Scilab Manual for Advanced Control System (Rev. 2015) by Prof Mrs. Deepti Khimani Instrumentation Engineering Vivekanand Education Society'S Institute Of Technology/ Mumbai University[1]

Solutions provided by
Prof Mrs. Deepti Khimani
Instrumentation Engineering
Vivekanand Education Society'S Institute Of Technology/ Mumbai University

November 23, 2024

[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 Experiments

List of Figures

Experiment: 1

Construct phase trajectory of the system using Delta method and verify the same using Scilab

This code can be downloaded from the website wwww.scilab.in

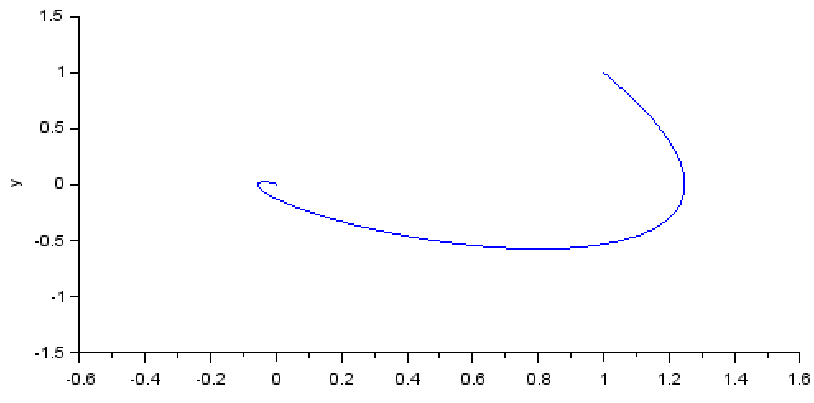

Figure 1.1: Lab

Experiment: 2

Check the existance and stability of limit cycles in Wien Bridge Oscillator.

This code can be downloaded from the website wwww.scilab.in

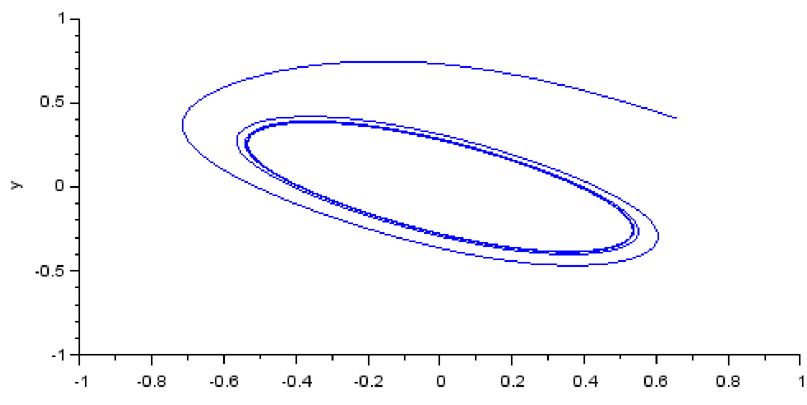

Figure 2.1: Lab

Experiment: 3

Plot amplitude versus describing function of saturation nonlinearity.

Scilab code Solution 3.1 Lab

```
//Lab 03:Plot amplitude versus describing function
   of saturation nonlinearity.

//scilab - 6.0.0
// Operating System : Windows 10, 64-bit

//
   ____________________________________________________________________________

close;
clc;
clear;
//
   ____________________________________________________________________________

```

```
//Response of the saturation nonlinearity to sinusoidal input
figure;
importXcosDiagram(".\Lab03_DF_Saturation_Model.xcos")
xcos_simulate(scs_m,4);
scs_m.props.context
plot(yt.time,yt.values(:,1),'r--','thickness',2)
plot(yt.time,yt.values(:,2),'b','thickness',2)
xlabel('Time (sec.)','fontsize',2);
ylabel('Amplitude','fontsize',2);
xset('font size',2);
title("Saturation nonlinearity output to sinusoidal input",'fontsize',4);
exec .\figure_properties.sci; //custom script for setting figure properties
//
//Describing Functin for saturation nonlinearity.
k=1;
N=1;
i=1;
Keq=[];

for a=0:0.2:10
    if k*a/N > 1 then
    Keq(i,1)=2/%pi*(k*asin(N/a/k)+N/a*sqrt(1-(N/k/a)^2))
    else
    Keq(i,1)=k
    end
    i=i+1;
end

a=0:0.2:10;
a=a';
figure,
```

```
plot(a,Keq,'thickness',2)
xlabel('$a$','fontsize',2);
ylabel('$K_{eq}$','fontsize',2);

xset('font size',2);
title("Describing Function for a saturation
    nonlinearity with k=N=1",'fontsize',4);
exec .\figure_properties.sci; //custom script for
    setting figure properties
zoom_rect([0 0 10 1.1])
//
```

This code can be downloaded from the website wwww.scilab.in

check Appendix AP 1 for dependency:

`figure_properties.sci`

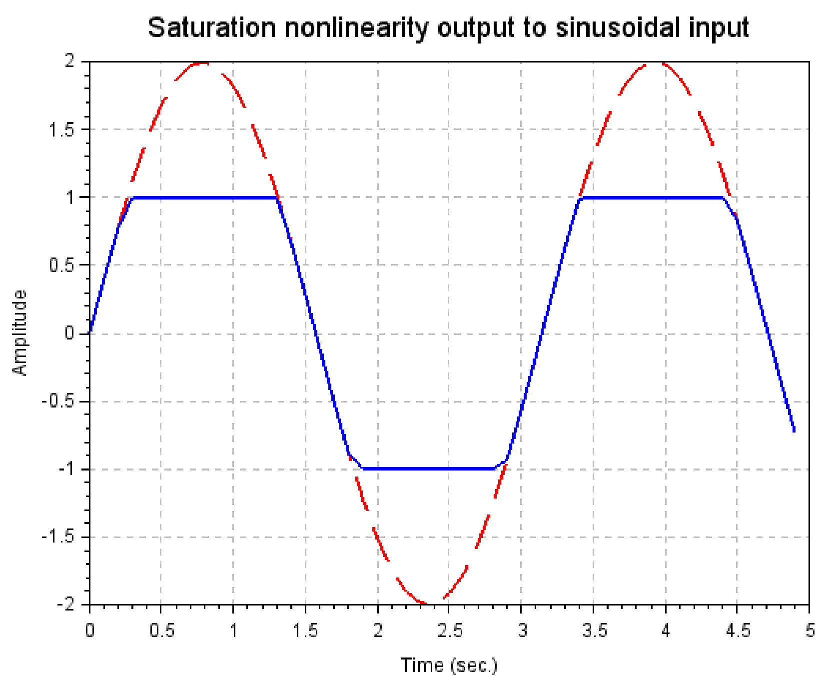

Figure 3.1: Lab

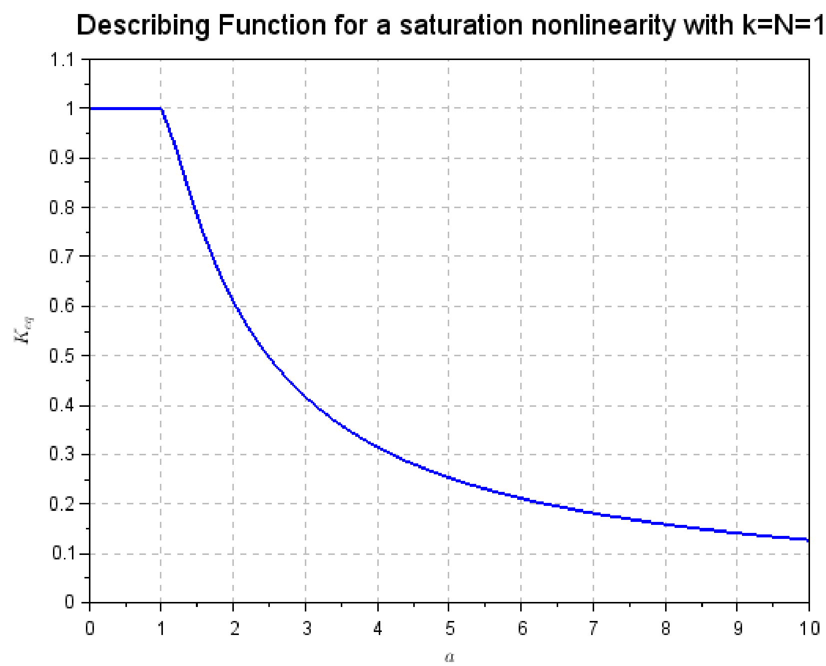

Figure 3.2: Lab

Experiment: 4

Plot amplitude and phase versus describing function of realy with hysterisis nonlinearity.

Scilab code Solution 4.1 Lab

```
//Lab 04. Plot amplitude and phase versus describing
    function of realy with hysterisis nonlinearity

//scilab - 6.0.0
// Operating System : Windows 10, 64-bit

//
    ____________________________________________________________________________

close;
clc;
clear;
//
    ____________________________________________________________________________
```

```
//Response of the Relay with hysteresis nonlinearity to sinusoidal input
figure;
importXcosDiagram(".\Lab04_DF_RelayNHyst_Model.xcos")
xcos_simulate(scs_m,4);
scs_m.props.context
plot(yt.time,yt.values(:,1),'r--','thickness',2)
plot(yt.time,yt.values(:,2),'b','thickness',2)

xlabel('Time (sec.)','fontsize',2);
ylabel('Amplitude','fontsize',2);
title("Relay with hysteresis nonlinearity output to sinusoidal input",'fontsize',4);
exec .\figure_properties.sci; //custom script for setting figure properties
zoom_rect([0 -1.2 5 1.2])
//------------------------------------------------------------------------
////Describing Functin for relay with hysteresis nonlinearity.
h=0.1;
N=1;
i=1;

for a=0.1:0.025:1
    if a<h then
        Keq(i,1)=0;
        ro(i,1)=0;
        theta(i,1)=0
    else
        Keq(i,1)=4*N/(%pi*a)*(sqrt(1-(h/a)^2)-%i*h/a)
        [r th]=polar(Keq(i,1));
        ro(i,1)=r; //magnitude
        theta(i,1)=clean(th); //angle in radians
    end
```

```
    i=i+1;
end

a=0.1:0.025:1
a=a';
figure,

subplot(2,1,1), plot(a,ro,'thickness',2)
xlabel('$a$','fontsize',2);
ylabel(['Magnitude', '$|K_{eq}|$'],'fontsize',2);

xset('font size',2);
exec .\figure_properties.sci; //custom script for setting figure properties
title("Describing Functin for relay with hysteresis nonlinearity with h=0.1 and N=1",'fontsize',4);

subplot(2,1,2), plot(a,theta*180/%pi,'thickness',2)
xlabel('$a$');
ylabel(['Phase', '$ \angle K_{eq}$','deg.']);
xset('font size',2);
exec .\figure_properties.sci; //custom script for setting figure properties
//
```

This code can be downloaded from the website wwww.scilab.in

check Appendix AP 1 for dependency:

`figure_properties.sci`

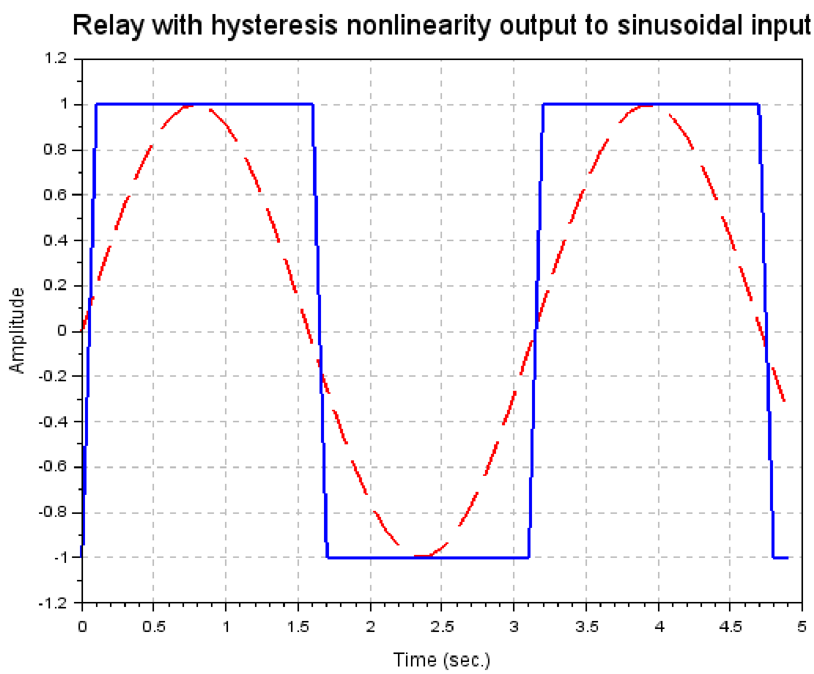

Figure 4.1: Lab

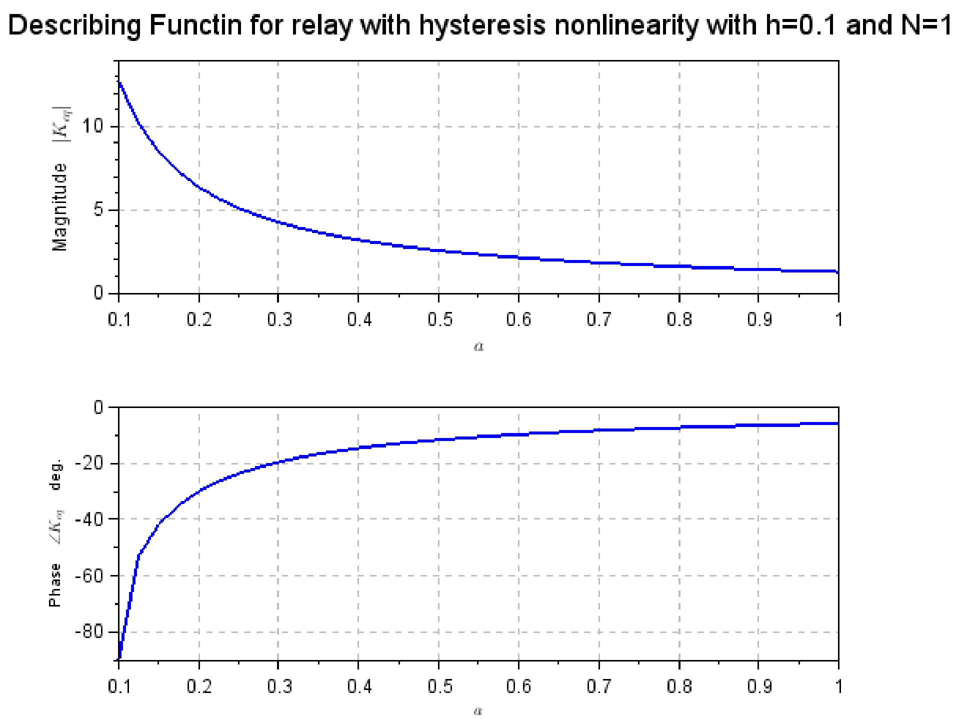

Figure 4.2: Lab

Experiment: 5

Investigate stability of a system with relay control using describing function.

Scilab code Solution 5.1 Lab

```
//Lab 05:Investigate stability of a system with
   relay control using describing function.
//scilab - 6.0.0
// Operating System : Windows 10, 64-bit

//
   ------------------------------------------------------------
close;
clc;
clear;
//
   ------------------------------------------------------------

// System Decription
s=poly(0,'s');
```

```
num=5;
den=(s+1)*(0.1*s+1)^2;
g=syslin('c',num/den)

//Nyquist plot of the system
nyquist(g)
title("Nyquist plot and describing function to determine limit cycle",'fontsize',4)
f=gca();
f.x_location = "origin"
f.y_location = "origin"
h=legend('');
h.visible = "off"
xset("color",2);

// Nyquist Plot of Describing Function for saturation nonlinearity.
omegat=0.05:0.05:%pi;
a=sin(omegat);
k=1;
Keq=4/%pi./a
DF_nyq=-1 ./Keq;

plot(DF_nyq,zeros(1,length(DF_nyq)),'m-.','thickness',2)
exec .\figure_properties.sci; //custom script for setting figure properties
//zoom_rect([-0.8 -1.4 0.4 1.4])

//limit cycle point (Identified on plot)
plot(-0.2064, 0,'bo');

// Frequency and Magnitude of limit cycle
p=roots(den);
p=abs(p);
omega=sqrt(sum(1./p)/prod(1./p)) //rad/sec
set(gca(),"grid",[1 1])
xset('font size',2)
```

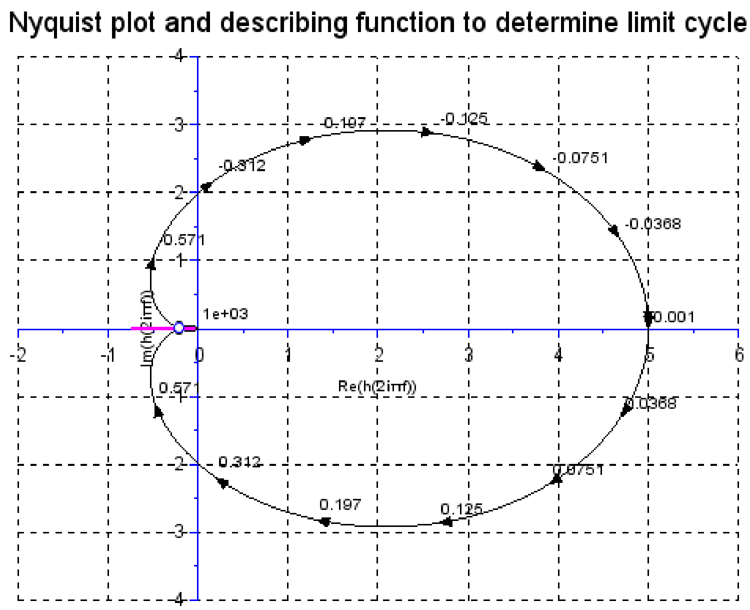

Figure 5.1: Lab

```
magn=prod(0.2064-roots(den))
```

check Appendix AP 1 for dependency:

`figure_properties.sci`

Experiment: 6

Design optimal control for a system to minimize performance index.

Scilab code Solution 6.1 Lab

```
//Lab 06: Design optimal control for a system to
    minimize quadratic
//performance index (P.I.)

//scilab - 6.0.0
// Operating System : Windows 10, 64-bit

//
    ------------------------------------------------------------------------

close;
clc;
clear;
//
    ------------------------------------------------------------------------

// System Decription
```

```
a=[0 1;1 -3];
b=[0 1]';
c=[1 0];
d=0;
sys=syslin('c',a,b,c,d)

// coefficients of P.I.
Q=[1 0;0 1];
R=1;

// Solution P of the riccati equation and feedback gain that minimizes P.I.
[K,P]=lqr(sys,Q,R);

//Closed loop system with optimal control

sysc=syslin('c',a+b*K,b,c,d)

// Close loop solution
t=0:0.1:10;
[y x]=csim('step',t,sysc);

//Closed loop response
plot(t,x,'thickness',2)
title('Evolution of states using optimal control','fontsize',4)
xlabel('Time t','fontsize',2)
ylabel('Response x(t)','fontsize',2)
set(gca(),"grid",[1 1])
h=legend('x1','x2');
h.font_size = 2;
xset('font size',2)
```

check Appendix AP 1 for dependency:

`figure_properties.sci`

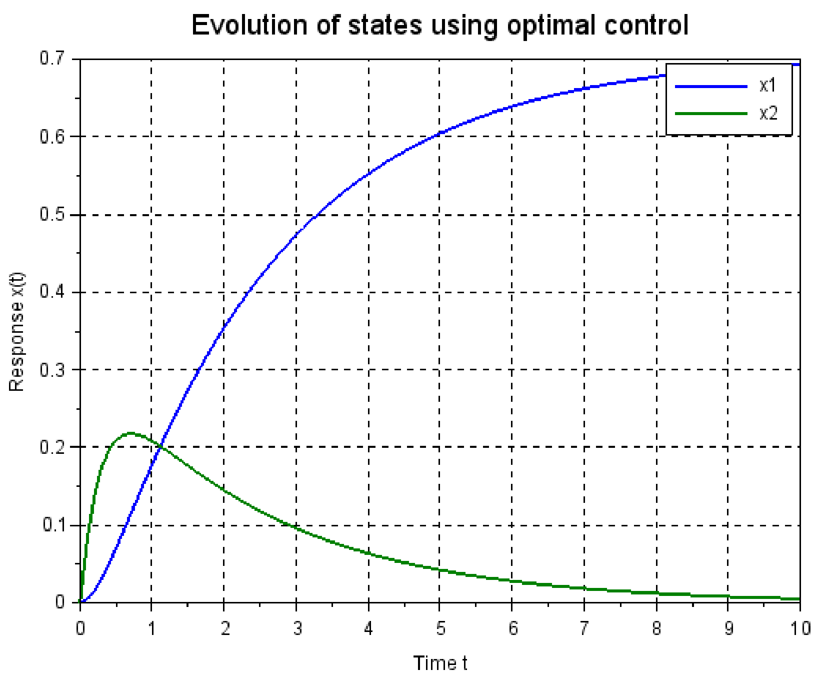

Figure 6.1: Lab

Experiment: 7

Design IMC controller for the non minimum phase delay system.

Scilab code Solution 7.1 Lab

```
//Lab 07: Design IMC controller for the Non-minimum
   Phase Delay system.

//scilab - 6.0.0
// Operating System : Windows 10, 64-bit

//
   ____________________________________________________________

close;
clc;
clear;
//
   ____________________________________________________________


//System transfer function without delay (delay
```

```
    introduced in zcos)
s=poly(0,'s');
num=0.5848*(-0.32*s+1);
den=0.18*s^2+0.87*s+1;

//model
num_m=0.5848*(-0.3549*s+1);
den_m=0.1858*s^2+0.8627*s+1;

// Response of the system with IMC Control
figure;
importXcosDiagram(".\Lab07_IMC_Model.xcos")
xcos_simulate(scs_m,4);
scs_m.props.context
plot(ye.time,ye.values(:,1),'thickness',2)
xlabel('t (sec.)','fontsize',2);
ylabel('y(t)','fontsize',2);
title("Response of the system with IMC controller.",
    'fontsize',4);
set(gca(),"grid",[1 1])
xset('font size',2)

exec .\figure_properties.sci; //custom script for
    setting figure properties

// Error in plant and model output.
figure;
plot(ye.time,ye.values(:,2),'thickness',2)
xlabel('t (sec.)','fontsize',2);
ylabel('e(t)','fontsize',2);
set(gca(),"grid",[1 1])
xset('font size',2)

title("Error in plant and model output.",'fontsize'
    ,4);
exec .\figure_properties.sci; //custom script for
    setting figure properties
```

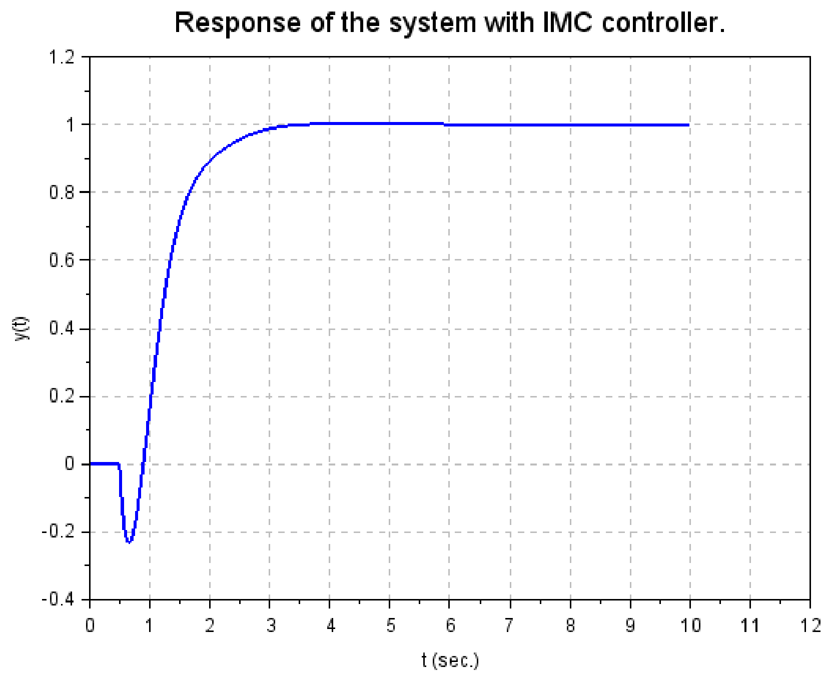

Figure 7.1: Lab

This code can be downloaded from the website wwww.scilab.in

check Appendix AP 1 for dependency:

`figure_properties.sci`

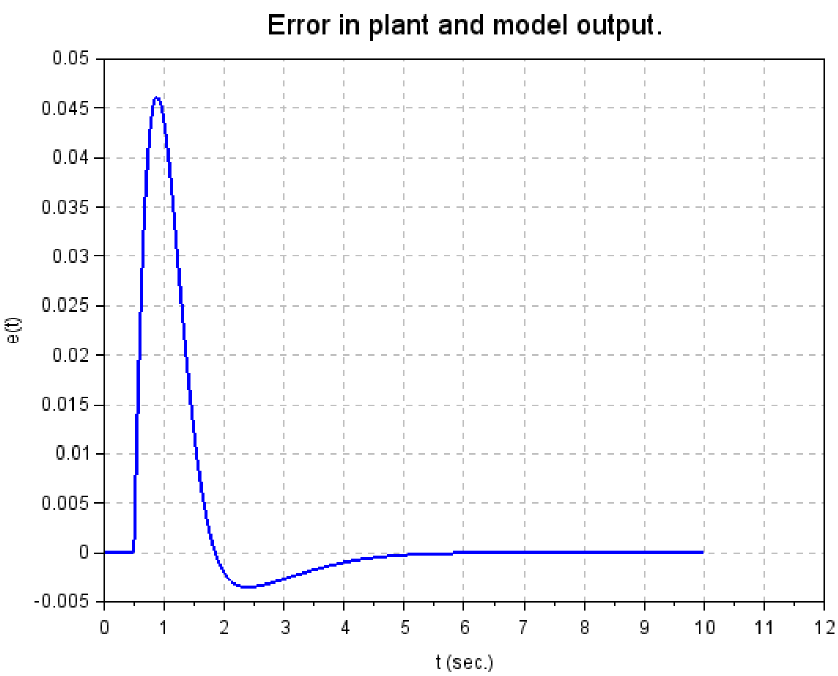

Figure 7.2: Lab

Experiment: 8

Computation of PID Controller gains based on IMC design for the first order delay system

check Appendix AP 1 for dependency:

`figure_properties.sci`

Scilab code Solution 8.1 Lab

```
//Lab 08:  Computation of PID Controller gains based
    on IMC Design for the First order Delay system.

//scilab - 6.0.0
// Operating System : Windows 10, 64-bit

//
    ________________________________________________________________

close;
clc;
clear;
```

```
//

//System transfer function
kp=0.5;
tp=1.5;
theta=0.8; //for Pade approximation of the delay of 0.8sec
s=poly(0,'s');
num=kp;
den=tp*s+1;

//IMC filter designed parameters
lambda=0.5;

//PID gain computation
kc=(tp+0.5*theta)/kp/(lambda+0.5*theta);
ti=tp+0.5*theta;
td=tp*theta/(2*tp+theta);

// Response of the system with IMC based PID control
figure;
importXcosDiagram(".\Lab08_IMC_PID_Model.xcos")
xcos_simulate(scs_m,4);
scs_m.props.context
plot(yt.time,yt.values,'thickness',2)
xlabel('t (sec.)','fontsize',2);
ylabel('y(t)','fontsize',2);
title("Response of the system with IMC based PID control.",'fontsize',4);
set(gca(),"grid",[1 1])
xset('font size',2)

exec .\\figure_properties.sci; //custom script for setting figure properties
```

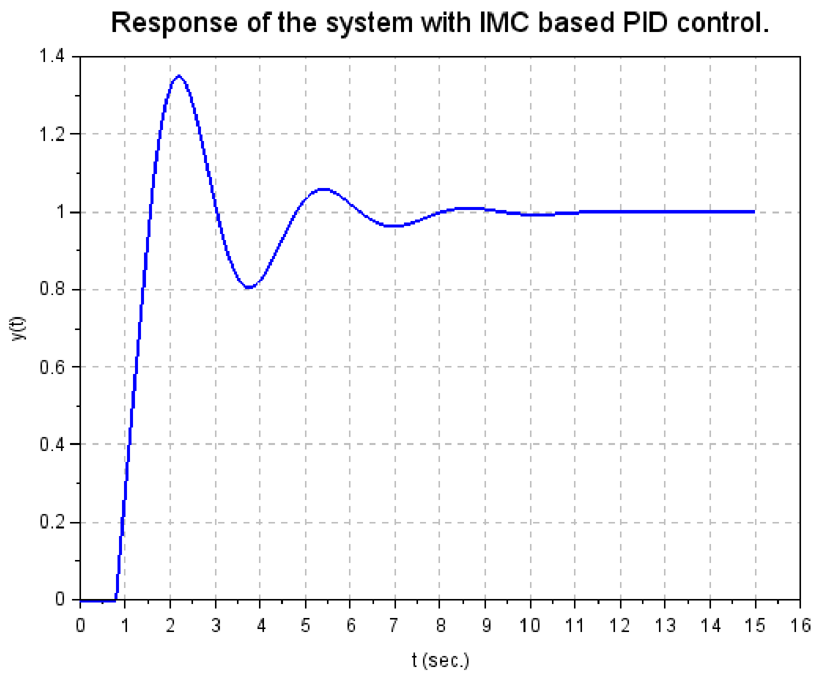

Figure 8.1: Lab

This code can be downloaded from the website wwww.scilab.in

Appendix

Scilab code AP 11

```
//________________________________________________________________________________
//figure handel settings
f=get("current_figure"); //Current figure handle
f.background=8; //make the figure window background
    white
l=f.children(1);
l.background=8 ;//make the text background white
id=color('grey');
xgrid(id);
//________________________________________________________________________________
```

figure properties