

Scilab Manual for  
Feedback Control Lab Practices  
by Prof Deepti Khimani  
Instrumentation Engineering  
Vivekanand Education Society'S Institute Of  
Technology<sup>1</sup>

Solutions provided by  
Prof Deepti Khimani  
Instrumentation Engineering  
Vivekanand Education Society'S Institute Of Technology

March 10, 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 Observe the effect of change in time constant of the first-order systems.	5
2 Plot the time response of second-order system with real poles and compare it with responses of it's individual poles.	8
3 Plot the time-responses of the second-order system with various damping factors.	12
4 Examine the steady state errors for Type 0 system.	16
5 Examine the steady state errors for Type 1 system.	20
6 Examine the steady state errors for Type 2 system.	24
7 Inspect the relative stability of systems by Root-Locus.	27
8 Inspect the stability of systems by Bode plot.	32
9 Determine the frequency response specifications from Nyquist plot of system.	35

# List of Experiments

Solution 1.1	Effect of time constant . . . . .	5
Solution 2.1	Time response of SOS . . . . .	8
Solution 3.1	Time response of SOS . . . . .	12
Solution 4.1	Steady state error analysis . . . . .	16
Solution 5.1	Steady state error analysis . . . . .	20
Solution 6.1	Steady state error analysis . . . . .	24
Solution 7.1	Stability analysis . . . . .	27
Solution 8.1	Stability analysis . . . . .	32
Solution 9.1	Nyquist plot . . . . .	35

# List of Figures

1.1	Effect of time constant . . . . .	7
2.1	Time response of SOS . . . . .	11
3.1	Time response of SOS . . . . .	15
4.1	Steady state error analysis . . . . .	19
5.1	Steady state error analysis . . . . .	23
6.1	Steady state error analysis . . . . .	26
7.1	Stability analysis . . . . .	30
7.2	Stability analysis . . . . .	31
8.1	Stability analysis . . . . .	34
9.1	Nyquist plot . . . . .	37

# Experiment: 1

Observe the effect of change in time constant of the first-order systems.

Scilab code Solution 1.1 Effect of time constant

```
1 //Lab 1 : Observe the effect of change in time
   constant of the first-order systems.
2 //


---


3 // Problem Statement: Compare step response of
4 //  $G_1=1/(2s+1)$ ,  $G_2=1/(s+1)$  and
5 //  $G_3=1/(4s+1)$ 
6 //


---


7 // Operating System OSX (Mac) 10.14.6
8 // Scilab Version 6.1.0
9 //
```

10

```

11 xdel(winsid()) // close the figure windows
12 clear; // clear the workspace
13 clc; // clear the console
14 //

```

---

```

15 // system representation
16
17 s=%s //Laplace operator
18
19 //system with various time constants.
20
21 G1=syslin('c',1/(2*s+1)) //system 1 represented
    by G1(s)=1/(2s+1)
22 G2=syslin('c',1/(s+1)) //system 2 represented
    by G2(s)=1/(s+1)
23 G3=syslin('c',1/(4*s+1)) //system 3 represented
    by G3(s)=1/(4s+1)
24
25 //

```

---

```

26 // Unit step response
27 u=1
28 t=0:0.2:15 // Simulation time
29 c1=csim('step',t,G1*u) // Response of system G1(s)
    =1/(2s+1)
30 c2=csim('step',t,G2*u) // Response of system G2(s)
    =1/(s+1)
31 c3=csim('step',t,G3*u) // Response of system G3(s)
    =1/(4s+1)
32
33 //plots
34
35 plot(t,c1,t,c2,'r-',t,c3,'LineWidth',2)
36
37 // plot of response value at one time-constant
38 plot(2,0.6321,'o',1,0.6321,'ro',4,0.6321,'o',')

```

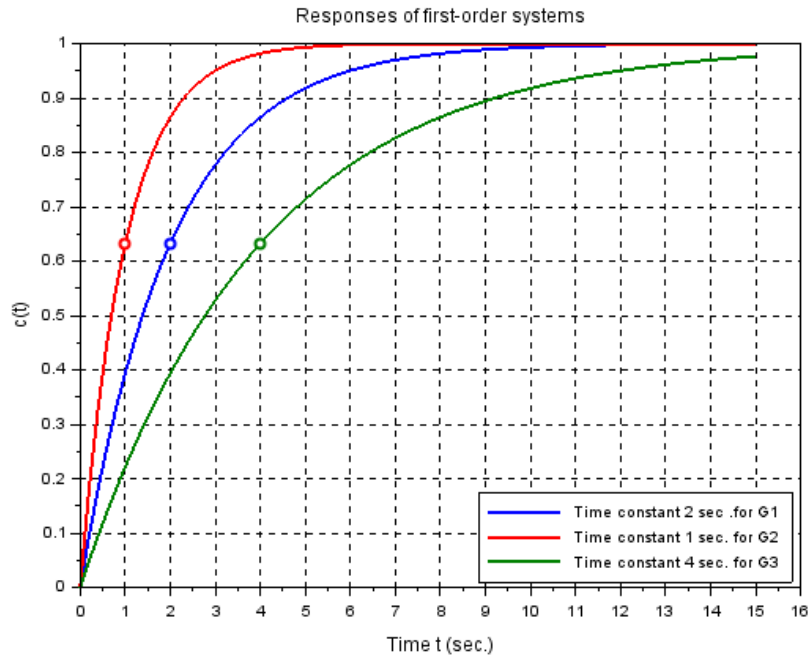


Figure 1.1: Effect of time constant

```

    LineWidth',2)
39 plot(t,0.6321,'—','LineWidth',1.5)
40 xgrid
41
42 //Title, labels and grid to the figure
43 legend('Time constant 2 sec .for G1','Time constant
    1 sec. for G2','Time constant 4 sec. for G3',4)
44 title('Responses of first-order systems','fontsize'
    ,2)
45 xlabel('Time t (sec.)','fontsize',2)
46 ylabel('c(t)','fontsize',2)

```

---



## Experiment: 2

**Plot the time response of second-order system with real poles and compare it with responses of it's individual poles.**

Scilab code Solution 2.1 Time response of SOS

```
1 //Lab 2 : Plot the time response of second-order
   system with real poles
2 // and compare it with responses of it's individual
   poles.
3 //


---


4 // Problem Statement: Plot the time response of
5 //G=1/(2s+1)/(3s+1) and comapare it with responses
   of
6 // G1=1/(2s+1) and G2=1/(3s+1)
7 //
```

---

```

8
9 //


---


10 // Operating System OSX (Mac) 10.14.6
11 // Scilab Version 6.1.0
12 //


---


13
14 xdel(winsid()) // close the figure windows
15 clear; // clear the workspace
16 clc; // clear the console
17 //


---


18 // system representation
19
20 s=%s //Laplace operator
21
22 //system with various time constants.
23
24 G=syslin('c',1/(2*s+1)/(3*s+1)) // representation
    of original system  $G(s)=1/(2s+1)/(3s+1)$ 
25
26 G1=syslin('c',1/(2*s+1)) //  $G1(s)=1/(2s+1)$ 
27 G2=syslin('c',1/(3*s+1)) //  $G2(s)=1/(3s+1)$ 
28
29 //


---


30 // Unit step response
31
32 t=0:0.2:30 // Simulation time
33 dim=size(t);
34 u=ones(dim(1),dim(2)) // unit step input
35

```

```
36 c=csim(u,t,G) // Response of system G
37
38 c1=csim(u,t,G1) // Response of system G1
39 c2=csim(u,t,G2) // Response of system G2
40
41 //plot
42 plot(t,c,t,c1,t,c2,'LineWidth',2)
43
44 xgrid
45
46 //Title, labels and grid to the figure
47 legend('System G (SO)', 'System G1(FO)', 'System G2(
    FO)',4)
48 title('Responses of second-order system and its
    first-order poles', 'fontsize',2)
49 xlabel('Time t (sec.)', 'fontsize',2)
50 ylabel('c(t)', 'fontsize',2)
```

---

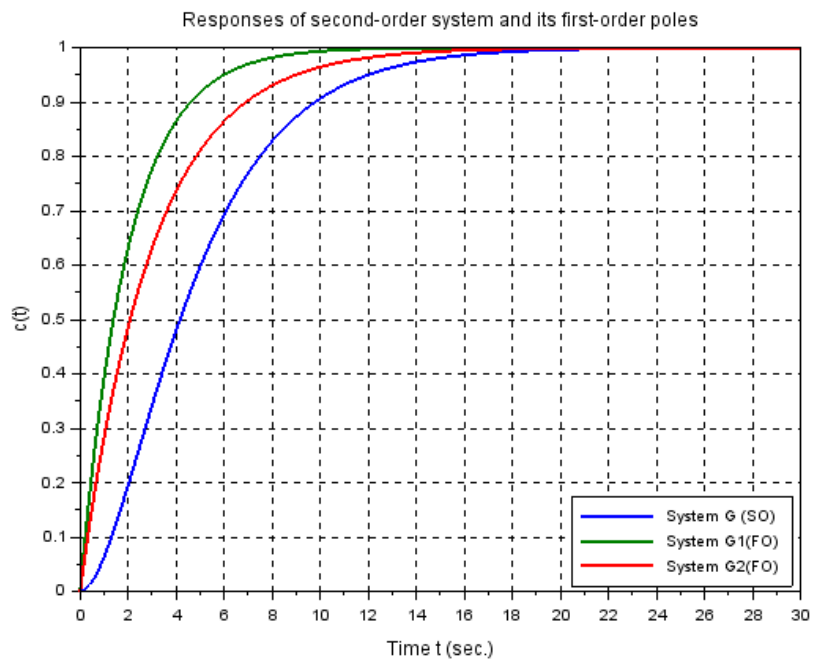


Figure 2.1: Time response of SOS

## Experiment: 3

Plot the time-responses of the second-order system with various damping factors.

Scilab code Solution 3.1 Time response of SOS

```
1 //Lab 3 : Plot the time-responses of the second-  
  order systems with  
2 //various damping factors.  
3 //  


---

4 // Problem Statement: Compare Step response of  $G=1/(s^2+2*\xi*wn*s+wn^2)$   
5 //with natural frequency  $wn=1$  rad/sec and  $\xi=0.3$  (  
  underdamped),  
6 // $\xi=1.5$  (overdamped) and  $\xi=0$  (undamped).  
7 //  


---

8  
9 //
```

---

```

10 // Operating System OSX (Mac) 10.14.6
11 // Scilab Version 6.1.0
12 //


---


13 xdel(winsid()) // close the figure windows
14 clear; // clear the workspace
15 clc; // clear the console
16 //


---


17 // system representation
18 s=%s // Laplace operator
19 s=poly(0,'s');
20 wn=1 // Natural frequency wn=1 rad/sec.
21
22 // case 1 : underdamped system with xi=0.3
23 xi=0.3
24 G1=syslin('c',1/(s^2+2*xi*wn*s+wn^2)) //G1(s)=1/(s
    ^2+0.3s+1)
25
26 // case 2 : overdamped system with xi=1.5
27 xi=1.5
28 G2=syslin('c',1/(s^2+2*xi*wn*s+wn^2)) //G2(s)=1/(s
    ^2+1.5s+1)
29
30 // case 3 : undamped system with xi=0
31 xi=0
32 G3=syslin('c',1/(s^2+2*xi*wn*s+wn^2)) //G3(s)=1/(s
    ^2+1)
33 //


---


34 // Unit step response
35 u=1
36 t=0:0.1:20 // Simulation time
37 c1=csim('step',t,G1*u) // Response of system G1

```

```

38 c2=csim('step',t,G2*u) // Response of system G2
39 c3=csim('step',t,G3*u) // Response of system G3
40
41 //plots
42
43 plot(t,c1,t,c2,'r-',t,c3,'LineWidth',2) //Responses
44 plot(t,u,'—','LineWidth',1) // input signal
45
46 xgrid
47
48 //Title, labels and grid to the figure
49 legend('Underdamped System','overdamped System','
        Undamped System',4)
50 title('Responses of second-order systems','fontsize
        ',3)
51 xlabel('Time t (sec.)','fontsize',2)
52 ylabel('c(t)','fontsize',2)

```

---

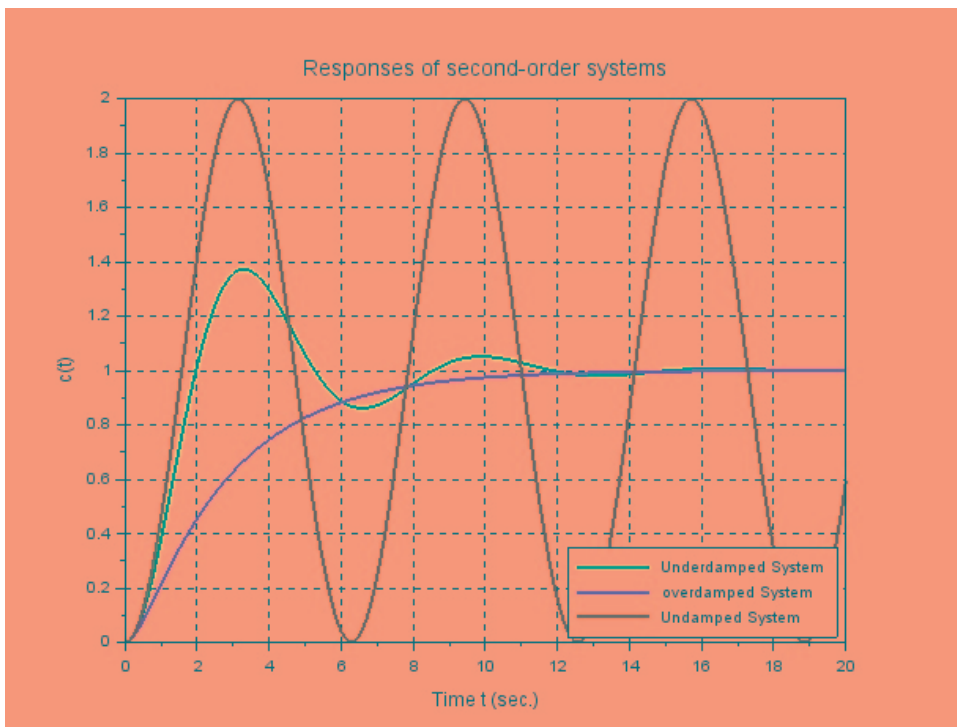


Figure 3.1: Time response of SOS



# Experiment: 4

## Examine the steady state errors for Type 0 system.

Scilab code Solution 4.1 Steady state error analysis

```
1 //Lab 4 : Examine the steady state errors for Type
  0 system.
2 //


---


3 // (a) Problem Statement: Examine the steady state
  errors in step ,
4 // ramp and parabolic response of Type 0 system G
  =0.5/(s+0.8).
5 //


---


6
7 //


---


8 // Operating System OSX (Mac) 10.14.6
9 // Scilab Version 6.1.0
10 //
```

---

```

11 xdel(winsid()) // close the figure windows
12 clear; // clear the workspace
13 clc; // clear the console
14 //

```

---

```

15 // system representation
16
17 s=%s //Laplace operator
18
19 //Type 0 system G=0.5/(s+0.8)
20
21 G=syslin('c',0.5/(s+0.8))
22
23 Gc1=G/(1+G) //closed loop Gc1=0.5/(s+1.3)
24
25 //

```

---

```

26
27 t=0:0.2:12 // Simulation time
28 dim=size(t);
29
30 u1=ones(dim(1),dim(2)) // unit step input
31 u2=t // unit ramp;
32 u3=t^2/2 // unit parabolic input
33
34 // Unit step response
35
36 c1=csim(u1,t,Gc1) // unit step response of closed
    loop system
37 c2=csim(u2,t,Gc1) // unit rmp response of closed
    loop system
38 c3=csim(u3,t,Gc1) // unit parabolic response closed
    loop system
39

```

```
40 plot(t,c1,t,c2,t,c3,t,'LineWidth',2)
41 plot(t,u1,'—',t,u2,'—',t,u3,'—','LineWidth',1.5)
42
43 zoom_rect([0,0,8,1.5]) //zoom the graphics window for
    clarity
44
45 xgrid
46
47 //Title, labels and grid to the figure
48 legend('unit step response','unit ramp response','
    unit parabolic response',4)
49 title('Responses of Type 0 system (dashed lines are
    inputs)', 'fontsize',2)
50 xlabel('Time t (sec.)', 'fontsize',2)
51 ylabel('c(t)', 'fontsize',2)
```

---

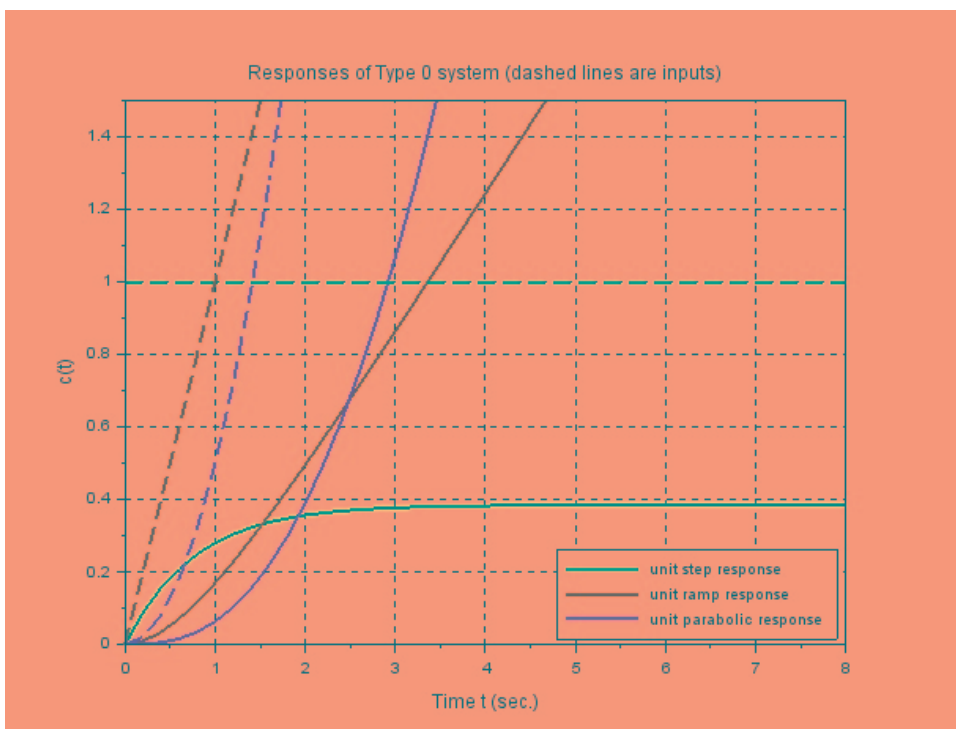


Figure 4.1: Steady state error analysis

# Experiment: 5

## Examine the steady state errors for Type 1 system.

Scilab code Solution 5.1 Steady state error analysis

```
1 //Lab 5 : Examine the steady state errors for Type
  1 system.
2 //


---


3 //Problem Statement: Examine the steady state errors
  in step ,
4 // ramp and parabolic response of Type 1 system G
  =0.5/s/(s+0.8)
5 //


---


6 //


---


7 // Operating System OSX (Mac) 10.14.6
8 // Scilab Version 6.1.0
9 //
```

---

```

10 xdel(winsid()) // close the figure windows
11 clear; // clear the workspace
12 clc; // clear the console
13 //

```

---

```

14 // system representation
15
16 s=%s //Laplace operator
17
18 //Type 1 system  $G=0.5/s/(s+0.8)$ 
19
20 G=syslin('c',0.5/s/(s+0.8))
21 Gcl=G/(1+G) //closed loop  $Gcl=0.5/(s^2+0.8s+0.5)$ 
22
23 //

```

---

```

24
25 t=0:0.2:12 // Simulation time
26 dim=size(t);
27
28 u1=ones(dim(1),dim(2)) // unit step input
29 u2=t // unit ramp;
30 u3=t^2/2 // unit parabolic input
31
32 // Unit step response
33
34 c1=csim(u1,t,Gcl) // unit step response of closed
    loop system
35 c2=csim(u2,t,Gcl) // unit rmp response of closed
    loop system
36 c3=csim(u3,t,Gcl) // unit parabolic response closed
    loop system
37
38 plot(t,c1,t,c2,t,c3,t,'LineWidth',2)
39 plot(t,u1,'—',t,u2,'—',t,u3,'—','LineWidth',1.5)

```

```
40
41 zoom_rect([0,0,8,20]) //zoom the graphics window for
    clarity
42
43 xgrid
44
45 //Title, labels and grid to the figure
46 legend('unit step response','unit ramp response', '
    unit parabolic response',2)
47 title('Responses of Type 1 system (dashed lines are
    inputs)', 'fontsize',2)
48 xlabel('Time t (sec.)', 'fontsize',2)
49 ylabel('c(t)', 'fontsize',2)
```

---

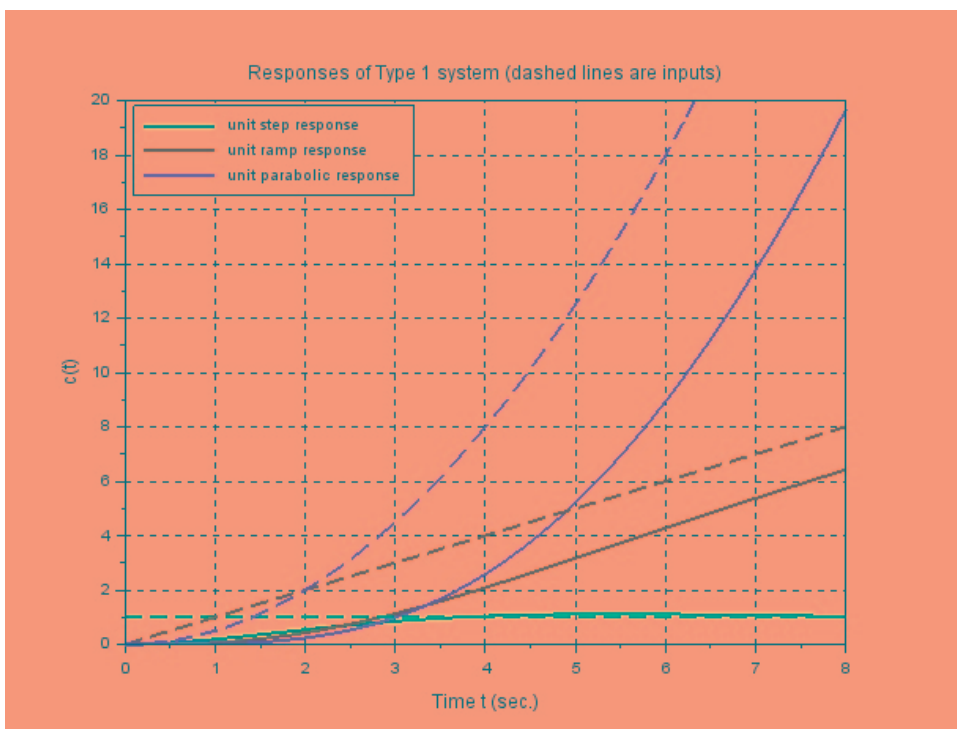


Figure 5.1: Steady state error analysis



# Experiment: 6

## Examine the steady state errors for Type 2 system.

Scilab code Solution 6.1 Steady state error analysis

```
1 //Lab 6 : Examine the steady state errors for Type
  2 system.
2 //


---


3 //Problem Statement: Examine the steady state errors
  4 in step ,
  5 //ramp and parabolic response of Type 2 system  $G=(s$ 
  6  $+0.5)/s^2/(s+0.8)$ 
  7 //


---


8 // Operating System OSX (Mac) 10.14.6
9 // Scilab Version 6.1.0
10 //
```

---

```

11 xdel(winsid()) // close the figure windows
12 clear; // clear the workspace
13 clc; // clear the console
14 //

15 // system representation
16
17 s=%s //Laplace operator
18
19 //Type 1 system  $G=(s+0.5)/s^2/(s+0.8)$ 
20
21 G=syslin('c',(s+0.5)/s^2/(s+0.8))
22 Gc1=G/(1+G) //closed loop  $Gc1=(s+0.5)/(s^3+0.8s^2+s$ 
    +0.5)
23
24 //

25
26 t=0:0.2:10 // Simulation time
27 dim=size(t);
28
29 u1=ones(dim(1),dim(2)) // unit step input
30 u2=t // unit ramp;
31 u3=t^2/2 // unit parabolic input
32
33 // Unit step response
34
35 c1=csim(u1,t,Gc1) // unit step response of closed
    loop system
36 c2=csim(u2,t,Gc1) // unit rmp response of closed
    loop system
37 c3=csim(u3,t,Gc1) // unit parabolic response closed
    loop system
38

```

---

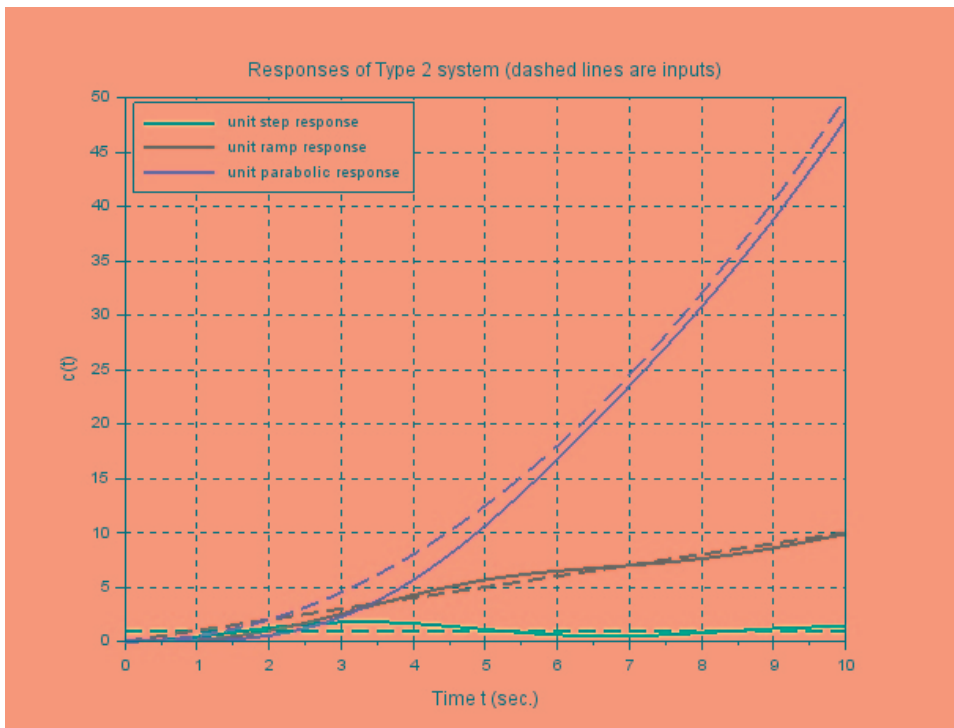


Figure 6.1: Steady state error analysis

```

39 plot(t,c1,t,c2,t,c3,t,'LineWidth',2)
40 plot(t,u1,'—',t,u2,'—',t,u3,'—','LineWidth',1.5)
41
42 xgrid
43
44 //Title, labels and grid to the figure
45 legend('unit step response','unit ramp response','
         unit parabolic response',2)
46 title('Responses of Type 2 system (dashed lines are
         inputs)','fontsize',2)
47 xlabel('Time t (sec.)','fontsize',2)
48 ylabel('c(t)','fontsize',2)

```

---

# Experiment: 7

## Inspect the relative stability of systems by Root-Locus.

Scilab code Solution 7.1 Stability analysis

```
1 //Lab 7 : Inspect the relative stability of systems
   by Root-Locus.
2 //
   _____

3 //Problem Statement: Draw the root locus of the
   system  $G=1/s/(s^2+1.5*s+1)$ .
4 //Observe settling time of of the system with
   different gains.
5 //
   _____

6
7 //
   _____

8 // Operating System OSX (Mac) 10.14.6
9 // Scilab Version 6.1.0
10 //
```

---

```

11 xdel(winsid()) // close the figure windows
12 clear; // clear the workspace
13 clc; // clear the console
14 //

15 // system representation
16 s=%s // Laplace operator
17 s=poly(0, 's');
18
19 G=syslin('c', 1/(s*(s^2+1.5*s+1))) // system
    representation G(s)=1/(s(s^2+1.5s+1))
20
21 //Compute marginal (critical) gain
22 //gain margin
23 gm= g_margin(G);
24 //gm=log10(kmar) ==> kmar=10^(gm/20)
25 kmar=10^(gm/20)
26
27 kmax=2 // maximum gain for plotting of root locus
28 evans(G, kmax)
29 xgrid
30 // for various k, system transfer functions are
31 k1=0.4
32 G1=k1*G/(1+k1*G) //G1=0.4/(s^3+1.5s^2+s+0.4)
33 [z1, p1, k1]=tf2zp(G1)
34 plot(real(p1), imag(p1), 'bx')
35
36 k2=0.8
37 G2=k2*G/(1+k2*G) //G2=0.8/(s^3+1.5s^2+s+0.8)
38
39 [z2, p2, k2]=tf2zp(G2)
40 plot(real(p2), imag(p2), 'gx')
41
42 k3=kmar //kmar=1.5
43 G3=k3*G/(1+k3*G) //G1=0.4/(s^3+1.5s^2+s+1.5)

```

```

44
45 [z3,p3,k3]=tf2zp(G3)
46 plot(real(p3),imag(p3),'cx')
47 //

```

---

```

48 // Unit step response
49 t=0:0.1:40 // Simulation time
50 dim=size(t);
51
52 u=ones(dim(1),dim(2)) // unit step input
53
54 c1=csim(u,t,G1) // Response of system G with k1
55 c2=csim(u,t,G2) // Response of system G2 with k2
56 c3=csim(u,t,G3) // Response of system G2 with k3
57
58 //plots
59 figure(1)
60 clf
61 plot(t,c1,t,c2,t,c3,'c:','LineWidth',2)
62 xgrid
63
64 //Title, labels and grid to the figure
65 legend('k=0.4','k=0.8','k=1.5',4)
66 title('Responses of system with different gains',
        'fontsize',3)
67 xlabel('Time t (sec.)','fontsize',2)
68 ylabel('c(t)','fontsize',2)

```

---

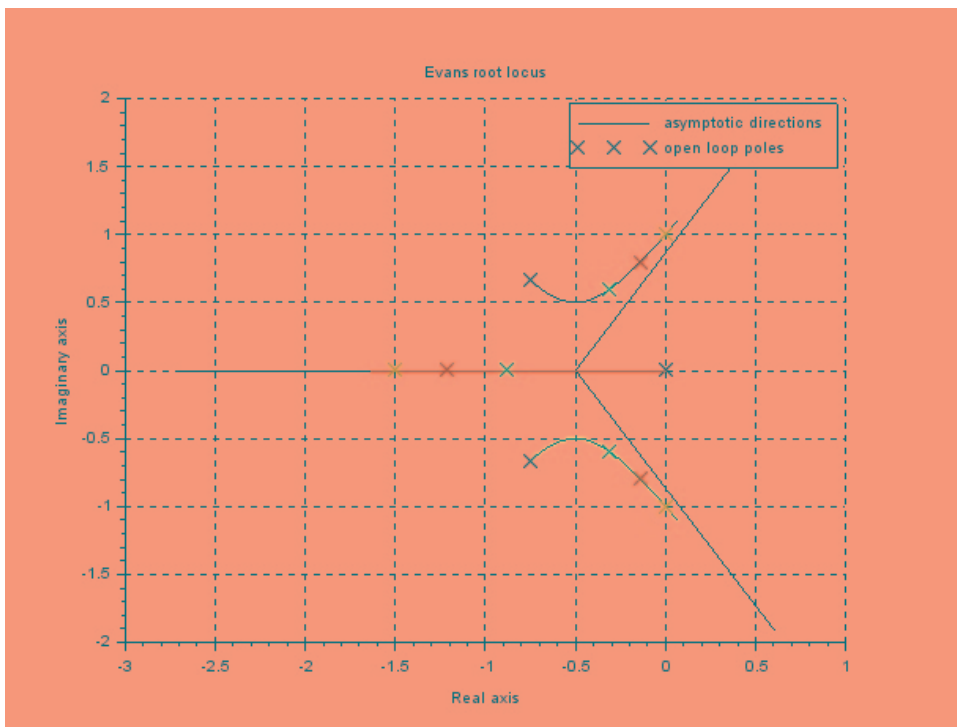


Figure 7.1: Stability analysis

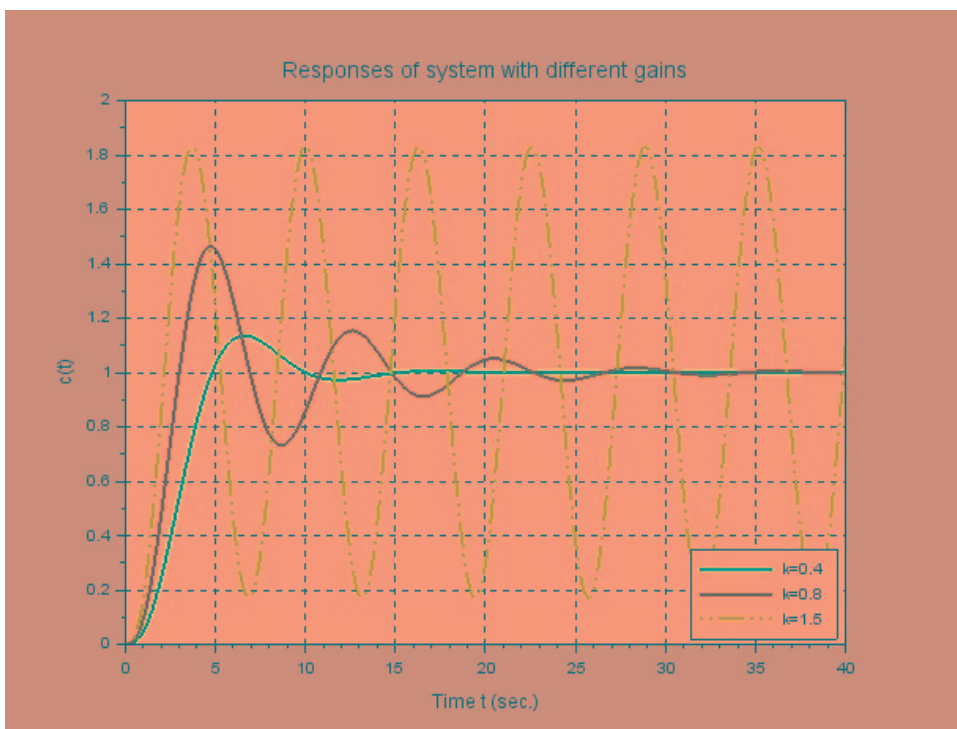


Figure 7.2: Stability analysis



# Experiment: 8

## Inspect the stability of systems by Bode plot.

Scilab code Solution 8.1 Stability analysis

```
1 //Lab 8 : Inspect the relative stability of systems
   from the Bode plot.
2 //
   _____

3 //Problem Statement: Draw the Bode of the system G
   =1/s/{s^2+1.8s+1}$.
4 //Determine the stability margins.
5 //
   _____

6
7 //
   _____

8 // Operating System OSX (Mac) 10.14.6
9 // Scilab Version 6.1.0
10 //
```

```
11 xdel(winsid()) // close the figure windows
12 clear; // clear the workspace
13 clc; // clear the console
14 //


---


15 // system representation
16 s=%s // Laplace operator
17 s=poly(0,'s');
18 G=syslin('c',1/s/(s^2+1.8*s+1))
19 //Bode plot with stability margins
20 show_margins(G,'bode')
21 [Gm,pcw] = g_margin(G)
22 [Pm,gcw] = p_margin(G)
23
24 disp("Gain Margin",Gm,"Phase crossover frequency",
      pcw)
25 disp("Phase Margin",Pm,"Gain crossover frequency",
      gcw)


---


```

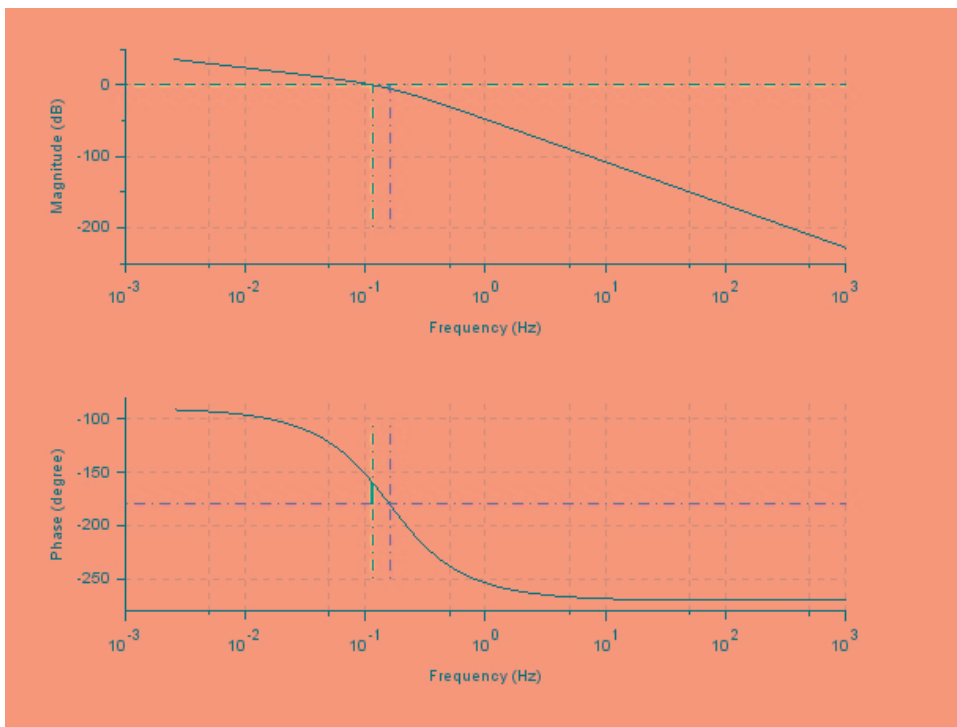


Figure 8.1: Stability analysis

## Experiment: 9

# Determine the frequency response specifications from Nyquist plot of system.

Scilab code Solution 9.1 Nyquist plot

```
1 //Lab 9 : Determine the frequency response
   specifications from Nyquist plot of system.
2 //
   _____

3 //Problem Statement: Draw the Nyquist plot of the
   system  $G=1/s/\{s^2+5s+1\}$ .
4 //Determine the stability margins.
5 //
   _____

6 //
   _____

7 // Operating System OSX (Mac) 10.14.6
8 // Scilab Version 6.1.0
9 //
```

---

```
10 xdel(winsid()) // close the figure windows
11 clear; // clear the workspace
12 clc; // clear the console
13 //

14 // system representation
15 s=%s // Laplace operator
16 s=poly(0, 's');
17 G=syslin('c', 1/s/(s^2+5*s+1)) //System
    representation G(s)=1/s/(s^2+5s+1)
18
19 //

20
21 //Nyquist plot with stability margins
22 show_margins(G, 'nyquist')
23 zoom_rect([-1.5 -1.2 0 1.2])
24
25 [Gm,pcw] = g_margin(G)
26 [Pm,gcw] = p_margin(G)
27
28 disp("Gain Margin",Gm,"Phase crossover frequency",
    pcw)
29 disp("Phase Margin",Pm,"Gain crossover frequency",
    gcw)
```

---

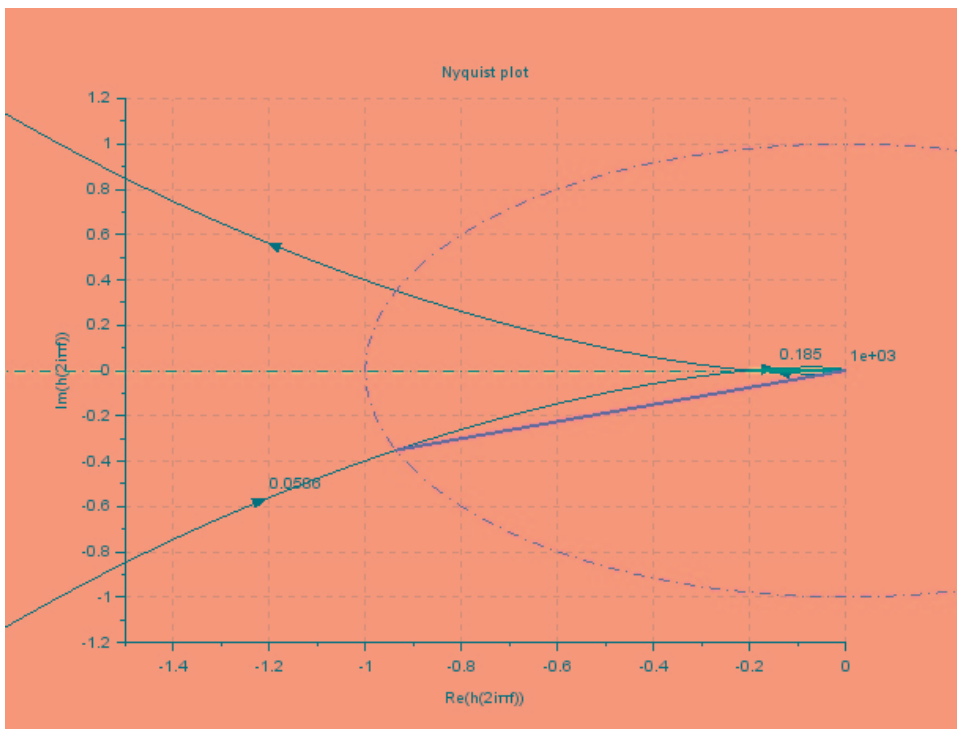


Figure 9.1: Nyquist plot