice examples:

(a) Obtain the transient step responses for various damping ratio (i) ζ = 0, (ii) ζ = 0.5, (iii) ζ= 1 and (iv) ζ = 10 of a simple mechanical system shown in figure below.



(Assume M = 1 Kg, K=1 Nm and B=1 Nm/s)

(b) Obtain the transient response of a second order system whose transfer function is

$$T(s) = \frac{100}{s^2 + 2\zeta\omega_n s + 100}$$

when the damping ratio (i) ζ = 0, (ii) ζ = 0.5, (iii) ζ= 1 and (iv) ζ = 100.

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result:

Ex. No. 5	TIME RESPONSE PARAMETERS FOR A GIVEN SYSTEM
Exp. Date:	

Aim:

To obtain the various time response parameters for the given system.

Mp=0.2;//Maximum peak overshoot, tp=1;//peak time, J=1;*//kg.m^2*, B=1;*//N−/rad/sec*
Eq=(s*pi)^2-log(1/Mp)^2*(1-s^2);

Tools Used:

(i) SCILAB 6.0.2. software

(ii) PC

Program:

```
clear;clc;
xdel(winsid());
mode(0);

Mp=0.2;
tp=1;
J=1;//kg.m^2
B=1;//N−/rad/sec

s=%s;
pi=%pi;
Eq=(s*pi)^2-log(1/Mp)^2*(1-s^2);
disp(Eq, 'The obtained equation')

x=roots(Eq);
disp(x,'x=')

zeta=abs(x(1));
disp(zeta,'zeta=')

wd=pi/tp;
disp(wd,'damped frequency wd=')

wn=wd/sqrt(1-zeta^2);
disp(wn,'natural frequency wn=')
```

```
K=J*wn^2;
disp(K, 'K=')

Kh=(2*sqrt(K*J)*zeta-B)/K;
disp(Kh,'Kh=')

sigma=wn*zeta;
disp(sigma, 'sigma=')

_beta=atan(wd/sigma);
disp(_beta,'Beta =')

tr=(pi-_beta)/wd;
disp(tr,'Rise time tr=')

ts_2percent=4/sigma;
disp(ts_2percent, 'Settling time for 2% error=')

ts_5percent=3/sigma;
disp(ts_5percent, 'Settling time for 5% error=')
```

Simulation result:

The obtained equation

-2.5902904 +12.459895s^2

x=

0.4559498
-0.4559498

zeta=

0.4559498

damped frequency wd=

3.1415927

natural frequency wn=

3.5298576

K=

12.459895

Kh=

0.1780814

sigma=

1.6094379

Beta =

1.0973572

Rise time tr=

0.6507004

Settling time for 2% error=

2.4853397

Settling time for 5% error=

1.8640048

Exp. No. 5 practice example:

Obtain the various time response parameters for the given system.

```
Mp=0.25;
tp=1.6;
J=1;//kg.m^2
B=1;//N−/rad/sec
Eq=(s*pi)^2-log(1/Mp)^2*(1-s^2);
```

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 6	STEP AND RAMP RESPONSE OF DIFFERENT CONTROLLERS
Exp. Date:	

Aim:

To obtain step and ramp response of the different controllers such as (i) proportional controller, (ii) integral controller, (iii) proportional integral controller, (iv) proportional derivative controller and (v) proportional plus integral plus derivative controller for the given values:

Proportional gain, K_p=4, integral gain, K_i=2, differential time, T_d=0.8 and integral time, T_i=2

Tools Used:

(i) SCILAB 6.0.2 software

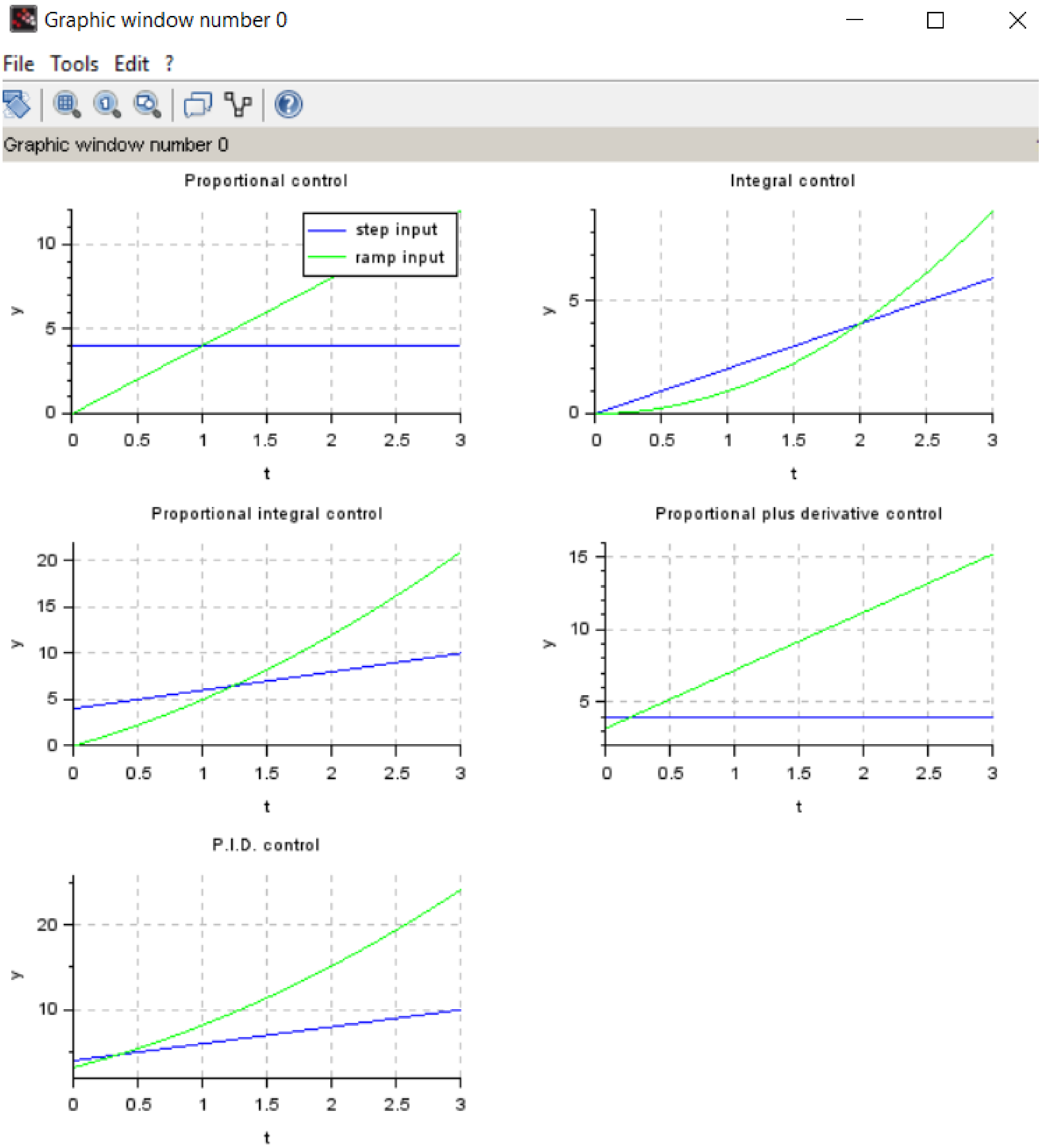
(ii) PC

Program:

```
clc;
clear;
xdel(winsid());

Kp=4;//proportional gain
Ki1=2;//integral gain
Td=0.8;//differential time
Ti=2;//integral time
Ki2=Kp/Ti;

s=%s;
Gi=syslin('c',Ki1/s);
t=0:0.05:3;
ramp=t;
subplot(321);
p1=Kp*ones(1,length(t));
p2=Kp*t;
plot2d(t,p1,style=2);
plot2d(t,p2,style=3);
xtitle('Proportional control','t','y');
legend('step input','ramp input');
xgrid(color('gray'));

subplot(322);
```

```
i1=csim("step",t,Gi);
i2=csim(ramp,t,Gi);
plot2d(t,i1,style=2);
plot2d(t,i2,style=3);
xtitle('Integral control','t','y');
xgrid(color('gray'));
i1=i1*Ki2/Ki1;//change of gain
i2=i2*Ki2/Ki1;

subplot(323);
plot2d(t,p1+i1,style=2);
plot2d(t,p2+i2,style=3);
xtitle('Proportional integral control','t','y');
xgrid(color('gray'));

subplot(324);
pd1=p1;
pd2=p2+Kp*Td*ones(1,length(t));//derivative term
plot2d(t,pd1,style=2);
plot2d(t,pd2,style=3);
xtitle('Proportional plus derivative control','t','y');
xgrid(color('gray'));

subplot(325);
plot2d(t,pd1+i1,style=2);
plot2d(t,pd2+i2,style=3);
xtitle('P.I.D. control','t','y');
xgrid(color('gray'));
```

Simulation result:



Exp. No. 6 practice examples:

Obtain step and ramp response of different controllers (i) proportional controller, (ii) integral controller, (iii) proportional integral controller, (iv) proportional derivative controller and (v) proportional plus integral plus derivative controller for the given values:

Proportional gain, K_p=100, integral gain, K_i=1, differential time, T_d=0.1 and integral time, T_i=10

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		**Faculty signature**
Date of record submission		

Result:

Ex. No. 7	P, PI & PID CONTROLLER USING PROCESS CONTROL SIMULATOR
Exp. Date:	

Aim:

To study the response of the P, PI and PID controller in a first order process for a unit feedback system by using process control simulator. (Assume K_p = 5, K_i = 0.5 and K_d = 2.5)

Tools Used:

(i) Process control simulator

(ii) Power cord

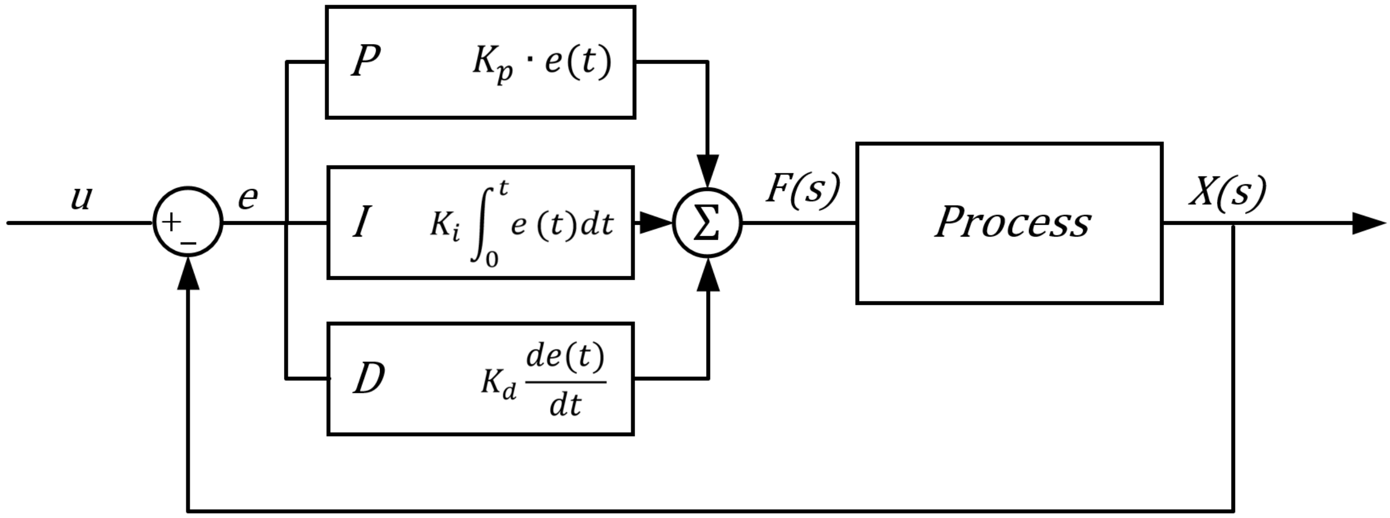
(iii) Patch cords

(iv) connecting wires if needed

Procedure:

1. Connections are made as per the schematic connection diagram.
2. The PID output is connected with the first order system and with unit feedback input.
3. Apply a DC input signal ranging from 0 to 5 V.
4. Now connect using patch cord for the input and set the desired input voltage Vin.
5. Now connect using patch cord for viewing the output.
6. Set the constants Kp, Ki and Kd values.
7. Set I and D ‘OFF’ for P controller, D ‘OFF’ for PI controller and all kept ON (P I and D) for PID
8. Observe the output voltage, Vout at time t = 10 sec and tabulate the readings of the output voltage.
9. Plot the graph between input voltage and output voltage.

Connection diagram:



Tabulation:

V_{in} (V)	V_{out} (V) at t=10 sec		
	P	**P+I**	**P+I+D**
0			
0.5			
1			
1.5			
2			
2.5			
3			
3.5			
4			
4.5			
5			

Exp. No. 7 practice examples:

Study the response of the PD, PI and PID controller in a first order process for a unit feedback system by using process control simulator. (Assume time t_1 = 0 sec, t_2=10 sec, t_3= 30 sec, K_p = 7.5, K_i = 0.3 and K_d = 3)

(Examples output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 8	BODE PLOT FOR A GIVEN TRANSFER FUNCTION
Exp. Date:	

Aim:

To sketch the bode plot for the given transfer function and determine the following (i) gain cross over frequency (ii) phase cross over frequencies, (iii) gain margin and (iv) phase margin.

$$G(s) = \frac{k}{s(1 + 0.5s)(1 + 0.1s)}$$

(Assume system gain K=10)

Tools Used:

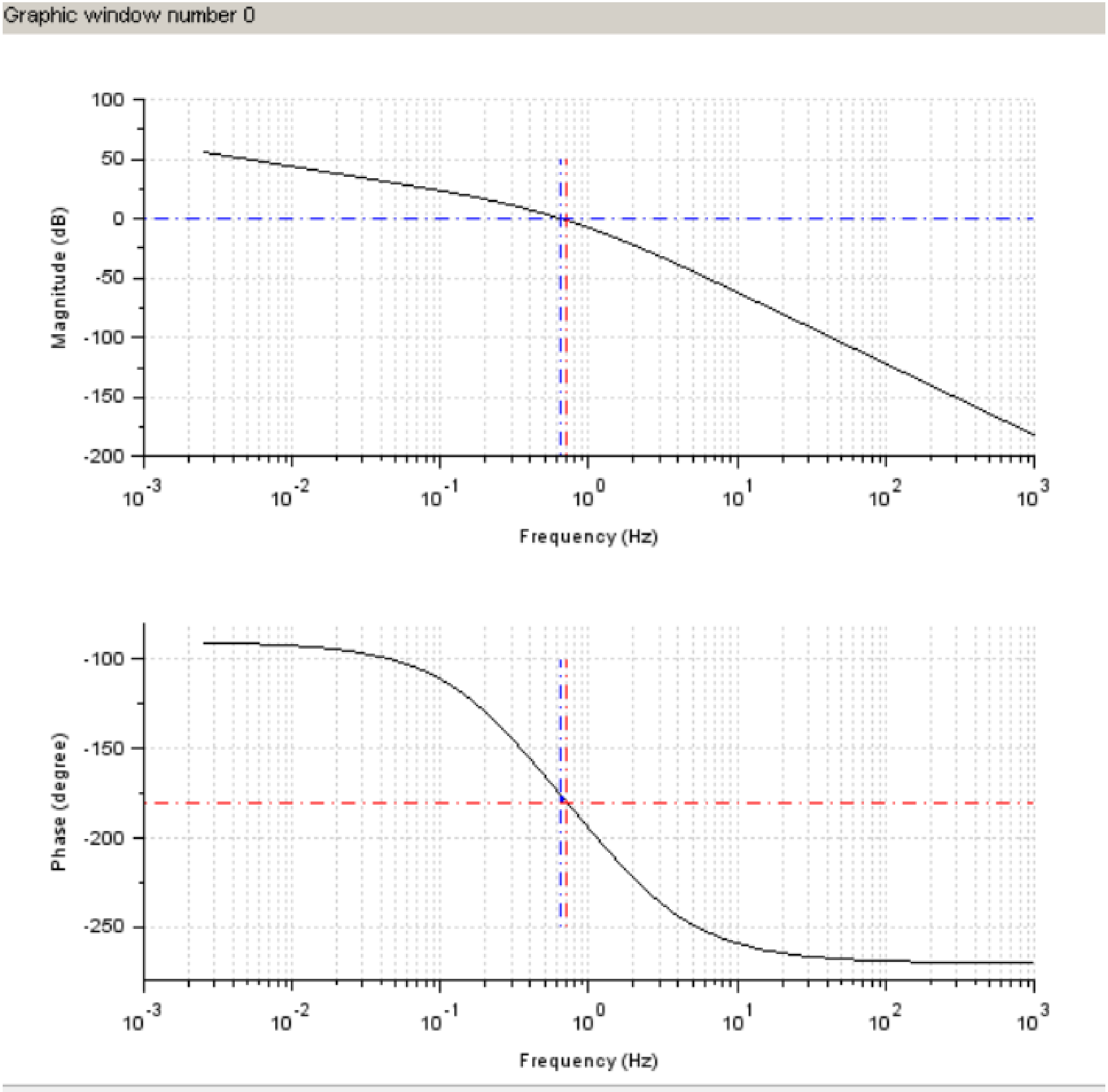
(i) SCILAB software

(ii) PC

Program:

```
clc;
clear;
k=10;
num=poly([1],"s","coeff");
den=poly([0 1 0.6 0.05],"s","coeff");
s1=syslin('c',num,den);
G=k*s1;
disp(G, 'The given transfer function G(s)=');
bode(G,0.01,1000);
xtitle('Bode plot of the given transfer function G(s)');
[g,frp]=g_margin(G);
[p,frg]=p_margin(G)
show_margins(G)
disp(frg,'Gain crossover frequency=',p,'Phase margin(degrees)=');
disp(frp,'Phase crossover frequency=',g,'Gain margin(dB)=');
```

Simulated output:


Graphic window number 0
100
50
0
-50
-100
-150
-200
Magnitude (dB)
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (Hz)
-100
-150
-200
-250
Phase (degree)
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (Hz)

Scilab 6.0.2 Console

```
 The given transfer function G(s)=


          10
   ----------------
           2       3
   s + 0.6s + 0.05s

 Phase margin(degrees)=

   3.9430653

 Gain crossover frequency=

   0.6489779

 Gain margin(dB)=

   1.5836249

 Phase crossover frequency=

   0.7117625

-->
```

Exp. No. 8 practice examples:

Sketch the bode plot for the given transfer function and determine the following (i) gain cross over frequency (ii) phase cross over frequencies, (iii) gain margin and (iv) phase margin.

(i) $G(s) = \frac{Ks^2}{(1+0.2s)(1+0.02s)}$ (Assume K=1)

(ii) $G(s) = \frac{K(9s^2+1.8s+9)}{(2s^3+1.8s^2+9s)}$ (Assume K=0.9)

(iii) G(s) = $\frac{K(1+0.2s)(1+0.025s)}{s^3(1+0.001s)(1+0.005s)}$ (Assume K=10)

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 9	POLAR PLOT FOR A TRANSFER FUNCTION
Exp. Date:	

Aim:

To sketch the polar plot for the given transfer function

$$G(s) = \frac{k}{s(1 + 0.5s)(1 + 0.1s)}$$

(Assume system gain K=10)

Tools Used:

(i) SCILAB software

(ii) PC

Program:

Function code:

```
//polar plot of a linear system
//repf=spolarplot(G,omega)
//G:linear system and omega: frquency in rad/s
//repf: complex frequency response
// save the function code program as spolarplot.sce

function repf=spolarplot(G, omega)
f=omega/2/%pi;
repf=repfreq(G,f);
r=abs(repf);
theta=atan(imag(repf),real(repf));
polarplot(theta,r,style=2);
endfunction
```

Main Program:

```
clear;
clc;
xdel(winsid()); //close all windows
//please edit the path
//cd"/<your code directory>/"
//exec("spolarplot.sce")
```

```
cd"C:\Users\admin\Documents"
exec("spolarplot.sce")
s=%s;
omega=logspace(-1,3,1000);
G=syslin('c',10,0.05*s^3+0.6*s^2+s);
disp(G,’The given transfer function G(s)=’);
spolarplot();
```

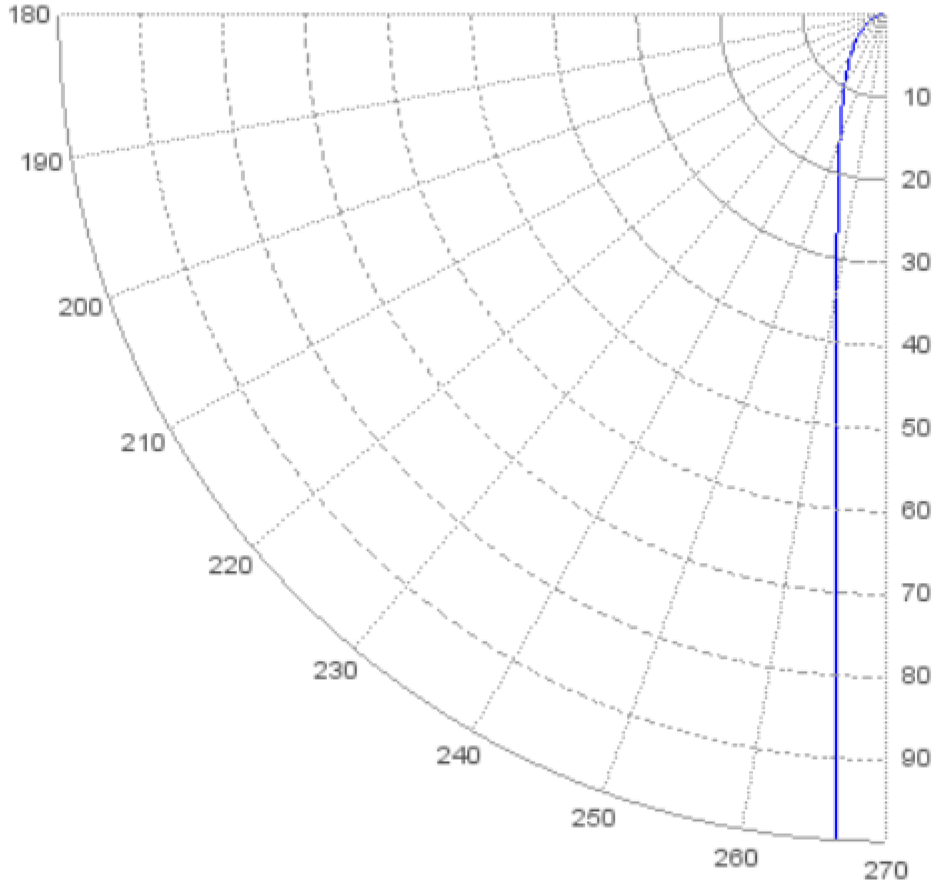
Executed output:

Console window:

Scilab 6.0.2 Console

```
--> //polar plot of a linear system

--> //repf=spolarplot(G,omega)

--> //G:linear system and omega: frquency in rad/s

--> //repf: complex frequency response

--> function repf=spolarplot(G,omega)
--> f=omega/2/%pi;
--> repf=repfreq(G,f);
--> r=abs(repf);
--> theta=atan(imag(repf),real(repf));
--> polarplot(theta,r,style=2);
--> endfunction

 The given transfer function is G(s)=


         10
   -----------------
          2       3
   s + 0.6s + 0.05s

-->
```

Graphic window number 0



Exp. No. 9 practice examples:

Sketch the polar plot for the given transfer function.

(i) $G(s) = \frac{Ks^2}{(1+0.2s)(1+0.02s)}$ (Assume K=1)

(ii) $G(s) = \frac{K(9s^2+1.8s+9)}{(2s^3+1.8s^2+9s)}$ (Assume K=0.9)

(iii) G(s) = $\frac{K(1+0.2s)(1+0.025s)}{s^3(1+0.001s)(1+0.005s)}$ (Assume K=10)

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 10	ROOT LOCUS FOR A TRANSFER FUNCTION
Exp. Date:	

Aim:

To sketch the root locus for the given transfer function

$$G(s) = \frac{1}{s(s+1)(s^2+4s+13)}$$

Tools Used:

(i) SCILAB software

(ii) PC

Program:

Function code:

```
function root1(G, box, text)
evans(G);
xgrid();
a=gca();
if box~=0 then
a.box="on";
a.data_bounds=box;
end
a.children(1).visible='off';//remove the legend block
xtitle(text);
endfunction//save the code as root1.sce
```

Main program:

```
//Root locus
clear;
clc;
xdel(winsid());

//cd"your directory"
cd"C:\Users\admin\Documents"
exec("root1.sce");
```

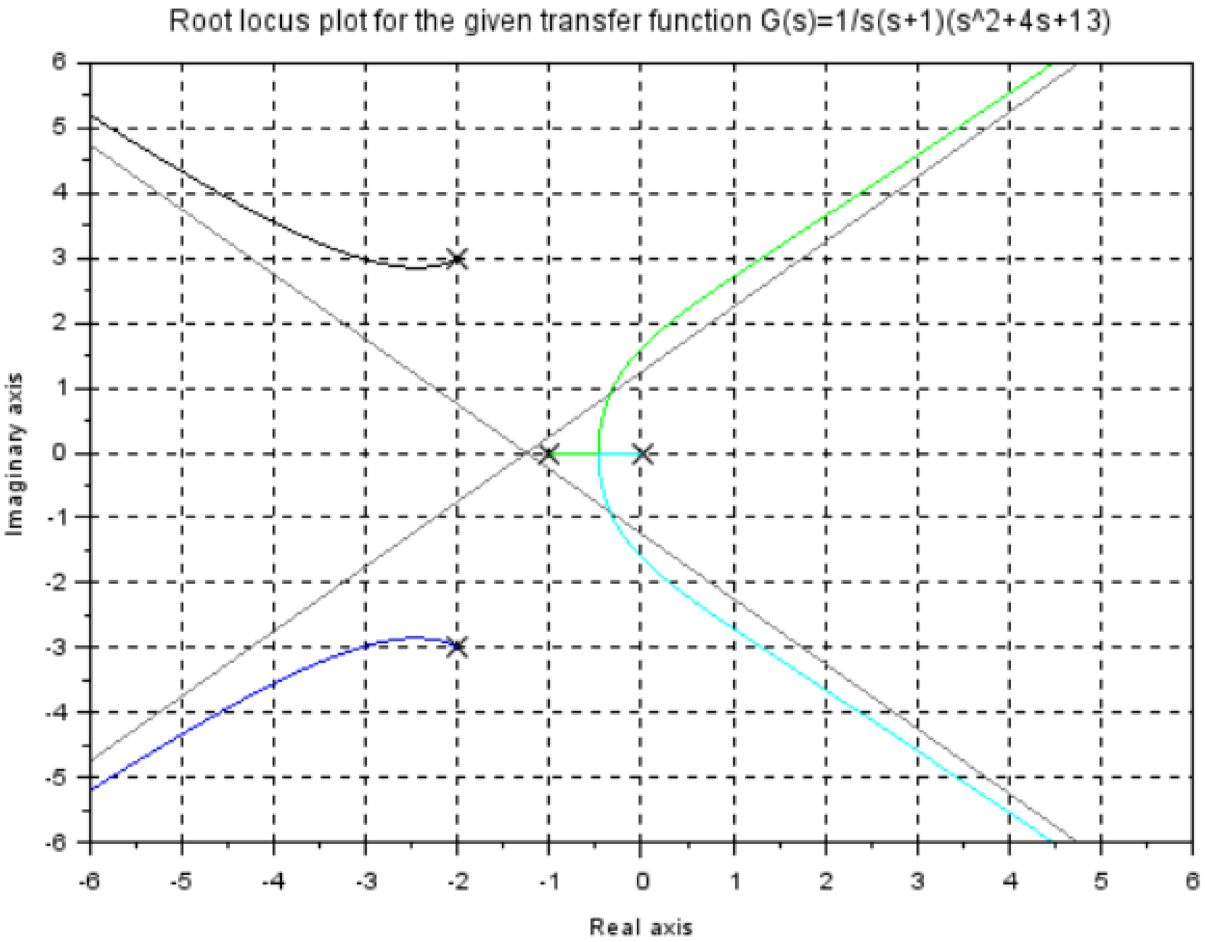
```
s=%s;
G=syslin('c',1,s*(s+1)*(s^2+4*s+13));
disp(G, 'The given transfer function G(s)=');
root1(G,[-6 -5;6 5],'Root locus plot for the given transfer function G(s)=1/s(s+1)(s^2+4s+13)');
//simply write the transfer function and choose
//suitable range [xmin ymin:xmax ymax]
```

Simulated result:

Scilab 6.0.2 Console

```
 The given transfer function G(s)=



           1
   --------------------
            2    3   4
   13s + 17s + 5s + s

-->
```

Graphic window number 0



Exp. No. 10 practice examples:

Sketch the root locus for the given transfer function

1. $G(s) = \frac{s+40}{s(s+20)(s^2+60s+100^2)}$
2. $G(s) = \frac{s+1}{s^2(s+3)(s+5)}$
3. $G(s) = \frac{1}{s(s+2)(s+5)}$
4. $G(s) = \frac{200}{(s+20)}$
5. $G(s) = \frac{1}{(s^2+10s+100)}$

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 11	NYQUIST PLOT FOR A TRANSFER FUNCTION
Exp. Date:	

Aim:

To sketch the Nyquist plot for the following transfer function

(a) $G(s) = \frac{20(s^2+s+0.5)}{s(s+1)(s+10)}$

(b) $G(s)H(s) = \frac{25+30s+5s^2}{168+206s+89s^2+16s^3+s^4}$

and

(c) $G(s) = \frac{K}{(T_1s+1)(T_2s+1)}$ (Assume K=1, T_1 = 5 sec and T_2=10 sec)

Tools Used:

(i) SCILAB software

(ii) PC

(a) Program code:

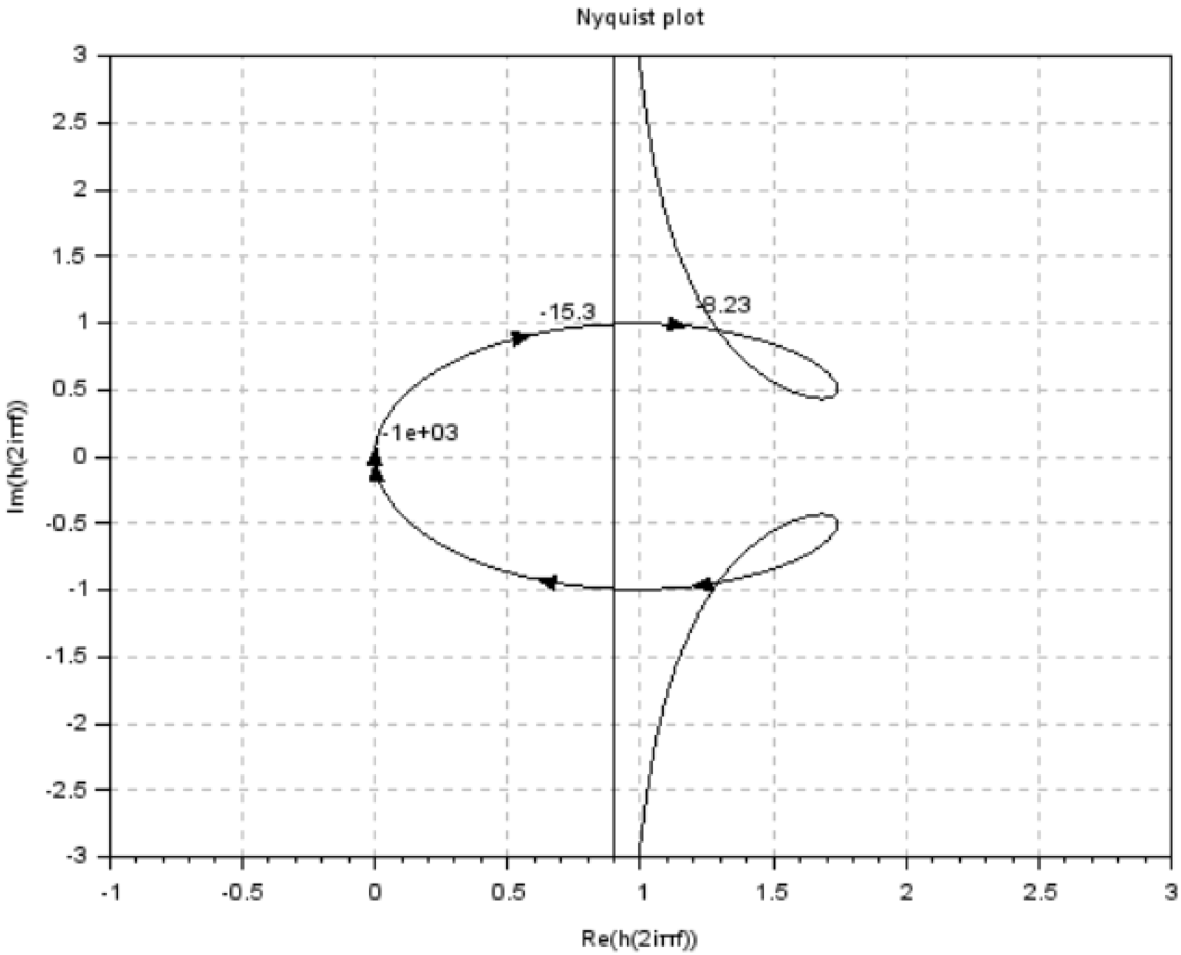
```
clear;
clc;
xdel(winsid()); //close all windows

s=%s/2/%pi;

G=syslin('c',20*(s^2+s+0.5),s*(s+1)*(s+10));
disp(G,'The given transfer function G(s)=')

a=gca();
a.clip_state='on';

nyquist(G,-1000,1000);

xgrid(color('gray'));
a.data_bounds=[-1 -3;3 3];
a.box='on';
```

Simulation Output:

Scilab 6.0.2 Console

```
 The given transfer function G(s)=

                  2
   10 + 20s + 20s
   ---------------
             2    3
   10s + 11s + s

-->
```

Graphic window number 0

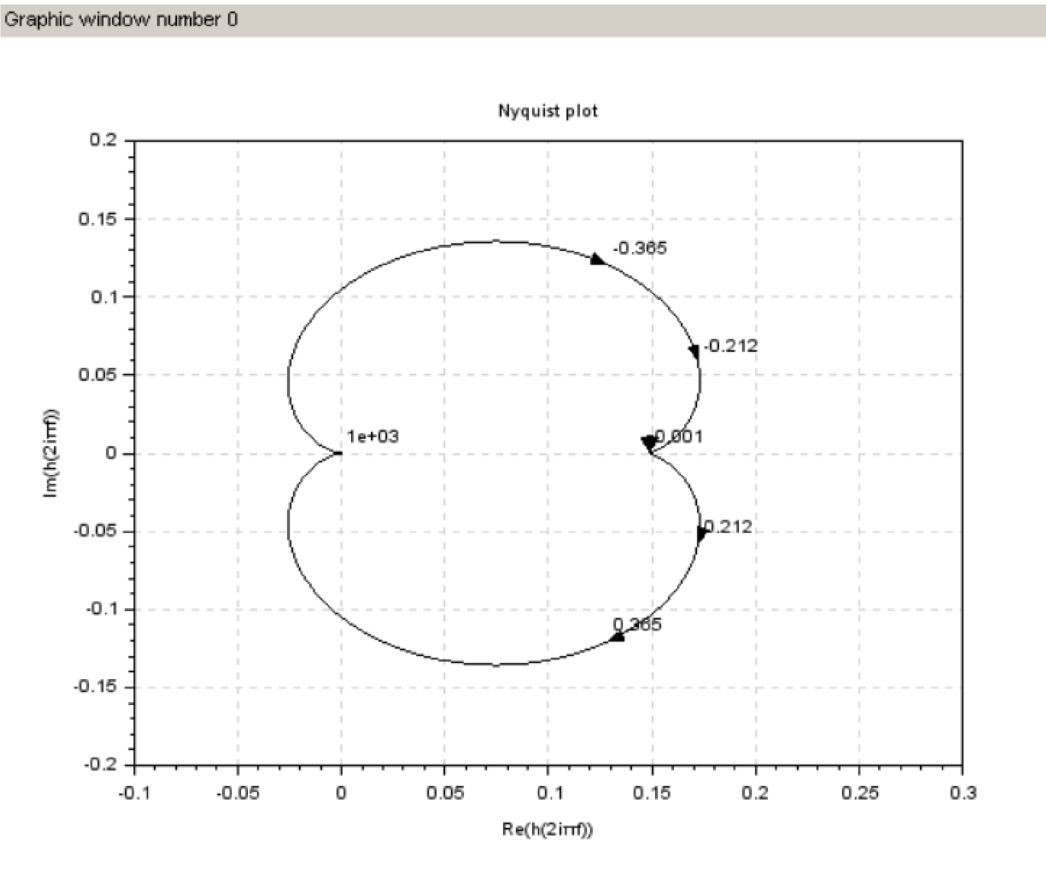


(b) Program code:

```
clear;
clc;
xdel(winsid()); //close all windows
s=%s;
T=syslin('c',25+30*s+5*s^2,168+206*s+89*s^2+16*s^3+s^4);
disp(T, 'The given loop transfer function G(s)H(s) = ');
nyquist(T)
a=gca();
a.clip_state='on';
a.data_bounds=[-0.1 -0.2;0.3 0.2];
a.box='on';
```

Simulation output:

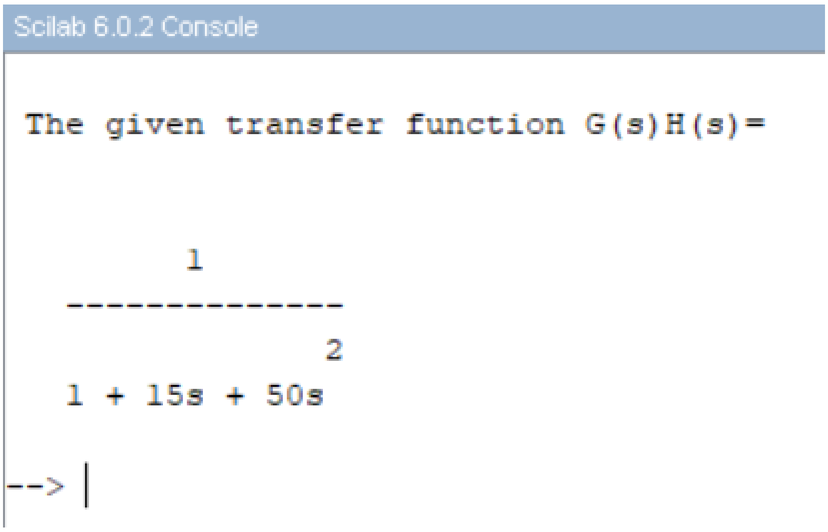
```
Scilab 6.0.2 Console

 The given loop transfer function G(s)H(s) =

                        2
        25 + 30s + 5s
   ---------------------------
                     2     3   4
   168 + 206s + 89s + 16s + s

--> |
```



(c) Program code:

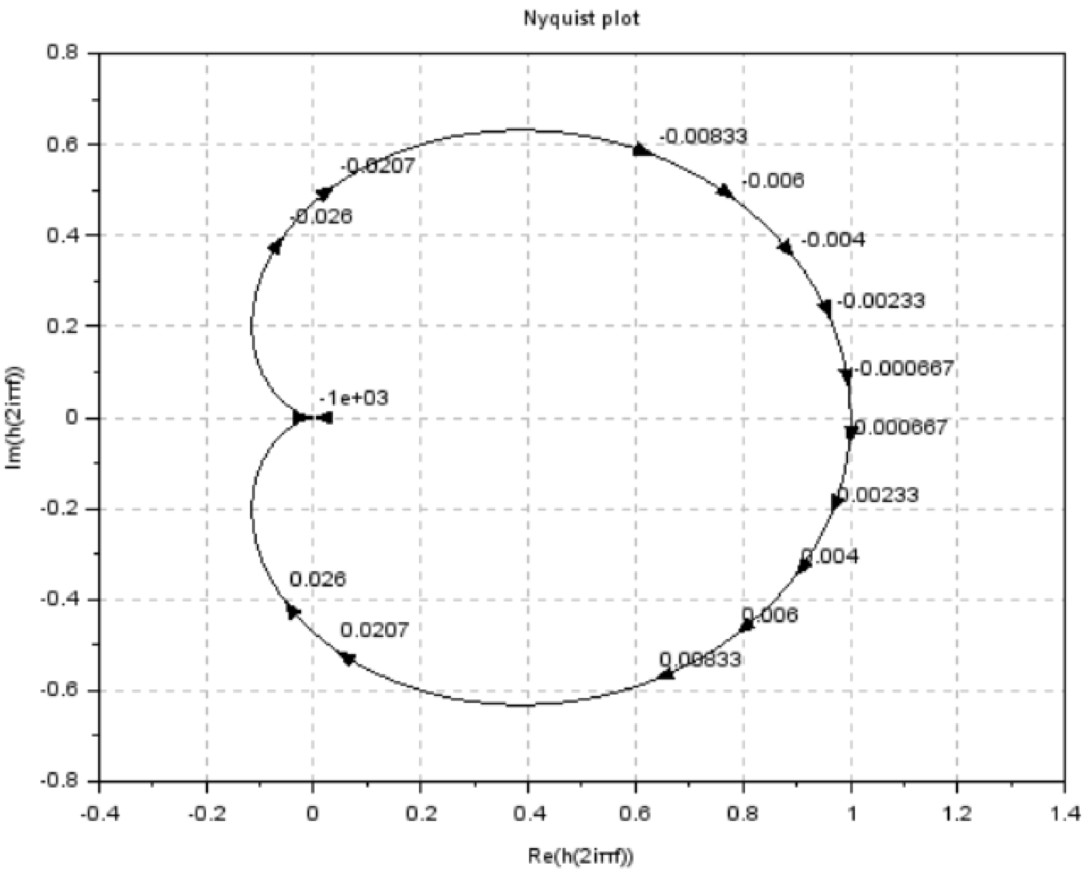
```
clear;
clc;
xdel(winsid()); //close all windows

s=%s;
T1=5; T2=10;
K=1;
den=(T1*s+1)*(T2*s+1);
GH=syslin('c',K,den); disp(GH, ‘The given transfer function G(s)H(s)=’)
nyquist(GH,-1000,1000);

xgrid(color('gray'));
a=gca();
a.clip_state='on';
a.data_bounds=[-0.3 -0.8;1.3 0.8];
a.box='on';
```

Simulation result:



```
Scilab 6.0.2 Console

 The given transfer function G(s)H(s)=

        1
   --------------
               2
   1 + 15s + 50s

--> 
```

Graphic window number 0



Exp. No. 11 practice examples:

Sketch the Nyquist plot for the given transfer function

1. $G(s) = \frac{s+40}{s(s+20)(s^2+60s+100^2)}$
2. $G(s) = \frac{s+1}{s^2(s+3)(s+5)}$
3. $G(s) = \frac{1}{s(s+2)(s+5)}$`
4. $G(s)H(s) = \frac{20s^2}{(s+2)\ (s+10)}$
5. $G(s) = \frac{1}{(s^2+10s+100)}$

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 12	FIND STABILITY OF A SYSTEM USING ROUTH HURWITZ CRITERION
Exp. Date:	

Aim:

To determine the stability of the closed-loop system using Routh Hurwitz Criterion for the given polynomial characteristics equations.

(i) $P(s) = s^4 + 2s^3 + 3s^2 + 4s + 5$ and

(ii) $P(s) = s^5 + 7s^4 + 6s^3 + 42s^2 + 8s + 56$

Tools Used:

(i) SCILAB 6.0.2 software

(ii) PC

Program code:

```
clear;
clc;
xdel(winsid());
mode(0);
s=%s;
H=s^4+2*s^3+3*s^2+4*s+5;
//H=s^5+7*s^4+6*s^3+42*s^2+8*s+56;
disp(H,'The given characteristics equation 1-G(s)H(s)=');
c=coeff(H);
len=length(c);
r=routh_t(H);
disp(r,"Rouths table=");
x=0;
for i=1:len
if(r(i,1)<0)
x=x+1;
end
end
if(x>=1)
printf("From Rouths table, it is clear that the system is unstable.")
else
printf("From Rouths table, it is clear that the system is stable.")
end
```

Simulation output:

Scilab 6.0.2 Console

```
 The given characteristics equation 1-G(s)H(s)=

                2    3    4
   5 +4s +3s  +2s  +s

 Rouths table=

   1.   3.   5.
   2.   4.   0.
   1.   5.   0.
  -6.   0.   0.
   5.   0.   0.
From Rouths table, it is clear that the system is unstable.
-->
```

Scilab 6.0.2 Console

```
 The given characteristics equation 1-G(s)H(s)=

                 2    3    4   5
   56 +8s +42s  +6s  +7s  +s

 Rouths table=

   1.           6.    8.
   7.           42.   56.
   28.          84.   0.
   21.          56.   0.
   9.3333333    0.    0.
   56.          0.    0.
From Rouths table, it is clear that the system is stable.
-->
```

Exp. No. 12 practice examples:

Determine the stability of the closed-loop system using Routh Hurwitz Criterion for the given polynomial equations.

(i) $P(s) = s^4 + 2s^3 + 3s^2 + 4s + 5$ and

(ii) $P(s) = s^5 + 7s^4 + 6s^3 + 42s^2 + 8s + 56$

(iii) P(s)=s^7-2s^6-s^5+2s^4+4s^3-8s^2-4s+8

(iv) P(s)=(s+1)(s+2)(s+3)(s+4)+240

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		**Faculty signature**
Date of record submission		

Result

Ex. No. 13	TRANSFER FUNCTION FROM STATE MODEL AND STATE MODEL FROM TRANSFER FUNCTION
Exp. Date:	

Aim:

To obtain (i) the transfer function from the given state model

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5.008 & -25.1026 & -5.03247 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 25.04 \\ -121.005 \end{bmatrix}, C = [1 \quad 0 \quad 0] \; and \; D = [0]$$

and (ii) the state model from the given transfer functions

$$T(s) = \frac{s}{s^3 + 14s^2 + 56s + 160}$$

Tools Used:

(i) SCILAB software

(ii) PC

Program code:

(a) The transfer function from the given state model

```
clear;
clc;
xdel(winsid());

A=[0 1 0;0 0 1;-5.008 -25.1026 -5.03247];
B=[0;25.04;-121.005];
C=[1 0 0];
D=[0];

disp('The given state model: State equation x*=Ax+Bu and Output Equation Cx+D where');
disp(D,'D=',C,'C=',B,'B=',A,'A=');

H=syslin('c',A,B,C,D);
G=clean(ss2tf(H));

disp(G,'The obtained transfer function is G(s) = ');
```

Simulation output:

```
Scilab 6.0.2 Console

 The given state model: State equation x*=Ax+Bu and Output Equation Cx+D where

 A=

   0.       1.         0.
   0.       0.         1.
  -5.008  -25.1026   -5.03247

 B=

   0.
   25.04
  -121.005

 C=

   1.   0.   0.

 D=

   0.

 The obtained transfer function is G(s) =


          5.0080488 + 25.04s
   --------------------------------
                                2   3
   5.008 + 25.1026s + 5.03247s + s

-->
```

(b) The state model from the given transfer function

Program code:

```
clear;
clc;
xdel(winsid());

s=%s;
Htf=syslin('c',s,160+56*s+14*s^2+s^3);

Hss=tf2ss(Htf);

disp(Hss,'The obtained state space model for the given transfer function: State Equation
x*=Ax+Bu, Output Equation Y=Cx+D');

//To print the answer, Use
//ssprint(Hss)
//Alternate:

[A,B,C,D]=abcd(Htf)

//To cross check, Use
//Htf2=clean(ss2tf(Hss)) //which matches with Htf

disp(D,'D=',C,'C=',B,'B=',A,'A=');
```

Simulation output:

Scilab 6.0.2 Console

```
 The obtained state space model for the given transfer function: State Equation x*=Ax
 +Bu, Output Equation Y=Cx+D
!lss  A  B  C  D  X0  dt  !
   0.  -8.     0.
   7.  -14.    5.
   4.   0.     0.
   0.
   1.
   0.
  -0.125   0.   0.
   0.
   0.
   0.
   0.
 c

 A=

   0.  -8.     0.
   7.  -14.    5.
   4.   0.     0.

 B=

   0.
   1.
   0.

 C=

  -0.125   0.   0.

 D=

   0.
```

Exp. No. 13 practice examples:

(a) Obtain the transfer function from the given state space model $\dot{X} = AX + BU; Y = CX + D$ where

(1) $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -4 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}, C = [1 \quad 0 \quad 0] \; and \; D = [0]$

(2) $A = \begin{bmatrix} -4 & 1 & 0 \\ -3 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 10 \\ 40 \end{bmatrix}, C = [1 \quad 0 \quad 0] \; and \; D = [0]$

(b) Obtain the state model from the given transfer functions

(1) $T(s) = \frac{10}{4s^2+2s+1}$

(2) $T(s) = \frac{2(s+5)}{(s+2)(s+3)(s+4)}$

(3) $T(s) = \frac{10(s+4)}{s(s+1)(s+3)}$

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		**Faculty signature**
Date of record submission		

Result

Ex. No. 14	IMPLEMENTATION OF THE BLOCK DIAGRAM AND STATE MODEL REDUCTION TECHNIQUE
Exp. Date:	

Aim:

To obtain the transfer function by implementing (i) the block diagram reduction technique, and (ii) the state model reduction technique for [a] single input and single output (SISO) system and [b] multiple input and multiple output (MIMO) system.

Tools Used:

(i) SCILAB software

(ii) PC

Program code:

(i) The block diagram reduction technique

```
clc;
s=poly(0,'s');
n1=[2*s];
d1=[3+s^2];
h1=syslin('c',n1/d1);
n2=[2];
d2=[s+4];
h2=syslin('c',n2/d2);
n3=[4];
d3=[s+6];
h3=syslin('c',n3/d3);
n4=[s];
d4=[1+2*s];
h4=syslin('c',n4/d4);
h5=h3/.h4; //h3 and h4 in negative feedback loop
h6=h1*h5; //h1 and h5 are in series
h7=h6/.h3; //h6 and h3 in negative feedback loop
disp(h7,"Reduced transfer function T(s)=");
```

Simulated output:

```
Scilab 6.0.2 Console

 Reduced transfer function T(s)=

                       2      3
           48s + 104s + 16s
   ------------------------------------
                     2       3      4     5
   108 + 356s + 187s + 114s + 29s + 2s

-->
```

(ii) The state model reduction technique

a) for single input and single output (SISO) system

```
clc;
s=poly(0,'s');
A=[0 1;-6 -5];
B=[0;1];
C=[8 1];
[n1,n2]=size(A);
I=eye(n1,n2); //Identity matrix
X=s*I-A;
phi=inv(X); //Inverse of Matrix
Y=C*phi;
Z=Y*B;
//sys=tf2ss(Z)
disp(Z,"The transfer function representation of system is T(S)=");
//disp(sys)
```

Simulated output:

```
Scilab 6.0.2 Console

 The transfer function representation of system is T(S)=

      8 + s
   -----------
             2
   6 + 5s + s

-->
```

b) for multiple input and multiple output (MIMO) system

```
clc;
clear;
close;

A=[0 1;-25 -4];
B=[1 1;0 1];
C=[1 0;0 1];
D=[0 0;0 0];

H=syslin('c',A,B,C,D);

disp(H,'The given state model matrices are');
disp(A,'A=');
disp(B,'B=');
disp(C,'C=');
disp(D,'D=');

G=clean(ss2tf(H));
disp(G,'The obtained transfer functions are');
```

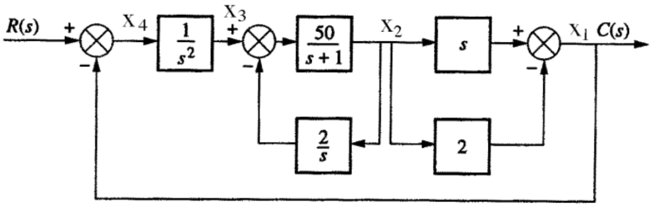
Simulated output:

Scilab 6.0.2 Console

```
 The given state model matrices are

!lss  A  B  C  D  X0  dt  !
   0.    1.
  -25.  -4.
   1.   1.
   0.   1.
   1.   0.
   0.   1.
   0.   0.
   0.   0.
   0.
   0.
 c




   A=

     0.    1.
   -25.  -4.

   B=

     1.   1.
     0.   1.

   C=

     1.   0.
     0.   1.

   D=

     0.   0.
     0.   0.




 The obtained transfer functions are



       4 + s          5 + s
   ------------   ------------
              2              2
   25 + 4s + s    25 + 4s + s

       -25          -25 + s
   ------------   ------------
              2              2
   25 + 4s + s    25 + 4s + s

-->
```

Exp. No. 14 practice examples:

Obtain the transfer function by implementing (i) the block diagram reduction technique for the system shown in figure below



and (ii) the state model reduction technique for [a] single input and single output (SISO) system and

A=[1 1;1 1]; B=[2;2]; C=[3 3];

[b] multiple input and multiple output (MIMO) system.

A=[0 1;-25 -4]; B=[1 1;0 1]; C=[1 0;0 1]; D=[0 0;0 0];

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 15	**TRANSFER FUNCTION TO CONTROLLABLE, OBSERVABLE AND JORDON CANONICAL FORMS**
Exp. Date:	

Aim:

To obtain the controllable, observable and Jordan canonical form from the given transfer function.

$$T(s) = \frac{s+3}{s^2+3s+2}$$

Tools Used:

(i) SCILAB software

(ii) PC

Program:

Function coding:

```
function [r, z, p]=pf_residue(N, D)
z=roots(N) //zeros
p=roots(D) //poles
q=round(p);
m=1; //to keep a count of the roots multiplicity

for i=1:length(p)
if(i<length(p)&q(i+1)==q(i))
m=m+1;
else
P1=N/pdiv(D,(s-p(i))^m);
r(i)=horner(P1,p(i));
for j=1:(m-1)
P1=derivat(P1);
r(i-j)=horner(P1/gamma(j+1),p(i));
end //gamma(j+1)=j! (factorial)
m=1;
end
end
endfunction
//save the file as pf_residue.sci
```

Main program:

```
clc;
clear;
xdel(winsid());

cd"C:\Users\admin\Documents\Scilab"
exec("pf_residue.sci")
s=%s;
N=s+3;
D=s^2+3*s+2;
Hc=cont_frm(N,D);
disp('Controllable form =');
ssprint(Hc);

Ho=syslin('c',(Hc.A)',(Hc.C)',(Hc.B)',Hc.D);
disp('Observable form =');
ssprint(Ho);

A=diag(roots(D));
B=[1;1];
C=pf_residue(N,D)';
D=Hc.D;

Hj=syslin('c',A,B,C,D);
disp('Jordan canonical form =');
ssprint(Hj);
```

Simulated output:

Scilab 6.0.2 Console

```
--> //Partial Fraction Residue gives the coeff of

--> //partial fraction expansion for the given polynomial


--> function [r,z,p]=pf_residue(N,D)
--> z=roots(N) //zeros
--> p=roots(D) //poles
--> q=round(p);
--> m=1; //to keep a count of the roots multiplicity
--> for i=1:length(p)
--> if(i<length(p)&q(i+1)==q(i))
--> m=m+1;
--> else
--> P1=N/pdiv(D,(s-p(i))^m);
--> r(i)=horner(P1,p(i));
--> for j=1:(m-1)
--> P1=derivat(P1);
--> r(i-j)=horner(P1/gamma(j+1),p(i));
--> end //gamma(j+1)=j! (factorial)
--> m=1;
--> end
--> end
--> endfunction

 Controllable form =

+    | 0   1 |      | 0 |
x = |-2  -3 |x + |  1 |u

y = | 3   1 |x

 Observable form =

.    | 0  -2 |      | 3 |
x = | 1  -3 |x + |  1 |u

y = | 0   1 |x

 Jordan canonical form =

.    |-2   0 |      | 1 |
x = | 0  -1 |x + |  1 |u

y = |-1   2 |x

-->
```

Exp. No. 15 practice examples:

Obtain the controllable, observable and Jordan canonical form from the given transfer function.

(a) $T(s) = \frac{s+1}{s^2+6s+25}$

(b) $T(s) = \frac{2(s+5)}{(s+2)(s+3)}$

(Examples simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result

Ex. No. 16	STATE AND OUPUT CONTROLLABILITY AND OBSERVABILITY (USING KALMANS TEST)
Exp. Date:	

Aim:

To obtain the state and output controllability and observability using Kalman’s test.

Tools Used:

(i) SCILAB software

(ii) PC

Program:

```
clear; clc;
xdel(winsid());//close all windows

A=[0 1;-0.4 -1.3];
B=[0;1];
C=[0.8 1];
D=[0];
G1=syslin('c',A,B,C,D); ssprint(G1);

G2=syslin('c',A',C',B',D);ssprint(G2);

Cc1=cont_mat(A,B);disp(Cc1,'state controllability matrix1=');
disp(det(Cc1),'det(Cc1)=');

if(det(Cc1)~=0)
   printf('The given system 1 is completely controllable')
else
   printf('The given system 1 is not completely controllable')
   end

Ob1=obsv_mat(A,C);disp(Ob1,'observability matrix1=');
disp(det(Ob1),'det(Ob1)');

if(det(Ob1)~=0)
   printf('The given system 1 is completely observable')
else
   printf('The given system 1 is not completely observable')
```

```
  end
Cc2=cont_mat(A',C');disp(Cc2,'state controllability matrix2=');
disp(det(Cc2),'det(Cc2)=');
if(det(Cc2)~=0)
  printf('The given system 2 is completely controllable')
else
  printf('The given system 2 is not completely controllable')
 end
Ob2=obsv_mat(A',B');disp(Ob2,'observability matrix2=');
disp(det(Ob2),'det(Ob2)');
if(det(Ob2)~=0)
  printf('The given system 2 is completely observable')
else
  printf('The given system 2 is not completely observable')
  end
Htf=ss2tf(G1);disp (Htf,'Reduced transfer function=');
e=spec(A); disp (e,'Eigen values=');
D=poly(e,'s'); disp(D,'actual denominator (characteristic poly)=');
//Htf=ss2tf(G1);disp (Htf,'Reduced transfer function=');
//e=spec(A); disp (e,'Eigen values=');
//D=poly(e,'s'); disp(D,'actual denominator (characteristic poly)=');
```

Simulated output:

```
.    | 0     1    |     | 0 |
x = |-0.4 -1.3 |x + | 1 |u

y = | 0.8   1 |x

.    | 0 -0.4 |     | 0.8 |
x = | 1 -1.3 |x + | 1   |u

y = | 0  1 |x

 state controllability matrix1=

   0.    1.
   1.  -1.3

 det(Cc1)=

  -1.
The given system 1 is completely controllable
 observability matrix1=

   0.8   1.
  -0.4  -0.5

 det(Ob1)

   0.
The given system 1 is not completely observable
```

```
 state controllability matrix2=

   0.8  -0.4
   1.   -0.5

 det(Cc2)=

   0.
The given system 2 is not completely controllable
 observability matrix2=

   0.   1.
   1.  -1.3

 det(Ob2)

  -1.
The given system 2 is completely observable
 Reduced transfer function=


      1
   --------

   0.5 + s

 Eigen values=

  -0.5
  -0.8

 actual denominator (characteristic poly)=

                2
   0.4 +1.3s +s

-->
```

Exp. No. 16 practice examples:

Obtain the state and output controllability and observability using Kalman's test.

For the system A=[0 1;-1 -2], B=[0;1], C=[2 4] and D=[0]

(Examples' simulated output separately attached)

Marks obtained:

Theoretical Calculations	**20**	
Observation	**20**	
Execution of practice examples	**30**	
Viva	**10**	
Record	**20**	
Total Score	**100**	
Date of experiment		
Date of record submission		**Faculty signature**

Result