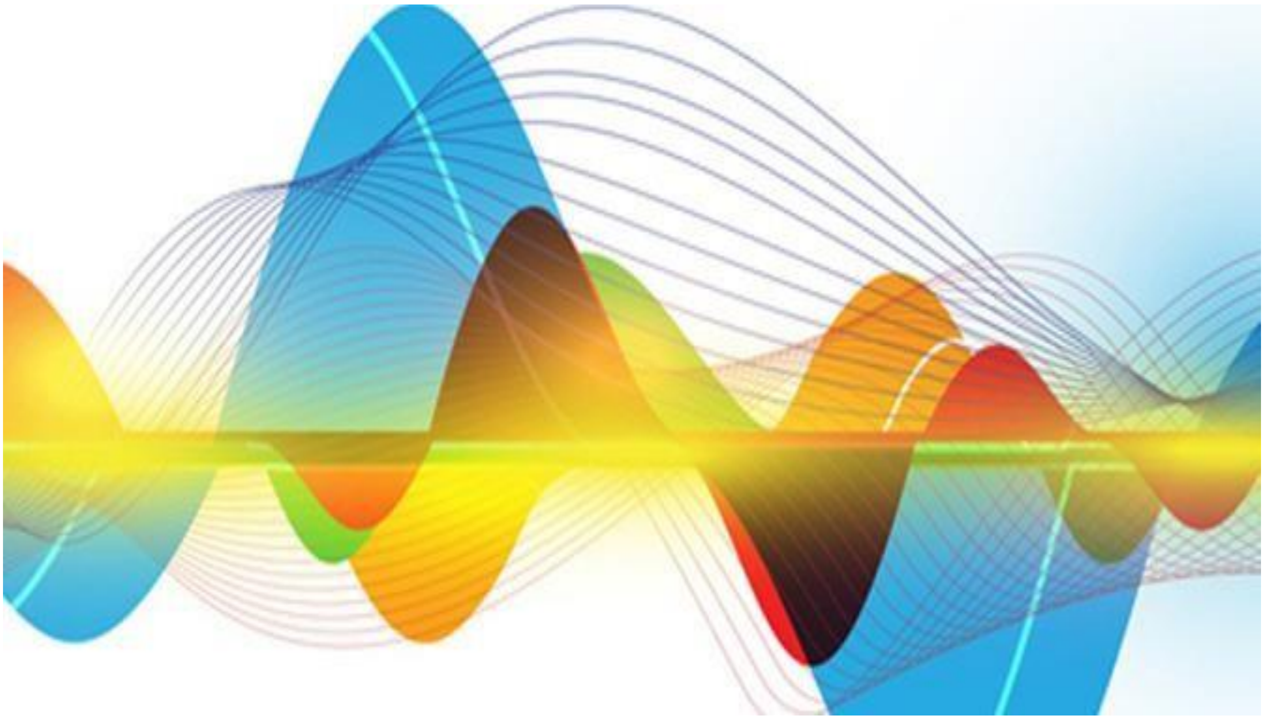
DIGITAL SIGNAL PROCESSING LAB

LAB MANUAL

B.Tech III yr - II sem (R-18 JNTUH)



Department of Electronics and Communication Engineering

MALLAREDDY INSTITUTE OF TECHNOLOGY

(Sponsored by Malla Reddy Educational society)

Approved by AICTE, New Delhi & Affiliated to JNTU Hyderabad

Maisammaguda, Dhulapally Post (via Hakimpet), Secunderabad-500 100

(2021-22)

CONTENTS

Vision of the Institute:

To provide plenty of opportunities for both students and faculty to monitor continuously their huge potential for meeting the global challenges in a good environment.

Mission of the Institute:

- To create both necessary and sufficient technical and non-technical competence at International level by providing an academic climate in which students get opportunities to constantly release their vast inner potential.
- To observe ethical, cultural, socio-economic and constitutional rights and responsibilities among engineering graduates.
- To create a platform of advancements in science and technology for technical interest.

Vision of the Department

To produce globally competent and socially responsible Electronics and Communication Engineers contributing to the advancement of engineering and technology which involves creativity and innovation by providing excellent learning environment with world class facilities.

Mission of the Department

1	To be a center of excellence in instruction, innovation in research, service to the stake holders, profession and public.
2	To prepare graduates to enter a rapidly changing field as an Electronics and Communication Engineer and update with latest developments.
3	To prepare a graduates to possess leadership and communication skills, to work effectively as a team and to take the social and ethical responsibility.

Program Outcomes:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives:

- To provide thorough knowledge in Electronics and Communication Engineering subjects including theoretical knowledge and practical training for preparing physical models pertaining.
- To provide training in soft skills via English language, communication, verbal, logical, analytical, comprehension, team building, inter personal relationship, group discussion etc… for placements.
- To inculcate the habit of lifelong learning for career development through successful completion of advanced degrees, professional development courses, industrial training etc…

Course Outcomes:

- Able to generate elementary signals/ waveforms and perform arithmetic operations on signals and to Plot DFT / IDFT of given DT signal.
- Able to plot frequency response of a given system and verify the properties of LTI system and to Implement FFT of given sequence and identify the reduction of computations using FFT
- Able to Implement LP & HP FIR filter for a given sequence and calculate the filter coefficients
- Able to Implement and design LP & HP IIR filter for a given sequence. And able to generate Sinusoidal signal through filtering.
- Able to Implement Decimation Process and vary (decrease) the sampling rate. And able to Implement Interpolation Process and vary (increase) the sampling rate.

Editorial Board:

Chief Editor	Head of the department
Faculty Editors	

Lab Instructions

CODE OF CONDUCT FOR THE LABORATORIES

- All students must follow the Dress Code while in the laboratory.
- Footwear is NOT allowed.
- Foods and drinks are NOT allowed.
- All bags must be left at the indicatedplace.
- The lab timetable must be strictly followed.
- Be PUNCTUAL for your laboratory session.
- Program must be executed within the given time.
- Noise must be kept to a minimum.
- Workspace must be kept clean and tidy at all time.
- Handle the systems and interfacing kits with care.
- All students are liable for any damage to the accessories due to their own negligence.
- All interfacing kits connecting cables must be RETURNED if you taken from the lab supervisor.
- Students are strictly PROHIBITED from taking out any items from the laboratory.
- Students are NOT allowed to work alone in the laboratory without the Lab Supervisor
- USB Ports have been disabled if you want to use USB drive consult lab supervisor.
- Report immediately to the Lab Supervisor if any malfunction of the accessories, is there.

Before leaving the lab

- Place the chairs properly.
- Turn off the system properly
- Turn off the monitor.
- Please check the laboratory notice board regularly forupdates.

GENERAL LABORATORY INSTRUCTIONS

Students should be punctual for laboratory session and should not leave the lab without the permission of the teacher.

Each student is expected to have his/her own lab book where they will take notes on the experiments as they are completed. The lab books will be checked at the end of each lab session. Lab notes are a primary source from which you will write your lab reports.

Organization of the Laboratory

It is important that the programs are done according to the timetable and completed within the scheduled time.

You should complete the pre-lab work in advance and utilize the laboratory time for verification only.

The aim of these exercises is to develop your ability to understand, analyze and test them in the laboratory.

A member of staff and a Lab assistant will be available during scheduled laboratory sessions to provide assistance. Always attempt program first without seeking help. When you get into difficulty; ask for assistance.

Assessment:

The laboratory work of a student will be evaluated continuously during the semester for 25 marks. Out of the 25 marks, 15 marks will be awarded for day-to-day work. For each program marks are awarded under three heads:

Internal lab test(s) conducted during the semester carries 25 marks.
End semester lab examination, conducted as per the JNTU regulations, carries 75 marks.

At the end of each laboratory session you must obtain the signature of the teacher along with the marks for the session out of 10 on the lab notebook.

Lab Reports

Note that, although students are encouraged to collaborate during lab, each must individually prepare a report and submit. They must be organized, neat and legible.

Your report should be complete, thorough, understandable and literate.

Your reports should follow the prescribed format, to give your report structure and to make sure that you address all of the important points.

Reports should be submitted within one week after completing a scheduled lab session.

Report Format:

Lab write ups should consist of the following sections:

Aim:

Equipments:

Theory:

Procedure:

Program:

Expected Waveform:

Observation and Calculations:

Results:

Conclusions:

Note: *Diagrams if any must be drawn neatly on left hand side.*

LIST OF EXPERIMENTS

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B.Tech. III Year II Sem.(R18) **Course Code: EC604PC**

Note:
1. The Programs shall be implemented in Software (Using MATLAB / Lab View / C Programming/ Equivalent) and Hardware (Using TI / Analog Devices / Motorola / Equivalent DSP processors).
2. Minimum of 12 experiments to be conducted.

List of Experiments

1. Generation of Sinusoidal Waveform / Signal based on Recursive Difference Equations
2. Histogram of White Gaussian Noise and Uniformly Distributed Noise.
3. To find DFT / IDFT of given DT Signal
4. To find Frequency Response of a given System given in Transfer Function/ Differential equation form.
5. Obtain Fourier series coefficients by formula and using FET and compare for half sine wave.
6. Implementation of FFT of given Sequence
7. Determination of Power Spectrum of a given Signal(s).
8. Implementation of LP FIR Filter for a given Sequence/Signal.
9. Implementation of HP IIR Filter for a given Sequence/Signal
10. Generation of Narrow Band Signal through Filtering
11. Generation of DTMF Signals
12. Implementation of Decimation Process
13. Implementation of Interpolation Process
14. Implementation of I/D Sampling Rate Converters
15. Impulse Response of First order and Second Order Systems.

INTRODUCTION TO MATLAB

MATLAB (MATRIX LABORATORY):

MATLAB is a software package for high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include the following

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow learning and applying specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include Image processing, signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The main features of MATLAB

1. Advance algorithm for high performance numerical computation ,especially in the Field matrix algebra
2. A large collection of predefined mathematical functions and the ability to define one's own functions.
3. Two-and three dimensional graphics for plotting and displaying data
4. A complete online help system
5. Powerful, matrix or vector oriented high level programming language for individual applications.
6. Toolboxes available for solving advanced problems in several application areas

Common Procedure to all Programs in MATLAB

1. Click on the MATLAB Icon on the desktop.

2. MATLAB window open.

3. Click on the 'FILE' Menu on menu bar.

4. Click on NEW M-File from the file Menu.

5. An editor window open, start typing commands.

6. Now SAVE the file in directory.

7. Then Click on DEBUG from Menu bar and Click Run.

Experiment 1

GENERATION OF SINUSOIDAL WAVEFORM

AIM : To Generate continuous time sinusoidal signal, Discrete time cosine signal.

Requirements : Computer with MATLAB software

Theory:

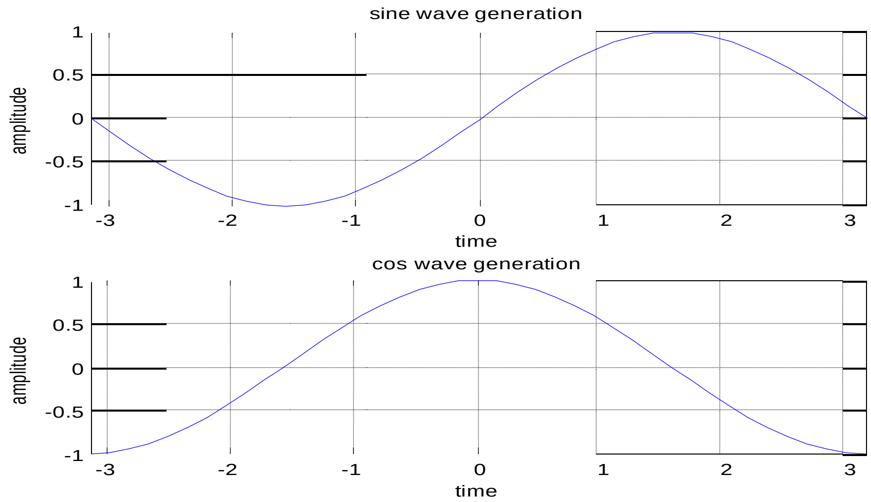
The sine wave or sinusoid is a mathematical function that describes a smooth repetitive oscillation. It occurs often in pure mathematics, as well as physics, signal processing, electrical engineering and many other fields. Its most basic form as a function of time (t) where: • A, the amplitude, is the peak deviation of the function from its center position. • ω, the angular frequency, specifies how many oscillations occur in a unit time interval, in radians per second • φ, the phase, specifies where in its cycle the oscillation begins at t = 0.

PROGRAM:

%**Sine and Cos Wave Generation**

```
clc;
clear all;
x=-2*pi:pi/20:2*pi;
y=sin(x);
subplot(2,1,1);
plot(x,y);
grid;
axis([-pi,pi,-1,1]);
xlabel('time');
ylabel('amplitude');
title('sine wave generation');
y1=cos(x);
subplot(2,1,2);
plot(x,y1);
grid;
axis([-pi,pi,-1,1]);
xlabel('time');
ylabel('amplitude');
title('cos wave generation');
```

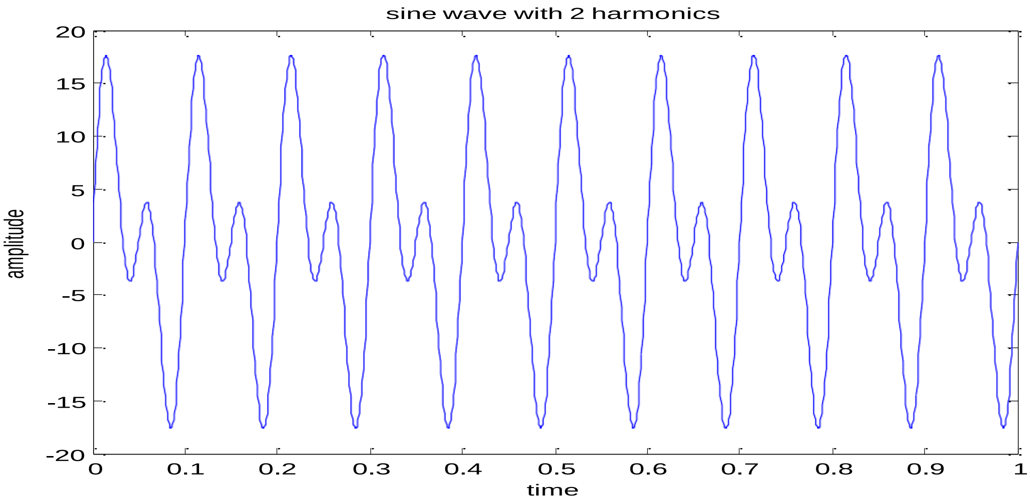
Output:

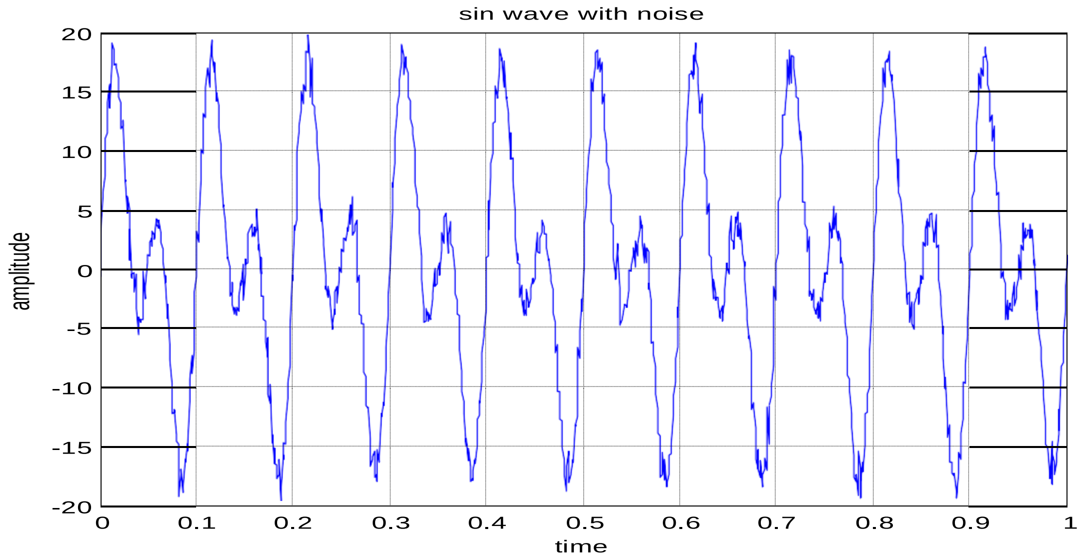


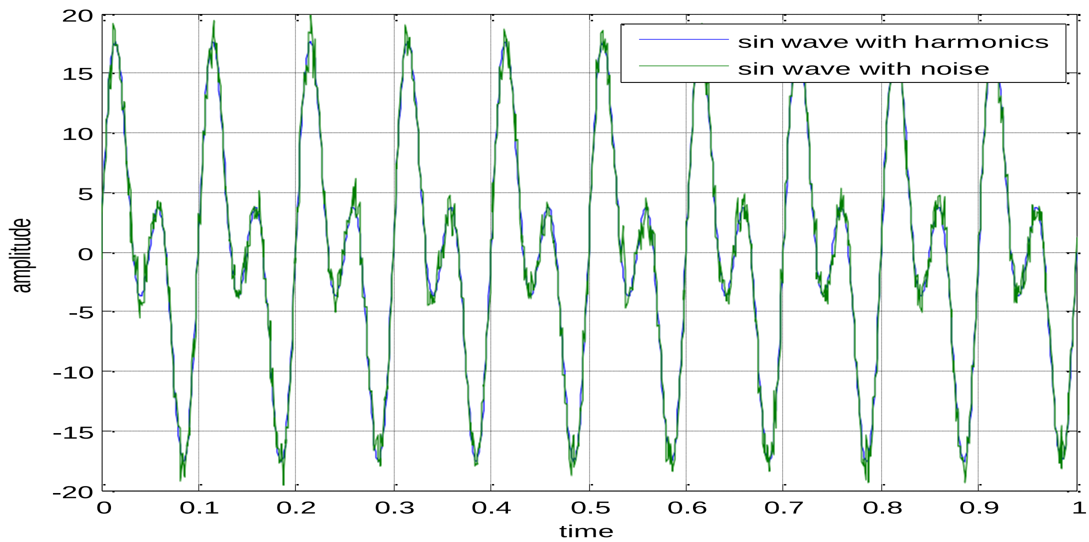
i) Sine wave with 2 harmonics and noise

```
clc;
clear all;
fs=1000;
ts=1/fs;
t=0:ts:1;
x=10*sin(2*pi*10*t)+10*sin(2*pi*2*10*t);
plot(t,x);
grid;
xlabel('time');
ylabel('amplitude');
title('sine wave with 2 harmonics');
grid;
x1=x+randn(size(t));
figure;
plot(t,x1);
grid;
xlabel('time');
ylabel('amplitude');
title('sin wave with noise');
figure;
plot(t,x,t,x1);
grid;
xlabel('time');
ylabel('amplitude');
legend('sin wave with harmonics','sin wave with noise');
```

<u>Output:</u>


sine wave with 2 harmonics
amplitude
time
20
15
10
5
0
-5
-10
-15
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

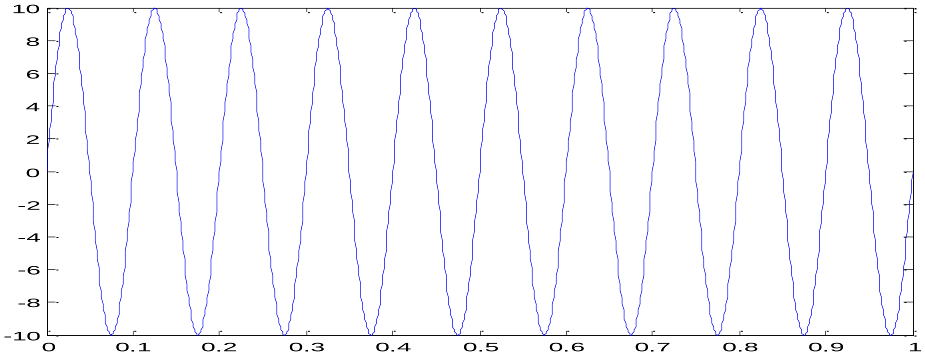

sin wave with noise
amplitude
time
20
15
10
5
0
-5
-10
-15
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

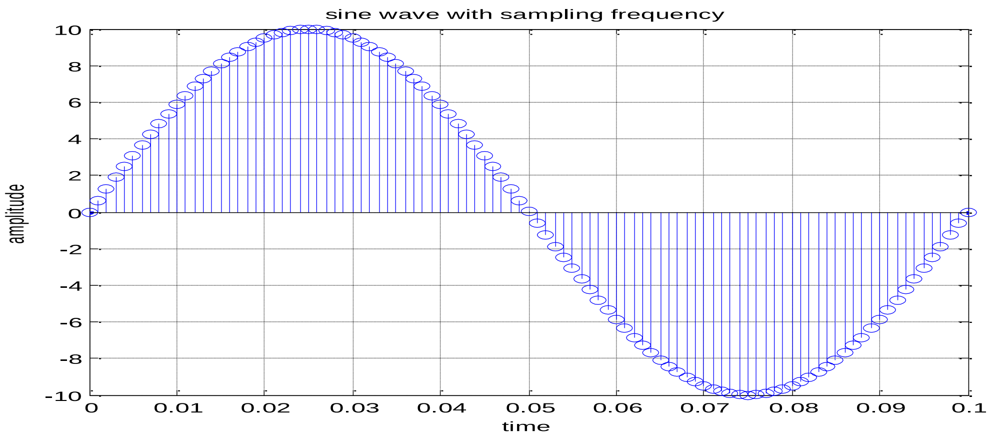

sin wave with harmonics
sin wave with noise
amplitude
time
20
15
10
5
0
-5
-10
-15
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

ii) Sine wave with sampling frequency

```
clc;
clear all;
fs=1000;
ts=1/fs;
t=0:ts:1;
y=10*sin(2*pi*10*t);
plot(t,y);
figure;
stem(t,y);
grid;
xlabel('time');
ylabel('amplitude');
title('sine wave with sampling frequency');
axis([0 0.1 10 -10]);
```

Output:





Result: Continuous time sinusoidal signal, discrete time cosine signal and sum of sinusoidal signal are designed.

Experiment 2

HISTOGRAM OF WHITE GAUSSIAN NOISE AND UNIFORMLY DISTRIBUTED NOISE

AIM: To plot the histogram of a Gaussian noise and uniform noise.

Requirements : Computer with MATLAB software

THEORY:

A white noise signal (process) is constituted by a set of independent and identically distributed (i.i.d) random variables. In discrete sense, the white noise signal constitutes a series of samples that are independent and generated from the same probability distribution. For example, you can generate a white noise signal using a random number generator in which all the samples follow a given Gaussian distribution. This is called White Gaussian Noise (WGN) or Gaussian White Noise. Similarly, a white noise signal generated from a Uniform distribution is called Uniform White Noise. White Gaussian Noise and Uniform White Noise are frequently used in system modelling. In modelling/simulation, a white noise can be generated using an appropriate random generator. White Gaussian Noise can be generated using "randn" function in Matlab which generates random numbers that follow a Gaussian distribution. Similarly, "rand" function can be used to generate Uniform White Noise in Matlab that follows a uniform distribution. When the random number generators are used, it generates a series of random numbers from the given distribution.

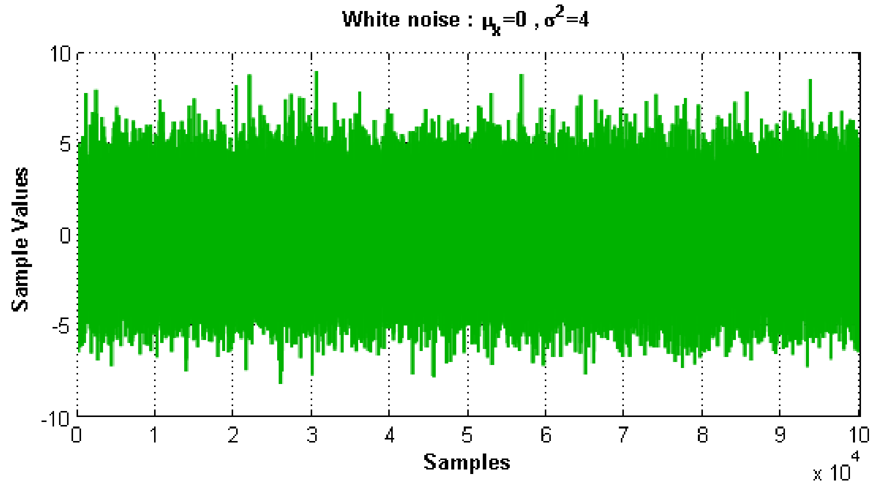
Theoretical PDF of the Gaussian random variable is

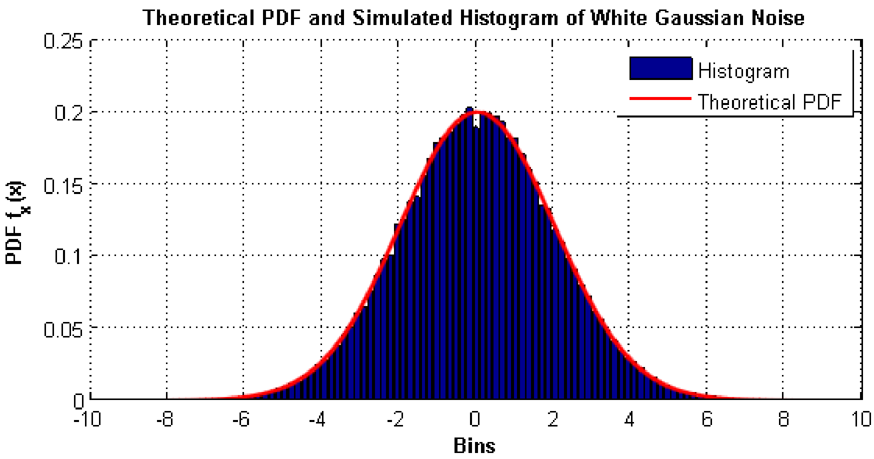
$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Program

```
clear all;
clc;
close all;

L=100000; %Sample length for the random signal
mu=0;
sigma=2;
X=sigma*randn(L,1)+mu;
figure();

subplot(2,1,1)
plot(X);
title(['White noise : \mu_x=',num2str(mu),' \sigma^2=',num2str(sigma^2)])
xlabel('Samples')
```

```
ylabel('Sample Values')
grid on;

subplot(2,1,2)
n=100; %number of Histrogram bins
[f,x]=hist(X,n);
bar(x,f/trapz(x,f)); hold on;
%Theoretical PDF of Gaussian Random Variable
g=(1/(sqrt(2*pi)*sigma))*exp(-((x-mu).^2)/(2*sigma^2));
plot(x,g);hold off; grid on;
title('Theoretical PDF and Simulated Histogram of White Gaussian Noise');
legend('Histogram','Theoretical PDF');
xlabel('Bins');
ylabel('PDF f_x(x)');
```





Result:

The histogram of the white Gaussian noise is plotted using matlab software.

Experiment 3

TO FIND DFT / IDFT OF GIVEN DT SIGNAL

AIM: To find Discrete Fourier Transform and Inverse Discrete Fourier Transform of given digital signal.

Requirements: Computer with MATLAB software

THEORY:

Basic equation to find the DFT of a sequence is given below.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

$$where \ \ W_N^{nk} = e^{-j\frac{2\pi nk}{N}} \text{ [TWIDDLE FACTOR]}$$

Basic equation to find the IDFT of a sequence is given below.

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N}kn} \qquad n = 0, \dots, N-1.$$

PROGRAM:

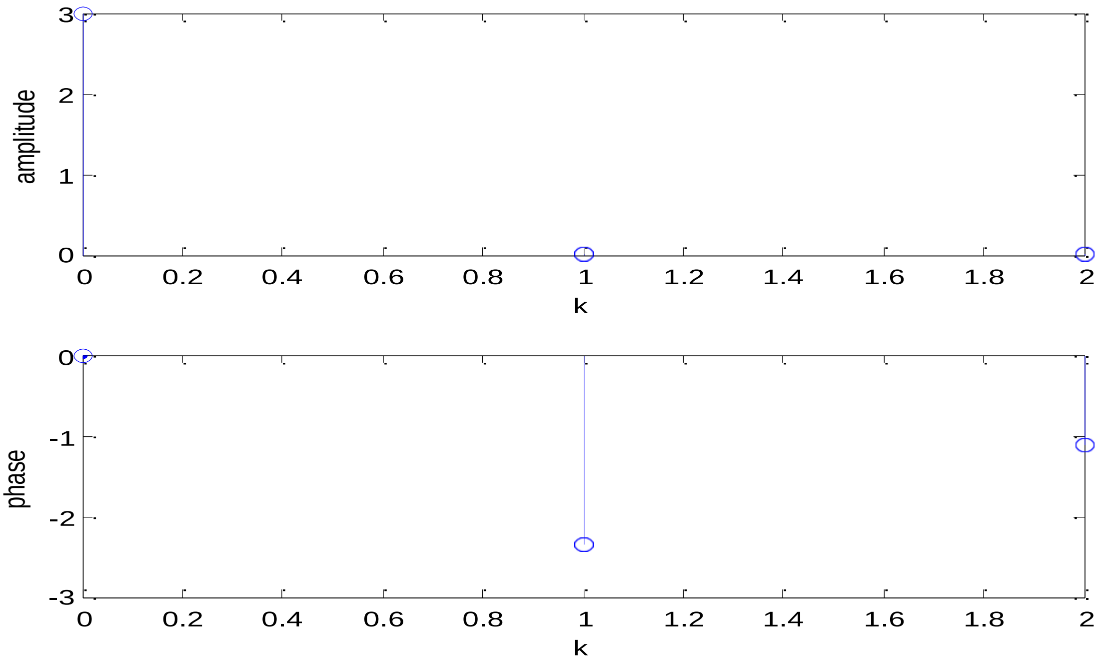
% DFT

```
clear all
x1=[1 1 1]
N=length(x1)
for k=0:1:N-1
   for n=0:1:N-1
p=exp(-i*2*pi*n*k/N)
x2(k+1,n+1)=p
   end
end
Y=(x1*x2)
k=0:1:N-1
subplot(2,1,1)
stem(k,abs(Y))
xlabel('k')
ylabel('amplitude')
subplot(2,1,2)
stem(k,angle(Y))
xlabel('k')
```

ylabel('phase')

Output:

Y = 3.0000 -0.0000 - 0.0000i 0.0000 - 0.0000i

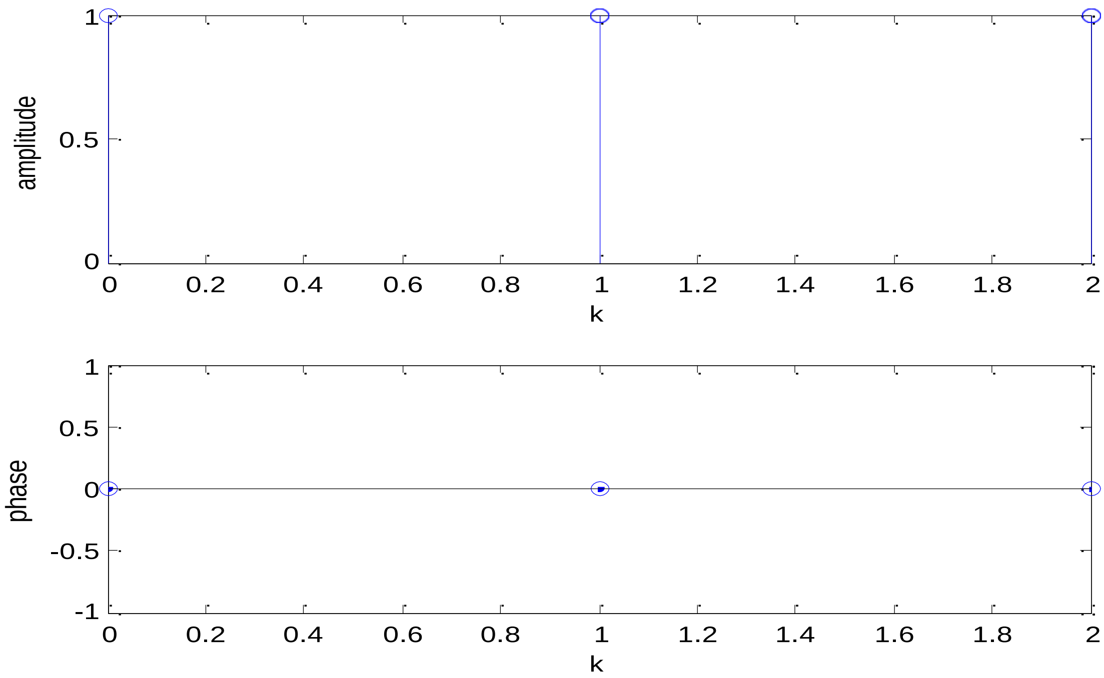


% ***IDFT***

```
clear all
x1=[3.0000 -0.0000 - 0.0000i  0.0000 - 0.0000i]
N=length(x1)
for k=0:1:N-1
   for n=0:1:N-1
p=exp(-i*2*pi*n*k/N)
x2(k+1,n+1)=p
   end
end
Y=(x1*x2)
k=0:1:N-1
subplot(2,1,1)
stem(k,abs(Y))
xlabel('k')
ylabel('amplitude')
subplot(2,1,2)
stem(k,angle(Y))
```

```
xlabel('k')
ylabel('phase')
```

Output: z = 3 3 3
Y = 1 1 1



RESULT: The DFT of given sequence is obtained. Hence the theory and practical value are proved.

Outcomes:

After finishing this experiment the students are able to :

1. Calculate DFT /IDFT of any given signal.
2. Plot DFT / IDFT of given DT signal.

Experiment 4

TO FIND FREQUENCY RESPONSE OF A GIVEN SYSTEM
GIVEN IN (TRANSFER FUNCTION/ DIFFERENTIAL EQUATION FORM)

AIM: To obtain the impulse response/step response of a system described by the given difference equation

Requirements : Computer with MATLAB software

THEORY:

The Difference equation is given as

y[n]-0.25y[n-1]+0.45y[n-2]=1.55x[n]+1.95x[n-1]+ 2.15x[n]

The frequency response is a representation of the system's response to sinusoidal inputs at varying frequencies. The output of a linear system to a sinusoidal input is a sinusoid of the same frequency but with a different magnitude and phase. Any linear system can be *completely* described by how it changes the amplitude and phase of cosine waves passing through it. This information is called the system's frequency response. Since both the impulse response and the frequency response contain complete information about the system, there must be a one-to-one correspondence between the two. Given one, you can calculate the other. The relationship between the impulse response and the frequency response is one of the foundations of signal processing: A system's frequency response is the Fourier Transform of its impulse response Since h [ ] is the common symbol for the impulse response, H [ ] is used for the frequency response.

PROGRAM:

```
clc;
clear all;
close all;
% Difference equation of a second order system
% y[n]-0.25y[n-1]+0.45y[n-2]=1.55x[n]+1.95x[n-1]+ 2.15x[n-2]
b=input('enter the coefficients of x(n),x(n-1)-----');
a=input('enter the coefficients of y(n),y(n-1)----');
N=input('enter the number of samples of frequency response ');
```

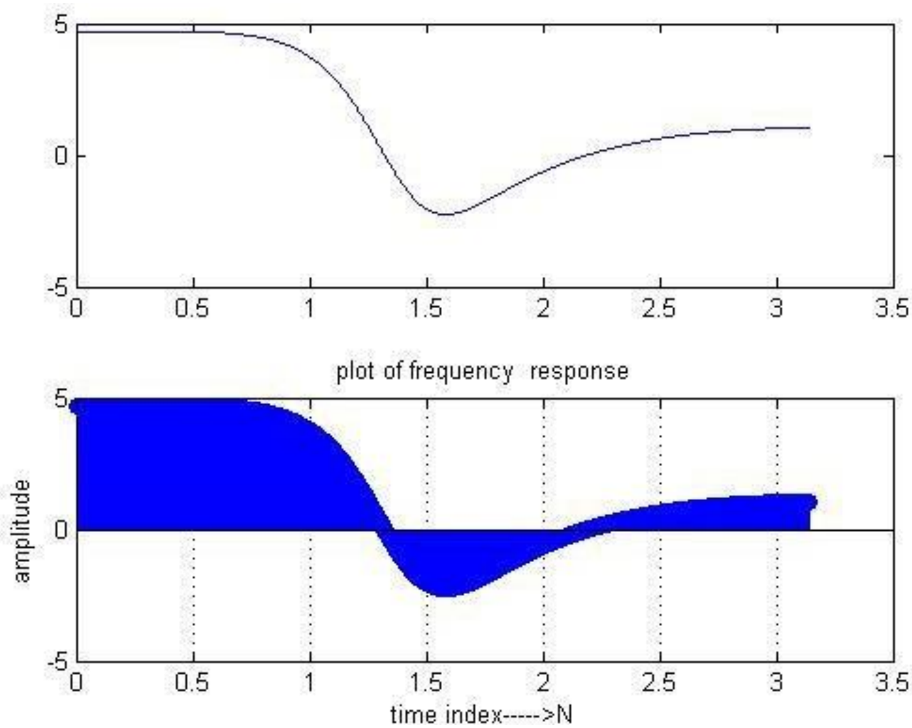
```
[h,t]=freqz(b,a,N);
subplot(2,1,1);
% figure(1);
plot(t,h);
subplot(2,1,2);
% figure(2);
stem(t,h);
title('plot of frequency response');
ylabel('amplitude');
xlabel('time index---- >N');
disp(h);
grid on;
```

OUTPUT:

enter the coefficients of x(n),x(n-1)-----[1.55 1.95 2.15]

enter the coefficients of y(n),y(n-1)----[1 -.25 .45]

enter the number of samples of frequency response 1500



RESULT: The frequency response of given Differential equation is obtained. Hence the theory and practical value are proved.

Experiment 5

OBTAIN FOURIER SERIES COEFFICIENTS BY FORMULA AND USING FET AND COMPARE FOR HALF SINE WAVE.

AIM: To obtain the impulse response/step response of a system described by the given difference equation

Requirements : Computer with MATLAB software

THEORY:

The Fourier series is a sum of sine and cosine functions that describes a periodic signal. It is represented in either the trigonometric form or the exponential form. The toolbox provides this trigonometric Fourier series form

$$y=a_0+\sum_{i=1}^{n} a_i\cos(iwx)+b_i\sin(iwx)$$

where a_0 models a constant (intercept) term in the data and is associated with the $i = 0$ cosine term, w is the fundamental frequency of the signal, n is the number of terms (harmonics) in the series, and $1 \le n \le 8$.

we have first to construct the original signal "Square(t)" so as to compare it with Fourier approximation :

Program:

```
clc;
clear all;
close all;
Fs=60;
 t=0:1/Fs:20-1/Fs;
 y=square(t,50);
 y(y>0)=2;
 y(y<0)=-12;
 figure, plot(t,y);
 axis ([0 20 -20 10])
 % Fourier Series
 a0=0;
 Fy=zeros(size(t));
 N=10;
 for n=1:2:N
```

```
    Fy=Fy+(4/n*pi)*sin(2*pi*n*t/(2*pi));
end
hold on,
plot(t,Fy,'r')
legend(' Square ','Fourier Approx');
```

Experiment 6

TO FIND THE FFT OF A GIVEN SEQUENCE

AIM: To find the FFT of a given sequence

Requirements: Computer with MATLAB software

Theory:

DFT of a sequence

$$X\left[k\right]=\sum_{n=0}^{N-1} x[n]e^{-j\left(2\pi / N\right)kn}$$

Where N= Length of sequence.

K= Frequency Coefficient.

n = Samples in time domain.

FFT: -Fast Fourier transform.

There are two methods.

1. Decimation in time (DIT FFT).

2. Decimation in Frequency (DIF FFT).

PROGRAM:

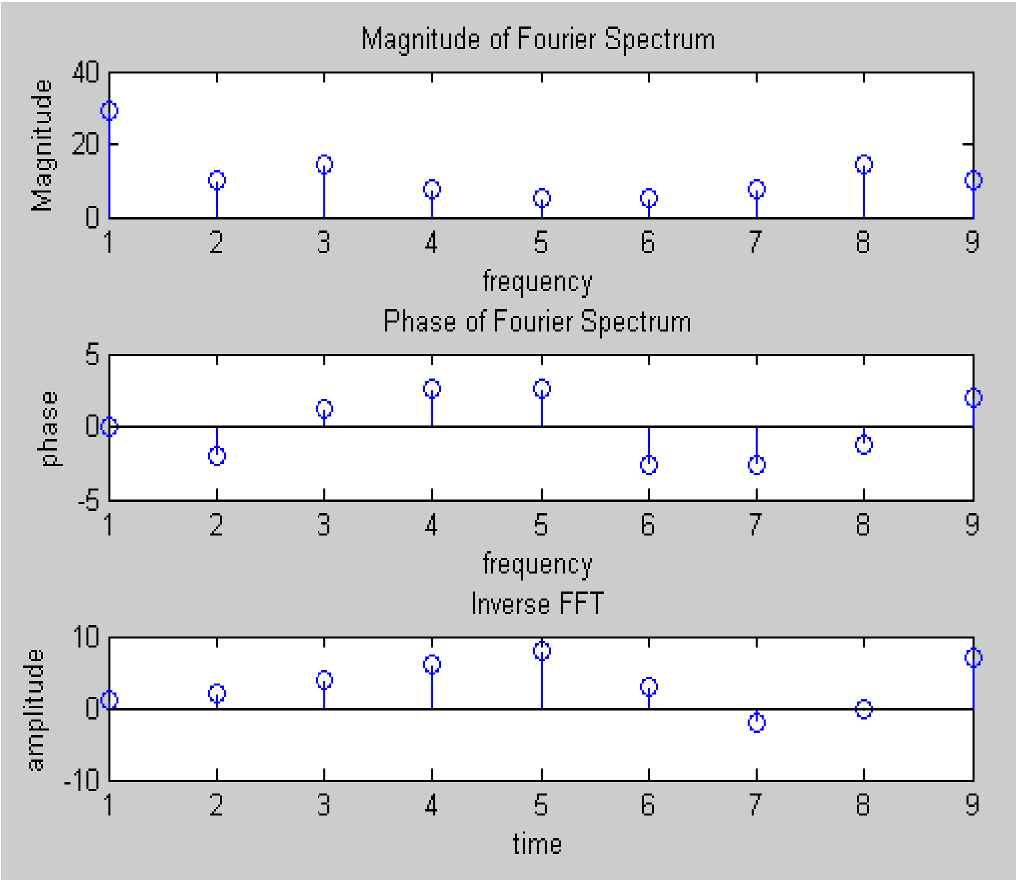
```
%Finding the Fourier Transform of a given signal and plotting its
%magnitude and phase spectrum.
clc;
clear all;
close all;
x=input('Enter the sequence : ')
N=length(x)
```

```
xK=fft(x,N)
xn=ifft(xK)
subplot(3,1,1)
stem(abs(xK))
title('Magnitude of Fourier Spectrum')
xlabel('frequency')
ylabel('Magnitude')
subplot(3,1,2)
stem(angle(xK))
title('Phase of Fourier Spectrum')
xlabel('frequency')
ylabel('phase')
subplot(3,1,3)
stem(xn)
title('Inverse FFT')
xlabel('time');
ylabel('amplitude');
```

Output:

```
Enter the sequence : [1 2 4 6 8 3 -2 0 7]

x =    1    2    4    6    8    3   -2    0    7

N =    9

xK =

  Columns 1 through 6

  29.0000             -3.7476 - 9.3636i   5.2306 +13.6981i  -7.0000 + 3.4641i  -4.4829 + 2.2771i  -
4.4829 - 2.2771i

  Columns 7 through 9

  -7.0000 - 3.4641i  5.2306 -13.6981i  -3.7476 + 9.3636i

xn =    1.0000    2.0000    4.0000    6.0000    8.0000    3.0000   -2.0000        0   7.0000
```

Plotting Magnitude and Phase spectrum



RESULT: The Fast Fourier Transform of given sequence is obtained. Hence the theory and practical value are proved.

Experiment 7

DETERMINATION OF POWER SPECTRUM OF A GIVEN SIGNAL

AIM: Determination of Power Spectrum of a given signals

Requirements: Computer with MATLAB software

THEORY:

The value of the auto-correlation function at zero-time equals the total power in the signal. To compute PSD we compute the auto-correlation of the signal and then take its FFT. The auto-correlation function and PSD are a Fourier transform pair.

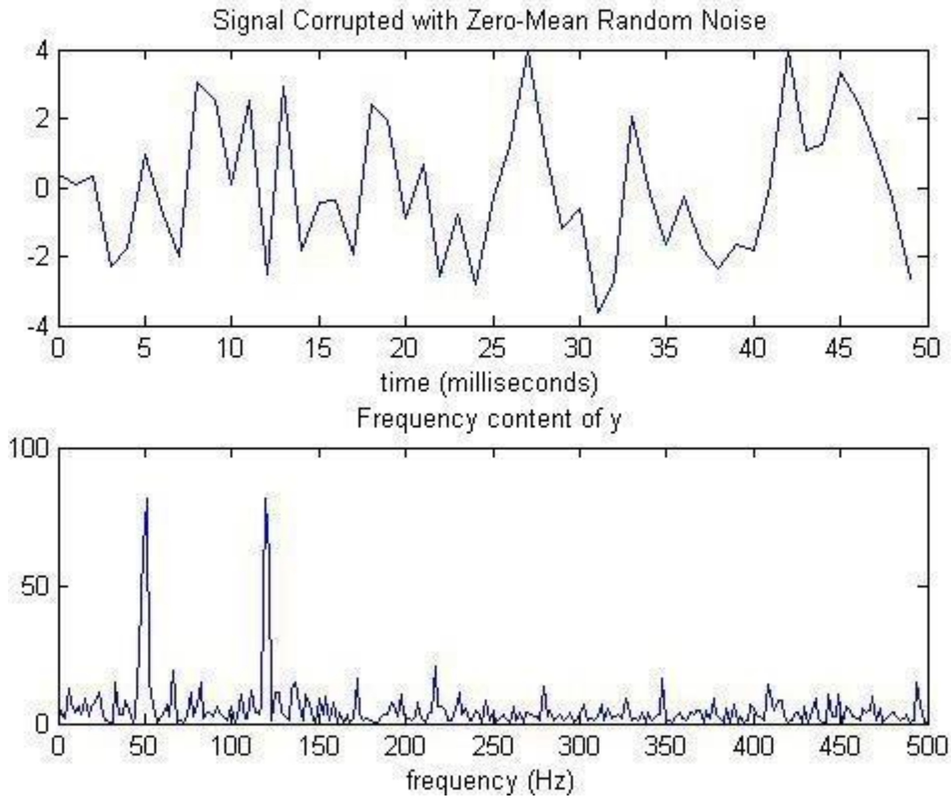
The power spectral density (P.S.D) is a measurement of the energy at various frequencies. In the previous section we saw how to unwrap the FFT and get back the sine and cosine coefficients. Usually we only care how much information is contained at a particular frequency and we don't really care whether it is part of the sine or cosine series. Therefore, we are interested in the absolute value of the FFT coefficients. The absolute value will provide you with the total amount of information contained at a given frequency, the square of the absolute value are considered the power of the signal. Remember that the absolute value of the Fourier coefficients is the distance of the complex number from the origin.

PROGRAM:

```
%Power spectral density
t = 0:0.001:0.6;
x = sin(2*pi*50*t)+sin(2*pi*120*t);
y = x + 2*randn(size(t));
subplot(2,1,1);
% figure(1);
plot(1000*t(1:50),y(1:50))
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)');
Y = fft(y,512);
```

```
%The power spectral density, a measurement of the energy at various frequencies, is:
Pyy = Y.* conj(Y) / 512;
f = 1000*(0:256)/512;
subplot(2,1,2);
% figure(2);
plot(f,Pyy(1:257))
title('Frequency content of y');
xlabel('frequency (Hz)');
```

OUTPUT:



RESULT: The power spectral density of given sequence is obtained. Hence the theory and practical value are proved.

Experiment 8

IMPLEMENTATION OF LP FIR FILTER FOR A GIVEN SEQUENCE

AIM: To implement LP FIR filter for a given sequence using windowing techniques.

Requirements: Computer with MATLAB software

THEORY:

Digital filters refers to the hard ware and software implementation of the mathematical algorithm which accepts a digital signal as input and produces another digital signal as output whose wave shape, amplitude and phase response has been modified in a specified manner. Digital filter play very important role in DSP. Compare with analog filters they are preferred in number of application due to following advantages.

- Truly linear phase response
- Better frequency response
- Filtered and unfiltered data remains saved for further use.

There are two type of digital filters.

1. FIR (finite impulse response) filter

2. IIR (infinite impulse response) filter

Window Name	Transition Width $\Delta\omega$ Approximate	Exact values	Min. Stop band Attenuation	Matlab Command
Rectangular	$\frac{4\Pi}{M}$	$\frac{1.8\Pi}{M}$	**21db**	**B = FIR1(N,Wn,boxcar)**
Bartlett	$\frac{8\Pi}{M}$	$\frac{6.1\Pi}{M}$	**25db**	**B = FIR1(N,Wn,bartlett)**
Hanning	$\frac{8\Pi}{M}$	$\frac{6.2\Pi}{M}$	**44db**	**B = FIR1(N,Wn,hanning)**
Hamming	$\frac{8\Pi}{M}$	$\frac{6.6\Pi}{M}$	**53db**	**B= FIR1(N,W_n,hamming)**

Blackman	$\frac{12\Pi}{M}$	$\frac{11\Pi}{M}$	**74db**	**B = FIR1(N,Wn,blackman)**

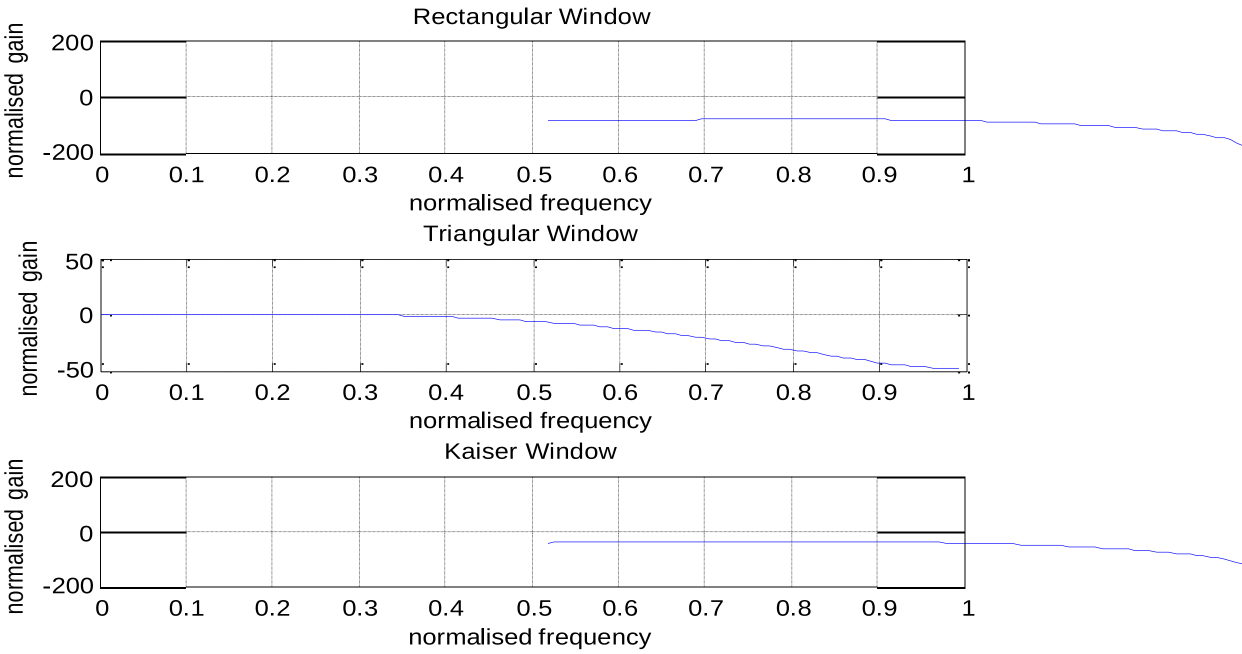
PROGRAM:

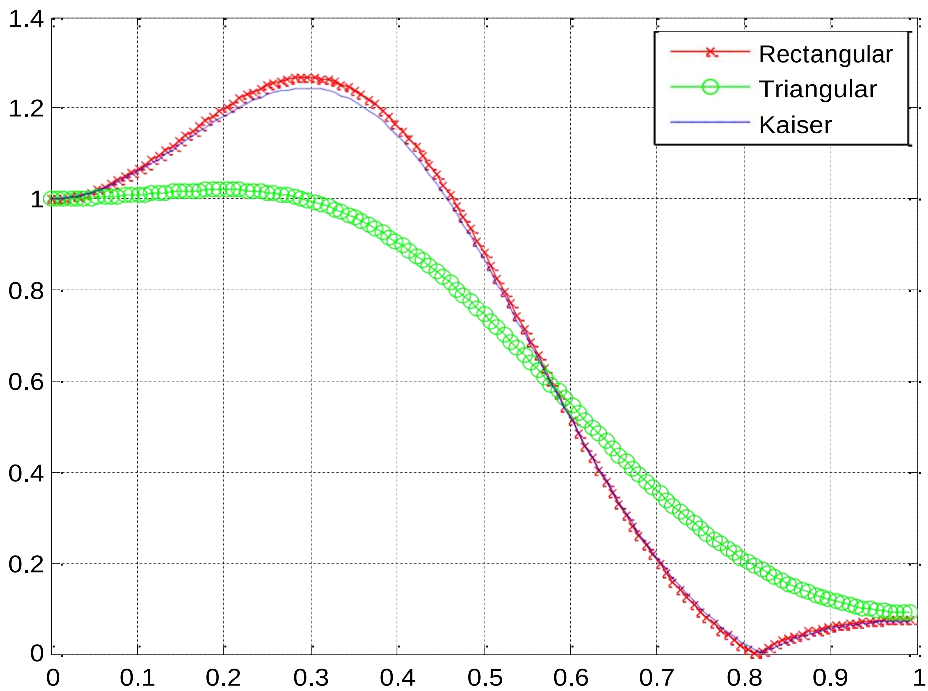
```
clc;
clear all;
n=input('enter window length');
wn=input('enter cutoff frequency');

window1=rectwin(n+1);
b1=fir1(n,wn,window1);
[h1,w1]=freqz(b1,1,128);
subplot(3,1,1);
plot(w1/pi,20*log(abs(h1)));
title('Rectangular Window');
xlabel('normalised frequency');
ylabel('normalised gain');
grid;

window2=triang(n+1);
b2=fir1(n,wn,window2);
[h2,w2]=freqz(b2,1,128);
subplot(3,1,2);
plot(w2/pi,20*log(abs(h2)));
title('Triangular Window');
xlabel('normalised frequency');
ylabel('normalised gain');
grid;

window3=kaiser(n+1);
b3=fir1(n,wn,window3);
[h3,w3]=freqz(b3,1,128);
subplot(3,1,3);
plot(w3/pi,20*log(abs(h3)));
title('Kaiser Window');
xlabel('normalised frequency');
ylabel('normalised gain');
grid;

figure
plot(w1/pi,abs(h1),'-rx',w2/pi,abs(h2),'-go',w3/pi,abs(h3),'-.b');
hleg1 = legend('Rectangular','Triangular','Kaiser');
```

grid;

Result: enter window length 6

enter cutoff frequency 0.6





RESULT: The FIR low pass filter for given values is obtained. Hence the ideal and practical response of FIR filter is proved.

Experiment 9

IMPLEMENTATION OF HP IIR FILTER FOR A GIVEN SEQUENCE

AIM: To implement HP IIR filter for a given sequence using windowing techniques.

Requirements: Computer with MATLAB software

THEORY:

An Infinite impulse response (IIR) filter possesses an output response to an impulse which is of an infinite duration. The impulse response is "infinite" since there is feedback in the filter that is if you put in an impulse, then its output must produced for infinite duration of time. The IIR filter can realize both the poles and zeroes of a system because it has a rational transfer function, described by polynomials in z in both the numerator and the denominator:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=1}^{N} a_k Z^{-k}} \qquad (1)$$

The difference equation for such a system is described by the following:

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) + \sum_{k=1}^{N} a_k y(n-k) \qquad (2)$$

M and N are order of the two polynomials b_k and a_k are the filter coefficients. These filter coefficients are generated using FDS (Filter Design software or Digital Filter design package).

PROGRAM:

```
clc;
clear all;
wp=input('enter the passband frequency');
ws=input('enter the stopband frequency');
rp=input('enter the passband attenuation');
rs=input('enter the stopband attenuation');
fs=input('enter the sampling frequency');
w1=2*wp/fs;
w2=2*ws/fs;
if(wp<ws)
   error('for HPF ws should be less than wp');
```

```
else
   [n,wn]=buttord(w1,w2,rp,rs,'s');
   [b,a]=butter(n,wn,'high','s');
   w=0:0.01:pi;
   [b1,a1]=bilinear(b,a,2);
   [h,omega]=freqz(b1,a1,w);
   gain=20*log(abs(h));
   an=angle(h);
   subplot(2,1,1);
   plot(omega/pi,gain);
   grid;
   title('magnitude plot');
   xlabel('normalized frequency');
   ylabel('gain in dB');
   subplot(2,1,2);
   plot(omega/pi,an);
   grid;
   title('phase plot');
   xlabel('normalized frequency');
   ylabel('phase angle');
end;
```

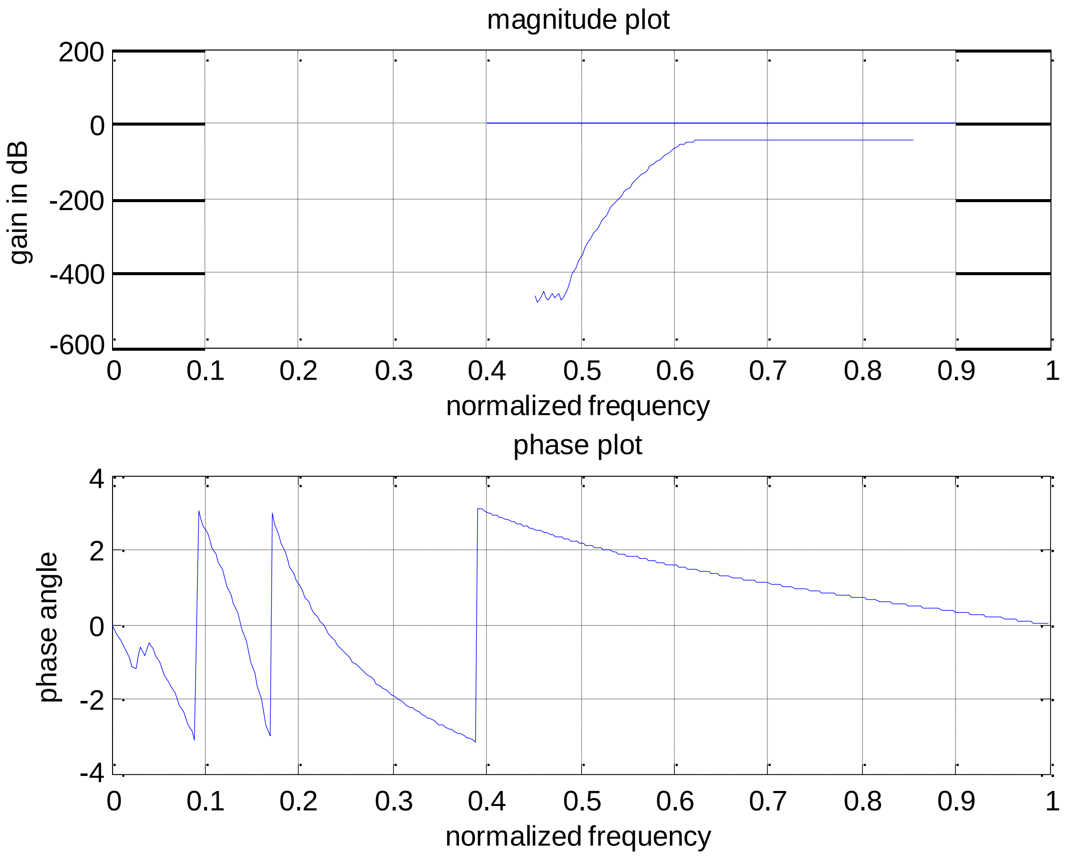
Result: enter the passband frequency 300

enter the stopband frequency 200

enter the passband attenuation .2

enter the stopband attenuation 30

enter the sampling frequency 500



RESULT: The IIR high pass filter for given values is obtained. Hence the ideal and practical response of IIR filter is proved.

Experiment 10

GENERATION OF NARROWBAND SIGNAL THROUGH FILTERING

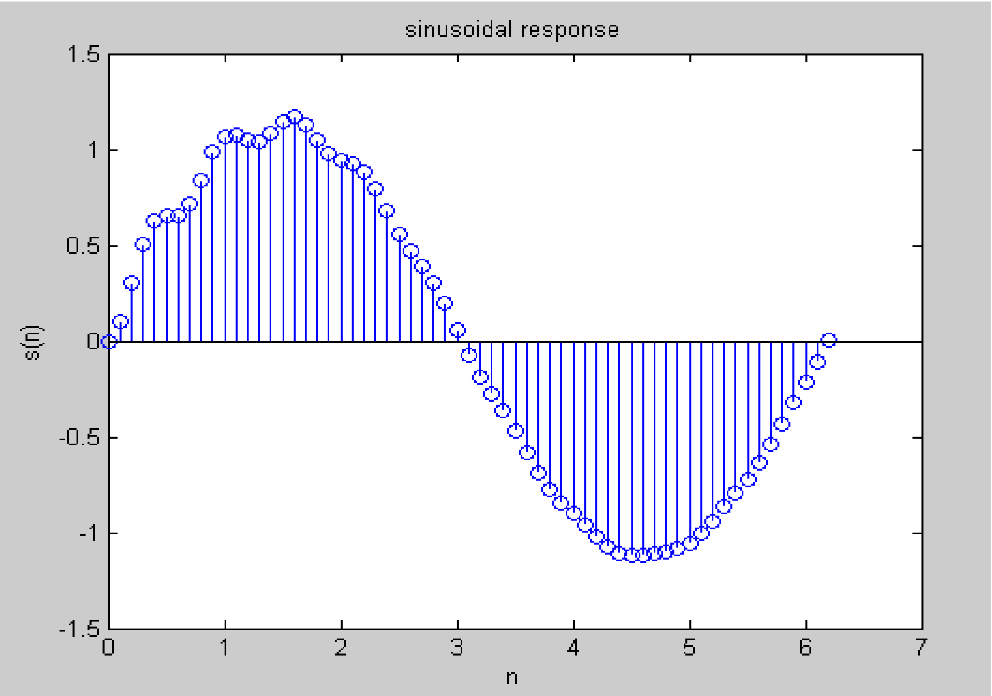
AIM: To generate a sinusoidal signal through filtering.

Requirements: Computer with MATLAB software

PROGRAM:

```
clc;
close all;
clear all;
b=[1];
a=[1,-1,0.9];
n=[-20:120];
t=0:0.1:2*pi;
x=sin(t);
s=filter(b,a,x);
stem(t,s);
title('sinusoidal response');
xlabel('n');
ylabel('s(n)');
```

OUTPUT:



Experiment 11

GENERATION OF DTMF SIGNALS

AIM: To generate DTMF signal using MATLAB.

Requirements: Computer with MATLAB software

THEORY:

The DTMF stands for "*Dual Tone Multi Frequency*", and is a method of representing digits with tone frequencies, in order to transmit them over an analog communications network, for example a telephone line. In telephone networks, DTMF signals are used to encode dial trains and other information. Dual-tone Multi-Frequency (DTMF) signaling is the basis for voice communications control and is widely used worldwide in modern telephony to dial numbers and configure switchboards. It is also used in systems such as in voice mail, electronic mail and telephone banking.

A DTMF signal consists of the sum of two sinusoids - or tones - with frequencies taken from two mutually exclusive groups. These frequencies were chosen to prevent any harmonics from being incorrectly detected by the receiver as some other DTMF frequency. Each pair of tones contains one frequency of the low group (697 Hz, 770 Hz, 852 Hz, 941 Hz) and one frequency of the high group (1209 Hz, 1336 Hz, 1477Hz) and represents a unique symbol.

Frequenc y	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

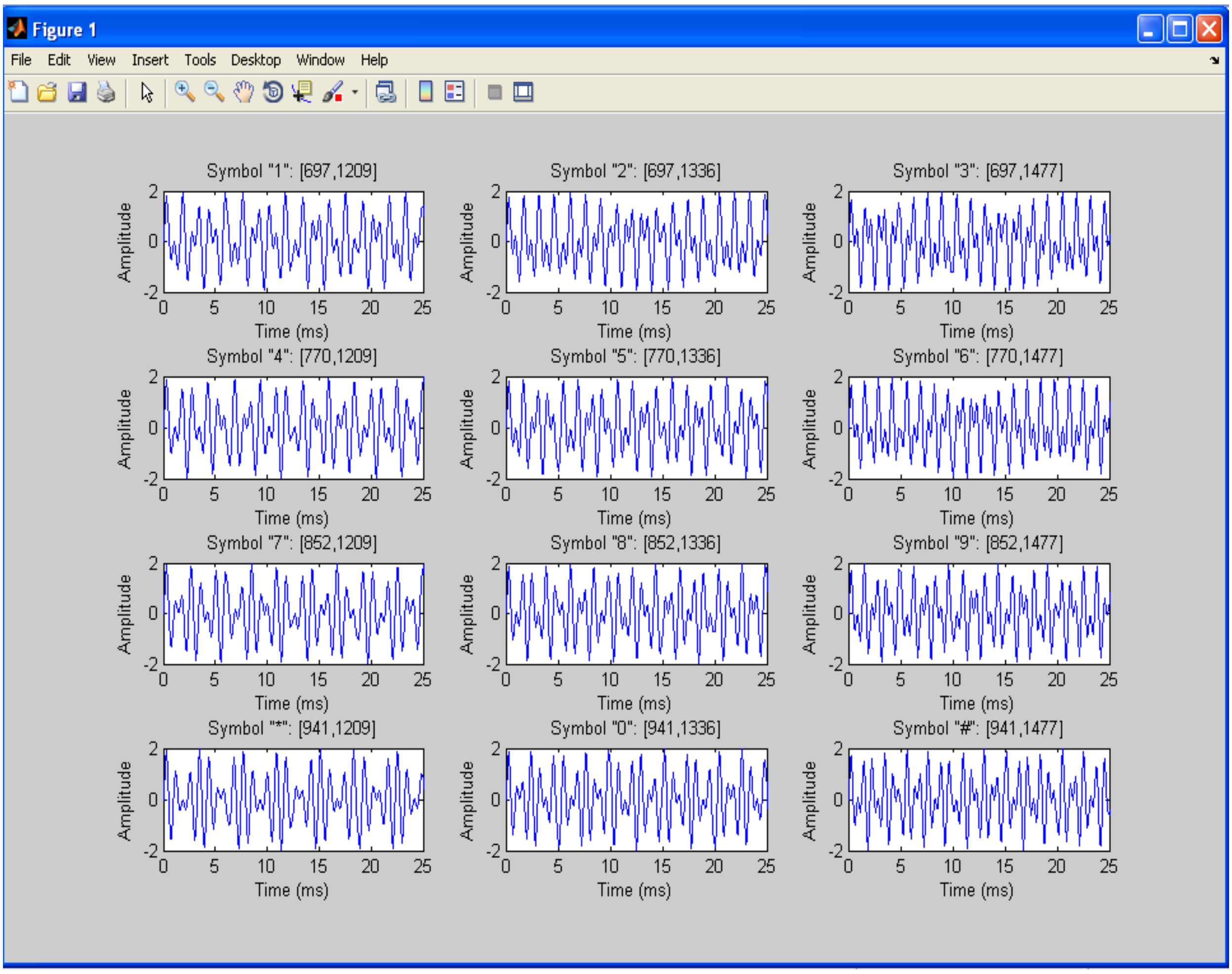
Program:

```
clc
close all
clear all
```

```
 %generate the twelve frequency pairs
symbol = {'1','2','3','4','5','6','7','8','9','*','0','#'};
lfg = [697 770 852 941]; % Low frequency group
hfg = [1209 1336 1477];  % High frequency group
f  = [];
for c=1:4,
   for r=1:3,
      f = [ f [lfg(c);hfg(r)] ];
   end
end
f';
 %generate and visualize the DTMF tones
Fs  = 8000;      % Sampling frequency 8 kHz
N = 800;         % Tones of 100 ms
t  = (0:N-1)/Fs; % 800 samples at Fs
tones = zeros(N,size(f,2));
for toneChoice=1:12,
   % Generate tone
   tones(:,toneChoice) = sum(sin(f(:,toneChoice)*2*pi*t))';
   % Plot tone
   subplot(4,3,toneChoice),plot(t*1e3,tones(:,toneChoice));
   title(['Symbol "', symbol{toneChoice},'":
[',num2str(f(1,toneChoice)),',',num2str(f(2,toneChoice)),']'])
   set(gca, 'Xlim', [0 25]);
   ylabel('Amplitude');
   xlabel('Time (ms)');
end
```

Generation of DTMF Signals:

Output:



RESULT: Different DTMF signal are generated using MATLAB.

Experiment 12

IMPLEMENTATION OF DECIMATION PROCESS

AIM: Program to verify the decimation of given sequence.

Requirements: Computer with MATLAB software

THEORY:

The sampling rate of a discrete time signal x(n) can be reduced by a factor M by taking every M^{th} value of the signal the block diagram representation of the down sampler is shown in figure below. The quadratic symbol in below figure with arrow pointing down words is called a down sampler. The output signal y(n) is a down sampled signal of the input signal x(n) and can be represented by

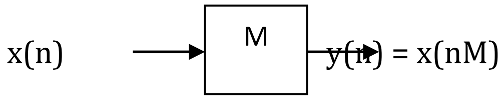


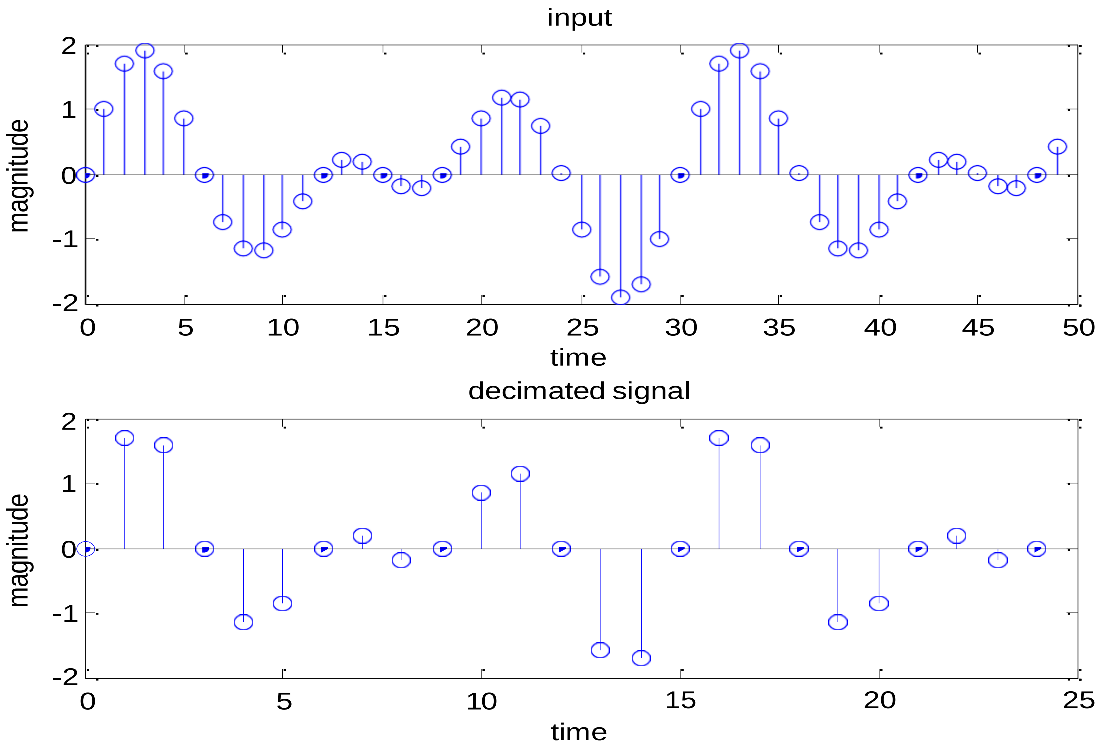
fig: down sampler

Let us assume the signal x(n) as shown in output figure. The down sampled signal x(n) can be obtained by simply keeping every M^{th} sample and removing (M-1) in between samples. This process is equal to reducing the sampling rate by a factor M.

PROGRAM:

```
clc;
clear all;
N=50;
n=0:1:N-1;
x=sin(2*pi*(n/10))+sin(2*pi*(n/15));
D=input('enter decimation factor');
x1=x(1:D:N);
n1=1:1:N/D;
subplot(2,1,1);
stem(n,x);
title('input');
xlabel('time');
ylabel('magnitude');
subplot(2,1,2);
stem(n1-1,x1);
title('decimated signal');
xlabel('time');
ylabel('magnitude');
```

OUTPUT: enter decimation factor 2



RESULT: The decimator for a given sequence is observed for chosen factor **M**. Hence the theory and practical is verified.

Experiment 13

IMPLEMENTATION OF INTERPOLATION PROCESS

AIM: Program to verify the Interpolation of given sequence.

Requirements: Computer with MATLAB software

THEORY:

The process of increasing the sampling rate is called interpolation. Interpolation is upsampling followed by appropriate filtering.

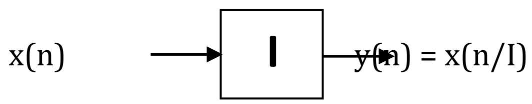


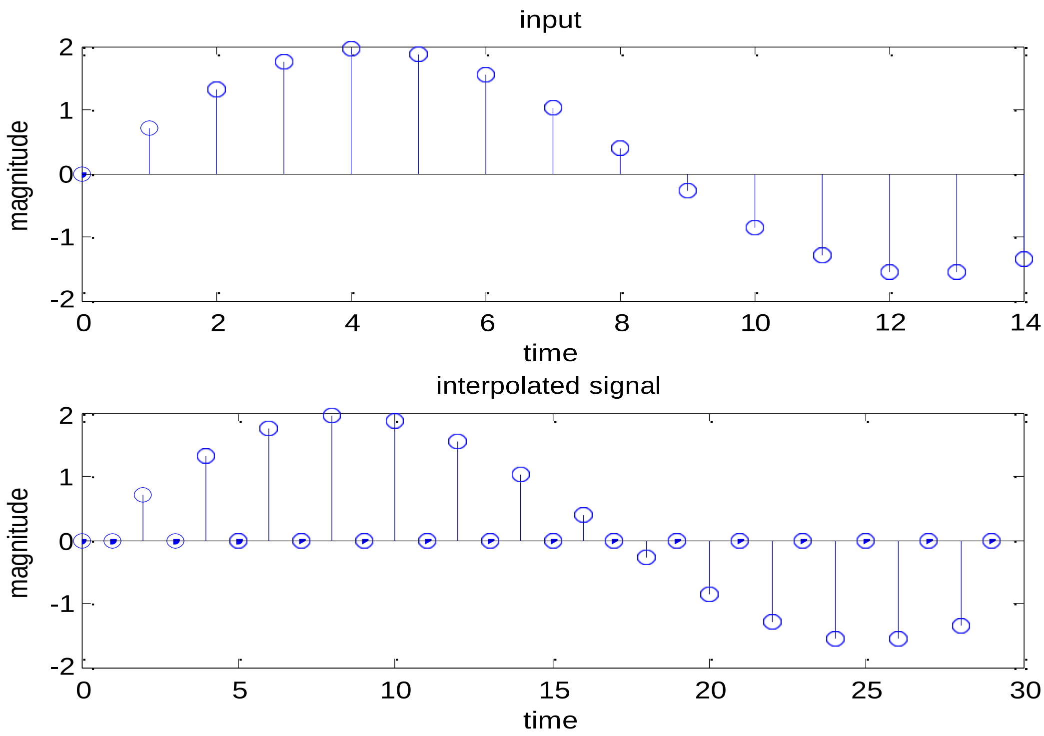
fig: Up sampler

The simplest method to interpolate by a factor of I is to add I-1 zeros in between the samples, multiply the amplitude by I and filter the generated signal, with a so-called anti-imaging low pass filter at the high sampling frequency.

PROGRAM:

```
clc;
clear all;
N=15;
n=0:1:N-1;
x=sin(2*pi*(n/20))+sin(2*pi*(n/15));
I=input('enter interpolation factor');
x1=zeros(1,I*N);
n1=1:1:N*I;
j=1:I:N*I;
x1(j)=x;
subplot(2,1,1);
stem(n,x);
title('input');
xlabel('time');
ylabel('magnitude');
subplot(2,1,2);
stem(n1-1,x1);
title('interpolated signal');
xlabel('time');
ylabel('magnitude');
```

OUTPUT: enter interpolation factor **2**



RESULT: The Interpolator for a given sequence is observed for chosen factor **I.** Hence the theory and practical is verified.

Experiment 14

IMPLEMENTATION OF I/D SAMPLING RATE CONVERTERS

AIM: Program to implement sampling rate conversion.

Requirements: Computer with MATLAB software

THEORY:

The aim of (digital) sample-rate conversion is to bring a digital audio signal from one sample frequency to another. The distortion of the audio signal, introduced by the sample-rate converter, should be as low as possible. The generation of output samples from the input samples may be performed by the application of various methods.

PROGRAM:

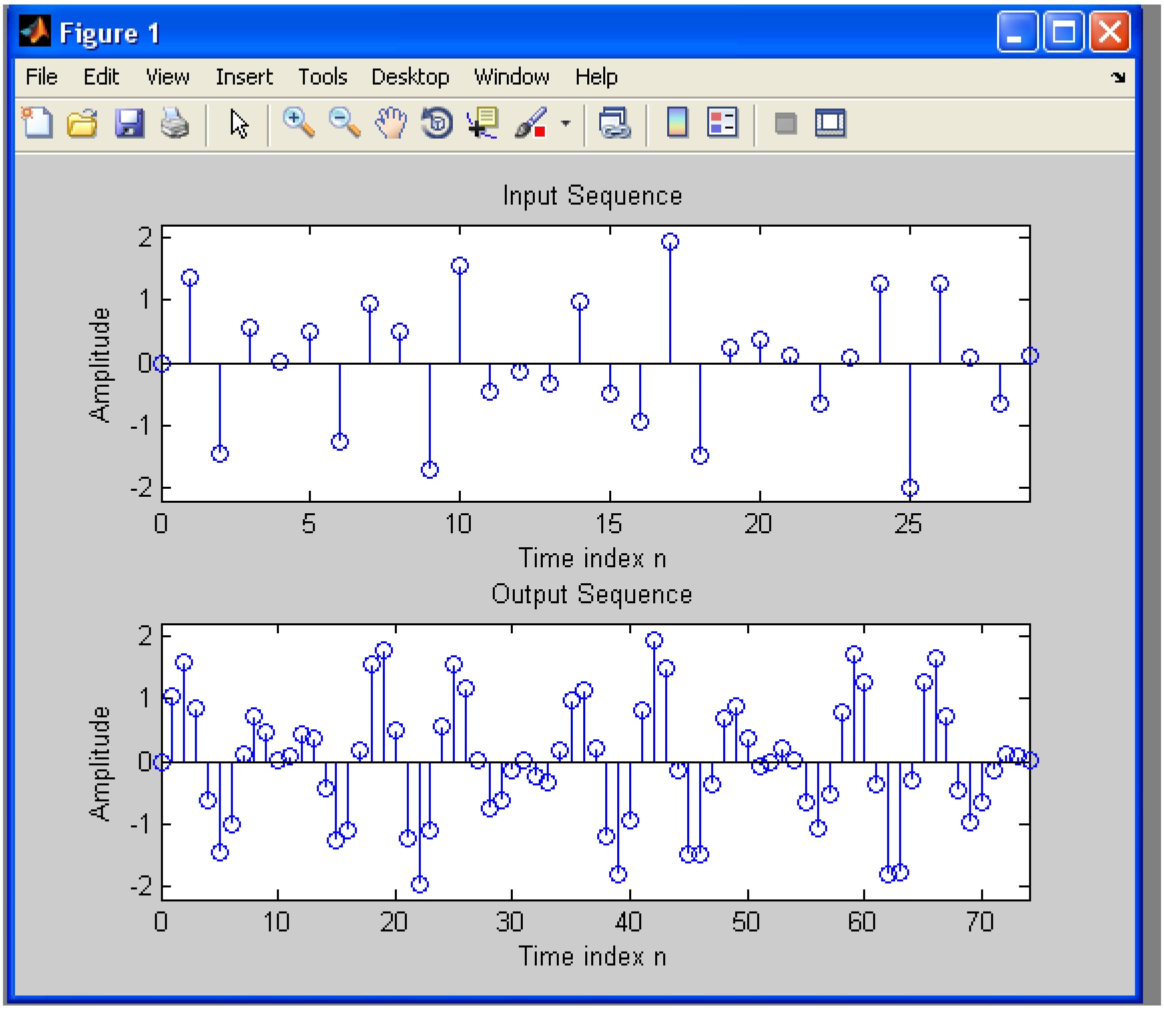
```
% Illustration of Sampling Rate Alteration by a Ratio of Two Integers
clc;
close all;
clear all;
L = input('Up-sampling factor = ');
M = input('Down-sampling factor = ');
n = 0:29;
x = sin(2*pi*0.43*n) + sin(2*pi*0.31*n);
y = resample(x,L,M);
subplot(2,1,1);
stem(n,x(1:30));axis([0 29 -2.2 2.2]);
title('Input Sequence');
xlabel('Time index n'); ylabel('Amplitude');
subplot(2,1,2);
m = 0:(30*L/M)-1;
stem(m,y(1:30*L/M));axis([0 (30*L/M)-1 -2.2 2.2]);
title('Output Sequence');
```

xlabel('Time index n'); ylabel('Amplitude');

OUTPUT:

Up-sampling factor = 5

Down-sampling factor = 2



RESULT: For a given sequence sampling rate conversion is verified.

Experiment 15

IMPULSE RESPONSE OF FIRST ORDER AND SECOND ORDER SYSTEM

AIM: To determine the response of first order and second order systems.

Requirements: Computer with MATLAB software

THEORY:-

LTI Discrete time system is completely specified by its impulse response i.e. knowing the impulse response we can compute the output of the system to any arbitrary input. Let h[n] denotes the impulse response of the LTI discrete time systems. Since discrete time system is time invariant, its response to [n-1] will be h[n-1] .Likewise the response to [n+2] , [n-4] and [n-6] will be h[n+2], h[n-4] and h[n-6] .

PROGRAM:

%This program finds the unit sample response of the first order discrete system, expressed by its difference equation as y(n)= x(n)+2.y(n-1)

```
a=input('enter the coefficient vector of input starting from the coefficient of x(n) term')
b=input('enter the  coefficient  vector of  output starting from the coefficient of y(n) term')
n1=input('enter the lower limit of the range of impulse response')
n2=input('enter the upper limit of the range of impulse response')
n=[n1:n2];
x=zeros(1,length(n));
for i=1:length(n)
   if n(i)==0
      x(i)=1;
   end
end
h=filter(a,b,x);
stem(n,h)
title('Unit sample response of the discrete system y(n)=x(n)+2.y(n-1)');
xlabel('Time ')
ylabel('Unit Sample Response')
axis([-1 7 0 35])
```

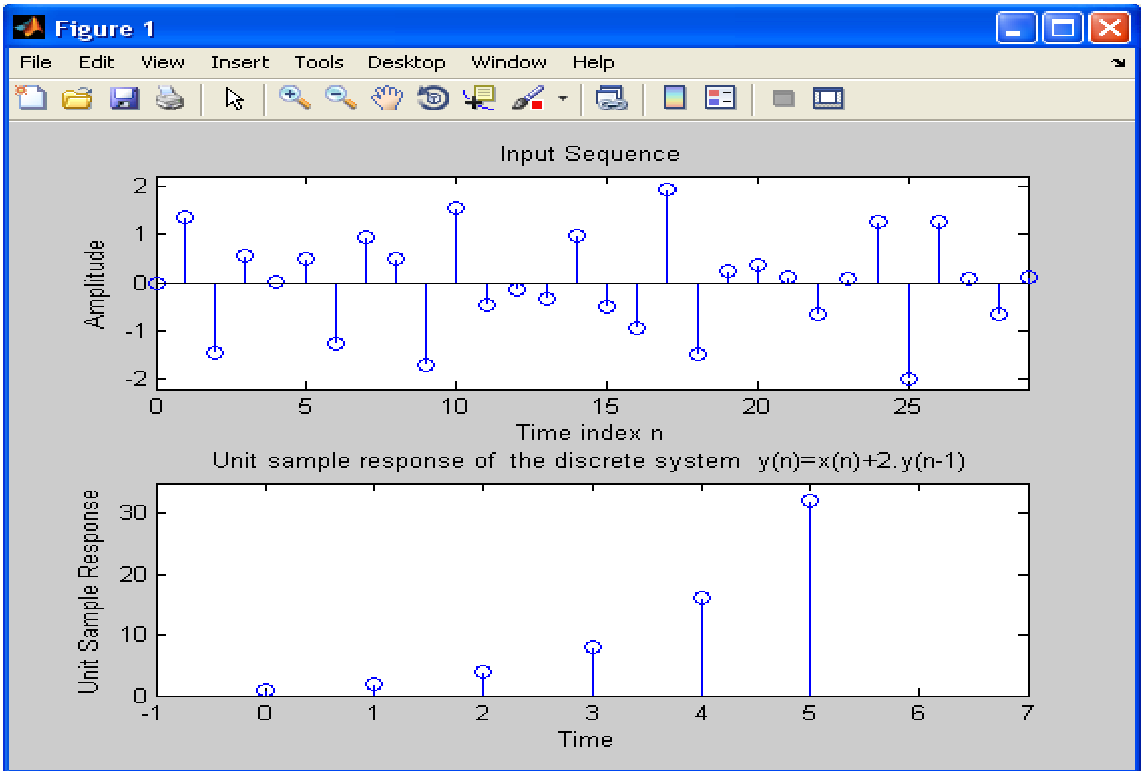
OUTPUT:

enter the coefficient vector of input starting from the coefficient of x(n) term1 : **a =1**

enter the coefficient vector of output starting from the coefficient of y(n) term[1 -2] b =1 -2

enter the lower limit of the range of impulse response0 n1 = 0

enter the upper limit of the range of impulse response5 n2 = 5



% This program find the Unit Sample response of the second order discrete system represented as y(n)-(3/4)y(n-1)+(1/8).y(n-2)=x(n)+(5/3).x(n-1).

```
a=input('enter the coefficient vector of input starting from the coefficient of x(n) term')
b=input('enter the  coefficient  vector of  output starting from the coefficient of y(n) term')
n1=input('enter the lower limit of the range of impulse response')
n2=input('enter the upper limit of the range of impulse response')
n=[n1:n2];
x=zeros(1,length(n));
for i=1:length(n)
  if n(i)==0
    x(i)=1;
```

```
  end
end
h=filter(a,b,x);
stem(n,h)
title('Unit sample response of the discrete system y(n)-(3/4)y(n-1)+(1/8).y(n-2)=x(n)+(5/3).x(n-1)')
xlabel('Time ')
ylabel('Amplitude')
axis([-1 7 0 3])
```

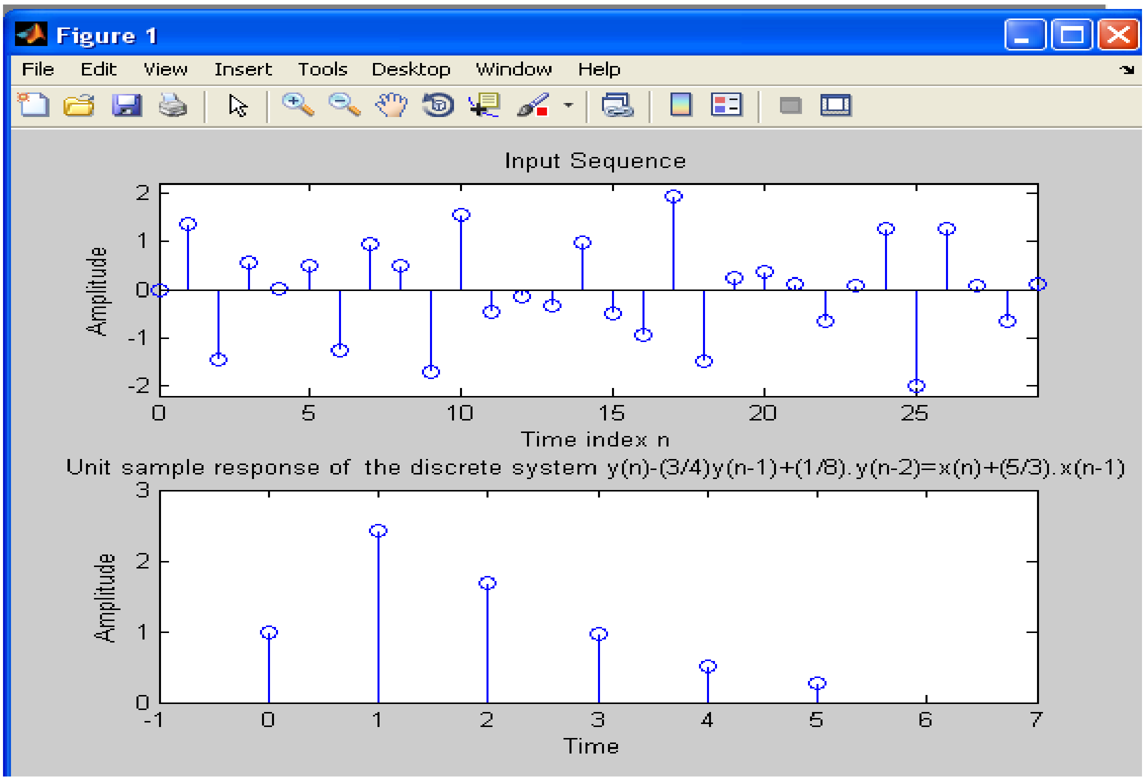
OUTPUT:

enter the coefficient vector of input starting from the coefficient of x(n) term[1.0000 1.6667]

enter the coefficient vector of output starting from the coefficient of y(n) term[1.0000 -0.7500 0.1250]

enter the lower limit of the range of impulse response0, n1 = 0

enter the upper limit of the range of impulse response5, n2 =5



RESULT: The impulse response for the given specifications is plotted using matlab