MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE- Accredited by NBA & NAAC 'A' Grade – ISO 9001:2015 Certified)

Certificate

Department of ***Electronics and Communication Engineering*** *Certified that in the bonafide Record of the work done by Mr./Miss. ______________________________ Reg.No_____________________ of B.Tech* ***ECE****____ year _____ semester for the Academic year 20___ to 20_____ in ____________________________Laboratory.*

Date: *Staff Incharge* *HOD*

Internal Examiner *External Examiner*

INDEX

S.No	Date	Name of the Experiment	Page No	Signature of faculty

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY
Maisammaguda, Dhulapally post, via Hakimpet, Secunderabad

BASIC SIMULATION MANUAL

II B.Tech I-SEM

Prepared by

Mr. V Shivaraj Kumar
ASSISTANT PROFESSOR

Mrs.N Saritha
ASSISTANT PROFESSOR

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

II Year B.Tech. ECE- I Sem **L/T/P/C**
-/-/3/1.5

(R20A0482) BASIC SIMULATION LAB

COURSE OBJECTIVES:
1) To learn basic Operations on Matrices.
2) To simulate generation of basic waveforms and general operations on signals.
3) To Understand the Concept of auto correlation, cross correlation and Convolution of given Signal/ sequence and simulate it accordingly.
4) To learn various transforms like Fourier and Z-transform of various signals.

NOTE:
1) All the experiments are to be simulated using MATLAB or equivalent software
2) Minimum of 10 experiments are to be completed

List of experiments:
1) Basic operations on matrices.
2) Generation on various signals and Sequences (periodic and aperiodic), such as unit impulse, unit step, square, saw tooth, triangular, sinusoidal, ramp, sinc.
3) Operations on signals and sequences such as addition, multiplication, scaling, shifting, folding, computation of energy and average power.
4) Finding the even and odd parts of signal/sequence and real and imaginary part of signal.
5) Convolution between signals and sequences.
6) Auto correlation and cross correlation between signals and sequences.
7) Verification of linearity properties of a given continuous /discrete system.
8) Verification of time invariance properties of a given continuous discrete system.
9) Computation of unit sample, unit step and sinusoidal response of the given LTI system and verifying its physical Realizability and stability properties.
10) Finding the Fourier transform of a given signal and plotting its magnitude and phase spectrum.
11) Locating the zeros and poles and plotting the pole zero maps in s-plane and z-plane for the given transfer function.
12) Sampling theorem verification.

COURSE OUTCOMES
After going through this course the student will be able to
1) Do the various operations on matrices.
2) Perform various operations on the signals including Time shifting, Scaling, Reversal, Amplitude Scaling.
3) Determine the correlation & Convolution between Signals and sequences.
4) Understand the various transforms of signals and sequences.

VISION

To establish a pedestal for the integral innovation, team spirit, originality and competence in the students, expose them to face the global challenges and become pioneers of Indian vision of modern society.

MISSION

* To become a model institution in the fields of Engineering, Technology and Management.
* To impart holistic education to the students to render them as industry ready engineers.
* To ensure synchronization of institute ideologies with challenging demands of International Pioneering Organizations.

QUALITY POLICY

* To implement best practices in Teaching and Learning process for both UG and PG courses meticulously
* To provide state of art infrastructure and expertise to impart quality education.
* To groom the students to become intellectually creative and professionally competitive.
* To channelize the activities and tune them in heights of commitment and sincerity, the requisites to claim the never-ending ladder of SUCCESS year after year.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: PROFESSIONALISM & CITIZENSHIP To create and sustain a community of learning in which students acquire knowledge and learn to apply it professionally with due consideration for ethical, ecological and economic issues.
PEO2: TECHNICAL ACCOMPLISHMENTS To provide knowledge based services to satisfy the needs of society and the industry by providing hands on experience in various technologies in core field.
PEO3: INVENTION, INNOVATION AND CREATIVITY To make the students to design, experiment, analyze, interpret in the core field with the help of other multi-disciplinary concepts wherever applicable.
PEO4: PROFESSIONAL DEVELOPMENT To educate the students to disseminate research findings with good soft skills and become a successful entrepreneur.
PEO5: HUMAN RESOURCE DEVELOPMENT To graduate the students in building national capabilities in technology, education and research

CODE OF CONDUCT FOR THE LABORATORIES

1. All students must observe the Dress Code while in the laboratory.
2. Sandals or open-toed shoes are NOT allowed.
3. Foods, drinks and smoking are NOT allowed.
4. All bags must be left at the indicated place.
5. The lab timetable must be strictly followed.
6. Be punctual for your laboratory session.
7. Program must be executed within the given time.
8. Noise must be kept to a minimum.
9. Workspace must be kept clean and tidy at all time.
10. Handle the systems and interfacing kits with care.
11. All students are liable for any damage to the accessories due to their own negligence.
12. All interfacing kits connecting cables must be RETURNED if you taken from supervisor.
13. Students are strictly PROHIBITED from taking out any items from the laboratory.
14. Students are NOT allowed to work alone in the laboratory without the Lab Supervisor
15. USB Ports have been disabled if you want to use USB drive consult lab supervisor.
16. 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 for updates.

EXPERIMENT NO-1

BASIC OPERATIONS ON MATRICES

AIM: Generate a matrix and perform basic operation on matrices using MATLAB software.

Software Required: MATLAB software

Theory:

MATLAB treats all variables as matrices. Vectors are special forms of matrices and contain only one row or one column. Whereas scalars are special forms of matrices and contain only one row and one column. A matrix with one row is called row vector and a matrix with single column is called column vector.

The first one consists of convenient matrix building functions, some of which are given below.
1. eye - identity matrix
2. zeros - matrix of zeros
3. ones - matrix of ones
4. diag - extract diagonal of a matrix or create diagonal matrices
5. triu - upper triangular part of a matrix
6. tril - lower triangular part of a matrix
7. rand - ran

commands in the second sub-category of matrix functions are
1. size- size of a matrix
2. det -determinant of a square matrix
3. inv- inverse of a matrix
4. rank- rank of a matrix
5. rref- reduced row echelon form
6. eig- eigenvalues and eigenvectors
7. poly- characteristic polynomialdomly generated matrix

Program:

```
% Creating a column vector
>> a=[1;2;3]
a =
1
2
3
% Creating a row vector
>> b=[1 2 3]
b=
1 2 3
% Creating a matrix
>> m=[1 2 3;4 6 9;2 6 9]
m =
1 2 3
```

```
4 6 9
2 6 9
% Extracting sub matrix from matrix
>> sub_m=m(2:3,2:3)
sub_m =
6 9
6 9
% extracting column vector from matrix
>> c=m (:,2)
c =
2
6
6
% extracting row vector from matrix
>> d=m (3,:)
d =
2 6 9
% creation of two matrices a and b
>> a=[2 4 -1;-2 1 9;-1 -1 0]
a =
2 4 -1
-2 1 9
-1 -1 0
>> b=[0 2 3;1 0 2;1 4 6]
b =
0 2 3
1 0 2
1 4 6

% matrix multiplication
>> x1=a*b
x1 =
3 0 8
10 32 50
-1 -2 -5
% element to element multiplication
>> x2=a.*b
x2 =
0 8 -3
-2 0 18
-1 -4 0
% matrix addition
>> x3=a+b
x3 =
2 6 2
-1 1 11
0 3 6
```

```
% matrix subtraction
>> x4=a-b
x4 =
2 2 -4
-3 1 7
-2 -5 -6
% matrix division
>> x5=a/b
x5 =
-9.0000 -3.5000 5.5000
12.0000 3.7500 -5.7500
3.0000 0.7500 -1.7500
% element to element division
>> x6=a./b
Warning: Divide by zero.
x6 =
Inf 2.0000 -0.3333
-2.0000 Inf 4.5000
-1.0000 -0.2500 0
% inverse of matrix a
>> x7=inv(a)
x7 =
-0.4286 -0.0476 -1.7619
0.4286 0.0476 0.7619
-0.1429 0.0952 -0.4762
% transpose of matrix a
>> x8=a'
x8 =
2 -2 -1
4 1 -1
-1 9 0
```

RESULT: Matrix operations are performed using Matlab software.

VIVA QUESTIONS:-

1. Expand MATLAB? And importance of MATLAB?
2. What is clear all and close all will do?
3. What is disp() and input()?
4. What is the syntax to find the Eigen values and eigenvectors of the matrix?
5. Define scalar and vector?

EXPERIMENT NO-2
GENERATION OF SIGNALS AND SEQUENCES

AIM: Generate various signals and sequences (Periodic and aperiodic), such as Unit Impulse, Unit Step, Square, Saw tooth, Triangular, Sinusoidal, Ramp, Sinc.

Software Required: Matlab software

Theory: If the amplitude of the signal is defined at every instant of time then it is called continuous time signal. If the amplitude of the signal is defined at only at some instants of time then it is called discrete time signal. If the signal repeats itself at regular intervals then it is called periodic signal. Otherwise they are called aperiodic signals.

EX: ramp,Impulse,unit step, sinc- Aperiodic signals

square,sawtooth,triangular sinusoidal – periodic signals.

Ramp sinal: The **ramp function** is a unitary real function, easily computable as the mean of the independent variable and its absolute value.This function is applied in engineering. The name *ramp function* is derived from the appearance of its graph.

$$r(t)= \begin{cases} t & \text{when } t \geq 0 \\ 0 & \text{else} \end{cases}$$

Unit impulse signal: One of the more useful functions in the study of linear systems is the "unit impulse function."An ideal impulse function is a function that is zero everywhere but at the origin, where it isinfinitely high. However, the *area* of the impulse is finite

Y(t)= 1 when t=0

=0 other wise

Unit step signal: The unit step function and the impulse function are considered to be fundamental functions in engineering, and it is strongly recommended that the reader becomes very familiar with both of these functions.

$$u(t)= \begin{cases} 0 & \text{if } t < 0 \\ 1 & \text{if } t > 0 \\ \frac{1}{2} & \text{If } t=0 \end{cases}$$

Sinc signal: There is a particular form that appears so frequently in communications engineering, that we give it its own name. This function is called the "Sinc function”.

The Sinc function is defined in the following manner:

$$sinc(x) = \frac{\sin \pi x}{\pi x} \quad \text{if } x \neq 0 \text{ and } sinc(0) = 1$$

The value of sinc(x) is defined as 1 at x = 0, since

$$\lim_{x \to 0} sinc(x) = 1$$

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

PROGRAM:

```
% Generation of signals and sequences
clc;
clear all;
close all;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%generation of unit impulse signal
t1=-1:0.01:1
y1=(t1==0);
subplot(2,2,1);
plot(t1,y1);
xlabel('time');
ylabel('amplitude');
title('unit impulse signal');
%generation of impulse sequence
subplot(2,2,2);
stem(t1,y1);
xlabel('n');
ylabel('amplitude');
title('unit impulse sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of unit step signal
t2=-10:1:10;
y2=(t2>=0);
subplot(2,2,3);
plot(t2,y2);
xlabel('time');
ylabel('amplitude');
title('unit step signal');
%generation of unit step sequence
subplot(2,2,4);
stem(t2,y2);
xlabel('n');
ylabel('amplitude');
title('unit step sequence');
```

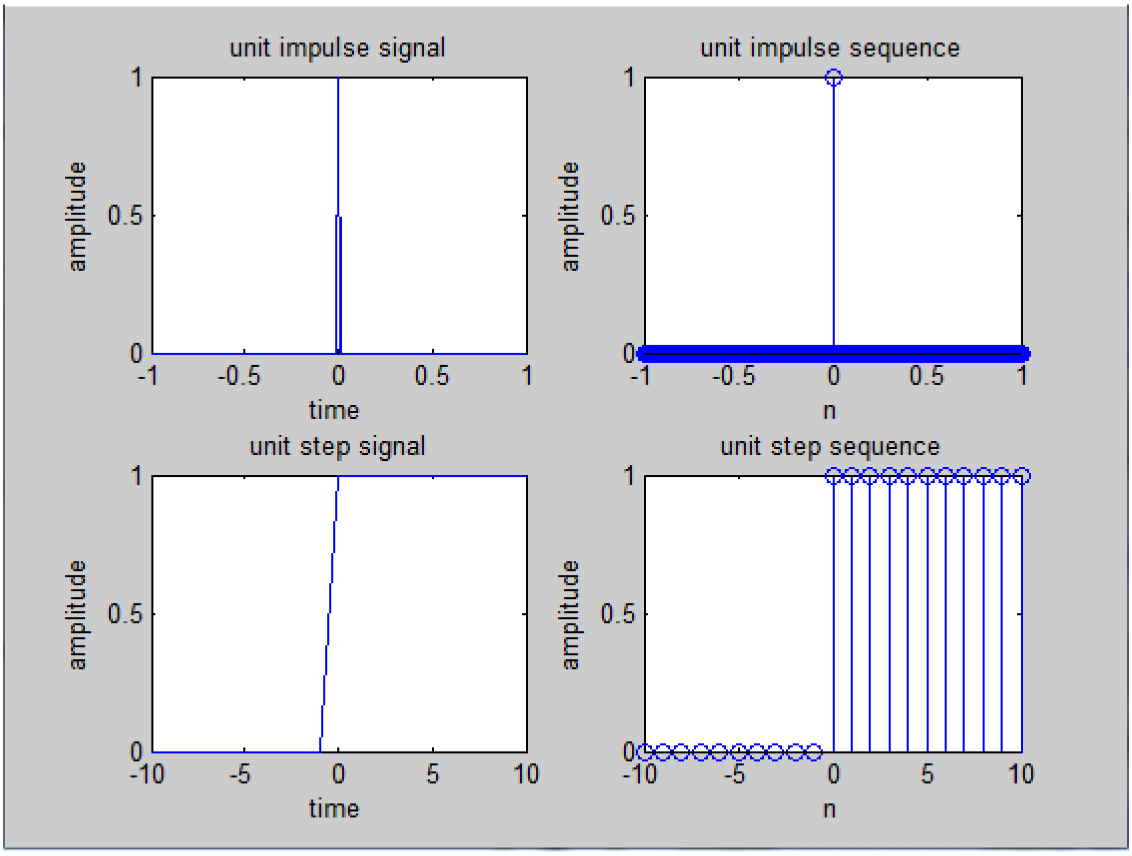
```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of square wave signal
t=0:0.002:0.1;
y3=square(2*pi*50*t);
figure;
subplot(2,2,1);
plot(t,y3);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title('square wave signal');
%generation of square wave sequence
subplot(2,2,2);
stem(t,y3);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('square wave sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of sawtooth signal
y4=sawtooth(2*pi*50*t);
subplot(2,2,3);
plot(t,y4);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title('sawtooth wave signal');
%generation of sawtooth sequence
subplot(2,2,4);
stem(t,y4);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('sawtooth wave sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of triangular wave signal
y5=sawtooth(2*pi*50*t,.5);
figure;
subplot(2,2,1);
plot(t,y5);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title(' triangular wave signal');
```

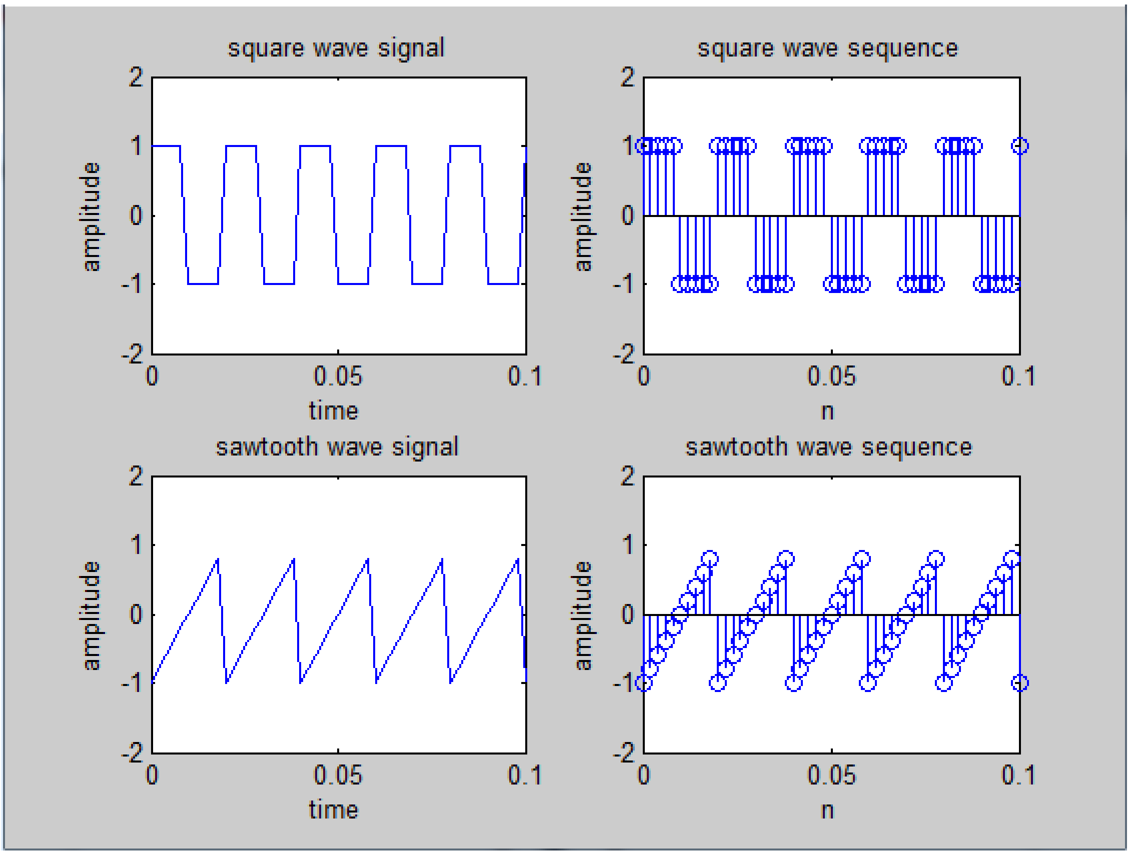
```
%generation of triangular wave sequence
subplot(2,2,2);
stem(t,y5);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('triangular wave sequence');
%generation of sinsoidal wave signal
y6=sin(2*pi*40*t);
subplot(2,2,3);
plot(t,y6);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title(' sinsoidal wave signal');
%generation of sin wave sequence
subplot(2,2,4);
stem(t,y6);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('sin wave sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%generation of ramp signal
y7=t;
figure;
subplot(2,2,1);
plot(t,y7);
xlabel('time');
ylabel('amplitude');
title('ramp signal');
%generation of ramp sequence
subplot(2,2,2);
stem(t,y7);
xlabel('n');
ylabel('amplitude');
title('ramp sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%generation of sinc signal
t3=linspace(-5,5);
y8=sinc(t3);
subplot(2,2,3);
plot(t3,y8);
xlabel('time');
ylabel('amplitude');
title(' sinc signal');
%generation of sinc  sequence
```

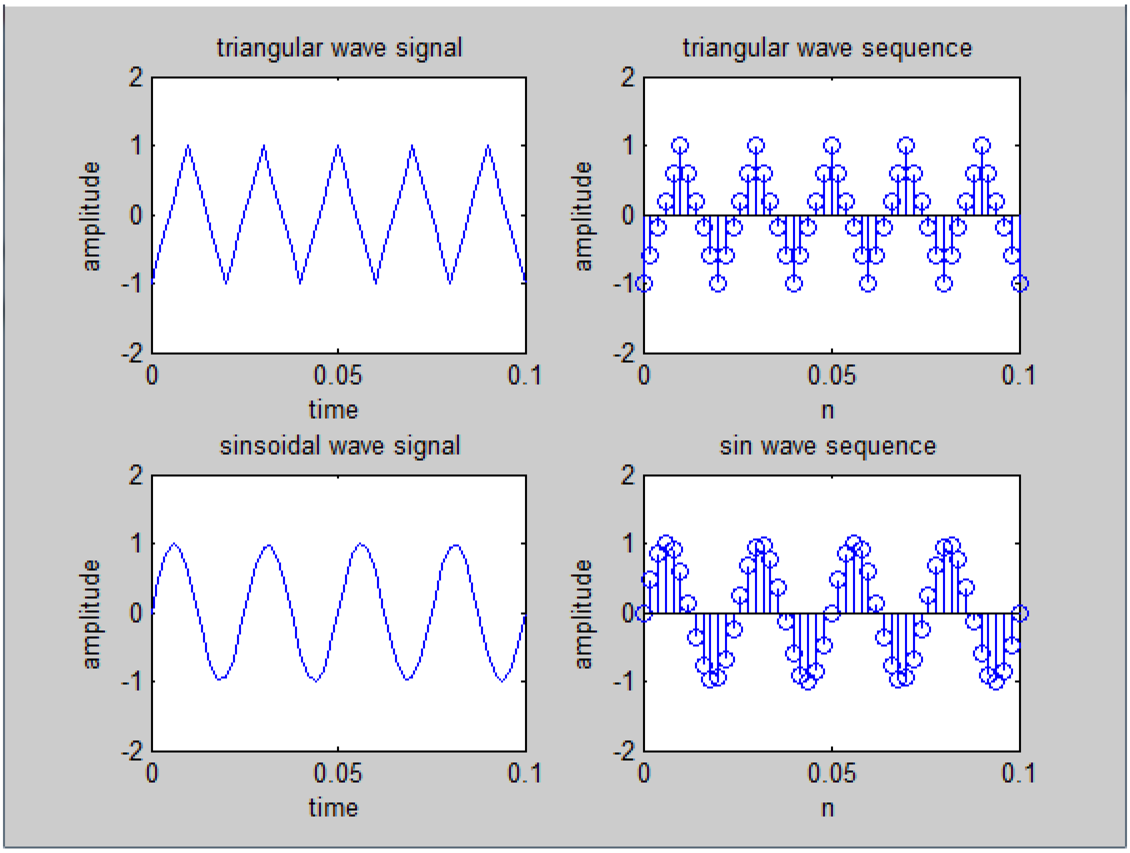
```
subplot(2,2,4);
stem(y8);
xlabel('n');
ylabel('amplitude');
title('sinc sequence');
```

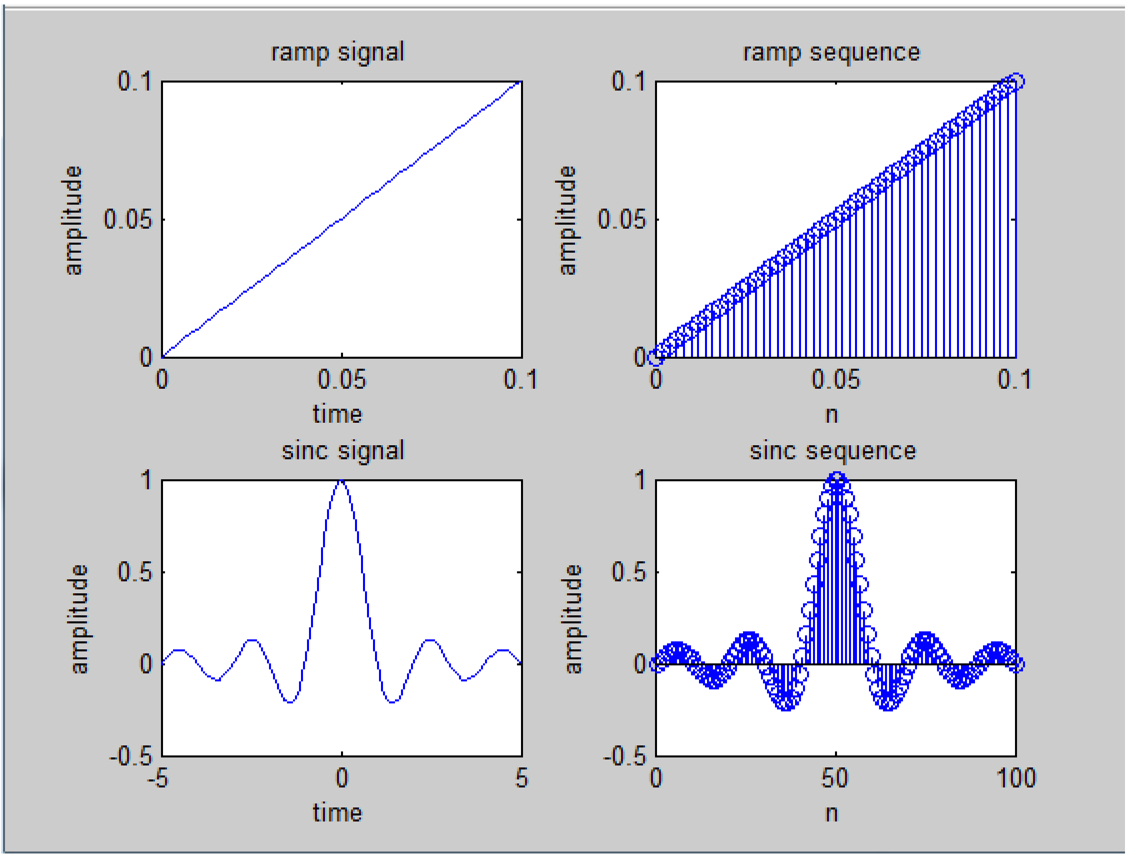
Result: Various signals & sequences generated using Matlab software.

Output:




square wave signal
amplitude
time
square wave sequence
amplitude
n
sawtooth wave signal
amplitude
time
sawtooth wave sequence
amplitude
n


triangular wave signal
amplitude
time
triangular wave sequence
amplitude
n
sinsoidal wave signal
amplitude
time
sin wave sequence
amplitude
n



VIVA QUESTIONS:-

1. Define Signal?
2. Define continuous and discrete Signals?
3. State the relation between step, ramp and Delta Functions?
4. Differentiate saw tooth and triangular signals?
5. Define Periodic and aperiodic Signal?

EXPERIMENT NO-3
BASIC OPERATIONS ON SIGNALS AND SEQUENCES

AIM: perform the operations on signals and sequences such as addition, multiplication, scaling, shifting, folding and also compute energy and power.

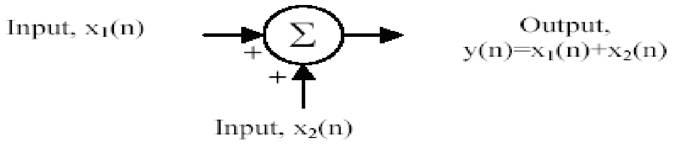
Software Required: Matlab software.

Theory:

Signal Addition

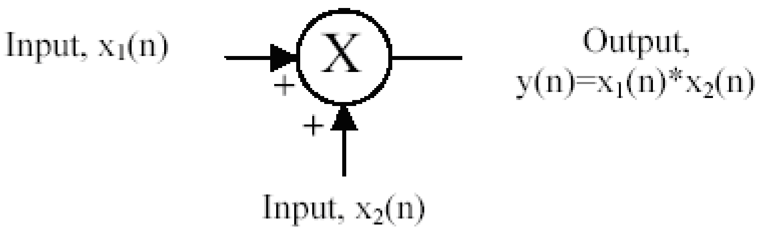
Addition: any two signals can be added to form a third signal,

z (t) = x (t) + y (t)



Multiplication :

Multiplication of two signals can be obtained by multiplying their values at every instants . z
z(t) = x (t) y (t)

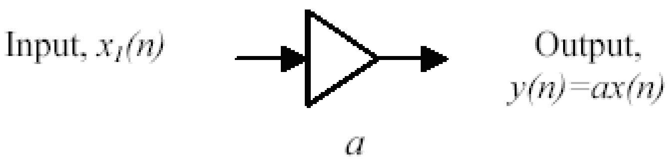


Time reversal/Folding:

Time reversal of a signal x(t) can be obtained by folding the signal about t=0. Y(t)=y(-t)

Signal Amplification/Scaling : Y(n)=ax(n) if a < 1 attnuation

a >1 amplification



Time shifting: The time shifting of x(n) obtained by delay or advance the signal in time by using y(n)=x(n+k)

If k is a positive number, y(n) shifted to the right i e the shifting delays the signal

If k is a negative number, y(n) it gets shifted left. Signal Shifting advances the signal

Energy:

$$E[n] = \lim_{N\to\infty} \sum_{n=-N}^{N} |x[n]|^2 = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

Average power:

$$P[n] = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2$$

Program:

```
clc;
clear all;
close all;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 % generating two input signals
t=0:.01:1;
x1=sin(2*pi*4*t);
x2=sin(2*pi*8*t);
subplot(2,2,1);
plot(t,x1);
xlabel('time');
ylabel('amplitude');
title('input signal 1');
subplot(2,2,2);
plot(t,x2);
xlabel('time');
ylabel('amplitude');
title('input signal 2');

% addition of signals
y1=x1+x2;
subplot(2,2,3);
plot(t,y1);
xlabel('time');
ylabel('amplitude');
title('addition of two signals');

% multiplication of signals
y2=x1.*x2;
subplot(2,2,4);
plot(t,y2);
xlabel('time');
ylabel('amplitude');
title('multiplication of two signals');
```

```
% scaling of a signal1
A=2;
y3=A*x1;
figure;
subplot(2,2,1);
plot(t,x1);
xlabel('time');
ylabel('amplitude');
title('input signal')
subplot(2,2,2);
plot(t,y3);
xlabel('time');
ylabel('amplitude');
title('amplified input signal');

% folding of a signal1
h=length(x1);
nx=0:h-1;
subplot(2,2,3);
plot(nx,x1);
xlabel('nx');
ylabel('amplitude');
title('input signal')
y4=fliplr(x1);
nf=-fliplr(nx);
subplot(2,2,4);
plot(nf,y4);
xlabel('nf');
ylabel('amplitude');
title('folded signal');

%shifting of a signal 1
figure;
subplot(3,1,1);
plot(t,x1);
xlabel('time t');
ylabel('amplitude');
title('input signal');
subplot(3,1,2);
plot(t+2,x1);
xlabel('t+2');
ylabel('amplitude');
title('right shifted signal');
subplot(3,1,3);
plot(t-2,x1);
xlabel('t-2');
ylabel('amplitude');
```

```
title('left shifted signal');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%operations on sequences
n1=1:1:9;
s1=[1 2 3 0 5 8 0 2 4];
figure;
subplot(2,2,1);
stem(n1,s1);
xlabel('n1');
ylabel('amplitude');
title('input sequence1');
s2=[1 1 2 4 6 0 5 3 6];
subplot(2,2,2);
stem(n1,s2);
xlabel('n2');
ylabel('amplitude');
title('input sequence2');

% addition of sequences
s3=s1+s2;
subplot(2,2,3);
stem(n1,s3);
xlabel('n1');
ylabel('amplitude');
title('sum of two sequences');

% multiplication  of sequences
s4=s1.*s2;
subplot(2,2,4);
stem(n1,s4);
xlabel('n1');
ylabel('amplitude');
title('product of two sequences');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% program for energy of a sequence
z1=input('enter the input sequence');
e1=sum(abs(z1).^2);
disp('energy of given sequence is');e1

% program for energy of a signal
t=0:pi:10*pi;
z2=cos(2*pi*50*t).^2;
e2=sum(abs(z2).^2);
disp('energy of given signal is');e2

% program for power of a sequence
p1= (sum(abs(z1).^2))/length(z1);
```

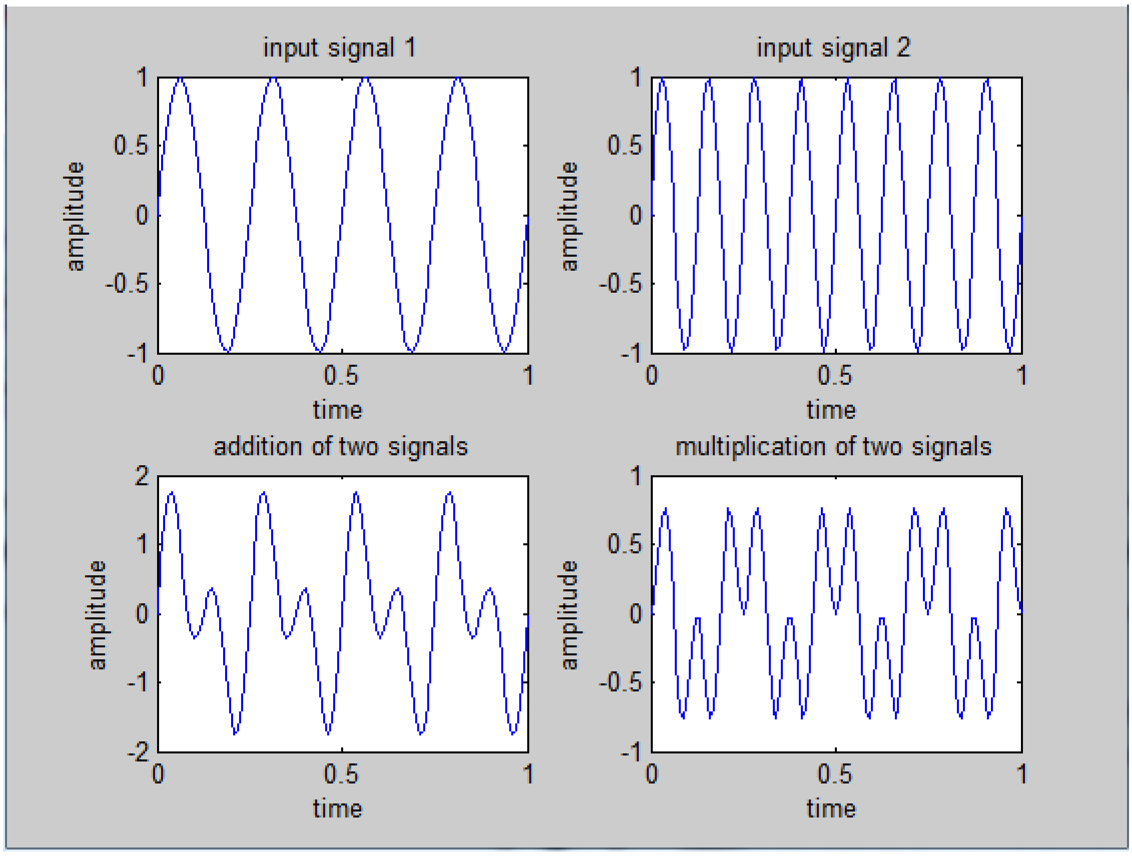
```
disp('power of given sequence is');p1
% program for power of a signal
p2=(sum(abs(z2).^2))/length(z2);
disp('power of given signal is');p2
```

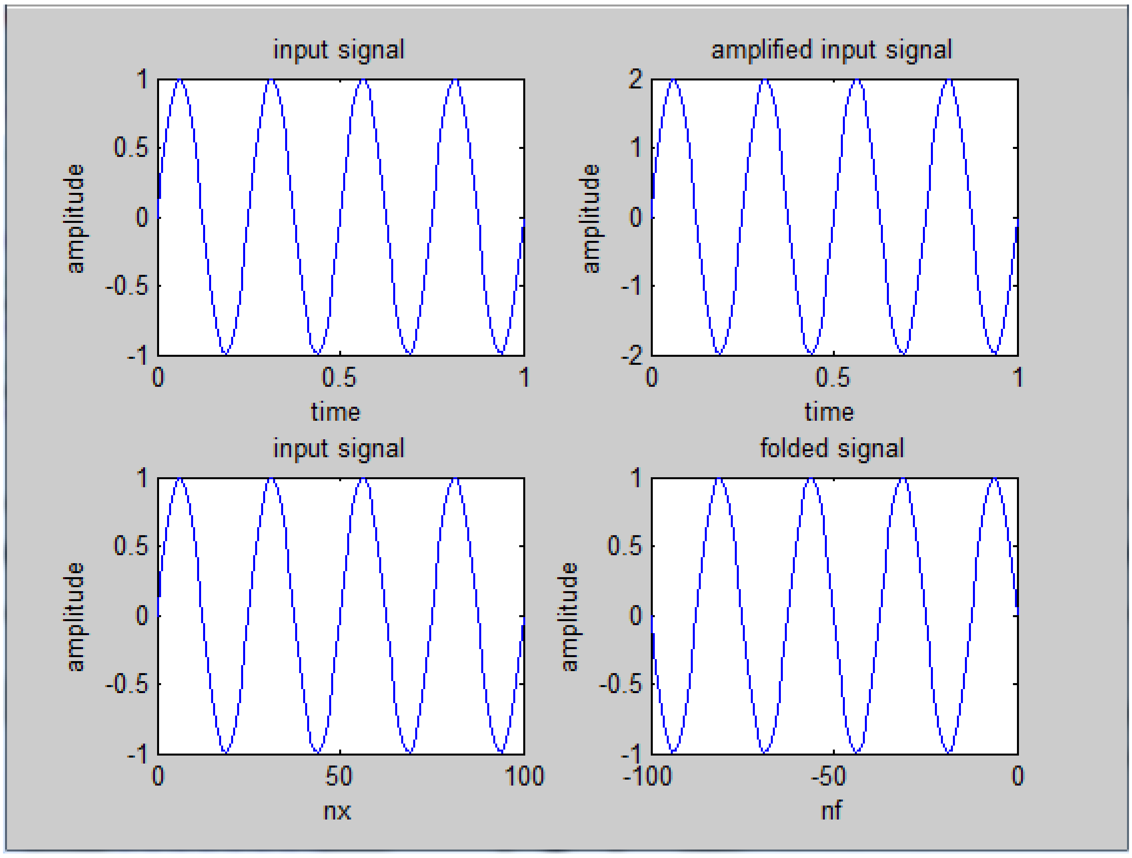
OUTPUT:

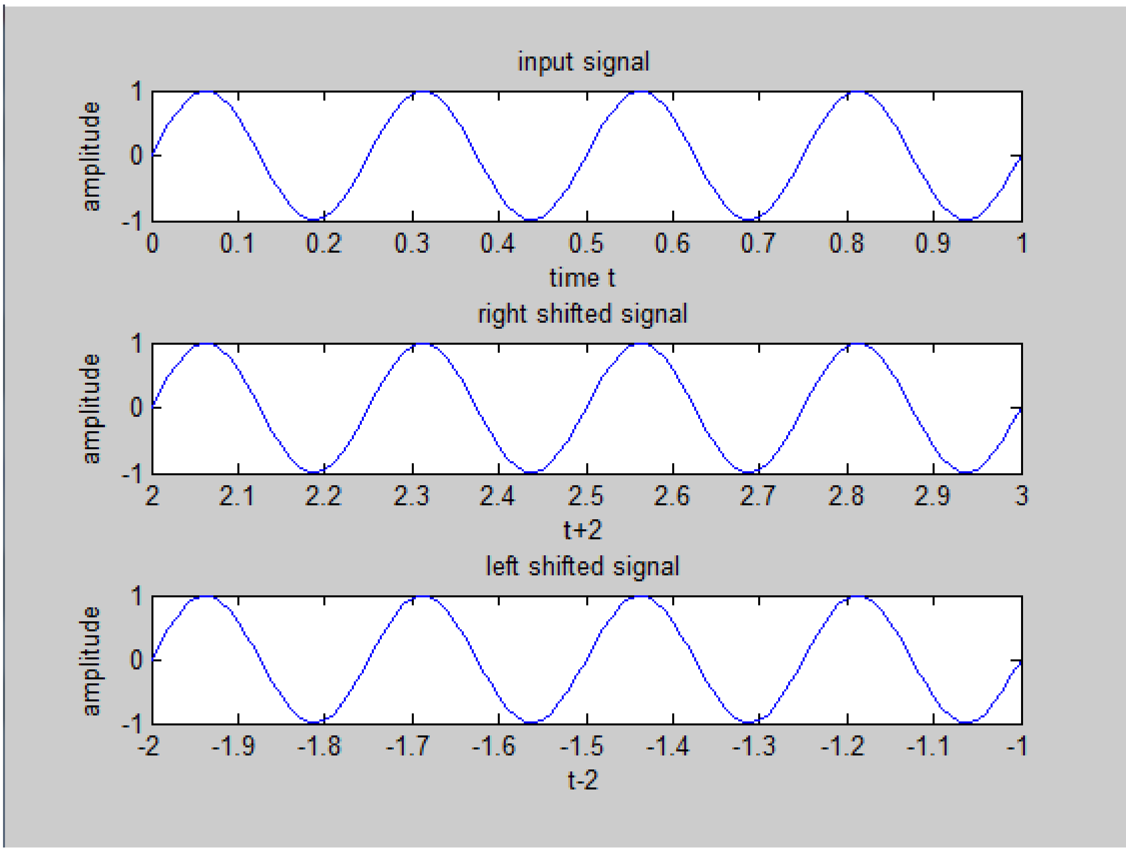
```
enter the input sequence[1 3 2 4 1]
energy of given sequence is
e1 = 31
energy of given signal is
e2 = 4.0388
power of given sequence is
p1 = 6.2000
power of given signal is
p2 =    0.3672
```

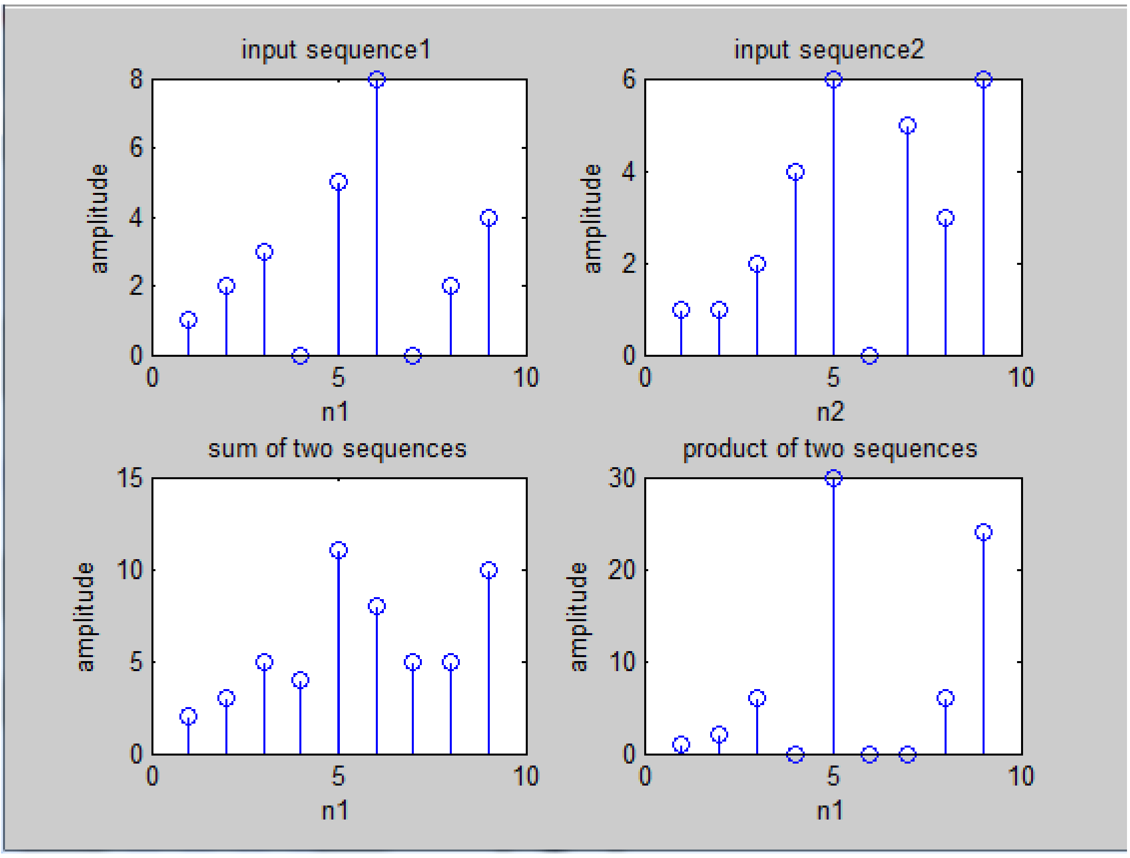
Result: Various operations on signals and sequences are performed.

Output:


input signal 1
input signal 2
addition of two signals
multiplication of two signals
amplitude
time
1
0.5
0
-0.5
-1
2
-2


input signal
amplified input signal
amplitude
time
input signal
folded signal
nx
nf
1
0.5
0
-0.5
-1
2
-2
50
100
-100
-50


input signal
amplitude
time t
right shifted signal
t+2
left shifted signal
t-2



VIVA QUESTIONS:-

1. Define Symmetric and Anti-Symmetric Signals?
2. Define scaling of a signal?
3. What are the Different types of representation of discrete time signals?
4. What are the Different types of Operation performed on signals?
5. What is System?

EXPERIMENT NO-4
EVEN AND ODD PARTS OF SIGNAL AND SEQUENCE & REAL AND IMAGINARY PARTS OF SIGNAL

AIM: Finding even and odd part of the signal and sequence and also find real and imaginary parts of signal.

Software Required: Matlab software

Theory: One of characteristics of signal is symmetry that may be useful for signal analysis. Even signals are symmetric around vertical axis, and Odd signals are symmetric about origin.

Even Signal: A signal is referred to as an even if it is identical to its time-reversed counterparts; x(t) = x(-t).

Odd Signal: A signal is odd if x(t) = -x(-t).

An odd signal must be 0 at t=0, in other words, odd signal passes the origin.

Using the definition of even and odd signal, any signal may be decomposed into a sum of its even part, xe(t), and its odd part, xo(t), as follows

Even and odd part of a signal: Any signal x(t) can be expressed as sum of even and odd components i.e .,

$x(t)=x_e(t)+x_o(t)$

$$x_e(t) = \frac{1}{2}\{x(t) + x(-t)\}$$

$$x_o(t) = \frac{1}{2}\{x(t) - x(-t)\}$$

$$\begin{aligned} x(t) &= x_e(t) + x_o(t) \\ &= \frac{1}{2}\{x(t) + x(-t)\} + \frac{1}{2}\{x(t) - x(-t)\} \end{aligned}$$

Program:

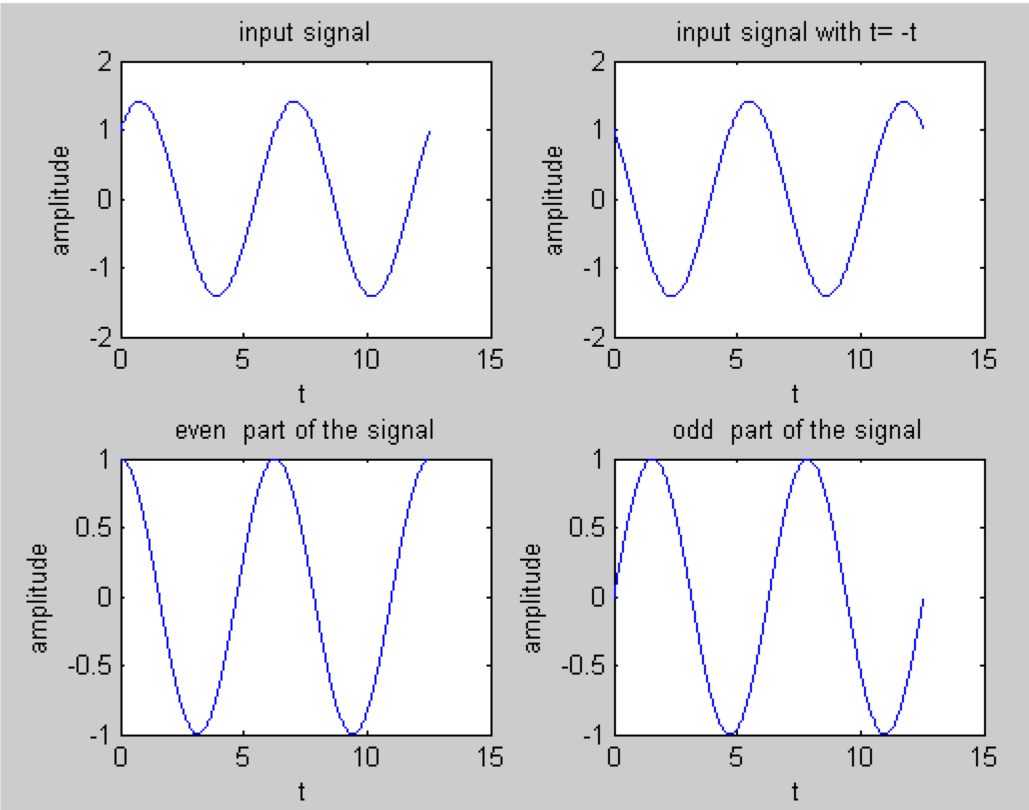
```
clc
close all;
clear all;

%Even and odd parts of a signal
t=0:.001:4*pi;
x=sin(t)+cos(t);     % x(t)=sint(t)+cos(t)
subplot(2,2,1)
plot(t,x)
xlabel('t');
ylabel('amplitude')
title('input signal')

y=sin(-t)+cos(-t);  % y(t)=x(-t)
subplot(2,2,2)
plot(t,y)
xlabel('t');
ylabel('amplitude')
```

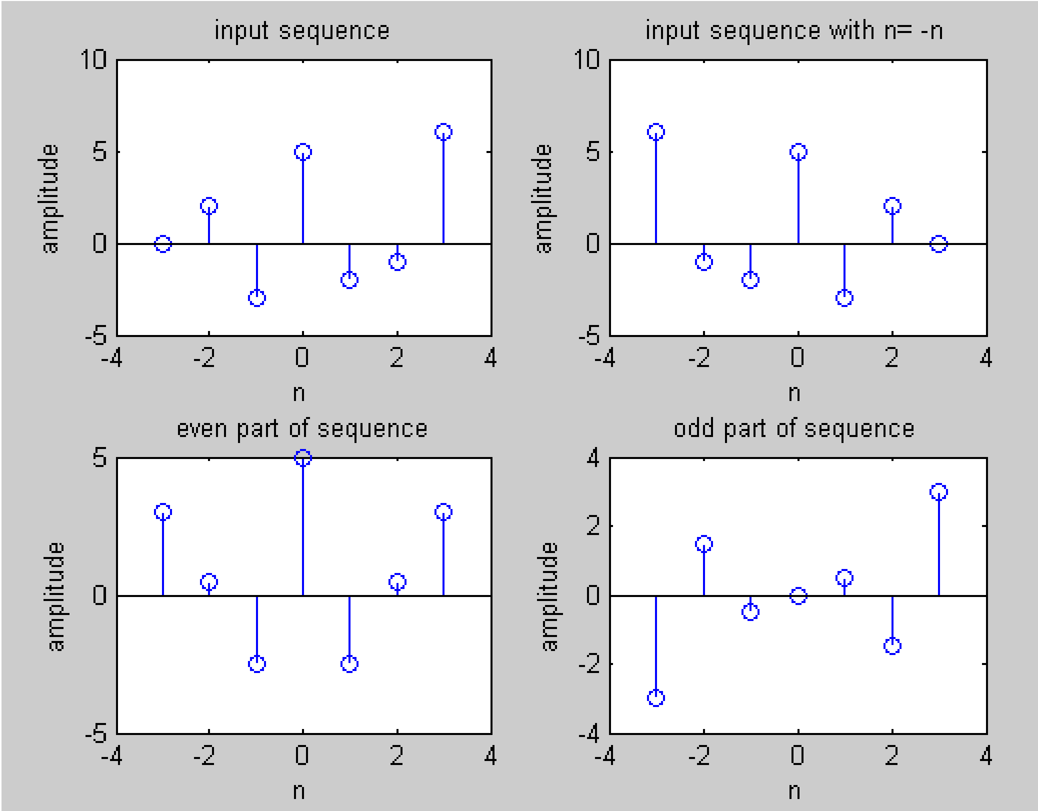
```
title('input signal with t= -t')
even=(x+y)/2;
subplot(2,2,3)
plot(t,even)
xlabel('t');
ylabel('amplitude')
title('even  part of the signal')
odd=(x-y)/2;
subplot(2,2,4)
plot(t,odd)
xlabel('t');
ylabel('amplitude');
title('odd  part of the signal');

% Even and odd parts of a sequence
x1=[0,2,-3,5,-2,-1,6];
n=-3:3;
y1= fliplr(x1);%y1(n)=x1(-n)
figure;
subplot(2,2,1);
stem(n,x1);
xlabel('n');
ylabel('amplitude');
title('input sequence');
subplot(2,2,2);
stem(n,y1);
xlabel('n');
ylabel('amplitude');
title('input sequence with n= -n');
even1=.5*(x1+y1);
odd1=.5*(x1-y1);

% plotting even and odd parts of the sequence
subplot(2,2,3);
stem(n,even1);
xlabel('n');
ylabel('amplitude');
title('even part of sequence');
subplot(2,2,4);
stem(n,odd1);
xlabel('n');
ylabel('amplitude');
title('odd part of sequence');

% plotting real and imginary parts of the signal
x2=sin(t)+j*cos(t);
figure;
```

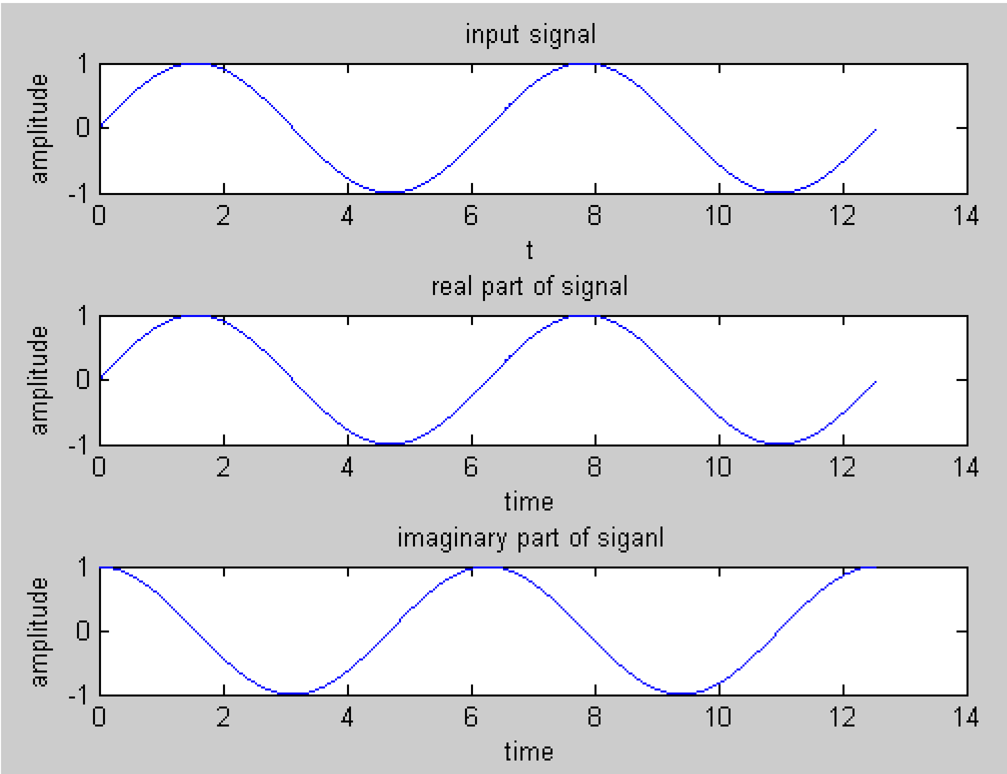
```
subplot(3,1,1);
plot(t,x2);
xlabel('t');
ylabel('amplitude');
title('input signal');
subplot(3,1,2)
plot(t,real(x2));
xlabel('time');
ylabel('amplitude');
title('real part of signal');
subplot(3,1,3)
plot(t,imag(x2));
xlabel('time');
ylabel('amplitude');
title('imaginary part of siganl');
```

RESULT: Even and odd part of the signal and sequence, real and imaginary parts of signal are computed.

Output:




input sequence
input sequence with n= -n
even part of sequence
odd part of sequence
amplitude
n


input signal
real part of signal
imaginary part of siganl
amplitude
t
time

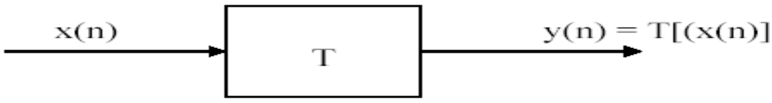
VIVA QUESTIONS:-

1. What is the formula to find odd part of signal?
2. What is Even Signal?
3. What is Odd Signal?
4. What is the formula to find even part of signal?
5. How to represent a signal with even and odd parts?

EXPERIMENT NO-5
CONVOLUTION BETWEEN SIGNALS& SEQUENCES

Aim: Write the program for convolution between two signals and also between two sequences.

Software Required: Matlab software
Theory:



Convolution involves the following operations.
1. Folding
2. Multiplication
3. Addition
4. Shifting

$$y[n] = \mathbf{T} \sum_{k=-\infty} x[k] . \delta(n\text{-}k)$$

These operations can be represented by a Mathematical Expression as follows:
x[n]= Input signal Samples
h[n-k]= Impulse response co-efficient.
y[n]= Convolution output.
n = No. of Input samples
h = No. of Impulse response co-efficient.
Example : X(n)={1 2 -1 0 1}, h(n)={ 1,2,3,-1}

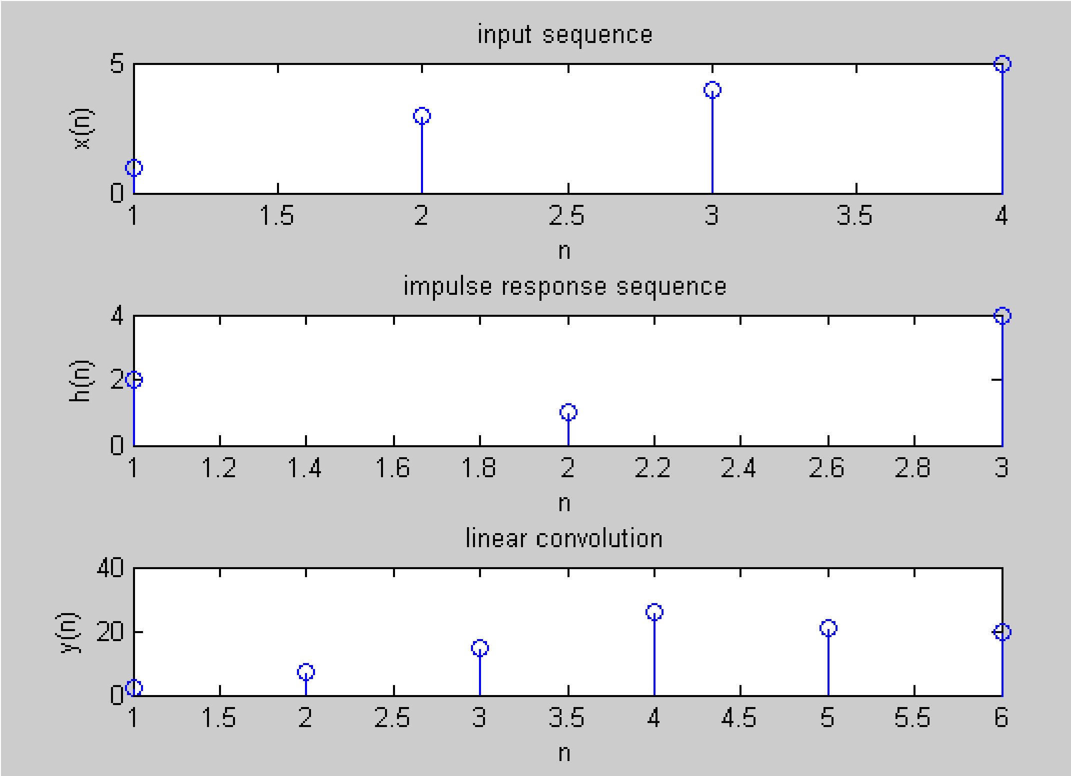
Program:
```
clc;
close all;
clear all;
%program for convolution of two sequences
x=input('enter input sequence: ');
h=input('enter impulse response: ');
y=conv(x,h);
subplot(3,1,1);
stem(x);
xlabel('n');
ylabel('x(n)');
title('input sequence')
subplot(3,1,2);
stem(h);
xlabel('n');
ylabel('h(n)');
```

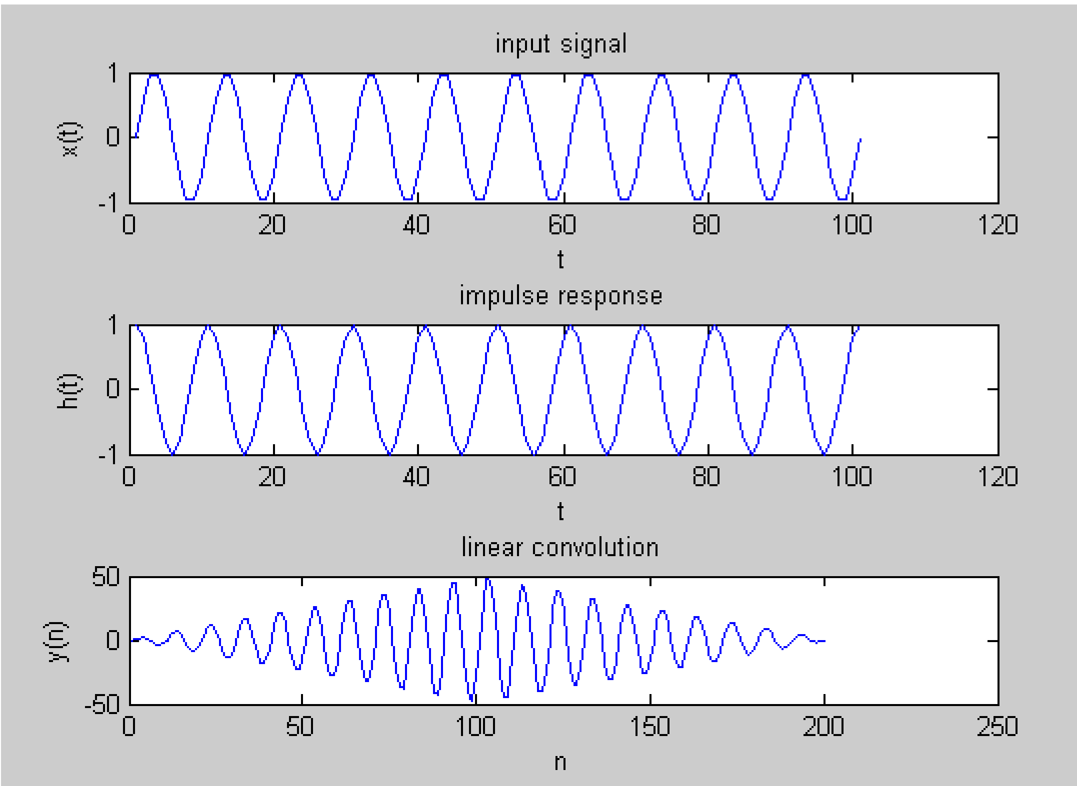
```
title('impulse response sequence')
subplot(3,1,3);
stem(y);
xlabel('n');
ylabel('y(n)');
title('linear convolution')
disp('linear convolution y=');
disp(y)
%program for signal convolution
t=0:0.1:10;
x1=sin(2*pi*t);
h1=cos(2*pi*t);
y1=conv(x1,h1);
figure;
subplot(3,1,1);
plot(x1);
xlabel('t');
ylabel('x(t)');
title('input signal')
subplot(3,1,2);
plot(h1);
xlabel('t');
ylabel('h(t)');
title('impulse response')
subplot(3,1,3);
plot(y1);
xlabel('n');
ylabel('y(n)');
title('linear convolution');
```

RESULT: convolution between signals and sequences is computed.

Output:

```
enter input sequence: [1 3 4 5]

enter impulse response: [2 1 4]

linear convolution y=

    2    7   15   26   21   20
```


input sequence
x(n)
n
impulse response sequence
h(n)
n
linear convolution
y(n)
n


input signal
x(t)
t
impulse response
h(t)
t
linear convolution
y(n)
n

VIVA QUESTIONS:-

1. Define Convolution?
2. Define Properties of Convolution?
3. What is the Difference between Convolution& Correlation?
4. Define impulse response?
5. What is Half Wave Symmetry?

EXPERIMENT NO-6
AUTO CORRELATION AND CROSS CORRELATION

Aim: To compute Auto correlation and Cross correlation between signals and sequences.

Software Required: Mat lab software

Theory:

Correlations of sequences:

It is a measure of the degree to which two sequences are similar. Given two real-valued sequences *x*(*n*) and *y*(*n*) of finite energy,

Convolution involves the following operations.

1. Shifting
2. Multiplication
3. Addition

These operations can be represented by a Mathematical Expression as follows:

Cross correlation

$$r_{x,y}(l) = \sum_{n=-\infty}^{+\infty} x(n)y(n-l)$$

The index *l* is called the shift or lag parameter

Autocorrelation

$$r_{x,x}(l) = \sum_{n=-\infty}^{+\infty} x(n)x(n-l)$$

Program:

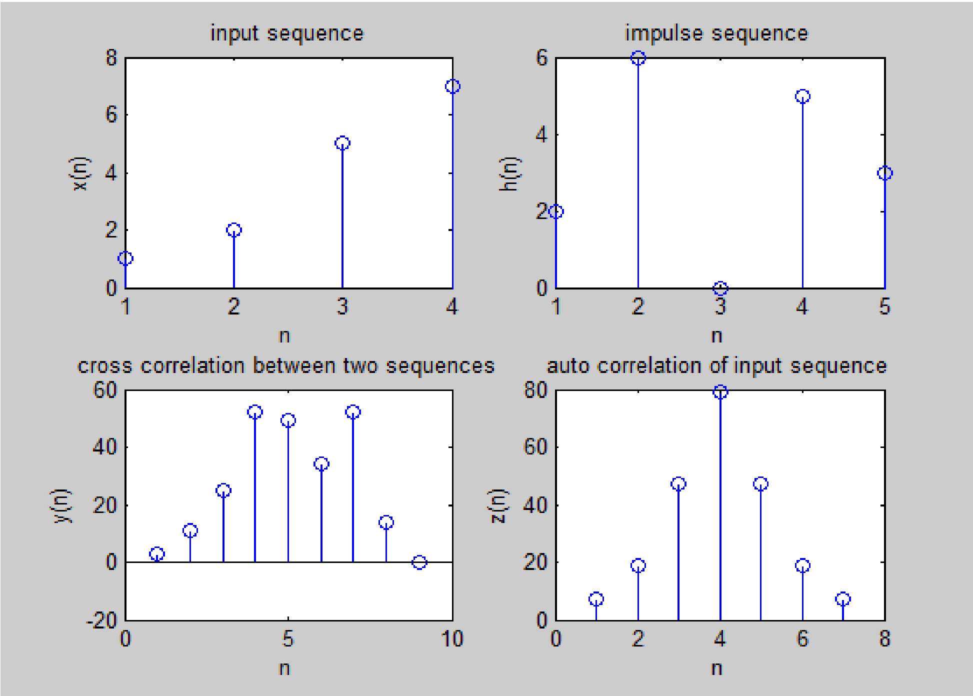
```
clc;
close all;
clear all;

% two input sequences
x=input('enter input sequence');
h=input('enter the impulse suquence');
subplot(2,2,1);
stem(x);
xlabel('n');
ylabel('x(n)');
title('input sequence');
subplot(2,2,2);
stem(h);
xlabel('n');
```

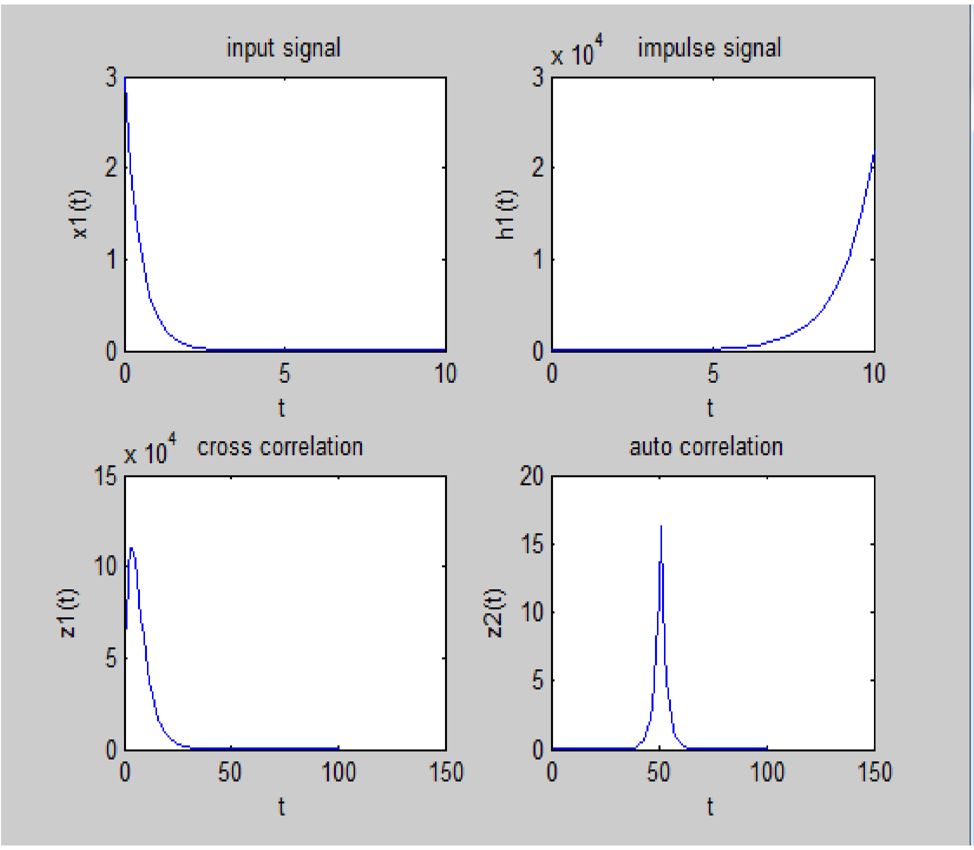
```
ylabel('h(n)');
title('impulse sequence');

% cross correlation between two sequences
y=xcorr(x,h);
subplot(2,2,3);
stem(y);
xlabel('n');
ylabel('y(n)');
title(' cross correlation between two sequences ');
% auto correlation of input sequence
z=xcorr(x,x);
subplot(2,2,4);
stem(z);
xlabel('n');
ylabel('z(n)');
title('auto correlation of input sequence');

%  cross correlation between two signals
% generating two input signals
t=0:0.2:10;
x1=3*exp(-2*t);
h1=exp(t);
figure;
subplot(2,2,1);
plot(t,x1);
xlabel('t');
ylabel('x1(t)');
title('input signal');
subplot(2,2,2);
plot(t,h1);
xlabel('t');
ylabel('h1(t)');
title('impulse signal');

%  cross correlation
subplot(2,2,3);
z1=xcorr(x1,h1);
plot(z1);
xlabel('t');
ylabel('z1(t)');
title('cross correlation ');
% auto correlation
subplot(2,2,4);
z2=xcorr(x1,x1);
plot(z2);
xlabel('t');
```

ylabel('z2(t)');
title('auto correlation ');

Result: Auto correlation and Cross correlation between signals and sequences is computed.

Output: enter input sequence [1 2 5 7]
enter the impulse sequence [2 6 0 5 3]




input signal
x1(t)
t
x 10^4
impulse signal
h1(t)
t
x 10^4
cross correlation
z1(t)
t
auto correlation
z2(t)
t

VIVA QUESTIONS:-

1. Define Correlation? And its properties?
2. Define Auto-Correlation?
3. Define Cross-Correlation?
4. What is the importance of correlation?
5. What is the difference b/w correlation and convolution?

EXPERIMENT NO-7

VERIFICATION OF LINEARITY OF A DISCRETE SYSTEM

AIM: Verify the Linearity of a given Discrete System.

Software Required:
Mat lab software 7.0 and above

Theory:

LINEARITY PROPERTY:
Any system is said to be linear if it satisfies the superposition principal. superposition principal state that Response to a weighted sum of input signal equal to the corresponding weighted sum of the outputs of the system to each of the individual input signals.

If x(n) is a input signal and y(n) is a output signal then

y(n)=T[x(n)]

y1(n)=T[x1(n)] and y2(n)=T[x2(n)]

x3=[a*x1(n) +b *x2(n)]

Y3(n)= T [x3(n)]

T [a*x1(n)+b*x2(n)] = a y1(n)+ b y2(n)

Program:

```
% Verification of Linearity of a given System.
% a) y(n)=nx(n)  b) y=x^2(n)
clc;
clear all;
close all;

n=0:40;
a1=input('enter the scaling factor a1=');
a2=input('enter the scaling factor a2=');
x1=cos(2*pi*0.1*n);
x2=cos(2*pi*0.4*n);
x3=a1*x1+a2*x2;
%y(n)=n.x(n);
y1=n.*x1;
y2=n.*x2;
y3=n.*x3;
yt=a1*y1+a2*y2;
yt=round(yt);
y3=round(y3);
```

```
if y3==yt
   disp('given system [y(n)=n.x(n)]is Linear');
else
   disp('given system [y(n)=n.x(n)]is non Linear');
end
%y(n)=x(n).^2
x1=[1 2 3 4 5];
x2=[1 4 7 6 4];
x3=a1*x1+a2*x2;
y1=x1.^2;
y2=x2.^2;
y3=x3.^2;
yt=a1*y1+a2*y2;
if y3==yt
   disp('given system [y(n)=x(n).^2 ]is Linear');
else
   disp('given system is [y(n)=x(n).^2 ]non Linear');
end
```

Result: The Linearity of a given Discrete System is verified.

Output:

```
enter the scaling factor a1=3
enter the scaling factor a2=5
given system [y(n)=n.x(n)]is Linear
given system is [y(n)=x(n).^2 ]non Linear
```

VIVA QUESTIONS:-

1. Define linear system with example?
2. Define non- linear system with example?
3. Define super position principle?
4. Give mathematical expression for Linearity.
5. Identity the system $y(n)=[x(n)]^3$ is linear or non-linear system.

EXPERIMENT NO -8
VERIFICATION OF TIME INVARIANCE OF A DISCRETE SYSTEM

AIM: Verify the Time Invariance of a given Discrete System.

Software Required: Mat lab software
Theory:

TIME INVARIENT SYSTEMS (TI):

A system is called time invariant if its input – output characteristics do not change with time

X(t)---- input : Y(t) ---output
X(t-k) -----delay input by k seconds : Y(t-k) ------ Delayed output by k seconds

If Y(t)=T[X(t)] then Y(t-k)=T[X(t-k)] then system is time invariant system.

Program:

```
% Verification of Time Invariance of a Discrete System
% a)y=x^2(n)   b) y(n)=nx(n)
clc;
clear all;
close all;
n=1:9;
x(n)=[1 2 3 4 5 6 7 8 9];
d=3; % time delay
xd=[zeros(1,d),x(n)];%x(n-k)
y(n)=x(n).^2;
yd=[zeros(1,d),y];%y(n-k)
disp('transformation of delay signal yd:');disp(yd)
dy=xd.^2; % T[x(n-k)]
disp('delay of transformation signal dy:');disp(dy)
if dy==yd
   disp('given system [y(n)=x(n).^2 ]is time invariant');
else
   disp('given system is [y(n)=x(n).^2 ]not time invariant');
end
y=n.*x;
yd=[zeros(1,d),y(n)];
disp('transformation of delay signal yd:');disp(yd);
n1=1:length(xd);
dy=n1.*xd;
disp('delay of transformation signal dy:');disp(dy);
if yd==dy
   disp('given system [y(n)=nx(n)]is a time invariant');
```

```
else
   disp('given system [y(n)=nx(n)]not a time invariant');
end
```

Result: The Time Invariance of a given Discrete System is verified.

Output:

```
Transformation of delay signal yd:
    0    0    0    1    4    9   16   25   36   49   64   81

Delay of transformation signal dy:
    0    0    0    1    4    9   16   25   36   49   64   81

Given system [y(n)=x(n).^2 ]is time invariant
Transformation of delay signal yd:
    0    0    0    1    4    9   16   25   36   49   64   81

Delay of transformation signal dy:
    0    0    0    4   10   18   28   40   54   70   88  108

Given system [y(n)=nx(n)]not a time invariant
```

VIVA QUESTIONS:-

1. Define time invariant system with example?
2. Define time variant system with example?
3. Define LTI system?
4. Give mathematical expression for time invariant system?
5. Give another name for time invariant system and time variant system?

EXPERIMENT NO-9
UNIT SAMPLE, UNIT STEP AND SINUSOIDAL RESPONSE OF THE GIVEN LTI SYSTEM AND VERIFYING ITS STABILITY

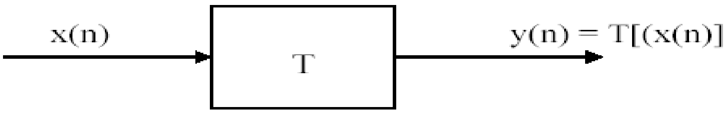
AIM: Compute the Unit sample, unit step and sinusoidal response of the given LTI system and verifying its stability

Software Required:

Mat lab software 7.0 and above

Theory:

A discrete time system performs an operation on an input signal based on predefined criteria to produce a modified output signal. The input signal x(n) is the system excitation, and y(n) is the system response. The transform operation is shown as,



If the input to the system is unit impulse i.e. x(n) = δ(n) then the output of the system is known as impulse response denoted by h(n) where,

$$h(n) = T[\delta(n)]$$

we know that any arbitrary sequence x(n) can be represented as a weighted sum of discrete impulses. Now the system response is given by,

$$y(n) = T[x(n)] = T\Big[\sum_{k=-\infty}^{\infty} x(k)\ \delta(n-k)\Big]$$

For linear system (1) reduces to

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\ T[\delta(n-k)]$$

%given difference equation y(n)-y(n-1)+.9y(n-2)=x(n);

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

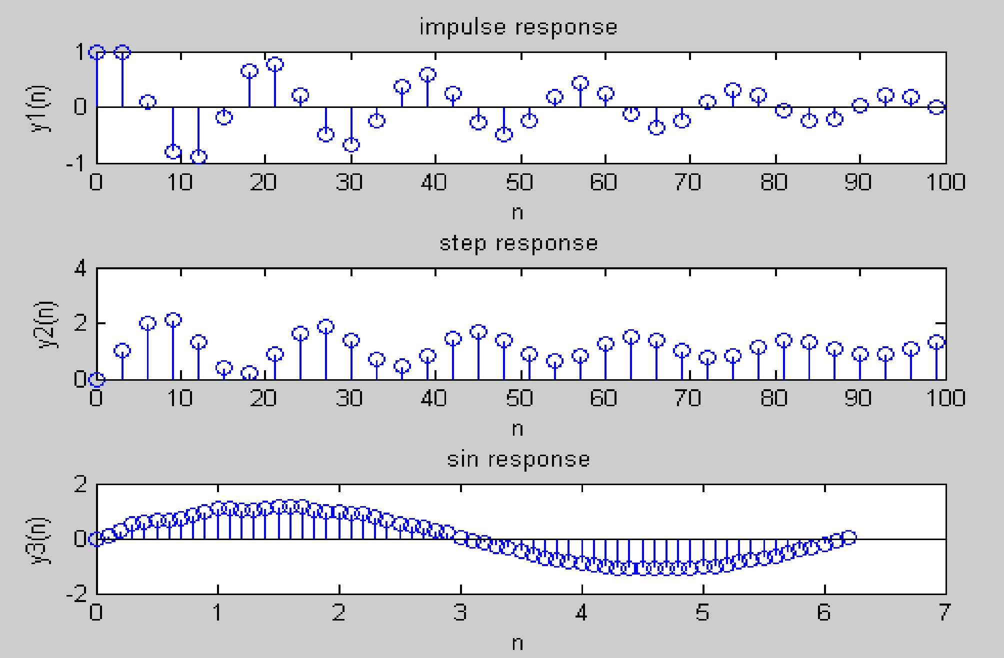
$$H(z) = \frac{b_0 + b_1 Z^{-1} + b_2 Z^{-2} + \dots\dots\dots\dots b_{N-1} Z^{(N-1)} + b_N Z^{-N}}{1 + a_1 Z^{-1} + a_2 Z^{-2} + \dots\dots\dots\dots a_{N-1} Z^{(N-1)} + a_N Z^{-N}}$$

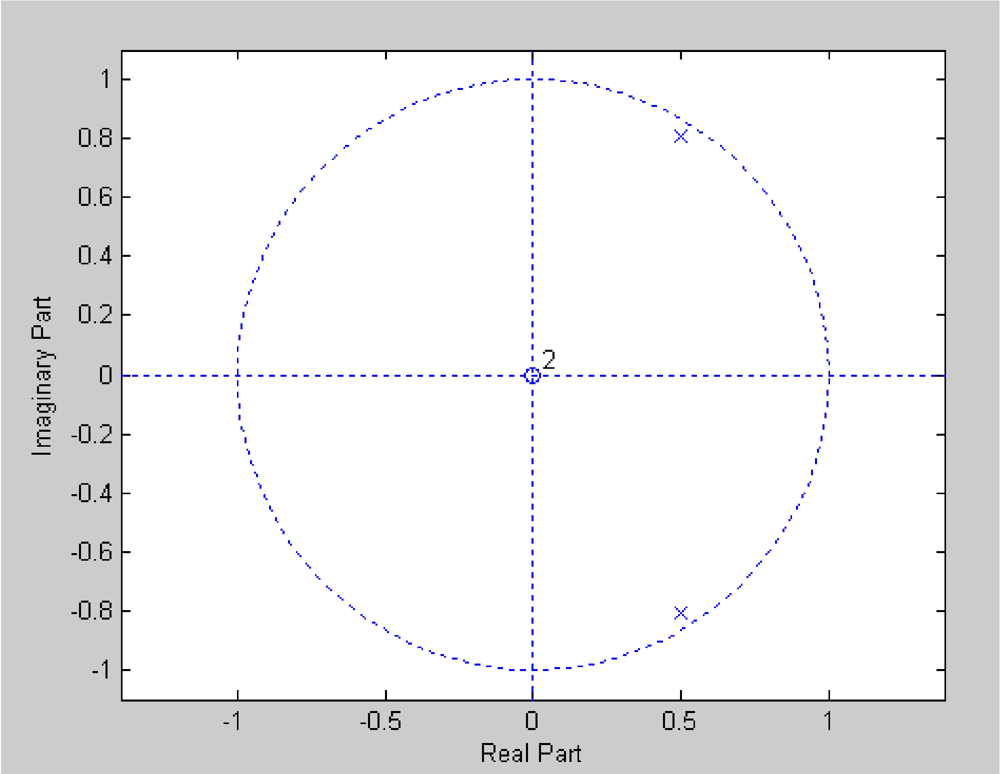
Program:

```
%given difference equation y(n)-y(n-1)+.9y(n-2)=x(n);
clc;
clear all;
close all;
b=[1];
a=[1,-1,.9];
n =0:3:100;

%generating impulse signal
x1=(n==0);
%impulse response
y1=filter(b,a,x1);
subplot(3,1,1);
stem(n,y1);
xlabel('n');
ylabel('y1(n)');
title('impulse response');

%generating step signal
x2=(n>0);
% step response
y2=filter(b,a,x2);
subplot(3,1,2);
stem(n,y2);
xlabel('n');
ylabel('y2(n)')
title('step response');
%generating sinusoidal signal
t=0:0.1:2*pi;
x3=sin(t);
% sinusoidal response
y3=filter(b,a,x3);
subplot(3,1,3);
stem(t,y3);
xlabel('n');
ylabel('y3(n)');
title('sin response');

% verifing stability
figure;
zplane(b,a);
```

Result: The Unit sample, unit step and sinusoidal response of the given LTI system is computed and its stability verified. Hence all the poles lie inside the unit circle, so system is stable.

Output:





VIVA QUESTIONS:-

1. What operations can be performed on signals and sequence?
2. Define causality and stability?
3. Define step response and impulse response of the system.
4. Define poles and zeros of the system?
5. What is the function of filter?

EXPERIMENT NO-10

FINDING THE FOURIER TRANSFORM OF A GIVEN SIGNAL AND PLOTTING ITS MAGNITUDE AND PHASE SPECTRUM

AIM: To find the Fourier Transform of a given signal and plotting its magnitude and phase spectrum.

Software Required: Matlab software

Theory:

Fourier Transform:

The Fourier transform as follows. Suppose that f is a function which is zero outside of some interval $[-L/2, L/2]$. Then for any $T \geq L$ we may expand f in a Fourier series on the interval $[-T/2, T/2]$, where the "amount" of the wave $e^{2\pi inx/T}$ in the Fourier series of f is given by

By definition Fourier Transform of signal f(t) is defined as

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t}\, dt$$

Inverse Fourier Transform of signal F(w) is defined as

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t}\, d\omega$$

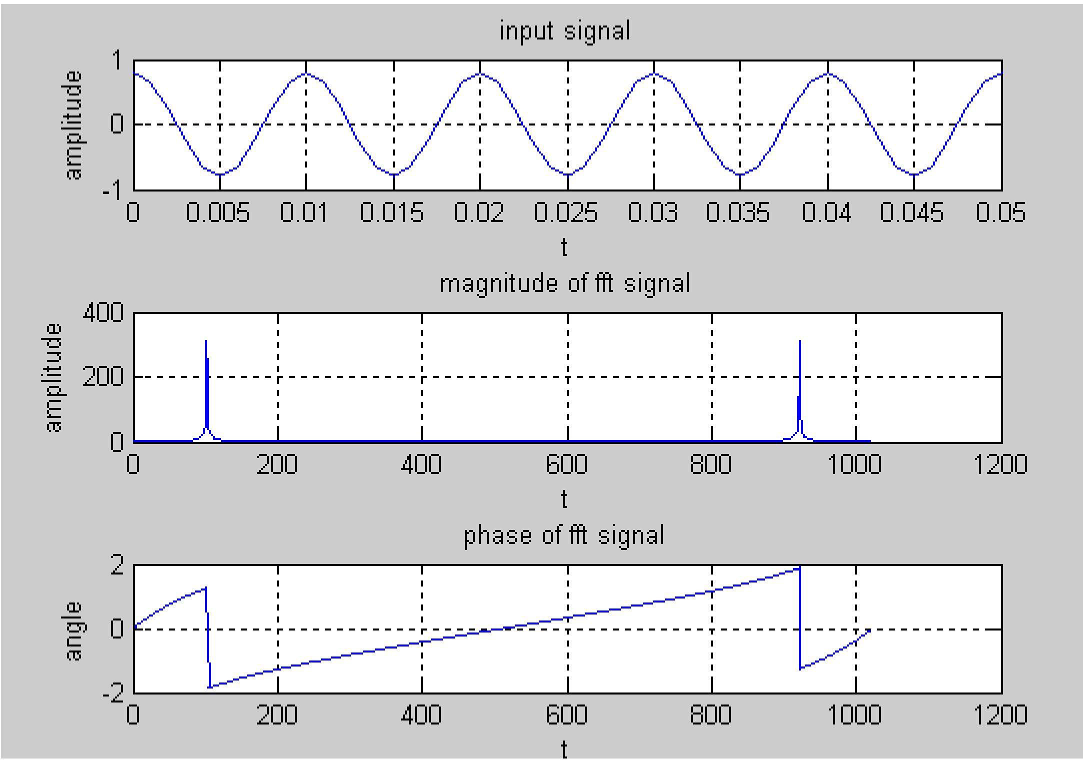
Program:

```
clc;
clear all;
close all;
 fs=1000;
 N=1024;   % length of fft sequence
t=[0:N-1]*(1/fs);
% input signal
x=0.8*cos(2*pi*100*t);
subplot(3,1,1);
plot(t,x);
axis([0 0.05 -1 1]);
grid;
xlabel('t');
ylabel('amplitude');
title('input signal');
% Fourier transformof given signal
x1=fft(x);
```

```
% magnitude spectrum
k=0:N-1;
Xmag=abs(x1);
subplot(3,1,2);
plot(k,Xmag);
grid;
xlabel('t');
ylabel('amplitude');
title('magnitude of fft signal')
%phase spectrum
Xphase=angle(x1);
subplot(3,1,3);
plot(k,Xphase);
grid;
xlabel('t');
ylabel('angle');
title('phase of fft signal');
```

Result: Magnitude and phase spectrum of FFT of a given signal is plotted.

Output:



VIVA QUESTIONS:

1. Define convolution property of Fourier transform?
2. What are the properties of Continuous-Time Fourier transform?
3. What is the sufficient condition for the existence of F.T?
4. Define the F.T of a signal?
5. What is the difference b/w F.T&F.S?

EXPERIMENT NO-11

LOCATING POLES AND ZEROS IN S-PLANE & Z-PLANE

AIM: Write the program for locating poles and zeros and plotting pole-zero maps in s-plane and z-plane for the given transfer function.

Software Required: Matlab software

Theory:

Z-transforms

The Z-transform, like many other integral transforms, can be defined as either a *one-sided* or *two-sided* transform.

Bilateral Z-transform

The *bilateral* or *two-sided* Z-transform of a discrete-time signal *x[n]* is the function *X(z)* defined as

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

Unilateral Z-transform

Alternatively, in cases where $x[n]$ is defined only for $n \geq 0$, the *single-sided* or *unilateral* Z-transform is defined as

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=0}^{\infty} x[n]z^{-n}$$

In signal processing, this definition is used when the signal is causal.

where $z = r.e^{j\omega}$

$$X(z) = \frac{P(z)}{Q(z)}$$

The roots of the equation P(z) = 0 correspond to the 'zeros' of X(z)
The roots of the equation Q(z) = 0 correspond to the 'poles' of X(z)

Example:

$$H(z) = \frac{z+1}{(z-\frac{1}{2})(z+\frac{3}{4})}$$

The zeros are: $\{-1\}$

The poles are: $\left\{\frac{1}{2}, -\left(\frac{3}{4}\right)\right\}$

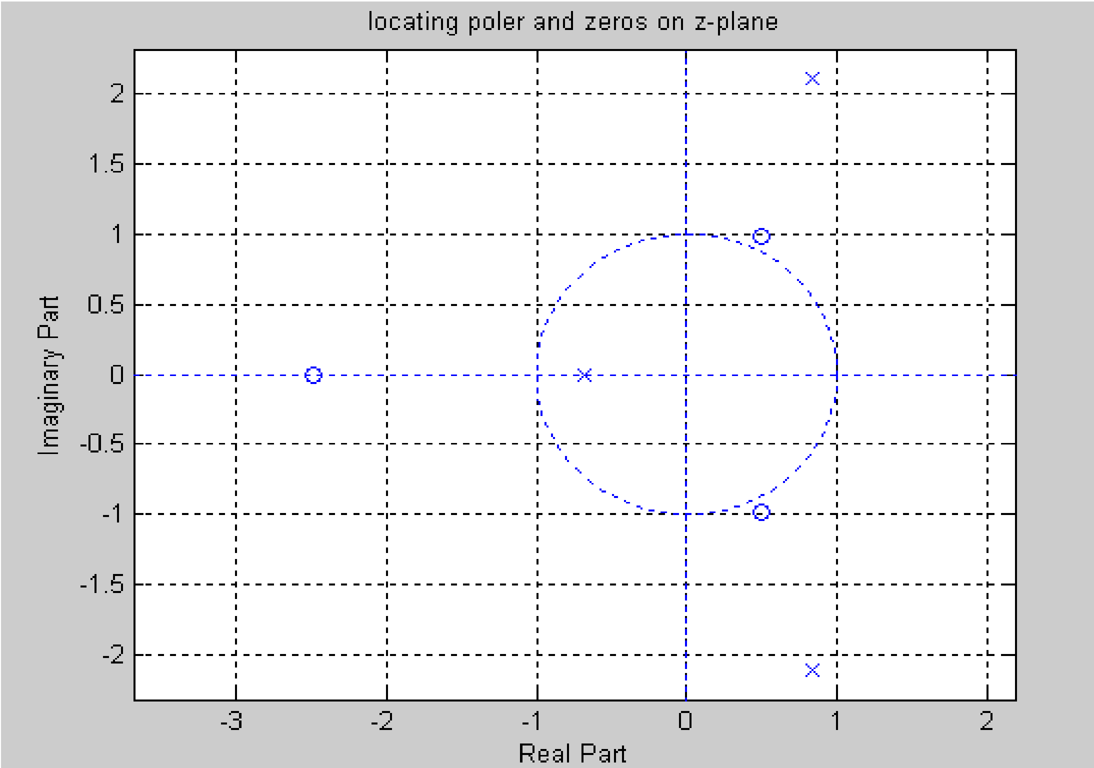
Program:

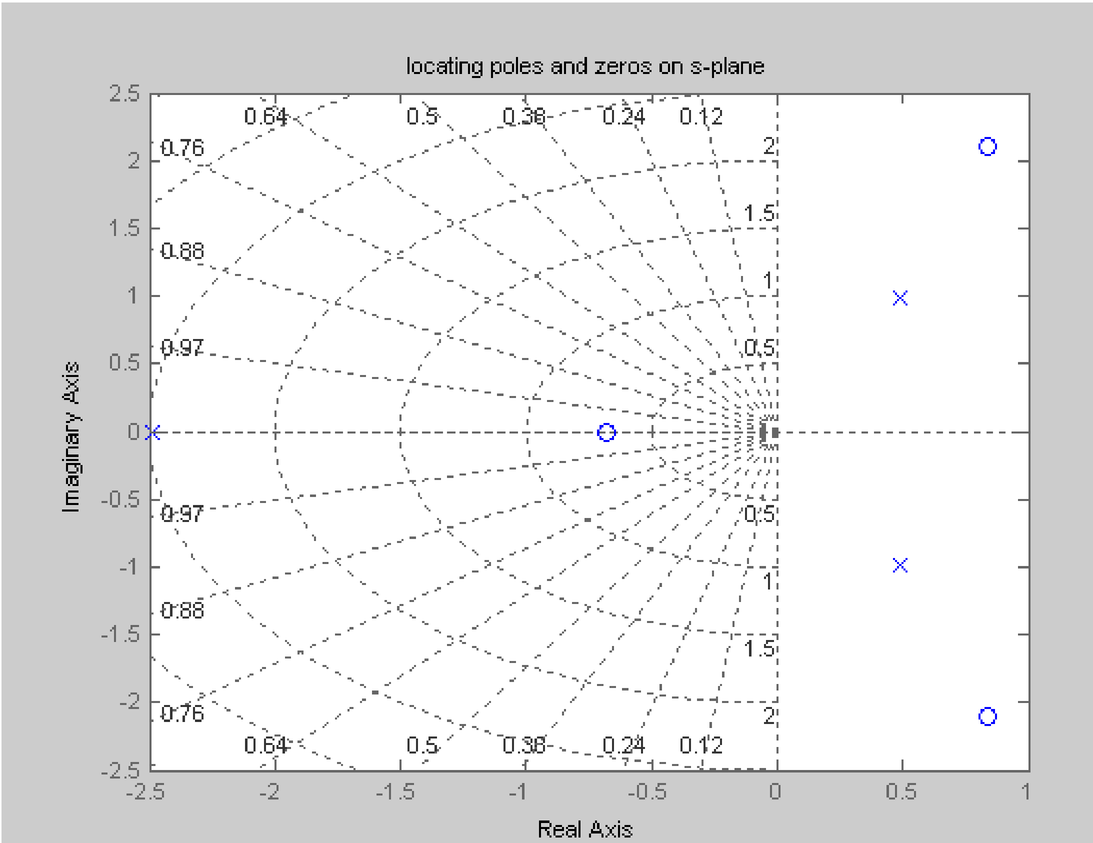
```
clc;
clear all;
close all;
%enter the numerator and denamenator cofficients in square brackets
num=input('enter numerator co-efficients');
den=input('enter denominator co-efficients');

% find poles and zeros
poles=roots(den)
zeros=roots(num)

% find transfer function H(s)
h=tf(num,den);

% plot the pole-zero map in s-plane
sgrid;
pzmap(h);
grid on;
title('locating poles and zeros on s-plane');

%plot the pole zero map in z-plane
figure
zplane(poles,zeros);
grid on;
title('locating poler and zeros on z-plane');
```

Result: Pole-zero maps are plotted in s-plane and z-plane for the given transfer function.

Output:

```
enter numerator co-efficients[1 -1 4 3.5]

enter denominator co-efficients[2 3 -2.5 6]

poles =

  -2.4874

   0.4937 + 0.9810i

   0.4937 - 0.9810i

zeros =

   0.8402 + 2.1065i

   0.8402 - 2.1065i

  -0.6805
```


locating poler and zeros on z-plane
Imaginary Part
Real Part
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-3
-2
-1
0
1
2


locating poles and zeros on s-plane
Imaginary Axis
Real Axis
0.64
0.5
0.38
0.24
0.12
0.76
0.88
0.97
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-2
-1.5
-1
-0.5
0
0.5
1

VIVA QUESTIONS:-

1. Study the details of pzmap() and zplane() functions?
2. What are poles and zeros?
3. How you specify the stability based on poles and zeros?
4. Define S-plane and Z-plane?
5. Define transfer function of the system?

EXPERIMENT NO-12
SAMPLING THEOREM VERIFICATION

AIM: Verify the sampling theorem.

Software Required: Matlab software

Theory:

Sampling Theorem:

A bandlimited signal can be reconstructed exactly if it is sampled at a rate atleast twice the maximum frequency component in it." Figure 1 shows a signal g(t) that is bandlimited.

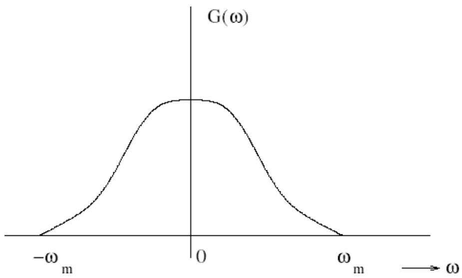


Figure 1: Spectrum of band limited signal g(t)

The maximum frequency component of g(t) is fm. To recover the signal g(t) exactly from its samples it has to be sampled at a rate fs ≥ 2fm.

The minimum required sampling rate fs = 2fm is called ' Nyquist rate

Proof: Let g(t) be a bandlimited signal whose bandwidth is fm (wm = 2πfm).

Figure 2: (a) Original signal g(t) (b) Spectrum G(w)

δ (t) is the sampling signal with fs = 1/T > 2fm.

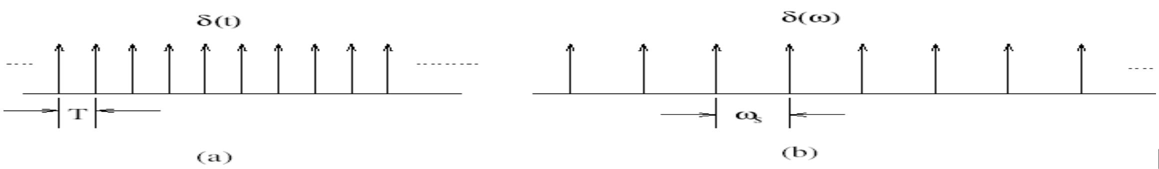


Figure 3: (a) sampling signal δ (t)) (b) Spectrum δ (w)

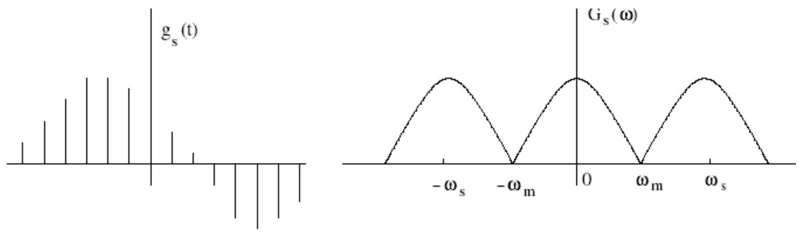


Figure 4: (a) sampled signal gs(t) (b) Spectrum Gs(w)

To recover the original signal G(w):

1. Filter with a Gate function, H2wm(w) of width 2wm

Scale it by T.

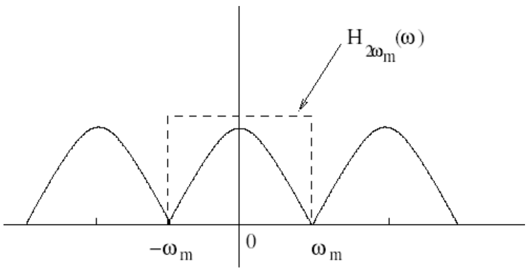


Figure 5: Recovery of signal by filtering with a fiter of width 2wm Aliasing ws < 2wm.

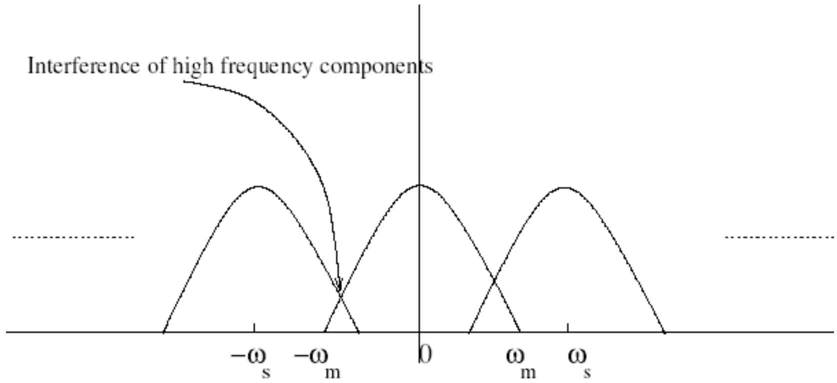


Figure 6: Aliasing due to inadequate sampling

Aliasing leads to distortion in recovered signal. This is the reason why sampling frequency should be atleast twice thebandwidth of the signal. Oversampling ws >2wm. This condition avoid aliasing.

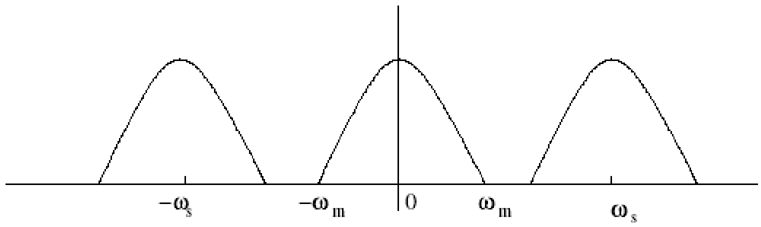


Figure 7: Oversampled signal-avoids aliasing

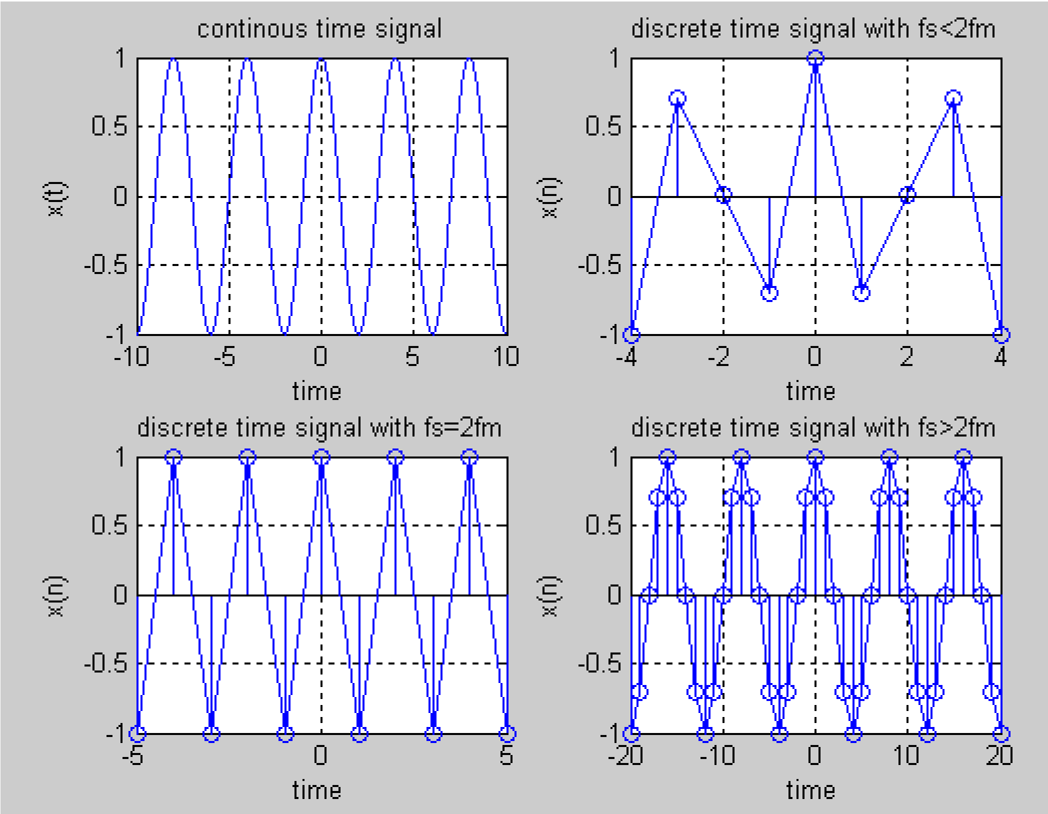
Program:

```
clc;
clear all;
close all;
t=-10:.01:10;
T=4;
fm=1/T;
x=cos(2*pi*fm*t);
subplot(2,2,1);
plot(t,x);
xlabel('time');
ylabel('x(t)');
title('continous time signal');
grid;
n1=-4:1:4;
fs1=1.6*fm;
fs2=2*fm;
fs3=8*fm;
x1=cos(2*pi*fm/fs1*n1);
subplot(2,2,2);
stem(n1,x1);
xlabel('time');
ylabel('x(n)');
title('discrete time signal with fs<2fm');
hold on;
subplot(2,2,2);
plot(n1,x1);
grid;
n2=-5:1:5;
x2=cos(2*pi*fm/fs2*n2);
subplot(2,2,3);
stem(n2,x2);
xlabel('time');
ylabel('x(n)');
title('discrete time signal with fs=2fm');
hold on;
subplot(2,2,3);
plot(n2,x2)
grid;
n3=-20:1:20;
x3=cos(2*pi*fm/fs3*n3);
subplot(2,2,4);
stem(n3,x3);
xlabel('time');
ylabel('x(n)');
```

```
title('discrete time signal with fs>2fm')
hold on;
subplot(2,2,4);
plot(n3,x3)
grid;
```

Result: Sampling theorem is verified.

OUTPUT:



VIVA QUESTIONS:-

1. State Paeseval's energy theorem for a periodic signal?
2. Define sampling Theorem?
3. What is Aliasing Effect?
4. What is under sampling?
5. What is over sampling?