

Scilab Manual for  
Control System Design  
by Prof Dilip Pawar  
Instrumentation Engineering  
Bharati Vidyapeeth College Of Engineering,  
Navi Mumbai<sup>1</sup>

Solutions provided by  
Mr Dilip N. Pawar  
Instrumentation Engineering  
Bharati Vidyapeeth College Of Engineering, Navi Mumbai

May 11, 2025

<sup>1</sup>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 State model for SISO and MIMO Systems          | 6  |
| 2 Similarity Transformation                      | 8  |
| 3 Controllability and Observability              | 10 |
| 4 Pole-Placement with State Feedback             | 12 |
| 5 Lead Compensator Design by Root – Locus        | 14 |
| 6 Lag Compensator Design by Root – Locus         | 18 |
| 7 Time Delay or Transportation Lag               | 22 |
| 8 Lead Compensator Design by Bode Plot Technique | 26 |
| 9 Lag Compensator Design by Bode Plot Technique  | 32 |
| 10 PID Controller Design                         | 38 |
| 11 Servomechanism or Tracking Problem            | 41 |

# List of Experiments

|              |  |    |
|--------------|--|----|
| Solution 1.1 | a . . . . .  | 6  |
| Solution 2.2 | a . . . . .  | 8  |
| Solution 3.3 | a . . . . .  | 10 |
| Solution 4.4 | a . . . . .  | 12 |
| Solution 5.5 | To design a Lead Compensator by Root Locus technique . . . . . | 14 |
| Solution 6.6 | To design a Lag Compensator by Root Locus technique . . . . .  | 18 |
| Solution 7.7 | To Study the Effect of Time Delay on System Response . . . . . | 22 |
| Solution 8.8 | To design a Lead Comensator by Bode plot technique . . . . .   | 26 |
| Solution 9.9 | To design a Lag Compensator by Bode plot technique . . . . .   | 32 |

# List of Figures

|      |  |    |
|------|--|----|
| 5.1  | To design a Lead Compensator by Root Locus technique . .                   | 16 |
| 5.2  | To design a Lead Compensator by Root Locus technique . .                   | 16 |
| 5.3  | To design a Lead Compensator by Root Locus technique . .                   | 17 |
| 5.4  | To design a Lead Compensator by Root Locus technique . .                   | 17 |
| 6.1  | To design a Lag Compensator by Root Locus technique . . .                  | 20 |
| 6.2  | To design a Lag Compensator by Root Locus technique . . .                  | 20 |
| 6.3  | To design a Lag Compensator by Root Locus technique . . .                  | 21 |
| 6.4  | To design a Lag Compensator by Root Locus technique . . .                  | 21 |
| 7.1  | To Study the Effect of Time Delay on System Response . . .                 | 24 |
| 7.2  | To Study the Effect of Time Delay on System Response . . .                 | 25 |
| 8.1  | To design a Lead Comensator by Bode plot technique . . . .                 | 29 |
| 8.2  | To design a Lead Comensator by Bode plot technique . . . .                 | 30 |
| 8.3  | To design a Lead Comensator by Bode plot technique . . . .                 | 30 |
| 8.4  | To design a Lead Comensator by Bode plot technique . . . .                 | 31 |
| 9.1  | To design a Lag Compensator by Bode plot technique . . . .                 | 35 |
| 9.2  | To design a Lag Compensator by Bode plot technique . . . .                 | 36 |
| 9.3  | To design a Lag Compensator by Bode plot technique . . . .                 | 36 |
| 9.4  | To design a Lag Compensator by Bode plot technique . . . .                 | 37 |
| 10.1 | To design a PID Controller by using Ziegler Nichols tuning rules . . . . . | 39 |
| 10.2 | To design a PID Controller by using Ziegler Nichols tuning rules . . . . . | 40 |
| 11.1 | To apply integral control to servo problem to minimize the error . . . . . | 42 |

|  |    |
|--|----|
| 11.2 To apply integral control to servo problem to minimize the<br>error . . . . . | 43 |
|--|----|

# Experiment: 1

## State model for SISO and MIMO Systems

Scilab code Solution 1.1 a

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Lab 1 a): To obtain State – Space Model of given
  system
4
5 clc ;
6 clear all;
7
8
9 // Define the Numerator and Denominator Polynomials
  of Transfer Function , G(s)
10 num=poly([30 10], "s", "coeff"); //Defines the
  numerator of G(s)
11 den=poly([0 8 6 1], "s", "coeff"); //Defines the
  denominator of G(s)
12
13 // Obtain the Controllable Phase Variable Form
14 sys=cont_frm(num,den);
15 [A,B,C,D]=abcd(sys)
```

```

16 disp(D,"D",C,"C",B,"B",A,"A")
17
18 // Result
19
20 // A
21
22 //      0.      1.      0.
23 //      0.      0.      1.
24 //      0.     -8.     -6.
25
26 // B
27
28 //      0.
29 //      0.
30 //      1.
31
32 // C
33
34 //      30.      10.      0.
35
36 // D
37
38 //      0.

```

---



## Experiment: 2

# Similarity Transformation

Scilab code Solution 2.2 a

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Lab 2 a): To obtain Similar State – Space Model
  of given system
4
5 clc ;
6 clear all;
7
8 // A system having the form  $\dot{x}=Ax+Bu$  and  $y=Cx+du$ ,
  where
9  $A=[-2,1;-3,0]$ ;  $B=[4;5]$ ;  $C=[1,0]$ ;  $D=0$ ; //and
10 //The transformation Matrix P is taken as –
11  $P = [2,1;4,3]$ ;
12 //the given system can be transformed to a similar
  system as –
13  $AA=\text{inv}(P)*A*P$ 
14  $BB=\text{inv}(P)*B$ 
15  $CC=C*P$ 
16 disp(C,"CC",B,"BB", A,"AA")
17 // Check the eigenvalues of both systems
18  $EV1=\text{spec}(A)$ ;  $EV2=\text{spec}(AA)$ ;
```

```

19 disp(EV2," Eigenvalues  of System2", EV1," Eigenvalues
    of System1")
20 // If both eigenvalues are same then systems are
    similar
21
22 // Result
23
24 // AA
25 //
26 //    -2.    1.
27 //    -3.    0.
28 //
29 // BB
30 //
31 //     4.
32 //     5.
33 //
34 // CC
35 //
36 //     1.    0.
37 //
38 // Eigenvalues  of System1
39 //
40 //    -1. + 1.4142136 i
41 //    -1. - 1.4142136 i
42 //
43 // Eigenvalues  of System2
44 //
45 //    -1. + 1.4142136 i
46 //    -1. - 1.4142136 i
47 //

```

---

# Experiment: 3

## Controllability and Observability

Scilab code Solution 3.3 a

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Lab 3 a): To Check the Controllability of given
   system by Gilbert's Test
4
5 clc ;
6 clear all;
7
8 // Let the system is described by state equations
9 A=[0,1,0;0,0,1;-6,-11,-6];
10 B=[0;0;1];
11 // Solution: –
12 // Gilbert's Test is required to find canonical
   state variable form as –
13 // Given system is in Phase Variable form, therefore
   Vander Monde Matrix is
14 // required as an Modal Matrix
15 // Find the eigenvalues and eigenvectors of matrix A
16 [V,D]=spec(A);
```

```

17 // Forming the Vander Monde Matrix from the
    eigenvalues as -
18 d = diag(D); dd=(d.^2);
19 M=[1,1,1;d';dd'];
20 Minv=inv(M);
21 // Finding the Bcap
22 Bcap=Minv*B;
23 disp(Bcap,"Bcap")
24 // Check the contents of B vector , whether any
    element is zero ,
25 // if not then system is controllable
26 // Result
27
28 // Bcap
29 //
30 //    0.5
31 //   -1.
32 //    0.5

```

---

# Experiment: 4

## Pole-Placement with State Feedback

Scilab code Solution 4.4 a

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Lab 4 a): To find a gain matrix K, by using Pole-
    placement by state feedback
4
5 clc ;
6 clear all;
7
8 // Problem: Consider the system as
9 // .
10 //  $\dot{X} = AX + Bu$ 
11 // where  $A = [0, 1, 0; 0, 0, 1; -1, -5, -6]$ ;  $B = [0; 0; 1]$ ;
12 // By using state feedback control  $u = -Kx$ , it is
    desired to have the closed
13 // - loop poles at  $s_1 = -2+j*4$ ;  $s_2 = -2-j*4$ ; and  $s_3$ 
    = -10;
14 // Determine the state feedback-gain matrix K with
    SCILAB
15
```

```

16 // First define the given A, B matrices as –
17 A = [0,1,0;0,0,1;-1,-5,-6];
18 B = [0;0;1];
19 // Then define the closed – loop poles as –
20 s1=-2+%i*4; s2=-2-%i*4; s3=-10;
21 S=[s1,s2,s3];
22 // Then invoking commands 'ppol' as –
23 K=ppol(A,B,S)
24 disp(K,"K")
25
26 // Result
27
28 // K
29 //
30 //      199.      55.      8.

```

---

## Experiment: 5

# Lead Compensator Design by Root – Locus

**Scilab code Solution 5.5** To design a Lead Compensator by Root Locus technique

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Lab 5: To design a Lead Compensator by Root–Locus
  technique
4
5 clc ;
6 clear all;
7
8
9
10 s=%s;
11 G=syslin('c',1/s^2)      // Transfer Function of Un-
  compensated System
12 Gc=syslin('c',(s+1.15)/((s^2)*(s+4.2))) // Transfer
  Function of Compensated System
13 // Root – Locus of Un–Compensated System
14 figure(0)
15 clf, evans(G), xgrid(5, 1, 7)
```

```

16
17 // Root – Locus of Compensated System
18 figure(1)
19 clf, evans(Gc), xgrid(5, 1, 7)
20 replot([-4.5, -10, 0.5, 10])
21
22 // Step Response of Un-Compensated System
23 figure(2)
24 t= 0:0.1:100;
25 x=[csim('step',t,G/(1+G),[0;0;0])]';
26 plot2d(t',x),
27 xlabel("t", "fontsize", 2,"color", "blue");
28 ylabel("Amplitude", "fontsize", 2, "color", "blue");
29 xgrid(5, 1, 7)
30 xtitle('Step Response of Un-Compensated System');//
    , 'X axis ', 'Y axis ');
31 // Step Response of Compensated System
32 figure(3)
33 t= 0:0.1:100;
34 x=[csim('step',t,Gc/(1+Gc),[0;0;0])]';
35 plot2d(t',x),
36 xlabel("t", "fontsize", 2,"color", "blue");
37 ylabel("Amplitude", "fontsize", 2, "color", "blue");
38 xtitle('Step Response of Compensated System');//, '
    X axis ', 'Y axis ');
39 xgrid(5, 1, 7)

```

---



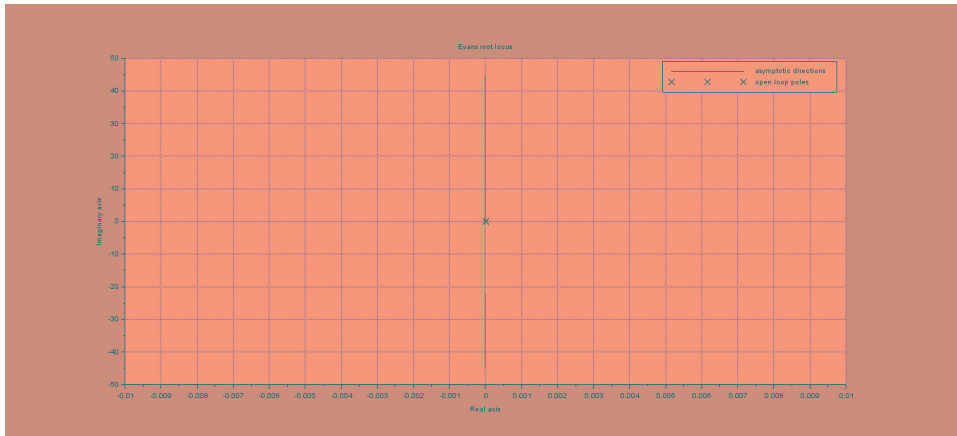


Figure 5.1: To design a Lead Compensator by Root Locus technique

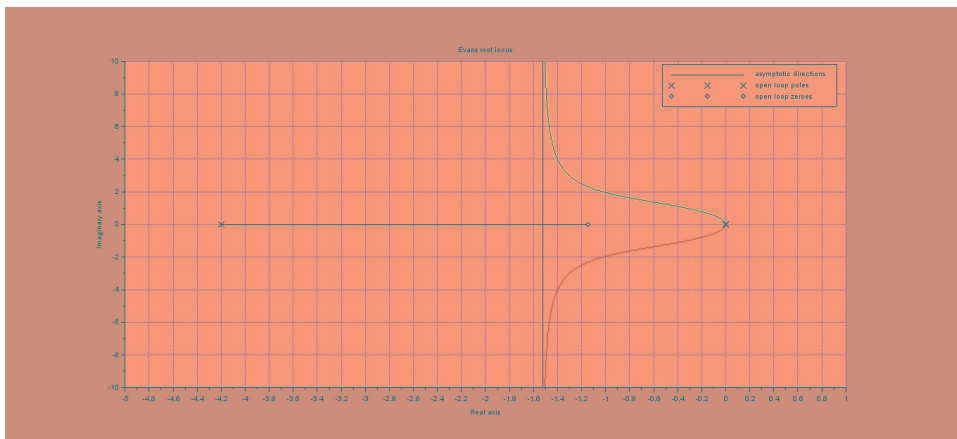


Figure 5.2: To design a Lead Compensator by Root Locus technique

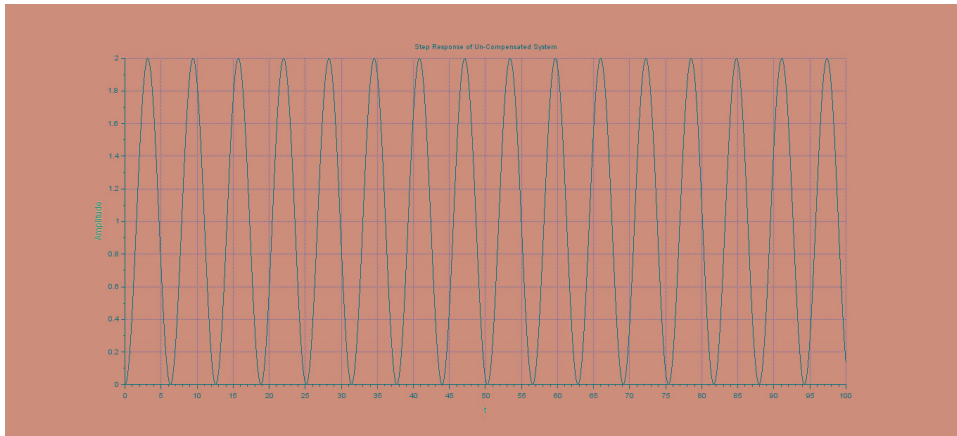


Figure 5.3: To design a Lead Compensator by Root Locus technique

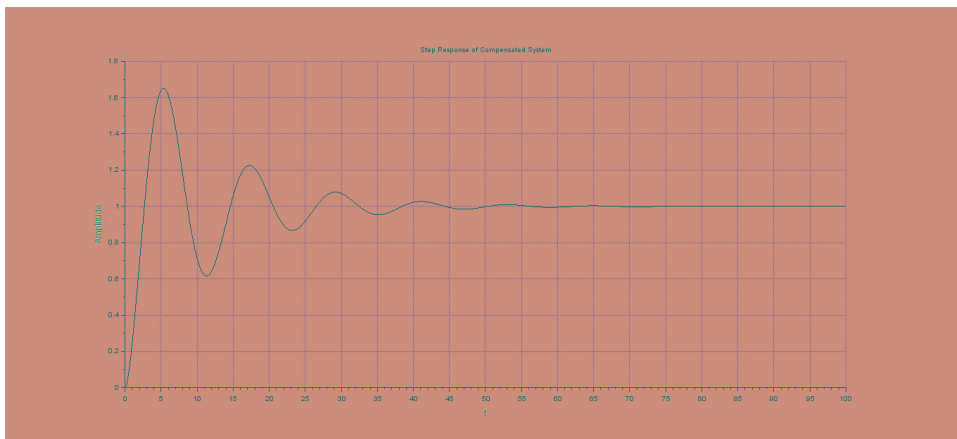


Figure 5.4: To design a Lead Compensator by Root Locus technique

## Experiment: 6

# Lag Compensator Design by Root – Locus

**Scilab code Solution 6.6** To design a Lag Compensator by Root Locus technique

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Lab 6: To design a Lag Compensator by Root–Locus
  technique
4
5 clc ;
6 clear all;
7
8 s=%s;
9 G=syslin('c',1/(s*(s+1)*(s+4))) // Transfer Function
  of Un–Compensated System
10 Gc=syslin('c',(s+0.2)/((s+0.02)*s*(s+1)*(s+4))) //
  Transfer Function of Compensated System
11 // Root – Locus of Un–Compensated System
12 figure(0)
13 clf, evans(G), xgrid//(5, 1, 7)
14 replot([-4.5,-10,0.5,10])
15 // Step Response of Un–Compensated System
```

```

16 figure(1)
17 t= 0:0.1:100;
18 x=[csim('step',t,G/(1+G),[0;0;0])]';
19 plot2d(t',x),xgrid//(5, 1, 7)
20 xlabel("t", "fontsize", 2,"color", "blue");
21 ylabel("Amplitude", "fontsize", 2, "color", "blue");
22 xtitle( 'Step Response of Un-Compensated System');
23
24 // Root – Locus of Compensated System
25 figure(2)
26 clf, evans(Gc), xgrid//(5, 1, 7)
27
28 replot([-4.5,-6,0.5,6])
29 // Step Response of Compensated System
30 figure(3)
31 t= 0:0.1:100;
32 x=[csim('step',t,Gc/(1+Gc))]';
33 plot2d(t',x),xgrid//(5, 1, 7)
34 xlabel("t", "fontsize", 2,"color", "blue");
35 ylabel("Amplitude", "fontsize", 2, "color", "blue");
36 xtitle( 'Step Response of Compensated System');

```

---

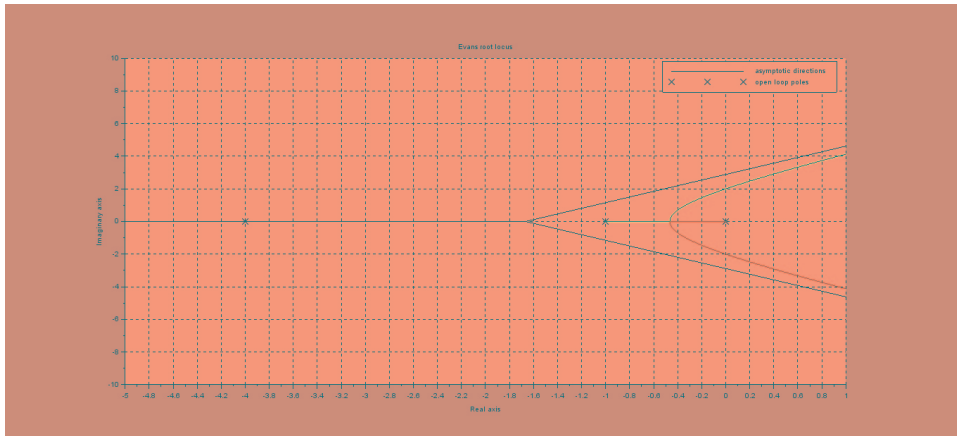


Figure 6.1: To design a Lag Compensator by Root Locus technique

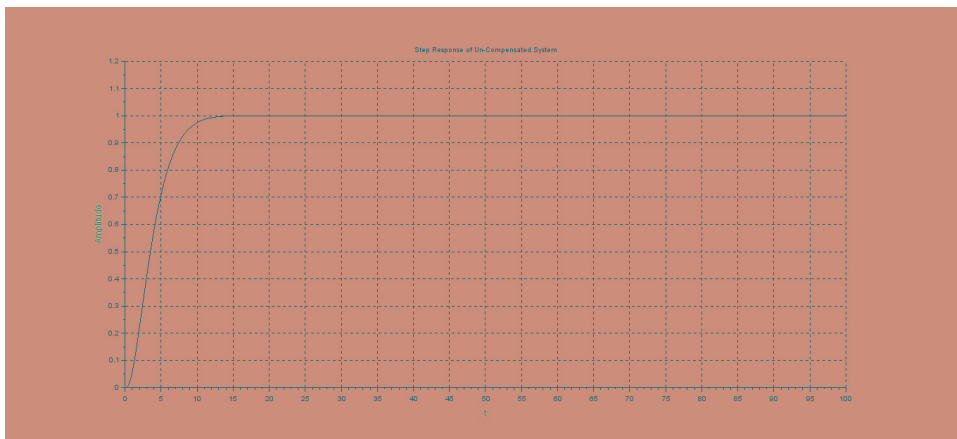


Figure 6.2: To design a Lag Compensator by Root Locus technique

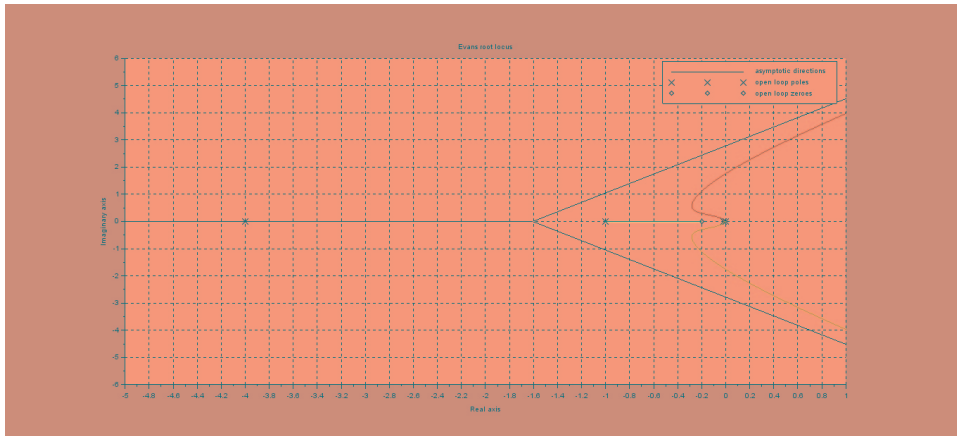


Figure 6.3: To design a Lag Compensator by Root Locus technique

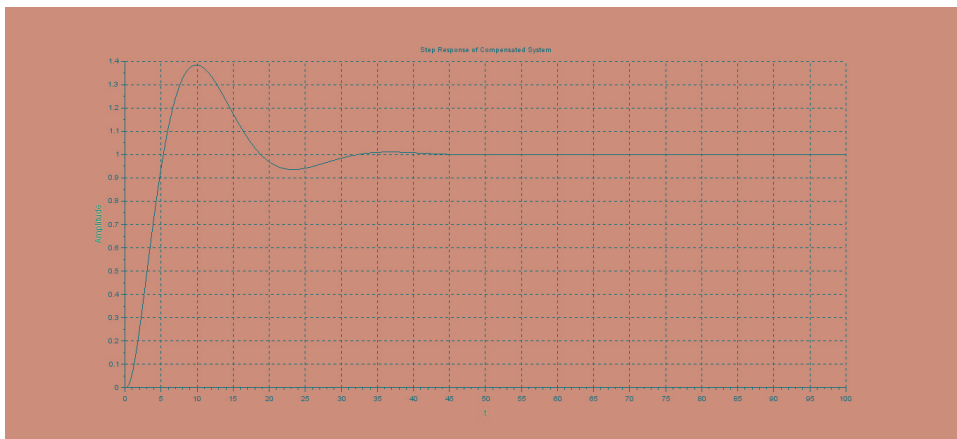


Figure 6.4: To design a Lag Compensator by Root Locus technique

## Experiment: 7

# Time Delay or Transportation Lag

**Scilab code Solution 7.7** To Study the Effect of Time Delay on System Response

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Lab 7: To Study the Effect of Time Delay on
  System Response
4 clc ;
5 clear all;
6
7 // For T = 0, i.e. No Delay
8 s=poly(0, 's')
9 // Transfer Function of Open-Loop System with T=0
10 G1=syslin('c',exp(0)*10/((s)*(s+1)));
11 figure(0)
12 bode(G1,0.1,100)
13 [gm1,frg1]=g_margin(G1);
14 disp(gm1,"Gain Margin,T=0",frg1,"Phase Cut-off
  Frequency, T=0")
15 show_margins(G1)
16 [phm1,frp1]=p_margin(G1);
```

```

17 disp(phm1,"Phase Margin , T=0",frp1,"Gain Cut-off
    Frequency , T=0")
18 show_margins(G1)
19
20 // For Time Delay T = 0.1
21 T=0.1;
22 nTd=[1-T*s/2]; dTd=[1+T*s/2]; // Num and Den of Delay
    System by Pade
23 sysTd=syslin('c',nTd/dTd);
24 G2=syslin('c',(sysTd*10)/((s)*(s+1)));
25 figure(1)
26 bode(G2,0.1,100)
27 [gm2,frg2]=g_margin(G2);
28 disp(gm2,"Gain Margin , T=0.1",frg2,"Phase Cut-off
    Frequency , T=0.1")
29 show_margins(G2)
30 [phm2,frp2]=p_margin(G2);
31 disp(phm2,"Phase Margin , T=0.1",frp2,"Gain Cut-off
    Frequency , T=0.1")
32 show_margins(G2)
33
34 // Results
35 // Frequency Response Specifications of Un-
    Compensated System
36 //Phase Cut-off Frequency , T=0
37 //
38 //      []
39 //
40 // Gain Margin ,T=0
41 //
42 //      Inf
43 //
44 // Gain Cut-off Frequency , T=0
45 //
46 //      0.4908709
47 //
48 // Phase Margin , T=0
49

```



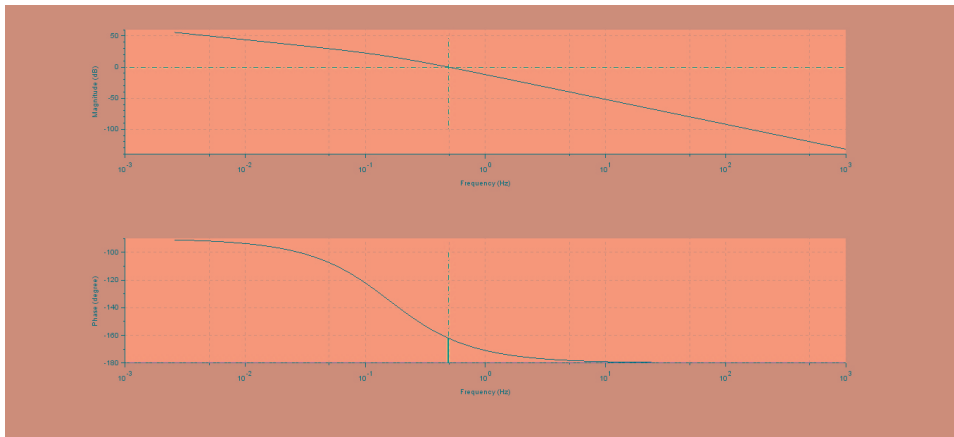


Figure 7.1: To Study the Effect of Time Delay on System Response

```

50 //      17.964236
51 // Frequency Response Specifications of Compensated
    System
52
53 // Phase Cut-off Frequency , T=0.1
54 //
55 //      0.4971165
56 //
57 // Gain Margin , T=0.1
58 //
59 //      0.2093087
60 //
61 // Gain Cut-off Frequency , T=0.1
62 //
63 //      0.4908709
64 //
65 // Phase Margin , T=0.1
66 //
67 //      0.4310002

```

---

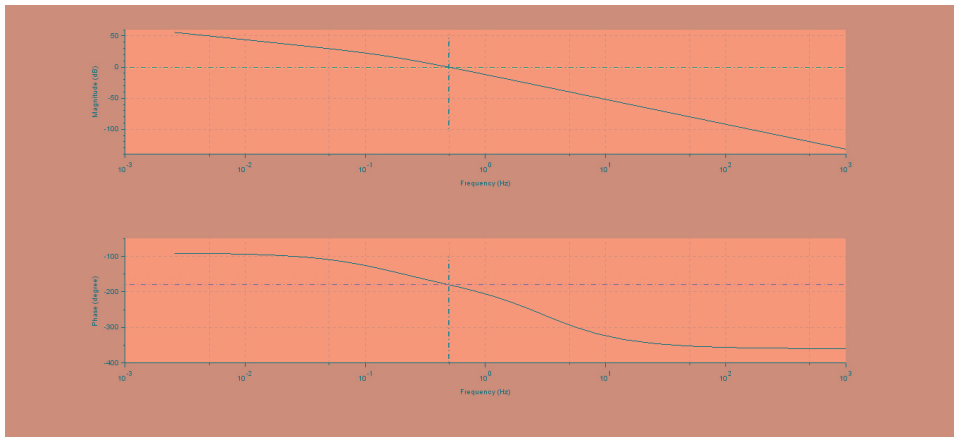


Figure 7.2: To Study the Effect of Time Delay on System Response

## Experiment: 8

# Lead Compensator Design by Bode Plot Technique

**Scilab code Solution 8.8** To design a Lead Comensator by Bode plot technique

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Expt. No. 8: To design a Lead Comensator by Bode-
  plot technique
4
5 clc
6 close
7 clf();
8
9 s=poly(0, 's')
10 // Transfer Function of Open-Loop System
11 G=syslin('c',12/((s)*(s+1)))
12 figure(0)
13 bode(G,0.1,100)
14 [gm,frg]=g_margin(G);
15 disp(gm,"Gain Margin",frg,"Phase Cut-off Frequency")
16 show_margins(G)
17 [phm,frp]=p_margin(G);
```

```

18 disp(phm,"Phase Margin",frp,"Gain Cut-off Frequency"
    )
19 show_margins(G)
20
21 // Transfer Function of Open-Loop Compensated System
    , GGc
22 GGc=syslin('c',12*(1+0.385*s)/((s)*(s+1)*(1+0.125*s)
    ))
23
24 // Bode - Plot of Compensated System
25 figure(1)
26 bode(GGc,0.1,100)
27 [gm,frg]=g_margin(GGc);
28 disp(gm,"Gain Margin",frg,"Phase Cut-off Frequency")
29 show_margins(GGc)
30 [phm,frp]=p_margin(GGc);
31 disp(phm,"Phase Margin",frp,"Gain Cut-off Frequency"
    )
32 show_margins(GGc)
33
34 // Step Response of Coles-Loop Un-Compensated System
35 figure(2)
36 t= 0:0.1:10;
37 x=[csim('step',t,G/(1+G))];
38 plot2d(t',x),xgrid(5, 1, 7)
39 xlabel("t", "fontsize", 2,"color", "blue");
40 ylabel("Amplitude", "fontsize", 2, "color", "blue");
41 xtitle('Step Response of Un-Compensated System');
42
43 // Step Response of Close-Loop Compensated System
44 figure(3)
45 t= 0:0.1:10;
46 x=[csim('step',t,GGc/(1+GGc))];
47 plot2d(t',x),xgrid(5, 1, 7)
48 xlabel("t", "fontsize", 2,"color", "blue");
49 ylabel("Amplitude", "fontsize", 2, "color", "blue");
50 xtitle('Step Response of Compensated System');
51 // Results

```

```

52 // Frequency Response Specifications of Un-
    Compensated System
53 //Phase Cut-off Frequency
54 //
55 //      []
56 //
57 // Gain Margin
58 //
59 //      Inf
60 //
61 // Gain Cut-off Frequency
62 //
63 //      0.5399649
64 //
65 // Phase Margin
66 //
67 //      16.422908
68
69 // Frequency Response Specifications of Compensated
    System
70
71 // Phase Cut-off Frequency
72 //
73 //      []
74 //
75 // Gain Margin
76 //
77 //      Inf
78 //
79 // Gain Cut-off Frequency
80 //
81 //      0.720447
82 //
83 // Phase Margin
84 //
85 //      43.107374

```

---

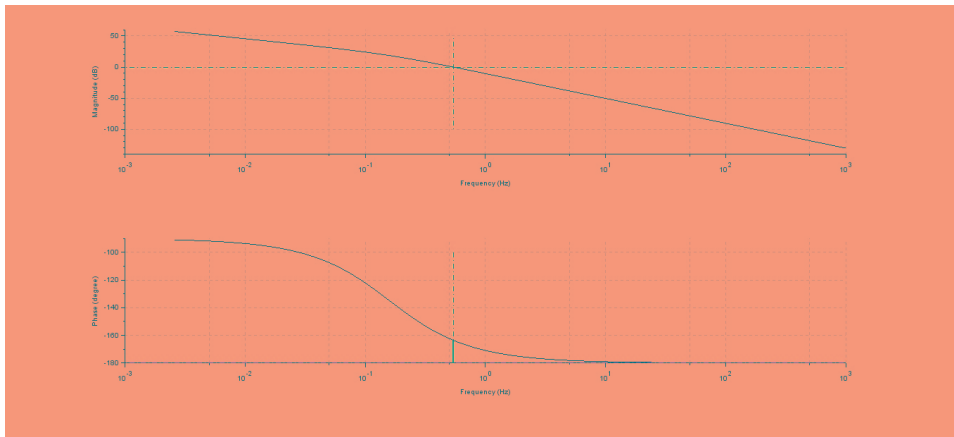


Figure 8.1: To design a Lead Comensator by Bode plot technique

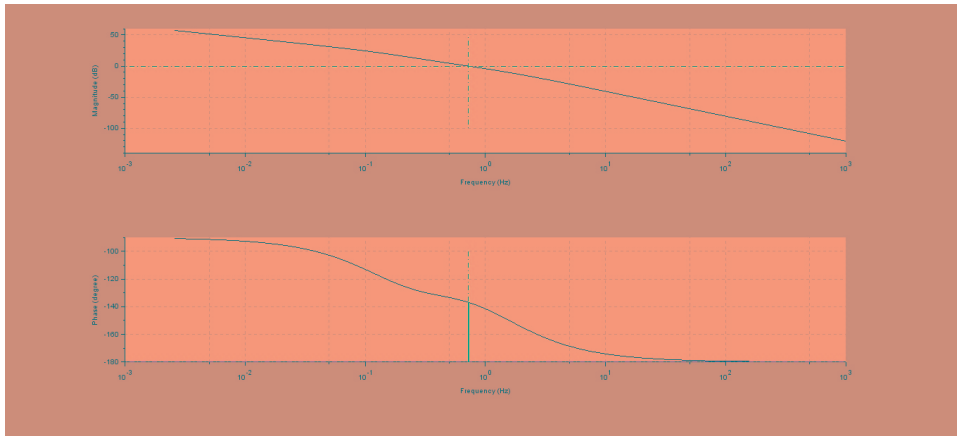


Figure 8.2: To design a Lead Comensator by Bode plot technique

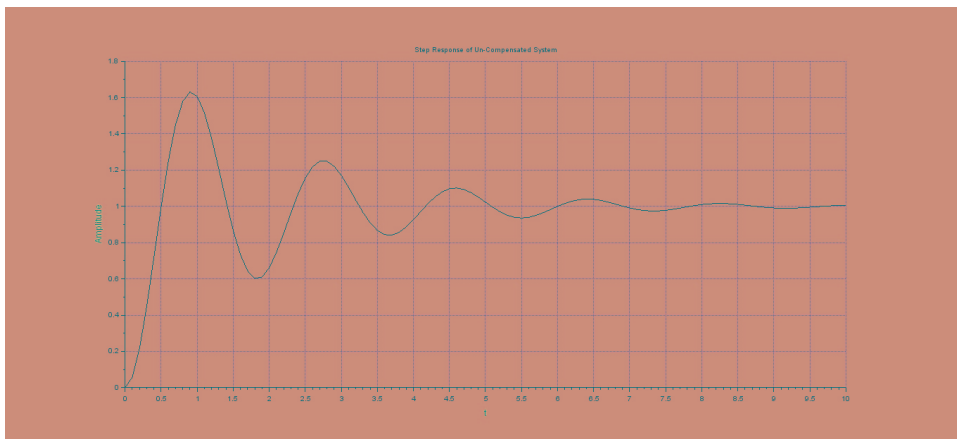


Figure 8.3: To design a Lead Comensator by Bode plot technique

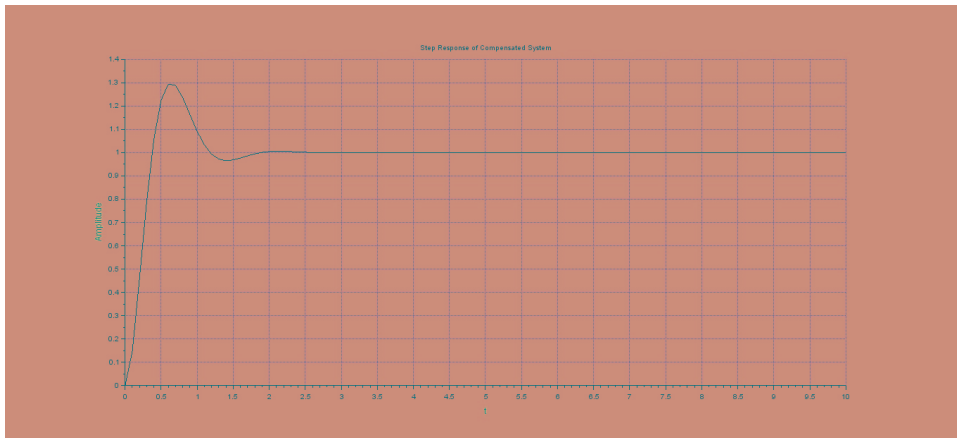


Figure 8.4: To design a Lead Comensator by Bode plot technique



## Experiment: 9

# Lag Compensator Design by Bode Plot Technique

**Scilab code Solution 9.9** To design a Lag Compensator by Bode plot technique

```
1 // Scilab : 6.0.0
2 // OS: Windows 7, 64 bit
3 // Expt. No. 9: To design a Lag Compensator by Bode-
  plot technique
4
5 clc
6 close
7 clf();
8
9 s=poly(0, 's')
10 // Transfer Function of Open-Loop System
11 G=syslin('c',30/((s)*(0.1*s+1)*(0.2*s+1)))
12 figure(0)
13 // Bode - Plot of Un-Compensated System
14 bode(G,0.1,100)
15
16 [gm,frg]=g_margin(G);
17 disp(gm,"Gain Margin",frg,"Phase Cut-off Frequency")
```

```

18 show_margins(G)
19 [phm,frp]=p_margin(G);
20 disp(phm,"Phase Margin",frp,"Gain Cut-off Frequency"
    )
21 show_margins(G)
22
23 // Transfer Function of Open-Loop Compensated System
    , GGc
24 GGc=syslin('c',30*(1+3.33*s)/((s)*(0.1*s+1)*(0.2*s
    +1)*(1+33.3*s)))
25
26 // Bode - Plot of Compensated System
27 figure(1)
28 bode(GGc,0.1,100)
29
30 [gm,frg]=g_margin(GGc);
31 disp(gm,"Gain Margin",frg,"Phase Cut-off Frequency")
32 show_margins(GGc)
33 [phm,frp]=p_margin(GGc);
34 disp(phm,"Phase Margin",frp,"Gain Cut-off Frequency"
    )
35 show_margins(GGc)
36
37 // Step Response of Closed-Loop Un-Compensated
    System
38 figure(2)
39 t= 0:0.1:10;
40 x=[csim('step',t,G/(1+G))];
41 plot2d(t',x),xgrid(5, 1, 7)
42 xlabel("t", "fontsize", 2,"color", "blue");
43 ylabel("Amplitude", "fontsize", 2, "color", "blue");
44 xtitle('Step Response of Un-Compensated System');
45
46 // Step Response of Closed-Loop Compensated System
47 figure(3)
48 t= 0:0.1:10;
49 x=[csim('step',t,GGc/(1+GGc))];
50 plot2d(t',x),xgrid(5, 1, 7)

```

```

51 xlabel("t", "fontsize", 2, "color", "blue");
52 ylabel("Amplitude", "fontsize", 2, "color", "blue");
53 xtitle( 'Step Response of Compensated System' );
54
55 // Result
56 // Frequency Response Specifications of Un-
    Compensated System
57 //Phase Cut-off Frequency
58 //
59 //     1.1253954
60 //
61 // Gain Margin
62 //
63 //     -6.0205999
64 //
65 // Gain Cut-off Frequency
66 //
67 //     1.555363
68 //
69 // Phase Margin
70 //
71 //     -17.245408
72 // Frequency Response Specifications of Compensated
    System
73
74 // Phase Cut-off Frequency
75 //
76 //     1.0788163
77 //
78 // Gain Margin
79 //
80 //     13.24356
81 //
82 // Gain Cut-off Frequency
83 //
84 //     0.4129329
85 //
86 // Phase Margin

```

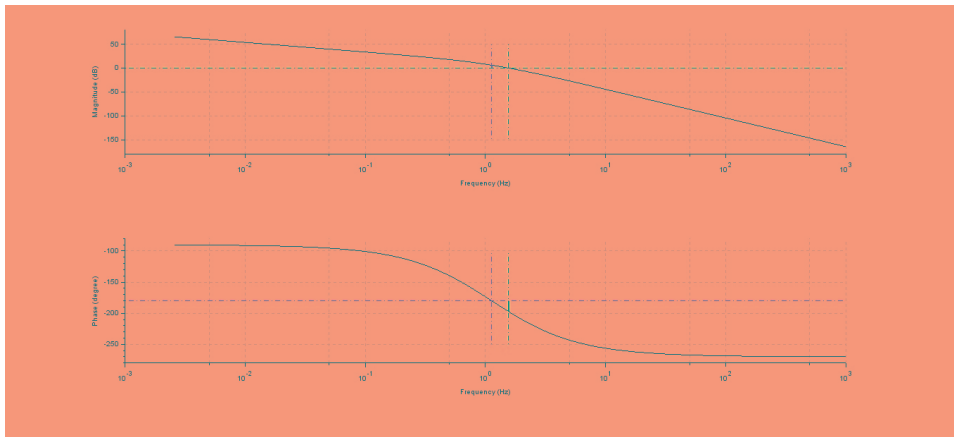


Figure 9.1: To design a Lag Compensator by Bode plot technique

87 //  
 88 // 42.090918  
 89 //

---

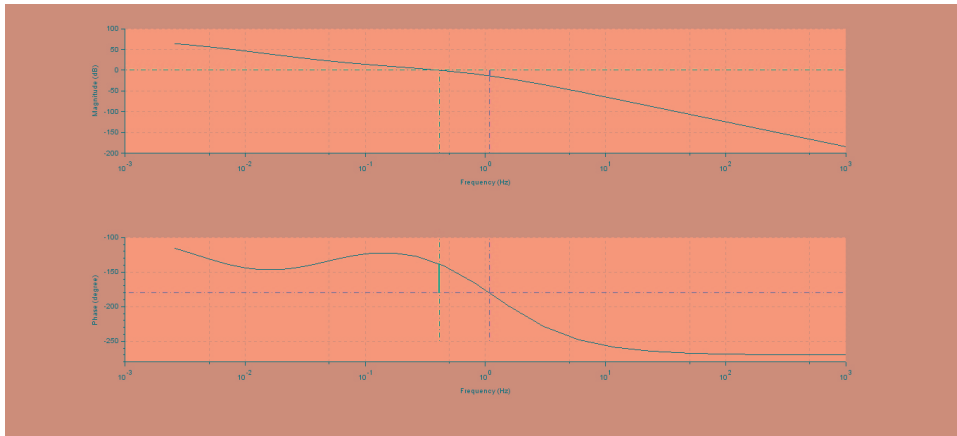


Figure 9.2: To design a Lag Compensator by Bode plot technique

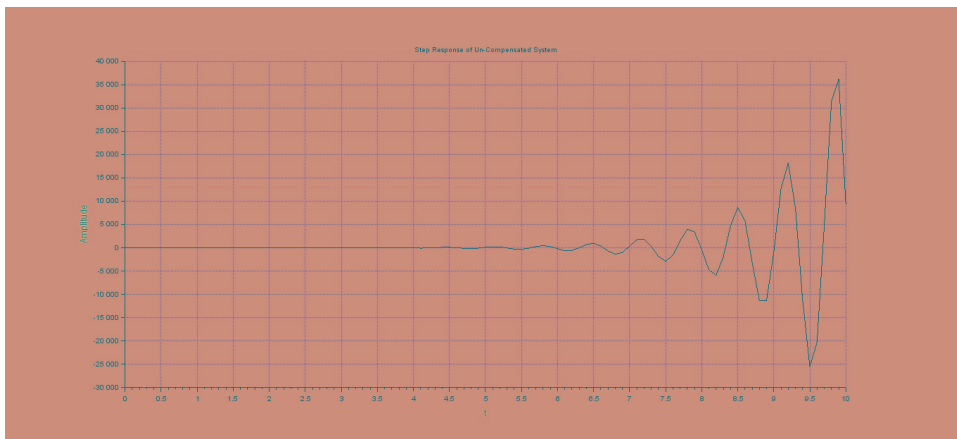


Figure 9.3: To design a Lag Compensator by Bode plot technique

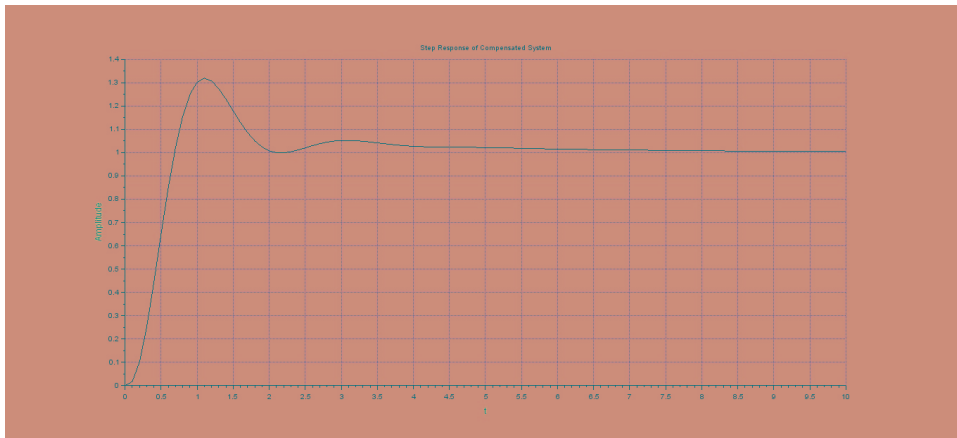


Figure 9.4: To design a Lag Compensator by Bode plot technique

# **Experiment: 10**

## **PID Controller Design**

This code can be downloaded from the website [www.scilab.in](http://www.scilab.in)

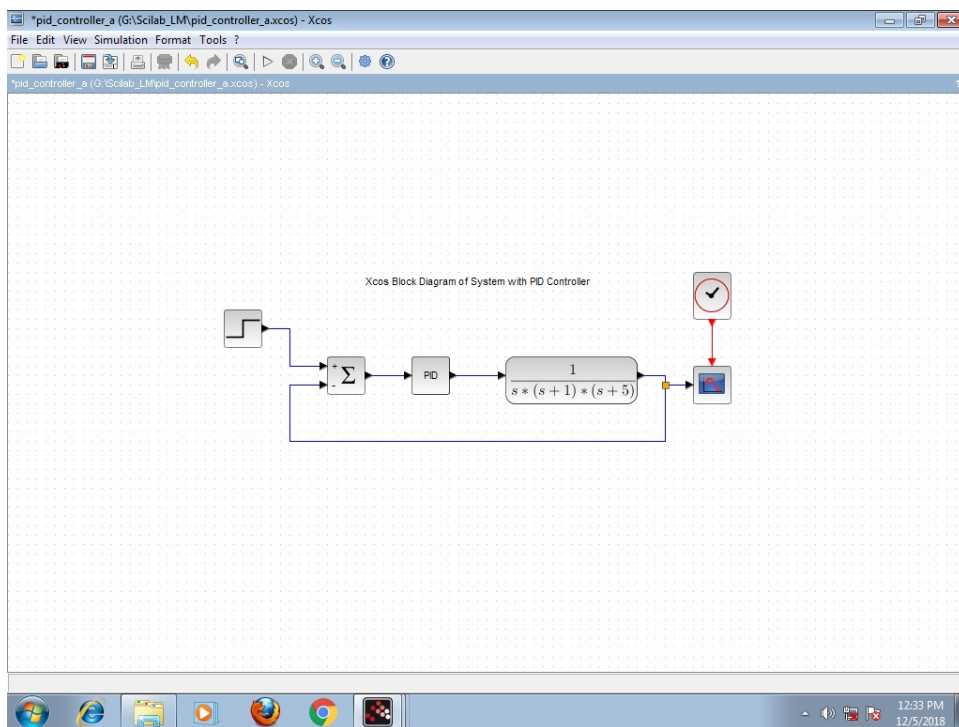


Figure 10.1: To design a PID Controller by using Ziegler Nichols tuning rules



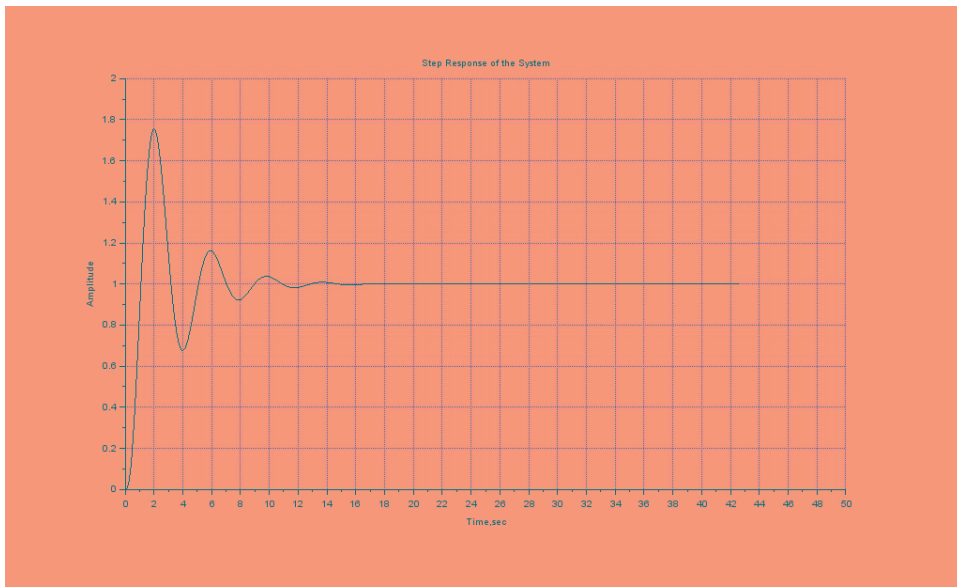


Figure 10.2: To design a PID Controller by using Ziegler Nichols tuning rules

## **Experiment: 11**

# **Servomechanism or Tracking Problem**

This code can be downloaded from the website [www.scilab.in](http://www.scilab.in)

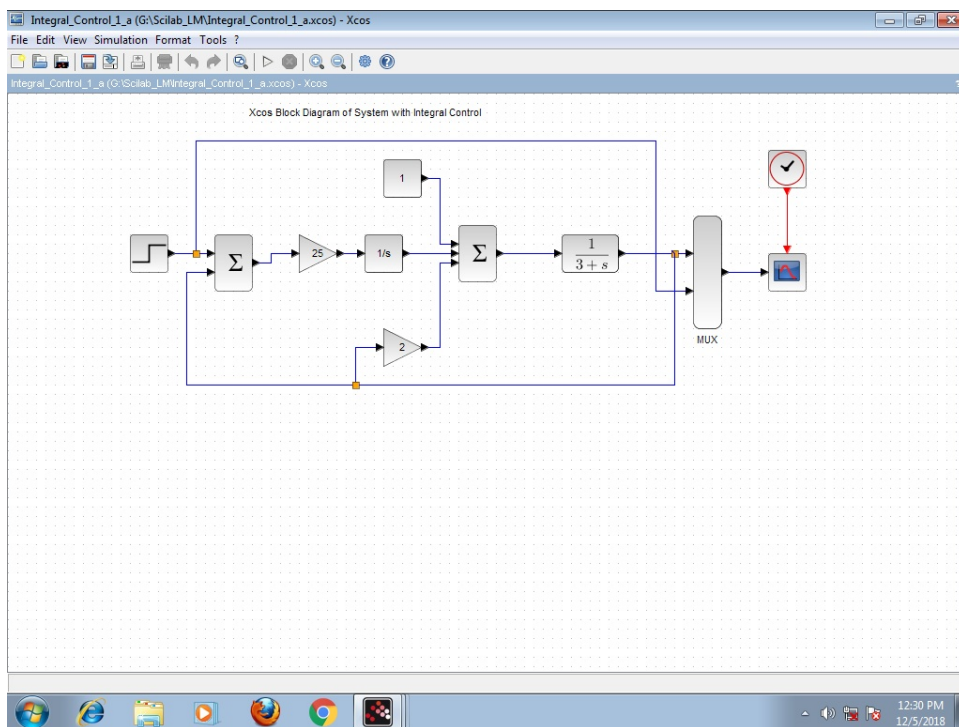


Figure 11.1: To apply integral control to servo problem to minimize the error

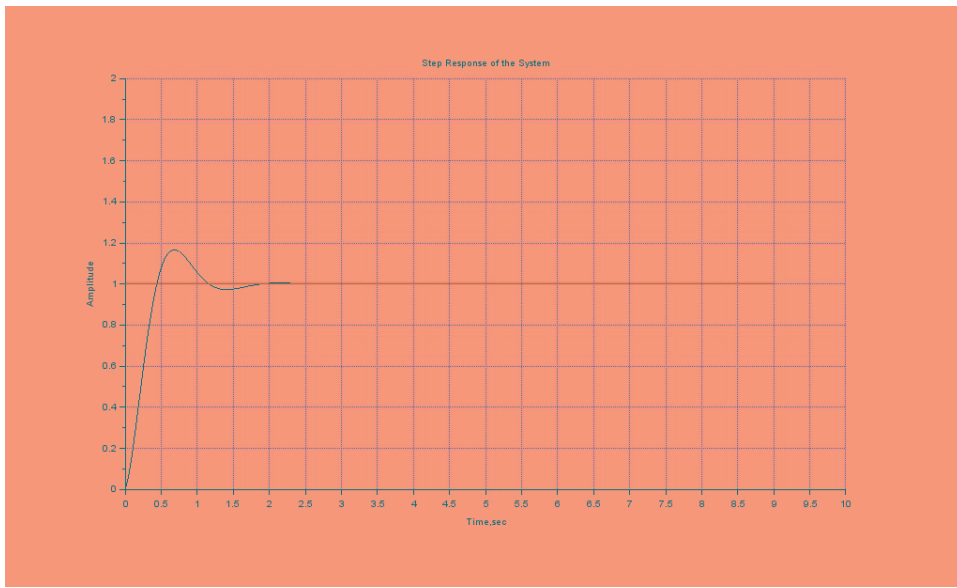


Figure 11.2: To apply integral control to servo problem to minimize the error