SIGNALS AND SYSTEMS

LAB MANUAL

EC-253 (R20)

Prepared By

Mrs. DIVYA GUDAPATI, **M.Tech, (Ph.D), MIEEE, MIE**

Assistant Professor

Department of ECE

Department of Electronics and Communications Engineering

Bapatla Women's Engineering College

(Affiliated to AcharyaNagarjunaUniversity,GUNTUR)

Bapatla-522101

BAPATLA WOMEN'S ENGINEERING COLLEGE::BAPATLA

AN ISO 9001-2015 CERTIFIED INSTITUTION

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

INSTITUTE VISION AND MISSION

Vision

Bapatla Women's Engineering College has been established with forethought to organize the growing needs of higher technical education for WOMEN, in order to provide them a competative edge and a unique niche in the society.

Mission

Our mission is to impart the quality education at par with global standards to the students from all over India and in particular those from the local and rural areas.We continuously try to maintain high standards so as to make them technologically competent and ethically strong individuals who shall be able to improve the quality of life and economy of our country.

DEPARTMENT VISION & MISSION

VISION

To produce highly professional and socially responsible women engineers through world class quality technical education and research to meet industry and Society needs in the field of Electronics and Communication Engineering.

MISSION

M1- To inculcate quality education for women through innovative learning practices with the use of modern tools.

M2- To impart professional ethics and moral values through co-curricular and extra-curricular activities as individual and team member.

M3- Empowering graduates through industrial trainings/on-job training in collaboration with industries with-a-view to life-long learning.

PROGRAM EDUCATIONALOBJECTIVES (PEO's)

PEO1.To develop and implementation of Electronics and Communication systems by imparting technical knowledge, professional leadership and communication skills as an individual and team in a multi-disciplinary stream.

PEO2. To train Graduates in solving complex problems, analyzing, designing and imparting research-based knowledge by using modern scientific tools to develop least cost-effective and innovative products.

PEO3.To produce women graduates through life-long learning, who ensure ethical and moral behavior.

PROGRAM SPECIFIC OUTCOMES (PSO's)

PSO1.To understand and analyze basic concepts of electronics and communication in various areas like signal processing, VLSI, embedded systems, communications, digital and analog devices.

PSO2.Ability to design and develop computing systems using concepts of mathematics, electronics engineering and other related inter disciplines to asses and resolve societal, environmental, health safety and cultural issues with professional ethics.

BAPATLA WOMEN'S ENGINEERING COLLEGE::BAPATLA
AN ISO 9001-2015 CERTIFIED INSTITUTION
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ProgrammeOutcomes(PO's)

- **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 analyse 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 prediction and modelling 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 the 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 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 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.

BAPATLA WOMEN'S ENGINEERING COLLEGE::BAPATLA
AN ISO 9001-2015 CERTIFIED INSTITUTION
DEPARTMENT OF ELECTRONICS AND COMMUNICATION NGINEERING

LIST OF EXPERIMENTS

S.No.	Name of the Experiment	Pg. No.
1.	Write a program to generate the discrete sequences (i) unit step (ii) unit impulse (iii) ramp (iv) periodic sinusoidal sequences. Plot all the sequences.	16
2.	Write a MATLAB program to perform addition, subtraction, multiplication, amplitude-scaling, time scaling and time-shifting on a given signal.	22
3.	Calculate the Energy and Power of a given Signal	33
4.	Write a program to find the trigonometric and exponential Fourier series coefficients of a rectangular periodic signal. Reconstruct the signal by combining the Fourier series coefficients with appropriate weightings.	36
5.	Write a MATLAB program to determine the Fourier transform of a given input signal and also plot its magnitude and phase response plots.	43
6.	Write a MATLAB program to determine the Inverse Fourier transform of a given signal and also plot its magnitude and phase response plots.	46
7.	Verification of Linearity and Time Invariance Properties of a Given Continuous/Discrete System	51
8.	Convolve the given two discrete time sequences. Plot all the sequences. Verify the result by analytical calculation.	55
9.	Program to find frequency response of analog LP/HP/BP/BS filters	61
10.	Sampling Theorem Verification.	70

11.	Write a program to find the autocorrelation and cross correlation of sequences.	75
12.	The signal X(t) is defined as below. The signal is sampled at a sampling rate of 1000 samples per second. Find the power content and power spectral density for the signal. X(t)=cos(2pi*47t)+cos(2pi*219t), 0<t<10 X(t)=0, otherwise	84
13.	Write a MATLAB program to perform waveform synthesis using Laplace Transform of a given signal.	86
14.	Write a MATLAB program to determine the Z-transform and plot the pole-zero locations in Z-plane.	89
15.	Program to find the impulse response of the system defined by a difference equation.	92

EXPERIMENTSBEYOND THE SYLLABUS

S.No.	Name of the Experiment	Pg. No.
1	GIBBS PHENOMENON	95
2	Gaussian Noise	98

COURSE OBJECTIVES:

1. To provide background and fundamentals of MATLAB tool for the analysis and processing of signals and to generate various continuous and discrete time signals.

2. To provide an overview of signal transmission through linear systems, convolution and correlation of signals and sampling.

3. To understand the concept and importance of Fourier and Z-Transforms

COURSE OUTCOMES

At the end of this course, the student will be able to

CO No.	Course Outcome Statement	Bloom's Taxonomy Level
C253.1	**Understand** the basic operations of MATLAB	**Understand**
C253.2	**Analysis** of time domain and frequency domain signals	**Analyze**
C312.3	**Implement** the concepts of Fourier Series and Fourier Transforms	**Analyze**
C312.4	**Determine** cross correlation, autocorrelation of sequence & impulse response, step response of a system	**Apply**
C312.5	**Verify** sampling theorem	**Evaluate**

INTRODUCTION

MATLAB is a 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 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.

MATLAB is an interactive system whose basic data element is an **array** that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or Fortran. The name MATLAB stands for **matrix laboratory**.

Starting MATLAB on Windows platforms, start MATLAB by double-clicking the MATLAB shortcut icon on your Windows desktop.

Quitting MATLAB: To end your MATLAB session, select File > Exit MATLAB in the desktop, or type quit in the Command Window. You can run a script file named finish.m each time MATLAB quits that, for example, executes functions to save the workspace.

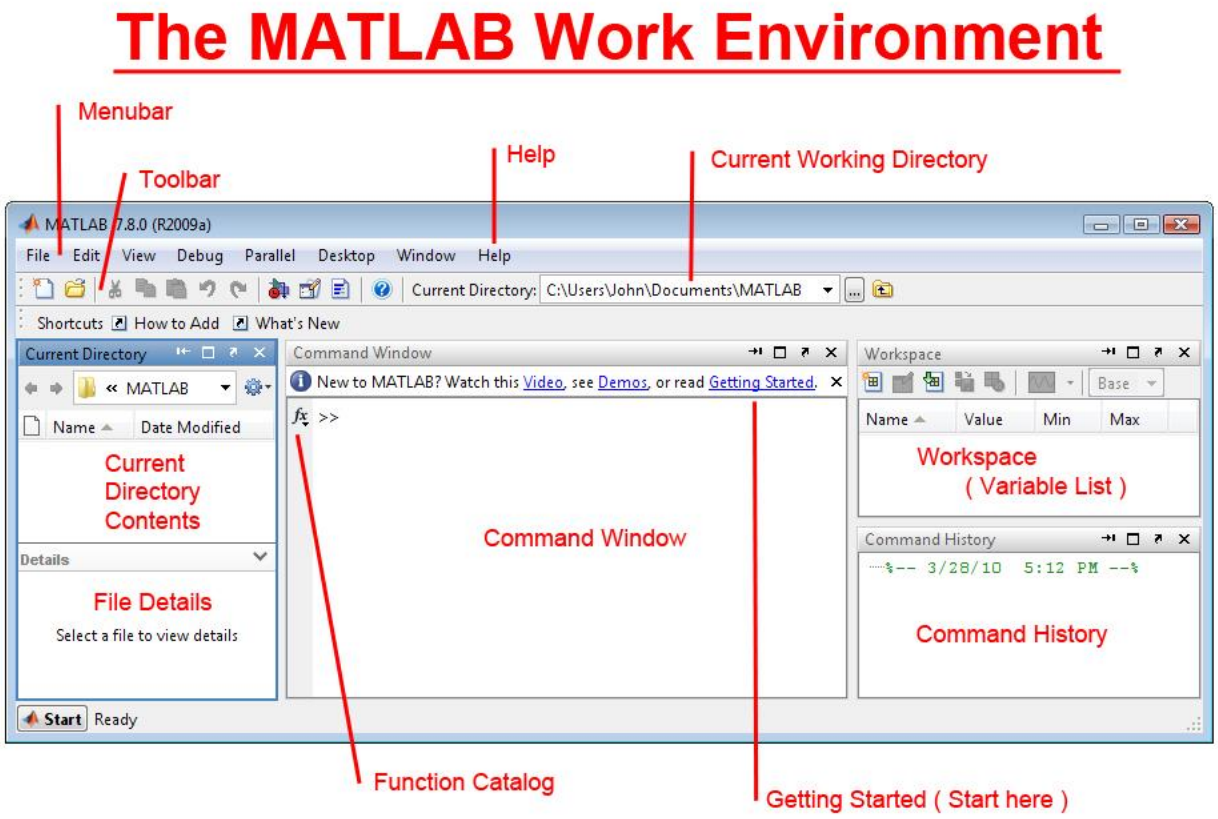
The main features of MATLAB

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

MATLAB Desktop:

On starting MATLAB the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files , variables and applications associated with MATLAB. It contains :



(i) Command Window and Command History

Command Window : Use the Command Window to enter variables and to run functions and M-file scripts. Press the up arrow key to recall a statement previously typed. Edit the statement as needed and then press Enter to run it.

Command History: Statements entered in the Command Window are logged in the Command History. From the Command History, previously run

statements can be viewed, copied and executed. M-file can be created from selected statements.

(ii) Current Directory Browser and Search Path : MATLAB file operations use the current directory and the search path as reference points.Any file required to run must either be in the current directory or on the search path.

(iii) Workspace Browser and Array Editor : The MATLAB workspace consists of the set of variables (named arrays) built up during aMATLAB session and stored in memory. Thevariables can be added to the workspace byusing functions, running M-files, and loading saved workspaces. To delete variables fromthe workspace, select the variables and select Edit > Delete.

Array Editor : Double-click a variable in the Workspace browser, or use open var variable name, to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of variables in the workspace.

(iv) Editor/Debugger : Editor/Debugger is used to create and debug M-files, which are programs to run MATLAB functions**.** The Editor/Debugger provides a graphicaluser interface for text editing, as wellas for M-file debugging. To create or edit an M-file use File > New or File > Open, or use the edit function.

Key Functions of MATLAB:

1. **Command Window and Workspace Management:**

Instruction	Definition/Description
who	Displays variables in the workspace
whos	Displays variables in the workspace with more details
clc	Clears command window
clear all	Clears MATLAB workspace (all variables)
close all	Close all figures
clear y / clear('y')	Deletes the variable named 'y'

X=1;	Suppresses echo
UP ↑ & DOWN ↓	Recall previously executed commands

2. Variables in MATLAB:

Instruction	Definition/Description
A=34	Integer
X=[1 2 3 4]	Vector
Y=[1 2 3;4 5 6;7 8 9]	Matrix
S='Hello'	String
R=3+2i or R=3+2j	Complex Number (Both declarations are same)
5*3/9+12	The result will be stored in ans
inf(1/0)	Infinity
pi	3.14 or π
eps	Very small number tends to zero

3. Complex Numbers:

Instruction	Definition/Description
S=3+4i or S=3+4j	Declaring a complex Number (Both declarations are same)
real(s)	Real Part of s
imag(s)	Imaginary Part of s
abs(s)	Absolute value of s: $\sqrt{3^2 + 4^2}$
angle(s)	Radian angle of s: $\tan^{-1}(\frac{4}{3})$
conj(s)	Conjugate of s: (real-j*imag)

4. Trigonometric Functions:

Instruction				
In radians	**In degrees**	**Inverse of the Original One**	**Hyperbolic Function**	**Inverse Hyperbolic Function**
Y=sin(x)	Y=sind(x)	Y=asin(x)	Y=sinh(x)	Y=asinh(x)
Y=cos(x)	Y=cosd(x)	Y=acos(x)	Y=cosh(x)	Y=acosh(x)

Y=tan(x)	Y=tand(x)	Y=atan(x)	Y=tanh(x)	Y=atanh(x)
Y=sec(x)	Y=secd(x)	Y=asec(x)	Y=sech(x)	Y=asech(x)
Y=csc(x)	Y=cscd(x)	Y=acsc(x)	Y=csch(x)	Y=acsch(x)
Y=cot(x)	Y=codt(x)	Y=acot(x)	Y=coth(x)	Y=acoth(x)

5. Exponential Functions:

Instruction	Definition/Description
x^n	x^n
exp(n)	Exponential Function
log10(x)	log(x)/log(10) [base 10 logarithm]
log2(x)	log(x)/log(2) [base 2 logarithm]
log(x)	ln [Natural Logarithm]
pow2(x)	2^x
Sqrt(x)	$\sqrt{x}$
factorial(x)	$x!$

6. Declaring Vectors and Operations on Vectors:

Instruction	Definition/Description
X=[1 5 9 -3 5]	X=[element1, element2, element3, ……]
Y=0:0.1:5	Start : step : end
Z=linspace(0,10,11)	Start, end, no. of points
Z=logpace(0,10,11)	Start, end, no. of points $(10^0, 10^1, 10^2, \ldots)$
X(7)	Displays 7^{th} element of the vector named X
X(5:9)	Displays 5^{th} to 9^{th} elements of X
X(6:end) = X(6:length(X))	Displays elements from 6^{th} to last one of X
X(1:1:6)	Displays elements 1,2,3,4,5 and 6 with step 1
X(1:3:7)	Displays elements 1,3,5 and 7 with step 2
X([1 2 4 9])	Displays 1^{st}, 2^{nd}, 4^{th} and 9^{th} elements
X'	Transpose of X, with conjugate if X is complex
X.'	The transpose of X without conjugation at all
[r,c]=size(x)	Returns: r = No. of rows of the vector x c = No. of columns of the vector x

sum(x)	Adds all the elements of the vector x
mean(x)	Mean value of x = sum(x)/length(x)
length(x)	No. of elements in the vector x
find(x==2)	Returns the locations of the elements in x which are equal to 2

7. Declaring Matrices and Operations on Matrices:

Instruction	Definition/Description
A=[1 5 9;-3 5 1;2 4 -7]	A=[row1 elements; row2 elements; ……]
B=zeros(3,4)	A 3×4 zeros matrix
C=ones(2,5)	A 2×5 ones matrix
D=eye(4)	A 4×4 identity matrix
E=rand(4)	A 4×4 random matrix ranging from 0 to 1
F=magic(4)	A 4×4 magic matrix (sum of any row = sum of any column)
a(2,3)	Displays the element of the matrix named a, in row 2 &column 3
a(2,3)=5	Sets the element in row 2 and colum 3 to 5
X=[1 2 3 4]	Vector / A 1×3 matrix
Y=[1 2 3;4 5 6;7 8 9]	A 3×3 matrix
a'	Transpose of a matrix a
inv(a)	Inverse of a
det(a)	Determinant of a
sum(a)	Sum of each column
prod(a)	Product of each column
B=a^2	Squaring the matrix a
B=a.^2	Squaring of each element in the matrix a
a*b	Multiplication of matrices
a.*b	Multiplying each element in 'a' by the one in 'b' of the same location/ (or) Array multiplication
a/b	A*inv(b)
a./b	Matrix Division
a\b	inv(a)/b
sort(a)	Sorting the vector ascending

rank(a)	Rank of the matrix

8. 2-D Plots:

Instruction	Definition/Description
Plot(a,b)	Plotting a continuous signal with a as x-axis and b as y-axis
stem(a,b)	Plotting a discrete signal with a as x-axis and b as y-axis
xlabel('time')	Print x-axis label as time
ylabel('amplitude')	Print y-axis label as amplitude
title('a')	Print title of the graph as a
hold on	The new plot doesn't replace the old one
hold off	The new plot replaces the old one
axis([-1 1 -2 4])	New axis limits for x and y respectively
Subplot (2,2,1)	Divides the figure into 2 rows and 2 columns and 1 represents no. of the figure

1 (2,2,1)	2 (2,2,2)
3 (2,2,3)	4 (2,2,4)

1 (3,1,1)
2(3,1,2)
2 (3,1,3)

1. **Write a program to generate the discrete sequences (i) unit step (ii) unit impulse (iii) ramp (iv) periodic sinusoidal sequences. Plot all the sequences.**

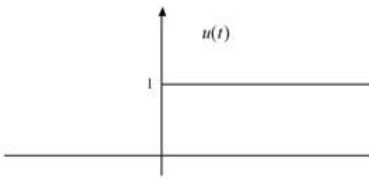
Aim: To generate Discrete sequences

Software:MATLAB R2015a(8.5.0.197613) version

Theory:A discrete signal or discrete-time signal is a time series consisting of a sequence of quantities. Unlike a continuous-time signal, a discrete-time signal is not a function of a continuous argument; however, it may have been obtained by samplingfrom a continuous-time signal, and then each value in the sequence is called a sample. When a discrete-time signal obtained by sampling a sequence corresponds to uniformly spaced times.

The unit step function, also known as the Heaviside function, is defined asthe function which has unit amplitude for the values t≥0 and zero amplitude for t<0.

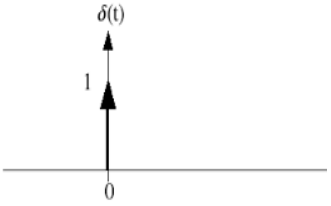
$$u(t) = \begin{cases} 1, for\ t \geq 0 \\ 0, for\ t0 \end{cases}$$



The Dirac delta function, or δ function, is
or distribution, on the real number line that is zero everywhere except at zero, with an integral of one over the entire real line

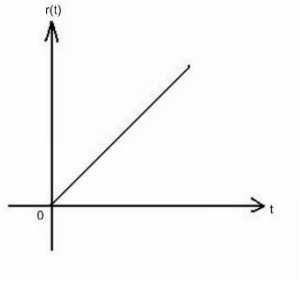
. $$\delta(t) = \begin{cases} 1, for\ t = 0 \\ 0, for\ t \neq 0 \end{cases}$$

.



The ramp function is a unary real function. It increases linearly with time for the values t≥0 and zero for t<0.

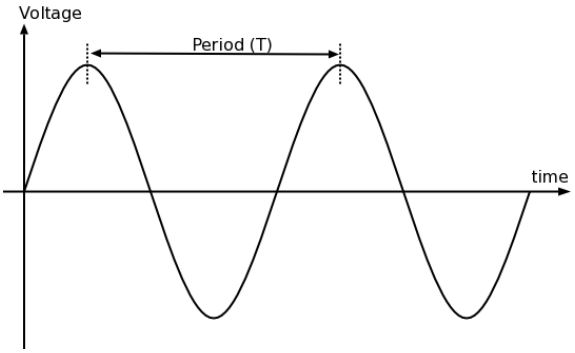
$$r(t) = \begin{cases} t, for\ t \geq 0 \\ 0, for\ t0 \end{cases}$$



A periodic function is a function that repeats its values in regular intervals or periods. The most important examples are the trigonometric functions, which repeat over intervals of 2π radians. Periodic functions are used throughout science todescribe oscillations, waves, and other phenomena that exhibit periodicity. Any function which is not periodic is called aperiodic.

The sine wave or sinusoid is a mathematical curve that describes a smooth repetitive oscillation. It is mathematically expressed as

$$y(t) = Asin(2\pi ft + \varphi)$$



Where,

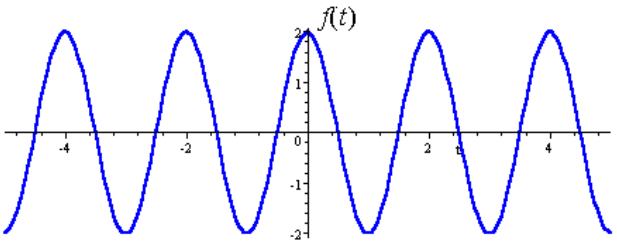
A= amplitude

f=frequency

φ= phase

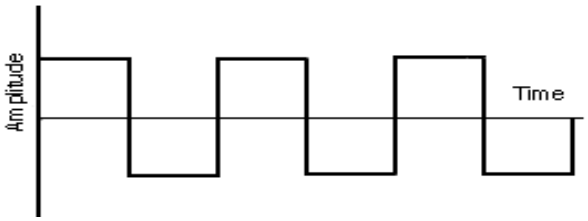
Periodic Cosine wave is mathematically expressed as:

$$y(t) = Acos(2\pi ft + \varphi)$$



Periodic square wave mathematica

$$y(t) = Asquare(2\pi ft + \varphi)$$



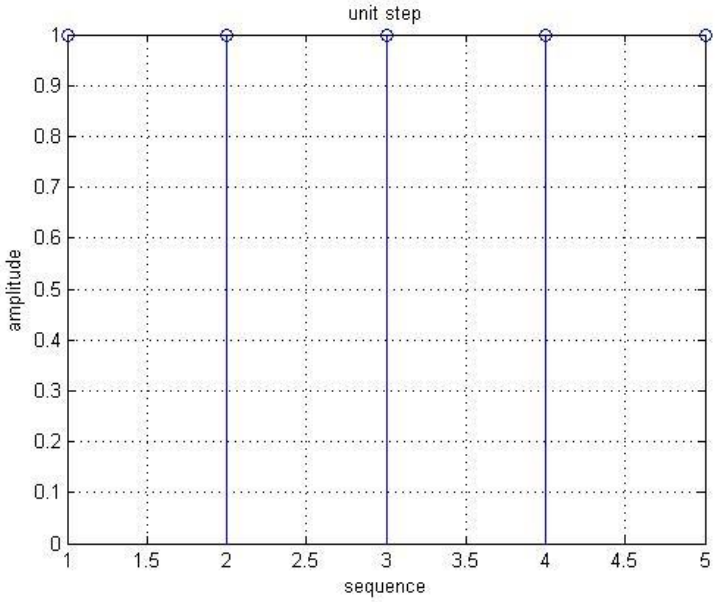
Procedure:-

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Programs:

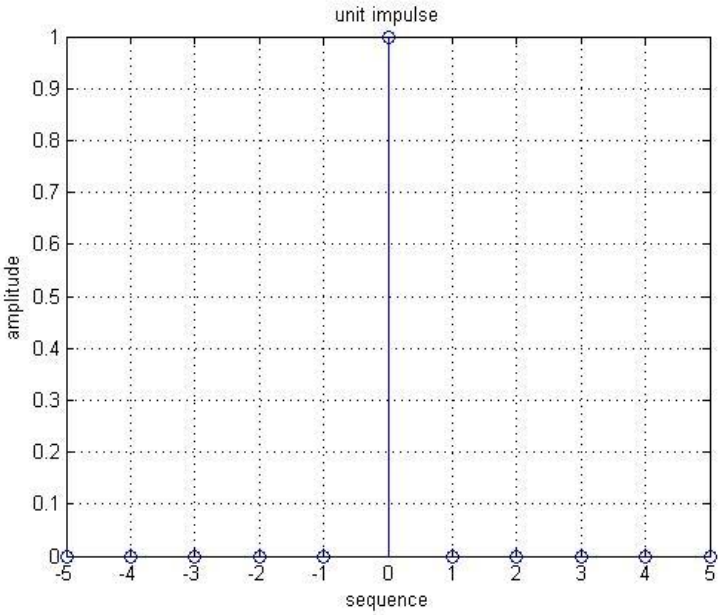
(i) Unit StepOutput:-**Enter the sequence 5**

```
clc;
close all;
clear all;
n=input('enter the sequence');
t=1:n;
u=ones(1,n);
stem(t,u);
xlabel('sequence');
ylabel('amplitude');
title('unit step');
grid on;
```



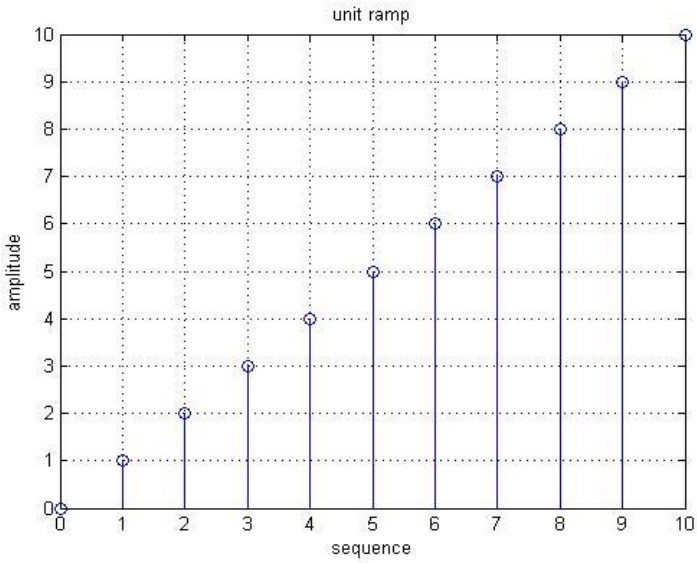
(ii) Unit impulse Output:-**Enter the sequence 5**

```
clc;
close all;
clear all;
N=input('enter the sequence');
n=-N:1:N;
x=[zeros(1,N),1,zeros(1,N)]
stem(n,x);
xlabel('sequence');
ylabel('amplitude');
title('unit impulse');
grid on;
```



(iii) RampOutput:-**Enter the sequence 5**

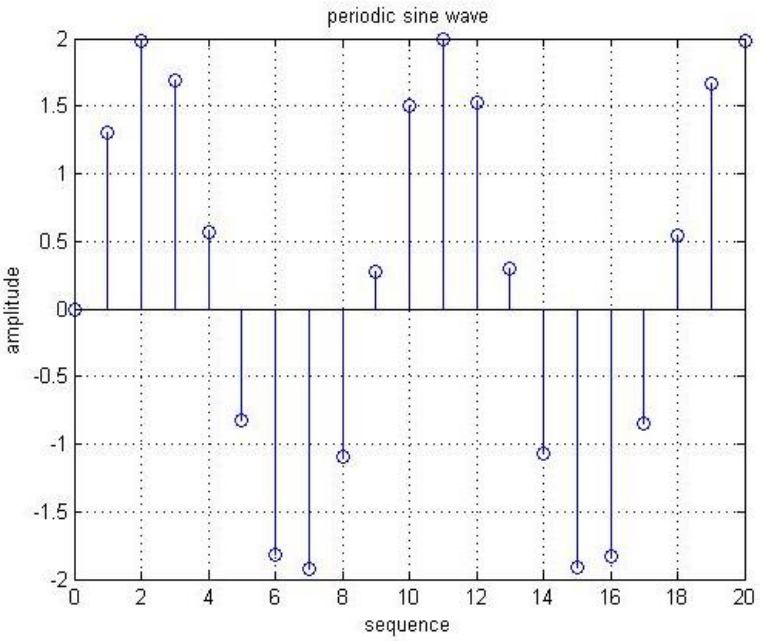
```
clc;
close all;
clear all;
n=0:1:10;
stem(n,n);
xlabel('sequence');
ylabel('amplitude');
title('unit ramp');
grid on;
```



(iv) Periodic Sinusoidal Sequences

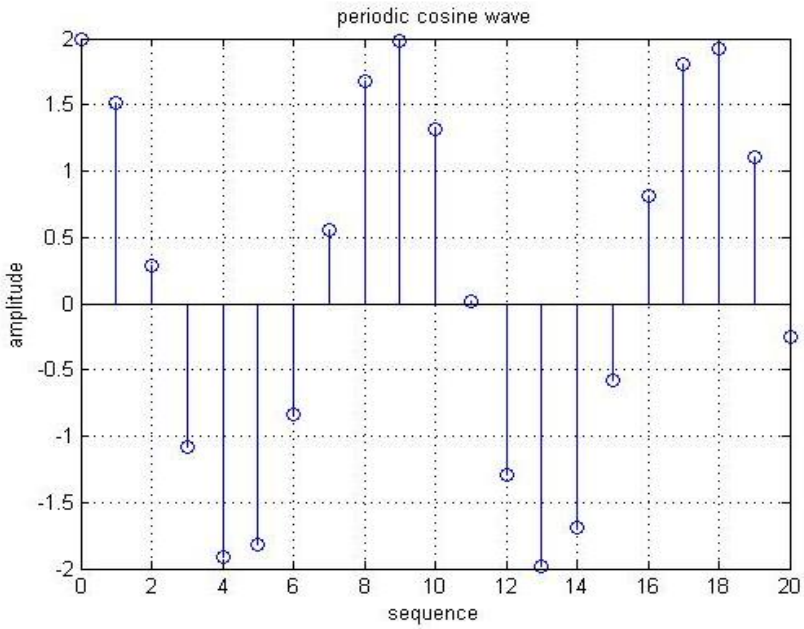
Periodic sine waveOutput:-**Enter the frequency 10000**

```
clc;Enter the amplitude 2
closeall;
clearall;
n=0:1:20;
pi=3.141;
f=input('enter the frequency');
a=input('enter the amplitude');
y=a*sin(2*pi*f*n);
stem(n,y);
xlabel('sequence');
ylabel('amplitude');
title('periodic sine wave');
gridon;
```



Cosine waveOutput:-**Enter the frequency 10000**

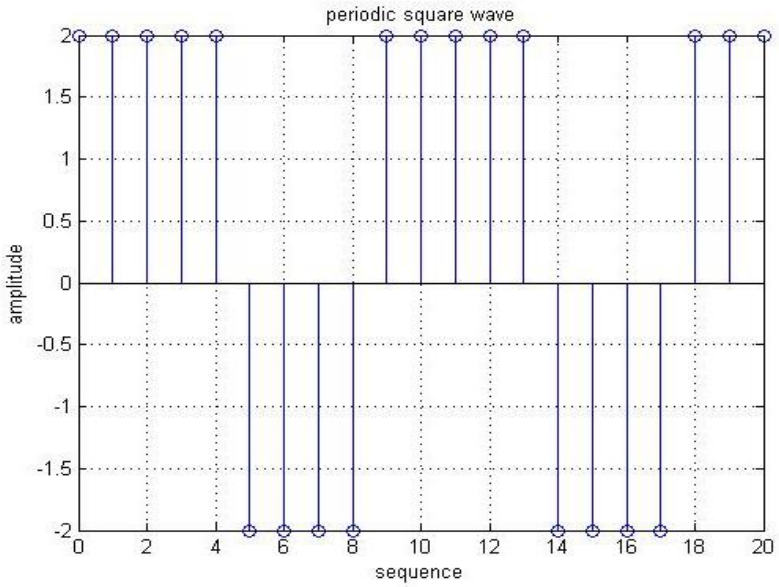
```
clc;Enter the amplitude 2
close all;
clear all;
n=0:1:20;
pi=3.141;
f=input('enter the frequency');
a=input('enter the amplitude');
y=a*cos(2*pi*f*n);
stem(n,y);
xlabel('sequence');
ylabel('amplitude');
title('periodic cosine wave');
grid on;
```



Square waveOutput:-**Enter the frequency 10000**

clc;**Enter the amplitude 2**

```
close all;
clear all;
n=0:1:10;
pi=3.141;
f=input('enter the frequency');
a=input('enter the amplitude');
y=a*square(2*pi*f*n);
stem(n,y);
xlabel('sequence');
ylabel('amplitude');
title('periodic square wave');
grid on;
```



Viva-Voce Questions:

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

2. Write a MATLAB program to perform addition, subtraction, multiplication, amplitude-scaling, time scaling and time-shifting on a given signal.

Aim: To perform addition, subtraction, multiplication, amplitude-scaling, time scaling and time-shifting of a given signal

Software:MATLAB R2015a(8.5.0.197613)version

Theory:

A signal, comprises of a set of information expressed as a function of any number of independent variables, that can be given as an input to a system, or derived as output from the system.

The mathematical transformation from one signal to another can be expressed as

$$\boldsymbol{Y(t) = T[X(t)]}$$

Where, Y(t) represents the modified signal derived from the original signal X(t), having only one independent variable t.

The **basic set of signal operations** can be broadly classified as below.

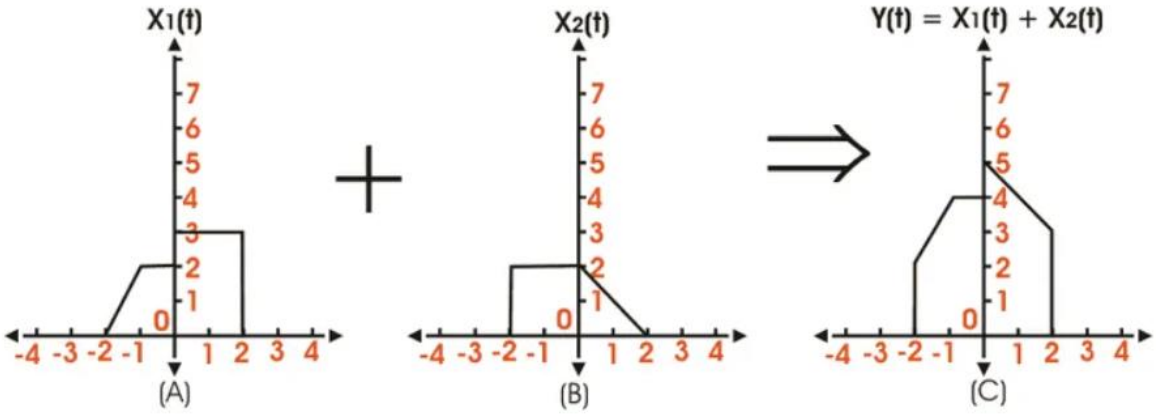
1. Addition of signals'
2. Subtraction of Signals
3. Multiplication of Signals
4. Amplitude-Scaling
5. Time-Scaling
6. Time-Shifting

Addition of Signals:

This particular operation involves the addition of amplitude of two or more signals at each instance of time or any other independent variables which are common between the signals. Addition of signals is illustrated in the diagram below, where $X_1(t)$ and $X_2(t)$ are two time dependent signals, performing the additional operation on them we getY(t)

$$X_1(t) = \sin(2\pi ft) \& X_2(t) = \cos(2\pi ft)$$

$$\boldsymbol{Y(t)} = X_1(t) + X_2(t)$$



Subtraction of Signals:

This particular operation involves the subtraction of amplitude of two or more signals at each instance of time or any other independent variables which are common between the signals. Addition of signals is illustrated in the diagram below, where $X_1(t)$ and $X_2(t)$ are two time dependent signals, performing the additional operation on them we get Y(t)

$$X_1(t) = \sin(2\pi ft) \& X_2(t) = \cos(2\pi ft)$$

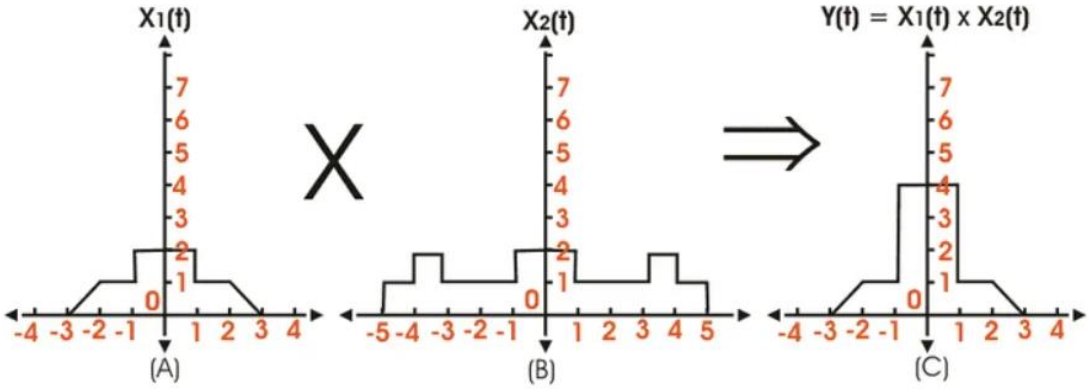
$$\boldsymbol{Y(t)} = X_1(t) - X_2(t)$$

Multiplication of Signals:

Multiplication of amplitude of two or more signals at each instance of time or any other independent variables is done which are common between the signals. The resultant signal we get has values equal to the product of amplitude of the parent signals for each instance of time. Multiplication of signals is illustrated in the diagram below, where $X_1(t)$ and $X_2(t)$ are two time dependent signals, on whom after performing the multiplication operation we get Y(t).

$$X_1(t) = \sin(2\pi ft) \& X_2(t) = \cos(2\pi ft)$$

$$\boldsymbol{Y(t)} = X_1(t) * X_2(t)$$



Amplitude Scaling:

Amplitude scaling rescales the amplitude of the signal. As a result, signal may be amplified or attenuated. Given a signal x=Asin(2πft) where A is the amplitude of the signal. We can rescale the amplitude by some constant multiplier alpha or beta. Amplitude scaled output will be

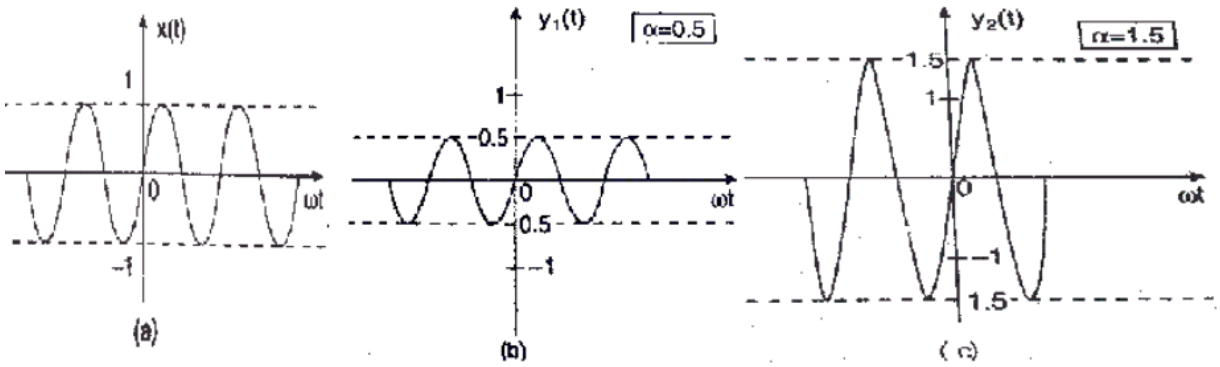
$$y = \alpha[Asin(2\pi ft)]$$

$$y = \beta[Asin(2\pi ft)]$$

$$where\ \alpha\ \&\ \beta\ are\ scaling\ factors$$

β< 1 → signal is attenuated.

α> 1 → signal is amplified.

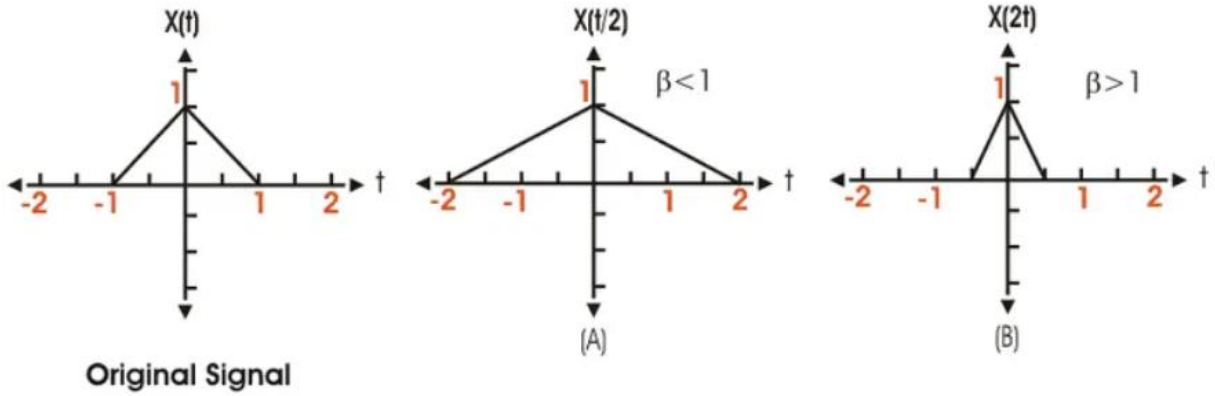


Time Scaling:

Time scaling of signals of signals involves the modification of a periodicity of the signal, keeping its amplitude constant. Its mathematically expressed as,

$$Y(t) = \beta X(t)$$

Where, X(t) is the original signal, and β is the scaling factor. If β > 1 implies, the signal is compressed and β < 1 implies, the signal is expanded. This is illustrated diagrammatically for better understanding.



Time-Shifting:

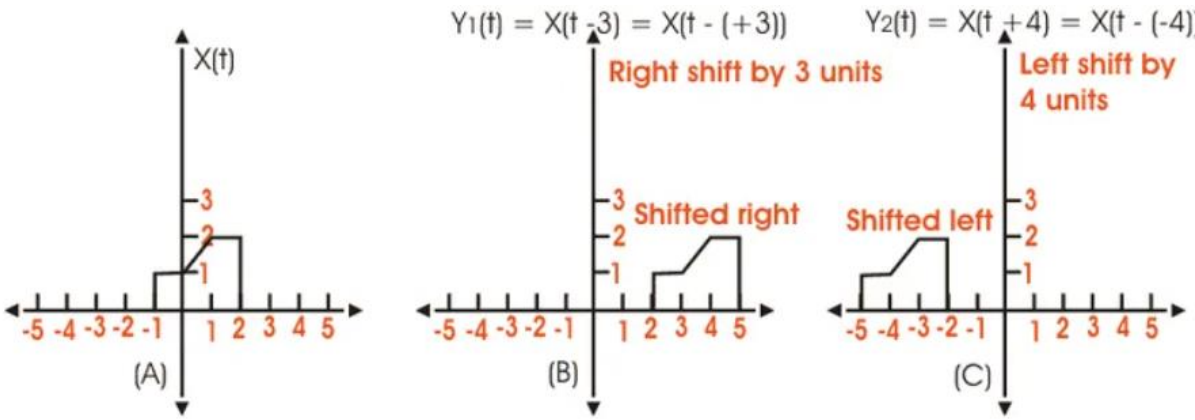
Time shifting of signals is probably the most important one, and most widely used amongst all **basic signal operations**.

It's generally used to fast-forward or delay a signal, as is necessary in most practical circumstances. Time shiftingismathematically expressed as,

$$Y(t) = X(t - t_0)$$

Where, X(t) is the original signal, and t_0 represents the shift in time.For a signal X(t) if the position shift $t_0 > 0$.

Then the signal is said to be right shiftedordelayed. In the same manner, if $t_0 < 0$, implies the signal is left shifted or delayed. This has been explained diagrammatically in the figure below. Where the original signal fig(a) is right shifted and also left shifted in figure (b) and (c) respectively.



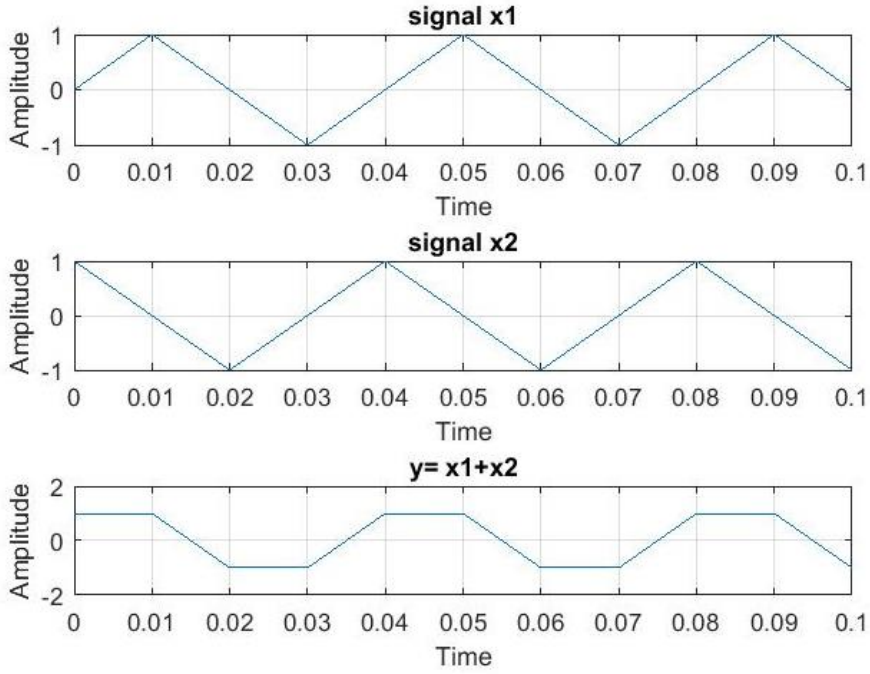
Procedure:-

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Programs:

(i) Addition of Two SignalsOutput:

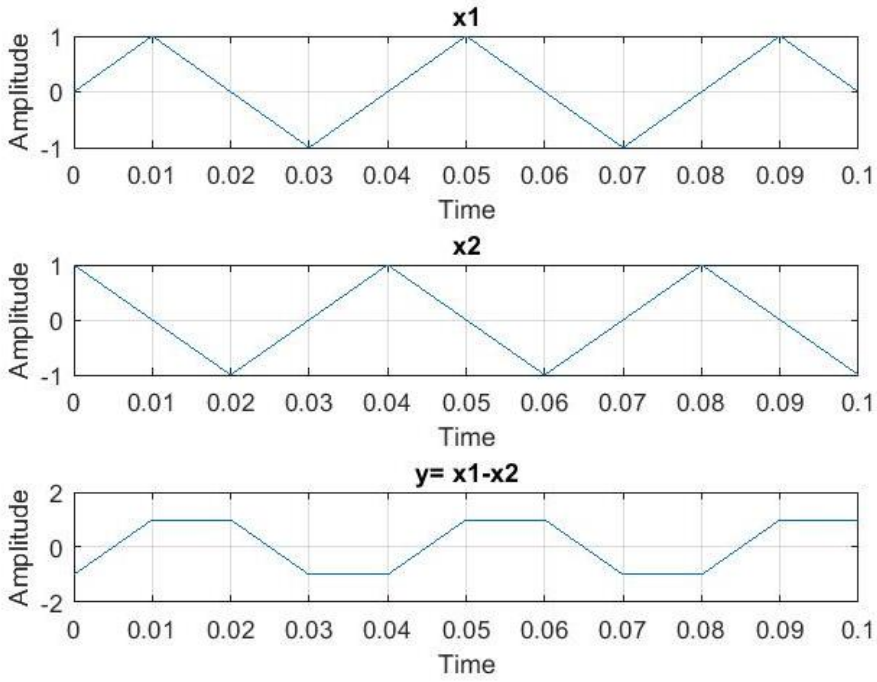
```
t=0:0.01:0.1;
f=25;
x1=sin(2*pi*f*t);
subplot(3,1,1);
plot(t,x1);
title('signal x1');
grid on;
xlabel('Time');;
ylabel('Amplitude');
x2=cos(2*pi*f*t);
subplot(3,1,2);
plot(t,x2);
grid on;
title('signal x2');
xlabel('Time');
```



```
ylabel('Amplitude');
y=x1+x2;
subplot(3,1,3);
plot(t,y);
grid on;
title('y= x1+x2');
xlabel('Time');
ylabel('Amplitude');
```

(ii) Subtraction of two signals Output:

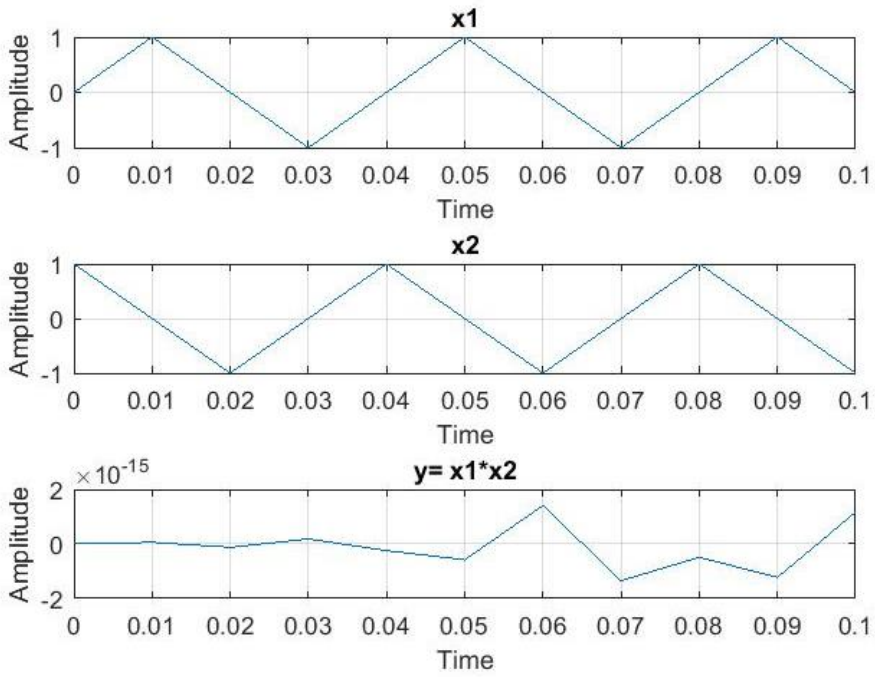
```
t=0:0.01:0.1;
f=25;
t1=2*pi*f*t;
x1=sin(2*pi*f*t);
subplot(3,1,1);
plot(t,x1);
title('x1');
xlabel('Time');
ylabel('Amplitude');
grid on;
x2=cos(2*pi*f*t);
subplot(3,1,2);
plot(t,x2);
title('x2');
xlabel('Time');
ylabel('Amplitude');
grid on;
y=x1-x2; % Here the subtraction takes place
subplot(3,1,3);
plot(t,y);
```



```
title('y= x1-x2');
xlabel('Time');
ylabel('Amplitude');
grid on;
```

(iii) Multiplication of Two Signals **Output:**

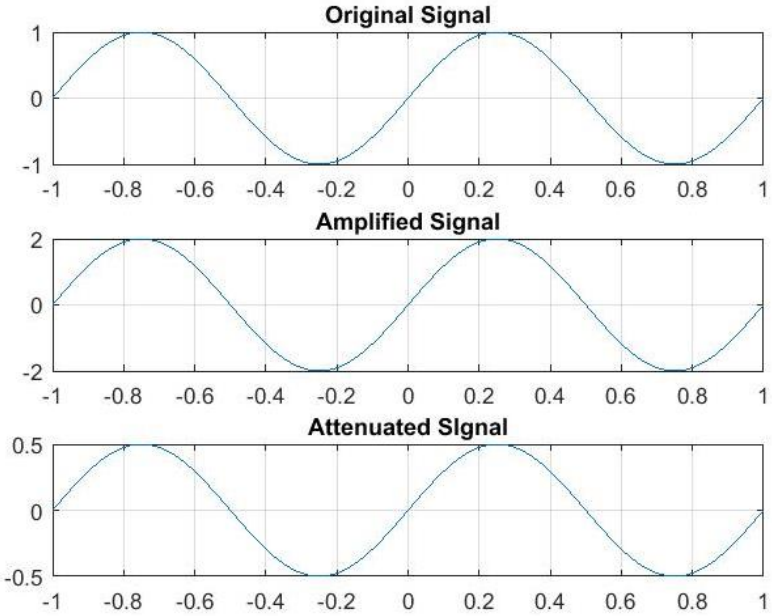
```
t=0:0.01:0.1;
f=25;
x1=sin(2*pi*f*t);
subplot(3,1,1);
plot(t,x1);
title('x1');
xlabel('Time');
ylabel('Amplitude');
grid on;
x2=cos(2*pi*f*t);
subplot(3,1,2);
plot(t,x2);
title('x2');
xlabel('Time');
ylabel('Amplitude');
grid on;
y=x1.*x2;% Here the multiplication takes place
subplot(3,1,3);
plot(t,y);
title('y= x1*x2');
xlabel('Time');
ylabel('Amplitude');
grid on;
```



(iv) Amplitude Scaling:

```
t=-1:0.01:1;
f=1;
a=1; % Amplitude
alpha=2; % Amplitude scale
beta=0.5; %Attenuated scale
x=a*sin(2*pi*f*t);
y=alpha*sin(2*pi*f*t);
z=beta*sin(2*pi*f*t);
subplot(3,1,1);
plot(t,x);
grid on;
title('Original Signal');
subplot(3,1,2);
plot(t,y);
grid on;
title('Amplified Signal');
subplot(3,1,3);
plot(t,z);
grid on;
title('Attenuated SIgnal');
```

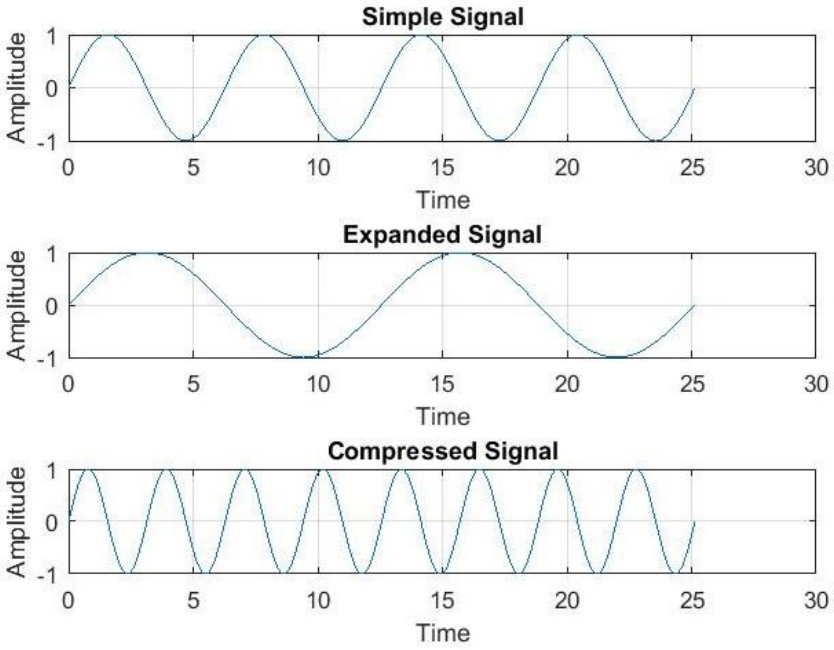
Output:



(v) Time Scaling:

```
t=0:0.01:8*pi;
x=sin(t);
subplot(3,1,1);
plot(t,x);
title('Simple Signal');
xlabel('Time');
ylabel('Amplitude');
grid on;
y=sin(t/2);
subplot(3,1,2);
plot(t,y);
title('Expanded Signal');
xlabel('Time');
ylabel('Amplitude');
grid on;
z=sin(t*2);
subplot(3,1,3);
plot(t,z);
grid on;
title('Compressed Signal');
xlabel('Time');
ylabel('Amplitude');
```

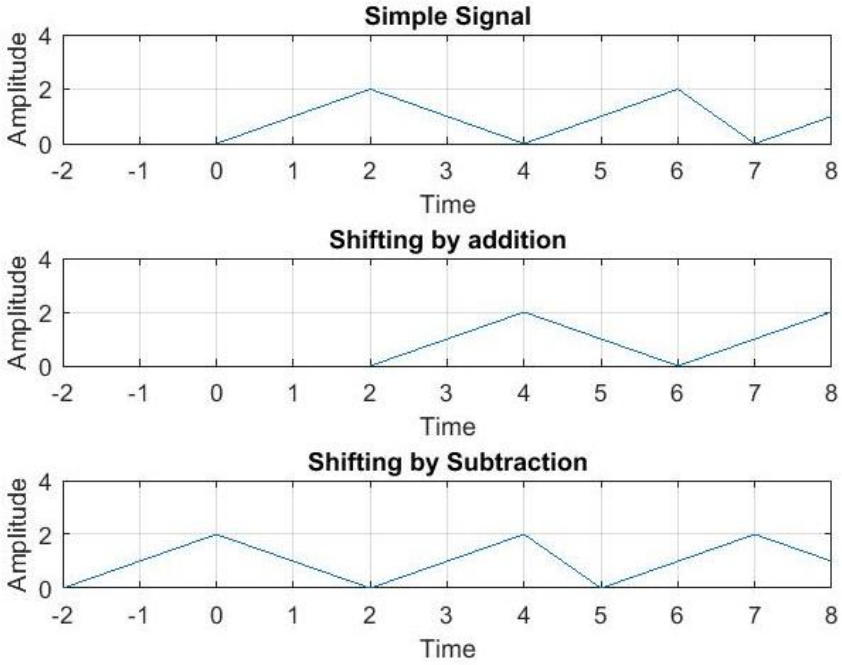
Output:



(vi) Time Shifting:

```
t=0:10;
x=[0 1 2 1 0 1 2 0 1 2 1 ];
subplot(3,1,1);
plot(t,x);
title('Simple Signal');
xlabel('Time');
ylabel('Amplitude');
grid on;
axis([-2 8 0 4]);
subplot(3,1,2);
plot(t+2,x);
grid on;
title('Shifting by addition');
xlabel('Time');;
ylabel('Amplitude');
grid on;
axis([-2 8 0 4]);
subplot(3,1,3);
plot(t-2,x);
grid on;
title('Shifting by Subtraction');
xlabel('Time');
ylabel('Amplitude')
grid on;
axis([-2 8 0 4]);
```

Output:



Viva-Voce Questions:

1. Define Symmetric and Anti-Symmetric Signals?
2. Define Continuous and Discrete Time Signals?
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?

3. Calculate the Energy and Power of a given Signal

Aim: To calculate the energy and power of a signal

Software:MATLAB R2015a(8.5.0.197613)version

Theory:A signal can be categorized into energy signal or power signal: An **energy signal** has a finite energy, 0 < E < ∞. In other words, energy signals have values only in the limited time duration. For example, a signal having only one square pulse is energy signal. A signal that decays exponentially has finite energy, so, it is also an energy signal. The power of an energy signal is 0, because of dividing finite energy by infinite time (or length).

$$E(t) = \lim_{N\to\infty} \int_{-N}^{N} |x(t)|^2 dt = \int_{-\infty}^{\infty} |x(t)|^2 dt$$

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

On the contrary, the **power signal** is not limited in time. It always exists from beginning to end and it never ends. For example, sine wave in infinite length is power signal. Since the energy of a power signal is infinite, it has no meaning to us. Thus, we use power (energy per given time) for power signal, because the power of power signal is finite, 0 < P < ∞.

$$P(t) = \lim_{N\to\infty} \frac{1}{2N} \int_{-N}^{N} |x(t)|^2 dt$$

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

Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.

5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Programs:

Energy of a signal:

```
t=0:pi:10*pi;
f=50;
a=2;
z=a*cos(2*pi*f*t);
e=sum(abs(z).^2);
disp('The energy of a signal is');
disp(e);
```

Power of a signal:

```
t=0:pi:10*pi;
f=50;
a=2;
z=a*cos(2*pi*f*t);
p=(sum(abs(z).^2))/length(z);
disp('the power of a signal is');
disp(p);
```

Output:

The energy of a signal is e=21.1591

The power of a signal is p= 1.9236

Viva-Voce Questions:

1. What is the difference between energy and power in the context of signal processing? How do you calculate them in MATLAB?

2. How do you define the energy of a discrete-time signal? What is the unit of energy in signal processing?

3. What is the relationship between the energy of a signal and its duration? Can you explain with an example?

4. How do you calculate the power of a signal in MATLAB? What is the unit of power in signal processing?

5. Consider a signal x[n] = A_sin(w_n). How would you calculate its energy and power in MATLAB? What would be the effect of changing the amplitude 'A' on the energy and power?

4. Write a program to find the trigonometric and exponential fourier series coefficients of a rectangular periodic signal. Reconstruct the signal by combining the Fourier series coefficients with appropriate weightings

Aim: To find the trigonometric and exponential fourier series coefficients of a rectangular periodic signal. Reconstruct the signal by combining the Fourier series coefficients with appropriate weightings

Software: MATLAB R2015a(8.5.0.197613)version

Theory:

Trigonometric Fourier Series:

A periodic signal can be represented over a certain interval of time in terms of the linear combination of orthogonal functions. If these orthogonal functions are trigonometric functions, then the Fourier series representation is known as ***trigonometric Fourier series***.

Consider a sinusoidal signal $f(t) = Asinw_0t$, which is periodic with time period$T$ such that $T = \frac{2\pi}{w_0}$. If the frequencies of two sinusoids are integral multiples of a fundamental frequency w_0, then the sum of these two sinusoids is also periodic.

A signal $f(t)$, represented by a sum of sine and cosine functions whose frequencies are integral multiples of the fundamental frequency $w_0$is a periodic signal.

Let the signal $f(t)$ is given by

$$f(t) = a_0 + a_1 cosw_0t + a_2 cos2w_0t + a_3 cos3w_0t + \cdots + a_k coskw_0t + b_1 sinw_0t$$
$$+ b_2 sin2w_0t + b_3 sin3w_0t + \cdots + b_k sinkw_0t$$

The trigonometric fourier series representation of a periodic signal $f(t)$ with infinite sum of sine and cosine functions with frequencies $0, w_0, 2w_0, 3w_0, \ldots kw_0$ is given by

$$f(t) = a_0 + \sum_{n=1}^{\infty} [a_n cosnw_0 t + b_n sinnw_0 t]$$

Where, $a_0, a_n \& b_n$ are trigonometric fourier series coefficients

$$a_0 = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)\ dt$$

$$a_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\, cos\ nw_0 t\ dt$$

$$b_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\ sin\ nw_0 t\ dt$$

The coefficient a_0 is known as DC component

$(a_1 cosw_0 t + b_1 sinw_0 t)$ is called first harmonic term

$(a_2 cos\ 2w_0 t + b_2 sin\ 2w_0 t)$ is called second harmonic term

$(a_n cos\ nw_0 t + b_n sin\ nw_0 t)$ is called n^{th} harmonic term

Exponential Fourier Series:

A periodic signal can be represented over a certain interval of time in terms of the linear combination of orthogonal functions. If these orthogonal functions are exponential functions, then it is called the exponential Fourier series

For any periodic signal $f(t)$, the exponential form of Fourier series is given by,

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{jnw_0 t}$$

Where $w_0 = \frac{2\pi}{T}$ *is the angular frequency of the periodic function*

The exponential fourier series coefficient is given by

$$C_n = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)\, e^{-jnw_0 t} \mathrm{dt}$$

The exponential Fourier series coefficients of a periodic function x(t) have only a discrete spectrum because the values of the coefficient Cn exists only for discrete values of n. As the exponential Fourier series represents a complex spectrum, thus, it has both magnitude and phase spectra.

Procedure:

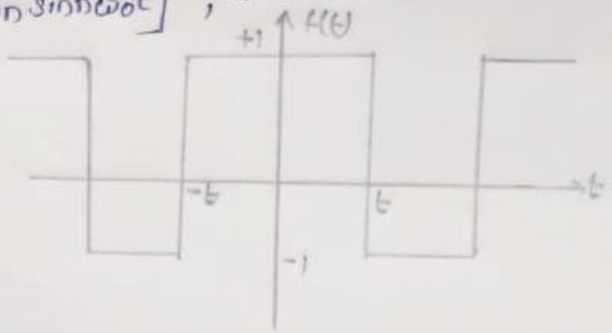
1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Calculations:

Trignometric fourier Series

Theoritical Calculation

$f(t) = a_0 + \sum_{n=1}^{\infty} [a_n \cos n\omega_0 t + b_n \sin n\omega_0 t]$, $t_0 < t < t_0 + T$



TFS Coefficient are a_0, a_n, b_n

$a_0 = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)\,dt$

$a_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\cos n\omega_0 t\,dt$

$b_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\sin n\omega_0 t\,dt$.

To find a_0: $a_0 = \frac{1}{T}\int_{-t}^{t} f(t)\,dt$

$= \frac{1}{T}\int_{-t}^{t} (1)\,dt$

$= \frac{1}{T}[t]_{-t}^{t} = \frac{1}{T}[t+t] = \frac{2t}{T}$

$\boxed{a_0 = \frac{2t}{T}}$

To find a_n: $\frac{2}{T}\int_{t_0}^{t_0+T} f(t)\cos n\omega_0 t\,dt$

$= \frac{2}{T}\int_{-t}^{t} (1)\cos n\omega_0 t\,dt$

$= \frac{2}{T}\left[\frac{\sin n\omega_0 t}{n\omega_0}\right]_{-t}^{t}$

$= \frac{2}{n\omega_0 T}\left[\sin n\omega_0 t - \sin n\omega_0(-t)\right]$

$= \frac{2}{n\omega_0 T}\left[\sin n\omega_0 t + \sin n\omega_0 t\right]$

$= \frac{2}{n\omega_0 T}\left[2\sin n\omega_0 t\right]$

$= \frac{2}{n(\frac{2\pi}{T})T}\left[2\sin n\omega_0 t\right] = \frac{2}{n\pi}\left[\sin\left(\frac{2\pi}{T}\right)t\right]$

$\boxed{a_n = \frac{2}{n\pi}\left[\sin\left(\frac{2\pi n}{T}\right)t\right]}$

To find b_n: $b_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\sin n\omega_0 t\,dt$

$= \frac{2}{T}\int_{-t}^{t} (1)\sin n\omega_0 t\,dt = \frac{2}{T}\left[\frac{-\cos n\omega_0 t}{n\omega_0}\right]_{-t}^{t}$

$= \frac{2}{T}\left[\frac{-\cos n\omega_0 t}{n\omega_0} + \frac{\cos n\omega_0(-t)}{n\omega_0}\right]$

$= \frac{2}{T}\left[\frac{-\cos n\omega_0 t}{n\omega_0} + \frac{\cos n\omega_0 t}{n\omega_0}\right]$

$\boxed{b_n = 0}$

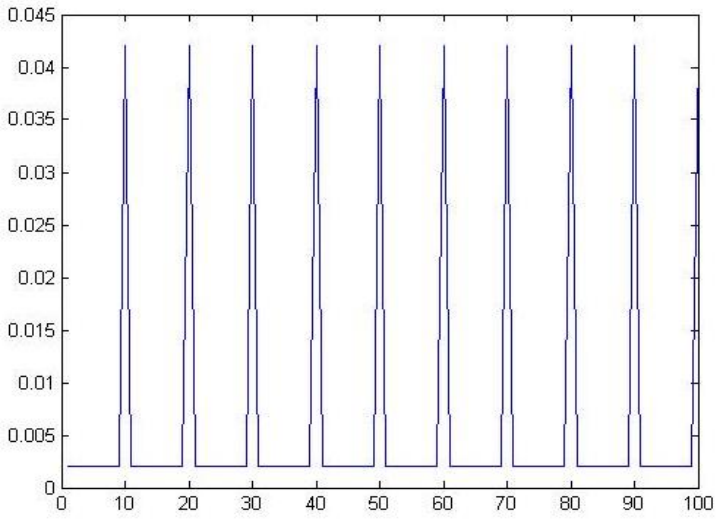
Now

$f(t) = \frac{2t}{T} + \sum_{n=1}^{\infty} \frac{2}{n\pi} \sin\left(\frac{2\pi n}{T}\right)t \cos n\omega_0 t + 0$

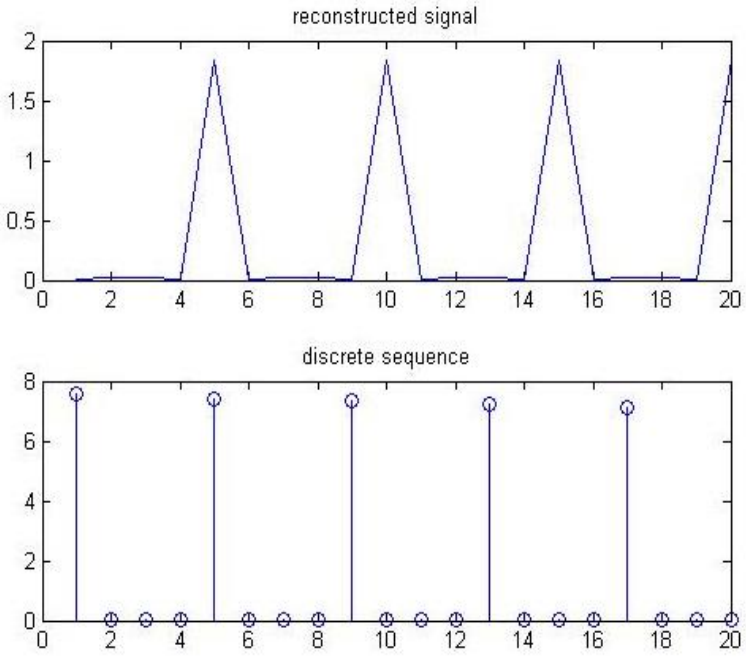
Program:

```
%Trignometric Fourier Series coefficients
clc;
clear all;
close all;
T=10;
n=10;
t=0.01;
w=(2*pi/T);
fori=1:n
a(i)=2*sin(2*pi*i*t/T)./(pi*i);
end;
disp('trignometric coefficients');
a0=(2*t/T);
%signal reconstruction
k=1:100
for j=1:length(k);
f(j)=a0;
fori=1:n
f(j)=f(j)+a(i)*cos(i*w*k(j));
end;
end;
plot(k,f);
title('Trignometric Fourier Series');
```

Output:



```
% Exponential Fourier Series coefficients
clc;
clear all;
close all;
T=5;
n=10;
t=0.08;
w=(2*pi/T);
fori=1:n
f(i)=(2*t*sinc(2*i*t/T)/T);
f1(i)=(2*t*sinc(2*(-i)*t/T));
end;
f0=(2*t/T);
j=sqrt(-1);
k=1:20;
for l=1:length(k)
y(l)=f0;
for m=1:n
y(l)=y(l)+f(m)*exp(-j*m*w*k(l));
end;
for m=n+1:2*n;
y(l)=y(l)+f1(m-n)*exp(j*(m-n)*w*k(l));
end;
end;
F=fft(y);
subplot(2,1,1);
plot(k,y);
title('reconstructed signal');
subplot(2,1,2);
stem(abs(F));
title('discrete sequence');
```



Viva-Voce Questions:

1. What is the Fourier series, and what are its coefficients?

2. How do you calculate the trigonometric Fourier series coefficients of a rectangular signal?

3. What is the difference between trigonometric and exponential Fourier series coefficients?

4. How do you reconstruct a signal from its Fourier series coefficients?

5. What is the purpose of weighting the Fourier series coefficients in signal reconstruction?

5. Write a MATLAB program to determine the Fourier transform of a given input signal and also plot its magnitude and phase response plots.

Aim: To find the Fourier transform of a square pulse .Plot its amplitude and phase spectrum.

Software: MATLAB R2015a(8.5.0.197613) version

Theory:

The Fourier Transform is a mathematical technique that transforms a function of time, $f(t)$, to a function of frequency, $F(\omega)$. It is closely related to the Fourier Series. The generalization of the complex Fourier series is known as the Fourier transform. The Fourier transform helps to extend the Fourier series to the non-periodic functions, which helps us to view any functions in terms of the sum of simple sinusoids.

Fourier transform of a signal $f(t)$ is given by

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

where, $\omega = 2\pi f$

F is a function of real variable ω and the function $F(\omega)$ is a complex number

$|F(\omega)|$is called the amplitude spectrum of $f(t)$ and

$\angle F(\omega)$is the phase spectrum of $f(t)$

Fourier transform is used in a wide range of applications, such as:

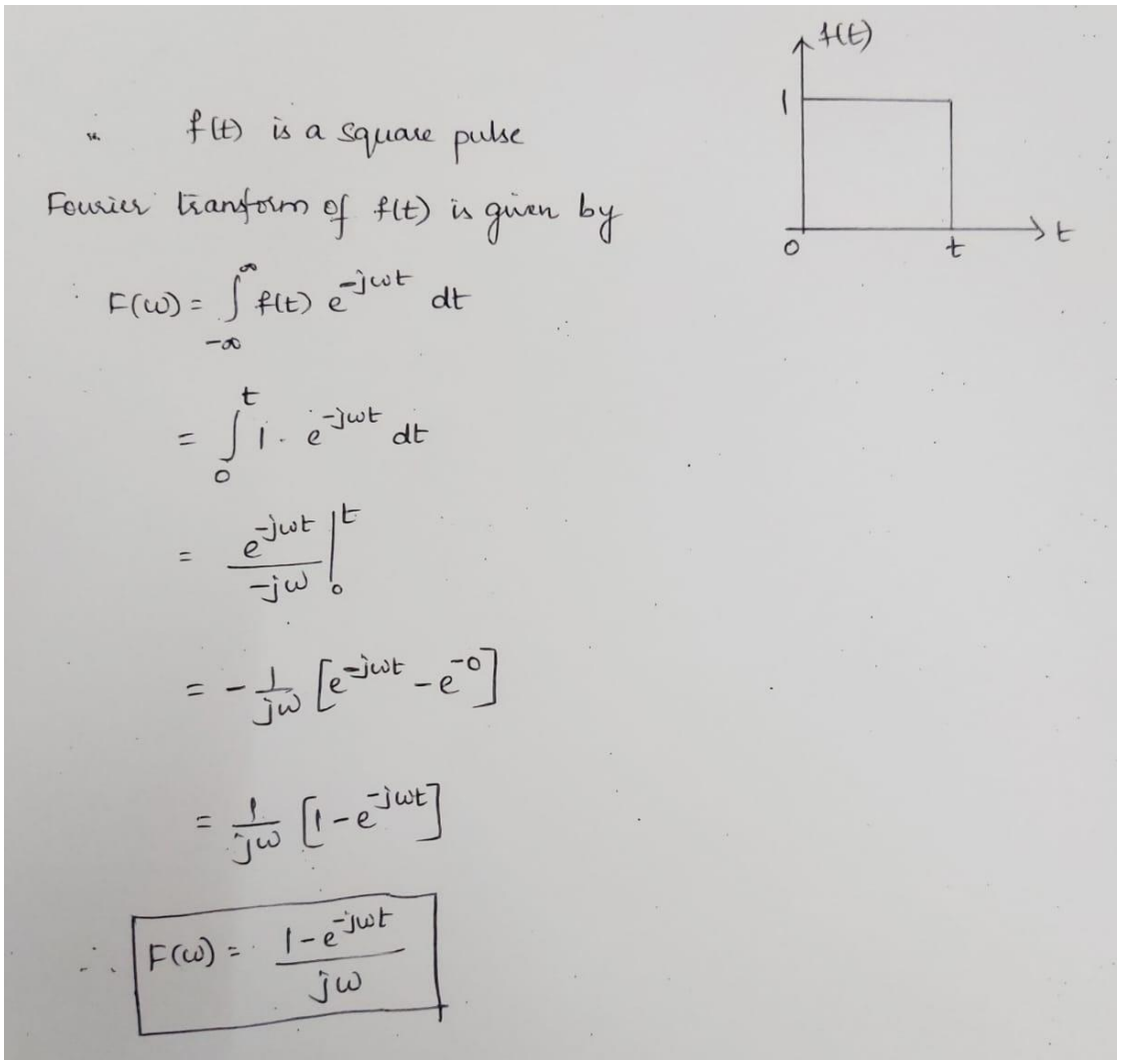
- Image Compression
- Image Analysis
- Image Filtering
- Image Reconstruction

Fourier transform is one of the important concepts used in image processing, which helps to decompose the image into the sine and cosine components.

Procedure:-

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Theoretical Calculations:



$\therefore$ f(t) is a square pulse

Fourier transform of f(t) is given by

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

$$= \int_{0}^{t} 1 \cdot e^{-j\omega t}\, dt$$

$$= \frac{e^{-j\omega t}}{-j\omega}\Big|_{0}^{t}$$

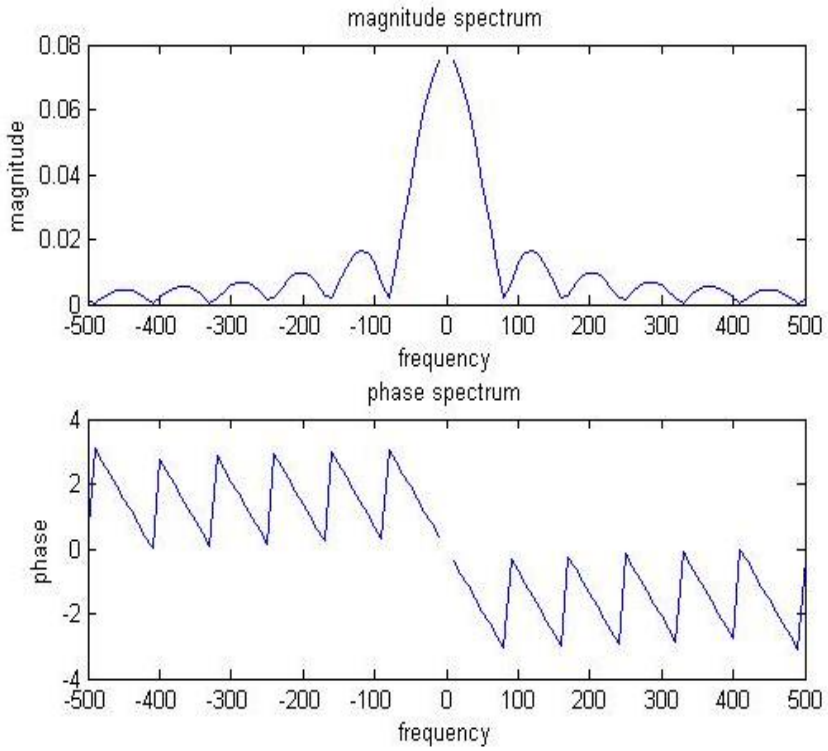
$$= -\frac{1}{j\omega}\left[e^{-j\omega t} - e^{-0}\right]$$

$$= \frac{1}{j\omega}\left[1 - e^{-j\omega t}\right]$$

$$\therefore F(\omega) = \frac{1 - e^{-j\omega t}}{j\omega}$$

Program: **Output:**

```
clc;
clearall;
closeall;
w=-500:10:500;
t=0.08/0.06;
j=sqrt(-1);
a=(1-exp(-j*w*t))./(j*w);
b=abs(a);
subplot(2,1,1);
plot(w,b)
xlabel('frequency');
ylabel('magnitude');
title('magnitude spectrum');
subplot(2,1,2);
p=angle(a);
plot(w,p);
xlabel('frequency');
ylabel('phase');
title('phase spectrum');
```



Viva-Voce Questions:

1. What is the Fourier transform, and what does it do?

2. How do you calculate the Fourier transform of a signal in MATLAB?

3. What is the magnitude response of a signal, and how is it plotted?

4. What is the phase response of a signal, and how is it plotted?

5. What information do the magnitude and phase response plots provide about the signal?

6. Write a MATLAB program to determine the Inverse Fourier transform of a given signal and plot its magnitude and phase response plots.

Aim: To find the Inverse Fourier Transform of a sinc function and plot its amplitude and phase spectrum.

Software:MATLAB R2015a(8.5.0.197613) version

Theory:

The inverse Fourier transform is a mathematical formula that converts a signal in the frequency domain ω to one in the time (or spatial) domain t. A time domain signal $f(t)$ isobtained by demodulating a frequency domain signal $F(\omega)$ using a special sinusoidal wave$e^{j\omega t}$across all time (from negative infinity to positive infinity) via:

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

Where j represents a square root of -1(an imaginary number);

$e^{j\omega t}$ Is a sinusoidal wave with an imaginary component as per *Euler's formula*:

$$e^{j\omega t} = \cos \omega t + j\, sin\, \omega t$$

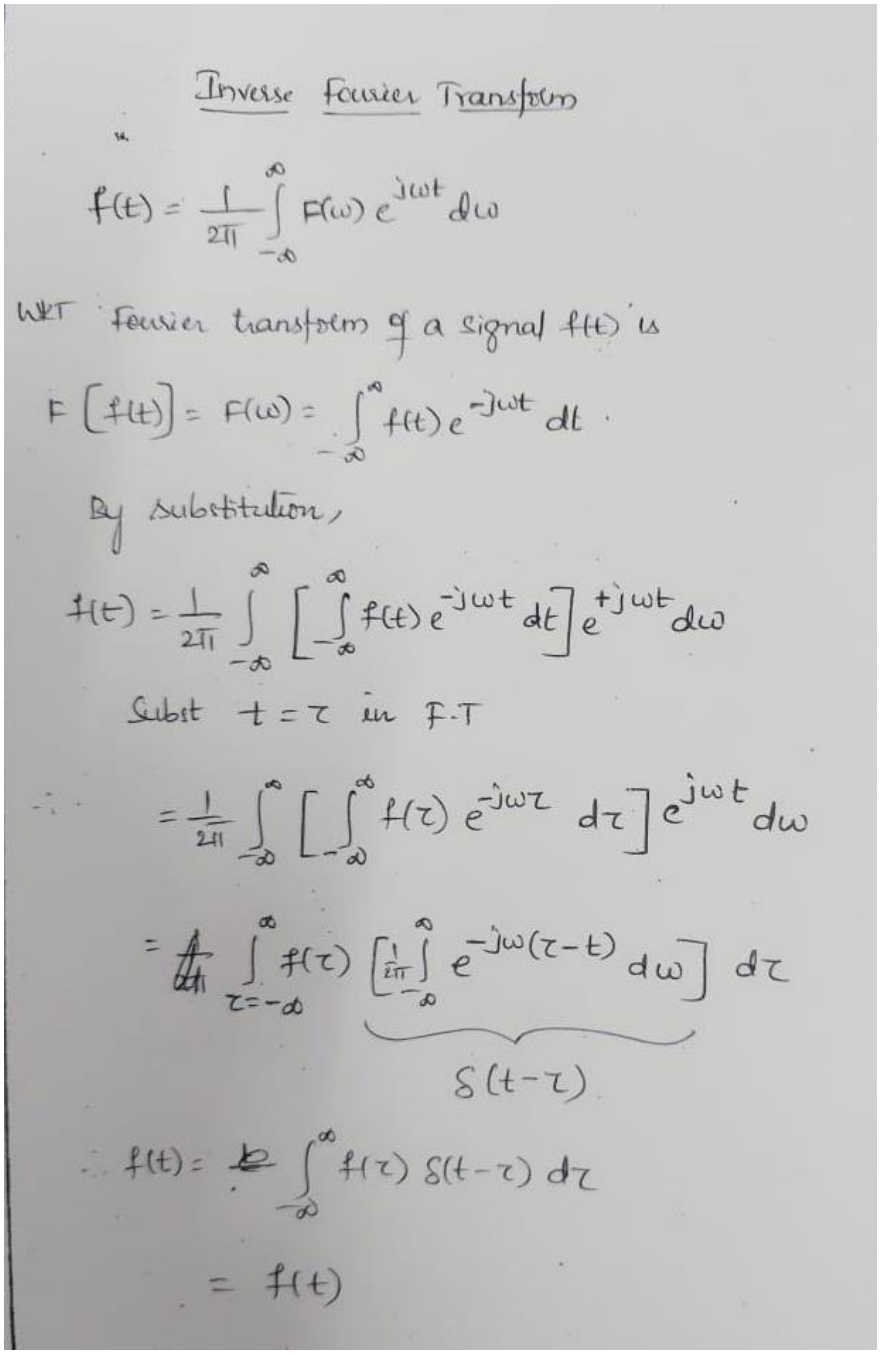
Conversely, time domain signals can be converted to frequency domain signals via the Fourier transform. So, $f(t)$ and $F(\omega)$ are called as Fourier transform pair.

Procedure:-

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.

6. **Run** the program.
7. Observe for resultant **graphs.**

Theoretical Calculations:



Inverse Fourier Transform

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

WKT Fourier transform of a signal $f(t)$ is

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

By substitution,

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt\right] e^{+j\omega t}\, d\omega$$

Subst $t = \tau$ in F.T

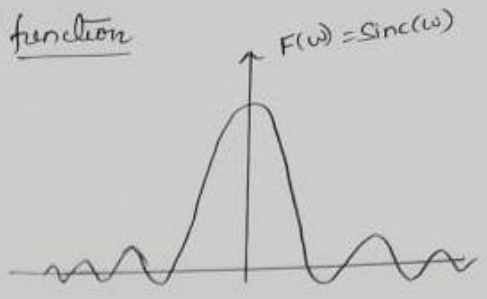
$$= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(\tau)\, e^{-j\omega\tau}\, d\tau\right] e^{j\omega t}\, d\omega$$

$$= \int_{\tau=-\infty}^{\infty} f(\tau)\underbrace{\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-j\omega(\tau-t)}\, d\omega\right]}_{\delta(t-\tau)}\, d\tau$$

$$\therefore f(t) = \int_{-\infty}^{\infty} f(\tau)\,\delta(t-\tau)\, d\tau$$

$$= f(t)$$

Inverse Fourier Transform of a 'Sinc' function



$$F(\omega) = \text{Sinc}(\omega) = \frac{\sin\omega}{\omega}$$

Inverse Fourier Transform is given by

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

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{\sin\omega}{\omega}\, e^{j\omega t}\, d\omega$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e^{j\omega} - e^{-j\omega}}{2j\omega}\, e^{j\omega t}\, d\omega \qquad \left(\because \sin\omega = \frac{e^{j\omega} - e^{-j\omega}}{2j}\right)$$

Let us consider

$$\int_{-a}^{a} e^{-j\omega\tau}\, d\tau = \frac{e^{-j\omega\tau}}{-j\omega}\Big|_{-a}^{a}$$

$$= -\frac{1}{j\omega}\left[e^{-j\omega a} - e^{j\omega a}\right]$$

$$= \frac{e^{j\omega a} - e^{-j\omega a}}{j\omega}$$

If $a=1$, $\int_{-1}^{1} e^{-j\omega\tau}\, d\tau = \frac{e^{j\omega} - e^{-j\omega}}{j\omega}$

By substituting in $f(t)$, we get

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\frac{1}{2}\int_{-1}^{1} e^{-j\omega\tau}\, d\tau\right] e^{j\omega t}\, d\omega$$

$$= \frac{1}{2}\int_{-1}^{1}\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{j\omega(t-\tau)}\, d\omega\right] d\tau$$

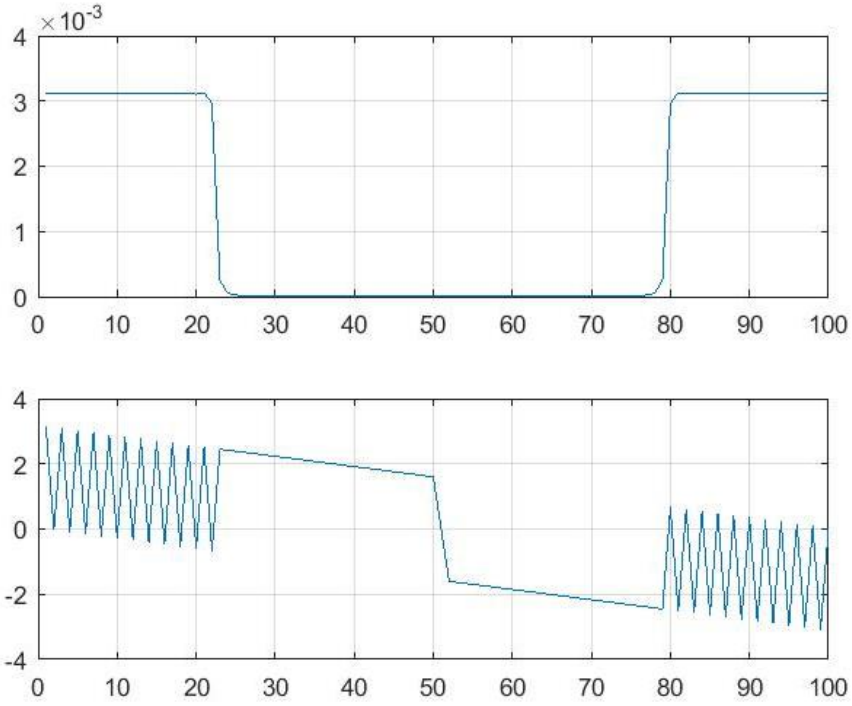
$$= \frac{1}{2}\int_{-1}^{1} \delta(t-\tau)\, d\tau \qquad \left(\because \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{j\omega(t-\tau)}\, d\omega = \delta(t-\tau)\right)$$

$\therefore f(t) = \frac{1}{2}$

Program:

```
clc;
clear all;
close all;
w=1;
T=1000;
t=linspace(-T,T,100);
s=2*sinc(w*t/pi);
z=ifft(s);
subplot(2,1,1);
plot(abs(z));
grid on;
subplot(2,1,2);
plot(angle(z));
grid on;
```

Output:



Viva-Voce Questions:

1. What is the Inverse Fourier Transform (IFT), and what does it do?

2. How do you calculate the IFT of a signal in MATLAB?

3. What is the purpose of plotting the magnitude response of the reconstructed signal?

4. How does the phase response plot of the reconstructed signal relate to the original signal?

5. What is the significance of comparing the original signal with the reconstructed signal (after IFT)?

7. Verification of Linearity and Time Invariance Properties of a Given Continuous/Discrete System

Aim: Verify the Linearity and time invariance of a given Discrete System.

Software:MATLAB R2015a(8.5.0.197613) version

Theory:

Linearity Property: Any system is said to be linear if it satisfies the superposition principalsuperposition 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

$$T[a_1x_1(n)+a_2x_2(n)]=a_1T[x_1(n)]+a_2T[x_2(n)]$$

Where, X(n)----------input signal

Y(n) --------- output signal

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

Y1(n)=T[x1(n)] : 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 [X2(n)]= Z3(n)

Let a [Y1(n)]+ b [X2(n)] =Z 3(n)

If Y3(n)- Z3(n)=0 then the system is stable otherwise it is not stable.

Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Program:

1. **Linearity Property:**

```
% 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');
```

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
```

Program (B) :

```
% 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)
ifdy==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);
ifyd==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
```

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)=n x(n)]not a time invariant

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

Viva-Voce Questions:

1. What does it mean for a system to be linear, and how do you test for it?
2. How do you verify the linearity property of a system using MATLAB?
3. What is time-invariance, and how do you test for it in a system?
4. How do you analyse a system's linearity and time-invariance properties mathematically?
5. What is the importance of linearity and time-invariance in signal processing?

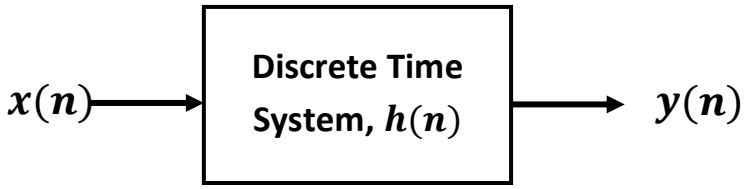
8. Convolve the given two discrete time sequences. Plot all the sequences. Verify the result by analytical calculation.

Aim: To find the auto correlation and cross correlation of sequences

Software:MATLAB R2015a(8.5.0.197613) version

Theory:

The Discrete-Time Convolution (DTC) is one of the most important operations in a discrete-time signal analysis. The operation relates the output sequence y(n) of a linear-time invariant (LTI) system, with the input sequence x(n) and the impulse response h(n), as shown in Fig.



The convolution sum of input x(n) and impulse response h(n) of an LTI system is defined as $y(n) = x(n) * h(n)$. $*$denotes convolution operation.

It is expressed mathematically as

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

(or)

$$y(n) = \sum_{k=-\infty}^{\infty} h(n+k)x(k)$$

The output of an LTI system is given by the weighted sum of time-shifted responses.

Convolution sum is an extremely powerful result to compute the system output for any input signal excitation. By using convolution, zero state response of the system can also be found.

Computation of Convolution:

Step 1: Express both the sequences x(n) and h(n) in terms of the index 'k'.

Step 2: Fold h(k) about k=0 to obtain h(-k) and shift right by 'n' if n is positive and left if n is negative to obtain h(n-k).

Step 3: Multiply the two sequences x(k) and h(n-k) element by element and sum the products to get y(n).

Step 4: Increment the index 'n' and shift the sequence h(n-k) to right by one sample and repeat step 3.

Step 5: Repeat step 4 until the sum of products is zero for all remaining values of n.

Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Theoretical Calculations:

Convolution

Method-I:

Consider two sequences $x(n)$ and $h(n)$

The linear convolution of the two sequences is $y(n) = \sum_{n=-\infty}^{\infty} x(k)h(n-k) = \sum_{n=-\infty}^{\infty} h(n+k)x(k)$

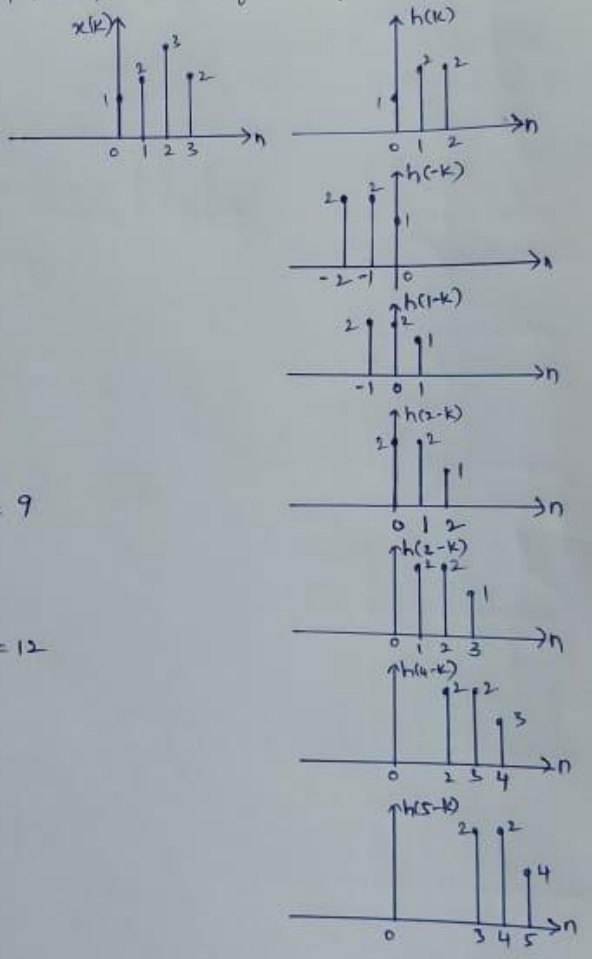
$h(n-k)$ represents right shift & $h(n+k)$ represents Left shift

Let $x(n) = [\underset{\uparrow}{1}\ 2\ 3\ 2]$ & $h(n) = [\underset{\uparrow}{1}\ 2\ 2]$

No. of samples in the output $= n_1 + n_2 - 1$

$= 4 + 3 - 1 = 6$ ($\because n_1$ & n_2 represents lengths of the sequences $x(n)$ & $h(n)$)

$$y(n) = \sum_{n=-\infty}^{\infty} x(k)h(n-k)$$



For $n=0 \Rightarrow y(0) = \sum_{n=-\infty}^{\infty} x(k)h(-k)$

$= (1)(1) = 1$

For $n=1 \Rightarrow y(1) = \sum_{n=-\infty}^{\infty} x(k)h(1-k)$

$= (1)(2) + (2)(1) = 4$

For $n=2 \Rightarrow y(2) = \sum_{n=-\infty}^{\infty} x(k)h(2-k)$

$= (1)(2) + (2)(2) + (3)(1) = 9$

For $n=3 \Rightarrow y(3) = \sum_{n=-\infty}^{\infty} x(k)h(3-k)$

$= (2)(2) + (3)(2) + (2)(1) = 12$

For $n=4 \Rightarrow y(4) = \sum_{n=-\infty}^{\infty} x(k)h(4-k)$

$= (3)(2) + (2)(2) = 10$

For $n=5 \Rightarrow y(5) = \sum_{n=-\infty}^{\infty} x(k)h(5-k)$

$= (2)(2) = 4$

$\therefore y(n) = \{1, 4, 9, 12, 10, 4\}$

Method-II:

	1	2	3	2
1	1	2	3	2
2	2	4	6	4
2	2	4	6	4

$\Rightarrow y(n) = \{1, 4, 9, 12, 10, 4\}$

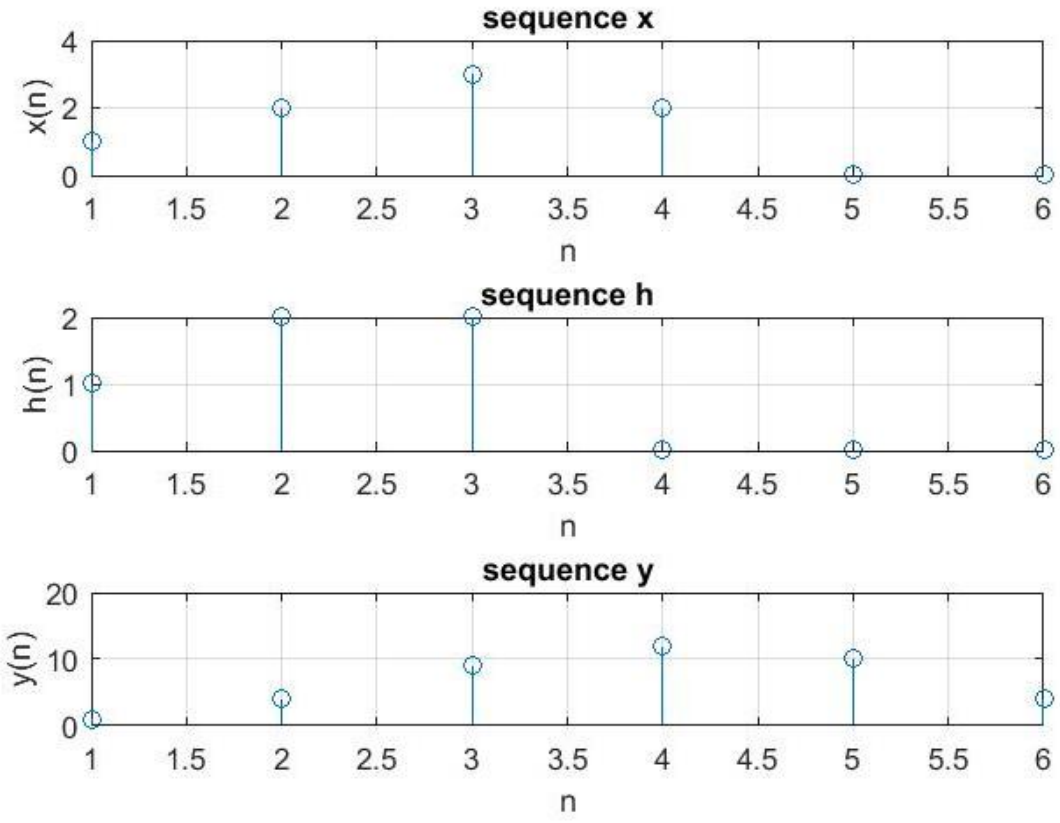
Program:

```
clc;
clear all;
close all;
x=input('enter the sequence: x');
h=input('enter the sequence: h');
n=length(x);
m=length(h);
k=n+m-1;
x=[x,zeros(1,m-1)];
h=[h,zeros(1,n-1)];
y=zeros(1,k);
for i=1:k;
        for j=1:i
        y(i)=y(i)+x(j)*h(i-j+1);
end;
end;
disp(y);
subplot(3,1,1);
stem(x);
xlabel('n');
ylabel('x(n)');
title('sequence x');
gridon;
subplot(3,1,2);
stem(h);
xlabel('n');
ylabel('h(n)');
title('sequence h');
gridon;
subplot(3,1,3);
stem(y);
```

```
xlabel('n');
ylabel('y(n)');
title('sequence y');
gridon;
```

Output:

```
enter the sequence: x=[1 2 3 2]
enter the sequence: h=[1 2 2]
    1     4     9    12    10     4
```



Viva-Voce Questions:

1. What is convolution in discrete-time systems?

2. How do you convolve two sequences in MATLAB?

3. Why do you need to verify the convolution result analytically?

4. What is the purpose of convolution in signal processing?

5. How do you handle sequences of different lengths during convolution?

9. Program to find frequency response of analog filters

Aim: To find the frequency response of analog filters

Software:MATLAB R2015a(8.5.0.197613)) version

Theory:

A filter is a frequency selective circuit designed to pass certainfrequencies and attenuate other frequencies. Filters are widely used in radio, television, audio-recording, radar, control-systems, music synthesis, image processing, and computer graphics.

The frequency response of the filter is understood by analyzing the characteristics o a filter. They are passband, stop band and cut-off frequency.

The band of frequencies of the input signal that pass through the filter without any attenuation is called **Passband**.The bandof frequencies of the input signal that are blocked or attenuated in the filter is called **Stopband.**The passband & stopband are distinguished from each other by the **cutoff frequency** or **corner frequency**. The output signal's voltage at the cutoff frequency is 70.7% of the input signal's voltage. It is also known as "**-3db frequency**" because -3db represents half power. And it is the frequency where the power of the output signal becomes half the power of the input signal.

In a band pass or band reject filter, there are two cutoff frequencies.

- Lower Cutoff Frequency:

The lower frequency at which the gain of the filter is half or -3db. It is denoted by f_1. Bandpass filter allows frequency after this point, whereas the band stop filter blocks it.

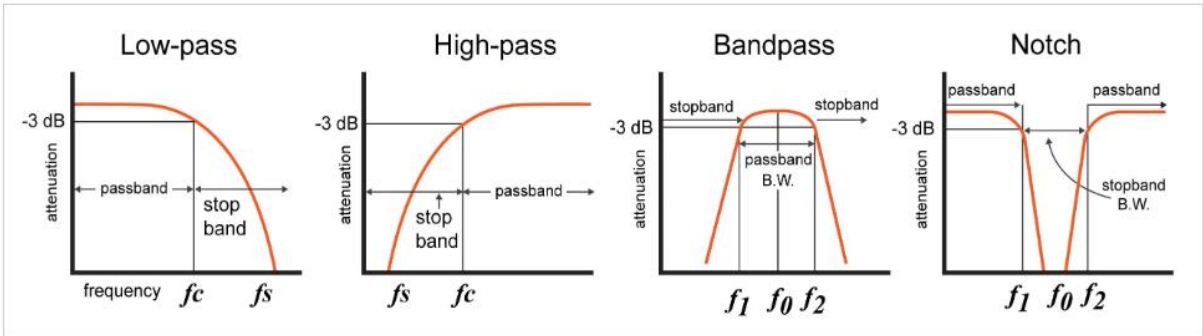
- Upper Cutoff Frequency

The upper frequency at which the output power is reduced by ½ of the input power. It is denoted by f_2. Bandpass filter does not allow frequency after this point, whereas the band stop filter allows it.

The frequency that lies at the center of the passband or stopband in a bandpass filter or band reject filter respectively is called **center frequency**. It lies in-between the two cut-off frequency i.e. lower & Upper cut-off frequency. In fact, it is the arithmetic mean of both cut-off frequencies.

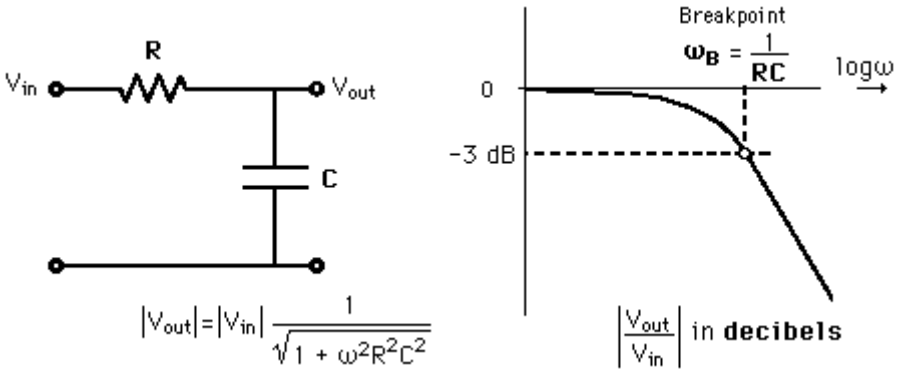
$$f_0 = \frac{f_1 + f_2}{2}$$

The range of frequencies that are passed (in case of bandpass filter) without any attenuation or the frequencies that are attenuated (in case of band reject filter) is called the **bandwidth** of the filter. The width of the frequencies before (in case of low pass filter) or after (in case of high pass filter) the cutoff frequency is called bandwidth.



LOW PASSFILTER:

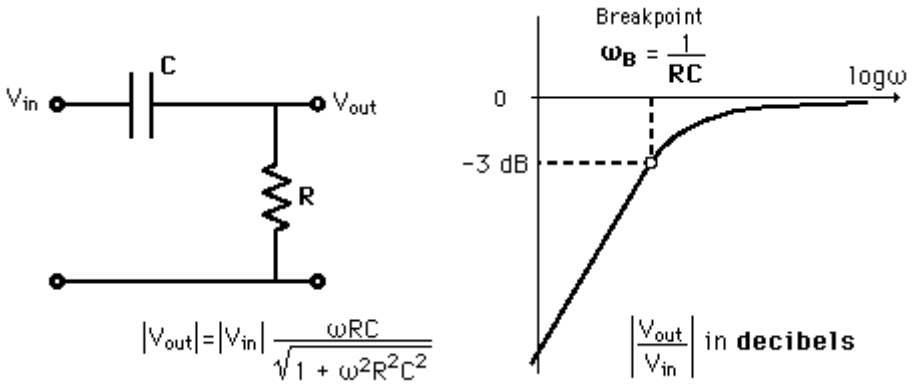
Alowpassfilter isonewhichpasses the frequencies below cut off frequency f_c and attenuates higher frequencies. Thepassbandfor low pass filteristhefrequency range below $f_c$andthestopbandis in thefrequency rangebelow f_c.



Magnitude response = $\frac{1}{1+j(\frac{f}{fc)=}}$ Phase response = $\tan^{-1}(\frac{f}{f_c})$

HIGH PASS FILTER:

A Highpassfilterattenuatesallfrequenciesbelow a designatedcutoff frequency$f_c$andpassesallfrequencies above f_c.Thepassband forhighpass filteris in thefrequencyrangeabove$f_c$andthestopbandis in thefrequency range below f_c.



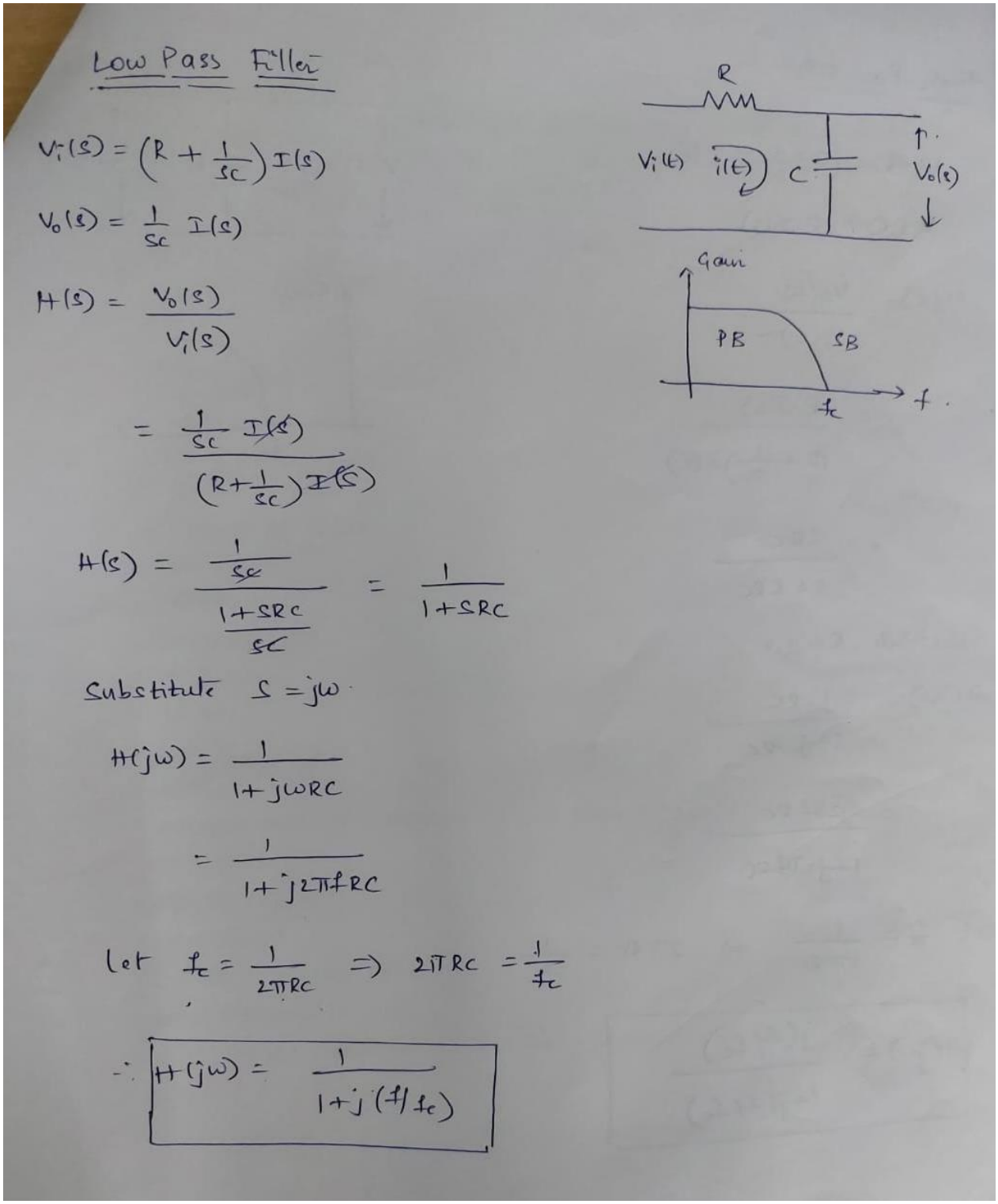
Magnitude response = $\frac{j(\frac{f}{f_c})}{1+j(\frac{f}{f_c})}$ Phase response = $\tan^{-1}(\frac{f}{f_c})$

Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

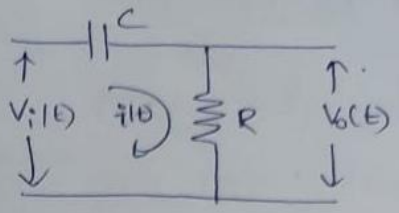
Theoretical Calculations:

(i) Low Pass Filter



(ii) High Pass Filter

High Pass Filter



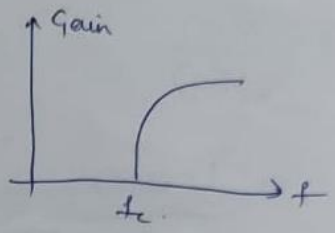
$$V_i(s) = \left(R + \frac{1}{sC}\right) I(s)$$

$$V_o(s) = R\,I(s)$$

$$H(s) = \frac{V_o(s)}{V_i(s)}$$

$$= \frac{R\,I(s)}{\left(R + \frac{1}{sC}\right) I(s)}$$

$$= \frac{sRC}{1 + sRC}$$



Substitute $s = j\omega$

$$H(j\omega) = \frac{j\omega RC}{1 + j\omega RC}$$

$$= \frac{j2\pi f RC}{1 + j2\pi f RC}$$

Let $f_c = \frac{1}{2\pi RC} \Rightarrow 2\pi RC = \frac{1}{f_c}$

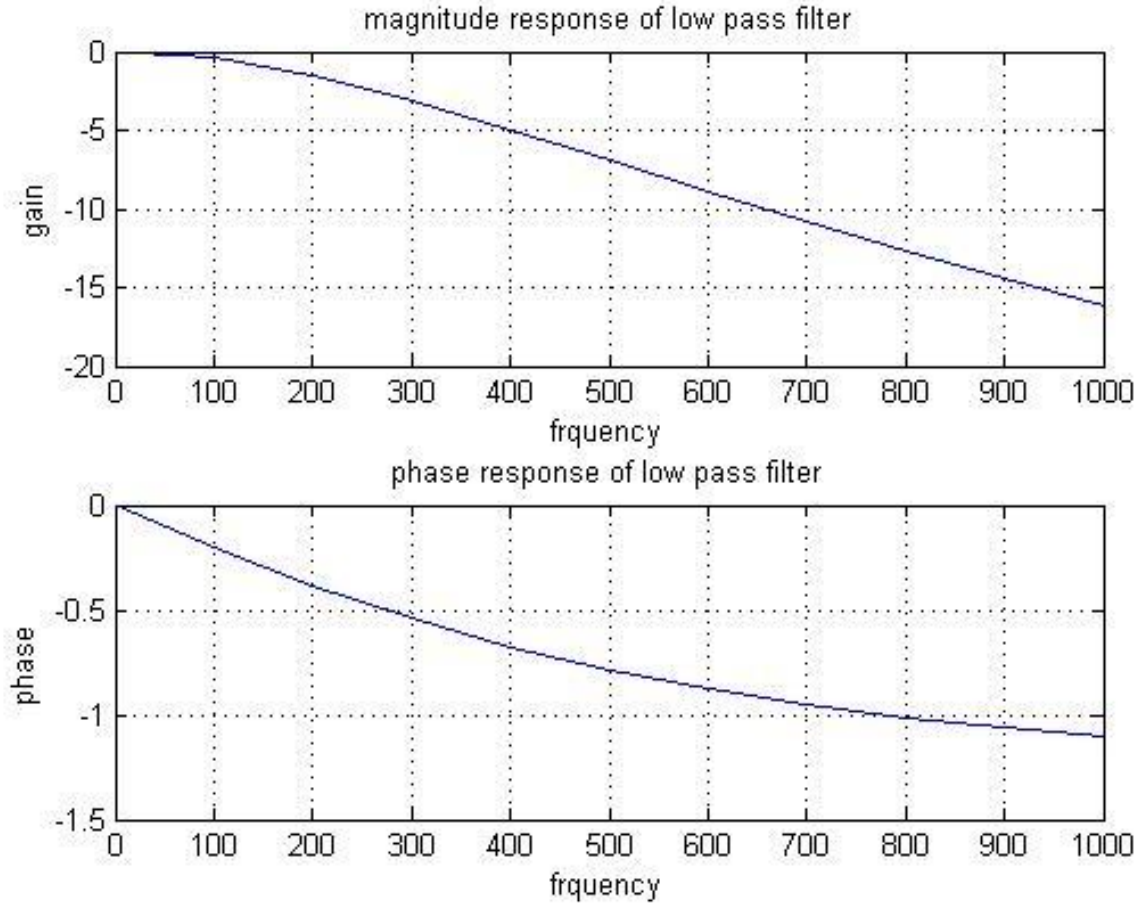
$$\therefore \boxed{H(j\omega) = \frac{j(f/f_c)}{1 + j(f/f_c)}}$$

Program:

Low Pass Filter:

```
clc
clearall;
closeall;
j=sqrt(-1);
f=0:100:1000;
fc=500;
a=1./(1+j*(f/fc));
h=abs(a);
g=20*log(h);
subplot(2,1,1);
plot(f,g);
gridon;
xlabel('frquency');
ylabel('gain');
title('magnitude response of low pass filter');
p=angle(a);
subplot(2,1,2);
plot(f,p);
xlabel('frquency');
ylabel('phase');
title('phase response of low pass filter');
gridon;
```

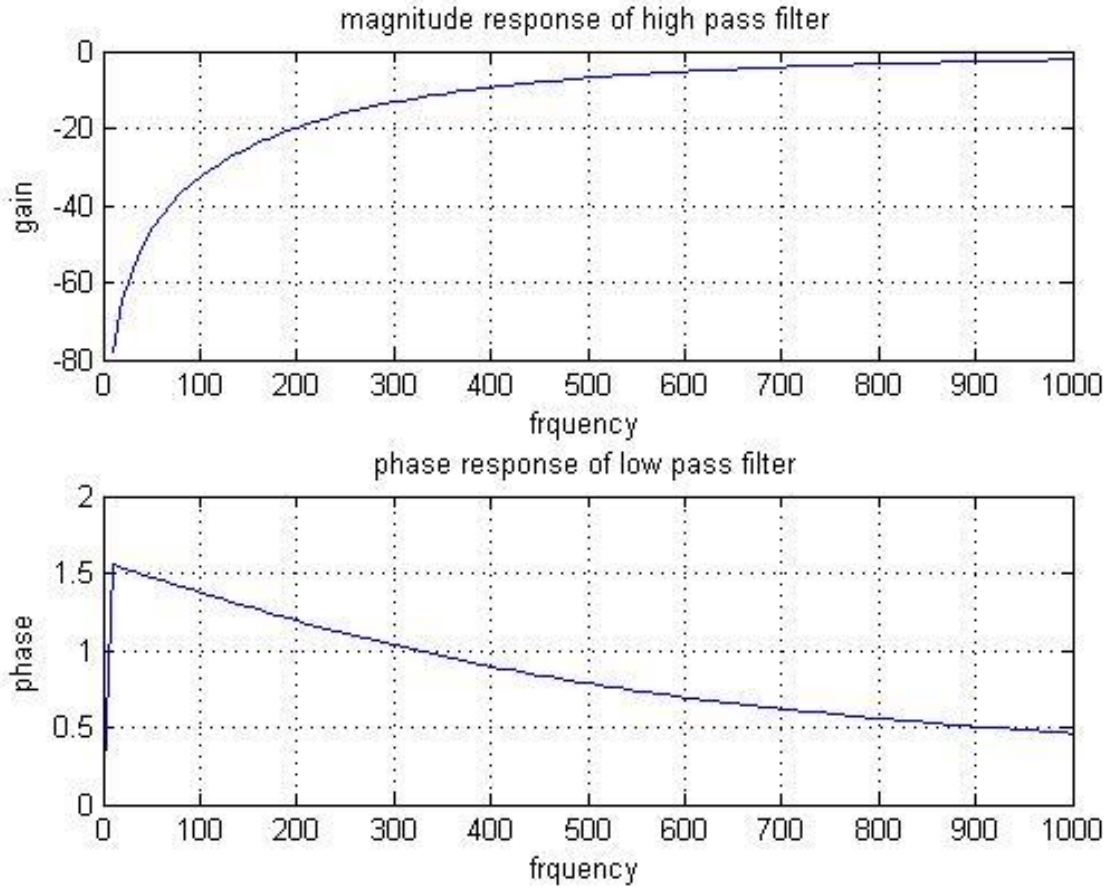
Output:-



High Pass Filter:

```
clc
clearall;
closeall;
j=sqrt(-1);
f=0:10:1000;
fc=500;
a=(j*(f/fc))./(1+j*(f/fc));
h=abs(a);
g=20*log(h);
subplot(2,1,1);
plot(f,g);
gridon;
xlabel('frquency');
ylabel('gain');
title('magnitude response of high pass filter');
p=angle(a);
subplot(2,1,2);
plot(f,p);
xlabel('frquency');
ylabel('phase');
title('phase response of low pass filter');
gridon;
```

Output:-



Viva-Voce Questions:

1. What is the frequency response of an analog filter, and why is it important?

2. How do you calculate the frequency response of an analog filter in MATLAB?

3. What is the significance of the magnitude and phase plots in the frequency response?

4. How do you relate the frequency response to the filter's transfer function?

5. What information does the frequency response provide about the filter's behavior?

10. Sampling Theorem Verification

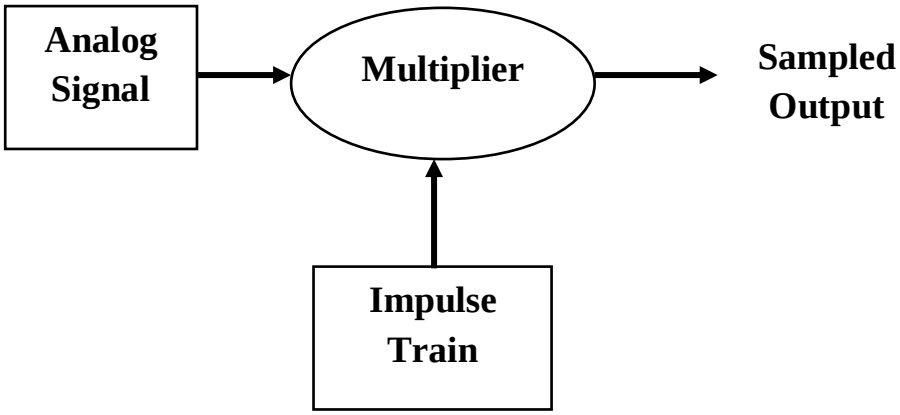
Aim: To find the Inverse Fourier Transform of a sinc function and plot its amplitude and phase spectrum.

Software:MATLAB R2015a(8.5.0.197613) version

Theory:

A signal has three properties like voltage or amplitude, frequency, phase. The signals are represented only in an analog form where the digital form of technology is not available. Analog signals are continuous in time and difference in voltage levels for different periods of the signal. Here, the main drawback of this is, the amplitude keeps on changing along with the period of the signal. This can be overcome by the digital form of signal representation. Here conversion of an analog form of the signal into digital form can be done using the sampling technique. The output of this technique represents the discrete version of its analog signal.

A continuous signal or an analog signal can be represented in the digital version in the form of samples. Here, these samples are also called as discrete points. In sampling theorem, the input signal is in an analog form of signal and the second input signal is a sampling signal, which is a pulse train signal and each pulse is equidistance with a period of “Ts”. This sampling signal frequency should be more than twice of the input analog signal frequency. If this condition satisfies, analog signal perfectly represented in discrete form else analog signal may be losing its amplitude values for certain time intervals. How many times the sampling frequency is more than the input analog signal frequency, in the same way, the sampled signal is going to be a perfect discrete form of signal. And these types of discrete signals are well performed in the reconstruction process for recovering the original signal.



Statement: A continuous time signal can be represented in its samples and can be recovered back when sampling frequency f_s is greater than or equal to the twice the highest frequency component of message signal. i. e. $f_s \geq 2f_m$

The output sample signal is represented by the samples. These samples are maintained with a gap, these gaps are termed as sample period or sampling interval (Ts). And the reciprocal of the sampling period is known as "sampling frequency" or "sampling rate". The number of samples is represented in the sampled signal is indicated by the sampling rate.

Sampling frequency **fs=1/Ts**

If the sampling frequency (Fs) equals twice the input signal frequency (Fm), then such a condition is called the Nyquist Criteria for sampling. When sampling frequency equals twice the input signal frequency is known as "Nyquist rate".

fs=2fm

If the sampling frequency (Fs) is less than twice the input signal frequency, such criteria called an Aliasing effect.

fs<2fm

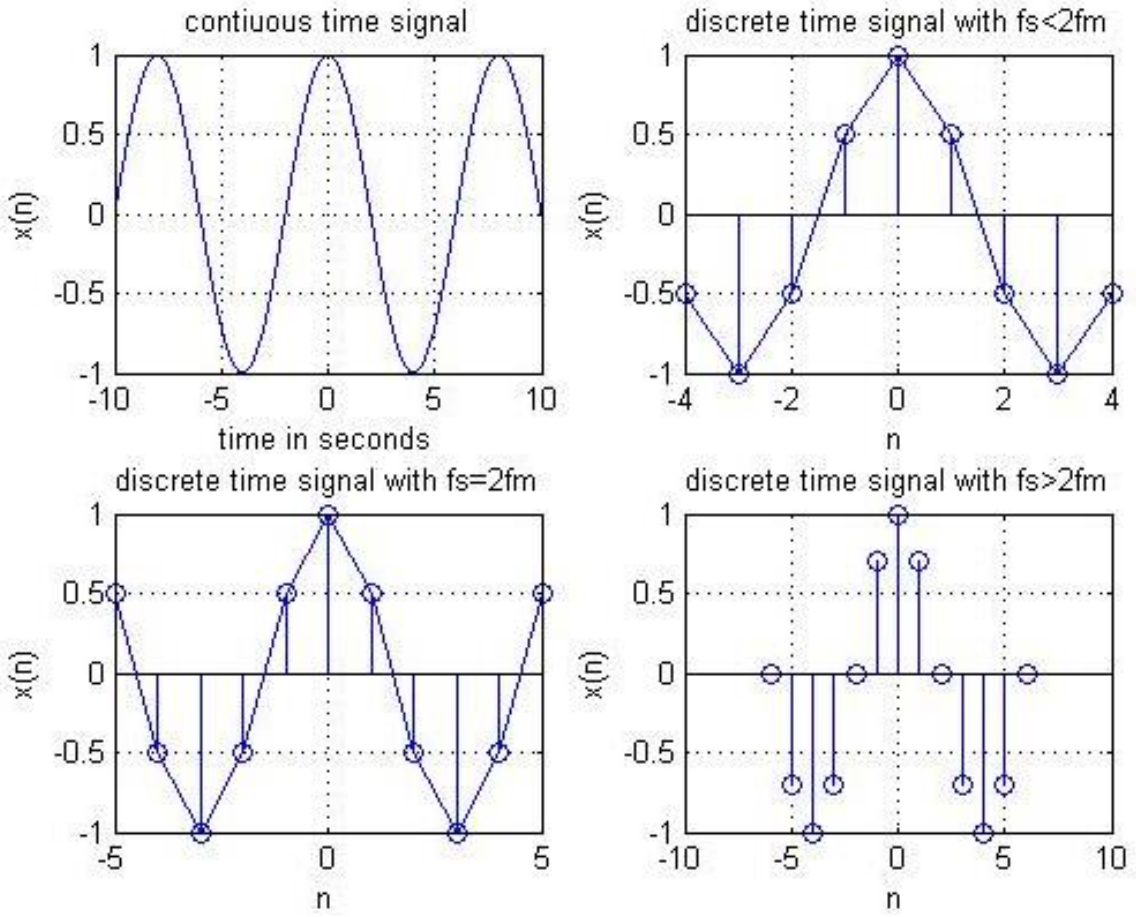
Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Program:

```
clc;
clearall;
closeall;
t=-10:0.01:10;
T=8;
fm=1/T;
x=cos(2*pi*fm*t);
fs1=1.2*fm;
fs2=1.2*fm;
fs3=8*fm;
n1=-4:1:4;
xn1=cos(2*pi*n1*fm/fs1);
subplot(2,2,1);
plot(t,x);
xlabel('time in seconds');
ylabel('x(n)');
title('continuous time signal');
```

```
gridon;
subplot(2,2,2);
stem(n1,xn1);
holdon;
gridon;
plot(n1,xn1);
xlabel('n');
ylabel('x(n)');
title('discrete time signal with fs<2fm');
n2=-5:1:5;
xn2=cos(2*pi*n2*fm/fs2);
subplot(2,2,3);
stem(n2,xn2);
holdon;
gridon;
plot(n2,xn2);
xlabel('n');
ylabel('x(n)');
title('discrete time signal with fs=2fm');
holdoff;
gridon;
n3=-6:1:6;
xn3=cos(2*pi*n3*fm/fs3);
subplot(2,2,4);
stem(n3,xn3);
xlabel('n');
ylabel('x(n)');
title('discrete time signal with fs>2fm');
holdoff;
gridon;
```



Result: The sampling theorem is verified

Viva-Voce Questions:

1. What is the Sampling Theorem, and what does it state?

2. How does the sampling rate affect the reconstruction of a signal?

3. What is aliasing, and how does it relate to the sampling rate?

4. How do you verify the Sampling Theorem experimentally?

5. What is the significance of the Nyquist rate in the Sampling Theorem?

11. Write a program to find the autocorrelation and cross correlation of sequences.

Aim: To find the auto correlation and cross correlation of sequences

Software:MATLAB R2015a(8.5.0.197613) version

Theory:

The *correlation* of two functions or signals or waveforms is defined as the measure of similarity between those signals.

There are two types of correlations.

- Autocorrelation
- Cross-correlation

The autocorrelation function is defined as the measure of similarity or coherence between a signal and its time delayed version. Therefore, the autocorrelation is the correlation of a signal with itself.

The autocorrelation of an energy or aperiodic signal x(t) is defined as

$$\boldsymbol{R_{11}(\tau) = \int_{-\infty}^{\infty} x(t)x^*(t-\tau)dt}$$

Where,the variable τ iscalled the delay parameter and here, the signal x(t) is time shifted by τ units in the positive direction.

If the signal x(t) is shifted by τ units in negative direction, then the autocorrelation of the signal is defined as,

$$\boldsymbol{R_{11}(\tau) = \int_{-\infty}^{\infty} x(t+\tau)x^*(t)dt}$$

The autocorrelation of a power signal or periodic signal x(t) having a time period T is defined as

$$\boldsymbol{R_{11}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)x^*(t-\tau)dt}$$

The autocorrelation of the sequence x[n] is given by

$$\boldsymbol{y(k) = \sum_{k=-\infty}^{\infty} x(n)x(n-k)}$$

Where, the variable k is called the delay parameter and here, the sequencex(n) is time shifted by n units in the positive direction.

If the sequence x(n) is shifted by k units in negative direction, then the autocorrelation of the sequence is defined as,

$$\boldsymbol{y(k) = \sum_{k=-\infty}^{\infty} x(n+k)x(n)}$$

The cross-correlation between two different signals or functions or waveforms is defined as the measure of similarity or coherence between one signal and the time-delayed version of another signal. The cross-correlation between two different signals indicates the degree of relatedness between one signal and the time-delayed version of another signal.

The cross-correlation of energy (or aperiodic) signals and power (or periodic) signals is defined separately.

Consider two complex signals $x_1(t)$ and $x_2(t)$ of finite energy. Then, the cross-correlation of these two energy signals is defined as

$$\boldsymbol{R_{12}(\tau) = \int_{-\infty}^{\infty} x_1(t)x_2^*(t-\tau)dt = \int_{-\infty}^{\infty} x_1(t+\tau)x_2^*(t)dt}$$

If the two signals x1(t)x1(t) andx2(t) are real, then the cross-correlation between them is,

$$\boldsymbol{R_{12}(\tau) = \int_{-\infty}^{\infty} x_1(t)x_2(t-\tau)dt = \int_{-\infty}^{\infty} x_1(t+\tau)x_2(t)dt}$$

If the energy signals $x_1(t)$ and $x_2(t)$ have some similarity. Then, the cross-correlation $R_{12}(\tau)$ between them will have some finite value over the range τ. The variable τ is called the *delay parameter or searching parameter or scanning parameter*. The time-delay parameter (τ) is the time delay or time shift of one of the two signals. This delay parameter τ determines the correlation between two signals.

Thecross-correlation $R_{12}(\tau)$ for two power(or periodic) signals$x_1(t)$ and $x_2(t)$ may be defined using the average form of correlation. If two power signals $x_1(t)$ and $x_2(t)$ have the same time period (say T), then the cross-correlation between them is defined as

$$\boldsymbol{R_{12}(\tau) = \frac{1}{T}\int_{-T/2}^{T/2} x_1(t)x_2{}^*(t-\tau)dt}$$

The cross correlation of the sequences x(n) and h(n) are defined as

$$\boldsymbol{y(k) = \sum_{k=-\infty}^{\infty} x(n)h(n-k) = \sum_{k=-\infty}^{\infty} h(n+k)x(n)}$$

Computation of Correlation:

Step 1: Obtain the sequence h(n-k) by shifting the sequence right by the time lag'k'

Step 2: Multiply the shifted sequence h(n-k) with x(n) and sum all the values to obtain y(k)

Step 3: Repeat steps 1 & 2 for left shift '-k'

Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Theoretical Calculations:

(i) AutoCorrelation:

Autocorelation

Method-I:-

Consider a sequence $x(n)$.

Autocorelation of the sequence is $y(k) = \sum_{k=-\infty}^{\infty} x(n)\,x(n-k)$

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

$x(n-k)$ represents shift towards right by k units ($\because n-k=0 \Rightarrow n=k$)

$x(n+k)$ represents shift towards left by k units ($\because n+k=0 \Rightarrow n=-k$)

let $x(n) = [1\ \underset{\uparrow}{2}\ 1\ 3]$

No. of samples in the output $= 2(n)-1 = 2(4)-1 = 7$

So, shift the sequence $x(n)$ 3 times to the left and 3 times to the right

$$y(k) = \sum_{k=-\infty}^{\infty} x(n)\,x(n-k)$$

For $k=0 \Rightarrow y(0) = \sum_{k=-\infty}^{\infty} x(n)\,x(n)$

$= (1)(1)+(2)(2)+(1)(1)+(3)(3) = 15$

For $k=1 \Rightarrow y(1) = \sum_{k=-\infty}^{\infty} x(n)\,x(n+1)$

$= (2)(1)+(2)(1)+(1)(3) = 7$

For $k=-2 \Rightarrow y(-2) = \sum_{k=-\infty}^{\infty} x(n)\,x(n+2)$

$= (1)(1)+(3)(2) = 7$

For $k=-3 \Rightarrow y(-3) = \sum_{k=-\infty}^{\infty} x(n)\,x(n+3)$

$= (1)(3) = 3$

For $k=1 \Rightarrow y(1) = \sum_{k=-\infty}^{\infty} x(n)\,x(n-1)$

$= (1)(2)+(2)(1)+(3)(1) = 7$

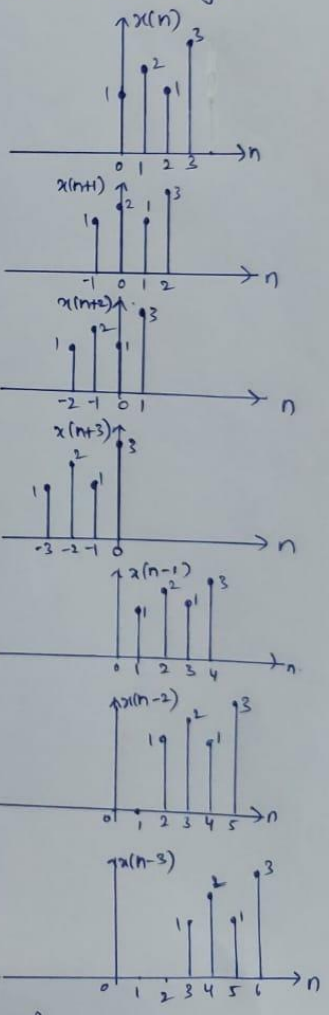
For $k=2 \Rightarrow y(2) = \sum_{k=-\infty}^{\infty} x(n)\,x(n-2)$

$= (1)(1)+(3)(2) = 7$

For $k=3 \Rightarrow y(3) = \sum_{k=-\infty}^{\infty} x(n)\,x(n-3)$

$= (3)(1) = 3$

$\therefore y(k) = \{3, 7, 7, \underset{\uparrow}{15}, 7, 7, 3\}$



Method II:

	1	2	1	3
3	3	6	3	9
1	1	2	1	3
2	2	4	2	6
1	1	2	1	3

$\Rightarrow y(k) = \{3, 7, 7, 15, 7, 7, 3\}$

(ii) Cross Correlation:

Cross Correlation

Method I :-

Consider two sequences $x(n)$ and $h(n)$

The cross correlation is given by $y(k) = \sum_{k=-\infty}^{\infty} x(n)\,h(n-k)$

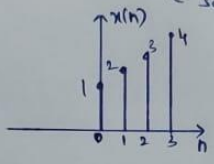
(or) $y(k) = \sum_{k=-\infty}^{\infty} h(n+k)\,x(n)$

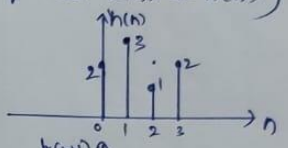
$h(n-k)$ represents shift towards right side of $h(n)$ by k units

$h(n+k)$ represents shift towards left side of $h(n)$ by k units

Let $x(n) = [\underset{\uparrow}{1}\ 2\ 3\ 4]$ & $h(n) = [\underset{\uparrow}{2}\ 3\ 1\ 2]$

No. of samples in the output $= n_1 + n_2 - 1 = 4+4-1 = 7$ ($\because n_1$ & n_2 are lengths of the sequences $x(n)$ & $h(n)$)

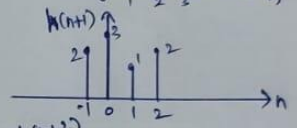




$y(k) = \sum_{k=-\infty}^{\infty} x(n)\,h(n-k)$

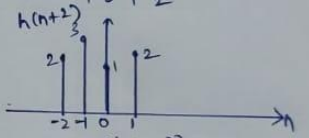
For $k=0 \Rightarrow y(0) = \sum_{k=-\infty}^{\infty} x(n)\,h(n)$

$= (1)(2)+(2)(3)+(3)(1)+(4)(2) = 19$



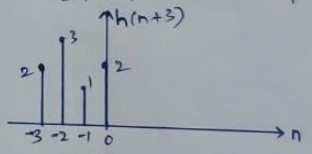
For $k=1 \Rightarrow y(1) = \sum_{k=-\infty}^{\infty} x(n)\,h(n+1)$

$= (3)(1)+(2)(1)+(3)(2) = 11$



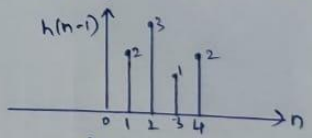
For $k=2 \Rightarrow y(2) = \sum_{k=-\infty}^{\infty} x(n)\,h(n+2)$

$= (1)(1)+(2)(2) = 1+4 = 5$



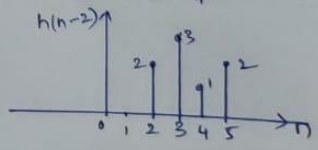
For $k=3 \Rightarrow y(3) = \sum_{k=-\infty}^{\infty} x(n)\,h(n+3)$

$= (1)(2) = 2$



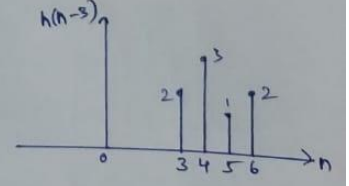
For $k=-1 \Rightarrow y(-1) = \sum_{k=-\infty}^{\infty} x(n)\,h(n-1)$

$= (2)(2)+(3)(3)+(4)(1) = 17$



For $k=-2 \Rightarrow y(-2) = \sum_{k=-\infty}^{\infty} x(n)\,h(n-2)$

$= (3)(2)+(3)(4) = 18$



For $k=-3 \Rightarrow y(3) = \sum_{k=-\infty}^{\infty} x(n)\,h(n-3)$

$= (4)(2) = 8$

$\therefore y(k) = \{2, 5, 11, \underset{\uparrow}{19}, 17, 18, 8\}$

Method II :-

	1	2	3	4
2	2	4	6	8
1	1	2	3	4
3	3	6	9	12
2	2	4	6	8

$\Rightarrow y(k) = \{2, 5, 11, \underset{\uparrow}{19}, 17, 18, 8\}$

Programs:

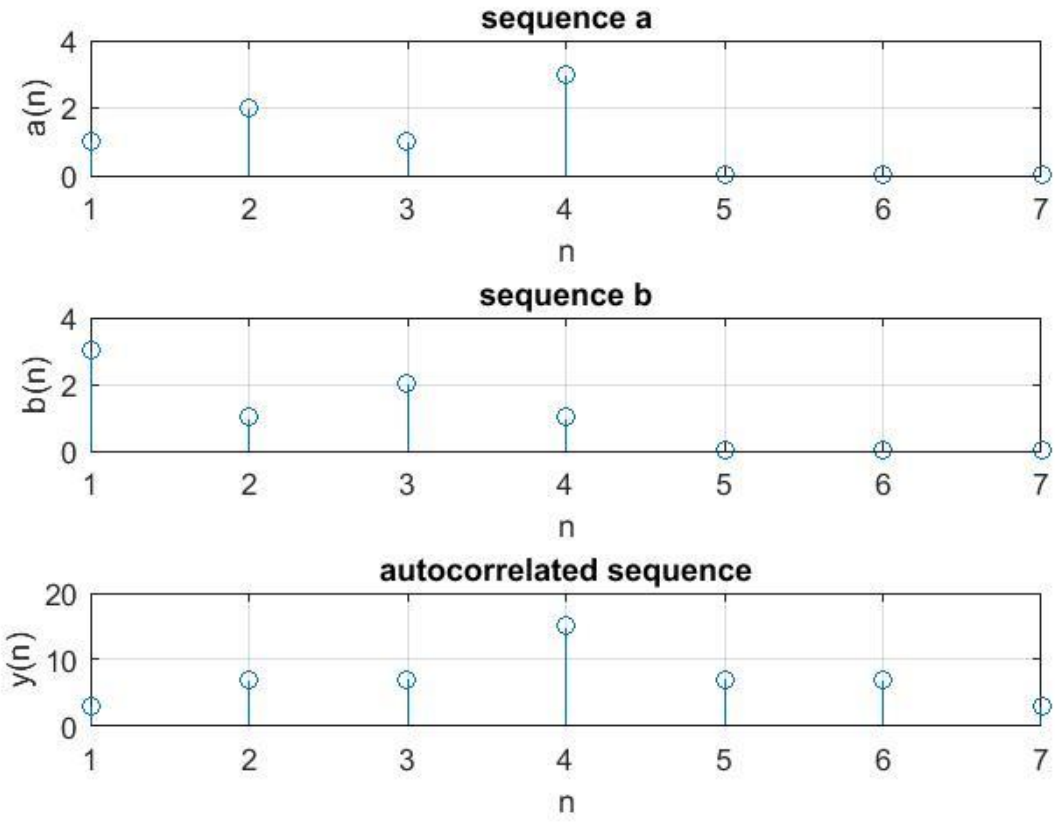
(i) Auto Correlation:

```
clc;
clearall;
closeall;
a=input('enter the sequence');
n=length(a);
k=(2*n)-1;
b(1:n)=a(n:-1:1);
a=[a,zeros(1,n-1)];
b=[b,zeros(1,n-1)];
y=zeros(1,k);
fori=1:k
        for j=1:i
        y(i)=y(i)+a(j)*b(i-j+1);
end;
end;
disp(y);
subplot(3,1,1);
stem(a);
xlabel('n');
ylabel('a(n)');
title('sequence a');
gridon;
subplot(3,1,2);
stem(b);
xlabel('n');
ylabel('b(n)');
title('sequence b');
gridon;
subplot(3,1,3);
stem(y);
xlabel('n');
```

```
ylabel('y(n)');
title('autocorrelated sequence');
gridon;
```

Output:

```
enter the sequence [1 2 1 3]
    3     7     7    15     7     7     3
```



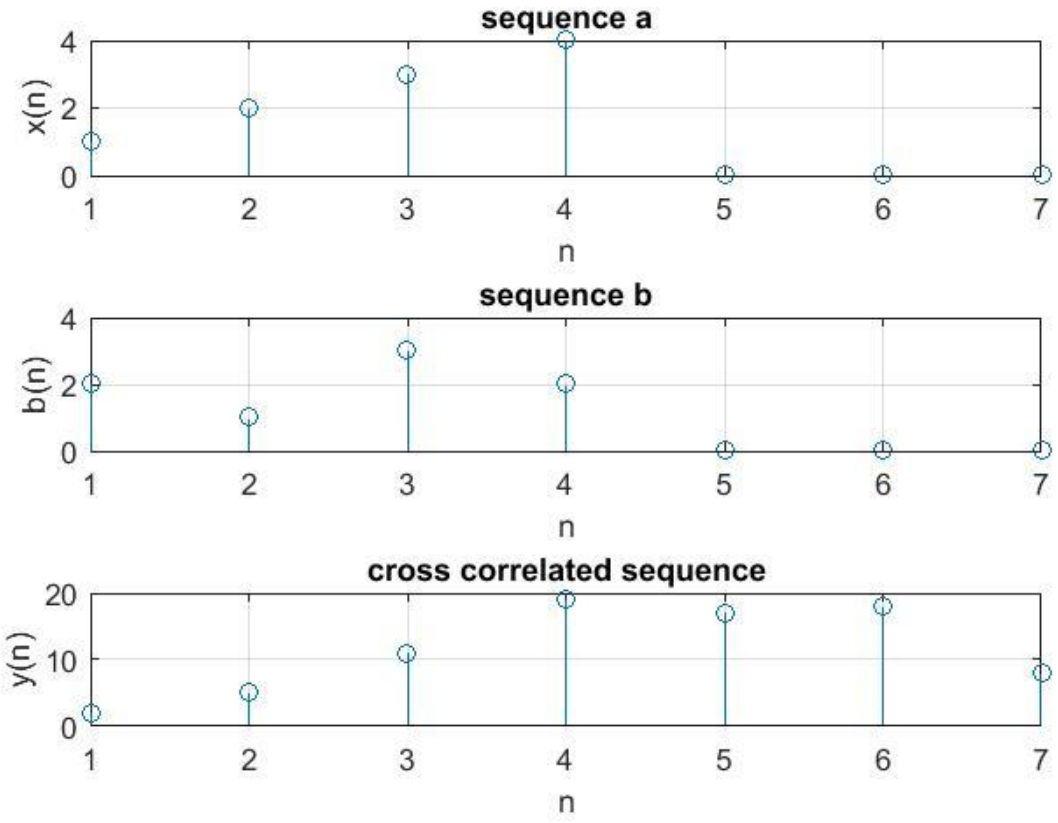
(ii) Cross Correlation:

```
clc;
clearall;
closeall;
x=input('enter first sequence');
h=input(enter second sequence');
n1=length(x);
n2=length(h);
k=n1+n2-1;
b(1:n1)=h(n2:-1:1);
x=[x,zeros(1,n2-1)];
b=[b,zeros(1,n1-1)];
y=zeros(1,k);
for i=1:k
        for j=1:i
y(i)=y(i)+x(j)*b(k-p+1);
end
end
disp(y);
subplot(3,1,1);
stem(a);
xlabel('n');
ylabel('a(n)');
title('sequence a');
gridon;
subplot(3,1,2);
stem(b);
xlabel('n');
ylabel('b(n)');
title('sequence b');
gridon;
subplot(3,1,3);
stem(y);
```

```
xlabel('n');
ylabel('y(n)');
title('cross correlated sequence');
gridon;
```

Output:

```
enter first sequence [1 2 3 4]
enter second sequence [2 3 1 2]
    2     5     11     19     17     18     8
```



Viva-Voce 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?

12. The signal x(t) is defined as below. The signal is sampled at a sampling rate of 1000 samples per second. Find the power content and power spectral density for the signal.
X(t)=cos(2pi*47t)+cos(2pi*219t), 0<t<10 X(t)=0, otherwise

Aim: To find the power and power spectral density of the signal

Software:MATLAB R2015a(8.5.0.197613) version

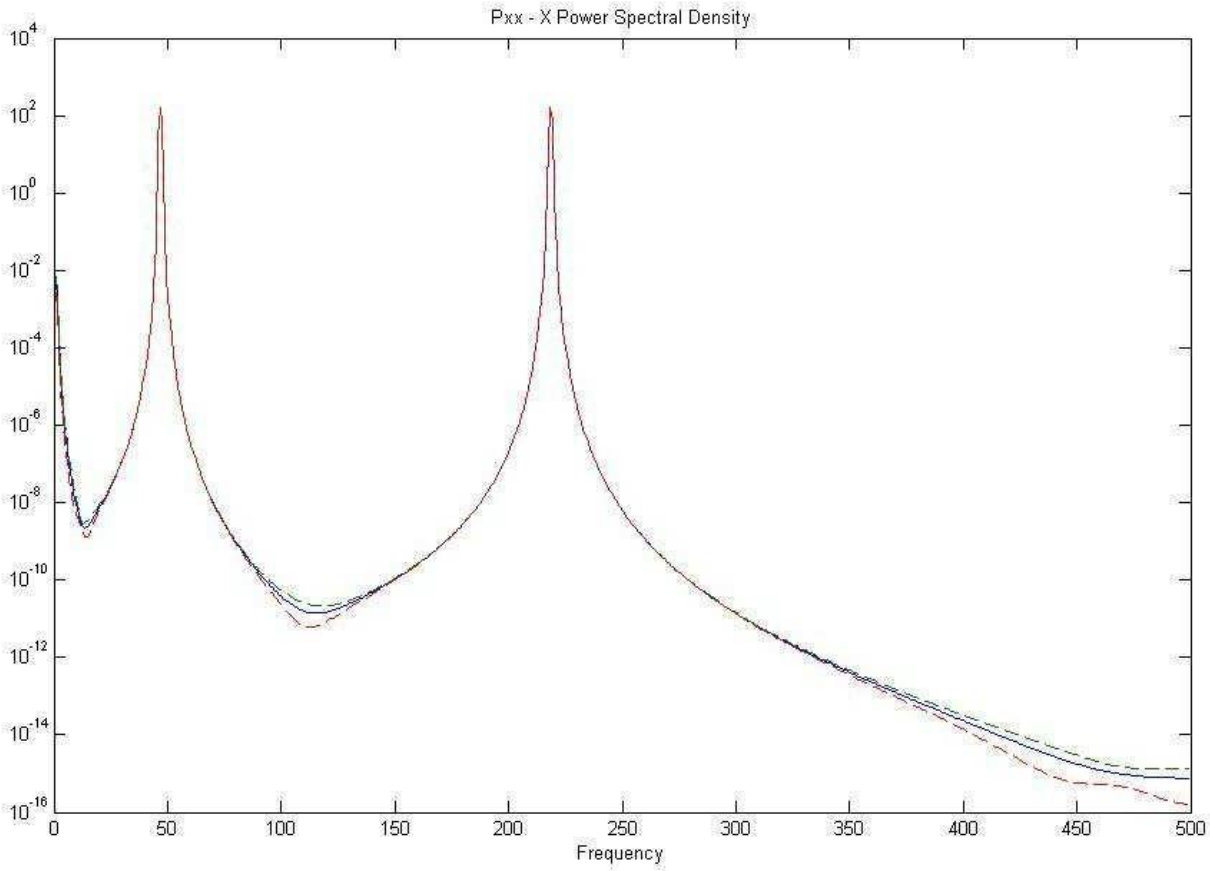
Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Program:

```
function p=psd()
t=0:(1/1000):10;
x=cos(2*pi*47*t)+cos(2*pi*219*T);
l=length(x);
p=(norm(x)^2)/l;
ps=spectrum(x,1024);
specplot(ps,1000)
```

Output:



Viva-Voce Questions:

1. What is the power content of a signal, and how is it calculated?

2. How do you calculate the power spectral density (PSD) of a signal?

3. What is the significance of the sampling rate in this experiment?

4. How do you relate the frequencies present in the signal to the power spectral density?

5. What information do the power content and PSD provide about the signal's properties?

13. Write a MATLAB program to perform waveform synthesis using Laplace Transform of a given signal.

Aim: Wave form synthesis using Laplace Transforms.

Software: MATLAB R2015a(8.5.0.197613) version

Theory:

Bilateral Laplace transforms:

The Laplace transform of a signal f(t) can be defined as follows:

$$\boldsymbol{F(S) = \mathcal{L}\{f(t)\} = \int_{-\infty}^{\infty} e^{-st} f(t)\,dt}$$

Inverse Laplace transform

The inverse Laplace transform is given by the following formula :

$$\boldsymbol{f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi i}\lim_{T\to\infty}\int_{\gamma-iT}^{\gamma+iT} e^{st}F(s)\,ds}$$

Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.Save the program using specified name other than program name with **.m** extension.
3. **Debug** the program for errors.
4. If errors are present correct them.
5. **Run** the program.
6. Observe for resultant **graphs.**

Program:

```
clc;
clear all;
close all;
%representation of symbolic variables
syms f t w s;
%laplace transform of t
f=t;
z=laplace(f);
```

```
disp('the laplace transform of f = ');
disp(z);
% laplace transform of a signal
%f1=sin(w*t);
f1=-1.25+3.5*t*exp(-2*t)+1.25*exp(-2*t);
v=laplace(f1);
disp('the laplace transform of f1 = ');
disp(v);
lv=simplify(v);
pretty(lv)
%inverse laplace transform
y1=ilaplace(z);
disp('the inverse laplace transform of z = ');
disp(y1);
y2=ilaplace(v);
disp('the inverse laplace transform of v = ');
disp(y2);
ezplot(y1);
figure;
ezplot(y2)
```

Output:

```
thelaplace transform of f =
1/s^2
thelaplace transform of f1 =
5/(4*(s + 2)) + 7/(2*(s + 2)^2) - 5/(4*s)
s - 5
----------
```

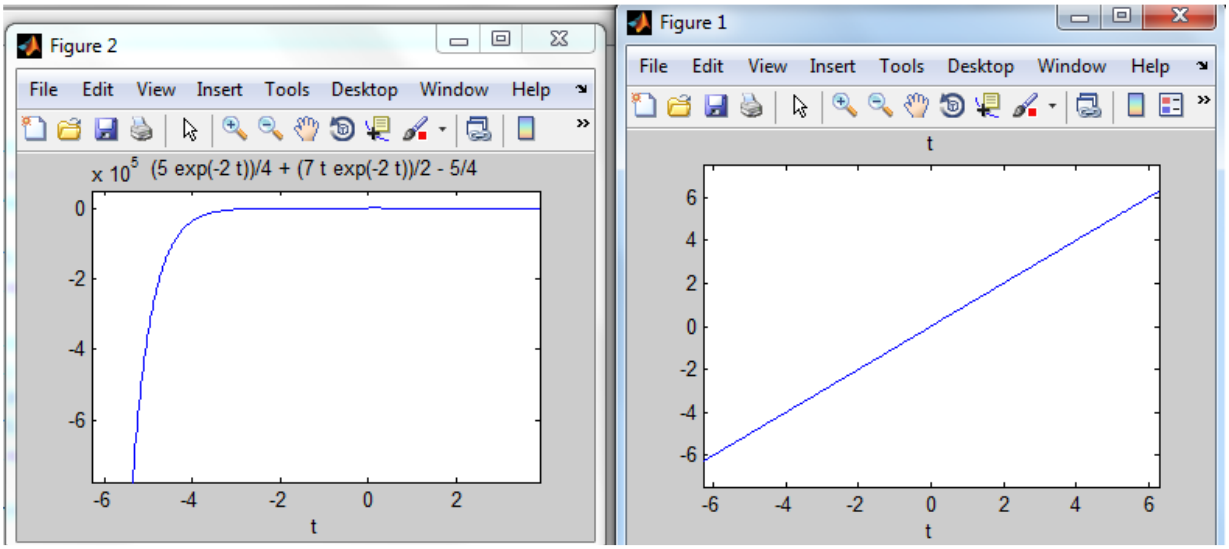
s (s + 2)2

the inverse laplace transform of z =

t

the inverse laplace transform of v =

5/(4*exp(2*t)) + (7*t)/(2*exp(2*t)) - 5/4



Viva-Voce Questions:

1. Define Laplace-Transform?

2. What is the Condition for Convergence of the L.T?

3. What is the Region of Convergence (ROC)?

4. State the Shifting property of L.T?

5. State convolution Property of L.T?

14. Write a MATLAB program to determine the Z-transform and plot the pole-zero locations in Z-plane.

Aim: To locating poles and zeros and plotting pole-zero maps in z-plane for the given transfer function.

Software Required: MATLAB R2015a(8.5.0.197613) version

Theory:

Z-transforms: The **Z-transform** converts a discrete time-domain signal, which is a sequence of real or complex numbers, into a complex frequency-domain representation.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

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

Unilateral Z-transform: Alternatively, in cases where x[n] is defined only for n = 0, the single-sided or unilateral Z-transform is defined as

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

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

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

Where

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

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)

The ROC of the Z-transform depends on the convergence of the Transfer function

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

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

and poles for H(z)

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

Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Program:

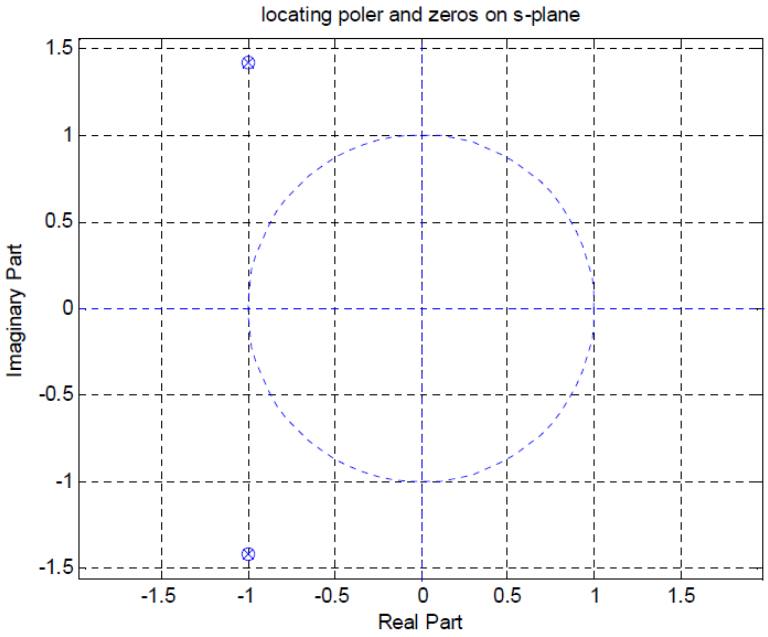
```
clc;

clear all;

close all;

num=input('enter numerator coefficient');

den=input('enter denominator coefficient');

p=roots(den);

z=roots(num);
```

```
zplane(p,z);

grid;

title('locating poles and zeros on z-plane');
```

Output:

locating poles &zeros on z-plane

enter numerator coefficient[1 2 3]



enter denominator coefficient[1 2 3]

Viva-Voce 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?

15. Program to find the impulse response of the system defined by a difference equation.

Aim: To write a MATLAB program to evaluate the impulse response of the system.

Software: MATLAB R2015a(8.5.0.197613) version

Theory: The Difference equation is given as

$$y(n) = x(n)+0.5x(n-1)+0.85x(n-2)+y(n-1)+y(n-2)$$

Procedure:

1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Program:

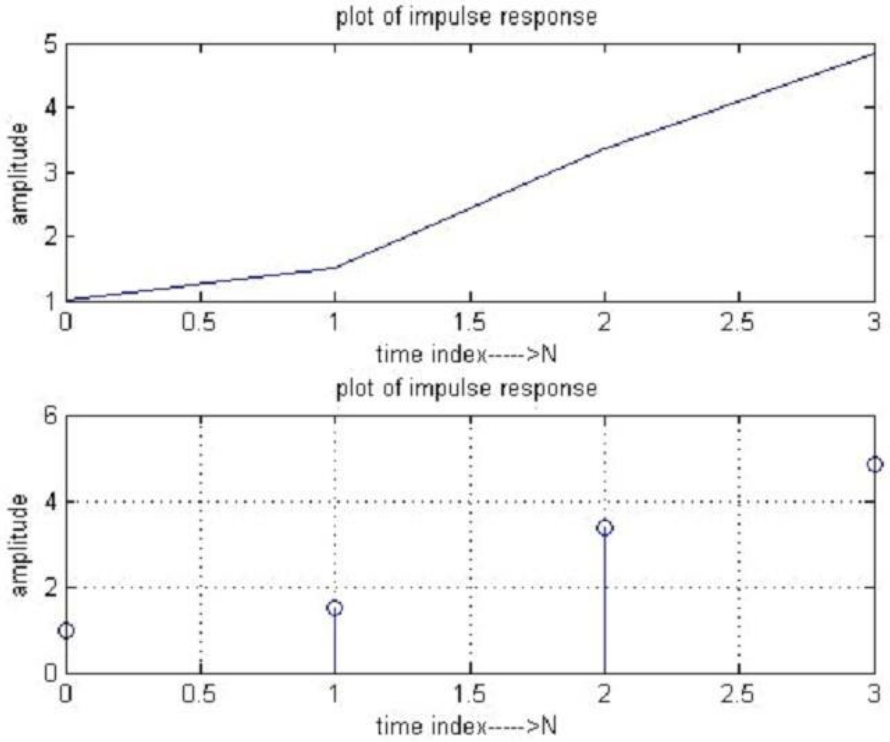
```
clc;
clear all;
close all;
% Difference equation of a second order system
% y(n) = x(n)+0.5x(n-1)+0.85x(n-2)+y(n-1)+y(n-2)
b=input('enter the coefficients of x(n),x(n-1)-----');
a=input('enter the coefficients of y(n),y(n-1)----');
N=input('enter the number of samples of imp response ');
[h,t]=impz(b,a,N);
```

```
subplot(2,1,1);
% figure(1);
plot(t,h);
title('plot of impulse response');
ylabel('amplitude');
xlabel('time index----->N');
subplot(2,1,2);
% figure(2);
stem(t,h);
title('plot of impulse response');
ylabel('amplitude');
xlabel('time index----->N');
disp(h);
grid on;
```

Output:

```
enter the coefficients of x(n),x(n-1)-----[1 0.5 0.85]
enter the coefficients of y(n),y(n-1)-----[1 -1 -1]
enter the number of samples of imp respons 4
1.0000
1.5000
3.3500
```

4.8500



Viva-Voce Questions:

1. What is an impulse response, and what does it represent in a system?

2. How do you calculate the impulse response from a difference equation?

3. What is the significance of the initial conditions in finding the impulse response?

4. How does the impulse response relate to the system's transfer function?

5. What information does the impulse response provide about the system's behavior?

EXPERIMENTS BEYOND THE SYLLABUS

1. GIBBS PHENOMENON

Aim: To verify the Gibbs Phenomenon.

Software:MATLAB R2015a(8.5.0.197613) version

Theory:

The Gibbs phenomenon, the Fourier series of a piecewise continuously differentiable periodicfunction behaves at a jump discontinuity. The *n* the approximated function shows amounts of ripples atthe points of discontinuity. This is known as the Gibbs Phenomina .partial sum of the Fourier serieshas large oscillations near the jump, which might increase the maximum of the partial sum above thatof the function itself. The overshoot does not die out as the frequency increases, but approaches a finite limit. The Gibbs phenomenon involves both the fact that Fourier sums overshoot at a jump discontinuity, and that this overshoot does not die out as the frequency increases Gibbs Phenomena.

Procedure:

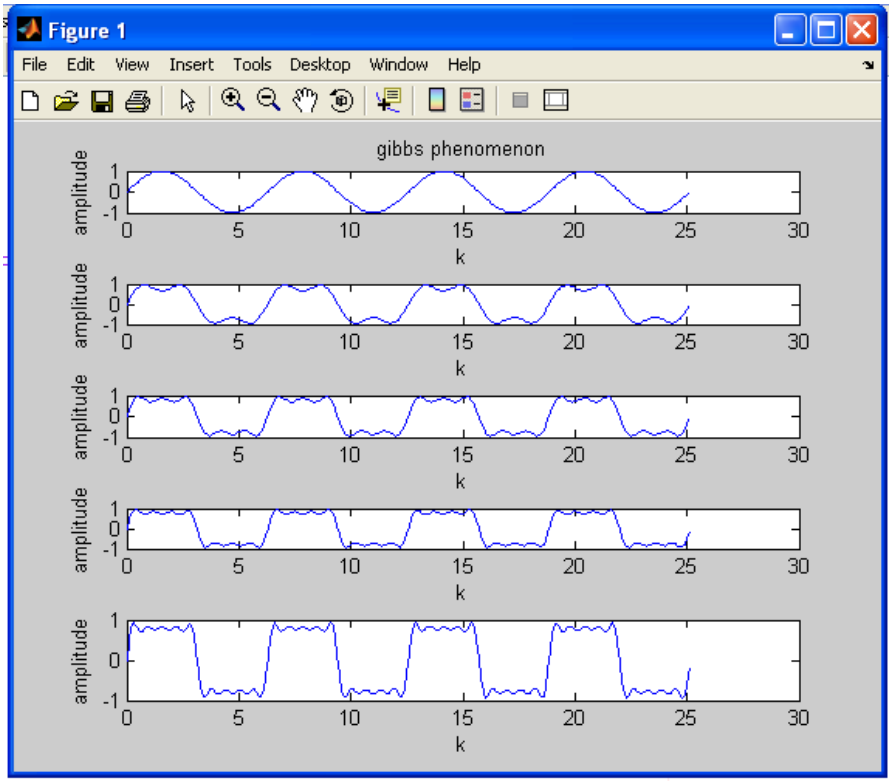
1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

Program:

```
t=0:0.1:(pi*8);
y=sin(t);
subplot(5,1,1);
plot(t,y);
xlabel('k'); ylabel('amplitude');
title('gibbs phenomenon'); h=2;
```

```
%k=3;
for k=3:2:9
y=y+sin(k*t)/k;
subplot(5,1,h);
plot(t,y);
xlabel('k');
ylabel('amplitude');
h=h+1;
end
```

Output:



Result: In this experiment Gibbs phenomenon have been demonstrated Using MATLAB

Viva-Voce Questions:

1. Define Gibb's Phenomenon?
2. What is the importance of Gibb's Phenomenon?
3. What is Static and Dynamic System?
4. Define Fourier series ?
5. What is Causality Condition of the Signal?

2.Gaussian Noise

Aim: To Verify the Gaussian noise.

Software:MATLAB R2015a(8.5.0.197613) version

Procedure:

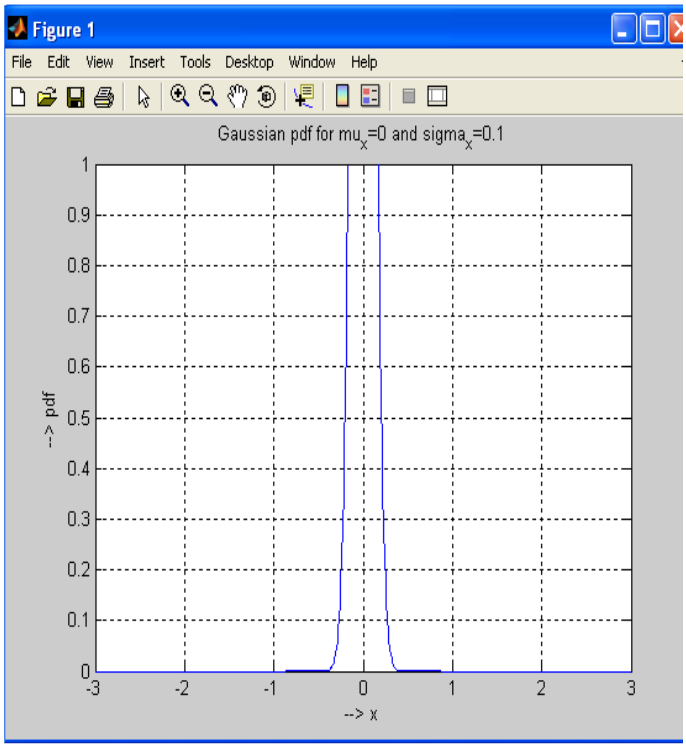
1. Click on the MATLAB icon, then**command window** will be opened.
2. Go to file and open new **Script**.
3. Save the program using specified name other than program name with **.m** extension.
4. **Debug** the program for errors.
5. If errors are present correct them.
6. **Run** the program.
7. Observe for resultant **graphs.**

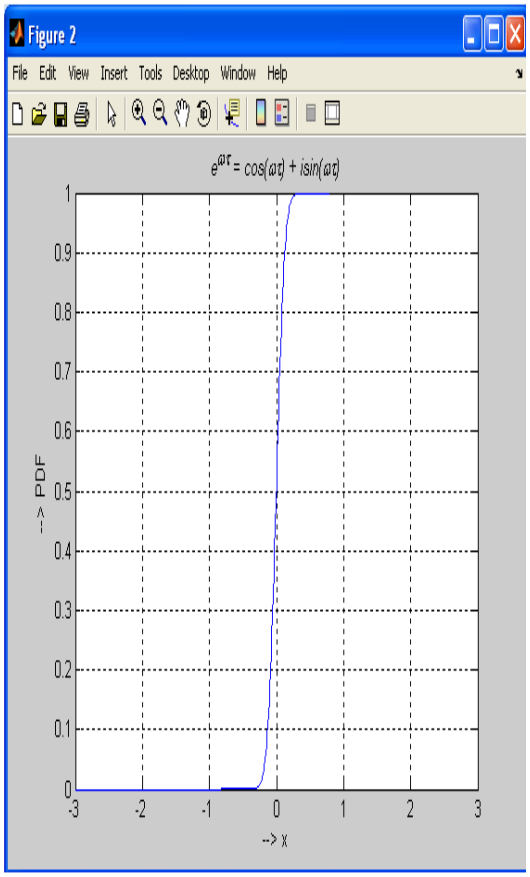
Program:

```
clc;
clear all;
close all;
%% Defining the range for the Random variable dx=0.01;
%delta x x=-3:dx:3;
[m,n]=size(x);
%% Defining the parameters of the pdfmu_x=0;
% mu_x=input('Enter the value of mean');
sig_x=0.1;
% sig_x=input('Enter the value of varience');
%% Computing the probability density function px1=[];
a=1/(sqrt(2*pi)*sig_x);
for j=1:n px1(j)=a*exp([-((x(j)-mu_x)/sig_x)^2]/2);
end
%% Computing the cumulative distribution function
cum_Px(1)=0;
for j=2:n cum_Px(j)=cum_Px(j-1)+dx*px1(j);
end
%% Plotting the results
```

```
figure(1)
plot(x,px1);
grid
axis([-3 3 0 1]);
title(['Gaussian pdf for mu_x=0 and sigma_x=', num2str(sig_x)]);
xlabel('--> x')
ylabel('-->pdf') figure(2)
plot(x,cum_Px);grid
axis([-3 3 0 1]);
title(['Gaussian Probability Distribution Function for mu_x=0 and
sigma_x=',
num2str(sig_x)]);
title('\ite^{\omega\tau} = cos(\omega\tau) + isin(\omega\tau)')
xlabel('--> x')
ylabel('--> PDF
```

Output:





Viva-Voce Questions:

1. What is a noise and how many types of noises are there?
2. What is Gaussian noise?
3. What is correlation? How many types of correlation are there?
4. State Paeseval's energy theorem for a periodic signal?
5. What is Signum function?