Scilab Manual for
Systems and Signal Processing Lab
by Prof Kotha Manohar
Electronics and Telecommunication Engineering
Matrusri Engineering College[1]

Solutions provided by
Prof Kotha Manohar
Electronics and Telecommunication Engineering
Matrusri Engineering College

April 20, 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 Experiments

List of Figures

Experiment: 1

Introduction to SCILAB and signal generation.

Scilab code Solution 1.1 Basic Signals Generation

```
// Experiment Number : 1
//Write a program to generate continuous and
   discrete time impulse, step, rectangual, saw
   tooth and sinusoidal functions
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                            Enrolment
   Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.1.0

clc;
clear;
close;
```

```

// Continuous time Unit Impulse function generation
t=-4:0.1:4;
a=[zeros(1,40) 1 zeros(1,40)];
k=input(”Enter the Amplitude of impulse function :”
    ); // reading amplitude value from keyboard
b=k*a;
subplot(2,2,1);
plot(t,b);
//Labeling the axes and title for the figure
xlabel('Time t');
ylabel('Amplitude');
title('Continuous time Impulse function');

// Discrete time Unit Impulse function generation
subplot(2 ,2 ,2);
plot2d3(t,b);
//Labeling the axes and title for the figure
xlabel('Time t');
ylabel('Amplitude');
title('Discrete time impulse function');

// Discrete time Unit step function generation
t=0:3;
y=ones(1,4);
subplot(2 ,2 ,3);
plot2d3(t,y);
//Labeling the axes and title for the figure
xlabel('Time t');
ylabel('Amplitude');
title('Discrete step function');

// Continuous time Unit step function generation
subplot(2 ,2 ,4);
plot(t,y);
//Labeling the axes and title for the figure
xlabel('Time t');
ylabel('Amplitude');
```

```
title('Continuous step function');

//Continuous time Square wave function generation

a=input("Enter  the amplitude of square wave function :" );
t =0:0.001:1;
d=a*squarewave(2* %pi *10* t);
figure;
subplot(2 ,2 ,1);
plot(t,d);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Continuous time Square wave signal');

//Discrete time Square wave function generation
n=0:0.01:1;
d=a* squarewave (2*%pi*10*n);
subplot (2 ,2 ,2);
plot2d3 (n,d);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Discrete time Square wave signal');

//Continuous time Sawtooth wave  generation
Fs = 20; // samples per second
t_total = 10; // seconds
n_samples = Fs * t_total ;
t = linspace (0, t_total , n_samples );
f =500; // sound frequency
saw_wave =2*( f*t- floor (0.5+ f*t));
subplot(2 ,2 ,3);
plot(t,saw_wave );
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
```

```
title('Continuous time Sawtooth wave');

//Discrete time Sawtooth wave generation
Fs = 20; // samples per second
t_total = 10; // seconds
n_samples = Fs * t_total ;
n = linspace (0, t_total , n_samples );
f =500; // sound frequency
saw_wave =2*(f*n- floor (0.5+ f*n));
subplot(2 ,2 ,4);
plot2d3(n, saw_wave );
//Labeling the axes and title for the figure
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Discrete time Sawtooth wave');

//Continuous time Sine wave  generation
//a=input(  Enter   amplitude     ) ;
n=0:0.001:1;
d=a* sin(2*%pi*10*n);
figure;
subplot (2 ,1 ,1);
plot(n,d);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Continuous time Sinusoidal signal');

//Discrete time Sine wave  generation
n=0:0.009:1;
d=a* sin(2*%pi*10*n);
subplot (2 ,1 ,2);
plot2d3 (n,d);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Discrete time sinusoidal signal');

```

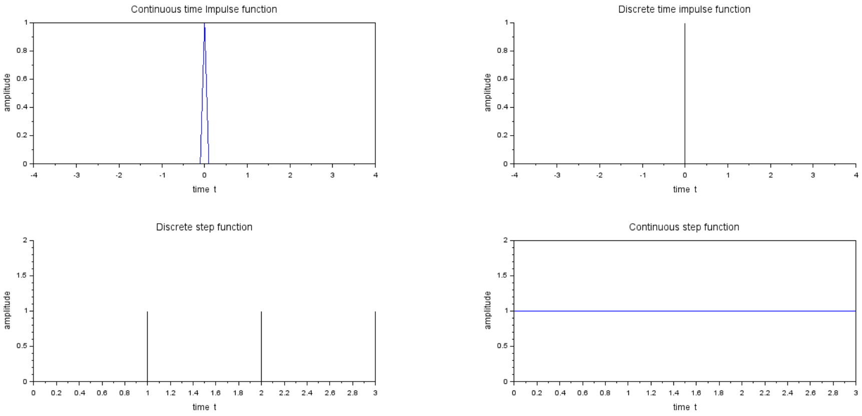

Figure 1.1: Basic Signals Generation

```

//Inputs in console window
//Enter the Amplitude of impulse function :1

//Enter  the amplitude of square wave function :1
```

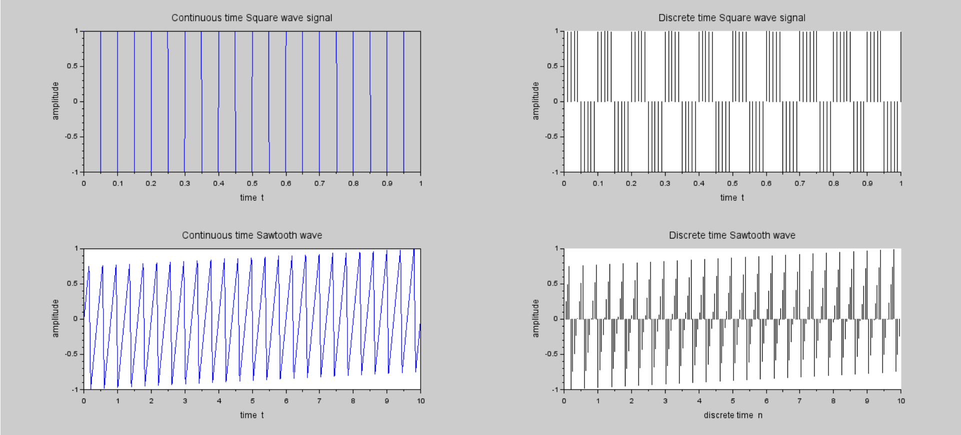

Figure 1.2: Basic Signals Generation

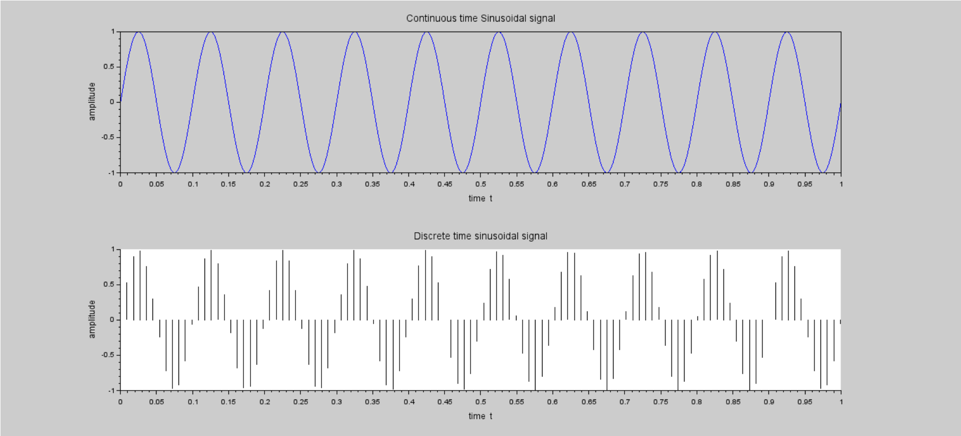

Figure 1.3: Basic Signals Generation

Experiment: 2

Perform Fast Fourier Transform

check Appendix AP 1 for dependency:

`idftfunction.sci`

Scilab code Solution 2.2 Inverse DFT

```
// Experiment Number : 2.2
//Write a program to find the IDFT of a discrete Frequncy Domain signal
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                          Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.0.2


clc;
```

```
clear;
close;

X=input('Enter the frequency domain signal  X=');
N=length(X);
x=idft(X,N);
disp('IDFT of X(K) is')
disp(x);
n=0:1:N-1
plot2d3(n,x);
xlabel('Discrete time n');
ylabel('Amplitude');
title('IDFT or time domain signal x(n)');




// Enter the frequency domain signal  X=[4 0 0 0]

//"IDFT of X(K) is"
 // 1.    1.    1.    1.
```

Scilab code Solution 2.3 FFT

```
// Experiment Number : 2
//Write a program to find the FFT and Inverse FFT of
    a discrete time signal
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                          Enrolment
   Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad
```

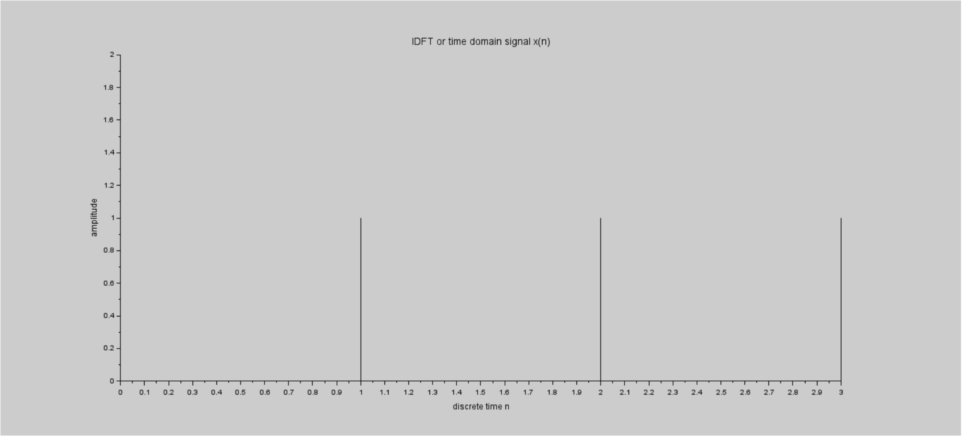

Figure 2.1: Inverse DFT

```
//


// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;

//Input time domain signal

x=input ('Enter the time domain sequence x=');
N=length(x);     //Finding the length of input signal

//FFT of  a signal
y=fft(x);  //Finding FFT of a sequence
disp('Frequency domain signal is');
disp(y);
y1=abs(y);  //Finding the magnitude response
disp('Magnitude Response is');
disp(y1);
y2=atan(imag(y),real(y));  //Finding the phase
```

```
   response
disp('Phase Response is');
disp(y2);

//Plotting the input time domain signal

n=0:1:N-1
subplot(2,2,1);
plot2d3(n,x);
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Input time domain signal');

//Plotting the magnitude spectrum

k=0:1:N-1
subplot(2,2,2);
plot2d3(k,y1);
xlabel('Discrete frequency  k');
ylabel('Amplitude');
title('Magnitude spectrum of FFT signal');

//Plotting the phase spectrum

subplot(2,2,3);
plot2d3(k,y2);
xlabel('Discrete frequency  k');
ylabel('Phase angle');
title('Phase spectrum of FFT signal');

// Finding Inverse Fast Fourier Transform
z=ifft(y);
disp('Inverse Fast Fourier Transform is');
disp(z);

//Plotting Inverse FFT signal

subplot(2,2,4);
```

```
plot2d3(n,z);
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Inverse FFT or time domain signal');


//Enter the time domain sequence x=[1 2 3 4]


// Frequency domain signal is

//    10.   -2. + 2.i   -2.   -2. - 2.i

// Magnitude Response is

//    10.    2.8284271    2.    2.8284271

// Phase Response is


//           column 1 to 3

//    0.    2.3561945    3.1415927

//           column 4

//  -2.3561945

// Inverse Fast Fourier Transform is

//    1.    2.    3.    4.
```

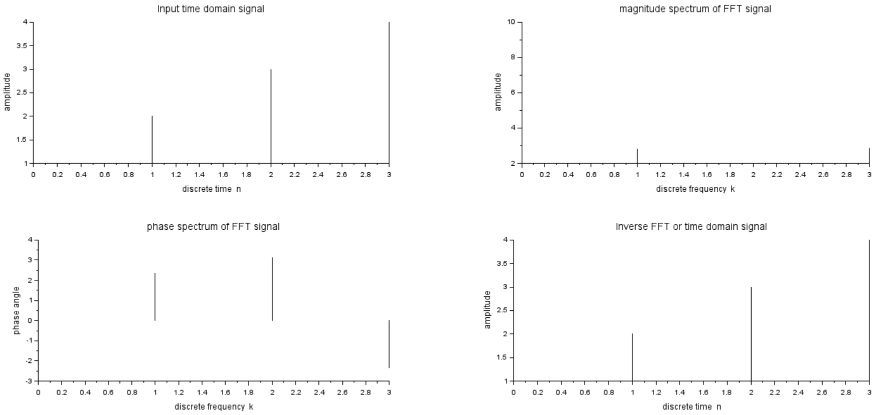

Figure 2.2: FFT

Experiment: 3

Perform Linear convolution.

Scilab code Solution 3.1 Linear Convolution

```
// Experiment Number : 3
//Write a program to find the linear convolution of two discrete time signals
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                              Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;

//Generating and plotting the input signal

```

```
x=input('Enter the time domain sequence  x=');
N1=length(x);   //Finding the length of input signal
n1=0:1:N1-1
subplot(3,1,1);
plot2d3(n1,x);
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Time domain input signal  x(n)');

//Generating and plotting the impulse response of the system

h=input('Enter the impulse response of the system  h=');
N2=length(h);
n2=0:1:N2-1
subplot(3,1,2);
plot2d3(n2,h);
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Impulse response of the system  h(n)');

//Generating and plotting the linear convolution of two discrete time signals
y=conv(x,h);
disp('Linear Convolution of input signal and impulse response of the system is:')
disp(y);
N=length(y);
n3=0:1:N-1
subplot(3,1,3);
plot2d3(n3,y);
xlabel('Discrete time  n3');
ylabel('Amplitude');
title('Linear convolution function  y(n3)');



```

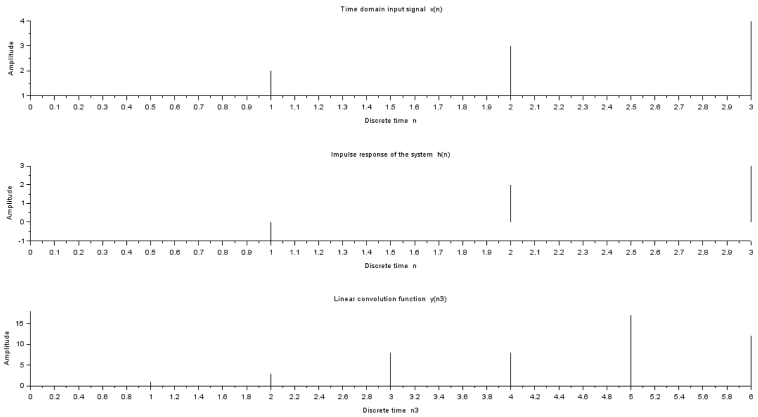

Figure 3.1: Linear Convolution

```

//Enter the time domain sequence  x=[1 2 3 4]

//Enter the impulse response of the system  h=[1 -1 2 3]


//  "Linear Convolution of input signal and impulse response of the system is:"

//    1.    1.    3.    8.    8.    17.    12.
```

Experiment: 4

Perform Circular Convolutions.

Scilab code Solution 4.1 Circular Convolution

```
// Experiment Number : 4
//Write a program to find the Circular convolution of two discrete time signals
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                    Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;

//Generating and plotting the input signal

```

```
x=input('Enter the time domain sequence  x=');
disp(x);
N1=length(x);   //finding the length of input signal
n1=0:1:N1-1
subplot(3,1,1);
plot2d3(n1,x);
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Time domain input signal  x(n)');

//Generating and plotting the impulse response of the system

h=input('Enter the impulse response of the system  h=');
disp(h);
N2=length(h);
n2=0:1:N2-1
subplot(3,1,2);
plot2d3(n2,h);
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Impulse response of the system  h(n)');

//Generating and plotting the circular convolution of two discrete time signals

N3=max(N1,N2);
x1=[x,zeros(1,N3-N1)];  //Zero padding to input sequence
h1=[h,zeros(1,N3-N2)];  //Zero padding to impulse response sequence
X1=fft(x1,-1);  //DFT Computation of input
H1=fft(h1,-1);  //DFT Computation of impulse response
Y=X1.*H1;   //output sequence in frequency domain
y=fft(Y,1) ;    ////IDFT Computation or output sequence in time domain
```

```
disp('Circular Convolution Sequence y[n]=',y);
n3=0:1:N3-1
subplot(3,1,3);
plot2d3(n3,y);
xlabel('Discrete time n3');
ylabel('Amplitude');
title('Circular convolution function y(n3)');




//Enter the time domain sequence x=[1 2 3 4]


//    1.    2.    3.    4.
//Enter the impulse response of the system h=[2 4 1]


//    2.    4.    1.

//  "Circular Convolution Sequence y[n]="

//    21.    12.    15.    22.
```

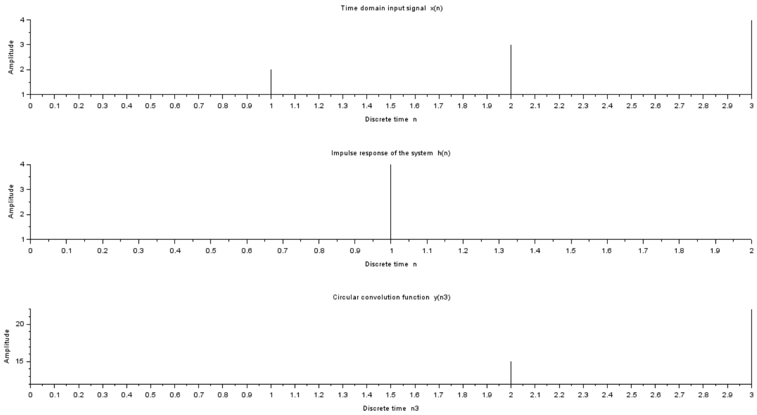

Figure 4.1: Circular Convolution

Experiment: 5

Perform FIR filters design using different window functions.

Scilab code Solution 5.1 FIR Filter Design

```
// Experiment Number : 5
//Write a program to generate Lowpass FIR Filter using different window techniques
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                    Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.1.0


clc;
```

```
clear;
close;

fc=input ('Enter the cutoff frequency  fc=');
fs=input ('Enter the sampling frequency  fs=');
N=input ('Enter the order of filter  N=');

//Finding the magnitude response of Lowpass FIR Filter  using Rectangular window technique

w1 =(2*%pi)*(fc/fs);
disp('Digital cutoff frequency in radians:');
disp(w1);
wc1 =w1/%pi;
disp('Normalized digital cutoff frequency in radians:');
disp(wc1);
[wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0],'re',[0,0]); //filter design
disp('Impulse response of Lowpass FIR filter using Rectangular window technique:h(n)=');
disp(wft);
a=gca();
subplot(2,2,1);
plot(2*fr,wfm);
//Labeling the axes and title for the figure
xlabel('Normalized digital frequency w');
ylabel('Magnitude ');
title('Magnitude response of Lowpass FIR Filter using Rectangular window');

//Finding the magnitude response of Lowpass FIR Filter  using Triangular window technique

[wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0],'tr',[0,0]); //filter design
disp('Impulse response of Lowpass FIR filter using Triangular window technique:h(n)=');
disp(wft);
a=gca();
subplot(2,2,2);
plot(2*fr,wfm);
//Labeling the axes and title for the figure
xlabel('Normalized digital frequency w');
ylabel('Magnitude ');
title('Magnitude response of Lowpass FIR Filter using Triangular window');

//Finding the magnitude response of Lowpass FIR Filter  using Hamming window technique

[wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0],'hm',[0,0]); //filter design
disp('Impulse response of Lowpass FIR filter using Hamming window technique:h(n)=');
disp(wft);
a=gca();
subplot(2,2,3);
plot(2*fr,wfm);
//Labeling the axes and title for the figure
xlabel('Normalized digital frequency w');
ylabel('Magnitude ');
title('Magnitude response of Lowpass FIR Filter using Hamming window ');

//Finding the magnitude response of Lowpass FIR Filter using Hanning window technique

[wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0],'hn',[0,0]); //filter design
disp('Impulse response of Lowpass FIR filter using Hanning window technique :h(n)=');
disp(wft);
a=gca();
subplot(2,2,4);
plot(2*fr,wfm);
```

```
//Labeling the axes and title for the figure
xlabel('Normalized digital frequency w');
ylabel('Magnitude ');
title('Magnitude response of Lowpass FIR Filter using Hanning window');


//Enter the cutoff frequency  fc=1200

//Enter the sampling frequency  fs=10000

//Enter the order of filter  N=3


//  "Digital cutoff frequency in radians:"

//   0.7539822

//  "Normalized digital cutoff frequency in radians:"

//   0.24

//  "Impulse response of Lowpass FIR filter using Rectangular window technique:h(n)="

//   0.1920103   0.2343554   0.2343554   0.1920103

//  "Impulse response of Lowpass FIR filter using Triangular window technique:h(n)="

//   0.0768041   0.1874843   0.1874843   0.0768041

//  "Impulse response of Lowpass FIR filter using Hamming window technique:h(n)="

//   0.0153608   0.1804536   0.1804536   0.0153608

```

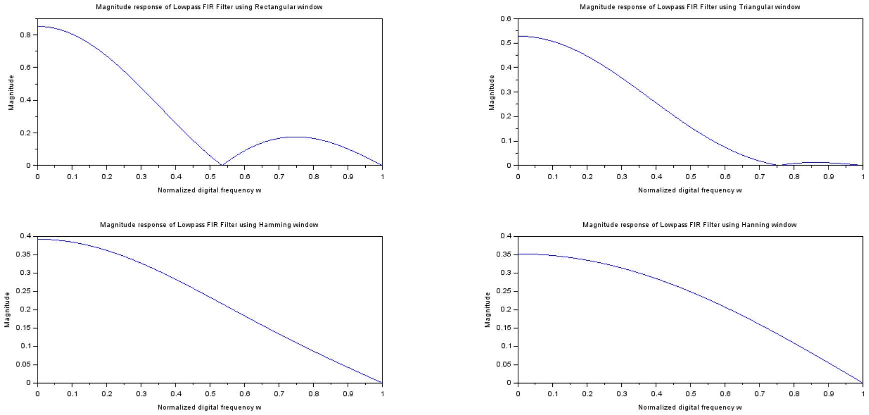

Figure 5.1: FIR Filter Design

```
//  "Impulse response of Lowpass FIR filter using Hanning window technique :h(n)="

//    0.    0.1757665    0.1757665    0.
```

Experiment: 6

Perform IIR filters design: Butterworth and Chebyshev.

Scilab code Solution 6.1 Butterworth Filter

```
// Experiment Number : 6.1
//Write a program to generate Butterworth lowpass,
   highpass, bandpass and bandstop IIR Filter
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                          Enrolment
   Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad
//


// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;

```

```
fc=input ('Enter the cutoff frequency  fc=');
fs=input ('Enter the sampling frequency  fs=');
N=input ('Enter the order of the filter  N=');
fp=2*fc/fs;

//Generating lowpass Butterworth IIR filter

[Hz1]=iir(N,'lp','butt',[fp/2,0],[0,0]);
[Hw1,w1]=frmag(Hz1,256);
subplot(2,2,1);
plot2d3(w1,abs(Hw1));
xlabel('Frequency');
ylabel('Magnitude');
title('Butterworth lowpass IIR filter');


//Generating Highpass Butterworth IIR filter

[Hz2]=iir(N,'hp','butt',[fp/2,0],[0,0]);
[Hw2,w2]=frmag(Hz2,256);
subplot(2,2,2);
plot2d3(w2,abs(Hw2));
xlabel('Frequency');
ylabel('Magnitude');
title('Butterworth highpass IIR filter');

//Generating Bandpass Butterworth IIR filter

[Hz3]=iir(N,'bp','butt',[fp/1.2,fp/3],[0,0]);
[Hw3,w3]=frmag(Hz3,256);
subplot(2,2,3);
plot2d3(w3,abs(Hw3));
xlabel('Frequency');
ylabel('Magnitude');
title('Butterworth Bandpass IIR filter');

//Generating bandstop Butterworth IIR filter

```

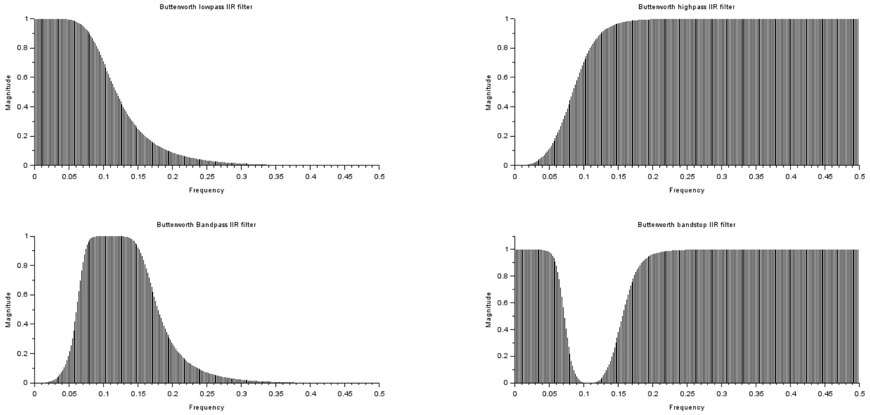

Figure 6.1: Butterworth Filter

```
[Hz4]=iir(N,'sb','butt',[fp/1.2,fp/3],[0,0]);
[Hw4,w4]=frmag(Hz4,256);
subplot(2,2,4);
plot2d3(w4,abs(Hw4));
xlabel('Frequency');
ylabel('Magnitude');
title('Butterworth bandstop IIR filter');



//Enter the cutoff frequency  fc=1000

//Enter the sampling frequency  fs=10000

//Enter the order of the filter  N=3
```

Scilab code Solution 6.2 Type 1 Chebyshev Filter

```
// Experiment Number : 6.2
```

```
//Write a program to generate Type-1 Chebyshev lowpass, highpass, bandpass and bandstop IIR Filter
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                          Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad
//


// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;

fc=input ('Enter the cutoff frequency  fc=');
fs=input ('Enter the sampling frequency  fs=');
N=input ('Enter the order of the filter  N=');
fp=2*fc/fs;

//Generating lowpass  Type-1 Chebyshev  IIR filter

[Hz1]=iir(N,'lp','cheb1',[fp/2,0],[0.2,0]);
[Hw1,w1]=frmag(Hz1,256);
subplot(2,2,1);
plot2d3(w1,abs(Hw1));
xlabel('Frequency');
ylabel('Magnitude');
title(' Lowpass Type-I chebyshev IIR filter');

//Generating Highpass Type-1 Chebyshev  IIR filter

[Hz2]=iir(N,'hp','cheb1',[fp/2,0],[0.2,0]);
[Hw2,w2]=frmag(Hz2,256);
```

```
subplot(2,2,2);
plot2d3(w2,abs(Hw2));
xlabel('Frequency');
ylabel('Magnitude');
title(' Highpass Type-I chebyshev IIR filter');

//Generating Bandpass Type-1 Chebyshev  IIR filter

[Hz3]=iir(N,'bp','cheb1',[fp/1.2,fp/3],[0.2,0]);
[Hw3,w3]=frmag(Hz3,256);
subplot(2,2,3);
plot2d3(w3,abs(Hw3));
xlabel('Frequency');
ylabel('Magnitude');
title(' Bandpass Type-I chebyshev IIR filter');

//Generating bandstop Type-1 Chebyshev  IIR filter

[Hz4]=iir(N,'sb','cheb1',[fp/1.2,fp/3],[0.2,0]);
[Hw4,w4]=frmag(Hz4,256);
subplot(2,2,4);
plot2d3(w4,abs(Hw4));
xlabel('Frequency');
ylabel('Magnitude');
title('Bandstop Type-I chebyshev  IIR filter');



//Enter the cutoff frequency  fc=1000

//Enter the sampling frequency  fs=10000

//Enter the order of the filter  N=3
```

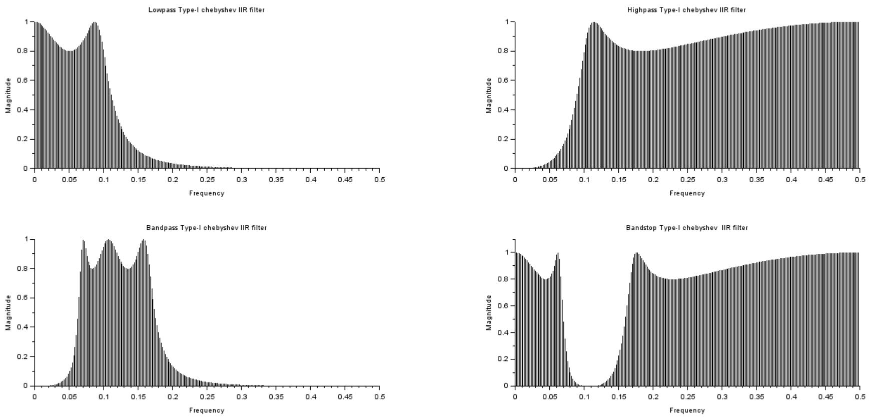

Figure 6.2: Type 1 Chebyshev Filter

Scilab code Solution 6.3 Type 2 Chebyshev Filter

```
// Experiment Number : 6.3
//Write a program to generate Type-2 Chebyshev lowpass , highpass , bandpass and bandstop IIR Filter
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                              Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College ,Hyderabad
//


// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;

fc=input ('Enter the cutoff frequency  fc=');
```

```
fs=input ('Enter the sampling frequency  fs=');
N=input ('Enter the order of the filter  N=');
fp=2*fc/fs;

//Generating lowpass  Type-2 Chebyshev  IIR filter

[Hz1]=iir(N,'lp','cheb2',[fp/2,0],[0,0.1]);
[Hw1,w1]=frmag(Hz1,256);
subplot(2,2,1);
plot2d3(w1,abs(Hw1));
xlabel('Frequency');
ylabel('Magnitude');
title(' Lowpass Type-2 chebyshev IIR filter');

//Generating Highpass Type-2 Chebyshev  IIR filter

[Hz2]=iir(N,'hp','cheb2',[fp/2,0],[0,0.1]);
[Hw2,w2]=frmag(Hz2,256);
subplot(2,2,2);
plot2d3(w2,abs(Hw2));
xlabel('Frequency');
ylabel('Magnitude');
title(' Highpass Type-2 chebyshev IIR filter');

//Generating Bandpass Type-2 Chebyshev  IIR filter

[Hz3]=iir(N,'bp','cheb2',[fp/1.2,fp/3],[0,0.1]);
[Hw3,w3]=frmag(Hz3,256);
subplot(2,2,3);
plot2d3(w3,abs(Hw3));
xlabel('Frequency');
ylabel('Magnitude');
title(' Bandpass Type-2 chebyshev IIR filter');

//Generating bandstop Type-2 Chebyshev  IIR filter

[Hz4]=iir(N,'sb','cheb2',[fp/1.2,fp/3],[0,0.1]);
[Hw4,w4]=frmag(Hz4,256);
```

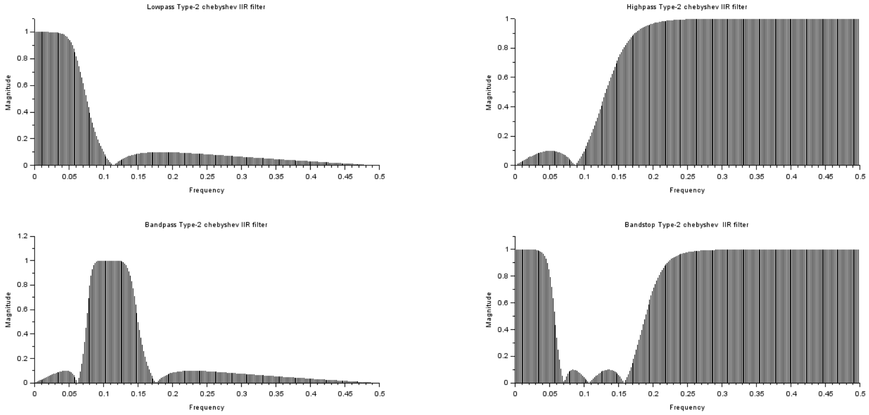

Figure 6.3: Type 2 Chebyshev Filter

```
subplot(2,2,4);
plot2d3(w4,abs(Hw4));
xlabel('Frequency');
ylabel('Magnitude');
title('Bandstop Type−2 chebyshev  IIR filter');



//Enter the cutoff frequency  fc=1000

//Enter the sampling frequency  fs=10000

//Enter the order of the filter  N=3
```

Experiment: 7

Perform Interpolation and Decimation.

Scilab code Solution 7.1 Interpolation

```
// Experiment Number : 7(a)
//Write a program to perform the interpolation of a discrete time signal
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                              Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;

```

```
//Generation of input signal

N=input ('Enter the number of points in input signal N=');
n=0:1:N-1
A=input ('Enter the amplitude of input sinusoidal signal A=');
fo=input ('Enter the frequency of input sinusoidal signal fo=');
x=A*sin(2*%pi*fo*n);
disp('Samples of input signal are:')
disp(x);

//Plotting the input signal

subplot(2,1,1);
plot2d3(n,x);
xlabel('Discrete time n');
ylabel('Amplitude');
title('Input signal x(n)');

//Generation of interpolation signal

L=input ('Enter the interpolation factor L=');
n1=1:1:L*N;
x1=[zeros(1,L*N)];
j=1:L:L*N;
x1(j)=x;
disp('Samples of interpolated signal are:')
disp(x1);

//Plotting the interpolated signal

subplot(2,1,2);
plot2d3(n1,x1);
xlabel('Discrete time n');
ylabel('Amplitude');
title('Upsampled signal x(n/L)');
```

```

//Enter the number of points in input signal  N=10

//Enter the amplitude of input sinusoidal signal  A =1

//Enter the frequency of input sinusoidal signal  fo =0.1


//"Samples of input signal are:"
//          column 1 to 3

//    0.    0.5877853    0.9510565

//          column 4 to 6

//    0.9510565    0.5877853    1.225D-16

//          column 7 to 9

//  -0.5877853  -0.9510565  -0.9510565

//          column 10

//  -0.5877853
//Enter the interpolation factor  L=2

//"Samples of interpolated signal are:"


//          column 1 to 4

//    0.    0.    0.5877853    0.

//          column 5 to 8

//    0.9510565    0.    0.9510565    0.
```

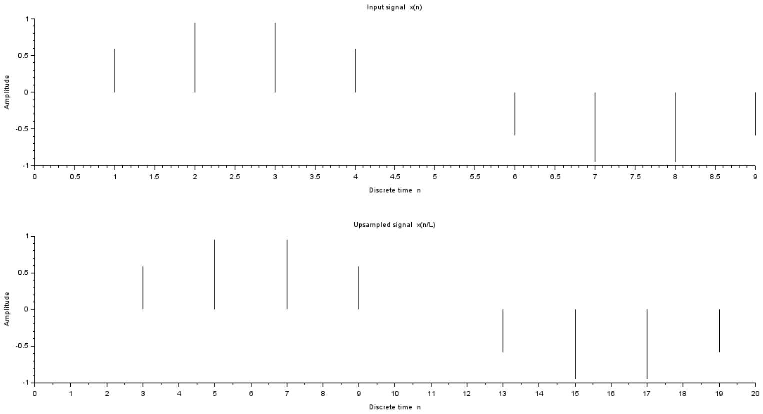

Figure 7.1: Interpolation

```

//           column 9 to 12

//    0.5877853    0.    1.225D-16    0.

//           column 13 to 16

//  -0.5877853    0.   -0.9510565    0.

//           column 17 to 20

//  -0.9510565    0.   -0.5877853    0.
```

Scilab code Solution 7.2 Decimation

```
// Experiment Number : 7(b)
//Write a program to perform the decimation of a
   discrete time signal
//Systems and Signal Processing Laboratory
```

```
//B.Tech III Year I Sem
// Student Name :                          Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad
//


// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;

//Generation of input signal

N=input ('Enter the number of points in input signal N=');
n=0:1:N-1
A=input ('Enter the amplitude of input sinusoidal signal A=');
fo=input ('Enter the frequency of input sinusoidal signal fo=');
x=A*sin(2*%pi*fo*n);
disp(x);

//Plotting the input signal

subplot(2,1,1);
plot2d3(n,x);
xlabel('Discrete time n');
ylabel('Amplitude');
title('Input signal x(n)');

//Generation of decimation signal
M=input ('Enter the decimation factor M=');
n1=1:1:N/M;
```

```
x1=x(1:M:N)
disp(x1);

// Plotting the decimation signal

subplot(2,1,2);
plot2d3(n1-1,x1);
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Decimated signal  x(Mn)');

//Enter the number of points in input signal  N=10

//Enter the amplitude of input sinusoidal signal  A=1

//Enter the frequency of input sinusoidal signal  fo=0.1



//          column 1 to 3

//    0.    0.5877853    0.9510565

//          column 4 to 6

//    0.9510565    0.5877853    1.225D-16

//          column 7 to 9

//   -0.5877853   -0.9510565   -0.9510565

//          column 10

//   -0.5877853
//Enter the decimation factor  M=2

```

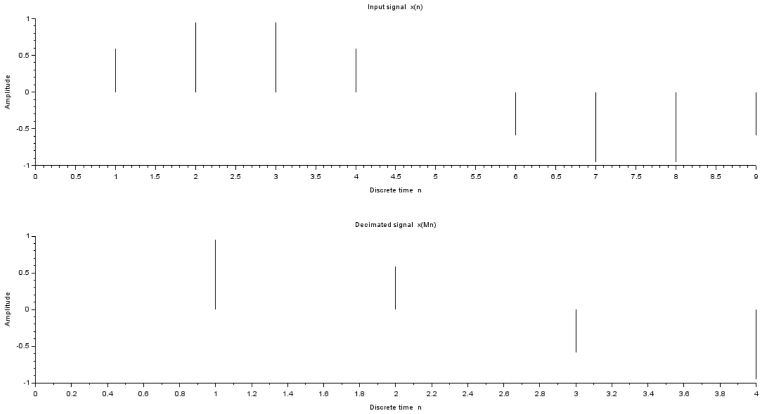

Figure 7.2: Decimation

```


//            column 1 to 3

//     0.     0.9510565     0.5877853

//            column 4 to 5

//   -0.5877853    -0.9510565
```

Experiment: 8

Perform Implementation of multi-rate systems.

Scilab code Solution 8.1 Multirate Systems

```
// Experiment Number : 8
//Write a program to implement the multirate systems with sampling rate conversion by a factor I/D or L/M
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                              Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.0.2

clc;
clear;
close;
```

```

L=input('Enter the upsampling factor  L=');
M=input('Enter the downsampling factor  M=');

//Generation of input signal

N=input ('Enter the number of points in input signal  N=');
A=input ('Enter the amplitude of input sinusoidal signal  A=');
fo=input ('Enter the frequency of input sinusoidal signal  fo=');
n=0:1:N-1
x=A*sin(2*%pi*fo*n);

//Plotting the input signal

subplot(2,1,1);
plot2d3(n,x(1:30));
xlabel('Discrete time  n');
ylabel('Amplitude');
title('Input signal  x(n)');

// Multirate system sampling rate converted by a factor L/M signal
y=intdec(x,L/M);
disp('Samples of output signal are:')


//Plotting the multirate system  sampling rate converted signal
m=0:(30*L/M)-1
subplot(2,1,2);
plot2d3(m,y(1:30*L/M));
xlabel('Discrete time  m');
ylabel('Amplitude');
title('Multirate system sampling rate converted signal');
```

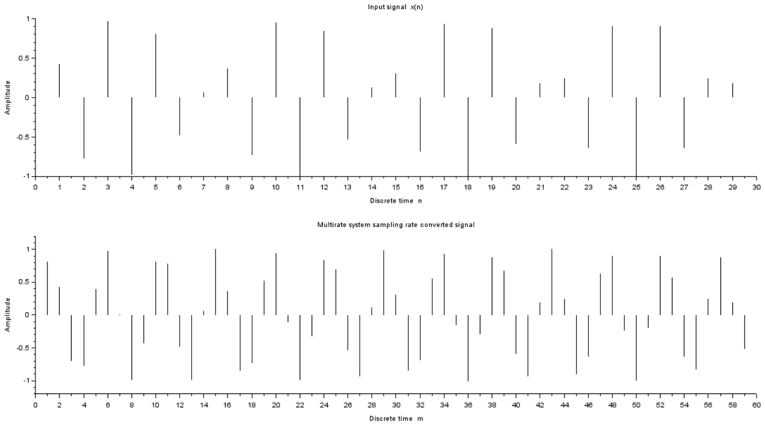

Figure 8.1: Multirate Systems

```


//Enter the upsampling factor  L=10

//Enter the downsampling factor  M=5

//Enter the number of points in input signal  N=30

//Enter the amplitude of input sinusoidal signal  A=1

//Enter the frequency of input sinusoidal signal  fo=0.43
```

Experiment: 9

Perform Time response of non-linear systems.

Scilab code Solution 9.1 Non Linear Systems

```
// Experiment Number : 9
//Write a program to find the time response of non linear systems
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                              Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.1.0

clc;
clear;
close;

```

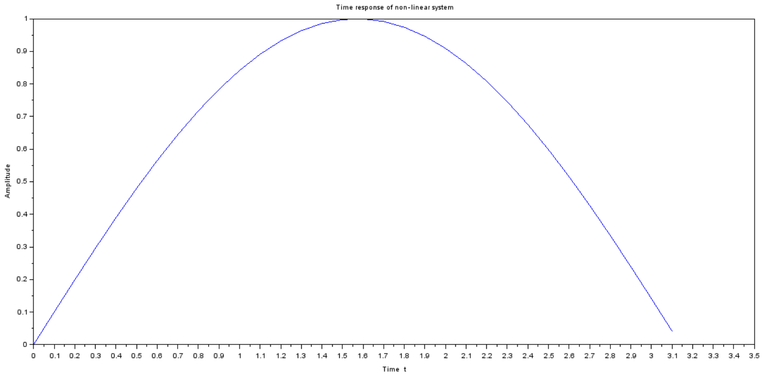

Figure 9.1: Non Linear Systems

```
//Non-linear system representation by a differential equation dy/dt=y^2-y*sin(t)+cos(t)
function ydot=f(t, y)
    ydot=y^2-y*sin(t)+cos(t)
endfunction

y0=0;//Initial condition y(0)=0
t0=0;//Initial time t(0)=0
t=0:0.1:%pi; //Time interval

//Finding the time response of non-linear system
y = ode(y0,t0,t,f);
plot(t,y) //Plotting the time response of non linear system

//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Time response of  non linear  system');
```

Experiment: 10

Design of P, PI, PD and PID controllers

Scilab code Solution 10.1 P Controller

```
// Experiment Number : 10.1
//Write a program to design P-controllers and to find its time response
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                              Enrolment Number :
// Course Instructor :K.Manohar
// Matrusri Engineering College ,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.1.0

clc;
clear;
close;

```

```
//Time Response of Open Loop System

num=poly([10],'s','coeff'); //Numerator input
den=poly([20 10 1],'s','coeff');   //Denominator input
q=syslin('c',num/den ) // Finging the transfer function of the system
t=0:0.05:2.5; //Time interval
p=csim('step',t,q); //Time Response of open loop system
subplot(211)
plot2d(t,p);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Time Response of open loop system');


//Design of P-Controller  and finding its time response

kp =300;     //Designing parameter of P-Controller
num=poly([kp], 's', 'coeff');
den=poly([20+kp 10 1], 's', 'coeff');
q=syslin('c',num/ den)  //P-Controller Transfer function
t =0:0.01:2;
p=csim('step',t,q); //Time Response of P-Controller
subplot (212)
plot2d(t,p);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Time Response of P-Controller');
```

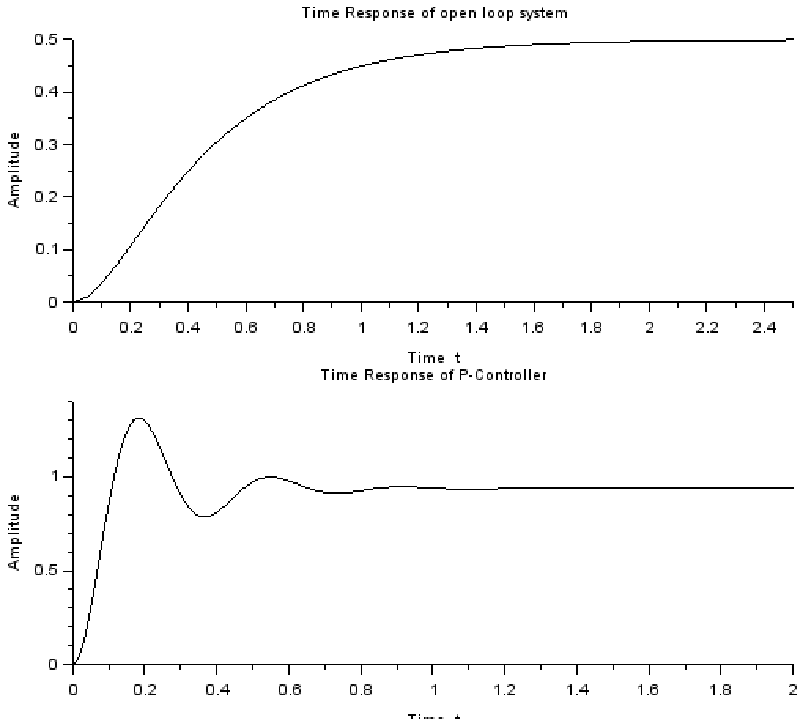

Figure 10.1: P Controller

Scilab code Solution 10.2 PI Controller

```
// Experiment Number : 10.2
//Write a program to design  PI-controllers and to find its time response
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                        Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.1.0

clc;
clear;
close;

//Time Response of Open Loop System

num=poly([10],'s','coeff'); //Numerator input
den=poly([20 10 1],'s','coeff');   //Denominator input
q=syslin('c',num/den ) // Finging the transfer function of the system
t=0:0.05:2.5; //Time interval
p=csim('step',t,q); //Time Response of open loop system
subplot(211)
plot2d(t,p);
//Labeling the axes and title for the figure
```

```
xlabel('Time  t');
ylabel('Amplitude');
title('Time Response of open loop system');





//Design of PI-Controller  and finding its time response

kp =30; //Designing parameter of PI-Controller
ki =70; //Designing parameter of PI-Controller
num=poly([ki kp], 's','coeff');
den=poly([ki 20+kp 10 1], 's', 'coeff');
q=syslin('c', num/den ) //PI-Controller Transfer function
t =0:0.01:2;
p= csim('step', t, q);  //Time Response of PI-Controller
subplot (212)
plot2d (t,p);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Time Response of PI Controller');
```

Scilab code Solution 10.3 PD Controller

```
// Experiment Number : 10.3
//Write a program to design  PD-controllers and to find its time response
//Systems and Signal Processing Laboratory
```

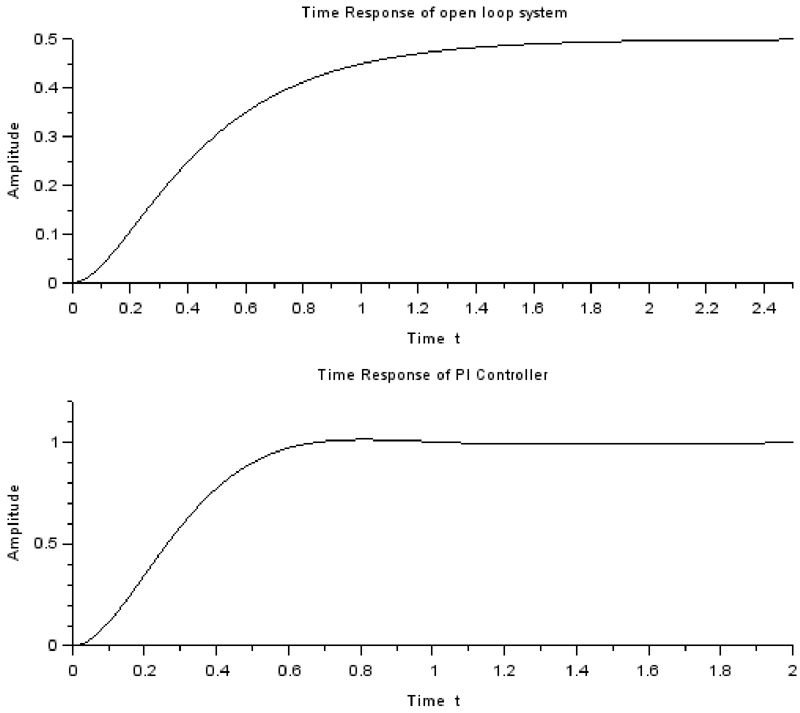

Figure 10.2: PI Controller

```
//B.Tech III Year I Sem
// Student Name :                       Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.1.0

clc;
clear;
close;

//Time Response of Open Loop System

num=poly([10],'s','coeff'); //Numerator input
den=poly([20 10 1],'s','coeff');   //Denominator input
q=syslin('c',num/den ) // Finging the transfer function of the system
t=0:0.05:2.5; //Time interval
p=csim('step',t,q); //Time Response of open loop system
subplot(211)
plot2d(t,p);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Time Response of open loop system');


//Design of PD-Controller  and finding its time response

kp =300;    //Designing parameter of PD-Controller
kd =10;    //Designing parameter of PD-Controller
```

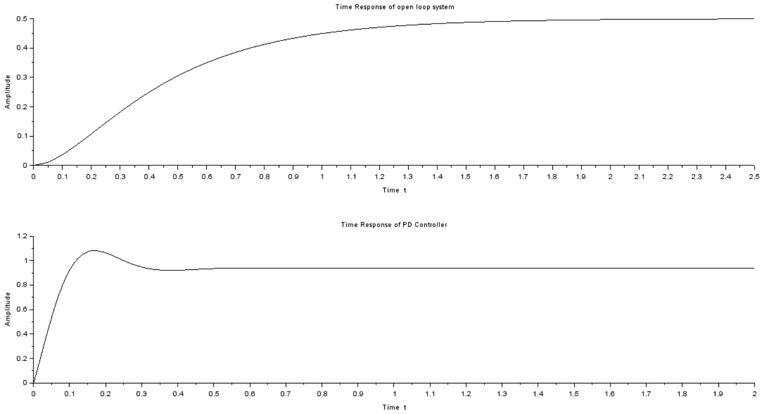

Figure 10.3: PD Controller

```
num=poly([kp kd],'s','coeff');
den=poly([20+kp 10+kd 1], 's', 'coeff');
q=syslin('c',num/den )  //PD Controller Transfer function
t=0:0.01:2;
p= csim('step',t,q);     //Time Response of PD Controller
subplot(212)
plot2d(t,p);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Time Response of PD Controller');
```

Scilab code Solution 10.4 PID Controller

```
// Experiment Number : 10.4
//Write a program to design PID controllers and to find its time response
```

```
//Systems and Signal Processing Laboratory
//B.Tech III Year I Sem
// Student Name :                         Enrolment Number :
// Course Instructor:K.Manohar
// Matrusri Engineering College,Hyderabad



// OS : Windows 10 . 1
// Scilab 6.1.0

clc;
clear;
close;

//Time Response of Open Loop System

num=poly([10],'s','coeff'); //Numerator input
den=poly([20 10 1],'s','coeff');   //Denominator input
q=syslin('c',num/den ) // Finging the transfer function of the system
t=0:0.05:2.5; //Time interval
p=csim('step',t,q); //Time Response of open loop system
subplot(211)
plot2d(t,p);
//Labeling the axes and title for the figure
xlabel('Time  t');
ylabel('Amplitude');
title('Time Response of open loop system');


//Design of PID-Controller  and finding its time response

kp =350;    //Designing parameter of PID-Controller
```

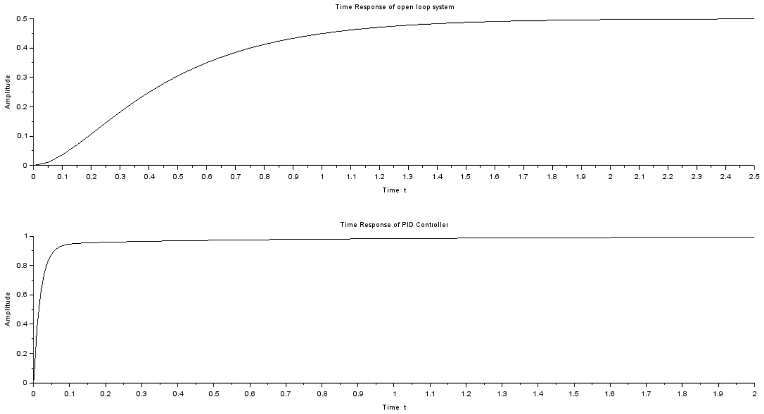

Figure 10.4: PID Controller

```
kd =50;      //Designing parameter of PID—Controller
ki =300;     //Designing parameter of PID—Controller
num=poly([ki kp kd],'s','coeff');
den=poly([ki 20+kp 10+kd 1], 's', 'coeff');
q=syslin('c',num/den)    //PID—Controller Transfer function
t=0:0.01:2;
p=csim('step',t,q); //Time Response of PID Controller
subplot(212)
plot2d(t,p);
//Labeling the axes and title for the figure
xlabel('Time t');
ylabel('Amplitude');
title('Time Response of PID Controller');
```

Appendix

Scilab code AP 1

```
//Generating the IDFT function
function x=idft(X,N)
    N=length(X);
        for k=0:1:N-1
        for n=0:1:N-1
            p=exp(%i*2*%pi*k*n/N);
            x2(k+1,n+1)=p;
        end
    end
    x=X*x2/N;
endfunction
```

IDFT Function