



## LABORATORY MANUAL

**B.Tech. Semester- VI**

**DIGITAL SIGNAL PROCESSING LAB**

**Subject code: LC-EE-308G**

**Prepared by:**

Mrs. Dimple Saproo

**Checked by:**

Mrs. Dimple Saproo

**Approved by:**

Name : Prof. (Dr.) Isha Malhotra

**Sign.: .....**

**Sign.: .....**

**Sign.: .....**

**DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING  
DRONACHARYA COLLEGE OF ENGINEERING  
KHENTAWAS, FARRUKH NAGAR, GURUGRAM (HARYANA)**

## Table of Contents

1. Vision and Mission of the Institute
2. Vision and Mission of the Department
3. Programme Educational Objectives (PEOs)
4. Programme Outcomes (POs)
5. Programme Specific Outcomes (PSOs)
6. University Syllabus
7. Course Outcomes (COs)
8. CO- PO and CO-PSO mapping
9. Course Overview
10. List of Experiments
11. DOs and DON'Ts
12. General Safety Precautions
13. Guidelines for students for report preparation
14. Lab assessment criteria
15. Details of Conducted Experiments
16. Lab Experiments

## Vision and Mission of the Institute

### **Vision:**

To impart Quality Education, to give an enviable growth to seekers of learning, to groom them as World Class Engineers and managers competent to match the expending expectations of the Corporate World has been ever enlarging vision extending to new horizons of Dronacharya College of Engineering

### **Mission:**

1. To prepare students for full and ethical participation in a diverse society and encourage lifelong learning by following the principle of 'Shiksha evam Sahayata' i.e. Education & Help.
2. To impart high-quality education, knowledge and technology through rigorous academic programs, cutting-edge research, & Industry collaborations, with a focus on producing engineers& managers who are socially responsible, globally aware, & equipped to address complex challenges.
3. Educate students in the best practices of the field as well as integrate the latest research into the academics.
4. Provide quality learning experiences through effective classroom practices, innovative teaching practices and opportunities for meaningful interactions between students and faculty.
5. To devise and implement programmes of education in technology that are relevant to the changing needs of society, in terms of breadth of diversity and depth of specialization.

## Vision and Mission of the Department

### **Vision:**

Our vision for the Electrical and Electronics Engineering (EEE) Department is to be a globally recognized Centre of excellence in education, research, and innovation in the field of electrical and electronics engineering. We strive to produce competent engineers with strong technical knowledge, ethical values, and a passion for lifelong learning. And also to contribute to the sustainable development of society through cutting-edge research, industry collaborations, and community engagement.

### **Mission:**

- M1. To provide a high-quality education that equips students with a strong foundation in electrical and electronics engineering.
- M2. To conduct pioneering research in diverse areas of electrical and electronics engineering.
- M3. To establish strong ties with industry partners to bridge the gap between academia and the professional world.
- M4. To instilling ethical values, social responsibility, and environmental consciousness in our students.
- M5. To regularly assess and upgrade our teaching methodologies, infrastructure, and facilities.

## **Program Educational Objectives (PEOs)**

PEO1: Engineers will practice the profession of engineering using a systems perspective and analyze, design, develop, optimize & implement engineering solutions and work productively as engineers, including supportive and leadership roles on multidisciplinary teams.

PEO2: Continue their education in leading graduate programs in engineering & interdisciplinary areas to emerge as researchers, experts, educators & entrepreneurs and recognize the need for, and an ability to engage in continuing professional development and life-long learning.

PEO3: Engineers, guided by the principles of sustainable development and global interconnectedness, will understand how engineering projects affect society and the environment.

PEO4: Promote Design, Research, and implementation of products and services in the field of Engineering through Strong Communication and Entrepreneurial Skills.

PEO5: Re-learn and innovate in ever-changing global economic and technological environments of the 21st century.

## Programme Outcomes (POs)

- PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- PO2: Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- PO5 : Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- PO6 :The engineer and society:** Apply reasoning informed by the contextual knowledge to Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- PO7 : Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- PO9 : Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- PO11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- PO12: Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## Program Specific Outcomes (PSOs)

**PSO1:** Equip themselves to potentially rich & employable field of Engineering. Analyze and design electrical machines, circuits, controls and systems which makes the part of Power generation, transmission, distribution, utilization and conservation

**PSO2:** Pursue higher studies in the contemporary Technologies and multidisciplinary fields with an inclination towards continuous learning in the area of Power quality, high voltage, power electronics and Renewable energy systems

**PSO3:** Take up-self- employment in Indian and global software market in designing, implementing and testing analog, digital, embedded and signal processing systems

**PSO4:** Meet the requirements of the Indian Standards and use knowledge in different domains to identify the research gaps and to provide innovative solutions.

## University Syllabus

1. Introduction to MATLAB.
2. Represent basic signals (unit step, unit impulse, ramp, exponential, sine and cosine).
3. To develop program for Z-Transform in MATLAB.
4. To develop program for Convolution of sequences in MATLAB.
5. To develop program for Correlation of sequences in MATLAB.
6. To develop program for DFT & IDFT of two sequences.
7. To develop program for FFT of two Sequences.
8. To develop program for Circular Convolution.
9. To design analog filter (low-pass, high pass, band-pass, band-stop).
10. To design digital IIR filters (low-pass, high pass, band-pass, band-stop).
11. To develop program for Interpolation and Decimation of sequences.
12. To design FIR filters using windows technique.
13. Detection of Signals buried in Noise.
14. Effect of noise on signals in MATLAB.



## Course Outcomes (COs)

After completion of this course, students will be able to :

CO1: Experiment concepts of DSP and its applications using MATLAB Software

CO2: To understand about the basic signal generation

CO3: Calculate linear and circular convolution of discrete sequences.

CO4: Implement Z transform and inverse Z transform of discrete signals

CO5: Model IIR and FIR filter using window techniques

## CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	3	1					2	1	3
CO2	3	3	2	3	1					2	1	3
CO3	3	3	2	3	1					2	1	3
CO4	3	3	2	3	1					2	1	3
CO5	3	3	2	3	1					2	1	3

## CO-PSO Mapping

	PSO1	PSO2	PSO3	PSO4
CO1	3	3		1
CO2	3	3		1
CO3	3	3		1
CO4	3	3		1
CO5	3	3		1

## Course Overview

Digital Signal Processing (DSP) is concerned with the representation of signals as a sequence of numbers and the operations carried out on the signals to extract specific information contained in them. The objective of the laboratory is to enable the students to simulate and experiment with digital signals and systems and apply the theory they have studied in DSP courses. Students can implement digital signal processing algorithms using different computational platforms and DSP tools. They can critically analyze the behavior of their implementation and observe the specific limitations inherent to the computational platform and tools.

The following guidelines should be followed regarding award of marks:

Class Work : 25 Marks

Theory : 25Marks

Total : 50 Marks

Duration of Exam. : 3 Hrs.

General instructions: Practical examination to be conducted immediately after the Pre university examinations covering entire lab experiments given above. Evaluation is a serious process that is to be conducted under the equal responsibility of both the internal and external examiners. The number of candidates evaluated per group should not exceed 5 and in a batch size of not more than 30. Students shall be allowed for the University examination only on submitting the duly certified internal examination record. The external examiner shall endorse the record and conduct the external examination as per guidelines laid by affiliated university

## List of Experiments mapped with Cos

S. No	List of Experiments	Course Outcomes
1.	Introduction to MATLAB.	CO1
2.	Represent basic signals (unit step, unit impulse, ramp, exponential, sine and cosine).	CO2
3.	To develop program for Z-Transform in MATLAB.	CO4
4.	To develop program for Convolution of sequences in MATLAB.	CO3
5.	To develop program for Correlation of sequences in MATLAB.	CO1
6.	To develop program for Circular Convolution.	CO3
7.	To design analog filter (low-pass, high pass, band-pass, band-stop).	CO5
8.	To design digital IIR filters (low-pass, high pass, band-pass, band-stop).	CO5
9.	To design FIR filters using windows technique.	CO5
10.	To develop program for Interpolation and Decimation of sequences.	CO1

## **DOs and DON'Ts**

### **DOs**

1. Login-on with your username and password.
2. Log off the Computer every time when you leave the Lab.
3. Arrange your chair properly when you are leaving the lab.
4. Put your bags in the designated area.
5. Ask permission to print.

### **DON'Ts**

1. Do not share your username and password.
2. Do not remove or disconnect cables or hardware parts.
3. Do not personalize the computer setting.
4. Do not run programs that continue to execute after you log off.
5. Do not download or install any programs, games or music on computer in Lab.
6. Personal Internet use chat room for Instant Messaging (IM) and Sites is strictly prohibited.
7. No Internet gaming activities allowed.
8. Tea, Coffee, Water & Eatables are not allowed in the Computer Lab.

## General Safety Precautions

### Precautions (In case of Injury or Electric Shock)

1. To break the victim with live electric source, use an insulator such as fire wood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.
2. Unplug the risk of faulty equipment. If main circuit breaker is accessible, turn the circuit off.
3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.
4. Immediately call medical emergency and security. Remember! Time is critical; be best.

### Precautions (In case of Fire)

1. Turn the equipment off. If power switch is not immediately accessible, take plug off.
2. If fire continues, try to curb the fire, if possible, by using the fire extinguisher or by covering it with a heavy cloth if possible isolate the burning equipment from the other surrounding equipment.
3. Sound the fire alarm by activating the nearest alarm switch located in the hallway.
4. Call security and emergency department immediately:

**Emergency : 201 (Reception)**

**Security : 231 (Gate No.1)**

## Guidelines to students for report preparation

All students are required to maintain a record of the experiments conducted by them. Guidelines for its preparation are as follows:-

- 1) All files must contain a title page followed by an index page. *The files will not be signed by the faculty without an entry in the index page.*
- 2) Student's Name, Roll number and date of conduction of experiment must be written on all pages.
- 3) For each experiment, the record must contain the following :
  - (i) Aim/Objective of the experiment .
  - (ii) Pre-experiment work (as given by the faculty) .
  - (iii) Lab assignment questions and their solutions.
  - (iv) Test Cases (if applicable to the course) .
  - (v) Results/ output.

**Note:**

1. Students must bring their lab record along with them whenever they come for the lab.
2. Students must ensure that their lab record is regularly evaluated.

## Lab Assessment Criteria

An estimated 10 lab classes are conducted in a semester for each lab course. These lab classes are assessed continuously. Each lab experiment is evaluated based on 5 assessment criteria as shown in following table. Assessed performance in each experiment is used to compute CO attainment as well as internal marks in the lab course.

Grading Criteria	Exemplary (4)	Competent (3)	Needs Improvement (2)	Poor (1)
<b>AC1: Pre-Lab written work (this may be assessed through viva)</b>	Complete procedure with underlined concept is properly written	Underlined concept is written but procedure is incomplete	Not able to write concept and procedure	Underlined concept is not clearly understood
<b>AC2: Program Writing/ Modeling</b>	Assigned problem is properly analyzed, correct solution designed, appropriate language constructs/ tools are applied, Program/solution written is readable	Assigned problem is properly analyzed, correct solution designed, appropriate language constructs/ tools are applied	Assigned problem is properly analyzed & correct solution designed	Assigned problem is properly analyzed
<b>AC3: Identification &amp; Removal of errors/ Troubleshooting</b>	Able to identify errors/ hardwiring and remove them	Able to identify errors/ bugs and remove them with little bit of guidance	Is dependent totally on someone for identification of errors and their removal	Unable to understand the reason for errors even after they are explicitly pointed out
<b>AC4: Execution &amp; Demonstration</b>	All variants of input /output are tested, Solution is well demonstrated and implemented concept is clearly explained	All variants of input /output are not tested, However, solution is well demonstrated and implemented concept is clearly explained	Only few variants of input /output are tested, Solution is well demonstrated but implemented concept is not clearly explained	Solution is not well demonstrated and implemented concept is not clearly explained
<b>AC5: Lab Record Assessment</b>	All assigned problems are well recorded with objective, design constructs and solution along with Performance analysis using all variants of input and output	More than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	Less than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	Less than 40 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input And output

# LAB EXPERIMENTS



## LAB EXPERIMENT No. 1

**OBJECTIVE:-** Introduction to MATLAB.

**BRIEF DESCRIPTION:-** MATLAB stands for Matrix Laboratory. It is a high performance language that is used for technical computing. It was developed by Cleve Moler of the company MathWorks.Inc in the year 1984.It is written in C,C++, Java. It allows matrix manipulations, plotting of functions, and implementation of algorithms and creation of user interfaces. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

### **PRE EXPERIMENT QUESTIONS:**

Q1.What is a MATLAB?.

Q2 What are the various functions of MATLAB?

### **EXPLANATION:**

#### **Getting started with MATLAB:**

It is both a programming language as well as a programming environment. It allows the computation of statements in the command window itself.

- **Command Window:**

In this window one must type and immediately execute the statements, as it requires quick prototyping. These statements cannot be saved. Thus, this is can be used for small, easily executable programs.

- **Editor (Script):**

In this window one can execute larger programs with multiple statements, and complex functions These can be saved and are done with the file extension ‘.m ‘

- **Workspace:**

In this window the values of the variables that are created in the course of the program (in the editor) are displayed.

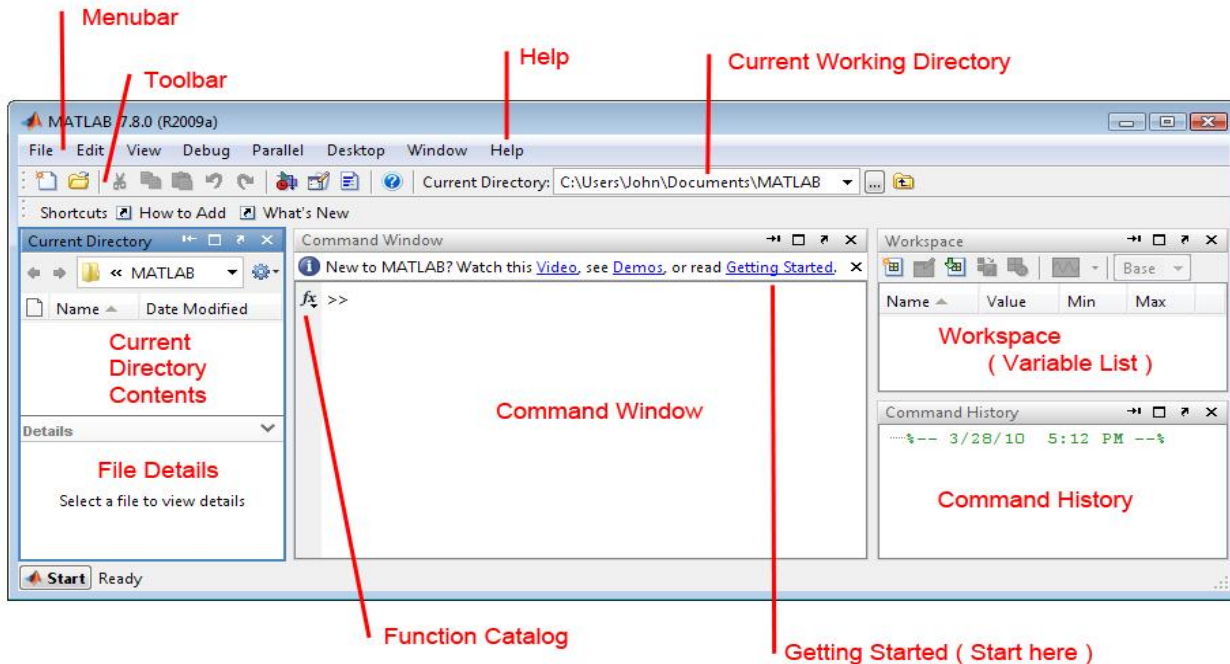
- **Current working directory:**

This window displays the exact location(path) of the program file being created.

MATLAB Library comes with a set of many inbuilt functions. These functions mostly perform mathematical operations like sine, cosine and tangent. They perform more complex functions too like finding the inverse and determinant of a matrix, cross product and dot product.

Although MATLAB is encoded in C, C++ and Java, it is a lot easier to implement than these three languages. For example, unlike the other three, no header files need to be initialised in the beginning of the document and for declaring a variable, the data type need not be provided. It provides an easier alternative for vector operations. They can be performed using one command instead of multiple statements in a for or while loop.

## The MATLAB Work Environment



ILLUSTRATE THE SIMPLE MATHEMATICAL EXPRESSIONS IN MATLAB.

**PROGRAM:-**

```

clc;
clear all;
close all;
x=2^5
y=x-1
z=x/y-1
t=x+y
r= pi^(1/3)-1
area=pi*r^2
a= exp(pi*r)
b=3*sqrt(5)-1
c=(sin(pi/6))^2
A = [1 2 0; 2 5 -1; 4 10 -1]
B = A'
C = A * B
D = A .* B
d = [1 2 3 4 6 4 3 4 5]
e = d + 2
plot(e)
    
```

```

grid on
bar(e)
xlabel('Sample #')
ylabel('Pounds')
plot(e, '*')
axis([0 10 0 10])

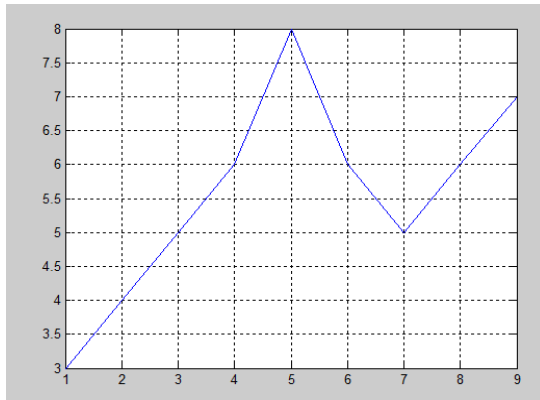
```

**RESULT:-**

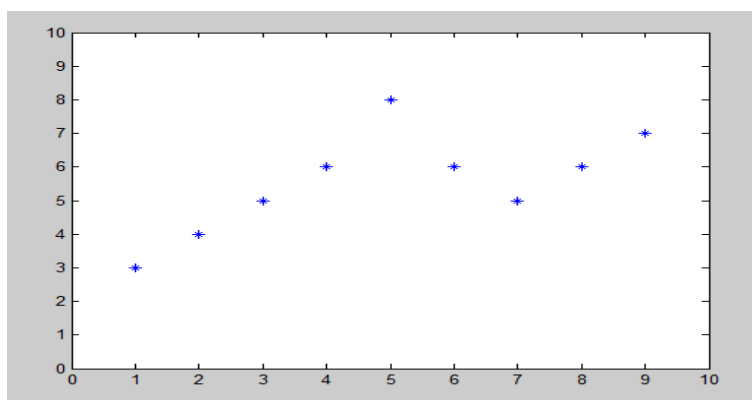
```

x = 32
y = 31
z = 0.0323
t = 63
r = 0.4646
area = 0.6781
a = 4.3041
b = 5.7082
c = 0.2500
A = 1  2  0
    2  5 -1
    4 10 -1
B = 1  2  4
    2  5 10
    0 -1 -1
C = 5 12 24
    12 30 59
    24 59 117
D = 1  4  0
    4 25 -10
    0 -10  1
d=1  2  3  4  6  4  3  4  5
e=3  4  5  6  8  6  5  6  7

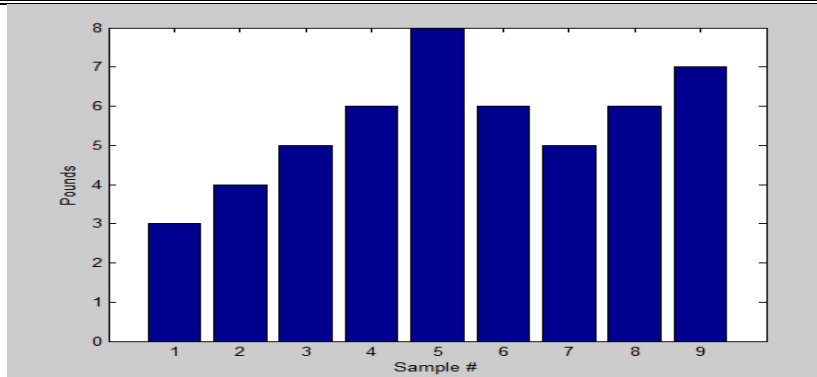
```



**Fig.1**



**Fig. 3**



**Fig.2**

**POST EXPERIMENT QUESTIONS:**

Q1. What is the difference between an array and a vector?

Q2. What are the various windows in MATLAB?

Q3 What are variables?

Q4 What are the various arithmetic operators used in MATLAB?

Q5.What are M-files ?

## LAB EXPERIMENT No. 2

**OBJECTIVE:-** Write a MATLAB program to represent the elementary discrete time signals

- a) unit impulse signal, b) unit step signal, c) ramp signal, and d) exponential signal  
e) sine and f) cosine signal.

**BRIEF DESCRIPTION:** A signal is a physical quantity, or quality, which conveys information.

Electrical engineers normally consider a signal to be an electric current or voltage, and these currents and voltages are functions of time. The step, ramp, impulse, exponential, and sinusoidal functions, etc., are the basic signals. These signals may be combined by addition or subtraction to build a variety of general waveforms used in practice. Signals play an important role in life. They are used in communication, speech processing, medical science, etc.

### **PRE EXPERIMENT QUESTIONS:**

Q.1 What are the basic signals?

Q.2 Explain continuous-time and discrete-time signals.

### **PROGRAM:-**

```

clc;
close all;
clear all;
% a) Unit Impulse signal
n=-3:1:3;
y1=[zeros(1,3),ones(1,1),zeros(1,3)];
subplot(3,3,1);
stem(n,y1);
title('Unit Impulse Signal')
ylabel('Amplitude ---->');
xlabel('(a) n-->');

% b) Unit Step Signal
N=input('Enter the N value');
n=0:1:N-1;
y2=ones(1,N);
subplot(3,3,2);
stem(n,y2);
title('Unit Step Signal')
ylabel('Amplitude of y2-->');
xlabel('(b) n-->');

% (c) Ramp Signal
n1=input('Enter the length of ramp sequence');
```

```

n=0:n1;
subplot(3,3,3);
stem(n,n);
title ('Ramp Signal')
ylabel('Amplitude of y3 -->');
xlabel('(c) n-->');

% (d) Exponential Signal
n2=input('Enter the length of exponential sequence');
n=0:n2;
a=input('Enter the a value');
y4=exp(a*n);
subplot(3,3,4);
stem(n,y4);
title ('Exponential Signal')
ylabel('Amplitude-->');
xlabel('(d) n-->');

% (e) sine signal.
t1=0:0.01:pi;
y5=sin(2*pi*t1);
subplot(3,3,5);
plot(t1, y5);
ylabel('Amplitude--->');
xlabel('(e) t--->');
title('Sinusoidal Signal');

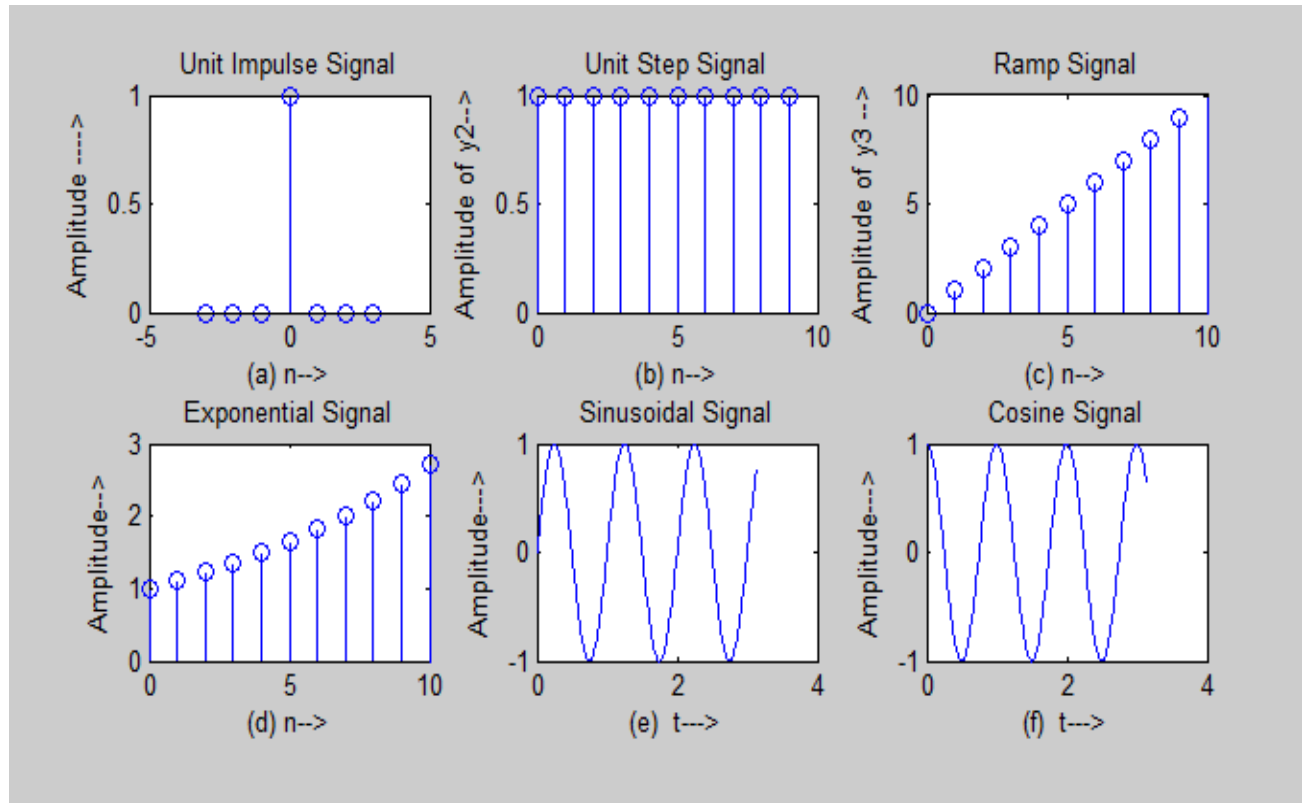
% (f) cosine signal.
t1=0:0.01:pi;
y6=cos(2*pi*t1);
subplot(3,3,6);
plot(t1, y6);
ylabel('Amplitude--->');
xlabel('(f) t--->');
title('Cosine Signal');

```

**RESULT:-**

Enter the N value 10  
Enter the length of ramp sequence 10  
Enter the length of exponential sequence 10  
Enter the a value 0.1

**OUTPUT WAVEFORM:-**



**POST EXPERIMENT QUESTIONS:**

- Q1. Which command is used to draw a continuous waveform?
- Q2. Which command is used to draw a discrete waveform?
- Q3. Which command is used to plot two or more graphs on the same set of axes?
- Q4. Which command is used to label the axes?
- Q5. Which command is used to give a title name to the plot?
- Q6. Which command is used to add a grid to the plot?
- Q7. Which command is used to plot more than one graph on the screen?
- Q8. For autoscaling of the axes which command is used?
- Q9. How the colour of plot can be changed?
- Q10. Which command is used to make the axis equal?

## LAB EXPERIMENT No. 3

**OBJECTIVE:-** To develop MATLAB program for z-transform

**BRIEF DESCRIPTION:** The Z-transform (ZT) is a mathematical tool which is used to convert the difference equations in time domain into the algebraic equations in z-domain.

The Z-transform is a very useful tool in the analysis of a linear shift invariant (LSI) system. An LSI discrete time system is represented by difference equations. To solve these difference equations which are in time domain, they are converted first into algebraic equations in z-domain using the Z-transform, then the algebraic equations are manipulated in z-domain and the result obtained is converted back into time domain using the inverse Z-transform.

### **PRE EXPERIMENT QUESTIONS:**

- Q.1 Explain Properties of Z Transform.  
Q.2 What is region of convergence (ROC)?

### **PROGRAM:-**

```
%(a) without function
clc;
clear all;
close all;
%Direct Z-transform
x=[1 2 3 4 5];
l=length(x);
X=0; % To initialize the output
z=sym('z', 'positive'); % It defines the symbol, The sym and syms commands allows to
set up assumptions for symbolic variables.
for i=0:l-1
    X=X+x(i+1)*z^(-i);
end
disp('Displaying Output');
disp(X);

%(b) using functions
clc;
clear all;
close all;
syms('n','Positive')
f=sin(n);
r=ztrans(f);
r=ztrans(sin(n))
disp('z transform using function');
disp(r);
syms ('p','q','Positive')
f1=exp(p+q);
```



```
rr=ztrans(f1);  
disp('z transform of exponential function');  
disp(rr);
```

**Results:**

- (a) Displaying Output  
 $2/z + 3/z^2 + 4/z^3 + 5/z^4 + 1$
- (b)  $r = (z \cdot \sin(1)) / (z^2 - 2 \cdot \cos(1) \cdot z + 1)$   
z transform using function  
 $(z \cdot \sin(1)) / (z^2 - 2 \cdot \cos(1) \cdot z + 1)$
- z transform of exponential function  
 $(z \cdot \exp(p)) / (z - \exp(1))$

**POST EXPERIMENT QUESTIONS:**

- Q.1 Define Z-transform of unit step signal.
- Q.2 Z-transform of various signals?

## **LAB EXPERIMENT No. 4**

**OBJECTIVE:-** To develop program for Convolution of sequences in MATLAB.

**BRIEF DESCRIPTION:** The convolution of two signals in the time domain is equivalent to the multiplication of their representation in frequency domain.

Mathematically, we can write the convolution of two signals as

$$Y(t) = x_1(t) * x_2(t)$$

### **PRE EXPERIMENT QUESTIONS:**

Q.1 What is convolution?

Q.2 What are the applications of convolution?

### **PROGRAM:-**

```

clc;
clear all;
close all;
x=input('Enter the first sequence');
h=input('Enter the second sequence');
y=conv(x,h);
subplot(3,1,1);
stem(x);
title('First Sequence');
ylabel('Amplitude---->');
xlabel('(a) n--->');
subplot(3,1,2);
stem(h);
title('Second Sequence');
ylabel('Amplitude--->');
xlabel('(b) n--->');
subplot(3,1,3);
stem(y);
title('Output Sequence');
ylabel('Amplitude--->');
xlabel('(c) n--->');
disp('The resultant signal is');y

```

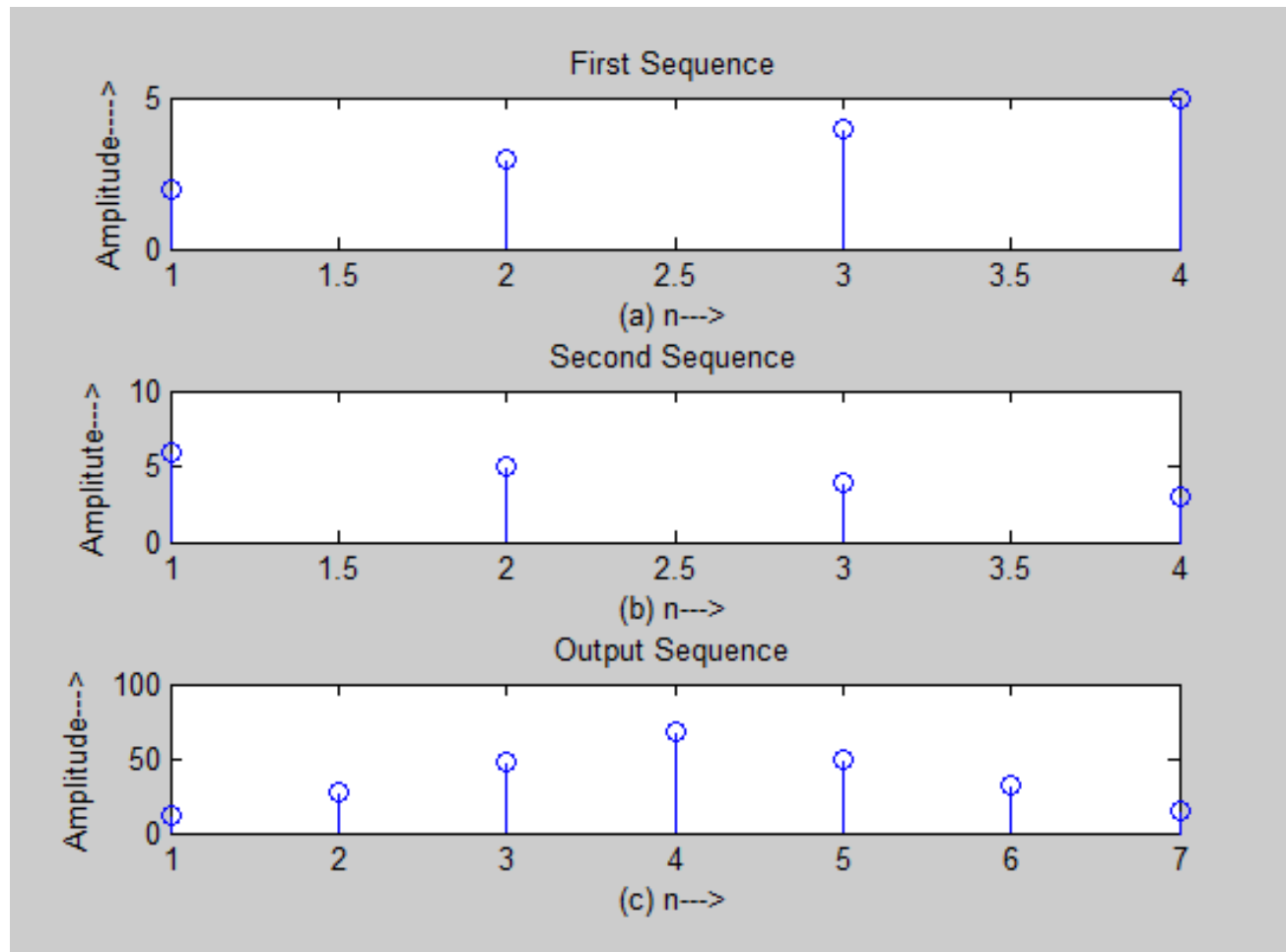
### **RESULT:-**

Enter the first sequence[2,3,4,5]

Enter the second sequence[6,5,4,3]

The resultant signal is

y =

**OUTPUT WAVEFORM:-****POST EXPERIMENT QUESTIONS:**

- Q1. Generate a simple convolution matrix.
- Q2. What is a convolution matrix?
- Q3. What is the symbol for convolution?

## LAB EXPERIMENT No. 5

**OBJECTIVE:-** To develop program for Correlation of sequences in MATLAB

**BRIEF DESCRIPTION:**

**Correlation** measures the relationship, or association, between two variables by looking at how the variables change with respect to each other. Statistical correlation also corresponds to simultaneous changes between two variables, and it is usually represented by linear relationships. Importantly, correlation does not necessarily mean causation. This is because a correlation describes how two or more variables are related, and not whether they cause changes in one another.

**PRE EXPERIMENT QUESTIONS:**

- Q.1 What is correlation?
- Q.2 What are the applications of correlation?

**PROGRAM:-**

```
clc;
clear all;
close all;
x=input('Enter the first sequence');
y=input('Enter the second sequence');
z=xcorr(x,y);
subplot(3,1,1);
stem(x);
title('First Sequence');
ylabel('Amplitude---->');
xlabel('(a) n--->');
subplot(3,1,2);
stem(y);
title('Second Sequence');
ylabel('Amplitude--->');
xlabel('(b) n--->');
subplot(3,1,3);
stem(z);
title('Output Sequence');
ylabel('Amplitude--->');
xlabel('(c) n--->');
disp('The resultant signal is');
disp(z)
fliplr(z)
```

**RESULT:-**

Enter the first sequence[1 2 3 5]

Enter the second sequence[2 3 5 6]

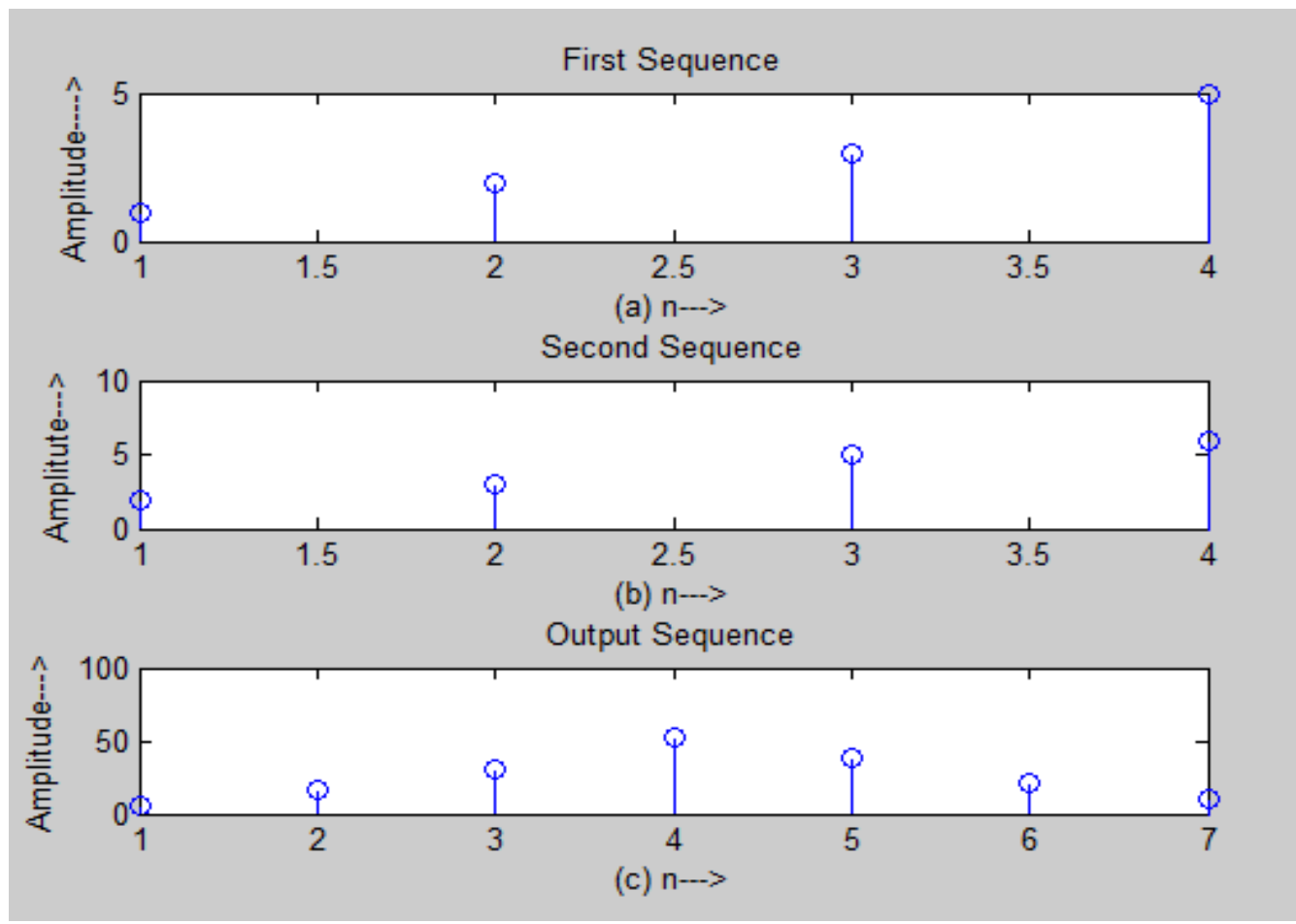
The resultant signal is

6.0000 17.0000 31.0000 53.0000 38.0000 21.0000 10.0000

ans =

10.0000 21.0000 38.0000 53.0000 31.0000 17.0000 6.0000

**OUTPUT WAVEFORM:-**



**QUESTIONS/ANSWERS**

Q1. What is auto-correlation?

Q2. What is cross-correlation?

Q3 Which command is used to find the correlation coefficients

Q4 Which command is used to find correlation between two sequences?

Q5. What command is used to find cross-correlation of two matrices?

Q6. Is convolution obeys the algebraic properties?

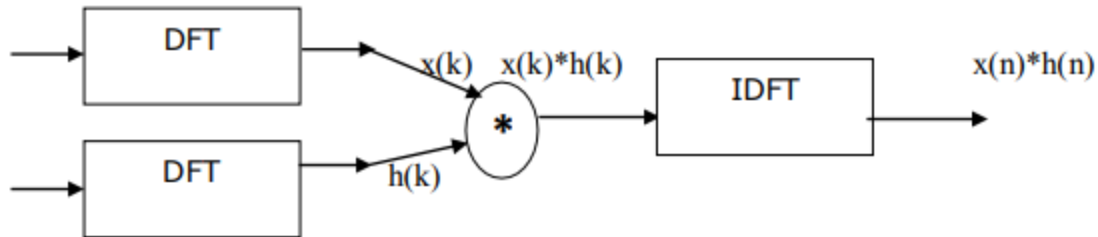
Q7. What is a convolution matrix?

Q8. Generate a simple convolution matrix.

## LAB EXPERIMENT No. 6

**OBJECTIVE:-** . To develop program for Circular Convolution.

**BRIEF DESCRIPTION:** Multiplication of two sequences in time domain is called as Linear convolution while Multiplication of two sequences in frequency domain is called as circular convolution. Results of both are totally different but are related with each other.



There are two different methods are used to calculate circular convolution

- 1) Graphical representation form
- 2) Matrix approach

### **PRE EXPERIMENT QUESTIONS:**

- Q1. What is the difference between circular convolution and discrete convolution?
- Q2. What is deconvolution?

### **PROGRAM:-**

```

clc;
close all;
clear all;
x1=input('Enter the input sequence1: ');
n1=input('Enter the time interval for input sequence 1:');
x2=input('Enter the input sequence 2: ');
n2=input('Enter the time interval for input sequence 2: ');
n=min(n1):1:max(n1);
y=cconv(x1,x2,4);
disp(x1);
disp(x2);
disp(y);
subplot(3,1,1);
stem(n1,x1);
xlabel('Time');
ylabel('Amplitude');
title('Input sequence 1');

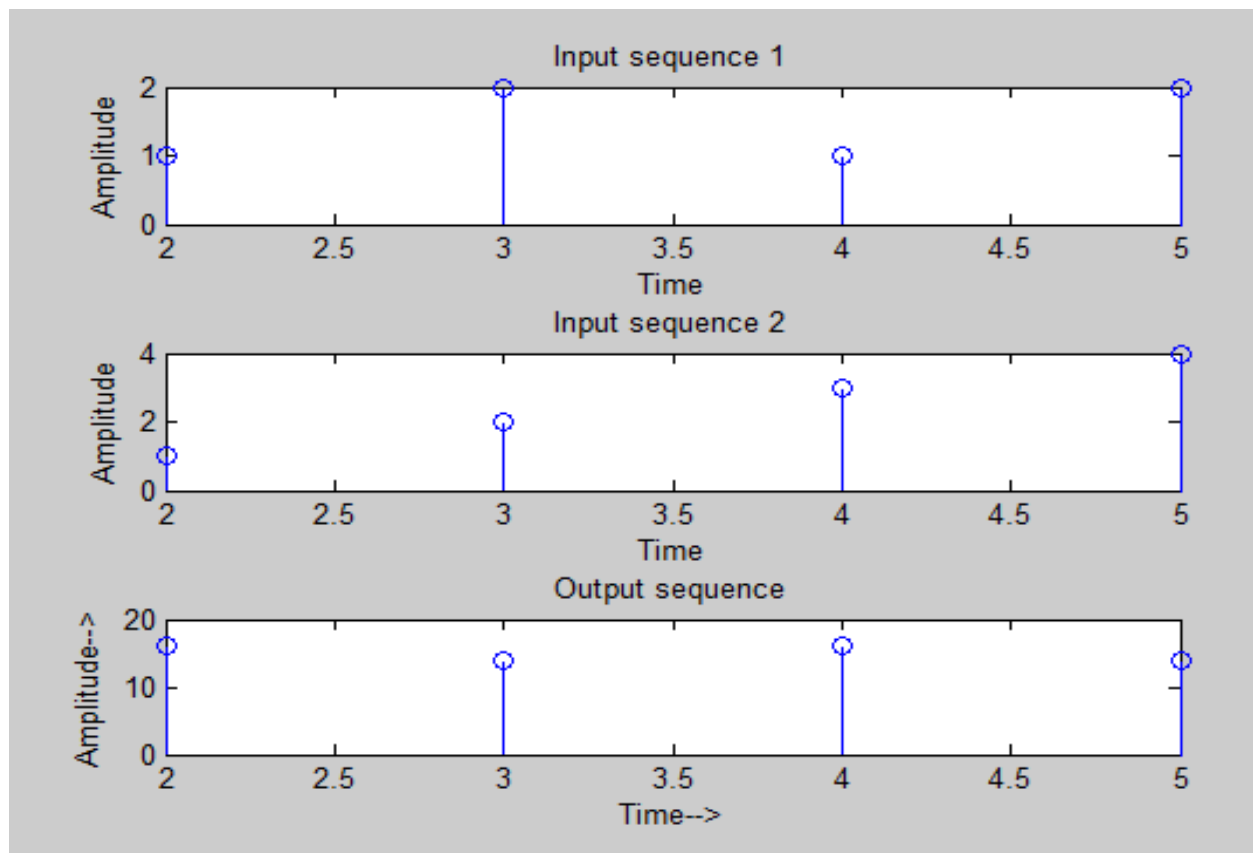
```

```
subplot(3,1,2);
stem(n2,x2);
xlabel('Time');
ylabel('Amplitude');
title('Input sequence 2');
subplot(3,1,3);
stem(n,y);
xlabel('Time-->');
ylabel('Amplitude-->');
title('Output sequence');
```

**RESULT:-**

Enter the input sequence1: [1 2 1 2]  
 Enter the time interval for input sequence 1:[2 3 4 5]  
 Enter the input sequence 2: [1 2 3 4]  
 Enter the time interval for input sequence 2: [2 3 4 5]  
 1 2 1 2  
 1 2 3 4  
 16 14 16 14

**OUTPUT WAVEFORM:-**





**POST EXPERIMENT QUESTIONS:**

Q1. Which command is used to find convolution between two sequences?

Q2. Is convolution obeys the algebraic properties?

## **LAB EXPERIMENT No. 7**

**OBJECTIVE:-** - To design analog filter (low-pass, high pass, band-pass, band-stop).

**BRIEF DESCRIPTION:** Filters are networks that process signals in a frequency-dependent manner. The basic concept of a filter can be explained by examining the frequency dependent nature of the impedance of capacitors and inductors. Analog filter is typically used in electronics and is considered as a basic building block of signal processing. These analog filters are used to separate audio signals before applying to loudspeakers. To separate and to combine several telephone conversations onto a single channel can be done using analog filters. To select a particular radio station from the radio receiver by rejecting all other channels can be done using analog filters.

### **PRE EXPERIMENT QUESTIONS:**

- Q.1 What is analog filter in DSP?  
Q.2 Different type of Analog filter ?

### **PROGRAM:-**

#### **Low Pass: -**

%Program for the design of butterworth low pass analog filter

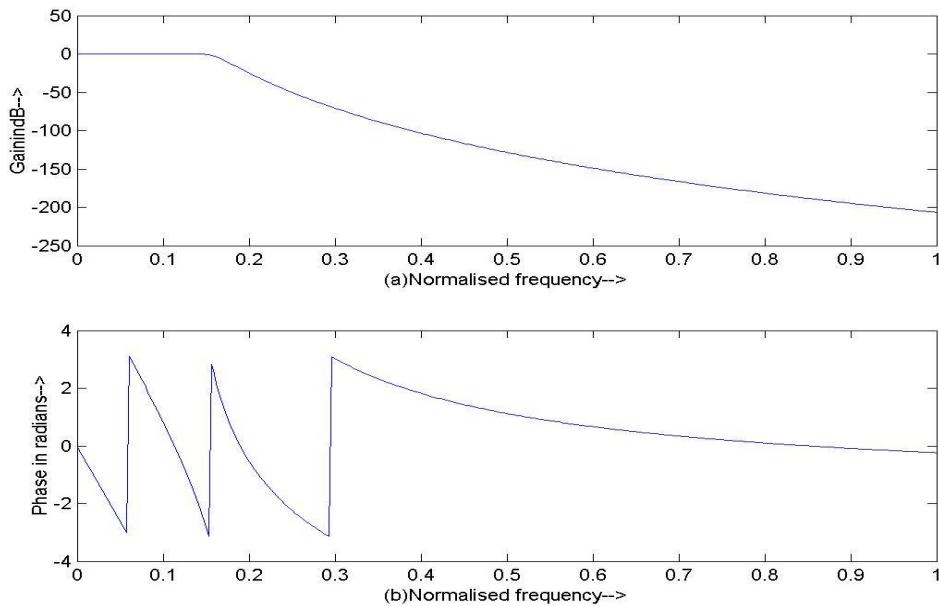
```
clc;
close all;
clear all;
format long
rs=input('enter the stopband ripple...');
rp=input('enter the passband ripple...');
ws=input('enter the stopband freq...');
wp=input('enter the passband freq...');
fs=input('enter the sampling freq...');
w1=2*wp/fs;
w2=2*ws/fs;
[n,wn]=buttord(w1,w2,rp,rs);
[z,p,k]=butter(n,wn);
[b,a]=zp2tf(z,p,k);
```

```
[b,a]=butter(n,wn);  
w=0:0.01:pi;  
[h,om]=freqs(b,a,w);  
m=20*log10(abs(h));  
an=angle(h);  
subplot(2,1,1);  
plot(om/pi,m);  
ylabel('GainindB-->');  
xlabel('(a)Normalised frequency-->');  
subplot(2,1,2);  
plot(om/pi,an)  
xlabel('(b)Normalised frequency-->');  
ylabel('Phase in radians-->');
```

**Output: -**

enter the stopband ripple...60  
enter the passband ripple...0.15  
enter the stopband freq...3000  
enter the passband freq...1500  
enter the sampling freq...7000

**OUTPUT WAVEFORM:-**



**High Pass: -**

%Program for the design of butterworth high pass analog filter

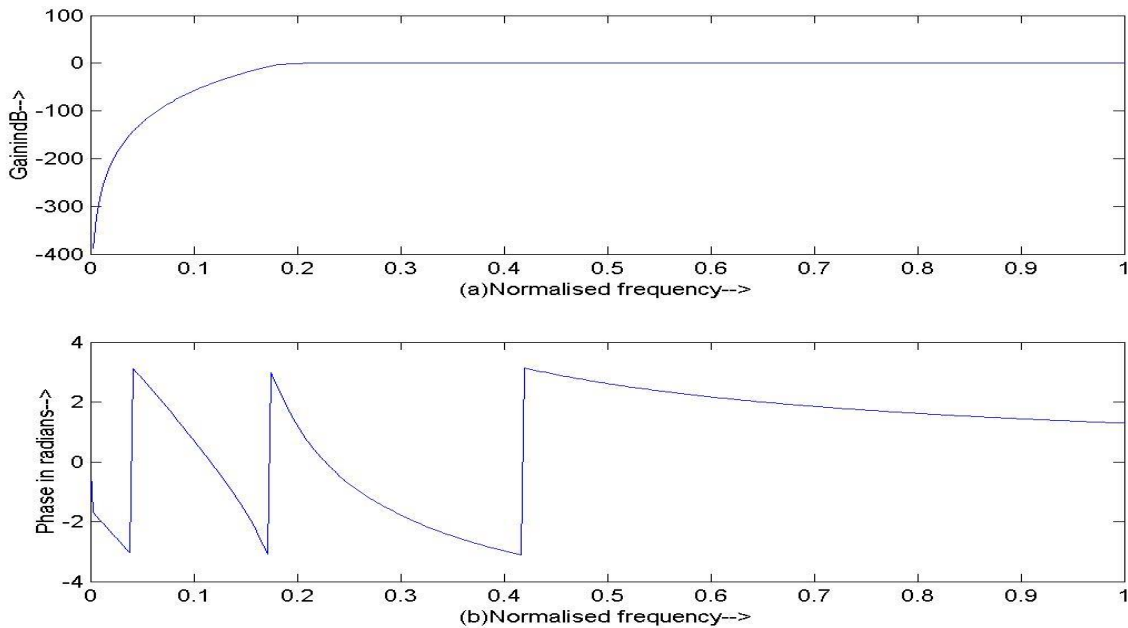
```

clc;
close all;
clear all;
format long
rp=input('enter the passband ripple...');
rs=input('enter the stopband ripple...');
wp=input('enter the passband freq.....');
ws=input('enter the stopband freq.....');
fs=input('enter the sampling freq.....');
w1=2*wp/fs;
w2=2*ws/fs;
[n,wn]=buttord(w1,w2,rp,rs,'s');
[b,a]=butter(n,wn,'high','s');
w=0:0.01:pi;
    
```

```
[h,om]=freqs(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('GainindB-->');
xlabel('(a)Normalised frequency-->');
subplot(2,1,2);
plot(om/pi,an)
xlabel('(b)Normalised frequency-->');
ylabel('Phase in radians-->');
```

**Output: -**

enter the passband ripple...0.2  
 enter the stopband ripple...40  
 enter the passband freq.....2000  
 enter the stopband freq.....3500  
 enter the sampling freq.....8000



**Band Pass: -**

%Program for the design of butterworth band pass analog filter

```

clc;
close all;
clear all;
format long
rp=input('enter the passband ripple...');
rs=input('enter the stopband ripple...');
wp=input('enter the passband freq...');
ws=input('enter the stopband freq...');
fs=input('enter the sampling freq...');
w1=2*wp/fs;
w2=2*ws/fs;
[n]=buttord(w1,w2,rp,rs,'s');
wn=[w1 w2];
[b,a]=butter(n,wn,'bandpass','s');
w=0:0.01:pi;
[h,om]=freqs(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('GainindB-->');
xlabel('(a)Normalised frequency-->');
subplot(2,1,2);
plot(om/pi,an)
xlabel('(b)Normalised frequency-->');
ylabel('Phase in radians-->');

```

**Output: -**

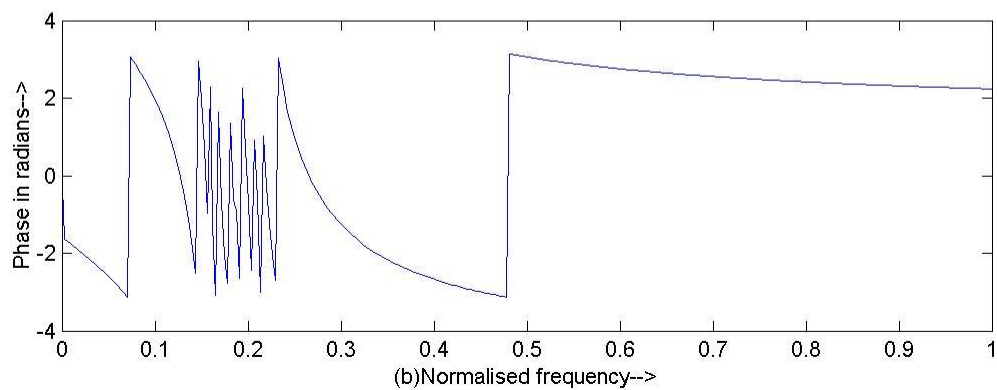
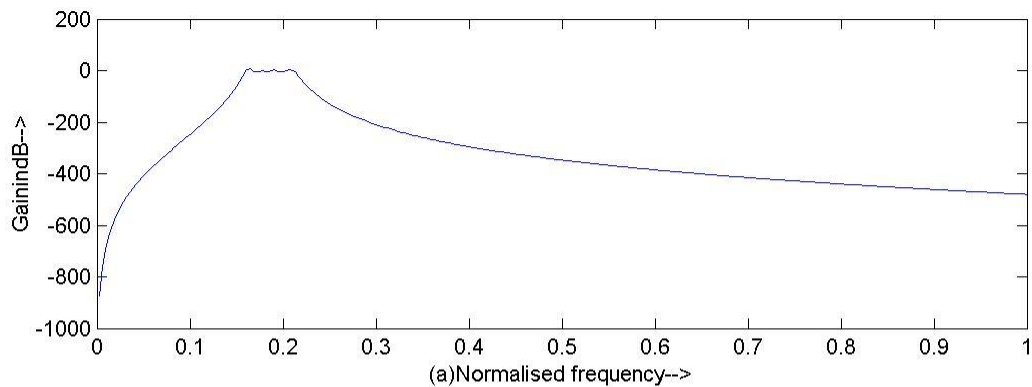
enter the passband ripple...0.36

enter the stopband ripple...36

enter the passband freq...1500

enter the stopband freq...2000

enter the sampling freq...6000



**Band Stop: -**

%Program for the design of butterworth band pass analog filter

clc;

close all;

clear all;

format long

rp=input('enter the passband ripple...');

```

rs=input('enter the stopband ripple...');
wp=input('enter the passband freq...');
ws=input('enter the stopband freq...');
fs=input('enter the sampling freq...');
w1=2*wp/fs;
w2=2*ws/fs;
[n]=buttord(w1,w2,rp,rs,'s');
wn=[w1 w2];
[b,a]=butter(n,wn,'stop','s');
w=0:0.01:pi;
[h,om]=freqz(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('Gain in dB-->');
xlabel('(a)Normalised frequency-->');
subplot(2,1,2);
plot(om/pi,an)
xlabel('(b)Normalised frequency-->');
ylabel('Phase in radians-->');

```

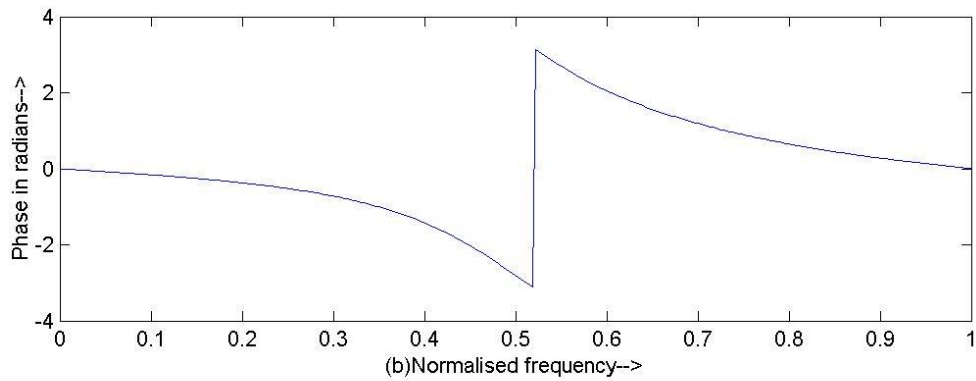
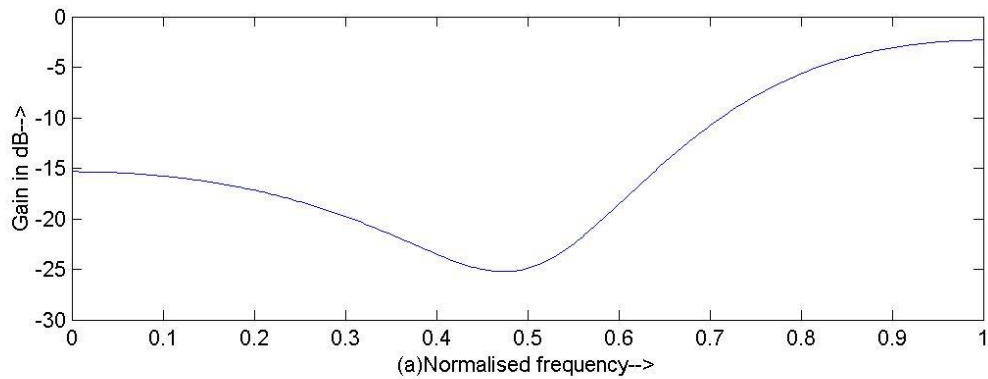
**Output: -**

```

enter the passband ripple...0.28
enter the stopband ripple...28
enter the passband freq...1000
enter the stopband freq...14000
enter the sampling freq...5000

```





**POST EXPERIMENT QUESTIONS:**

- Q.1 Explain different types of filters.
- Q.2 Which of the following methods are used to convert analog filter into digital filter?
- Q.3 Explain the analog filter design methods.

## **LAB EXPERIMENT No. 8**

**OBJECTIVE:-** To design digital filters (low pass, high pass, band pass, band stop).

**BRIEF DESCRIPTION:** A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialised DSP (Digital Signal Processor) chip. Digital filters are used for two general purposes:

(1) separation of signals that have been combined, and (2) restoration of signals that have been distorted in some way. Analog (electronic) filters can be used for these same tasks; however, digital filters can achieve far superior results.

### **PRE EXPERIMENT QUESTIONS:**

Q.1 What is digital filters in DSP?

Q.2 How digital filters are classified.

### **PROGRAMS:-**

#### **Low Pass: -**

%program for the design of chebyshev lowpass filter

clc;

close all;

clear all;

format long

rs=input('enter the stopband ripple...');

rp=input('enter the passband ripple...');

ws=input('enter the stopband freq...');

wp=input('enter the passband freq...');

fs=input('enter the sampling freq...');

w1=2\*wp/fs;

w2=2\*ws/fs;

[n,wn]=cheb1ord(w1,w2,rp,rs);

[b,a]=cheby1(n,rp,wn);

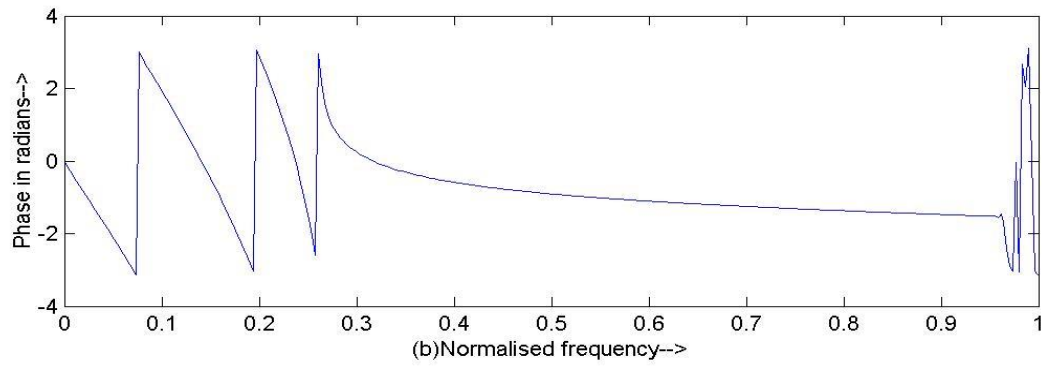
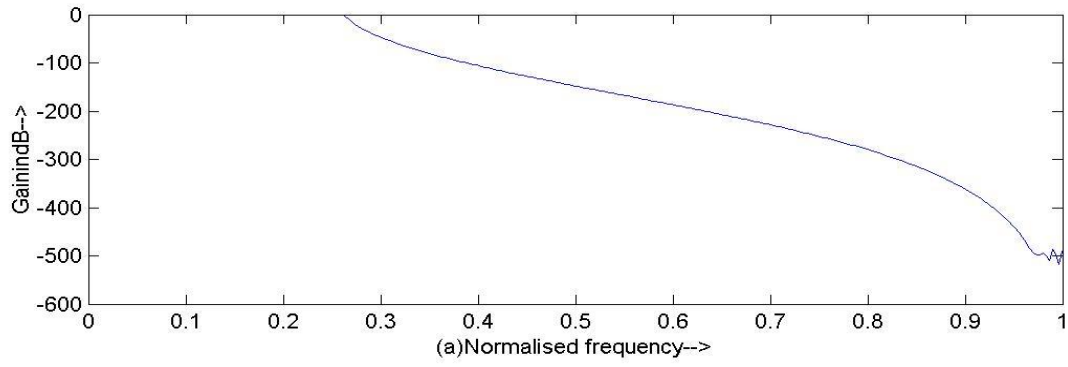
w=0:0.01:pi;

[h,om]=freqz(b,a,w);

```
m=20*log10(abs(h));  
an=angle(h);  
subplot(2,1,1);  
plot(om/pi,m);  
ylabel('GainindB-->');  
xlabel('(a)Normalised frequency-->');  
  
subplot(2,1,2);  
plot(om/pi,an)  
xlabel('(b)Normalised frequency-->');  
ylabel('Phase in radians-->');
```

**Output: -**

enter the stopband ripple...45  
enter the passband ripple...0.2  
enter the stopband freq...1500  
enter the passband freq...1300  
enter the sampling freq...10000



**High Pass: -**

```
%program for the design of chebyshev high pass filter
```

```
clc;
```

```
close all;
```

```
clear all;
```

```
format long
```

```
rp=input('enter the passband ripple...');
```

```
rs=input('enter the stopband ripple...');
```

```
wp=input('enter the passband freq.....');
```

```
ws=input('enter the stopband freq.....');
```

```
fs=input('ente the sampling freq.....');
```

```
w1=2*wp/fs;
```

```
w2=2*ws/fs;
```

```
[n,wn]=cheb1ord(w1,w2,rp,rs,'s');
```

```
[b,a]=cheby1(n,rp,wn,'high','s');
```

```
w=0:0.01:pi;
```

```
[h,om]=freqs(b,a,w);
```

```
m=20*log10(abs(h));
```

```
an=angle(h);
```

```
subplot(2,1,1);
```

```
plot(om/pi,m);
```

```
ylabel('GainindB-->');
```

```
xlabel('(a) Nomralised frequency-->');
```

```
subplot(2,1,2);
```

```
plot(om/pi,an);
```

```
xlabel('(b) Normalised frequency')
```

```
ylabel('Phase in radians')
```

**Output: -**

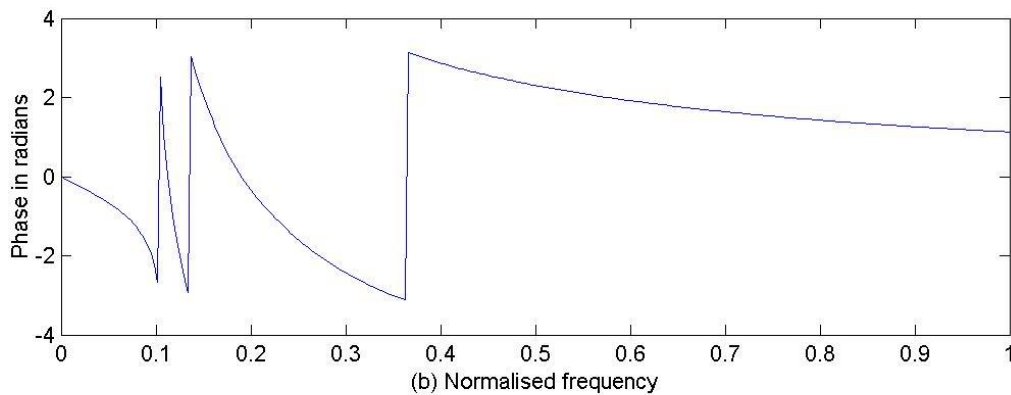
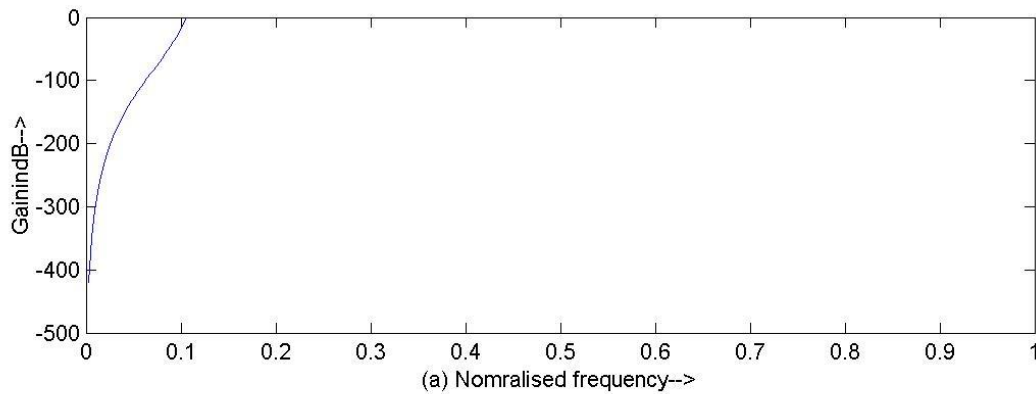
enter the passband ripple...0.3

enter the stopband ripple...60

enter the passband freq.....1500

enter the stopband freq.....2000

enter the sampling freq.....9000



**Band Pass: -**

%program for the design of chebyshev bandpass filter

clc;

close all;

clear all;

format long

rp=input('enter the passband ripple...');

```

rs=input('enter the stopband ripple...');
wp=input('enter the passband freq...');
ws=input('enter the stopband freq...');
fs=input('enter the sampling freq...');
w1=2*wp/fs;
w2=2*ws/fs;
[n]=cheb1ord(w1,w2,rp,rs,'s');
wn=[w1 w2];
[b,a]=cheby1(n,rp,wn,'bandpass','s');
w=0:0.01:pi;
[h,om]=freqs(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('GainindB-->');
xlabel('(a)Normalised frequency-->');
subplot(2,1,2);
plot(om/pi,an)
xlabel('(b)Normalised frequency-->');
ylabel('Phase in radians-->');

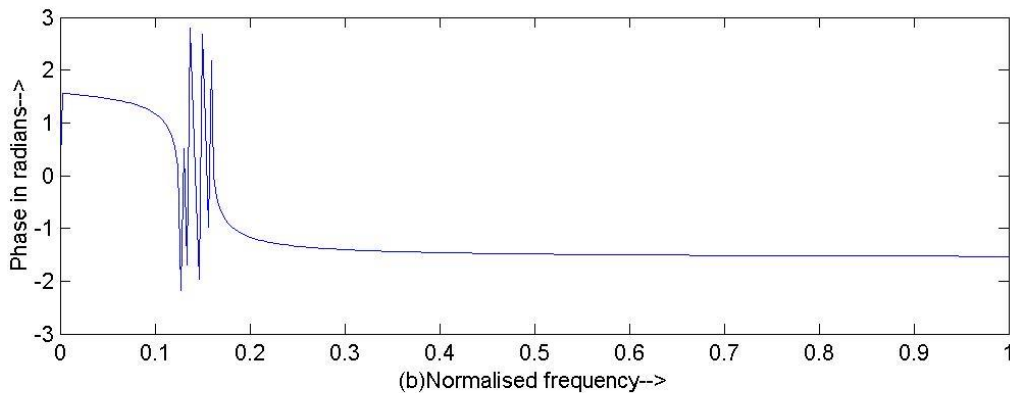
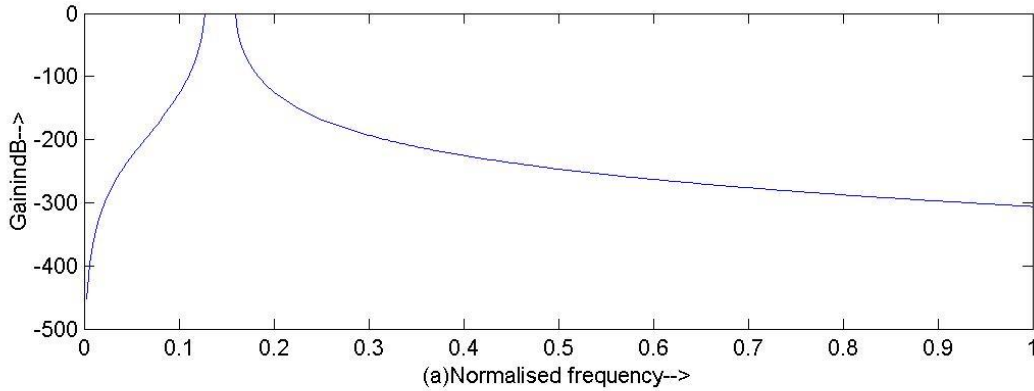
```

**Output: -**

```

enter the passband ripple...0.4
enter the stopband ripple...35
enter the passband freq...2000
enter the stopband freq...2500
enter the sampling freq...10000

```



**Band Stop: -**

%program for the design of chebyshev bandstop filter

clc;

close all;

clear all;

format long

rs=input('enter the stopband ripple...');

rp=input('enter the passband ripple...');

ws=input('enter the stopband freq...');

wp=input('enter the passband freq...');

fs=input('enter the sampling freq...');

w1=2\*wp/fs;

w2=2\*ws/fs;

[n]=cheb1ord(w1,w2,rp,rs);

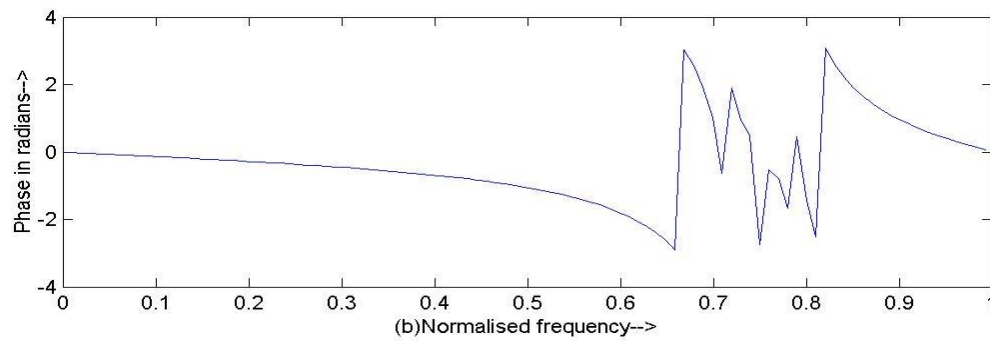
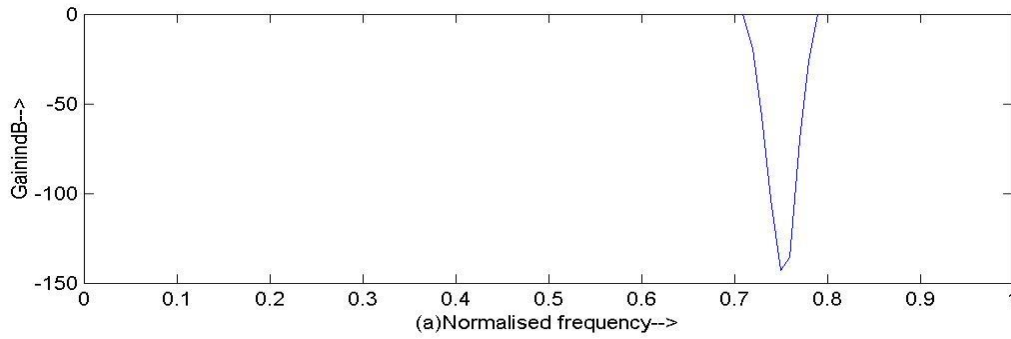
wn=[w1 w2];



```
[b,a]=cheby1(n,rp,wn,'stop');  
w=0:0.1/pi:pi;  
[h,om]=freqz(b,a,w);  
m=20*log10(abs(h));  
an=angle(h);  
subplot(2,1,1);  
plot(om/pi,m);  
ylabel('GainindB-->');  
xlabel('(a)Normalised frequency-->');  
subplot(2,1,2);  
plot(om/pi,an);  
xlabel('(b)Normalised frequency-->');  
ylabel('Phase in radians-->');
```

**Output: -**

```
enter the stopband ripple...40  
enter the passband ripple...0.25  
enter the stopband freq...2750  
enter the passband freq...2500  
enter the sampling freq...7000
```



**POST EXPERIMENT QUESTIONS:**

- Q.1 Differentiate FIR and IIR filter concepts.
- Q.2 What is the order of a recursive (IIR) digital filter?

## LAB EXPERIMENT No. 9

**OBJECTIVE:-** . To design FIR filters using windows technique.

**BRIEF DESCRIPTION:** The window method for digital filter design is fast, convenient, and robust, but generally suboptimal. Digital FIR filters have many favourable properties, which is why they are extremely popular in digital signal processing. One of these properties is that they may exhibit linear phase, which means that signals in the passband will suffer no dispersion. Dispersion occurs when different frequency components of a signal have a different delay through a system. The simplest design method for FIR filters is impulse response truncation (IRT), but unfortunately it has undesirable frequency-domain characteristics, owing to the Gibb's phenomenon. The second design method for a FIR filter is the windowing technique. The windowing method can be used to mitigate the adverse effects of impulse response truncation.

It is easily understood in terms of the convolution theorem for Fourier transforms, making it instructive to study after the Fourier theorems and windows for spectrum analysis. Tapering better preserves the shape of the desired frequency response, as we will see. By choosing the window carefully, we can manage various trade-offs so as to maximize the filter-design quality in a given application.

### **PRE EXPERIMENT QUESTIONS:**

**Q.1 What are the merits and demerits of FIR filters?**

**Q.2 What are the properties of FIR filters?**

### **PROGRAM:-**

```

clc;
clear all;
close all;
rp= input('Enter the passband ripple');
rs=input('Enter the stopband ripple');
fp=input('Enter the passband frequency');
fs=input('Enter the stopband frequency');
f=input('Enter the sampling frequency');
wp=2*fp/f;
ws=2*fs/f;
num=-20*log(sqrt(rp*rs))-13;
dem=14.6*(fs-fp)/f;
n=ceil(num/dem); %B = ceil(A) rounds the elements of A to the nearest
                %integers greater than or equal to A. For complex A,
                %the imaginary and real parts are rounded independently.
n1=n+1;
if(rem(n,2)~=0)
    n1=n;

```

```
n=n-1;
end
y=boxcar(n1);%specify win using windows from the Signal Processing Toolbox
%software, such as boxcar, hamming, hanning, bartlett,
%blackman, kaiser, or chebwin.

%LOW PASS FILTER
b=fir1(n,wp,y); %fir1 implements the method of windowed linear-phase FIR
%digital filter design
[h,o]=freqz(b,1,256); %freqz returns the instantaneous frequency response
%based on the current filter coefficients.
m=20*log10(abs(h));
subplot(2,2,1);
plot(o/pi,m);
title('Low pass filter response');
ylabel('Gain in dB-->');
xlabel('(a) Normalized frequency--->');

%HIGH PASS FILTER
b=fir1(n,wp,'high',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,2);
plot(o/pi,m);
```

```

title('High pass filter response');

ylabel('Gain in dB-->');
xlabel('(b) Nomalized frequency--->');

%BAND PASS FILTER
wn=[wp ws];
b=fir1(n,wn,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,3);
plot(o/pi,m);
title('Band pass filter response');
ylabel('Gain in dB-->');
xlabel('(c) Nomalized frequency--->');

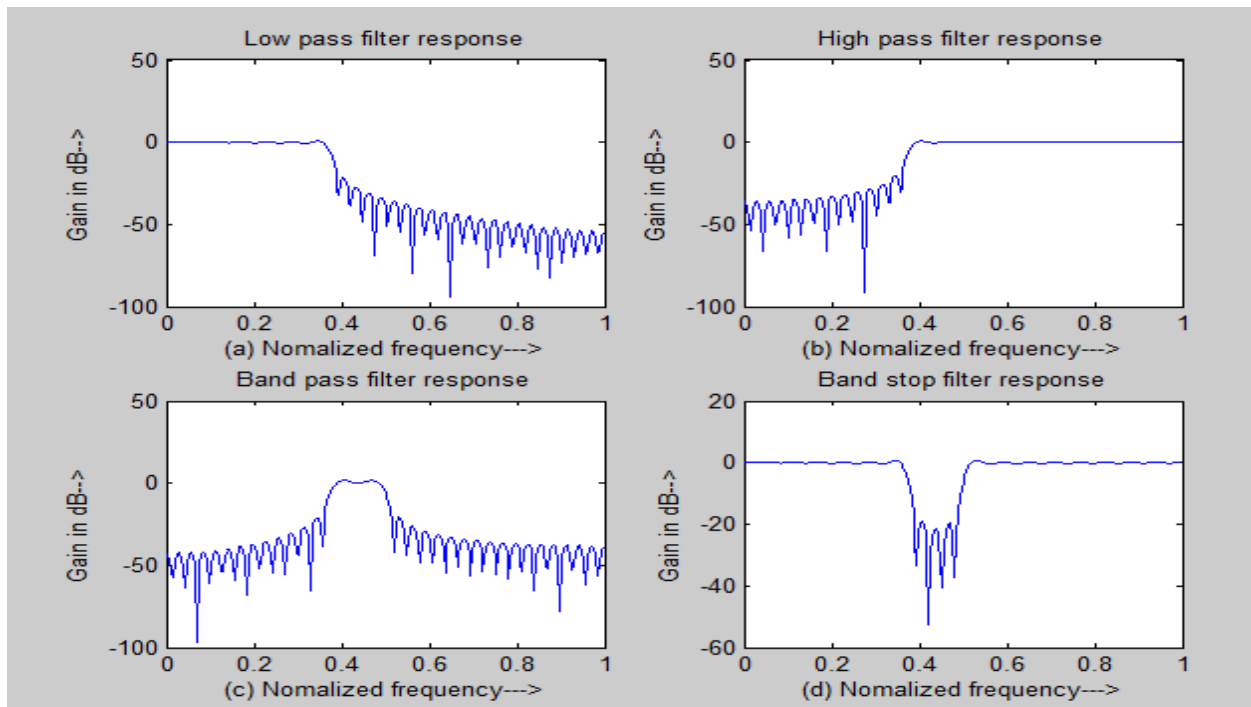
%BAND STOP FILTER
b=fir1(n,wn,'stop',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,4);
plot(o/pi,m);
title('Band stop filter response');
ylabel('Gain in dB-->');
xlabel('(d) Nomalized frequency--->');

```

### **RESULT:-**

Enter the passband ripple 0.02  
 Enter the stopband ripple 0.03  
 Enter the passband frequency 1500  
 Enter the stopband frequency 2000  
 Enter the sampling frequency 8000

**OUTPUT WAVEFORM:-**



**Q.1** What is the condition satisfied by linear phase FIR filter?

**Q.2** What are the desirable characteristics of the frequency response of window function?

**Q.3** List the characteristic of FIR filter designed using window.

## LAB EXPERIMENT No. 10

**OBJECTIVE:** To perform interpolation and decimation using MATLAB.

**BRIEF DESCRIPTION:** Decimation and interpolation are the two basic building blocks in the multirate digital signal processing systems. The decimator is utilized to decrease the sampling rate and

interpolator to increase the sampling rate. In many applications the sampling rate of a system needs to be changed for a lower or higher sampling rate for an appropriate processing of the signal.

**PRE EXPERIMENT QUESTIONS:**

Q.1 What is interpolation and decimation?

Q.2 What is the difference between interpolation and decimation?

**PROGRAM:**

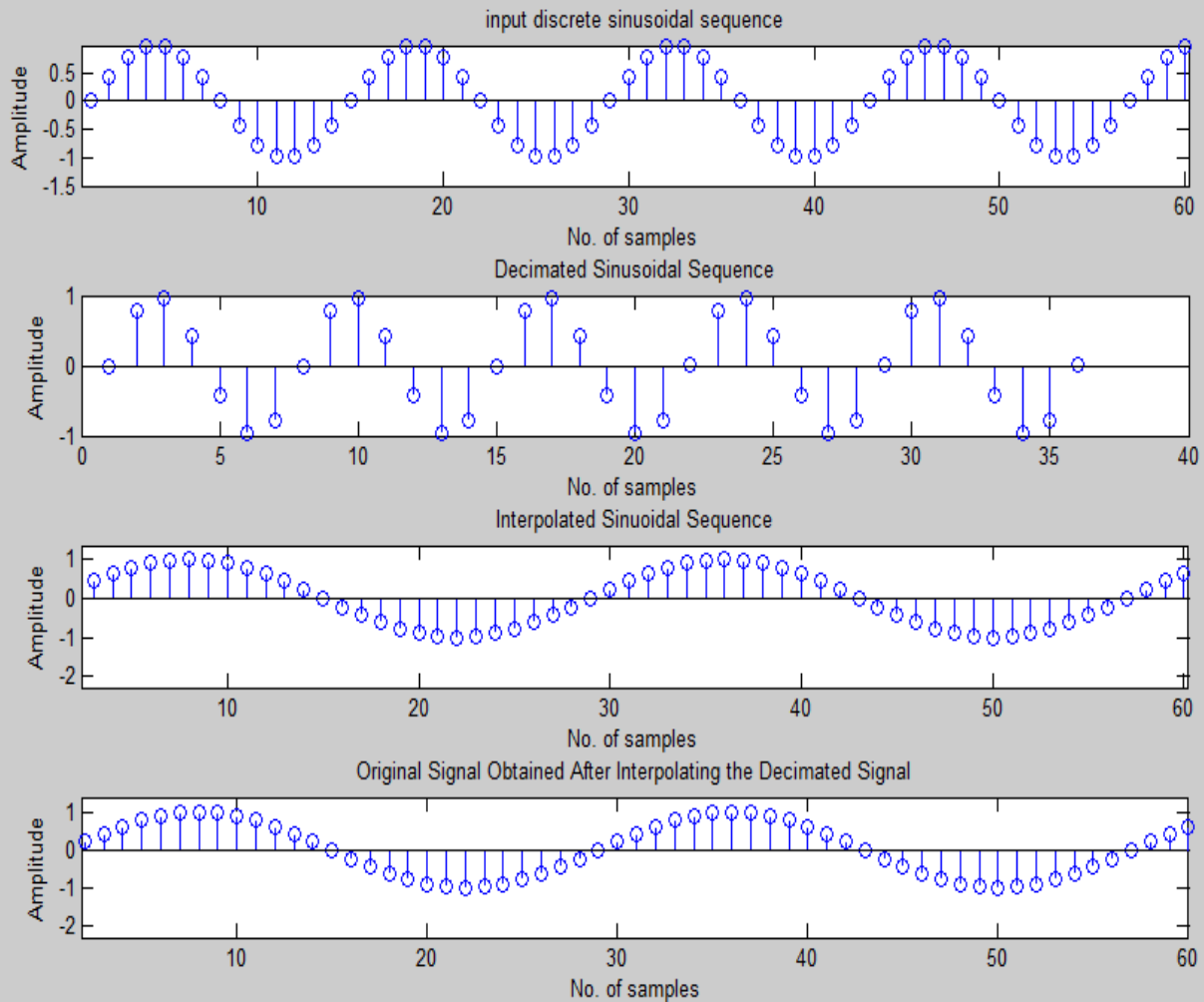
```

clc;
close all;
clear all;
% clears the command window
% closes all the previously open window
% clears previously stored values
% Generating Input Sequence
% input signal frequency
% sampling frequency
% time range for the input sequence
% input sinusoidal signal
fm = 10;
Fs = 140;
t = 0: 1/Fs:0.5;
x = sin(2*pi*fm*t);
subplot(4, 1, 1)
stem(x); % Discrete plot of the input sequence, ...
... where x-axis corresponds to the number of samples
xlabel('No. of samples'); % labelling x-axis
ylabel('Amplitude'); % labelling y-axis
title('input discrete sinusoidal sequence'); % giving title to the plot
% Decimation of the Input Sequence
% factor by which the input sequence is decimated
% resamples the sample in x with a rate (1/ M) times
M = 2;
xd = decimate(x,M);
%the original rate
subplot(4,1,2)
stem(xd) % Discrete plot of the input sequence, ...
... where x-axis corresponds to the number of samples

```

```
xlabel('No. of samples'); % labelling x-axis
ylabel('Amplitude'); % labelling y-axis
title('Decimated Sinusoidal Sequence'); % giving title to the plot
L = 2;
x1 = interp(x,L);
%rate
subplot(4,1,3);
% Interpolation of the Input Sequence
% factor by which the input sequence is interpolated
% resamples the sample in x with a rate L times the original
stem(x1); % Discrete plot of the input sequence, ...
... where x-axis corresponds to the number of samples
xlabel('No. of samples'); % labelling x-axis
ylabel('Amplitude'); % labelling y-axis
title('Interpolated Sinusoidal Sequence') % giving title to the plot
% Interpolation of the Decimated Signal
% coefficient by which the signal is interpolated
% resamples the sample in x with a rate L times the original
L = 2;
x1 = interp(xd,L);
%rate
subplot(4, 1,4)
stem(x1); % Discrete plot of the input sequence, ...
... where x-axis corresponds to the number of samples
xlabel(' No. of samples'); % labelling x-axis
ylabel('Amplitude'); % labelling y-axis
title('Original Signal Obtained After Interpolating the Decimated Signal'); %
%giving title to the graph
```





**POST EXPERIMENT QUESTIONS:**

- Q.1 What is the need for multirate signal processing?
- Q.2. Give some examples of multirate digital systems.
- Q.3 Write the input output relationship for a decimator.
- Q.4. Write the input output relationship for an interpolator.