

Scilab Manual for  
Systems and Signal Processing Lab  
by Prof Kotha Manohar  
Electronics and Telecommunication  
Engineering  
Matrusri Engineering College<sup>1</sup>

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

April 20, 2026

<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 Introduction to SCILAB and signal generation.	5
2 Perform Fast Fourier Transform	11
3 Perform Linear convolution.	17
4 Perform Circular Convolutions.	20
5 Perform FIR filters design using different window functions.	24
6 Perform IIR filters design: Butterworth and Chebyshev.	29
7 Perform Interpolation and Decimation.	37
8 Perform Implementation of multi-rate systems.	44
9 Perform Time response of non-linear systems.	47
10 Design of P, PI, PD and PID controllers	49

# List of Experiments

Solution 1.1	Basic Signals Generation . . . . .	5
Solution 2.2	Inverse DFT . . . . .	11
Solution 2.3	FFT . . . . .	12
Solution 3.1	Linear Convolution . . . . .	17
Solution 4.1	Circular Convolution . . . . .	20
Solution 5.1	FIR Filter Design . . . . .	24
Solution 6.1	Butterworth Filter . . . . .	29
Solution 6.2	Type 1 Chebyshev Filter . . . . .	31
Solution 6.3	Type 2 Chebyshev Filter . . . . .	34
Solution 7.1	Interpolation . . . . .	37
Solution 7.2	Decimation . . . . .	40
Solution 8.1	Multirate Systems . . . . .	44
Solution 9.1	Non Linear Systems . . . . .	47
Solution 10.1	P Controller . . . . .	49
Solution 10.2	PI Controller . . . . .	52
Solution 10.3	PD Controller . . . . .	53
Solution 10.4	PID Controller . . . . .	56
AP 1	IDFT Function . . . . .	59

# List of Figures

1.1	Basic Signals Generation . . . . .	9
1.2	Basic Signals Generation . . . . .	10
1.3	Basic Signals Generation . . . . .	10
2.1	Inverse DFT . . . . .	13
2.2	FFT . . . . .	16
3.1	Linear Convolution . . . . .	19
4.1	Circular Convolution . . . . .	23
5.1	FIR Filter Design . . . . .	28
6.1	Butterworth Filter . . . . .	31
6.2	Type 1 Chebyshev Filter . . . . .	34
6.3	Type 2 Chebyshev Filter . . . . .	36
7.1	Interpolation . . . . .	40
7.2	Decimation . . . . .	43
8.1	Multirate Systems . . . . .	46
9.1	Non Linear Systems . . . . .	48
10.1	P Controller . . . . .	51
10.2	PI Controller . . . . .	54
10.3	PD Controller . . . . .	56
10.4	PID Controller . . . . .	58

# Experiment: 1

## Introduction to SCILAB and signal generation.

Scilab code Solution 1.1 Basic Signals Generation

```
1 // Experiment Number : 1
2 //Write a program to generate continuous and
   discrete time impulse , step , rectangular , saw
   tooth and sinusoidal functions
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.1.0
13
14 clc;
15 clear;
16 close;
```

```

17
18 // Continuous time Unit Impulse function generation
19 t=-4:0.1:4;
20 a=[zeros(1,40) 1 zeros(1,40)];
21 k=input("Enter the Amplitude of impulse function :")
    ); // reading amplitude value from keyboard
22 b=k*a;
23 subplot(2,2,1);
24 plot(t,b);
25 //Labeling the axes and title for the figure
26 xlabel('Time t');
27 ylabel('Amplitude');
28 title('Continuous time Impulse function');
29
30 // Discrete time Unit Impulse function generation
31 subplot(2,2,2);
32 plot2d3(t,b);
33 //Labeling the axes and title for the figure
34 xlabel('Time t');
35 ylabel('Amplitude');
36 title('Discrete time impulse function');
37
38 // Discrete time Unit step function generation
39 t=0:3;
40 y=ones(1,4);
41 subplot(2,2,3);
42 plot2d3(t,y);
43 //Labeling the axes and title for the figure
44 xlabel('Time t');
45 ylabel('Amplitude');
46 title('Discrete step function');
47
48 // Continuous time Unit step function generation
49 subplot(2,2,4);
50 plot(t,y);
51 //Labeling the axes and title for the figure
52 xlabel('Time t');
53 ylabel('Amplitude');

```

```

54 title('Continuous step function');
55
56 //Continuous time Square wave function generation
57
58 a=input("Enter the amplitude of square wave
        function :");
59 t =0:0.001:1;
60 d=a*squarewave(2* %pi *10* t);
61 figure;
62 subplot(2 ,2 ,1);
63 plot(t,d);
64 //Labeling the axes and title for the figure
65 xlabel('Time t');
66 ylabel('Amplitude');
67 title('Continuous time Square wave signal');
68
69 //Discrete time Square wave function generation
70 n=0:0.01:1;
71 d=a* squarewave (2*%pi*10*n);
72 subplot (2 ,2 ,2);
73 plot2d3 (n,d);
74 //Labeling the axes and title for the figure
75 xlabel('Time t');
76 ylabel('Amplitude');
77 title('Discrete time Square wave signal');
78
79 //Continuous time Sawtooth wave generation
80 Fs = 20; // samples per second
81 t_total = 10; // seconds
82 n_samples = Fs * t_total ;
83 t = linspace (0, t_total , n_samples );
84 f =500; // sound frequency
85 saw_wave =2*( f*t- floor (0.5+ f*t));
86 subplot(2 ,2 ,3);
87 plot(t,saw_wave );
88 //Labeling the axes and title for the figure
89 xlabel('Time t');
90 ylabel('Amplitude');

```

```

91 title('Continuous time Sawtooth wave');
92
93 //Discrete time Sawtooth wave generation
94 Fs = 20; // samples per second
95 t_total = 10; // seconds
96 n_samples = Fs * t_total ;
97 n = linspace (0, t_total , n_samples );
98 f =500; // sound frequency
99 saw_wave =2*(f*n- floor (0.5+ f*n));
100 subplot(2 ,2 ,4);
101 plot2d3(n, saw_wave );
102 //Labeling the axes and title for the figure
103 xlabel('Discrete time n');
104 ylabel('Amplitude');
105 title('Discrete time Sawtooth wave');
106
107 //Continuous time Sine wave generation
108 //a=input( Enter amplitude );
109 n=0:0.001:1;
110 d=a* sin(2*%pi*10*n);
111 figure;
112 subplot (2 ,1 ,1);
113 plot(n,d);
114 //Labeling the axes and title for the figure
115 xlabel('Time t');
116 ylabel('Amplitude');
117 title('Continuous time Sinusoidal signal');
118
119 //Discrete time Sine wave generation
120 n=0:0.009:1;
121 d=a* sin(2*%pi*10*n);
122 subplot (2 ,1 ,2);
123 plot2d3 (n,d);
124 //Labeling the axes and title for the figure
125 xlabel('Time t');
126 ylabel('Amplitude');
127 title('Discrete time sinusoidal signal');
128

```

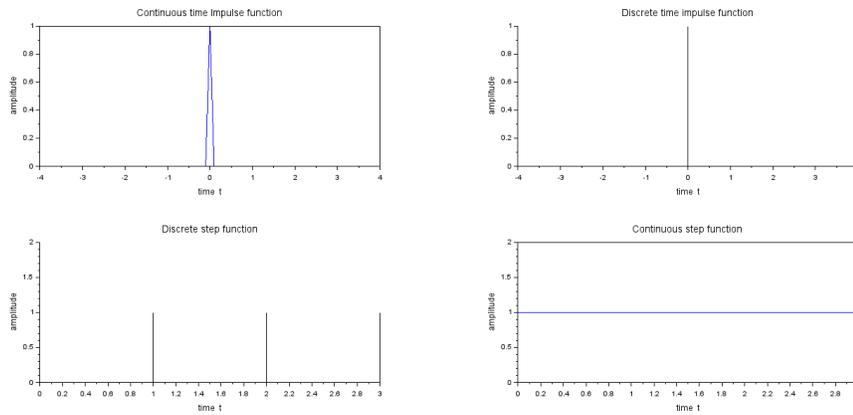


Figure 1.1: Basic Signals Generation

```

129
130 //Inputs in console window
131 //Enter the Amplitude of impulse function :1
132
133 //Enter the amplitude of square wave function :1

```

---

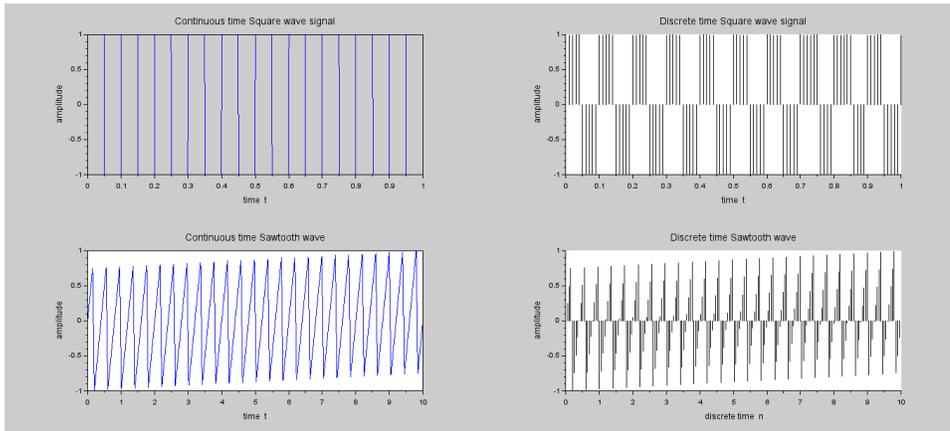


Figure 1.2: Basic Signals Generation

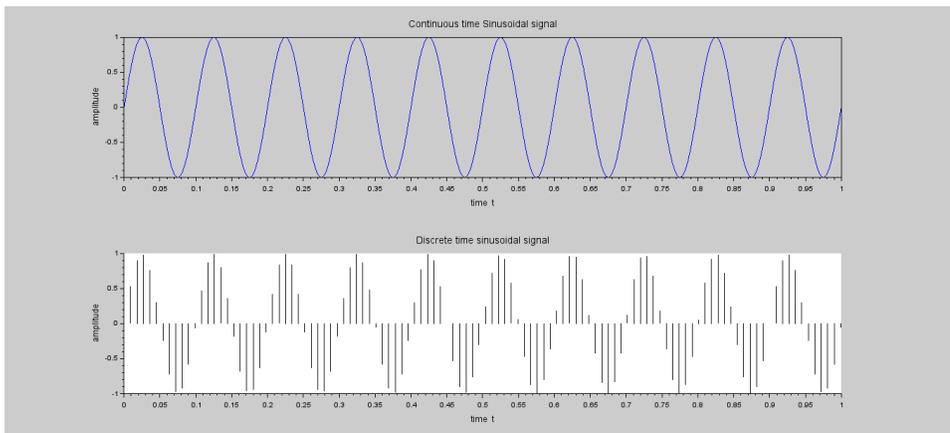


Figure 1.3: Basic Signals Generation

# Experiment: 2

## Perform Fast Fourier Transform

check Appendix [AP 1](#) for dependency:

```
idftfunction.sci
```

### Scilab code Solution 2.2 Inverse DFT

```
1 // Experiment Number : 2.2
2 //Write a program to find the IDFT of a discrete
   Frequency Domain signal
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14
15 clc;
```

```

16 clear;
17 close;
18
19 X=input('Enter the frequency domain signal X=');
20 N=length(X);
21 x=idft(X,N);
22 disp('IDFT of X(K) is ')
23 disp(x);
24 n=0:1:N-1
25 plot2d3(n,x);
26 xlabel('Discrete time n');
27 ylabel('Amplitude');
28 title('IDFT or time domain signal x(n)');
29
30
31
32
33 // Enter the frequency domain signal X=[4 0 0 0]
34
35 //”IDFT of X(K) is”
36 // 1. 1. 1. 1.

```

---

### Scilab code Solution 2.3 FFT

```

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

```

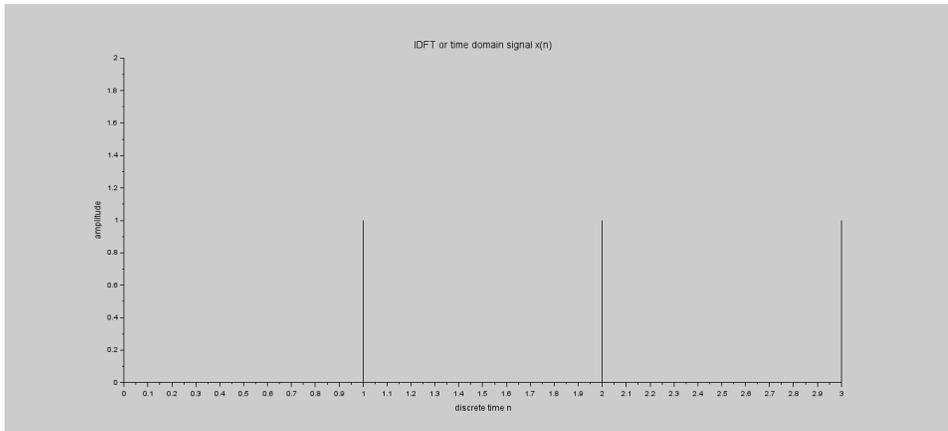


Figure 2.1: Inverse DFT

```

8 //
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 //Input time domain signal
19
20 x=input ('Enter the time domain sequence x=');
21 N=length(x); //Finding the length of input signal
22
23 //FFT of a signal
24 y=fft(x); //Finding FFT of a sequence
25 disp('Frequency domain signal is ');
26 disp(y);
27 y1=abs(y); //Finding the magnitude response
28 disp('Magnitude Response is ');
29 disp(y1);
30 y2=atan(imag(y),real(y)); //Finding the phase

```

```

        response
31 disp('Phase Response is ');
32 disp(y2);
33
34 //Plotting the input time domain signal
35
36 n=0:1:N-1
37 subplot(2,2,1);
38 plot2d3(n,x);
39 xlabel('Discrete time n');
40 ylabel('Amplitude');
41 title('Input time domain signal');
42
43 //Plotting the magnitude spectrum
44
45 k=0:1:N-1
46 subplot(2,2,2);
47 plot2d3(k,y1);
48 xlabel('Discrete frequency k');
49 ylabel('Amplitude');
50 title('Magnitude spectrum of FFT signal');
51
52 //Plotting the phase spectrum
53
54 subplot(2,2,3);
55 plot2d3(k,y2);
56 xlabel('Discrete frequency k');
57 ylabel('Phase angle');
58 title('Phase spectrum of FFT signal');
59
60 // Finding Inverse Fast Fourier Transform
61 z=ifft(y);
62 disp('Inverse Fast Fourier Transform is ');
63 disp(z);
64
65 //Plotting Inverse FFT signal
66
67 subplot(2,2,4);

```

```

68 plot2d3(n,z);
69 xlabel('Discrete time n');
70 ylabel('Amplitude');
71 title('Inverse FFT or time domain signal');
72
73
74 //Enter the time domain sequence x=[1 2 3 4]
75
76
77 // Frequency domain signal is
78
79 // 10. -2. + 2.i -2. -2. - 2.i
80
81 // Magnitude Response is
82
83 // 10. 2.8284271 2. 2.8284271
84
85 // Phase Response is
86
87
88 // column 1 to 3
89
90 // 0. 2.3561945 3.1415927
91
92 // column 4
93
94 // -2.3561945
95
96 // Inverse Fast Fourier Transform is
97
98 // 1. 2. 3. 4.

```

---

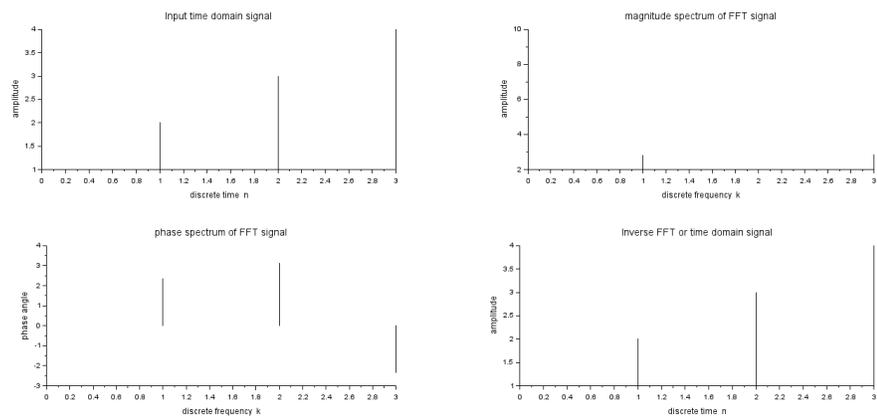


Figure 2.2: FFT

# Experiment: 3

## Perform Linear convolution.

### Scilab code Solution 3.1 Linear Convolution

```
1 // Experiment Number : 3
2 //Write a program to find the linear convolution of
   two discrete time signals
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 //Generating and plotting the input signal
19
```

```

20 x=input('Enter the time domain sequence x=');
21 N1=length(x); //Finding the length of input signal
22 n1=0:1:N1-1
23 subplot(3,1,1);
24 plot2d3(n1,x);
25 xlabel('Discrete time n');
26 ylabel('Amplitude');
27 title('Time domain input signal x(n)');
28
29 //Generating and plotting the impulse response of
    the system
30
31 h=input('Enter the impulse response of the system h
    =');
32 N2=length(h);
33 n2=0:1:N2-1
34 subplot(3,1,2);
35 plot2d3(n2,h);
36 xlabel('Discrete time n');
37 ylabel('Amplitude');
38 title('Impulse response of the system h(n)');
39
40 //Generating and plotting the linear convolution of
    two discrete time signals
41 y=conv(x,h);
42 disp('Linear Convolution of input signal and impulse
    response of the system is:')
43 disp(y);
44 N=length(y);
45 n3=0:1:N-1
46 subplot(3,1,3);
47 plot2d3(n3,y);
48 xlabel('Discrete time n3');
49 ylabel('Amplitude');
50 title('Linear convolution function y(n3)');
51
52
53

```

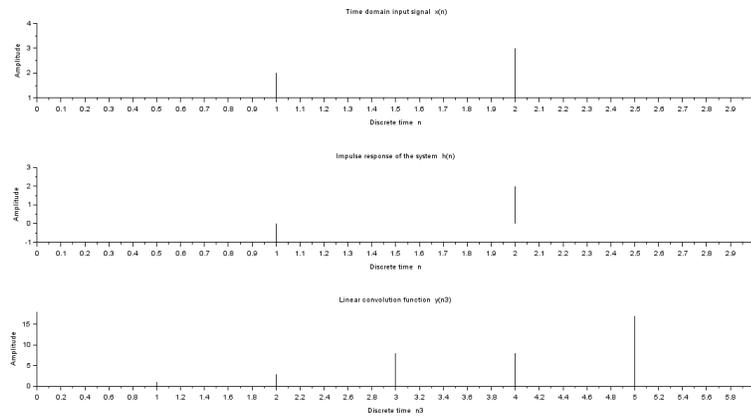


Figure 3.1: Linear Convolution

```

54
55 //Enter the time domain sequence  x=[1 2 3 4]
56
57 //Enter the impulse response of the system  h=[1 -1
    2 3]
58
59
60 //  "Linear Convolution of input signal and impulse
    response of the system is:"
61
62 //  1.   1.   3.   8.   8.   17.   12.

```

---

# Experiment: 4

## Perform Circular Convolutions.

### Scilab code Solution 4.1 Circular Convolution

```
1 // Experiment Number : 4
2 //Write a program to find the Circular convolution
   of two discrete time signals
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 //Generating and plotting the input signal
19
```

```

20 x=input('Enter the time domain sequence x=');
21 disp(x);
22 N1=length(x); //finding the length of input signal
23 n1=0:1:N1-1
24 subplot(3,1,1);
25 plot2d3(n1,x);
26 xlabel('Discrete time n');
27 ylabel('Amplitude');
28 title('Time domain input signal x(n)');
29
30 //Generating and plotting the impulse response of
    the system
31
32 h=input('Enter the impulse response of the system h
    =');
33 disp(h);
34 N2=length(h);
35 n2=0:1:N2-1
36 subplot(3,1,2);
37 plot2d3(n2,h);
38 xlabel('Discrete time n');
39 ylabel('Amplitude');
40 title('Impulse response of the system h(n)');
41
42 //Generating and plotting the circular convolution
    of two discrete time signals
43
44 N3=max(N1,N2);
45 x1=[x,zeros(1,N3-N1)]; //Zero padding to input
    sequence
46 h1=[h,zeros(1,N3-N2)]; //Zero padding to impulse
    response sequence
47 X1=fft(x1,-1); //DFT Computation of input
48 H1=fft(h1,-1); //DFT Computation of impulse
    response
49 Y=X1.*H1; //output sequence in frequency domain
50 y=fft(Y,1) ; //IDFT Computation or output
    sequence in time domain

```

```

51 disp('Circular Convolution Sequence y[n]=',y);
52 n3=0:1:N3-1
53 subplot(3,1,3);
54 plot2d3(n3,y);
55 xlabel('Discrete time n3');
56 ylabel('Amplitude');
57 title('Circular convolution function y(n3)');
58
59
60
61
62 //Enter the time domain sequence x=[1 2 3 4]
63
64
65 // 1. 2. 3. 4.
66 //Enter the impulse response of the system h=[2 4
67 1]
68
69 // 2. 4. 1.
70
71 // "Circular Convolution Sequence y[n]="
72
73 // 21. 12. 15. 22.

```

---

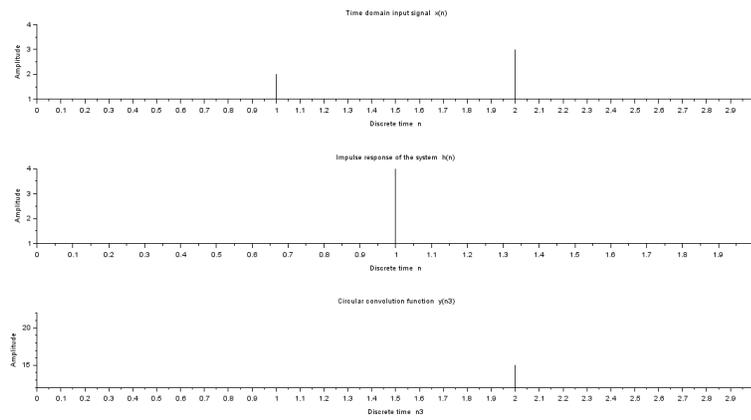


Figure 4.1: Circular Convolution

# Experiment: 5

## Perform FIR filters design using different window functions.

Scilab code Solution 5.1 FIR Filter Design

```
1 // Experiment Number : 5
2 //Write a program to generate Lowpass FIR Filter
   using different window techniques
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.1.0
13
14
15 clc;
```

```

16 clear;
17 close;
18
19 fc=input ('Enter the cutoff frequency fc=');
20 fs=input ('Enter the sampling frequency fs=');
21 N=input ('Enter the order of filter N=');
22
23 //Finding the magnitude response of Lowpass FIR
    Filter using Rectangular window technique
24
25 w1 =(2*%pi)*(fc/fs);
26 disp('Digital cutoff frequency in radians:');
27 disp(w1);
28 wc1 =w1/%pi;
29 disp('Normalized digital cutoff frequency in radians
    :');
30 disp(wc1);
31 [wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0], 're',[0,0]);
    //filter design
32 disp('Impulse response of Lowpass FIR filter using
    Rectangular window technique:h(n)=');
33 disp(wft);
34 a=gca();
35 subplot(2,2,1);
36 plot(2*fr,wfm);
37 //Labeling the axes and title for the figure
38 xlabel('Normalized digital frequency w');
39 ylabel('Magnitude ');
40 title('Magnitude response of Lowpass FIR Filter
    using Rectangular window');
41
42 //Finding the magnitude response of Lowpass FIR
    Filter using Triangular window technique
43
44 [wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0], 'tr',[0,0]);
    //filter design
45 disp('Impulse response of Lowpass FIR filter using
    Triangular window technique:h(n)=');

```

```

46 disp(wft);
47 a=gca();
48 subplot(2,2,2);
49 plot(2*fr,wfm);
50 //Labeling the axes and title for the figure
51 xlabel('Normalized digital frequency w');
52 ylabel('Magnitude ');
53 title('Magnitude response of Lowpass FIR Filter
        using Triangular window');
54
55 //Finding the magnitude response of Lowpass FIR
    Filter using Hamming window technique
56
57 [wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0],'hm',[0,0]);
    //filter design
58 disp('Impulse response of Lowpass FIR filter using
        Hamming window technique:h(n)=');
59 disp(wft);
60 a=gca();
61 subplot(2,2,3);
62 plot(2*fr,wfm);
63 //Labeling the axes and title for the figure
64 xlabel('Normalized digital frequency w');
65 ylabel('Magnitude ');
66 title('Magnitude response of Lowpass FIR Filter
        using Hamming window ');
67
68 //Finding the magnitude response of Lowpass FIR
    Filter using Hanning window technique
69
70 [wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0],'hn',[0,0]);
    //filter design
71 disp('Impulse response of Lowpass FIR filter using
        Hanning window technique :h(n)=');
72 disp(wft);
73 a=gca();
74 subplot(2,2,4);
75 plot(2*fr,wfm);

```

```

76 //Labeling the axes and title for the figure
77 xlabel('Normalized digital frequency w');
78 ylabel('Magnitude ');
79 title('Magnitude response of Lowpass FIR Filter
      using Hanning window');
80
81
82 //Enter the cutoff frequency fc=1200
83
84 //Enter the sampling frequency fs=10000
85
86 //Enter the order of filter N=3
87
88
89 // "Digital cutoff frequency in radians:"
90
91 // 0.7539822
92
93 // "Normalized digital cutoff frequency in radians
   :":
94
95 // 0.24
96
97 // "Impulse response of Lowpass FIR filter using
   Rectangular window technique:h(n)="
98
99 // 0.1920103 0.2343554 0.2343554 0.1920103
100
101 // "Impulse response of Lowpass FIR filter using
   Triangular window technique:h(n)="
102
103 // 0.0768041 0.1874843 0.1874843 0.0768041
104
105 // "Impulse response of Lowpass FIR filter using
   Hamming window technique:h(n)="
106
107 // 0.0153608 0.1804536 0.1804536 0.0153608
108

```

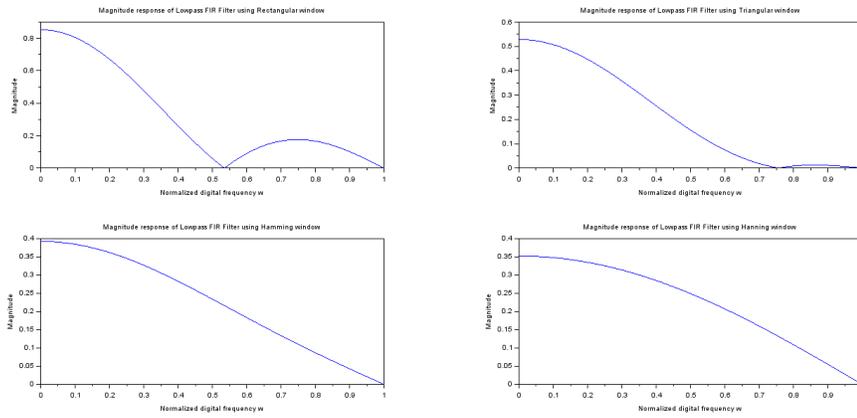


Figure 5.1: FIR Filter Design

```

109 // "Impulse response of Lowpass FIR filter using
110 // Hanning window technique :h(n)="
111 // 0. 0.1757665 0.1757665 0.

```

---

# Experiment: 6

## Perform IIR filters design: Butterworth and Chebyshev.

Scilab code Solution 6.1 Butterworth Filter

```
1 // Experiment Number : 6.1
2 //Write a program to generate Butterworth lowpass ,
   highpass , bandpass and bandstop IIR Filter
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8 //
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
```

```

18 fc=input ('Enter the cutoff frequency fc=');
19 fs=input ('Enter the sampling frequency fs=');
20 N=input ('Enter the order of the filter N=');
21 fp=2*fc/fs;
22
23 //Generating lowpass Butterworth IIR filter
24
25 [Hz1]=iir(N, 'lp', 'butt', [fp/2,0], [0,0]);
26 [Hw1,w1]=frmag(Hz1,256);
27 subplot(2,2,1);
28 plot2d3(w1,abs(Hw1));
29 xlabel('Frequency');
30 ylabel('Magnitude');
31 title('Butterworth lowpass IIR filter');
32
33
34 //Generating Highpass Butterworth IIR filter
35
36 [Hz2]=iir(N, 'hp', 'butt', [fp/2,0], [0,0]);
37 [Hw2,w2]=frmag(Hz2,256);
38 subplot(2,2,2);
39 plot2d3(w2,abs(Hw2));
40 xlabel('Frequency');
41 ylabel('Magnitude');
42 title('Butterworth highpass IIR filter');
43
44 //Generating Bandpass Butterworth IIR filter
45
46 [Hz3]=iir(N, 'bp', 'butt', [fp/1.2,fp/3], [0,0]);
47 [Hw3,w3]=frmag(Hz3,256);
48 subplot(2,2,3);
49 plot2d3(w3,abs(Hw3));
50 xlabel('Frequency');
51 ylabel('Magnitude');
52 title('Butterworth Bandpass IIR filter');
53
54 //Generating bandstop Butterworth IIR filter
55

```

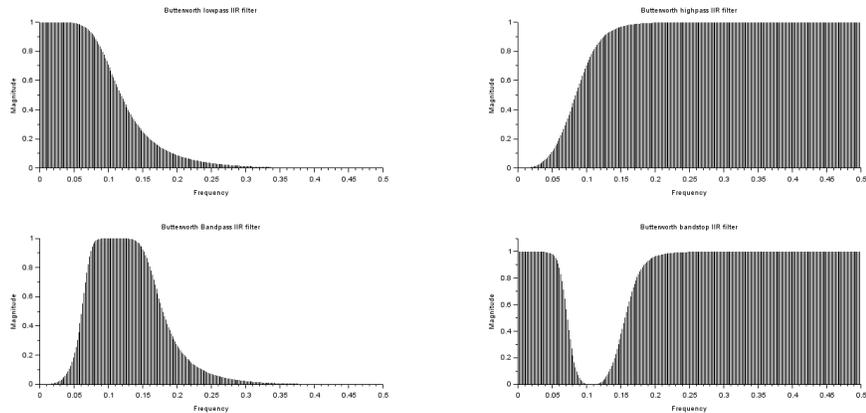


Figure 6.1: Butterworth Filter

```

56 [Hz4]=iir(N, 'sb', 'butt', [fp/1.2, fp/3], [0,0]);
57 [Hw4,w4]=frmag(Hz4,256);
58 subplot(2,2,4);
59 plot2d3(w4, abs(Hw4));
60 xlabel('Frequency');
61 ylabel('Magnitude');
62 title('Butterworth bandstop IIR filter');
63
64
65
66 //Enter the cutoff frequency fc=1000
67
68 //Enter the sampling frequency fs=10000
69
70 //Enter the order of the filter N=3

```

---

## Scilab code Solution 6.2 Type 1 Chebyshev Filter

```

1 // Experiment Number : 6.2

```

```

2 //Write a program to generate Type-1 Chebyshev
   lowpass, highpass, bandpass and bandstop IIR
   Filter
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor:K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8 //
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 fc=input ('Enter the cutoff frequency fc=');
19 fs=input ('Enter the sampling frequency fs=');
20 N=input ('Enter the order of the filter N=');
21 fp=2*fc/fs;
22
23 //Generating lowpass Type-1 Chebyshev IIR filter
24
25 [Hz1]=iir(N, 'lp', 'cheb1', [fp/2, 0], [0.2, 0]);
26 [Hw1, w1]=frmag(Hz1, 256);
27 subplot(2, 2, 1);
28 plot2d3(w1, abs(Hw1));
29 xlabel('Frequency');
30 ylabel('Magnitude');
31 title(' Lowpass Type-I chebyshev IIR filter ');
32
33 //Generating Highpass Type-1 Chebyshev IIR filter
34
35 [Hz2]=iir(N, 'hp', 'cheb1', [fp/2, 0], [0.2, 0]);
36 [Hw2, w2]=frmag(Hz2, 256);

```

```

37 subplot(2,2,2);
38 plot2d3(w2,abs(Hw2));
39 xlabel('Frequency');
40 ylabel('Magnitude');
41 title(' Highpass Type-I chebyshev IIR filter ');
42
43 //Generating Bandpass Type-1 Chebyshev IIR filter
44
45 [Hz3]=iir(N,'bp','cheb1',[fp/1.2,fp/3],[0.2,0]);
46 [Hw3,w3]=frmag(Hz3,256);
47 subplot(2,2,3);
48 plot2d3(w3,abs(Hw3));
49 xlabel('Frequency');
50 ylabel('Magnitude');
51 title(' Bandpass Type-I chebyshev IIR filter ');
52
53 //Generating bandstop Type-1 Chebyshev IIR filter
54
55 [Hz4]=iir(N,'sb','cheb1',[fp/1.2,fp/3],[0.2,0]);
56 [Hw4,w4]=frmag(Hz4,256);
57 subplot(2,2,4);
58 plot2d3(w4,abs(Hw4));
59 xlabel('Frequency');
60 ylabel('Magnitude');
61 title('Bandstop Type-I chebyshev IIR filter ');
62
63
64
65 //Enter the cutoff frequency fc=1000
66
67 //Enter the sampling frequency fs=10000
68
69 //Enter the order of the filter N=3

```

---

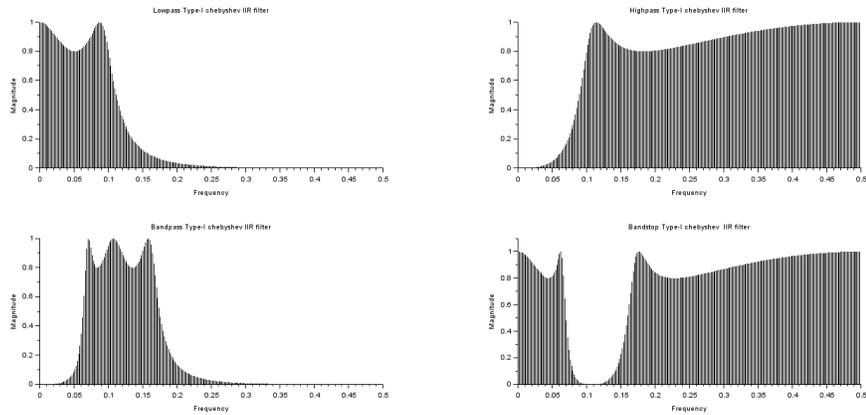


Figure 6.2: Type 1 Chebyshev Filter

### Scilab code Solution 6.3 Type 2 Chebyshev Filter

```

1 // Experiment Number : 6.3
2 //Write a program to generate Type-2 Chebyshev
  lowpass , highpass , bandpass and bandstop IIR
  Filter
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
  Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8 //
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 fc=input ('Enter the cutoff frequency fc=');

```

```

19 fs=input ('Enter the sampling frequency fs=');
20 N=input ('Enter the order of the filter N=');
21 fp=2*fc/fs;
22
23 //Generating lowpass Type-2 Chebyshev IIR filter
24
25 [Hz1]=iir(N, 'lp', 'cheb2', [fp/2, 0], [0, 0.1]);
26 [Hw1, w1]=frmag(Hz1, 256);
27 subplot(2, 2, 1);
28 plot2d3(w1, abs(Hw1));
29 xlabel('Frequency');
30 ylabel('Magnitude');
31 title(' Lowpass Type-2 chebyshev IIR filter ');
32
33 //Generating Highpass Type-2 Chebyshev IIR filter
34
35 [Hz2]=iir(N, 'hp', 'cheb2', [fp/2, 0], [0, 0.1]);
36 [Hw2, w2]=frmag(Hz2, 256);
37 subplot(2, 2, 2);
38 plot2d3(w2, abs(Hw2));
39 xlabel('Frequency');
40 ylabel('Magnitude');
41 title(' Highpass Type-2 chebyshev IIR filter ');
42
43 //Generating Bandpass Type-2 Chebyshev IIR filter
44
45 [Hz3]=iir(N, 'bp', 'cheb2', [fp/1.2, fp/3], [0, 0.1]);
46 [Hw3, w3]=frmag(Hz3, 256);
47 subplot(2, 2, 3);
48 plot2d3(w3, abs(Hw3));
49 xlabel('Frequency');
50 ylabel('Magnitude');
51 title(' Bandpass Type-2 chebyshev IIR filter ');
52
53 //Generating bandstop Type-2 Chebyshev IIR filter
54
55 [Hz4]=iir(N, 'sb', 'cheb2', [fp/1.2, fp/3], [0, 0.1]);
56 [Hw4, w4]=frmag(Hz4, 256);

```

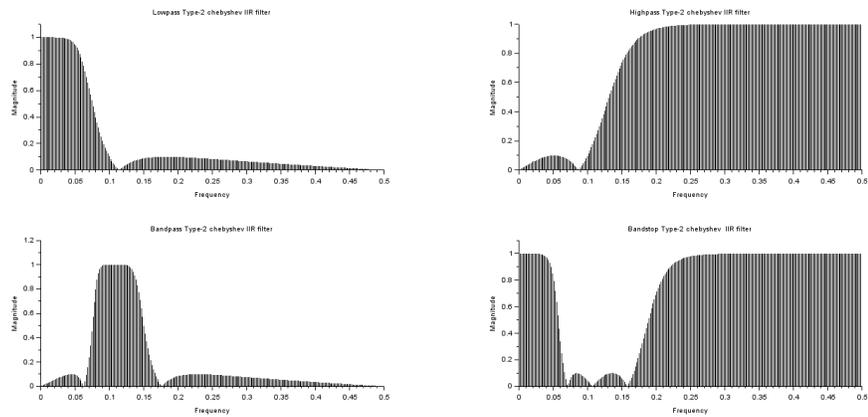


Figure 6.3: Type 2 Chebyshev Filter

```

57 subplot(2,2,4);
58 plot2d3(w4,abs(Hw4));
59 xlabel('Frequency');
60 ylabel('Magnitude');
61 title('Bandstop Type-2 chebyshev IIR filter');
62
63
64
65 //Enter the cutoff frequency fc=1000
66
67 //Enter the sampling frequency fs=10000
68
69 //Enter the order of the filter N=3

```

---

# Experiment: 7

## Perform Interpolation and Decimation.

Scilab code Solution 7.1 Interpolation

```
1 // Experiment Number : 7(a)
2 //Write a program to perform the interpolation of a
  discrete time signal
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
  Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
```

```

18 //Generation of input signal
19
20 N=input ('Enter the number of points in input signal
           N=');
21 n=0:1:N-1
22 A=input ('Enter the amplitude of input sinusoidal
           signal A=');
23 fo=input ('Enter the frequency of input sinusoidal
           signal fo=');
24 x=A*sin(2*%pi*fo*n);
25 disp('Samples of input signal are:')
26 disp(x);
27
28 //Plotting the input signal
29
30 subplot(2,1,1);
31 plot2d3(n,x);
32 xlabel('Discrete time n');
33 ylabel('Amplitude');
34 title('Input signal x(n)');
35
36 //Generation of interpolation signal
37
38 L=input ('Enter the interpolation factor L=');
39 n1=1:1:L*N;
40 x1=[zeros(1,L*N)];
41 j=1:L:L*N;
42 x1(j)=x;
43 disp('Samples of interpolated signal are:')
44 disp(x1);
45
46 //Plotting the interpolated signal
47
48 subplot(2,1,2);
49 plot2d3(n1,x1);
50 xlabel('Discrete time n');
51 ylabel('Amplitude');
52 title('Upsampled signal x(n/L)');

```

```

53
54 //Enter the number of points in input signal N=10
55
56 //Enter the amplitude of input sinusoidal signal A
    =1
57
58 //Enter the frequency of input sinusoidal signal fo
    =0.1
59
60
61 //”Samples of input signal are:”
62 //          column 1 to 3
63
64 //    0.    0.5877853    0.9510565
65
66 //          column 4 to 6
67
68 //    0.9510565    0.5877853    1.225D-16
69
70 //          column 7 to 9
71
72 //   -0.5877853   -0.9510565   -0.9510565
73
74 //          column 10
75
76 //   -0.5877853
77 //Enter the interpolation factor L=2
78
79 //”Samples of interpolated signal are:”
80
81
82 //          column 1 to 4
83
84 //    0.    0.    0.5877853    0.
85
86 //          column 5 to 8
87
88 //    0.9510565    0.    0.9510565    0.

```

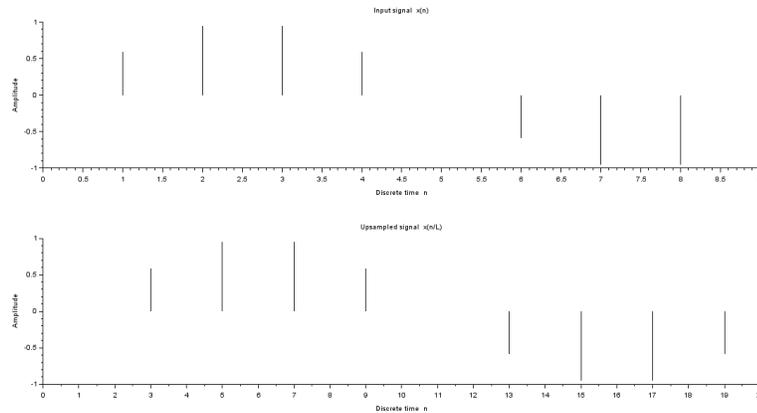


Figure 7.1: Interpolation

```

89
90 //          column 9 to 12
91
92 //    0.5877853    0.    1.225D-16    0.
93
94 //          column 13 to 16
95
96 //   -0.5877853    0.   -0.9510565    0.
97
98 //          column 17 to 20
99
100 //   -0.9510565    0.   -0.5877853    0.

```

---

### Scilab code Solution 7.2 Decimation

```

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

```

```

4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8 //
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 //Generation of input signal
19
20 N=input ('Enter the number of points in input signal
   N=');
21 n=0:1:N-1
22 A=input ('Enter the amplitude of input sinusoidal
   signal A=');
23 fo=input ('Enter the frequency of input sinusoidal
   signal fo=');
24 x=A*sin(2*%pi*fo*n);
25 disp(x);
26
27 //Plotting the input signal
28
29 subplot(2,1,1);
30 plot2d3(n,x);
31 xlabel('Discrete time n');
32 ylabel('Amplitude');
33 title('Input signal x(n)');
34
35 //Generation of decimation signal
36 M=input ('Enter the decimation factor M=');
37 n1=1:1:N/M;

```

```

38 x1=x(1:M:N)
39 disp(x1);
40
41 // Plotting the decimation signal
42
43 subplot(2,1,2);
44 plot2d3(n1-1,x1);
45 xlabel('Discrete time n');
46 ylabel('Amplitude');
47 title('Decimated signal x(Mn)');
48
49 //Enter the number of points in input signal N=10
50
51 //Enter the amplitude of input sinusoidal signal A
    =1
52
53 //Enter the frequency of input sinusoidal signal fo
    =0.1
54
55
56
57 //          column 1 to 3
58
59 //    0.    0.5877853    0.9510565
60
61 //          column 4 to 6
62
63 //    0.9510565    0.5877853    1.225D-16
64
65 //          column 7 to 9
66
67 //   -0.5877853   -0.9510565   -0.9510565
68
69 //          column 10
70
71 //   -0.5877853
72 //Enter the decimation factor M=2
73

```

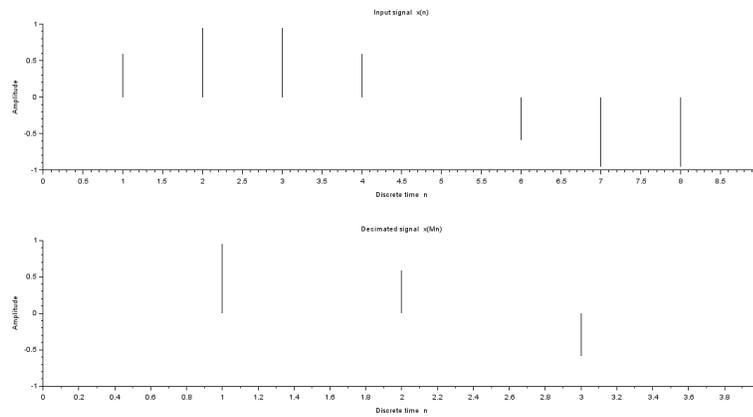


Figure 7.2: Decimation

```

74
75
76 //          column 1 to 3
77
78 //    0.    0.9510565    0.5877853
79
80 //          column 4 to 5
81
82 //   -0.5877853   -0.9510565

```

---

# Experiment: 8

## Perform Implementation of multi-rate systems.

Scilab code Solution 8.1 Multirate Systems

```
1 // Experiment Number : 8
2 //Write a program to implement the multirate systems
   with sampling rate conversion by a factor I/D or
   L/M
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
```

```

17
18 L=input('Enter the upsampling factor L=');
19 M=input('Enter the downsampling factor M=');
20
21 //Generation of input signal
22
23 N=input ('Enter the number of points in input signal
           N=');
24 A=input ('Enter the amplitude of input sinusoidal
           signal A=');
25 fo=input ('Enter the frequency of input sinusoidal
           signal fo=');
26 n=0:1:N-1
27 x=A*sin(2*%pi*fo*n);
28
29 //Plotting the input signal
30
31 subplot(2,1,1);
32 plot2d3(n,x(1:30));
33 xlabel('Discrete time n');
34 ylabel('Amplitude');
35 title('Input signal x(n)');
36
37 // Multirate system sampling rate converted by a
           factor L/M signal
38 y=intdec(x,L/M);
39 disp('Samples of output signal are:')
40
41
42 //Plotting the multirate system sampling rate
           converted signal
43 m=0:(30*L/M)-1
44 subplot(2,1,2);
45 plot2d3(m,y(1:30*L/M));
46 xlabel('Discrete time m');
47 ylabel('Amplitude');
48 title('Multirate system sampling rate converted
           signal');

```

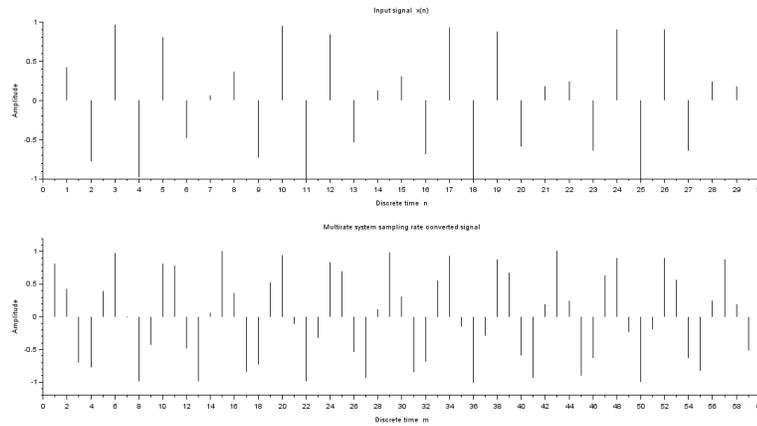


Figure 8.1: Multirate Systems

```

49
50
51 //Enter the upsampling factor L=10
52
53 //Enter the downsampling factor M=5
54
55 //Enter the number of points in input signal N=30
56
57 //Enter the amplitude of input sinusoidal signal A
    =1
58
59 //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

```
1 // Experiment Number : 9
2 //Write a program to find the time response of
   n o n linear systems
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.1.0
13
14 clc;
15 clear;
16 close;
17
```

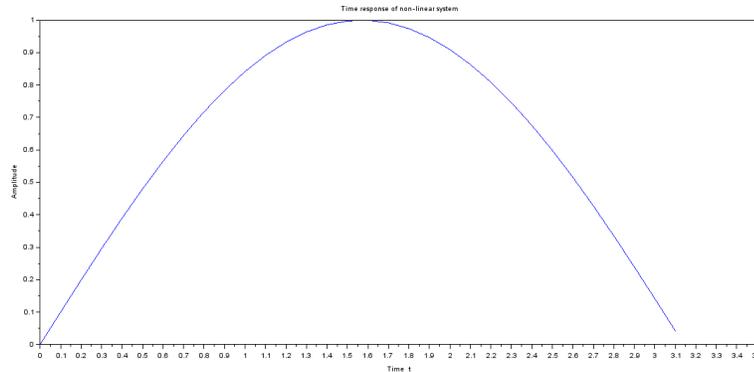


Figure 9.1: Non Linear Systems

```

18 //Non-linear system representation by a differential
    equation  $dy/dt=y^2-y*\sin(t)+\cos(t)$ 
19 function ydot=f(t, y)
20     ydot=y^2-y*sin(t)+cos(t)
21 endfunction
22
23 y0=0; //Initial condition  $y(0)=0$ 
24 t0=0; //Initial time  $t(0)=0$ 
25 t=0:0.1:%pi; //Time interval
26
27 //Finding the time response of non-linear system
28 y = ode(y0,t0,t,f);
29 plot(t,y) //Plotting the time response of non linear
    system
30
31 //Labeling the axes and title for the figure
32 xlabel('Time t');
33 ylabel('Amplitude');
34 title('Time response of non linear system');

```

---

# Experiment: 10

## Design of P, PI, PD and PID controllers

Scilab code Solution 10.1 P Controller

```
1 // Experiment Number : 10.1
2 //Write a program to design P-controllers and to
  find its time response
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
  Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.1.0
13
14 clc;
15 clear;
16 close;
17
```

```

18 //Time Response of Open Loop System
19
20 num=poly([10], 's', 'coeff'); //Numerator input
21 den=poly([20 10 1], 's', 'coeff'); //Denominator
    input
22 q=syslin('c',num/den ) // Finding the transfer
    function of the system
23 t=0:0.05:2.5; //Time interval
24 p=csim('step',t,q); //Time Response of open loop
    system
25 subplot(211)
26 plot2d(t,p);
27 //Labeling the axes and title for the figure
28 xlabel('Time t');
29 ylabel('Amplitude');
30 title('Time Response of open loop system');
31
32
33 //Design of P-Controller and finding its time
    response
34
35 kp =300; //Designing parameter of P-Controller
36 num=poly([kp], 's', 'coeff');
37 den=poly([20+kp 10 1], 's', 'coeff');
38 q=syslin('c',num/ den) //P-Controller Transfer
    function
39 t =0:0.01:2;
40 p=csim('step',t,q); //Time Response of P-Controller
41 subplot (212)
42 plot2d(t,p);
43 //Labeling the axes and title for the figure
44 xlabel('Time t');
45 ylabel('Amplitude');
46 title('Time Response of P-Controller');

```

---

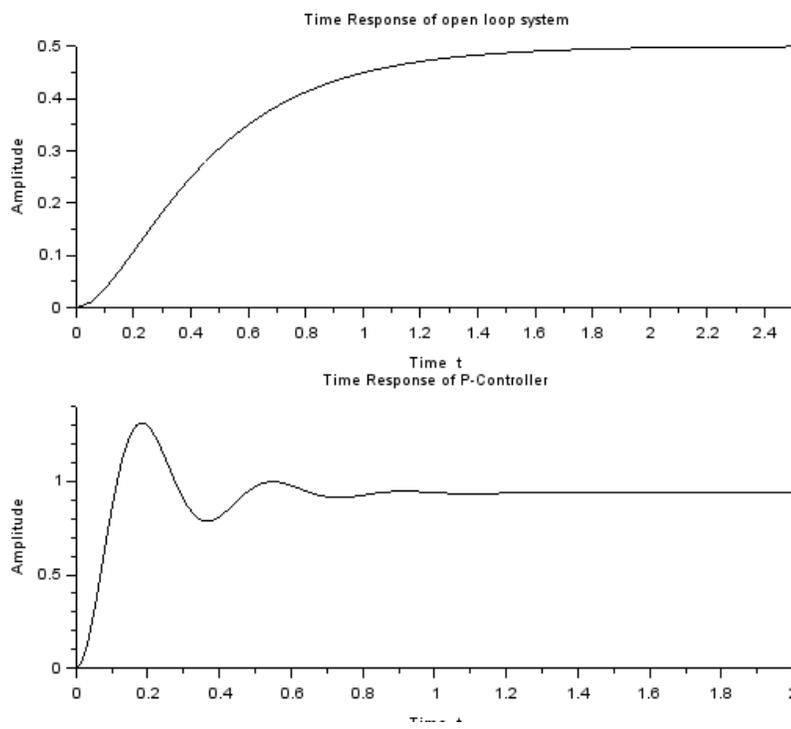


Figure 10.1: P Controller

## Scilab code Solution 10.2 PI Controller

```
1 // Experiment Number : 10.2
2 //Write a program to design PI-controllers and to
  find its time response
3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
  Number :
6 // Course Instructor:K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.1.0
13
14 clc;
15 clear;
16 close;
17
18 //Time Response of Open Loop System
19
20 num=poly([10], 's', 'coeff'); //Numerator input
21 den=poly([20 10 1], 's', 'coeff'); //Denominator
  input
22 q=syslin('c',num/den ) // Finging the transfer
  function of the system
23 t=0:0.05:2.5; //Time interval
24 p=csim('step',t,q); //Time Response of open loop
  system
25 subplot(211)
26 plot2d(t,p);
27 //Labeling the axes and title for the figure
```

```

28 xlabel('Time t');
29 ylabel('Amplitude');
30 title('Time Response of open loop system');
31
32
33
34
35
36 //Design of PI-Controller and finding its time
    response
37
38 kp =30; //Designing parameter of PI-Controller
39 ki =70; //Designing parameter of PI-Controller
40 num=poly([ki kp], 's', 'coeff');
41 den=poly([ki 20+kp 10 1], 's', 'coeff');
42 q=syslin('c', num/den ) //PI-Controller Transfer
    function
43 t =0:0.01:2;
44 p= csim('step', t, q); //Time Response of PI-
    Controller
45 subplot (212)
46 plot2d (t,p);
47 //Labeling the axes and title for the figure
48 xlabel('Time t');
49 ylabel('Amplitude');
50 title('Time Response of PI Controller');

```

---

### Scilab code Solution 10.3 PD Controller

```

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

```

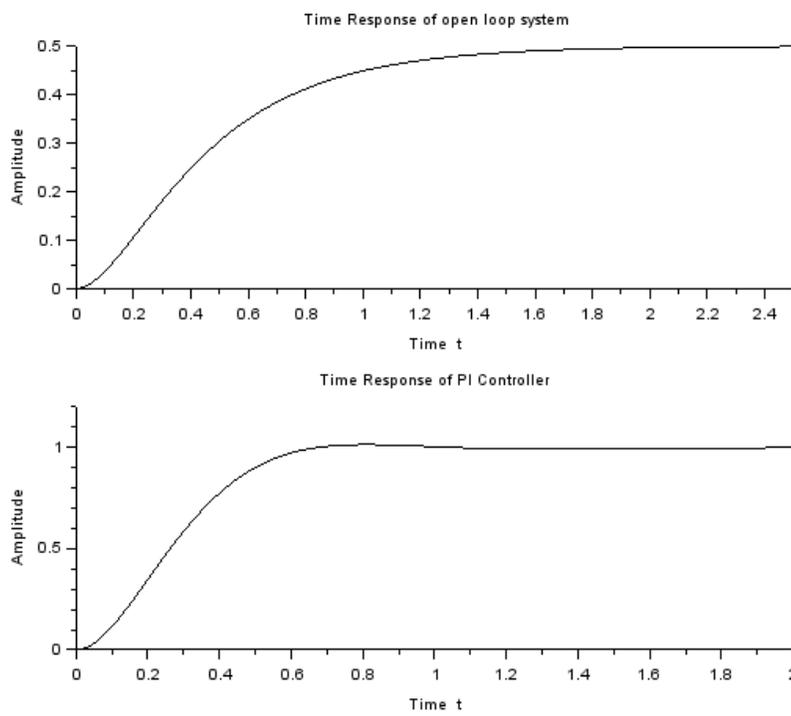


Figure 10.2: PI Controller

```

4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor :K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.1.0
13
14 clc;
15 clear;
16 close;
17
18 //Time Response of Open Loop System
19
20 num=poly ([10] , 's' , 'coeff' ); //Numerator input
21 den=poly ([20 10 1] , 's' , 'coeff' ); //Denominator
   input
22 q=syslin ('c' , num/den ) // Finging the transfer
   function of the system
23 t=0:0.05:2.5; //Time interval
24 p=csim ('step' , t , q); //Time Response of open loop
   system
25 subplot (211)
26 plot2d (t , p);
27 //Labeling the axes and title for the figure
28 xlabel ('Time t');
29 ylabel ('Amplitude');
30 title ('Time Response of open loop system');
31
32
33 //Design of PD-Controller and finding its time
   response
34
35 kp =300; //Designing parameter of PD-Controller
36 kd =10; //Designing parameter of PD-Controller

```

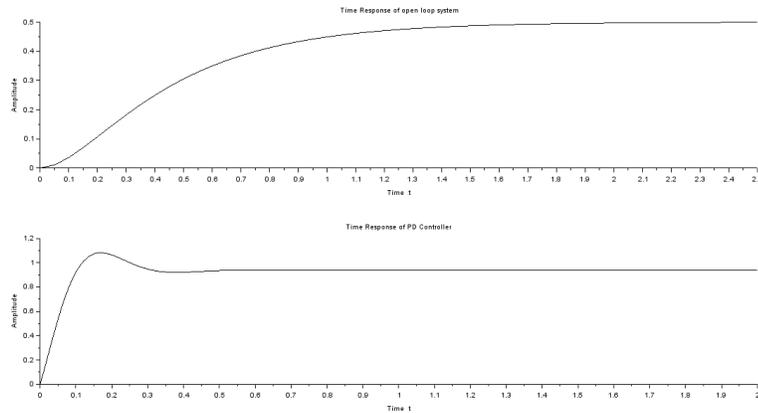


Figure 10.3: PD Controller

```

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

```

---

#### Scilab code Solution 10.4 PID Controller

```

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

```

```

3 //Systems and Signal Processing Laboratory
4 //B.Tech III Year I Sem
5 // Student Name :                               Enrolment
   Number :
6 // Course Instructor:K.Manohar
7 // Matrusri Engineering College ,Hyderabad
8
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.1.0
13
14 clc;
15 clear;
16 close;
17
18 //Time Response of Open Loop System
19
20 num=poly([10], 's', 'coeff'); //Numerator input
21 den=poly([20 10 1], 's', 'coeff'); //Denominator
   input
22 q=syslin('c', num/den ) // Finging the transfer
   function of the system
23 t=0:0.05:2.5; //Time interval
24 p=csim('step', t, q); //Time Response of open loop
   system
25 subplot(211)
26 plot2d(t, p);
27 //Labeling the axes and title for the figure
28 xlabel('Time t');
29 ylabel('Amplitude');
30 title('Time Response of open loop system');
31
32
33 //Design of PID-Controller and finding its time
   response
34
35 kp =350; //Designing parameter of PID-Controller

```

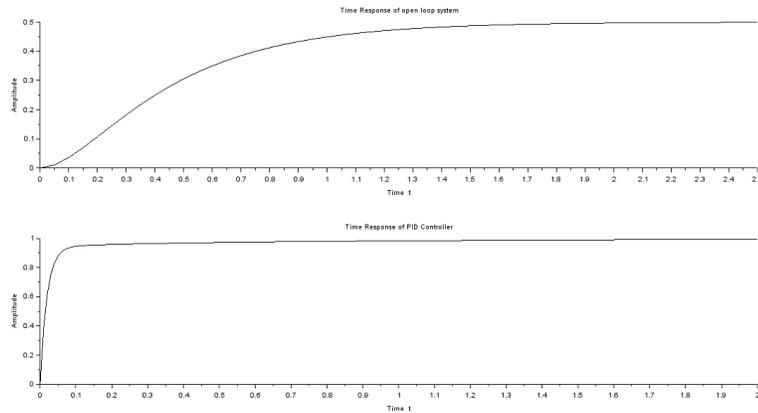


Figure 10.4: PID Controller

```

36 kd =50;      //Designing parameter of PID-Controller
37 ki =300;    //Designing parameter of PID-Controller
38 num=poly([ki kp kd], 's', 'coeff');
39 den=poly([ki 20+kp 10+kd 1], 's', 'coeff');
40 q=syslin('c', num/den) //PID-Controller Transfer
    function
41 t=0:0.01:2;
42 p=csim('step', t, q); //Time Response of PID
    Controller
43 subplot(212)
44 plot2d(t, p);
45 //Labeling the axes and title for the figure
46 xlabel('Time t');
47 ylabel('Amplitude');
48 title('Time Response of PID Controller');

```

---

# Appendix

Scilab code AP 11 //Generating the IDFT function

```
2 function x=idft(X,N)
3     N=length(X);
4     for k=0:1:N-1
5         for n=0:1:N-1
6             p=exp(%i*2*pi*k*n/N);
7             x2(k+1,n+1)=p;
8         end
9     end
10    x=X*x2/N;
11 endfunction
```

IDFT Function

---