Scilab Manual for
Control and Instrumentation
by Prof A. George Ansfer
Electrical Engineering
St. Xavier's Catholic College of Engineering[1]

Solutions provided by
Prof A.George Ansfer
Electrical Engineering
St.Xavier's Catholic College of Engineering

August 6, 2025

[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

Digital Simulation of P, PI, PD & PID Controller

Scilab code Solution 1.1 Exp1

```
//Digital Simulation of P, PI, PD, and PID
    controllers
// 1- Open Loop Program
num=poly([10],'s','coeff');//Numerator input
den=poly([20 10 1],'s','coeff');//Denominator input
q=syslin('c',num/den)//Ratio of the numerator to the
    denominator
t=0:0.05:2.5;//time interval
p=csim('step',t,q);
subplot(321)
plot2d(t,p);
xtitle(['Open Loop '],'Time(Second)', 'Amplitude' );

// 2- P Control Program
kp=300;
num=poly([kp],'s','coeff');
den=poly([20+kp 10 1],'s','coeff');
q=syslin('c',num/den)
t=0:0.01:2;
```

```
p=csim('step',t,q);
subplot(322)
plot2d(t,p);
xtitle(['P Control '],'Time(Second)', 'Amplitude' );

// 3      PI Control Program
kp=30;
ki=70;
num=poly([ki kp],'s','coeff');
den=poly([ki 20+kp 10 1],'s','coeff');
q=syslin('c',num/den)
t=0:0.01:2;
p=csim('step',t,q);
subplot(323)
plot2d(t,p);
xtitle(['PI Control '],'Time(Second)', 'Amplitude' );

// 4      PD Control Program
kp=300;
kd=10;
num=poly([kp kd],'s','coeff');
den=poly([20+kp 10+kd 1],'s','coeff');
q=syslin('c',num/den)
t=0:0.01:2;
p=csim('step',t,q);
subplot(324)
plot2d(t,p);
xtitle(['PD Control '],'Time(Second)', 'Amplitude' );

// 5      PID Control Program
kp=350;
kd=50;
ki=300;
num=poly([ki kp kd],'s','coeff');
den=poly([ki 20+kp 10+kd 1],'s','coeff');
q=syslin('c',num/den)
```

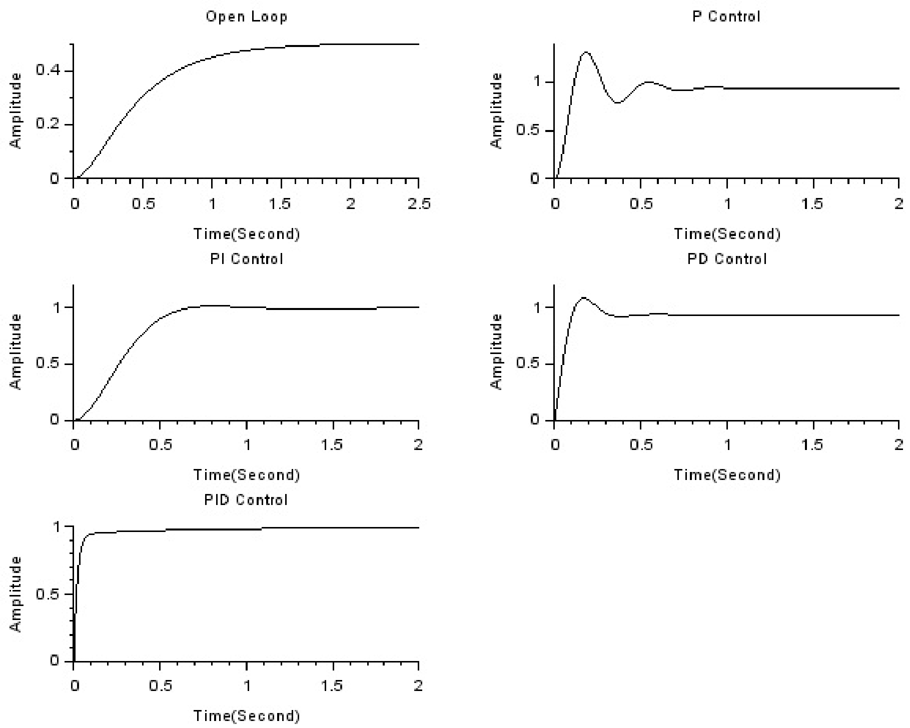

Figure 1.1: Exp1

```
t=0:0.01:2;
p=csim('step',t,q);
subplot(325)
plot2d(t,p);
xtitle(['PID Control '],'Time(Second)', 'Amplitude'
    );
```

Experiment: 2

Digital Simulation of Linear System

Scilab code Solution 2.2 exp2

```
//Second order system step response for damping
    conditions
t=0:0.0000001:0.0002;
d=[0.5 1 1.5];//Entering the damping conditions
    values
cv=[1 2 3];
s=%s;
for n=1:3
num = 10^10;
den = s^2 + 2*d(n)*100000*s  +10^10;
 P = syslin('c',num,den);
Ps=csim('step',t,P);
plot2d(t,Ps,style=cv(n));
end;
xgrid(6);
xtitle(['Second order step response '],'Time(Second)
    ', 'Amplitude' );
legends(['d=0.5(underdamped)';'d=1(critically damped
    )';'d=1.5(overdamped)'],[1,2,3],opt=4);
```

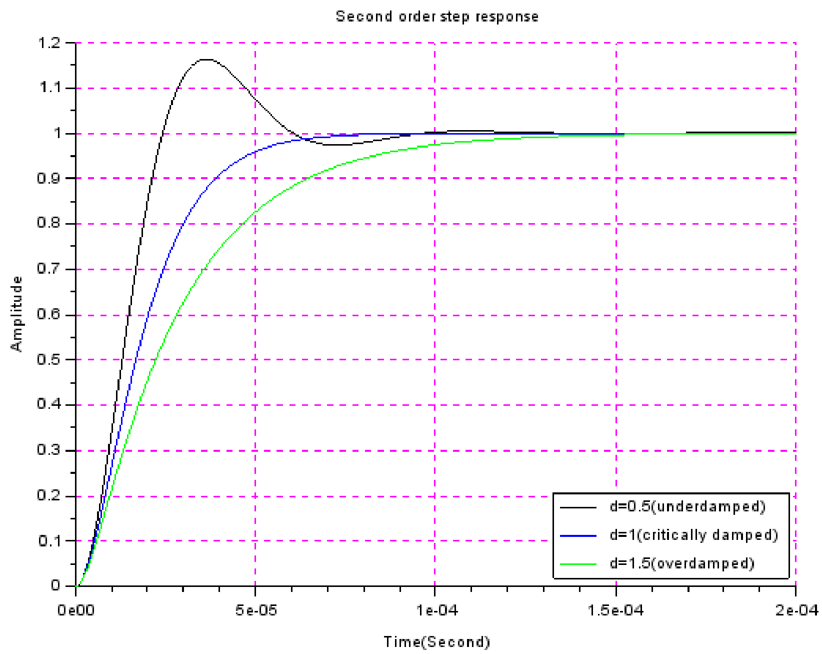

Figure 2.1: exp2

Experiment: 3

Simulation of Control Systems

Scilab code Solution 3.3 exp3

```
// Effects on Stability.
// G(s) = a^2 / s(s+2b a) , H(s) = C , T(s) = C a^2 / ( s(s+2b a) + C a^2 )
s=%s;
t=0:0.01:10;
a=1;b=1;
C=[1,2,5,10]
for n=1:4
    T=syslin('c', C(n)*a^2 , s*(s + 2*b*a) + C(n)*a^2 );
    Ts=csim('step',t,T);
    xset("line style",n);
    plot2d(t,Ts);
    xgrid(5);
end
xtitle('Effects on Stability .','Time(sec)','C(t)');
legends(['C=1';'C=2';'C=5';'C=10';],[[1;1],[1;2],[1;3],[1;4]], opt=4);
```

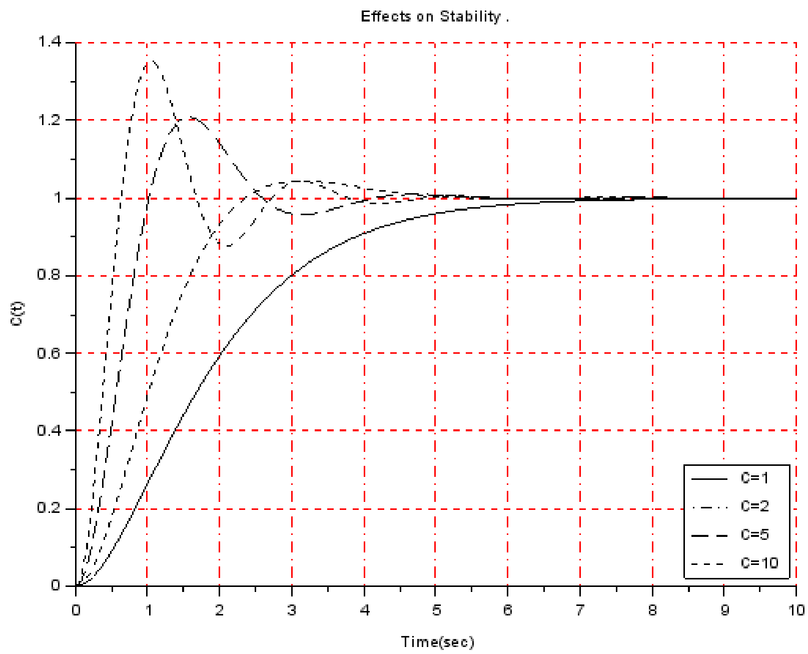

Figure 3.1: exp3

Scilab code Solution 3.4 exp4

```
//Frequency Domain Methods for controller design

// 1- Normalized bandwidth vs Damping factor (second
    order system)
deff("[a]=f1(b)","a=sqrt(1-2*b^2+sqrt(2-4*b^2+4*b^4)
   )")
b=[0:0.01:0.9];
subplot(311)
fplot2d(b,f1,[1])
xgrid(2)
xtitle(['Normalized bandwidth vs Damping Factor'],'
   Damping ratio', ' Normalized bandwidth ' );

// 2- Peak overshoot  vs  Resonance Peak  (second
   order system)
deff("[C]=f2(b)","C=exp((-%pi*b)/sqrt(1-b^2))")
deff("[D]=f3(b)","D=1/(2*b*sqrt(1-b^2))")
b=[0.05:0.01:0.9];
subplot(312)
xset("line style",4);
fplot2d(b,f2,[1])
xset("line style",1);
fplot2d(b,f3,[1])
xgrid(3)
xtitle([' Peak overshoot  vs  Resonance Peak'  ],'
   Damping ratio', 'Peak Gain, Resonance Gain' );
legends([' Peak Gain';' Resonance Gain'
   ],[[1;4],[1;1]],opt=1);

// 3- Resonant Frequency vs Damping Frequency (
   Second order system)
deff("[e]=f4(b)","e=sqrt(1-2*b^2)/sqrt(1-b^2)")
```

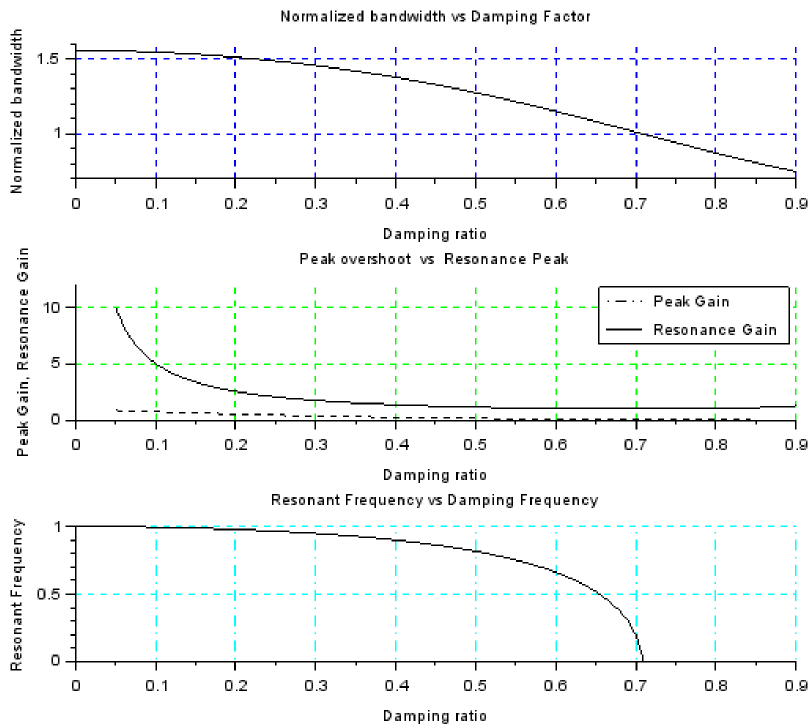

Figure 3.2: exp4

```
b=[0:0.01:0.9];// don't end with 1 bec, division by
    0 error
subplot(313)
fplot2d(b,f4,[1])
xgrid(4)
xtitle(['Resonant Frequency vs Damping Frequency'],'
    Damping ratio',' Resonant Frequency ');
```