Scilab Manual for Control System Design by Prof Deepti Khimani Instrumentation Engineering VESIT[1]

Solutions provided by
Prof Mrs. Deepti Khimani
Instrumentation Engineering
Mumbai University/VES Institute of Technology

September 5, 2026

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Scilab Solutions 3

1 Obtain state model of the second order system cascaded with active lead circuit and show its step response. 5

2 Determine eigen values of the state model. Also convert the state model into transfer function. 8

3 Transform the given system having distinct eigen values into controllable canonical and diagonal form. 10

4 Obtain the step and impulse response of the state model. 13

5 Check for the controllability and observability of a given system. 17

6 Obtain state feedback gain matrix for the given system. 19

7 Design a full state observer for the system. 22

8 Determine steady state error of the given system. 25

9 Compensation of system using lead compensator designed via root locus technique. 28

10 Design a lead compensator for the given system using bode plot. 30

List of Experiments

Solution 1.01	Lab 01	5
Solution 2.02	Lab02	8
Solution 3.03	Lab3	10
Solution 4.04	Lab 04	13
Solution 5.05	Lab05	17
Solution 6.06	Lab 06	19
Solution 7.07	Lab 07	22
Solution 8.08	Lab 08	25
Solution 9.09	Lab 09	28
Solution 10.10	Lab 10	30

List of Figures

1.1 Lab 01 7

4.1 Lab 04 15
4.2 Lab 04 16

6.1 Lab 06 21

7.1 Lab 07 24

8.1 Lab 08 27

9.1 Lab 09 29

10.1 Lab 10 32
10.2 Lab 10 33

Experiment: 1

Obtain state model of the second order system cascaded with active lead circuit and show its step response.

Scilab code Solution 1.01 Lab 01

```
1 //
    ------------------------------------------------------------------------

2 // Lab. 01: Obtain state model of the second order
    system cascaded with active
3 // lead circuit. Show its step response.
4 //
    ------------------------------------------------------------------------

5
6 //scilab -5.5.0
7 //Operating System : OS X 10.9.3
8
9 //Clean the environment
10 clc;
```

```
clear all;
clf;

// Compensator model
R1=1000; R2=5e3;C1=1e-6;C2=1e-5;
kc=5;
s=poly(0,'s');
g=kc*(R1*C1*s+1)/(R2*C2*s+1);

// System transfer function
g1=0.2/(s^2+1.7*s+1);

// Overall transfer function
sys=tf2ss(g*g1);

// Unit step response
t=linspace(0,10,1000);
y=csim('step',t,sys);
plot(t,y);
title('Unit step response of the electrical system',
   'fontsize',4)
xlabel('Time t','fontsize',2)
ylabel('Response y(t)','fontsize',2)
//set(gca(),"grid",[0.3 0.3])
```

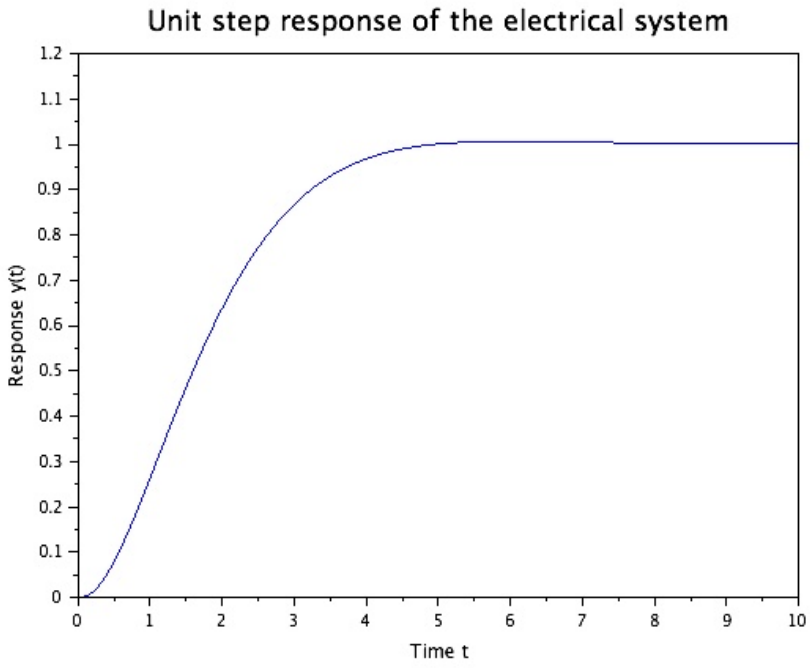

Figure 1.1: Lab 01

Experiment: 2

Determine eigen values of the state model. Also convert the state model into transfer function.

Scilab code Solution 2.02 Lab02

```

//
// Lab. 02: Determine eigen values of the state model.
// Convert the state model into transfer function.
//

//scilab-5.5.0
//Operating System : OS X 10.9.3

//Clean the environment
```

```
clc;
clear all;
//clf;

// State space representation
A=[0 1 0; 0 0 1; -5 -25 -5];
B=[0; 25;-120];
C=[1 0 0];
D=0;

sys1=syslin('c',A,B,C,D);
mprintf('State space representation of the given
   system is')
disp(sys1)

// Eigen values of system matrix
eig_val=spec(A)
mprintf('Eigen values of the system matrix are')
disp(eig_val)

// Transfer function of the given system
g1=ss2tf(sys1)
mprintf('Transfer function representation of the
   given system is')
disp(g1)
```

Experiment: 3

Transform the given system having distinct eigen values into controllable canonical and diagonal form.

Scilab code Solution 3.03 Lab3

```

//
// Lab. 03: Transform the given system having distinct eigen values into
// controllable canonical and diagonal form.
//

//scilab-5.5.0
//Operating System : OS X 10.9.3

//Clean the environment
```

```
clc;
clear all;
//clf;

// State space model
A=[-3 1; 1 -3];
B=[1;2];
C=[2 3];
D=0;

sys=syslin('c',A,B,C,D)
mprintf('State space representation of the given
   system is')
disp(sys)


// Eigen values of system matrix
eig_val=spec(A)
mprintf('Eigen values of the system matrix are')
disp(eig_val)

// Controllable canonical form
[Ac, Bc T]=canon(A,B)
T=flipdim(T,2);
Ac=T\A*T;
Bc=T\B;
Cc=C*T;
Dc=D;
sysc=syslin('c',Ac,Bc,Cc,Dc)
mprintf('State space representation of the given
   system in Controllable canonical form is')
disp(sysc)

// Diagonal form
[Ad M]=bdiag(A);
Bd=M\B;
Cd=C*M;
Dd=D;
```

```
47 sysd=syslin('c',Ad,Bd,Cd,Dd)
48 mprintf('State space representation of the given
     system in Diagonal form is')
49 disp(sysd)
```

Experiment: 4

Obtain the step and impulse response of the state model.

Scilab code Solution 4.04 Lab 04

```
1 //
  ------------------------------------------------------------------------

2 // Lab. 04: Obtain the step and impulse response of
     the state model.
3 //
  ------------------------------------------------------------------------

4
5 //scilab -5.5.0
6 //Operating System : OS X 10.9.3
7
8 //Clean the environment
9 clc;
10 clear all;
11 clf;
12
13 // State space representation
14 A=[-2 -1; -1 -1];
```

```
15 B=[1;1];
16 C=[0 2];
17 D=0;
18 x0=[0;5]; // Initial condition
19 sys=syslin('c',A,B,C,D)
20
21 // Response to a given input
22 figure(0)
23 t=linspace(0,20,1001);
24 temp=size(t);
25 u=ones(temp(1),temp(2)); // Exogenous signal(step)
26 y=csim(u,t,sys,x0)
27 plot(t,y)
28 title('Unit step response of the system','fontsize'
      ,4)
29 xlabel('Time t','fontsize',2)
30 ylabel('Response y(t)','fontsize',2)
31
32 // Response to a given input
33 figure(1)
34 t=linspace(0,10,1001);
35 y=csim('impuls',t,sys)
36 plot(t,y)
37 title('Impulse response of the system','fontsize',4)
38 xlabel('Time t','fontsize',2)
39 ylabel('Response y(t)','fontsize',2)
```

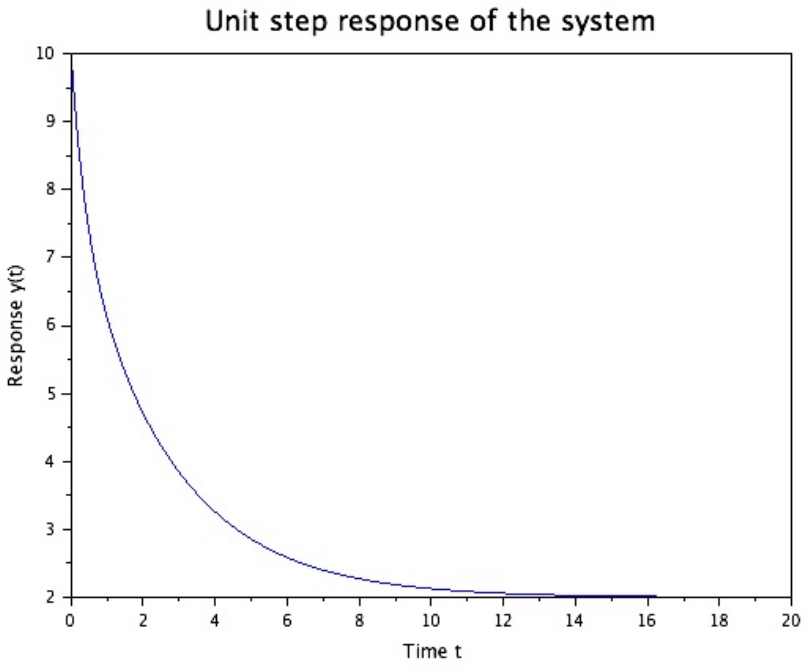

Figure 4.1: Lab 04

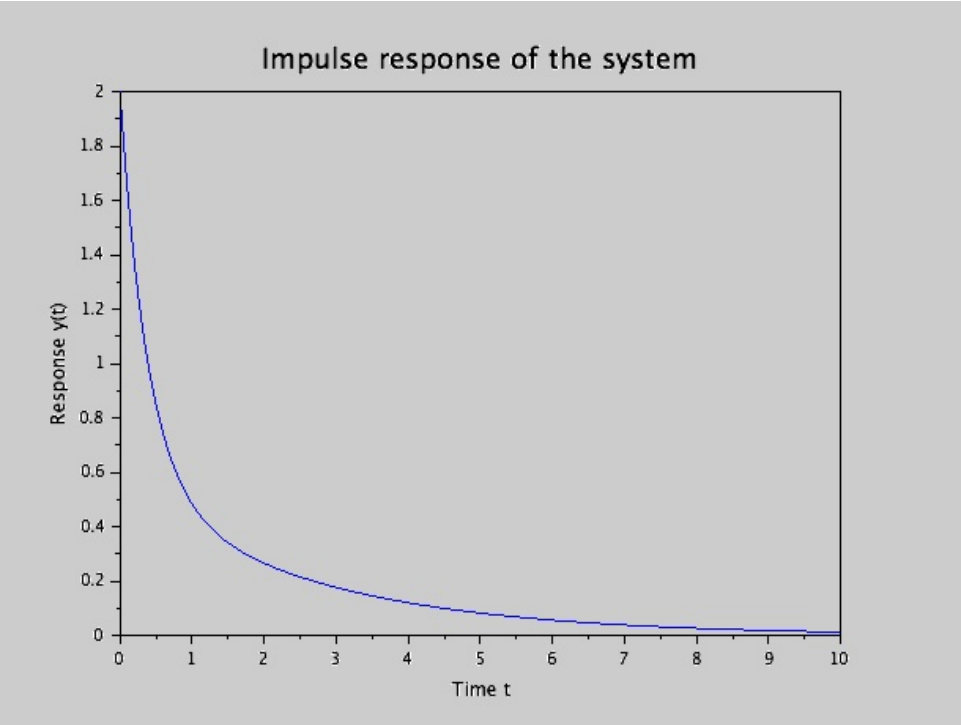

Figure 4.2: Lab 04

Experiment: 5

Check for the controllability and observability of a given system.

Scilab code Solution 5.05 Lab05

```

//
// Lab. 05: Check for the controllability and
   observability of a given system.
//

//scilab-5.5.0
//Operating System : OS X 10.9.3

//Clean the environment
clc;
clear all;
//clf;
```

```
13
14 // State space representation
15 A=[-5 1 0; 0 -2 1; 0 0 -1];
16 B=[6 0 1]';
17 C=[1 0 0];
18 D=0;
19 sys=syslin('c',A,B,C,D)
20
21 // Controllability test
22 n=cont_mat(sys)
23 mprintf('Controllability matrix is')
24 disp(n)
25
26 if rank(n)==3 then
27     disp('System is controllable')
28 else
29     disp('System is uncontrollable')
30 end
31
32 // Observability test
33 m=obsv_mat(sys)
34 mprintf('Observability matrix is')
35 disp(m)
36
37 if rank(m)==3 then
38     disp('System is observable')
39 else
40     disp('System is unobservable')
41 end
```

Experiment: 6

Obtain state feedback gain matrix for the given system.

Scilab code Solution 6.06 Lab 06

```
1 //
    ____________________________________________________________________

2 // Lab. 06: Obtain state feedback gain matrix for
    the given system.
3 //
    ____________________________________________________________________

4
5 //scilab -5.5.0
6 //Operating System : OS X 10.9.3
7
8 //Clean the environment
9 clc;
10 clear all;
11 clf;
12
13 // State space representation
14 A=[0 1 0; 0 0 1; -1 -5 -6];
```

```
15 B=[0 0 1]';
16 C=[0 0 1];
17 D=0;
18
19 // Desired poles
20 Pd=[-1+2*%i -1-2*%i -10];
21
22 // State feedback gain matrix
23 K=ppol(A,B,Pd)
24
25 //Closed loop system
26 sys=syslin('c',A-B*K,B,C,D)
27
28 //Response of closed loop system
29 t=linspace(0,20,1001);
30 y=csim('step',t,sys)
31 plot(t,y)
32 title('Response of the closed loop system','fontsize
     ',4)
33 xlabel('Time t','fontsize',2)
34 ylabel('Response y(t)','fontsize',2)
```

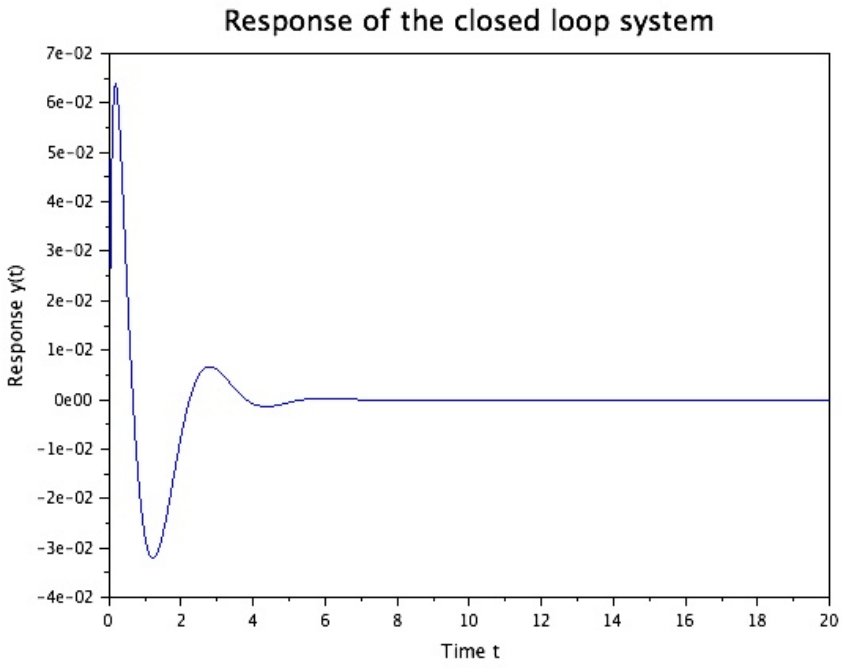

Figure 6.1: Lab 06

Experiment: 7

Design a full state observer for the system.

Scilab code Solution 7.07 Lab 07

```
1 //
  ____________________________________________________________________

2 // Lab. 07: Design a full state observer for the
  system.
3 //
  ____________________________________________________________________


4
5
6 //scilab -5.5.0
7 //Operating System : OS X 10.9.3
8
9 //Clean the environment
10 clc;
11 clear all;
12 clf;
13
14 //State space model
```

```
A=[1 -1 2; 2 -1 3; -1 -2 4];
B=[1 1 0]';
C=[1 1 0];
D=0;

//Stabilizer design
// Desired poles
Pd=[-7 -5 -10];

// State feedback gain matrix
K=ppol(A,B,Pd)

//Computation of observer gain
obsr_pol=[-20+0.5*%i -20-0.5*%i -60];
L=ppol(A',C',obsr_pol)'

// Augmented system
temp=size(A);
Aa=[A-B*K    B*K; zeros(temp(1),temp(2))     A-L*C
   ];
temp=size(Aa);
Ba=zeros(temp(1),1);
Ca=eye(6,6);
sys=syslin('c',Aa,Ba,Ca,zeros(6,1))

//Observer error
figure(0)
t=linspace(0,0.6,1001);
x0=[0 0 0 1 1 1 ]';
temp=size(t);
u=zeros(temp(1),temp(2)); // Exogenous signal(step)
y=csim(u,t,sys,x0)
plot(t,y(4:6,:))
title('Observer error','fontsize',4)
xlabel('$t$','fontsize',2)
ylabel('$x(t)-\hat x(t)$','fontsize',2)
legend('$x_1$','$x_2$','$x_3$')
```

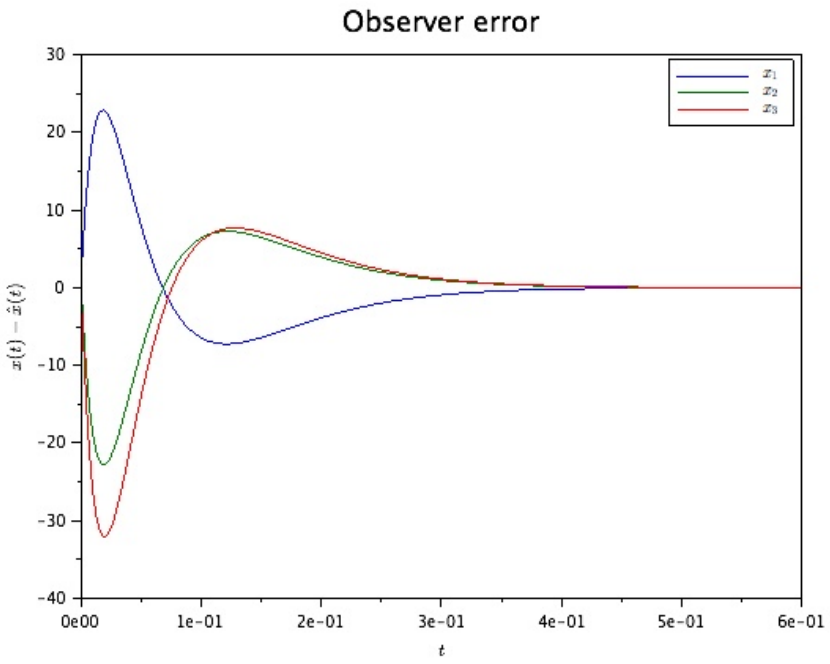

Figure 7.1: Lab 07

Experiment: 8

Determine steady state error of the given system.

Scilab code Solution 8.08 Lab 08

```
1 //
  ------------------------------------------------------------------------

2 // Lab. 08: Determine steady state error of the
     given system.
3 //
  ------------------------------------------------------------------------

4
5 //scilab -5.5.0
6 //Operating System : OS X 10.9.3
7
8 //Clean the environment
9 clc;
10 clear all;
11 clf;
12
13 //State space model
14 a=[0 1;-7 -9];
```

```
15 b=[0 1]';
16 c=[4 1];
17 d=0;
18 sys=syslin('c',a,b,c,d)
19
20 //Error in response of the system
21 t=linspace(0,20,1001);
22 y=csim('step',t,sys)
23 plot(t,1-y)
24 title('Error in response','fontsize',4)
25 xlabel('Time t','fontsize',2)
26 ylabel('Response y(t)','fontsize',2)
27
28 // Steady state error computation
29 ess=1+c*inv(a)*b
```

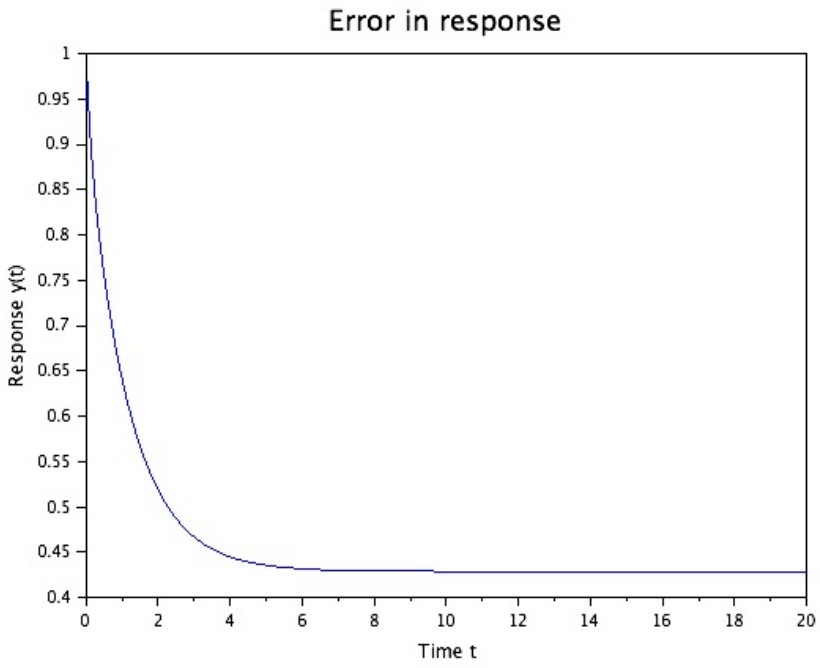

Figure 8.1: Lab 08

Experiment: 9

Compensation of system using lead compensator designed via root locus technique.

Scilab code Solution 9.09 Lab 09

```
1 //
  ------------------------------------------------------------------------

2 // Lab. 09: Compensation of system using lead
     compensator designed via root
3 //locus technique.
4 //
  ------------------------------------------------------------------------

5
6 //scilab -5.5.0
7 //Operating System : OS X 10.9.3
8
9 //Clean the environment
10 clc;
11 clear all;
12 clf;
```

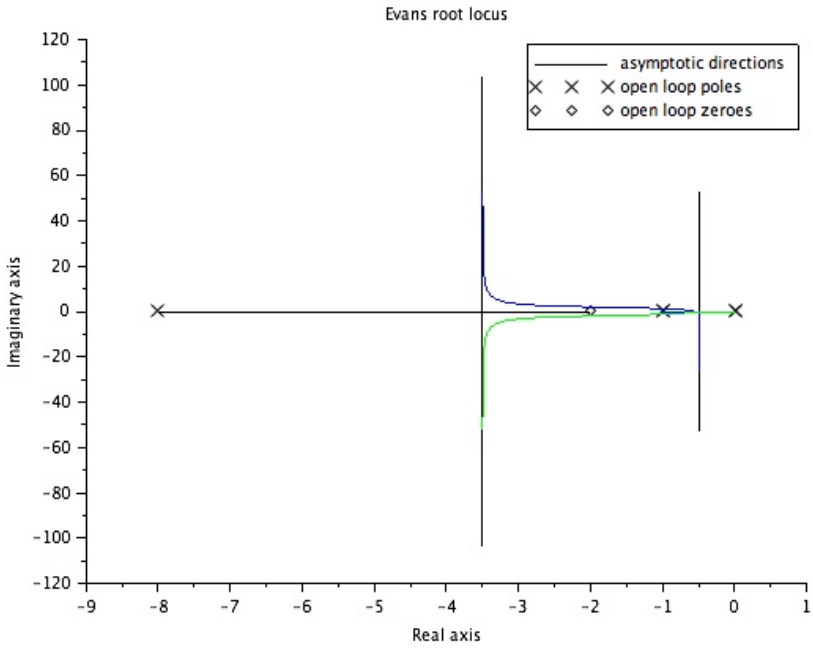

Figure 9.1: Lab 09

```
13
14 //System transfer function and its root locus
15 s=poly(0,'s');
16 g=1/(s*(s+1));
17 evans(g)
18
19 //Designed compensator
20 gc=(s+2)/(s+8);
21
22 //Root locus of compensated system
23 evans(g*gc)
```

Experiment: 10

Design a lead compensator for the given system using bode plot.

Scilab code Solution 10.10 Lab 10

```
1 //
  ____________________________________________________________________

2 // Lab.10: Design a lead compensator for the given
    system using bode plot.
3 // System is g=K/s(s+2). Design specifications: Kv
    =20 sec^-1 and PM=45 deg.
4 //
  ____________________________________________________________________

5
6 //scilab -5.5.0
7 //Operating System : OS X 10.9.3
8
9 //Clean the environment
10 clc;
11 clear all;
```

```
clf;

//Desired specifications
Phi_s=45;
K=40;

//Uncompenstated system
s=poly(0,'s');
g=syslin('c',40/(s*(s+2)));

//Bode plot of the uncompenmsated system
bode(g,0.001,1000)
title('uncompensated system')
gm=g_margin(g)
pm=p_margin(g)
eps1=10;
Phi_m=(Phi_s-pm+eps1)*%pi/180
alpha=(1-sin(Phi_m))/(1+sin(Phi_m))
gain_phi_m=-10*log10(1/alpha)

// Observed frequency at gain_phi_m
wc2=9.3

// Corner frequency
w1=wc2*sqrt(alpha)
w2=wc2/sqrt(alpha)
Gc=(s+w1)/(s+w2)

//The bode plot of compensated system
figure(1);
bode(Gc*g,0.001,1000),
title('Compensated system')
```

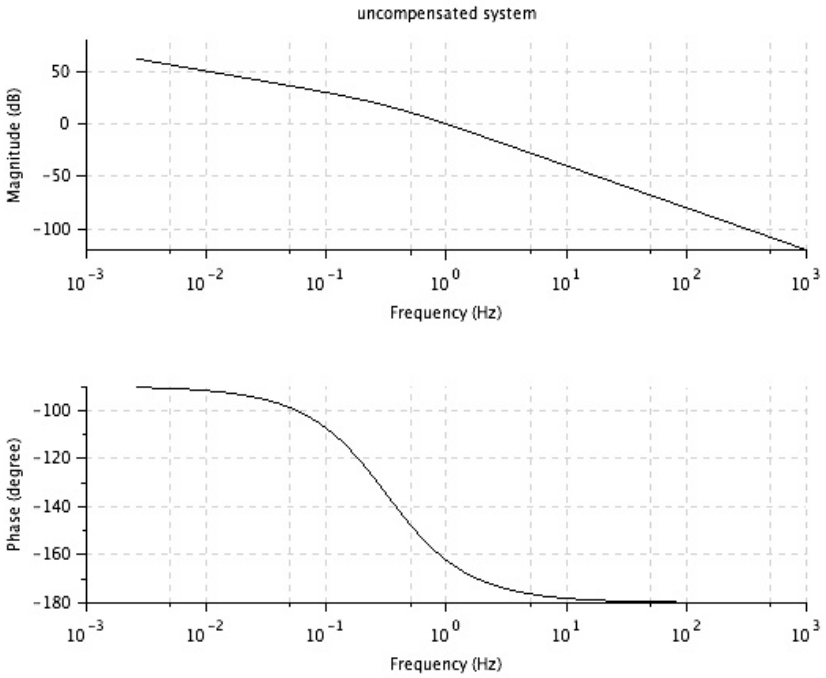

Figure 10.1: Lab 10

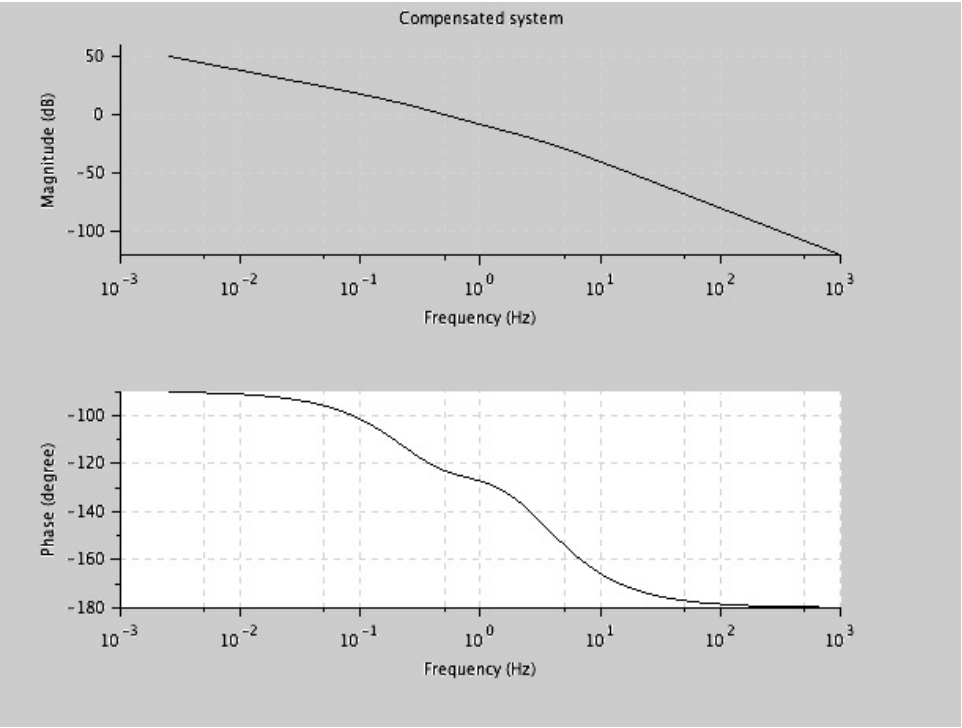

Figure 10.2: Lab 10