Scilab Manual for
Digital Signal Processing Lab
by Prof K.manohar
Electronics Engineering
Sreyas Institute Of Engineering And Technology[1]

Solutions provided by
Prof K.manohar
Electronics Engineering
Sreyas Institute Of Engineering And Technology

May 10, 2026

Contents

List of Experiments

List of Figures

Experiment: 1

DFT / IDFT of given Discrete Time Signal

check Appendix AP 2 for dependency:

```
dft.sci
```

Scilab code Solution 1.1 DFT of a sequence

```
1  // Experiment Number : 1.1
2  //Write a program to find the Discrete Fourier
       Transform (DFT) of a discrete time signal
3  // Digital Signal Processing Laboratory
4  //B.Tech III Year II Sem
5  // Student Name :                          Enrolment
       Number :
6  // Course Instructor:K.Manohar
7  // Sreyas Institute of Engineering and Technology,
       Hyderabad
8
9
10 // OS : Windows 10 . 1
11 // Scilab 6.0.2
12
```

```scilab
13
14  clc;
15  clear;
16  close;
17
18  exec("C:\Users\HAI\Documents\New folder sci\dft.sci"
        );    //path of dft.sce file in my computer
19  x=input ('enter the time domain signal x='); //time
        domain signal
20  N=input ('enter the DFT length N=');
21  X=dft(x,N);    //computind the DFT by calling the dft
          function
22  X1=abs(X);   //magnitude of frequency domain signal X
        (k)
23  disp('Magnitude Response of DFT is');
24  disp(X1);
25  X2=atan(imag(X),real(X));     //phase of frequency
        domain signal X(k)
26  disp('Phase Response of DFT is');
27  disp(X2);
28
29  //ploting the magnitude spectrum
30
31  subplot(2,1,1);
32  k=0:1:N-1
33  plot2d3(k,X1);
34  xlabel('frequency f');
35  ylabel('amplitude');
36  title('magnitude spectrum of X(k)');
37
38  //plotting the phase spectrum
39
40  subplot(2,1,2);
41  plot2d3(k,X2);
42  xlabel('frequency f');
43  ylabel('phase angle');
44  title('phase spectrum of X(k)');
45
```

```
46
47
48  // enter the time domain signal x=[1 2 1 0]
49
50  // enter the DFT length N=4
51
52
53  //  Magnitude  Response  of  DFT  is
54
55  //     4.     2.     0.     2.
56
57  //  Phase  Response  of  DFT  is
58
59  //     0.   −1.5707963     0.    1.5707963
```

check Appendix AP 1 for dependency:

`idft.sci`

Scilab code Solution 1.2 IDFT of a sequence

```
1  //  Experiment  Number  :  1.2
2  //Write  a  program  to  find  the  Inverse  Discrete
       Fourier  Transform  (IDFT)  of  a  discrete  time
       signal
3  //Digital  Signal  Processing  Laboratory
4  //B.Tech  III  Year  II  Sem
5  //  Student  Name  :                           Enrolment
       Number  :
6  //  Course  Instructor:K.Manohar
7  //  Sreyas  Institute  of  Engineering  and  Technology,
       Hyderabad
8
9
```
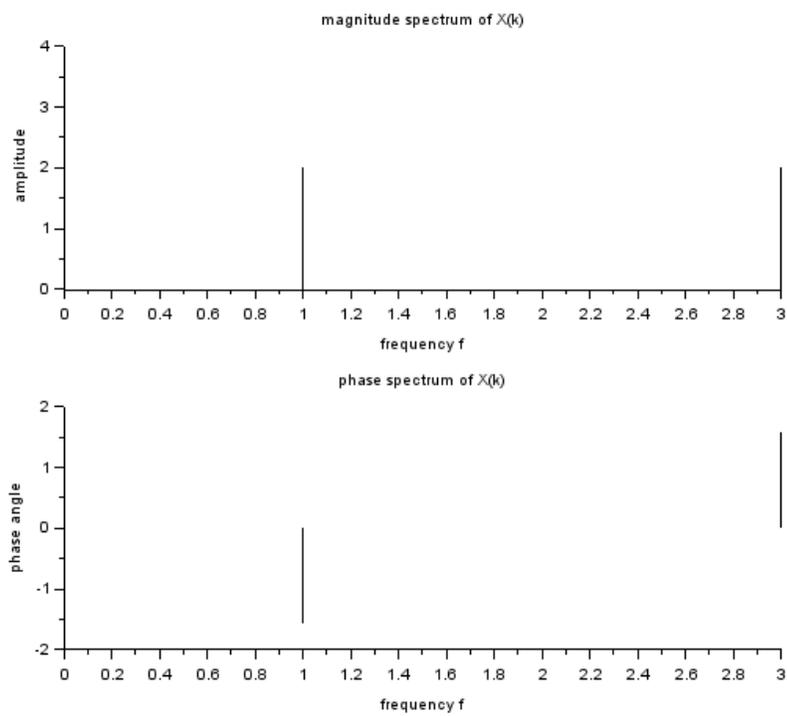
Figure 1.1: DFT of a sequence

```scilab
10  // OS : Windows 10 . 1
11  // Scilab 6.0.2
12
13
14  clc;
15  clear;
16  close;
17
18  exec("C:\ Users\HAI\Documents\New folder sci\idft.sci
        ");    //path of idft.sce file in my computer
19
20  X=input('enter the frequency domain signal  X='); //
        frequency domain signal
21  N=length(X);
22  x=idft(X,N);    //computing the IDFT by calling the
        idft function
23  disp('IDFT of given signal is ');
24  disp(x);
25  n=0:1:N-1
26  plot2d3(n,x);
27  xlabel('discrete time n');
28  ylabel('amplitude');
29  title('IDFT or time domain signal x(n)');
30
31
32
33  //enter the frequency domain signal  X=[4 0 0 0]
34
35
36  // IDFT of given signal is
37
38  //    1.    1.    1.    1.
```
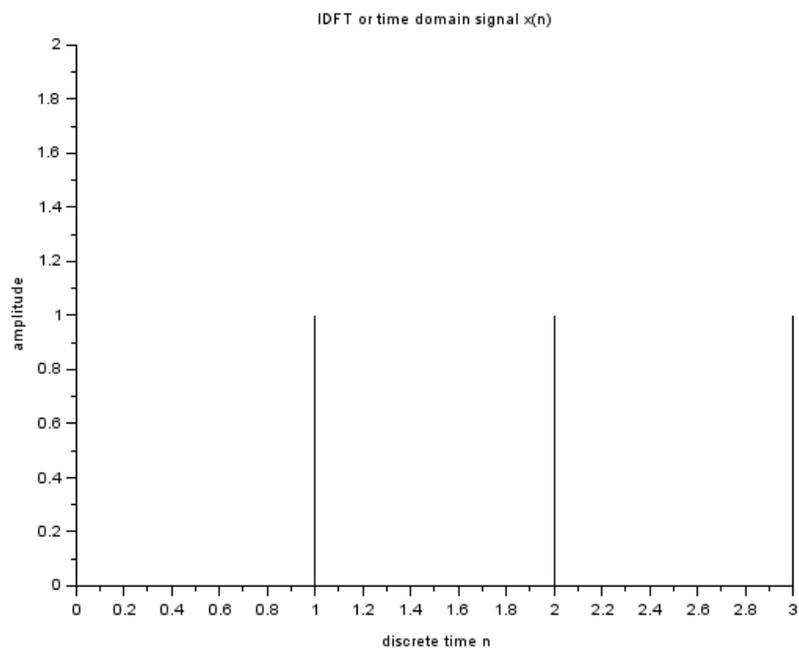
Figure 1.2: IDFT of a sequence

Experiment: 2

Frequency Response of a System

Scilab code Solution 2.0 Frequency Response

```
1  // Experiment Number : 2
2  // Write a program to find the Frequency Response of
      a system with transfer function H(Z)=1/[1-0.9Z
      ^-1]
3  // Digital Signal Processing Laboratory
4  //B.Tech III Year II Sem
5  // Student Name :                          Enrolment
      Number :
6  // Course Instructor:K.Manohar
7  // Sreyas Institute of Engineering and Technology,
      Hyderabad
8
9
10 // OS : Windows 10 . 1
11 // Scilab 6.0.2
12
13 clc;
14 clear;
15 close;
```

```
16
17  n=input ('enter the number of points for frequency
        response  n=');
18
19  //frequency response of system
20  w=0:2*%pi/n:2*%pi;
21  h=zeros(1,length(w));
22  for x=1:length(w)
23      h(x)=1/(1-0.9*exp(-%i*w(x)));
24  end
25  h1=abs(h);  //magnitude of transfer function
26  h2=atan(imag(h),real(h));    //phase of the transfer
        function
27
28  //plotting the magnitude spectrum
29  subplot(2,1,1);
30  plot(w,h1);
31  xlabel('frequency w');
32  ylabel('amplitude');
33  title('magnitude response of sytem H(w)');
34
35  //plotting the phase spectrum
36  subplot(2,1,2);
37  plot(w,h2);
38  xlabel('frequency w');
39  ylabel('phase angle');
40  title('phase response of sytem H(w)');
41
42
43  //enter the number of points for frequency response
        n=50
```
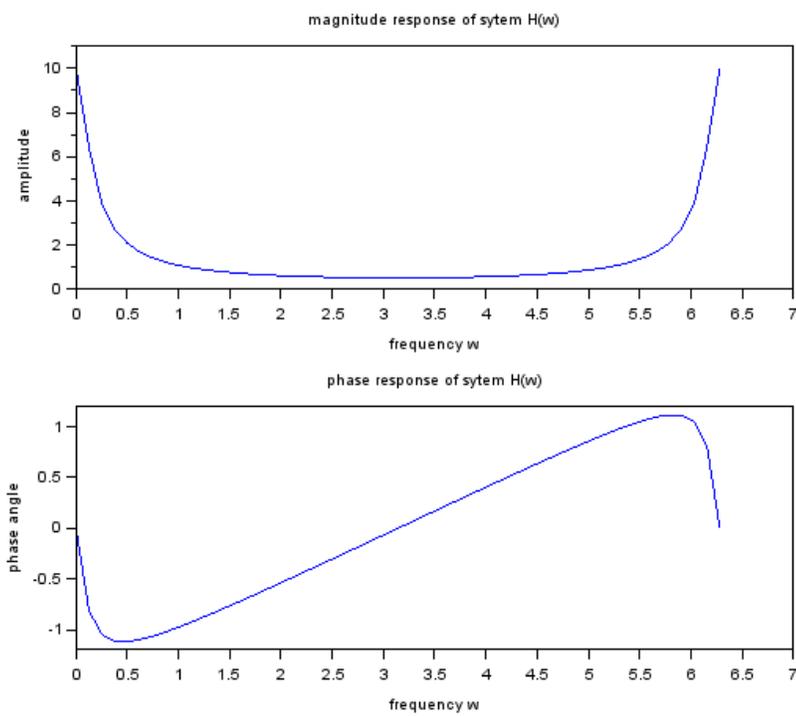
Figure 2.1: Frequency Response

Experiment: 3

Implementation of FFT of a given Sequence

Scilab code Solution 3.0 FFT of a Sequence

```
1  // Experiment Number : 3
2  // Write a program to find the FFT and Inverse FFT of
       a discrete time signal
3  // Digital Signal Processing Laboratory
4  // B. Tech III Year II Sem
5  // Student Name :                        Enrolment
      Number :
6  // Course Instructor:K. Manohar
7  // Sreyas Institute of Engineering and Technology,
      Hyderabad
8  //
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc ;
15 clear ;
16 close ;
```

```scilab
17
18  //input time domain signal
19
20  x=input ('enter the time domain sequence x=');
21  N=length(x);
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 magnitude spectrum
35
36  k=0:1:N-1
37  subplot(2,2,1);
38  plot2d3(k,y1);
39  xlabel('discrete frequency  k');
40  ylabel('amplitude');
41  title('magnitude spectrum of FFT signal');
42
43  //plotting the phase spectrum
44
45  subplot(2,2,2);
46  plot2d3(k,y2);
47  xlabel('discrete frequency  k');
48  ylabel('phase angle');
49  title('phase spectrum of FFT signal');
50
51  // Finding Inverse Fast Fourier Transform
52  z=ifft(y);
53  disp('Inverse Fast Fourier Transform is');
```

```
54  disp(z);
55
56  //Plotting Inverse FFT signal
57
58  n=0:1:N-1
59  subplot(2,2,3);
60  plot2d3(n,z);
61  xlabel('discrete time n');
62  ylabel('amplitude');
63  title('Inverse FFT or time domain signal');
64
65
66  //enter the time domain sequence x=[1 2 3 4]
67
68
69  // Frequency domain signal is
70
71  //    10.   -2. + 2.i   -2.   -2. - 2.i
72
73  // Magnitude Response is
74
75  //    10.    2.8284271    2.    2.8284271
76
77  // Phase Response is
78
79
80  //          column 1 to 3
81
82  //    0.    2.3561945    3.1415927
83
84  //          column 4
85
86  //   -2.3561945
87
88  // Inverse Fast Fourier Transform is
89
90  //    1.    2.    3.    4.
```
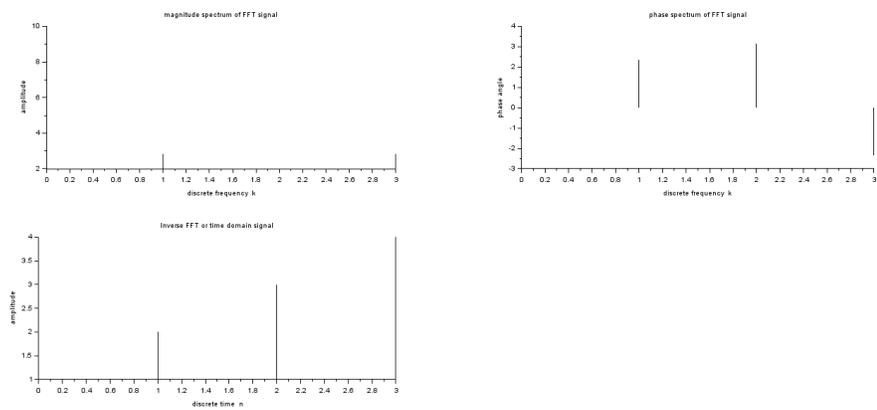
Figure 3.1: FFT of a Sequence

Experiment: 4

Determination of Power Spectrum of a given Signal

Scilab code Solution 4.0 Power Spectral Density of a sequence

```scilab
1  // Experiment Number : 4
2  //Write a program to find the power spectral density
        of a signal
3  //Digital Signal Processing Laboratory
4  //B.Tech III Year II Sem
5  // Student Name :                          Enrolment
     Number :
6  // Course Instructor:K.Manohar
7  // Sreyas Institute of Engineering and Technology,
     Hyderabad
8  //
9
10
11 // OS : Windows 10 . 1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
```

```scilab
17
18  //generating and plotting the input signal
19
20  x=input('enter the time domain sequence  x=');
21  disp(x);
22  N=length(x);
23  n=0:1:N-1
24  subplot(3,1,1);
25  plot2d3(n,x);
26  xlabel('discrete time  n');
27  ylabel('amplitude');
28  title('time domain input signal  x(n)');
29
30  //generating and plotting autocorrelation signal
31  R=xcorr(x,x);
32  disp(R);
33  N1=length(R);
34  n1=0:1:N1-1
35  subplot(3,1,2);
36  plot2d3(n1,R);
37  xlabel('discrete time  n1');
38  ylabel('amplitude');
39  title('Autocorrelation function  R(n1)');
40
41  //generating and plotting the power spectral density
        signal
42  P=fft(R);
43  disp(P);
44  N2=length(P);
45  k=0:1:N2-1
46  subplot(3,1,3);
47  plot2d3(k,P);
48  xlabel('discrete frequency  k');
49  ylabel('amplitude');
50  title('power spectral density  P(k)');
51
52
53  //enter the time domain sequence  x=[1 2 3 4]
```

```
54
55
56  //    1.    2.    3.    4.
57
58
59  //          column  1  to  6
60
61  //    4.    11.    20.    30.    20.    11.
62
63  //          column  7
64
65  //    4.
66
67
68  //          column  1  to  2
69
70  //    100.    -38.594245 - 18.586009 i
71
72  //          column  3
73
74  //    3.9066412 + 4.8987731 i
75
76  //          column  4
77
78  //  -1.3123959 - 5.749982 i
79
80  //          column  5
81
82  //  -1.3123959 + 5.749982 i
83
84  //          column  6
85
86  //    3.9066412 - 4.8987731 i
87
88  //          column  7
89
90  //  -38.594245 + 18.586009 i
```
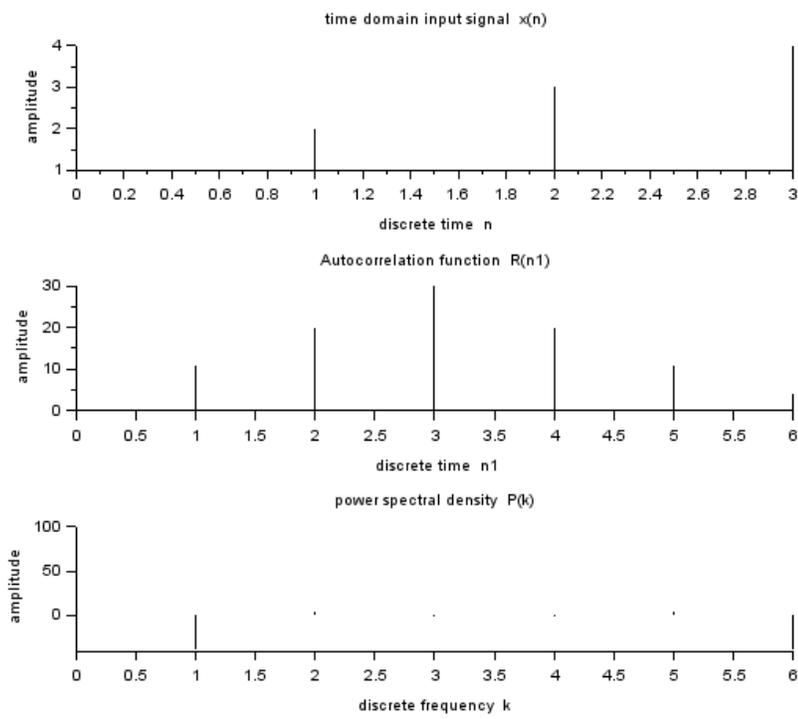
Figure 4.1: Power Spectral Density of a sequence

Experiment: 5

Implementation of Lowpass FIR Filter for given specifications

Scilab code Solution 5.0 Lowpass FIR Filter

```
1 // Experiment Number : 5
2 //Write a program to generate Lowpass FIR Filter for
      given specifications
3 //Digital Signal Processing Laboratory
4 //B.Tech III Year II Sem
5 // Student Name :                    Enrolment
    Number :
6 // Course Instructor:K.Manohar
7 // Sreyas Institute of Engineering and Technology,
    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 filter  N=');
21
22  // finding the magnitude response of Lowpass FIR
        Filter
23
24  w1 =(2*%pi)*(fc/fs);
25  disp(w1,'digital cutoff frequency in radians');
26  wc1 =w1/%pi;
27  disp(wc1,'normalized digital cutoff frequency in
        radians');
28  [wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0],'re',[0,0]);
29  disp(wft,'impulse response of Lowpass FIR filter:h(n
        )=');
30  a=gca();
31  plot(2*fr,wfm);
32  xlabel('normalized digital frequency w');
33  ylabel('magnitude ');
34  title('magnitude response of Lowpass FIR Filter');
35
36
37  //enter the cutoff frequency  fc=1200
38
39  //enter the sampling frequency  fs=10000
40
41  //enter the order of filter  N=3
42
43
44  // digital cutoff frequency in radians
45
46  //    0.7539822
47
48  // normalized digital cutoff frequency in
49  //  radians
```
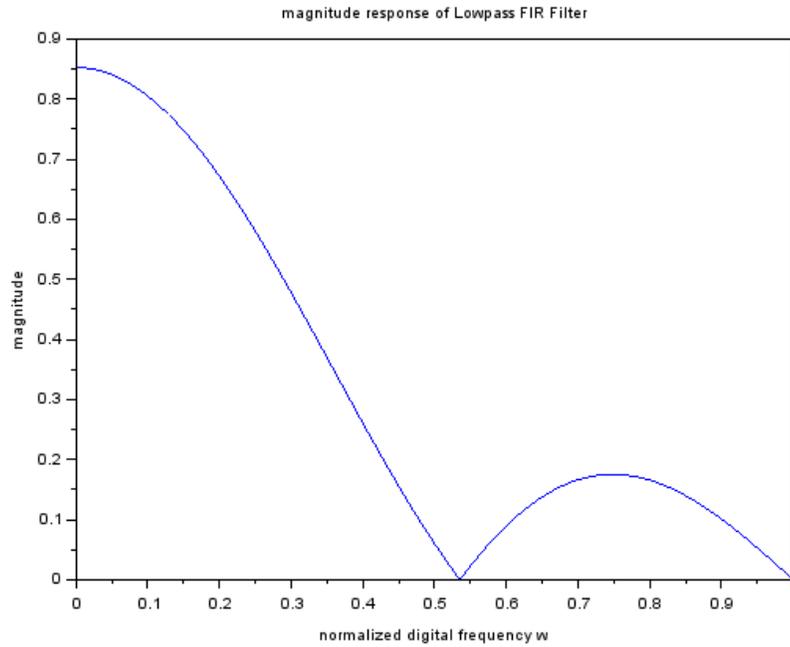
Figure 5.1: Lowpass FIR Filter

```
50  //    0.24
51
52  // impulse  response  of  Lowpass  FIR  filter
53  // :h(n)=
54
55  //        column  1  to  3
56
57  //    0.1920103    0.2343554    0.2343554
58
59  //        column  4
60
61  //    0.1920103
```

Experiment: 6

Implementation of IIR Filter for given specifications

Scilab code Solution 6.1 Lowpass IIR Filter

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

```scilab
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 lowpass  Type-1 Chebyshev  IIR filter
35
36  [Hz3]=iir(N,'lp','cheb1',[fp/2,0],[0.2,0]);
37  [Hw3,w3]=frmag(Hz3,256);
38  subplot(2,2,2);
39  plot2d3(w3,abs(Hw3));
40  xlabel('frequency');
41  ylabel('magnitude');
42  title('type-I chebyshev Lowpass  IIR filter');
43
44  //generating lowpass  Type-II Chebyshev  IIR filter
45
46  [Hz4]=iir(N,'lp','cheb2',[fp/2,0],[0,0.1]);
47  [Hw4,w4]=frmag(Hz4,256);
48  subplot(2,2,3);
49  plot2d3(w4,abs(Hw4));
50  xlabel('frequency');
51  ylabel('magnitude');
52  title('type-II chebyshev Lowpass  IIR filter');
53
54  //enter the cutoff frequency   fc=1000
55
```
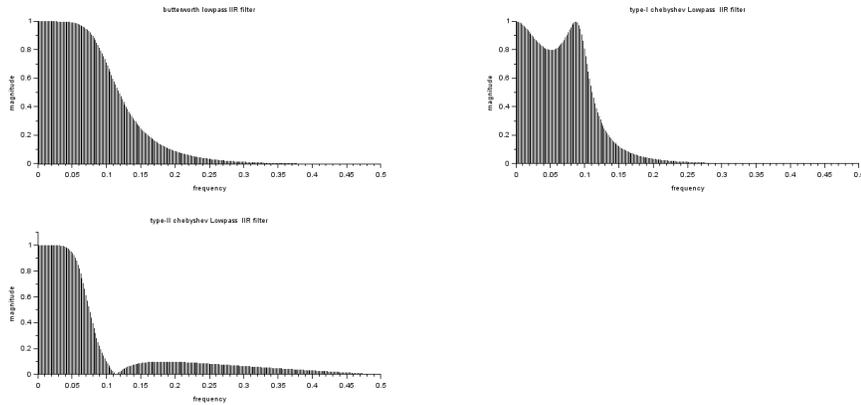
Figure 6.1: Lowpass IIR Filter

```
56  // enter  the  sampling  frequency    fs =10000
57
58  // enter  the  order  of  the  filter    N=3
```

Scilab code Solution 6.2 Highpass IIR Filter

```
1  //  Experiment  Number  :  6.2
2  // Write  a  program  to  generate  Highpass  IIR  Filter
3  // Digital  Signal  Processing  Laboratory
4  //B. Tech  III  Year  II  Sem
5  //  Student  Name  :                              Enrolment
       Number  :
6  //  Course  Instructor :K. Manohar
7  //  Sreyas  Institute  of  Engineering  and  Technology ,
       Hyderabad
8  //
9
10
11  //  OS  :  Windows  10  . 1
```

```scilab
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 Highpass Butterworth IIR filter
24
25  [Hz2]=iir(N,'hp','butt',[fp/2,0],[0,0]);
26  [Hw2,w2]=frmag(Hz2,256);
27  subplot(2,2,1);
28  plot2d3(w2,abs(Hw2));
29  xlabel('frequency');
30  ylabel('magnitude');
31  title('butterworth highpass IIR filter');
32
33  //generating Highpass  Type-1 Chebyshev  IIR  filter
34
35  [Hz3]=iir(N,'hp','cheb1',[fp/2,0],[0.2,0]);
36  [Hw3,w3]=frmag(Hz3,256);
37  subplot(2,2,2);
38  plot2d3(w3,abs(Hw3));
39  xlabel('frequency');
40  ylabel('magnitude');
41  title('type-I chebyshev Highpass filter');
42
43  //generating Highpass  Type-II Chebyshev  IIR  filter
44
45  [Hz4]=iir(N,'hp','cheb2',[fp/2,0],[0,0.1]);
46  [Hw4,w4]=frmag(Hz4,256);
47  subplot(2,2,3);
48  plot2d3(w4,abs(Hw4));
49  xlabel('frequency');
```
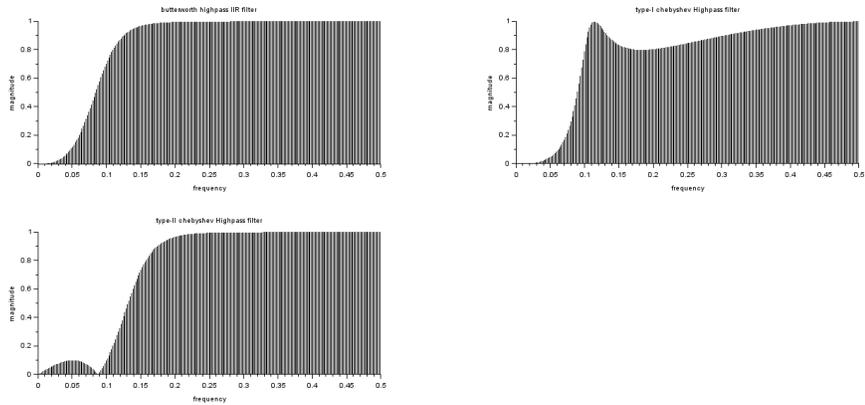
Figure 6.2: Highpass IIR Filter

```
50  ylabel('magnitude');
51  title('type-II chebyshev Highpass filter');
52
53  //enter the cutoff frequency   fc=1000
54
55  //enter the sampling frequency   fs=10000
56
57  //enter the order of the filter   N=3
```

Experiment: 7

Generation of DTMF Signals

Scilab code Solution 7.0 DTMF Signals Generation

```scilab
1  //  Experiment  Number  :  7
2  // Write  a  program  to  generate  the  Dual  Tone  Multi
       Frequency (DTMF)  Signal
3  // Digital  Signal  Processing  Laboratory
4  //B. Tech  III  Year  II  Sem
5  //  Student  Name  :                              Enrolment
       Number  :
6  //  Course  Instructor :K. Manohar
7  //  Sreyas  Institute  of  Engineering  and  Technology ,
       Hyderabad
8  //
9
10
11  //  OS  :  Windows  10  .  1
12  //  Scilab  6.0.2
13
14  clc ;
15  clear ;
16  close ;
17
18  fs = input ( ' enter  the  sampling  frequency   fs = ' ) ;
```

```scilab
19  Ts =1/ fs ;
20  num_samples = input ( 'enter the number of samples=') ;
21  dial_number = input ( 'enter the dial number=') ;
22  T= Ts *(0: num_samples -1) ;
23
24  // Generating DTMF Signals
25
26  select dial_number    // selecting dial number on
        keypad
27  case 0 then
28      f1 =941;
29      f2 =1336;
30  case 1 then
31      f1 =697;
32      f2 =1209;
33   case 2 then
34      f1 =697;
35      f2 =1336;
36   case 3 then
37      f1 =697;
38      f2 =1477;
39   case 4 then
40      f1 =770;
41      f2 =1209;
42   case 5 then
43      f1 =770;
44      f2 =1336;
45   case 6 then
46      f1 =770;
47      f2 =1477;
48   case 7 then
49      f1 =852;
50      f2 =1209;
51   case 8 then
52      f1 =852;
53      f2 =1336;
54   case 9 then
55      f1 =852;
```

```
56          f2=1477;
57    case 'A' then
58          f1=697;
59          f2=1633;
60    case 'B' then
61          f1=770;
62          f2=1633;
63    case 'C' then
64          f1=852;
65          f2=1633;
66    case '*' then
67          f1=941;
68          f2=1209;
69    case '#' then
70          f1=941;
71          f2=1477;
72    case 'D' then
73          f1=941;
74          f2=1633;
75    end
76    first_sine=cos(2**%pi*f1*T);
77    second_sine=cos(2**%pi*f2*T);
78    dtmf_signal=first_sine+second_sine;
79    plot(dtmf_signal);
80    xlabel(' time t');
81    ylabel('amplitude');
82    title('Dual Tone Multi Frequency(DTMF) signal');
83
84
85    //enter the sampling frequency    fs=8000
86
87    //enter the number of samples=100
88
89    //enter the dial number=9
```
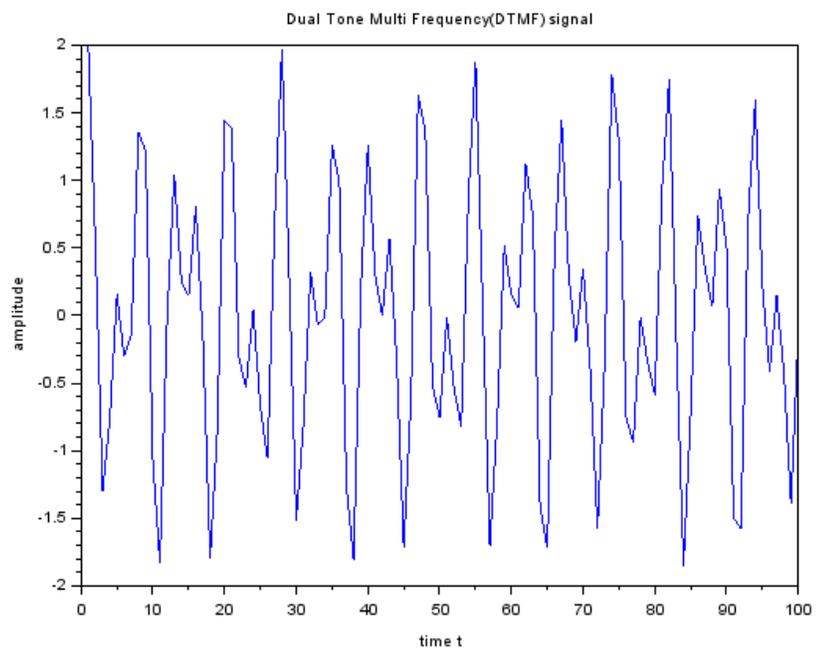
Figure 7.1: DTMF Signals Generation

Experiment: 8

Implementation of Decimation Process

Scilab code Solution 8.0 Decimation of a Signal

```
1  // Experiment Number : 8
2  //Write a program to implement the decimation
       process
3  //Digital Signal Processing Laboratory
4  //B.Tech III Year II Sem
5  // Student Name :                    Enrolment
       Number :
6  // Course Instructor:K.Manohar
7  // Sreyas Institute of Engineering and Technology,
       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
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
```
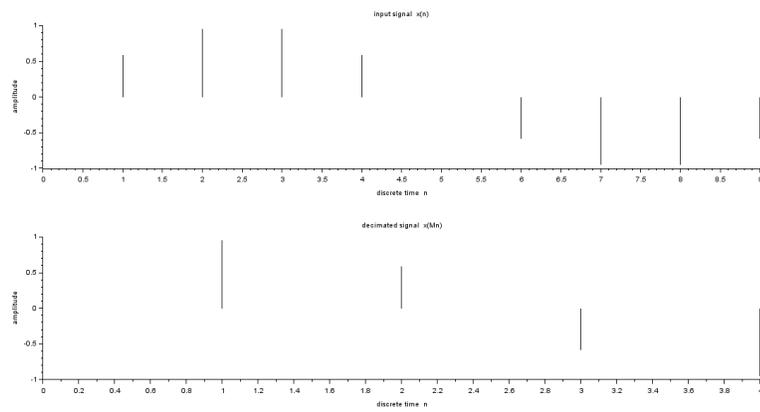
Figure 8.1: Decimation of a Signal

Experiment: 9

Implementation of Interpolation Process

Scilab code Solution 9.0 Interpolation of a Signal

```scilab
1  // Experiment Number : 9
2  //Write a program to implement the Interpolation
       process
3  //Digital Signal Processing Laboratory
4  //B.Tech III Year II Sem
5  // Student Name :                         Enrolment
       Number :
6  // Course Instructor:K.Manohar
7  // Sreyas Institute of Engineering and Technology,
       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 interpolation signal
36
37  L=input ('enter the interpolation factor L=');
38  n1=1:1:L*N;
39  x1=[zeros(1,L*N)];
40  j=1:L:L*N;
41  x1(j)=x;
42  disp(x1);
43
44  //plotting the interpolated signal
45
46  subplot(2,1,2);
47  plot2d3(n1,x1);
48  xlabel('discrete time n');
49  ylabel('amplitude');
50  title('upsampled signal x(n/L)');
51
```

```
52  //enter  the  number  of  points  in  input  signal   N=10
53
54  //enter  the  amplitude  of  input  sinusoidal  signal   A
       =1
55
56  //enter  the  frequency  of  input  sinusoidal  signal   fo
       =0.1
57
58
59
60  //           column  1  to  3
61
62  //    0.     0.5877853     0.9510565
63
64  //           column  4  to  6
65
66  //    0.9510565     0.5877853     1.225D−16
67
68  //           column  7  to  9
69
70  //   −0.5877853    −0.9510565    −0.9510565
71
72  //           column  10
73
74  //   −0.5877853
75  //enter  the  interpolation  factor   L=2
76
77
78
79  //           column  1  to  4
80
81  //    0.     0.     0.5877853     0.
82
83  //           column  5  to  8
84
85  //    0.9510565     0.     0.9510565     0.
86
87  //           column  9  to  12
```

41
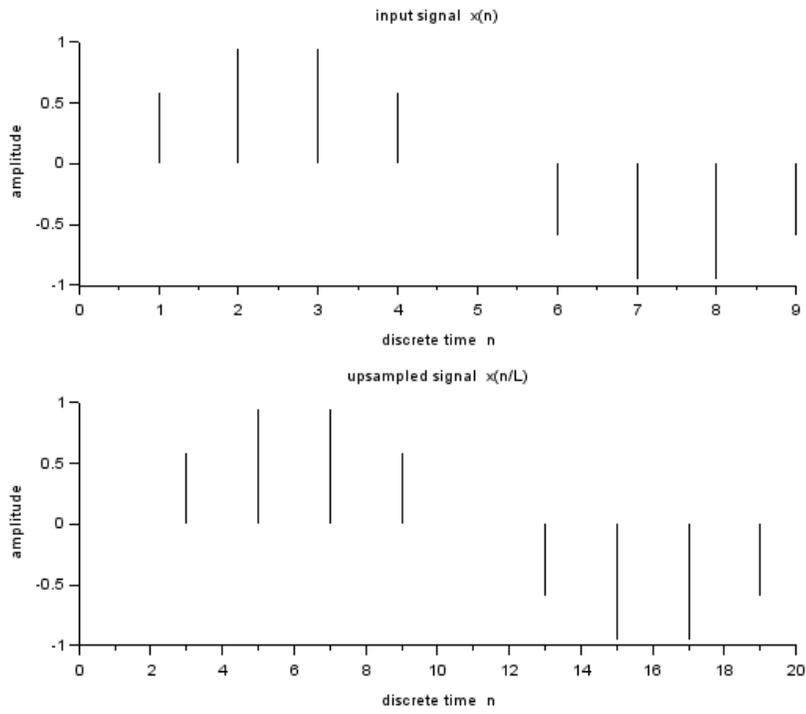
Figure 9.1: Interpolation of a Signal

```
88
89  //    0.5877853    0.    1.225D-16    0.
90
91  //         column  13  to  16
92
93  //   -0.5877853    0.   -0.9510565    0.
94
95  //         column  17  to  20
96
97  //   -0.9510565    0.   -0.5877853    0.
```

Experiment: 10

Implementation of Sampling rate conversion by a factor I/D

Scilab code Solution 10.0 Sampling Rate converter

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

```scilab
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  disp(x);
29
30  //plotting  the  input  signal
31
32  subplot(2,1,1);
33  plot2d3(n,x(1:30));
34  xlabel('discrete  time   n');
35  ylabel('amplitude');
36  title('input  signal   x(n)');
37
38  // sampling  rate  converted  by  a  factor  L/M signal
39  y=intdec(x,L/M);
40  disp(y);
41
42  //plotting  the  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('sampling  rate  converted  signal');
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
60
61
62
63  //            column  1  to  3
64
65  //    0.    0.4257793   −0.7705132
66
67  //            column  4  to  6
68
69  //    0.9685832   −0.9822873    0.809017
70
71  //            column  7  to  9
72
73  //   −0.4817537    0.0627905    0.3681246
74
75  //            column  10  to  12
76
77  //   −0.7289686    0.9510565   −0.9921147
78
79  //            column  13  to  15
80
81  //    0.8443279   −0.5358268    0.1253332
82
83  //            column  16  to  18
84
85  //    0.309017   −0.6845471    0.9297765
86
87  //            column  19  to  21
```

```
 88
 89  //   -0.9980267     0.8763067    -0.5877853
 90
 91  //          column 22 to 24
 92
 93  //    0.1873813     0.2486899    -0.637424
 94
 95  //          column 25 to 27
 96
 97  //    0.9048271    -1.     0.9048271
 98
 99  //          column 28 to 30
100
101  //   -0.637424     0.2486899     0.1873813
102
103
104  //          column 1 to 3
105
106  //   -0.0044437     0.8162176     0.430223
107
108  //          column 4 to 6
109
110  //   -0.6991301    -0.774957     0.3930428
111
112  //          column 7 to 9
113
114  //    0.9730269     0.0130957    -0.986731
115
116  //          column 10 to 12
117
118  //   -0.431533     0.8134607     0.7777726
119
120  //          column 13 to 15
121
122  //   -0.4861974    -0.983244     0.0672342
123
124  //          column 16 to 18
125
```

```
126  //    1.0072059     0.3636808    −0.8440229
127
128  //          column  19  to  21
129
130  //   −0.7245249     0.5240108     0.9466128
131
132  //          column  22  to  24
133
134  //   −0.1075454    −0.987671    −0.3264978
135
136  //          column  25  to  27
137
138  //    0.8398842     0.6957976    −0.5313831
139
140  //          column  28  to  30
141
142  //   −0.930282     0.1208895     0.9854806
143
144  //          column  31  to  33
145
146  //    0.3134607    −0.8509991    −0.6889908
147
148  //          column  34  to  36
149
150  //    0.5525013     0.9342202    −0.1468245
151
152  //          column  37  to  39
153
154  //   −1.0024705    −0.2888443     0.8807504
155
156  //          column  40  to  42
157
158  //    0.67167    −0.592229    −0.9289559
159
160  //          column  43  to  45
161
162  //    0.191825     1.0120117     0.2442462
163
```

```
164  //            column  46  to  48
165
166  //   -0.9054772    -0.6329803     0.6303453
167
168  //            column  49  to  51
169
170  //    0.9003833    -0.2401604    -0.9955563
171
172  //            column  52  to  54
173
174  //   -0.1887578     0.9003833     0.5708187
175
176  //            column  55  to  57
177
178  //   -0.6329803    -0.8243498     0.2442462
179
180  //            column  58  to  60
181
182  //    0.8735432     0.191825     -0.5116523
```
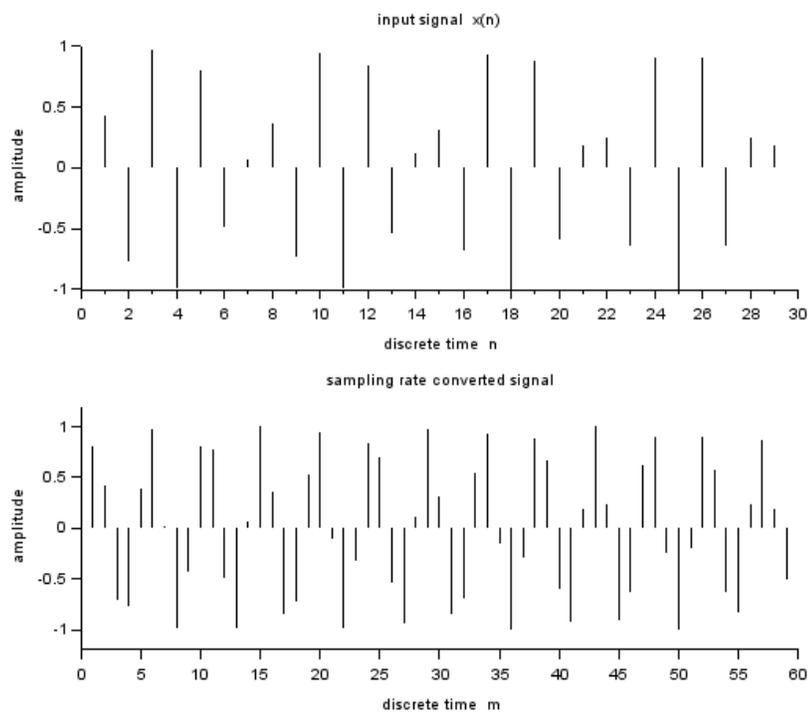
Figure 10.1: Sampling Rate converter

Experiment: 11

Impulse response of First order and Second order systems

Scilab code Solution 11.0 Impulse Response of a system

```
1  // Experiment Number : 11
2  //Write a program to find the impulse response of
       first order and second order systems
3  //Digital Signal Processing Laboratory
4  //B.Tech III Year II Sem
5  // Student Name :                        Enrolment
       Number :
6  // Course Instructor:K.Manohar
7  // Sreyas Institute of Engineering and Technology,
       Hyderabad
8
9
10 // OS : Windows 10 . 1
11 // Scilab 6.0.2
12 //
13
14 clc;
15 clear;
16 close;
```

```
17
18  // Impulse Response of first order system described
        by difference equation    y(n)+2y(n-1)=x(n)
19
20  a1=input ('enter the coefficients of input vector of
         first order system   a1=');
21  b1=input ('enter the coefficients of output vector
        of first order system   b1=');
22  n1=input ('enter the lower of impulse response   n1='
        );
23  n2=input ('enter the upper of impulse response   n2='
        );
24  n=n1:n2
25
26  //generate the impulse input function
27  x=zeros(1,length(n));
28  for i=1:length(n)
29      if n(i)==0
30          x(i)=1;
31  end
32  end
33
34  h1=filter(a1,b1,x);       //finds the impulse response
         of first order system
35  disp(h1);              //display the values of impulse
        response in console window
36  subplot(2,1,1)
37  plot2d3(n,h1);         //to plot the impulse response
        of sytem in graphical window
38  xlabel('discrete time   n');
39  ylabel('amplitude');
40  title('impulse response of first order system   h1(n)
        ');
41
42  // Impulse Response of Second order system described
         by difference equation
43  //  y(n)+0.5y(n-1)+0.3y(n-2)=x(n)+5x(n-1)
44
```

```scilab
45  a2=input ('enter the coefficients of input vector of
        second order system   a2=');
46  b2=input ('enter the coefficients of output vector
        of second order system   b2=');
47  h2=filter(a2,b2,x);        //finds the impulse response
        of system
48  disp(h2);             //display the values of impulse
        response in console window
49  subplot(2,1,2)
50  plot2d3(n,h2);         //to plot the impulse response
        of sytem in graphical window
51  xlabel('discrete time   n');
52  ylabel('amplitude');
53  title('impulse response of second order system   h2(n
        )');
54
55  //enter the coefficients of input vector of first
        order system   a1=[1]
56
57  //enter the coefficients of output vector of first
        order system   b1=[1  2]
58
59  //enter the lower of impulse response   n1=0
60
61  //enter the upper of impulse response   n2=5
62
63
64  //    1.   −2.    4.   −8.    16.   −32.
65
66  //enter the coefficients of input vector of second
        order system   a2=[1  5]
67
68  //enter the coefficients of output vector of second
        order system   b2=[1  0.5  0.3]
69
70
71
72  //   column 1 to 5
```
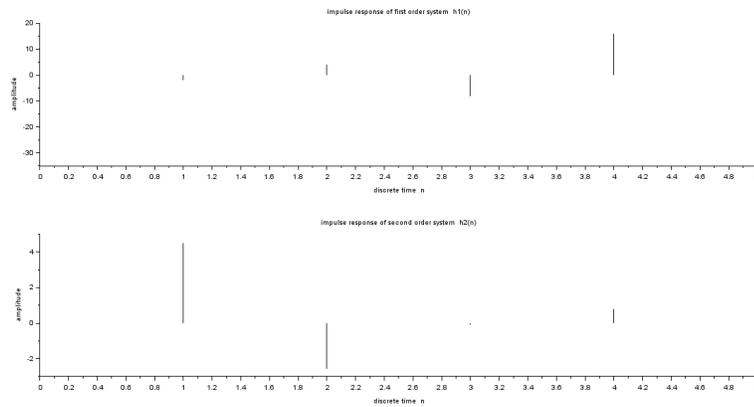
Figure 11.1: Impulse Response of a system

```
73
74  //      1.      4.5     −2.55    −0.075     0.8025
75
76  //           column  6
77
78  //   −0.37875
```

Experiment: 12

Finding the Fourier Series Coefficients of a Periodic Discrete Time Signal

Scilab code Solution 12.0 Fourier Series Coefficients

```
1  // Experiment Number : 12
2  //Write a program to generate Fourier Series
      Coefficients of a Periodic Signal
3  //Digital Signal Processing Laboratory
4  //B.Tech III Year II Sem
5  // Student Name :                    Enrolment
      Number :
6  // Course Instructor :K.Manohar
7  // Sreyas Institute of Engineering and Technology,
      Hyderabad
8
9
10 // OS : Windows 10 . 1
11 // Scilab 6.0.2
12
13 clc;
14 clear;
```

```
15  close;
16
17  //generating input signal
18  n = 0:0.01:5;
19  N = input('enter the time period N=');
20  Wo = 2*%pi/N;   //fundamental frequency
21  A = input('enter the amplitude of sinusoidal signal
       A=');
22  x = A*sin(Wo*n);   //input signal x(n)
23
24  //finding fourier series coefficients
25  for k =0:N-2
26         D(k+1,:) = exp(-%i*Wo*n.*k);
27         C(k+1) = x*D(k+1,:)'/length(n);
28         end
29  C =C'
30  C_conju = conj(C);
31  Ck = [C_conju($:-1:1),C(2:$)]
32  k = -(N-2):(N-2);
33  //
34  figure
35  C = gca();
36  C.y_location = "origin";
37  C.x_location = "origin";
38  C.data_bounds=[-8,-1;8,1];
39  pol1 = C;
40  pol1.thickness = 3;
41  plot2d3('gnn',k,-imag(Ck),5)
42  xlabel("discrete time");
43  ylabel("Amplitude");
44  pol1 = C;
45  pol1.thickness = 3;
46  plot2d3('gnn',N+k,-imag(Ck),5)
47  pol1 = C.children(1).children(1);
48  pol1.thickness = 3;
49  plot2d3('gnn',-(N+k),-imag(Ck($:-1:1)),5)
50  pol1 = C;
51  pol1.thickness = 3;
```
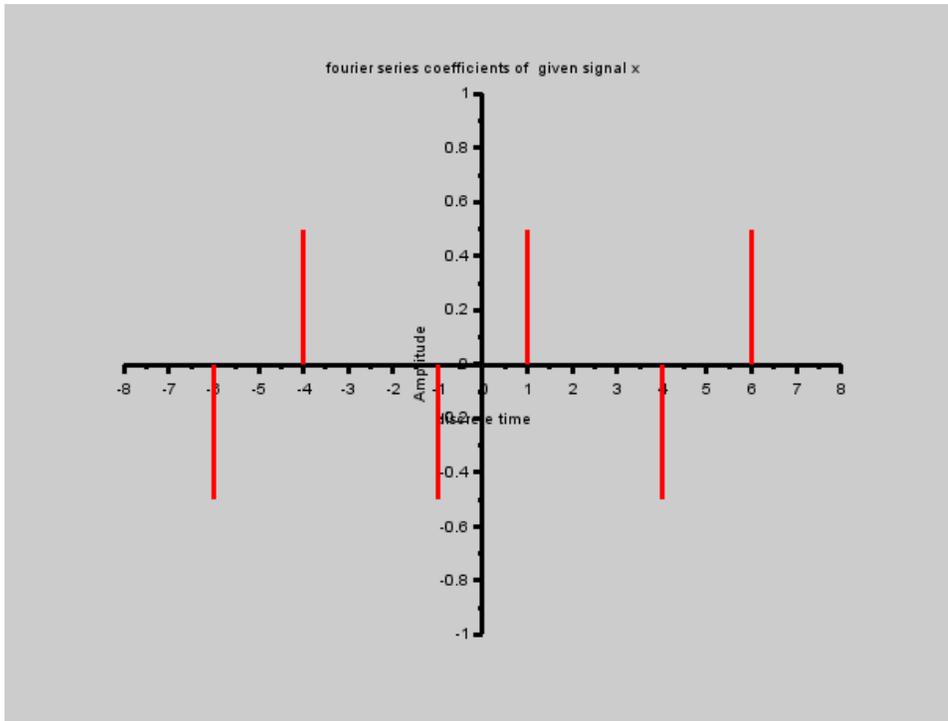
Figure 12.1: Fourier Series Coefficients

```
52  title('fourier series coefficients of  given signal
       x')
53
54  //enter the time period N=5
55
56  //enter the amplitude of sinusoidal signal A=1
```

Experiment: 13

Generation of Sinusoidal signal based on recursive difference equation

Scilab code Solution 13.0 recursive sinusoid generation

```
1  // Experiment Number : 13
2  // Write a program to generate a sinusoidal signal
      based on a recursive difference equation p(k+2)=a
      *p(k+1)+b*p(k)
3  // Digital Signal Processing Laboratory
4  // B. Tech III Year II Sem
5  // Student Name :                        Enrolment
      Number :
6  // Course Instructor :K. Manohar
7  // Sreyas Institute of Engineering and Technology,
      Hyderabad
8
9
10 // OS : Windows 10 . 1
11 // Scilab 6.0.2
12
13 clc;
```

```
14 clear;
15 close;
16
17 //generating the input signal
18 n=0:3;
19 A=input ('enter the amplitude of input sinusoidal
      signal  A=');
20 N=input ('enter the fundamental time period of input
       sinusoidal signal  N=');
21 y=A*sin(2*%pi*n/N);
22
23 //coefficients of difference equation
24 a=y(3)/y(2);
25 b=(y(4)-a*y(3))/y(2);
26 disp('The coefficients of the   difference   equation
      are');
27 disp(a);
28 disp(b);
29
30 //generation of sinusoidal signal through recursive
     equation p(k+2)=a*p(k+1)+b*p(k)
31
32 for k=1:1:119
33     p(1)=y(1);   //initial values
34     p(2)=y(2);   //initial values
35     p(k+2)=a*p(k+1)+b*p(k);
36 end
37
38 plot2d3(p);
39 xlabel('discrete time');
40 ylabel('amplitude');
41 title('discrete time sinusoidal signal');
42
43
44 //enter the amplitude of input sinusoidal signal  A
     =10
45
46 //enter the fundamental time period of input
```
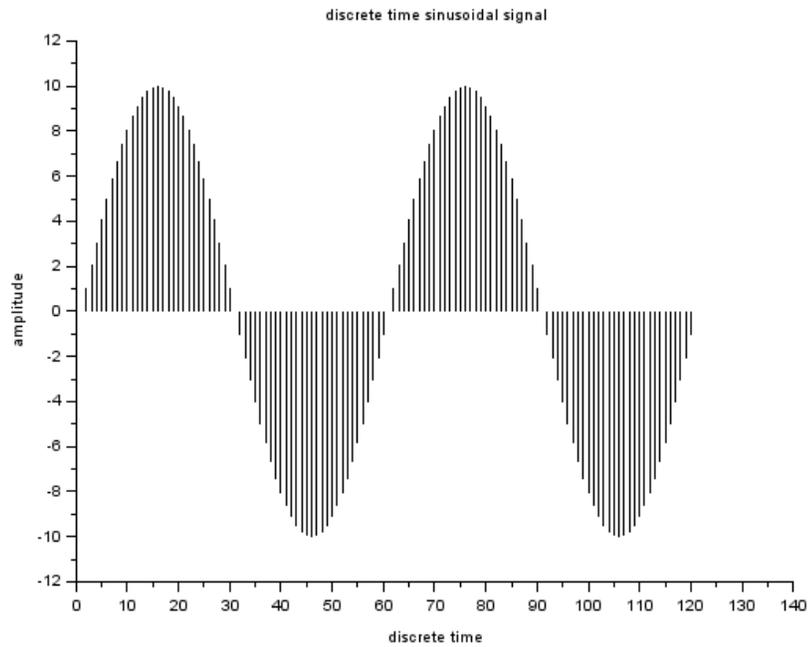
Figure 13.1: recursive sinusoid generation

```
     sinusoidal  signal   N=60
47
48
49  // The  coefficients  of  the    difference
50  // equation  are
51  //    1.9890438
52
53  //   −1.
```

Appendix

Scilab code AP 11 // generating the IDFT function

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

Scilab code AP 12 // Generating the DFT function

```
2
3  function X=dft(x,N)
4      L=length(x);
5      if(L>N)
6          disp('error because  L should be less than N
                ');
7      end
8      x1=[x zeros(1,N-L)];    // zero padding the
            sequence x(n)
9      for k=0:1:N-1
10         for n=0:1:N-1
```

60

```
11                    p=exp(-%i*2*%pi*k*n/N);
12                    x2(k+1,n+1)=p;
13              end
14         end
15        X=x1*x2;
16  endfunction
```

dft function