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

August 2, 2025

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

List of Figures

Experiment: 1

DFT / IDFT of given Discrete Time Signal

check Appendix AP 2 for dependency:

`dft.sci`

Scilab code Solution 1.1 DFT of a sequence

```
// Experiment Number : 1.1
//Write a program to find the Discrete Fourier
   Transform (DFT) of a discrete time signal
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                          Enrolment
   Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
   Hyderabad


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

```

```
clc;
clear;
close;

exec("C:\Users\HAI\Documents\New folder sci\dft.sci"); //path of dft.sce file in my computer
x=input ('enter the time domain signal x='); //time domain signal
N=input ('enter the DFT length N=');
X=dft(x,N); //computind the DFT by calling the dft function
X1=abs(X); //magnitude of frequency domain signal X(k)
disp('Magnitude Response of DFT is');
disp(X1);
X2=atan(imag(X),real(X)); //phase of frequency domain signal X(k)
disp('Phase Response of DFT is');
disp(X2);

//ploting the magnitude spectrum

subplot(2,1,1);
k=0:1:N-1
plot2d3(k,X1);
xlabel('frequency f');
ylabel('amplitude');
title('magnitude spectrum of X(k)');

//plotting the phase spectrum

subplot(2,1,2);
plot2d3(k,X2);
xlabel('frequency f');
ylabel('phase angle');
title('phase spectrum of X(k)');

```

```


//enter the time domain signal x=[1 2 1 0]

//enter the DFT length N=4


// Magnitude Response of DFT is

//    4.    2.    0.    2.

// Phase Response of DFT is

//    0.  -1.5707963    0.    1.5707963
```

check Appendix AP 1 for dependency:

`idft.sci`

Scilab code Solution 1.2 IDFT of a sequence

```
// Experiment Number : 1.2
//Write a program to find the Inverse Discrete Fourier Transform (IDFT) of a discrete time signal
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                              Enrolment Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology, Hyderabad


```

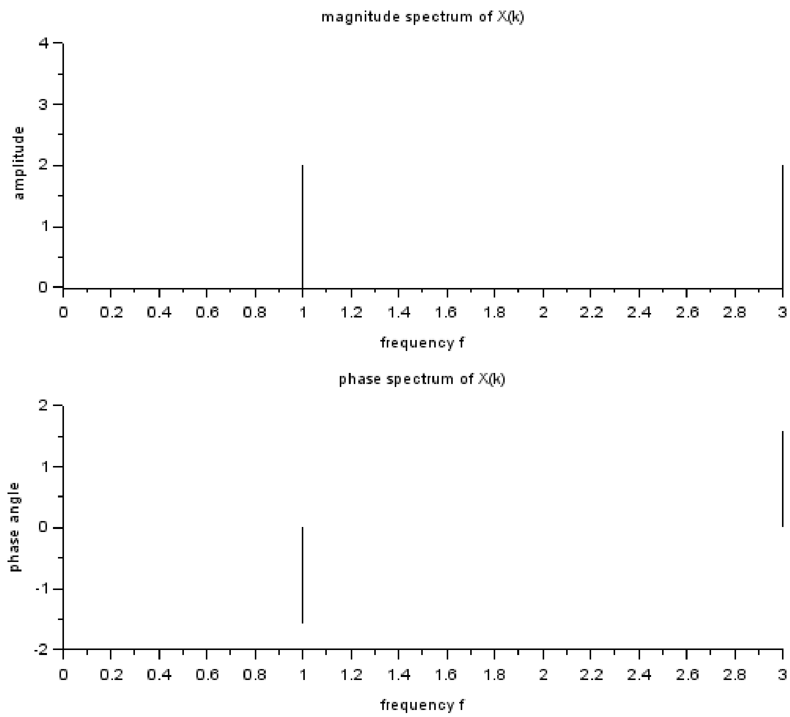

Figure 1.1: DFT of a sequence

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


clc;
clear;
close;

exec("C:\Users\HAI\Documents\New folder sci\idft.sci"); //path of idft.sce file in my computer

X=input('enter the frequency domain signal X='); // frequency domain signal
N=length(X);
x=idft(X,N); //computing the IDFT by calling the idft function
disp('IDFT of given signal 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 given signal is

// 1. 1. 1. 1.
```

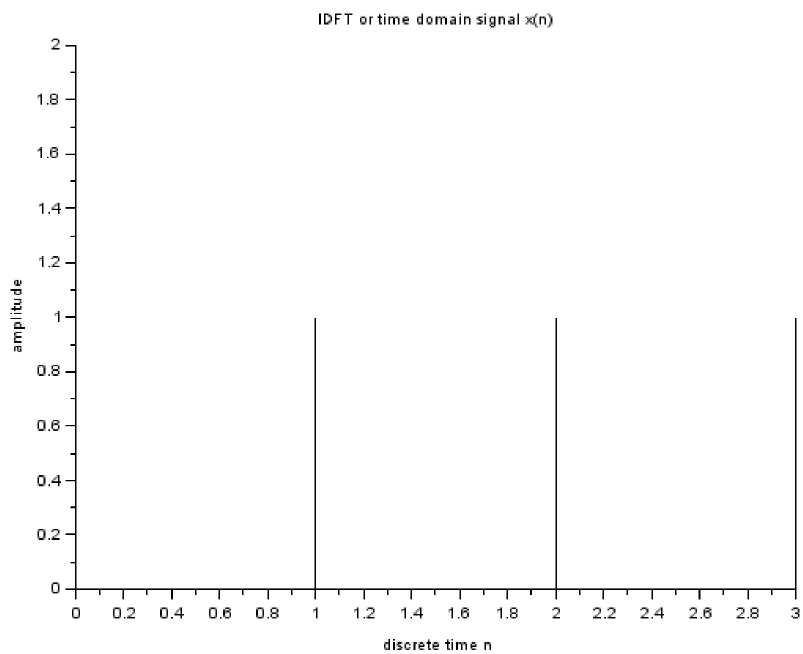

Figure 1.2: IDFT of a sequence

Experiment: 2

Frequency Response of a System

Scilab code Solution 2.0 Frequency Response

```
// Experiment Number : 2
//Write a program to find the Frequency Response of
    a system with transfer function H(Z)=1/[1-0.9Z
    ^-1]
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                        Enrolment
    Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
    Hyderabad


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

clc;
clear;
close;
```

```

n=input ('enter the number of points for frequency response  n=');

//frequency response of system
w=0:2*%pi/n:2*%pi;
h=zeros(1,length(w));
for x=1:length(w)
    h(x)=1/(1-0.9*exp(-%i*w(x)));
end
h1=abs(h);  //magnitude of transfer function
h2=atan(imag(h),real(h));    //phase of the transfer function

//plotting the magnitude spectrum
subplot(2,1,1);
plot(w,h1);
xlabel('frequency w');
ylabel('amplitude');
title('magnitude response of sytem H(w)');

//plotting the phase spectrum
subplot(2,1,2);
plot(w,h2);
xlabel('frequency w');
ylabel('phase angle');
title('phase response of sytem H(w)');


//enter the number of points for frequency response n=50
```

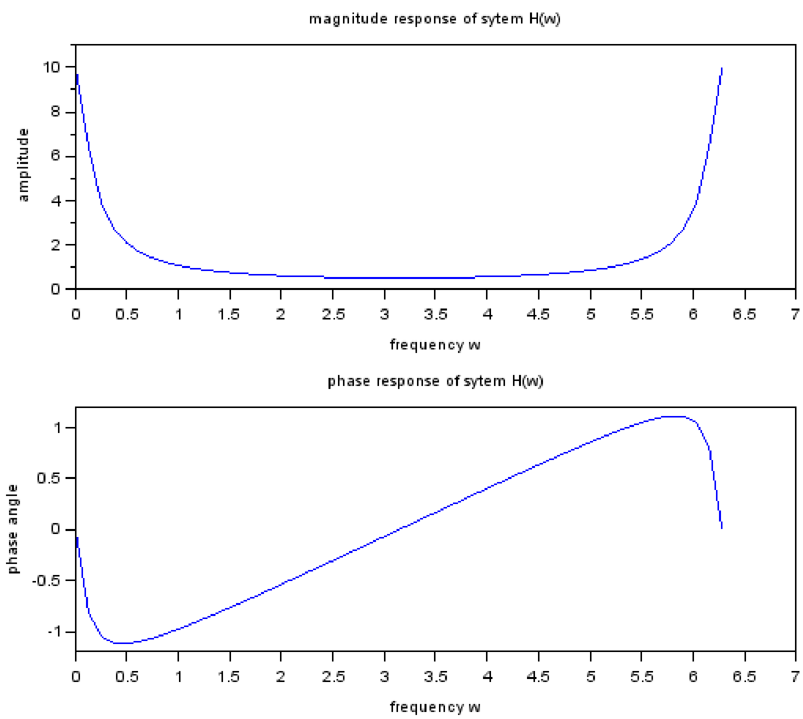

Figure 2.1: Frequency Response

Experiment: 3

Implementation of FFT of a given Sequence

Scilab code Solution 3.0 FFT of a Sequence

```
// Experiment Number : 3
//Write a program to find the FFT and Inverse FFT of
    a discrete time signal
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                         Enrolment
   Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
   Hyderabad
//


// 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);

//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 magnitude spectrum

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

//plotting the phase spectrum

subplot(2,2,2);
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

n=0:1:N-1
subplot(2,2,3);
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.
```

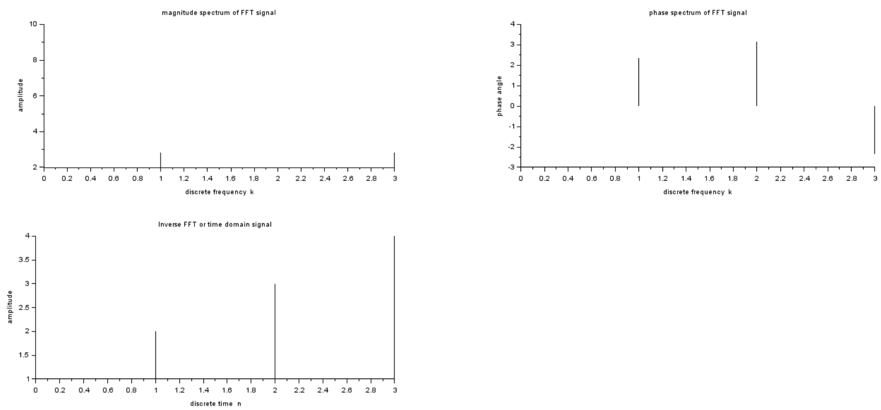

Figure 3.1: FFT of a Sequence

Experiment: 4

Determination of Power Spectrum of a given Signal

Scilab code Solution 4.0 Power Spectral Density of a sequence

```
// Experiment Number : 4
//Write a program to find the power spectral density of a signal
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                    Enrolment Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology, 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);
N=length(x);
n=0:1:N-1
subplot(3,1,1);
plot2d3(n,x);
xlabel('discrete time  n');
ylabel('amplitude');
title('time domain input signal  x(n)');

//generating and plotting autocorrelation signal
R=xcorr(x,x);
disp(R);
N1=length(R);
n1=0:1:N1-1
subplot(3,1,2);
plot2d3(n1,R);
xlabel('discrete time  n1');
ylabel('amplitude');
title('Autocorrelation function  R(n1)');

//generating and plotting the power spectral density
    signal
P=fft(R);
disp(P);
N2=length(P);
k=0:1:N2-1
subplot(3,1,3);
plot2d3(k,P);
xlabel('discrete frequency  k');
ylabel('amplitude');
title('power spectral density  P(k)');


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

```


//     1.     2.     3.     4.


//            column 1 to 6

//     4.     11.     20.     30.     20.     11.

//            column 7

//     4.


//            column 1 to 2

//     100.   -38.594245 - 18.586009i

//            column 3

//     3.9066412 + 4.8987731i

//            column 4

//   -1.3123959 - 5.749982i

//            column 5

//   -1.3123959 + 5.749982i

//            column 6

//     3.9066412 - 4.8987731i

//            column 7

//   -38.594245 + 18.586009i
```

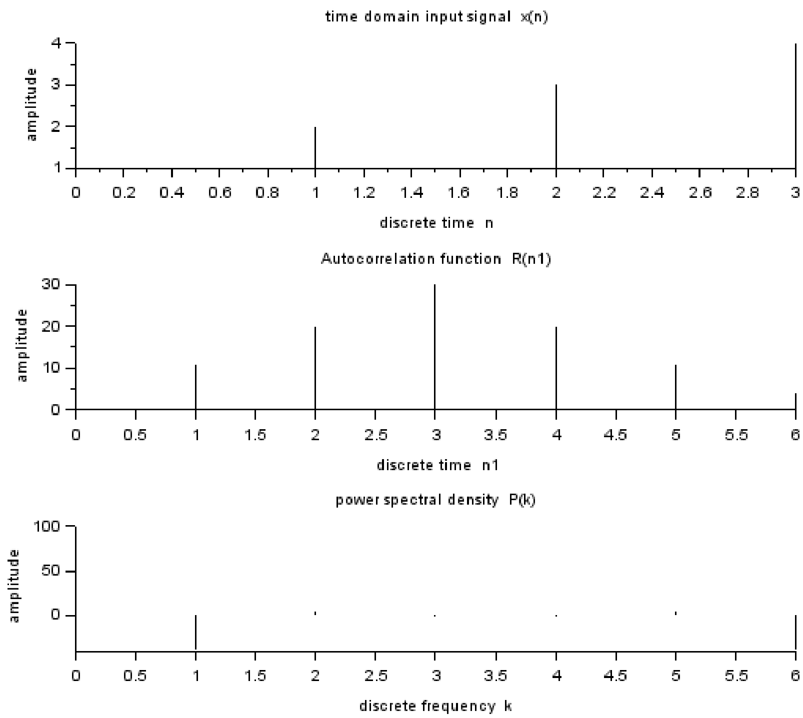

Figure 4.1: Power Spectral Density of a sequence

Experiment: 5

Implementation of Lowpass FIR Filter for given specifications

Scilab code Solution 5.0 Lowpass FIR Filter

```
// Experiment Number : 5
//Write a program to generate Lowpass FIR Filter for
    given specifications
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                          Enrolment
   Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
   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 filter  N=');

// finding the magnitude response of Lowpass FIR Filter

w1 =(2*%pi)*(fc/fs);
disp(w1,'digital cutoff frequency in radians');
wc1 =w1/%pi;
disp(wc1,'normalized digital cutoff frequency in radians');
[wft,wfm,fr]=wfir('lp',N +1,[wc1/2,0],'re',[0,0]);
disp(wft,'impulse response of Lowpass FIR filter:h(n)=');
a=gca();
plot(2*fr,wfm);
xlabel('normalized digital frequency w');
ylabel('magnitude ');
title('magnitude response of Lowpass FIR Filter');


//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
```

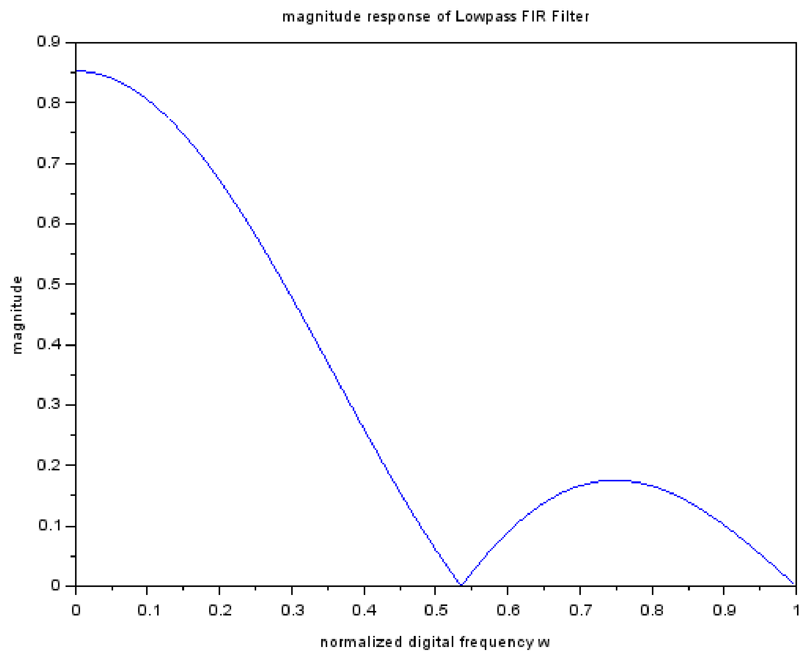

Figure 5.1: Lowpass FIR Filter

```
//    0.24

// impulse response of Lowpass FIR filter
// :h(n)=

//           column 1 to 3

//    0.1920103    0.2343554    0.2343554

//           column 4

//    0.1920103
```

Experiment: 6

Implementation of IIR Filter for given specifications

Scilab code Solution 6.1 Lowpass IIR Filter

```
// Experiment Number : 6.1
//Write a program to generate lowpass IIR Filter
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                          Enrolment
   Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
   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 lowpass  Type-1 Chebyshev  IIR filter

[Hz3]=iir(N,'lp','cheb1',[fp/2,0],[0.2,0]);
[Hw3,w3]=frmag(Hz3,256);
subplot(2,2,2);
plot2d3(w3,abs(Hw3));
xlabel('frequency');
ylabel('magnitude');
title('type-I chebyshev Lowpass  IIR filter');

//generating lowpass  Type-II Chebyshev  IIR filter

[Hz4]=iir(N,'lp','cheb2',[fp/2,0],[0,0.1]);
[Hw4,w4]=frmag(Hz4,256);
subplot(2,2,3);
plot2d3(w4,abs(Hw4));
xlabel('frequency');
ylabel('magnitude');
title('type-II chebyshev Lowpass  IIR filter');

//enter the cutoff frequency  fc=1000

```

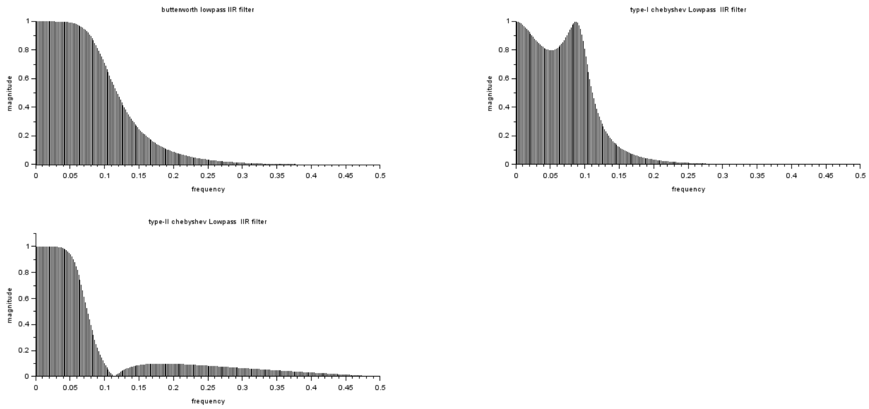

Figure 6.1: Lowpass IIR Filter

```
//enter the sampling frequency  fs=10000

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

Scilab code Solution 6.2 Highpass IIR Filter

```
// Experiment Number : 6.2
//Write a program to generate Highpass IIR Filter
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                         Enrolment
    Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
    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 Highpass Butterworth IIR filter

[Hz2]=iir(N,'hp','butt',[fp/2,0],[0,0]);
[Hw2,w2]=frmag(Hz2,256);
subplot(2,2,1);
plot2d3(w2,abs(Hw2));
xlabel('frequency');
ylabel('magnitude');
title('butterworth highpass IIR filter');

//generating Highpass  Type-1 Chebyshev  IIR filter

[Hz3]=iir(N,'hp','cheb1',[fp/2,0],[0.2,0]);
[Hw3,w3]=frmag(Hz3,256);
subplot(2,2,2);
plot2d3(w3,abs(Hw3));
xlabel('frequency');
ylabel('magnitude');
title('type-I chebyshev Highpass filter');

//generating Highpass  Type-II Chebyshev  IIR filter

[Hz4]=iir(N,'hp','cheb2',[fp/2,0],[0,0.1]);
[Hw4,w4]=frmag(Hz4,256);
subplot(2,2,3);
plot2d3(w4,abs(Hw4));
xlabel('frequency');
```

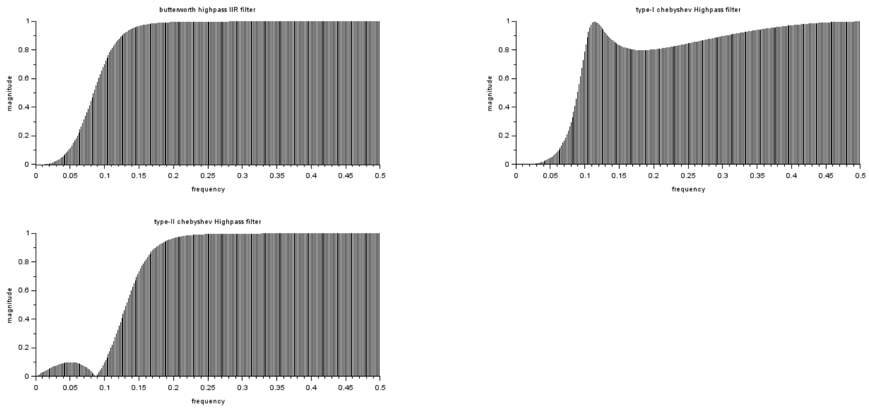

Figure 6.2: Highpass IIR Filter

```
ylabel('magnitude');
title('type-II chebyshev Highpass filter');

//enter the cutoff frequency  fc=1000

//enter the sampling frequency  fs=10000

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

Experiment: 7

Generation of DTMF Signals

Scilab code Solution 7.0 DTMF Signals Generation

```
// Experiment Number : 7
//Write a program to generate the Dual Tone Multi
    Frequency(DTMF) Signal
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                           Enrolment
    Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
    Hyderabad
//



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

clc;
clear;
close;

fs=input('enter the sampling frequency  fs=');
```

```
Ts=1/fs;
num_samples=input ('enter the number of samples=');
dial_number=input ('enter the dial number=');
T=Ts*(0:num_samples-1);

//Generating DTMF Signals

select dial_number    //selecting dial number on
   keypad
case 0 then
    f1=941;
    f2=1336;
case 1 then
     f1=697;
     f2=1209;
 case 2 then
     f1=697;
     f2=1336;
 case 3 then
     f1=697;
     f2=1477;
 case 4 then
     f1=770;
     f2=1209;
 case 5 then
     f1=770;
     f2=1336;
 case 6 then
     f1=770;
     f2=1477;
 case 7 then
     f1=852;
     f2=1209;
 case 8 then
     f1=852;
     f2=1336;
 case 9 then
     f1=852;
```

```
        f2=1477;
  case 'A' then
        f1=697;
        f2=1633;
  case 'B' then
        f1=770;
        f2=1633;
  case 'C' then
        f1=852;
        f2=1633;
  case '*' then
        f1=941;
        f2=1209;
  case '#' then
        f1=941;
        f2=1477;
  case 'D' then
        f1=941;
        f2=1633;
end
first_sine=cos(2**%pi*f1*T);
second_sine=cos(2**%pi*f2*T);
dtmf_signal=first_sine+second_sine;
plot(dtmf_signal);
xlabel(' time t');
ylabel('amplitude');
title('Dual Tone Multi Frequency(DTMF) signal');


//enter the sampling frequency  fs=8000

//enter the number of samples=100

//enter the dial number=9
```

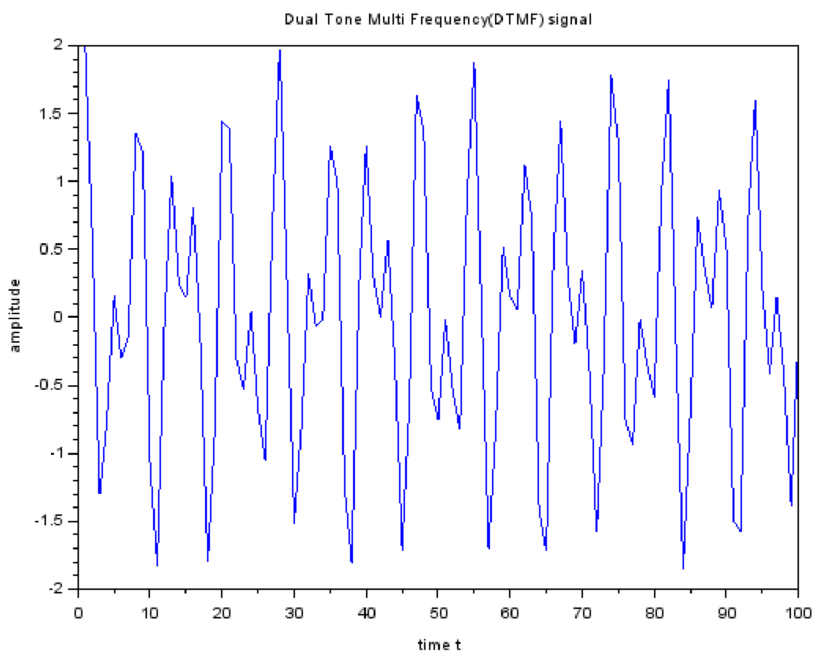

Figure 7.1: DTMF Signals Generation

Experiment: 8

Implementation of Decimation Process

Scilab code Solution 8.0 Decimation of a Signal

```
// Experiment Number : 8
//Write a program to implement the decimation process
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                    Enrolment Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology, 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



//          column 1 to 3

//    0.    0.9510565    0.5877853

//          column 4 to 5

//  -0.5877853  -0.9510565
```

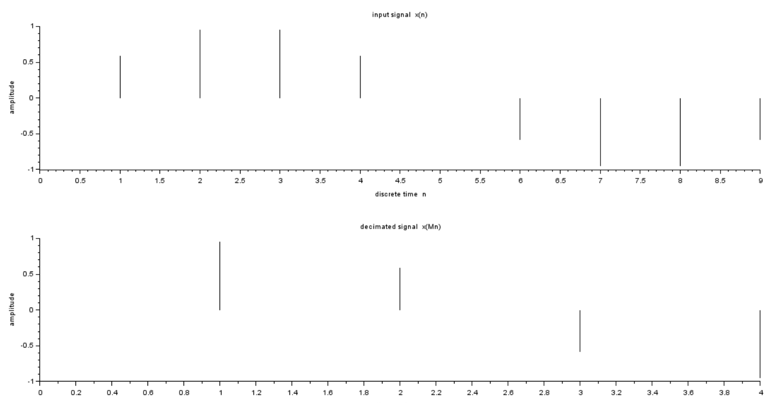

Figure 8.1: Decimation of a Signal

Experiment: 9

Implementation of Interpolation Process

Scilab code Solution 9.0 Interpolation of a Signal

```
// Experiment Number : 9
//Write a program to implement the Interpolation
    process
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                          Enrolment
    Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
    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 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(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
//            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
//            column 1 to 4
//    0.    0.    0.5877853    0.
//            column 5 to 8
//    0.9510565    0.    0.9510565    0.
//            column 9 to 12
```

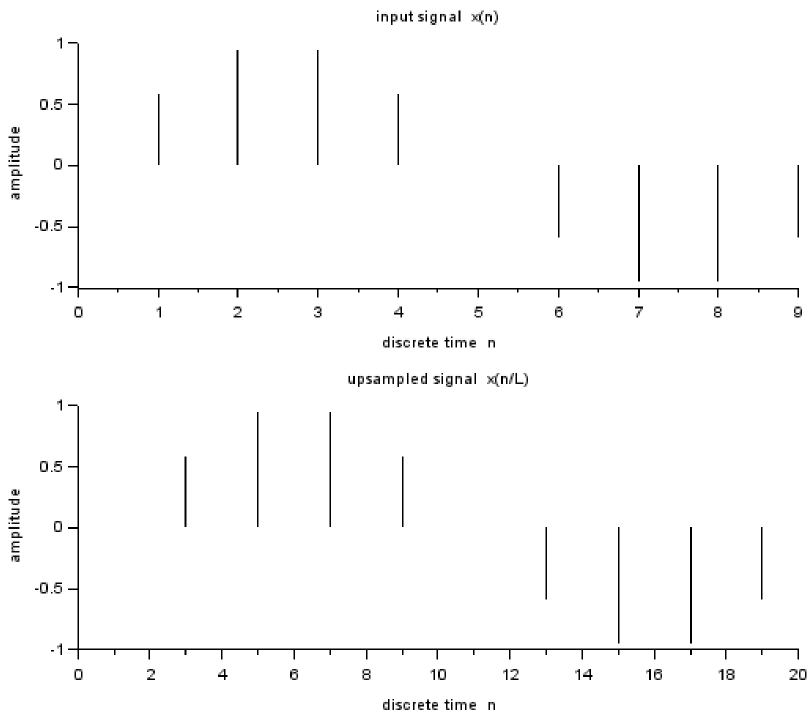

Figure 9.1: Interpolation of a Signal

```

//      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.
```

Experiment: 10

Implementation of Sampling rate conversion by a factor I/D

Scilab code Solution 10.0 Sampling Rate converter

```
// Experiment Number : 10
//Write a program to implement the sampling rate
    conversion by a factor I/D or L/M
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                          Enrolment
    Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
    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);
disp(x);

//plotting the input signal

subplot(2,1,1);
plot2d3(n,x(1:30));
xlabel('discrete time n');
ylabel('amplitude');
title('input signal x(n)');

// sampling rate converted by a factor L/M signal
y=intdec(x,L/M);
disp(y);

//plotting the 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('sampling rate converted signal');


//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



//          column 1 to 3

//    0.    0.4257793  -0.7705132

//          column 4 to 6

//    0.9685832  -0.9822873    0.809017

//          column 7 to 9

//  -0.4817537    0.0627905    0.3681246

//          column 10 to 12

//  -0.7289686    0.9510565  -0.9921147

//          column 13 to 15

//    0.8443279  -0.5358268    0.1253332

//          column 16 to 18

//    0.309017  -0.6845471    0.9297765

//          column 19 to 21
```

```
//   -0.9980267    0.8763067   -0.5877853

//             column 22 to 24

//    0.1873813    0.2486899   -0.637424

//             column 25 to 27

//    0.9048271   -1.    0.9048271

//             column 28 to 30

//   -0.637424    0.2486899    0.1873813


//             column 1 to 3

//   -0.0044437    0.8162176    0.430223

//             column 4 to 6

//   -0.6991301   -0.774957    0.3930428

//             column 7 to 9

//    0.9730269    0.0130957   -0.986731

//             column 10 to 12

//   -0.431533    0.8134607    0.7777726

//             column 13 to 15

//   -0.4861974   -0.983244    0.0672342

//             column 16 to 18

```

```
//    1.0072059    0.3636808   -0.8440229

//            column  19  to  21

//   -0.7245249    0.5240108    0.9466128

//            column  22  to  24

//   -0.1075454   -0.987671   -0.3264978

//            column  25  to  27

//    0.8398842    0.6957976   -0.5313831

//            column  28  to  30

//   -0.930282    0.1208895    0.9854806

//            column  31  to  33

//    0.3134607   -0.8509991   -0.6889908

//            column  34  to  36

//    0.5525013    0.9342202   -0.1468245

//            column  37  to  39

//   -1.0024705   -0.2888443    0.8807504

//            column  40  to  42

//    0.67167   -0.592229   -0.9289559

//            column  43  to  45

//    0.191825    1.0120117    0.2442462

```

```
//             column  46  to  48

//   -0.9054772   -0.6329803    0.6303453

//             column  49  to  51

//    0.9003833   -0.2401604   -0.9955563

//             column  52  to  54

//   -0.1887578    0.9003833    0.5708187

//             column  55  to  57

//   -0.6329803   -0.8243498    0.2442462

//             column  58  to  60

//    0.8735432    0.191825   -0.5116523
```

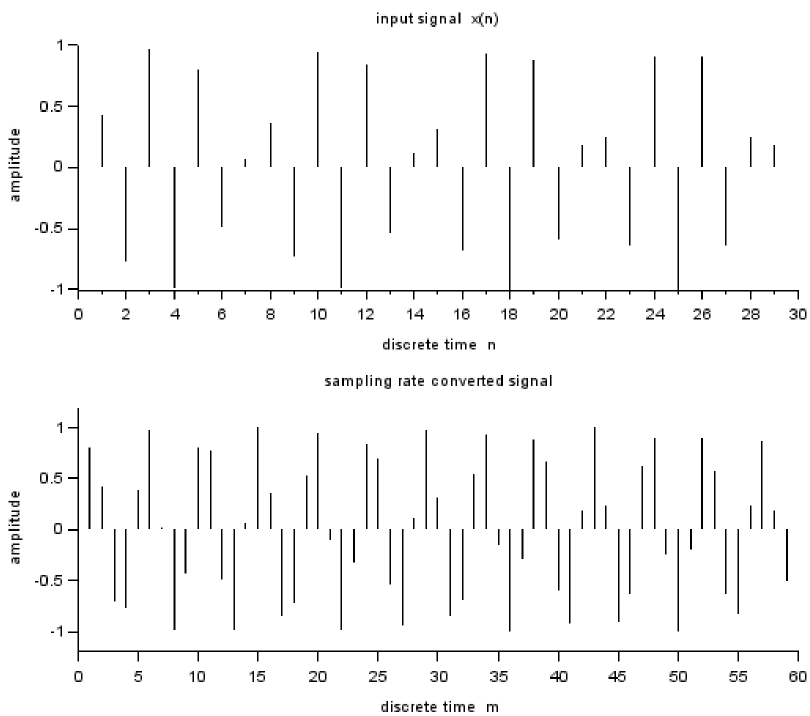

Figure 10.1: Sampling Rate converter

Experiment: 11

Impulse response of First order and Second order systems

Scilab code Solution 11.0 Impulse Response of a system

```
// Experiment Number : 11
//Write a program to find the impulse response of
   first order and second order systems
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                    Enrolment
   Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
   Hyderabad


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

clc;
clear;
close;
```

```

// Impulse Response of first order system described by difference equation   y(n)+2y(n-1)=x(n)

a1=input ('enter the coefficients of input vector of first order system  a1=');
b1=input ('enter the coefficients of output vector of first order system  b1=');
n1=input ('enter the lower of impulse response  n1=');
n2=input ('enter the upper of impulse response  n2=');
n=n1:n2

//generate the impulse input function
x=zeros(1,length(n));
for i=1:length(n)
    if n(i)==0
        x(i)=1;
end
end

h1=filter(a1,b1,x);     //finds the impulse response of first order system
disp(h1);          //display the values of impulse response in console window
subplot(2,1,1)
plot2d3(n,h1);       //to plot the impulse response of sytem in graphical window
xlabel('discrete time  n');
ylabel('amplitude');
title('impulse response of first order system  h1(n)');

// Impulse Response of Second order system described by difference equation
// y(n)+0.5y(n-1)+0.3y(n-2)=x(n)+5x(n-1)

```

```
a2=input ('enter the coefficients of input vector of second order system  a2=');
b2=input ('enter the coefficients of output vector of second order system  b2=');
h2=filter(a2,b2,x);      //finds the impulse response of system
disp(h2);           //display the values of impulse response in console window
subplot(2,1,2)
plot2d3(n,h2);       //to plot the impulse response of sytem in graphical window
xlabel('discrete time  n');
ylabel('amplitude');
title('impulse response of second order system  h2(n)');

//enter the coefficients of input vector of first order system  a1=[1]

//enter the coefficients of output vector of first order system  b1=[1  2]

//enter the lower of impulse response  n1=0

//enter the upper of impulse response  n2=5


//    1.   -2.    4.   -8.    16.   -32.

//enter the coefficients of input vector of second order system  a2=[1  5]

//enter the coefficients of output vector of second order system  b2=[1  0.5  0.3]



//   column 1 to 5
```

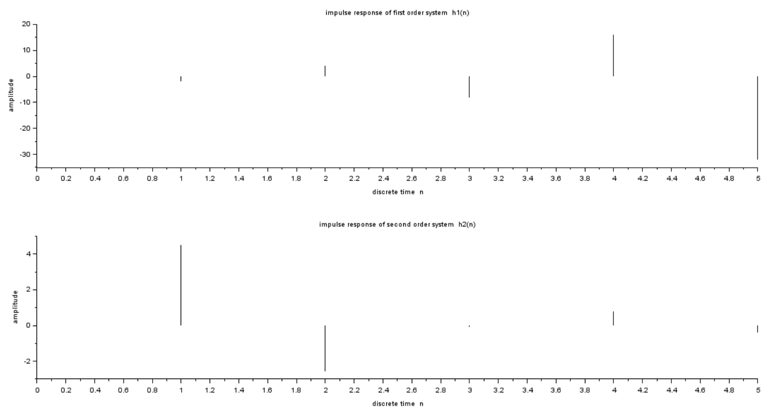

Figure 11.1: Impulse Response of a system

```

//       1.       4.5     -2.55     -0.075       0.8025

//                column  6

//     -0.37875
```

Experiment: 12

Finding the Fourier Series Coefficients of a Periodic Discrete Time Signal

Scilab code Solution 12.0 Fourier Series Coefficients

```
// Experiment Number : 12
//Write a program to generate Fourier Series
    Coefficients of a Periodic Signal
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                        Enrolment
    Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
    Hyderabad


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

clc;
clear;
```

```
close;

//generating input signal
n = 0:0.01:5;
N = input('enter the time period N=');
Wo = 2*%pi/N;  //fundamental frequency
A = input('enter the amplitude of sinusoidal signal
   A=');
x = A*sin(Wo*n);  //input signal x(n)

//finding fourier series coefficients
for k =0:N-2
      D(k+1,:) = exp(-%i*Wo*n.*k);
      C(k+1) = x*D(k+1,:)'/length(n);
      end
C =C'
C_conju = conj(C);
Ck = [C_conju($:-1:1),C(2:$)]
k = -(N-2):(N-2);
//
figure
C = gca();
C.y_location = "origin";
C.x_location = "origin";
C.data_bounds=[-8,-1;8,1];
pol1 = C;
pol1.thickness = 3;
plot2d3('gnn',k,-imag(Ck),5)
xlabel("discrete time");
ylabel("Amplitude");
pol1 = C;
pol1.thickness = 3;
plot2d3('gnn',N+k,-imag(Ck),5)
pol1 = C.children(1).children(1);
pol1.thickness = 3;
plot2d3('gnn',-(N+k),-imag(Ck($:-1:1)),5)
pol1 = C;
pol1.thickness = 3;
```

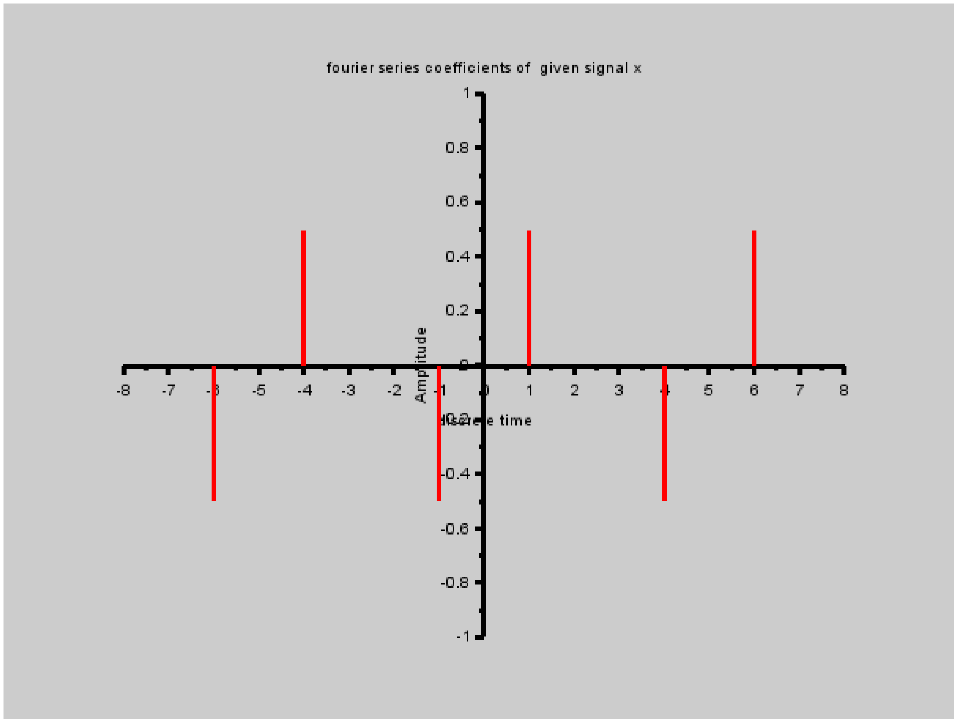

Figure 12.1: Fourier Series Coefficients

```
title('fourier series coefficients of  given signal
   x')

//enter the time period N=5

//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

```
// Experiment Number : 13
//Write a program to generate a sinusoidal signal
   based on a recursive difference equation p(k+2)=a
   *p(k+1)+b*p(k)
//Digital Signal Processing Laboratory
//B.Tech III Year II Sem
// Student Name :                          Enrolment
   Number :
// Course Instructor:K.Manohar
// Sreyas Institute of Engineering and Technology,
   Hyderabad


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

clc;
```

```
clear;
close;

//generating the input signal
n=0:3;
A=input ('enter the amplitude of input sinusoidal
   signal  A=');
N=input ('enter the fundamental time period of input
    sinusoidal signal  N=');
y=A*sin(2*%pi*n/N);

//coefficients of difference equation
a=y(3)/y(2);
b=(y(4)-a*y(3))/y(2);
disp('The coefficients of the  difference  equation
    are');
disp(a);
disp(b);

//generation of sinusoidal signal through recursive
   equation p(k+2)=a*p(k+1)+b*p(k)

for k=1:1:119
    p(1)=y(1);  //initial values
    p(2)=y(2);  //initial values
    p(k+2)=a*p(k+1)+b*p(k);
end

plot2d3(p);
xlabel('discrete time');
ylabel('amplitude');
title('discrete time sinusoidal signal');


//enter the amplitude of input sinusoidal signal  A
   =10

//enter the fundamental time period of input
```

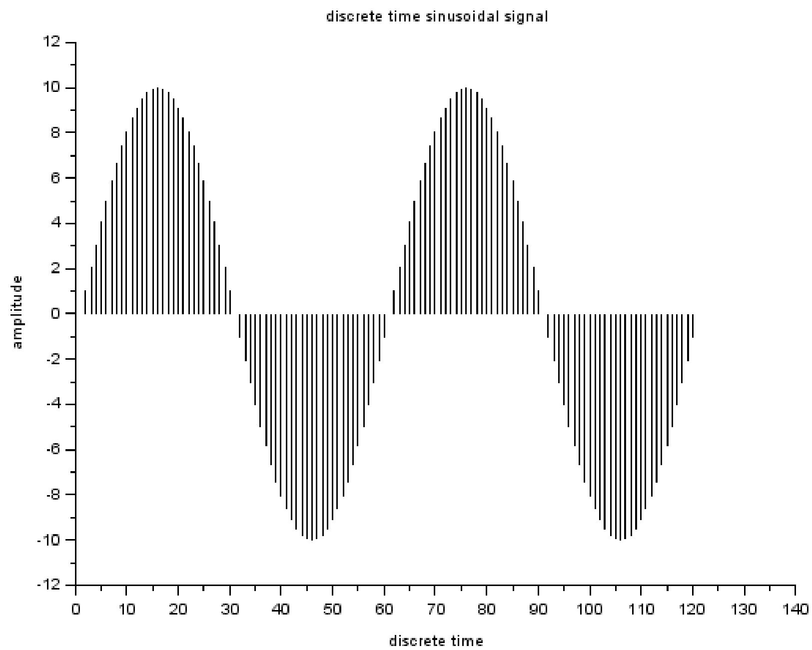

Figure 13.1: recursive sinusoid generation

```
    sinusoidal signal  N=60


// The coefficients of the   difference
// equation are
//   1.9890438

//  -1.
```

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

Scilab code AP 2

```
// Generating the DFT function

function X=dft(x,N)
    L=length(x);
    if(L>N)
        disp('error because  L should be less than N');
    end
    x1=[x zeros(1,N-L)];    // zero padding the sequence x(n)
    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=x1*x2;
endfunction
```

dft function