

QMP 7.5 R/C-1 Rev.1

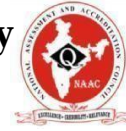


Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)

(NAAC Accredited & ISO 9001:2015 Certified Institution)

NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka.



Department of Electrical & Electronics Engineering

DIGITAL SIGNAL PROCESSING LABORATORY

18EEL67

(2022 – '23)

Version 2.0

MARCH 2023

Prepared by:

Mrs. Roopa S
Assistant Professor
Dept. of ECE

Reviewed by:

Mr. Lokanathan M S
Assistant Professor
Dept. of EEE

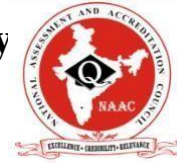
Approved by:

Mr. V C Kumar
Assoc.Professor & Head
Dept.of EEE



Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)
(NAAC Accredited & ISO 9001:2015 Certified Institution)
 NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka.



INSTITUTE VISION

To create centres of excellence in education and to serve the society by enhancing the quality of life through value based professional leadership.

INSTITUTE MISSION

- To provide high quality technical and professionally relevant education in a diverse learning environment.
- To provide the values that prepare students to lead their lives with personal integrity, professional ethics and civic responsibility in a global society.
- To prepare the next generation of skilled professionals to successfully compete in the diverse global market.
- To promote a campus environment that welcomes and honors women and men of all races, creeds and cultures, values and intellectual curiosity, pursuit of knowledge and academic integrity and freedom.
- To offer a wide variety of off-campus education and training programmes to individuals and groups.
- To stimulate collaborative efforts with industry, universities, government and professional societies.
- To facilitate public understanding of technical issues and achieve excellence in the operations of the institute.

EEE- DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Vision

Reach excellence in Electrical and Electronics Engineering education and to facilitate technically competent professionals in Electrical Sciences and allied fields with ethics to serve the society.

Mission

M1	To provide quality education to meet the modern needs in electrical & electronics engineering and allied fields.
M2	To empower each individual to apply knowledge and skills for the betterment of the society.
M3	To create centre of excellence in electrical sciences through industry-institute interactions and by adopting modern technology.
M4	To motivate research activities among students and faculty to meet the evolving needs of the society.

Program Educational Objectives (PEOs)	
PEO1	To exhibit strong knowledge in electrical sciences, mathematics and to analyze, apply, design and develop products of real time applications.
PEO2	To utilize technical knowledge, effective communication, leadership qualities and engaging with lifelong learning for the progress of Society.
PEO3	To facilitate a holistic academic environment and multidisciplinary approach for pursuing higher studies and to innovate through continuous research.
Program Specific Objectives (PSOs)	
PSO1	Analyse and apply principles of electrical science, mathematics and various techniques to evaluate different circuits and to assess the performance of machines, transmission and distribution, protection mechanisms in power system.
PSO2	Design and development of electrical and electronics circuits, measuring instruments and their Testing, control systems and strategies for power electronics, digital electronics circuits and application of microcontrollers.
PSO3	Able to effectively communicate and work in a team with ethical attitude and to apply holistic knowledge in Design, development and demonstration of project.

Engineering Graduates will be able to (Program Outcomes)	
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and engg. specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, research literature, and analyze engineering problems to arrive at substantiated conclusions using first principles of mathematics, natural, and engineering sciences
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components, processes to meet the specifications with consideration for the public health and safety, and the cultural, societal, and environmental considerations
PO4	Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid Conclusions
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable Development
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings
PO10	Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions
PO11	Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments
PO12	PO 12. Life-long learning Program: 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



Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)

(NAAC Accredited & ISO 9001:2015 Certified Institution)

NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka.

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING



COURSE: DIGITAL SIGNAL PROCESSING LABORATORY: 18EEL67

Course objectives:

1. To explain the use of MATLAB/ software in evaluating the DFT and IDFT of given Sequence.
2. To verify the convolution property of the DFT.
3. To design and implementation of IIR and FIR filters for given frequency specifications.
4. To realize IIR and FIR filters.
5. To help the students in developing software skills.

COURSE OUTCOMES:

At the end of the course the student will be able to:

CO1 : Apply fast and efficient algorithms for computing DFT and inverse DFT.

CO2 : Design different types of filters using various transformation techniques.

CO3: Analyze FIR filters by using different window function and frequency sampling method

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

INDEX PAGE

18EEL67: DIGITAL SIGNAL PROCESSING LAB

Sl. No.	Name of the Experiment	Date			Observation Marks (Max. 20)	Record Marks (Max. 10)	Signature (Student)	Signature (Faculty)
		Conduction	Repetition	Submission of Record				
01								
02								
03								
04								
05								
06								
07								
08								
09								
10								
11								
12								
Average								

Note: If the student fails to attend the regular lab, the experiment has to be completed in the same week. Otherwise manual/observation and record will be evaluated for 50% of maximum marks.

Syllabus

DIGITAL SIGNAL PROCESSING LABORATORY

Course Code	18EEL67	CIE Marks	40
Number of practical Hours/week	03(1 Hour Instruction + 2 Hours Laboratory)	SEE Marks	60
RBT levels	L1,L2,L3	Exam Hours	03
Credits-02			

1. Verification of Sampling Theorem both in time and frequency domains
 2. Evaluation of impulse response of a system
 3. To perform linear convolution of given sequences
 4. To perform circular convolution of given sequences using (a) the convolution summation formula (b) the matrix method and (c) Linear convolution from circular convolution with zero padding.
 5. Computation of N – point DFT and to plot the magnitude and phase spectrum.
 6. Linear and circular convolution by DFT and IDFT method.
 7. Solution of a given difference equation.
 8. Calculation of DFT and IDFT by FFT
 9. Design and implementation of IIR filters to meet given specification (Low pass, high pass, band pass and band reject filters)
 10. Design and implementation of FIR filters to meet given specification (Low pass, high pass, band pass and band reject filters) using different window functions
 11. Design and implementation of FIR filters to meet given specification (Low pass, high pass, band pass and band reject filters) using frequency sampling technique
 12. Realization of IIR and FIR filters
-


DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG

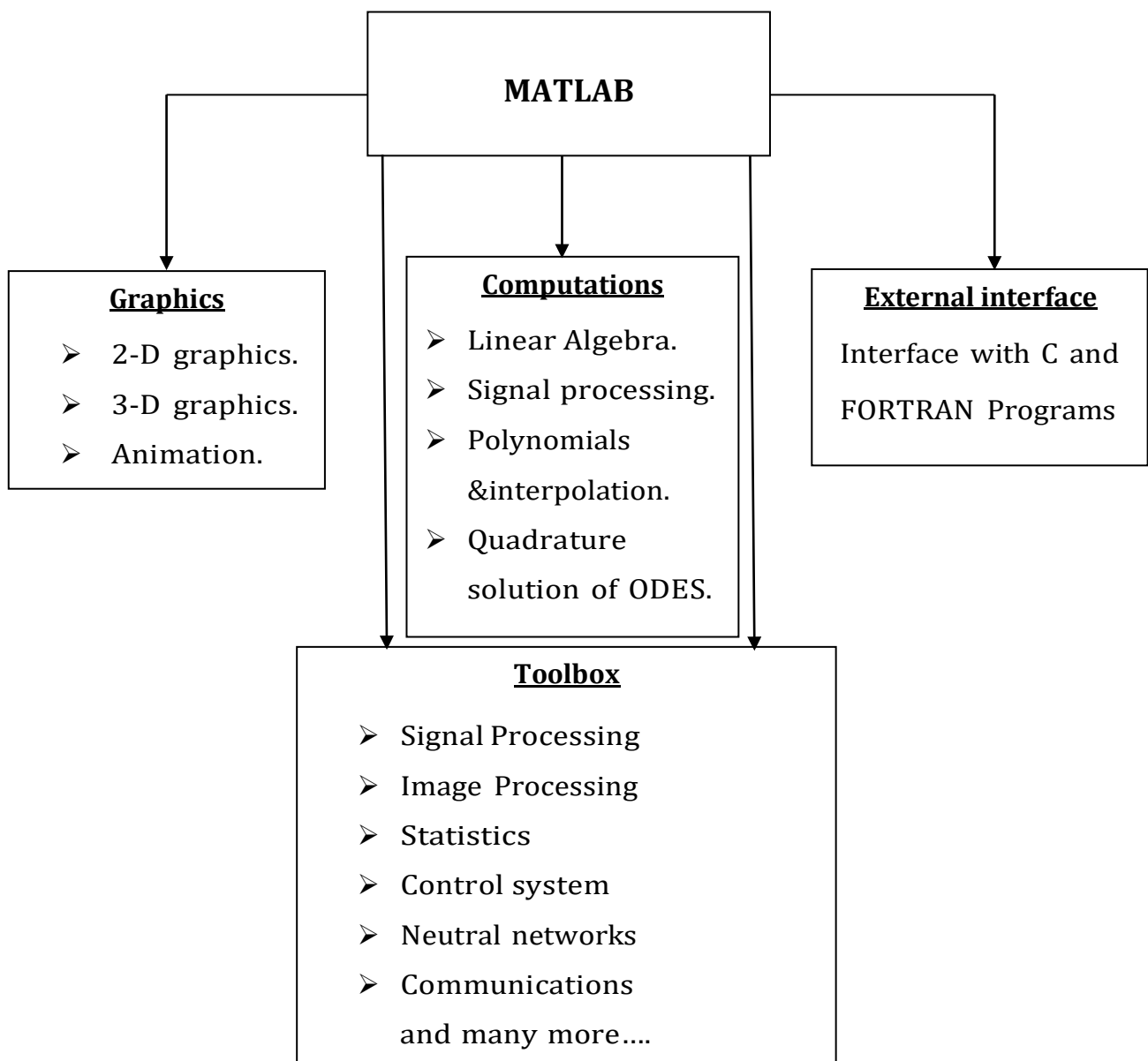
Contents

Exp. No.	Title of the Experiments	Page No.
First cycle Experiments		
	Introduction to MATLAB	1
	Warm up cycle: Generation of signals	10
1	Verification of Sampling Theorem both in time and frequency domains.	12
2	Evaluation of impulse response of a system	17
3	To perform linear convolution of given sequences	19
4	To perform circular convolution of given sequences	21
5	Computation of N – point DFT and to plot the magnitude and phase spectrum.	24
6	Linear and circular convolution by DFT and IDFT	26
Second cycle Experiments		
7	Solution of a given difference equation.	32
8	Calculation of DFT and IDFT by FFT	33
9	Design and implementation of IIR filters to meet given specification	34
10	Design and implementation of FIR filters to meet given specification using different window functions	44
11	Design and implementation of FIR filters to meet given specification using frequency sampling technique	59
12	Realization of IIR and FIR filters	61

INTRODUCTION TO MATLAB

MATLAB is a software package for high performance numerical computation and visualization provides an interactive environment with hundreds of built in functions for technical computation, graphics and animation. The MATLAB name stands for MATrix LABoratory.

The diagram shows the main features and capabilities of MATLAB.



At its core, MATLAB is essentially a set (a “toolbox”) of routines (called “m files’ or “mex files”) that sit on your computer and a window that allows you to create new variables with names (e.g. voltage and time) and process those variables with any of those routines (e.g. plot voltage against time, find the largest voltage, etc.)

MATLAB® is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran.

Key Features:

- High-level language for technical computing.
- Development environment for managing code, files, and data.
- Interactive tools for iterative exploration, design, and problem solving.
- Mathematical functions for linear algebra, statistics, Fourier analysis,
- Filtering, optimization, and numerical integration.
- 2-D and 3-D graphics functions for visualizing data.
- Tools for building custom graphical user interfaces.
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft® Excel®.

MATLAB Windows:

MATLAB works with through three basic windows

Command Window: This is the main window .It is characterized by MATLAB command prompt(>>) when you launch the application program MATLAB puts you in this window all commands including those for user-written programs ,are typed in this window at the MATLAB prompt.

Graphics window: The output of all graphics commands typed in the command window are flushed to the graphics or figure window, a separate gray window with white background color the user can create as many windows as the system memory will allow.

Edit window: This is where you write edit, create and save your own programs in files called M files.

Input-output:

MATLAB supports interactive computation taking the input from the screen and flushing the output to the screen. In addition it can read input files and write output files.

Data Type: The fundamental data-type in MATLAB is the array. It encompasses several distinct data objects- integers, real numbers, matrices. Character strings, structures and cells. There is no need to declare variables as real or complex, MATLAB automatically sets the variable to be real.

Dimensioning: Dimensioning is automatic in MATLAB. No dimension statements are required for vectors or arrays. We can find the dimensions of an existing matrix or a vector with the size and length commands.

Basic Instructions in Matlab**1) T = 0:1:10**

This instruction indicates a vector T which as initial value 0 and final value 10 with an increment of 1.

Therefore,

T =

0 1 2 3 4 5 6 7 8 9 10

2) T= Zeros (2, 3)

The above instruction creates a vector of two rows and three columns whose values are zero.

T =

0 0 0
0 0 0

3) T=Ones (3,2)

The above instruction creates a vector of three rows and two columns whose values are one

T =

1 1
1 1
1 1

$$4) \mathbf{a} = [1 \ 2 \ 3] \quad \mathbf{b} = [4 \ 5 \ 6]$$

$\mathbf{a}.*\mathbf{b} = [4 \ 10 \ 18]$, which is the result of individual elements.

5) Matrix and Array Operations

MATLAB allows you to process all of the values in a matrix using a single arithmetic operator or function.

To create a matrix that has multiple rows, separate the rows with semicolons.

- $\mathbf{a} = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 10]$

$\mathbf{a} =$

```
1  2  3
4  5  6
7  8 10
```

- $\mathbf{a} + 10$

$\mathbf{ans} =$

```
11 12 13
14 15 16
17 18 20
```

- $\sin(\mathbf{a})$

$\mathbf{ans} =$

```
0.8415    0.9093    0.1411
-0.7568   -0.9589   -0.2794
0.6570    0.9894   -0.5440
```

- To transpose a matrix, use a single quote ('):

\mathbf{a}'

$\mathbf{ans} =$

```
1  4  7
2  5  8
3  6 10
```

You can perform standard matrix multiplication, which computes the inner products between rows and columns, using the `*` operator. For example, confirm that a matrix times its inverse returns the identity matrix:

- **`p = a*inv(a)`**

`p =`

```
1.0000      0      -0.0000
0           1.0000     0
0           0           1.0000
```

Notice that `p` is not a matrix of integer values. MATLAB stores numbers as floating-point values, and arithmetic operations are sensitive to small differences between the actual value and its floating-point representation. You can display more decimal digits using the format command:

- **Format long:**

`p = a*inv(a)`

`p =`

```
1.0000000000000000  0  -0.0000000000000000
0                   1.0000000000000000  0
0                   0                   0.9999999999999998
```

Reset the display to the shorter format using `format short`. `format short` affects only the display of numbers, not the way MATLAB computes or saves them. To perform element-wise multiplication rather than matrix multiplication, use the `.*` operator:

- **`p = a.*a`**

`p =`

```
1     4     9
16    25    36
49    64    100
```

The matrix operators for multiplication, division, and power each have a corresponding array operator that operates element-wise. For example, raise each element of `a` to the third power:

- $a.^3$

ans =

1	8	27
64	125	216
343	512	1000

Concatenation:

Concatenation is the process of joining arrays to make larger ones. In fact, you made your first array by concatenating its individual elements. The pair of square brackets `[]` is the concatenation operator.

- $A = [a,a]$

A =

1	2	3	1	2	3
4	5	6	4	5	6
7	8	10	7	8	10

Concatenating arrays next to one another using commas is called *horizontal* concatenation. Each array must have the same number of rows. Similarly, when the arrays have the same number of columns, you can concatenate *vertically* using semicolons.

- $A = [a; a]$

A =

1	2	3
4	5	6
7	8	10
1	2	3
4	5	6
7	8	10

Complex Numbers:

Complex numbers have both real and imaginary parts, where the imaginary unit is the square root of -1 .

- **sqrt(-1)**

ans = 0 + 1.0000i

To represent the imaginary part of complex numbers, use either i or j .

c = [3+4i, 4+3j, -i, 10j]

c = 3.0000 + 4.0000i 4.0000 + 3.0000i 0 - 1.0000i 0
+10.0000i

Array Indexing:

Every variable in MATLAB is an array that can hold many numbers. When you want to access selected elements of an array, use indexing.

For example, consider the 4-by-4 magic square A:

- **A = magic(4)**

A =

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

There are two ways to refer to a particular element in an array. The most common way is to specify row and column subscripts, such as

- **A(4,2)**

ans = 14

- Less common, but sometimes useful, is to use a single subscript that traverses down each column in order:

- **A(8)**

ans = 14

- Using a single subscript to refer to a particular element in an array is called *linear indexing*.
- If you try to refer to elements outside an array on the right side of an assignment statement, MATLAB throws an error.
- **test = A(4,5)**
- Attempted to access A(4,5); index out of bounds because size(A)=[4,4]. However, on the left side of an assignment statement, you can specify elements outside the current dimensions. The size of the array increases to accommodate the newcomers.
- **A(4,5) = 17**

Example:

A =

16	2	3	13	0
5	11	10	8	0
9	7	6	12	0
4	14	15	1	17

To refer to multiple elements of an array, use the colon operator, which allows you to specify a range of the form start:end. For example, list the elements in the first three rows and the second column of A:

- **A(1:3,2)**

ans = 2

11

7

The colon alone, without start or end values, specifies all of the elements in that dimension. For example, select all the columns in the third row of A:

- **A(3,:)**

ans = 9 7 6 12 0

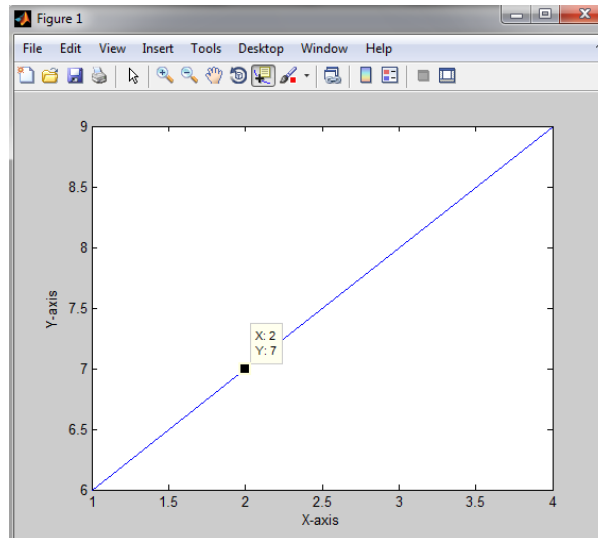
The colon operator also allows you to create an equally spaced vector of values using the more general form start:step:end.

6) Plot (t, m)

If $m = [6 \ 7 \ 8 \ 9]$

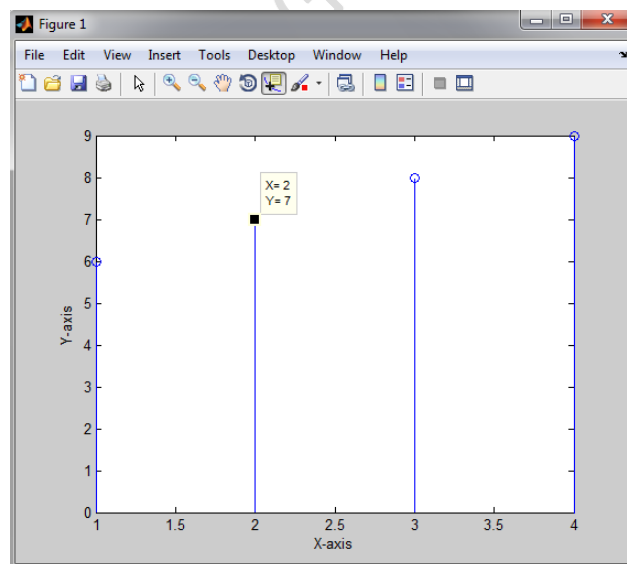
$t = [1 \ 2 \ 3 \ 4]$

This instruction will display a figure window which indicates the plot of m versus t .



7) Stem (t,m)

This Instruction will display a figure window as shown below,



8) Subplot: This function divides the figure window into rows and Columns.

Example: `subplot(3,1,2)` ---- \rightarrow 3-rows, 1-column, 2-location.

WARM UP CYCLE: GENERATION OF SIGNALS

Aim: Generation of impulse signal.

```
clc;
clear all;
close all;
t=-5:1:5;
y=[zeros(1,5),ones(1,1),zeros(1,5)];
stem(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Discrete time impulse function');
```

Aim: Generation of unit step signal.

```
clc;
clear all;
close all;
n=input('Enter the n value:');
t=0:1:n-1;
y=zeros(1,n);
stem(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Discrete time unit step function');
```

Aim: Generation of ramp signal.

```
clc;
clear all;
close all;
n=input('Enter the n value: ');
t=0:n;
stem(t,t,'r');
hold on;
plot(t,t,'k');
xlabel('Time');
ylabel('Amplitude');
title('Ramp function');
legend('Discrete','continuous');
```

Aim: Generation of exponential signal.

```
clc;
clear all;
close all;
n=input('enter the N value:');
t=0:0.1:n;
a=input('enter the A value:');
y=exp(a*t);
stem(t,y,'y');
hold on;
plot(t,y,'r');
xlabel('Time');
ylabel('Amplitude');
title('Exponential function');
legend('Discrete','continuous');
```

Aim: Generation of sine wave $y(n)=\sin(2\pi fn)$.

```
clc;
clear all;
close all;
t=0:0.001:5;
f=1;
y=sin(2*pi*f*t);
plot(t,y,'g');
xlabel('Frequency');
ylabel('amplitude');
title('sine wave');
```

Aim: Generation of cosine wave $y(n)=\cos(2\pi fn)$.

```
clc;
clear all;
close all;
t=0:0.001:5;
f=1;
y=cos(2*pi*f*t);
plot(t,y,'g');
xlabel('Frequency');
ylabel('amplitude');
title('cosine wave');
```

1. SAMPLING THEOREM

Aim: To verify sampling theorem for the following cases in both time and frequency domains:

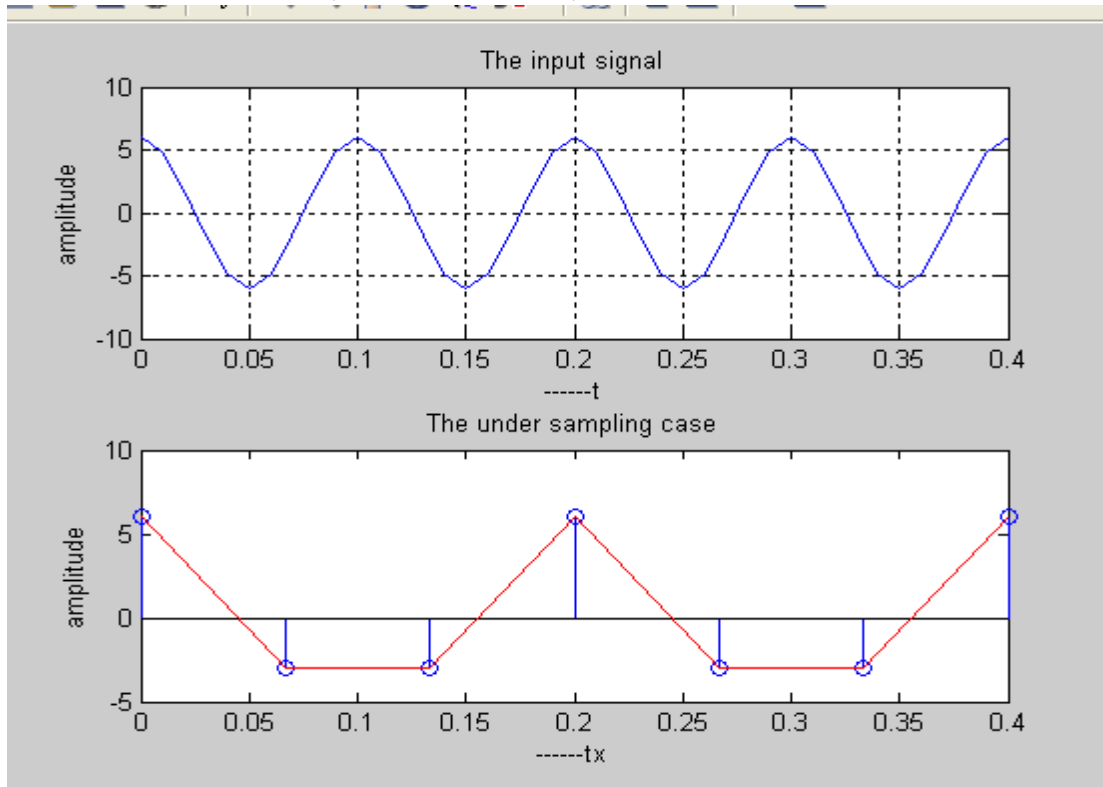
i) Under sampling ii) Nyquist sampling iii) Over sampling

Program:

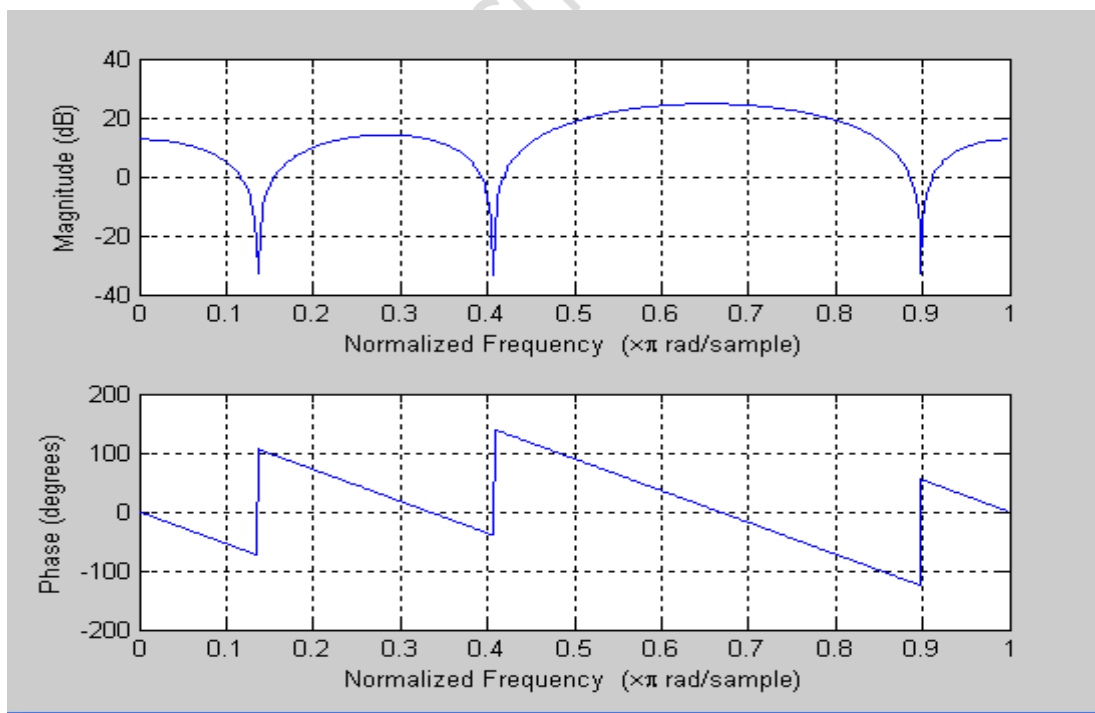
```
(a) UNDER SAMPLING
clear all;
close all;% clears the variables declared
k=input ('Enter the no of cycles = ');
a=input ('Enter the input signal amplitude= ');
fm=input ('Enter the input frequency = ');
t=0:1/(fm*fm):k/fm;
y=a*cos(2*pi*fm*t)
figure;
subplot(2,1,1)
plot(t,y)
grid on;
xlabel('----- t');ylabel('amplitude');
title('The input signal');
fnq=2*fm
fs=3/4*fnq;
tx=0:1/fs:k/fm;
ys=a*cos(2*fm*pi*tx)
subplot(2,1,2);
stem(tx,ys);
hold on
plot(tx,ys,'r');xlabel('----- tx');ylabel('amplitude');
title('The under sampling case');
t1=(k/fm)+1
figure;
freqz(ys,t1)
```

Result:

UNDER SAMPLING (Time domain)



UNDER SAMPLING (Frequency domain)



(b) NYQUIST SAMPLING

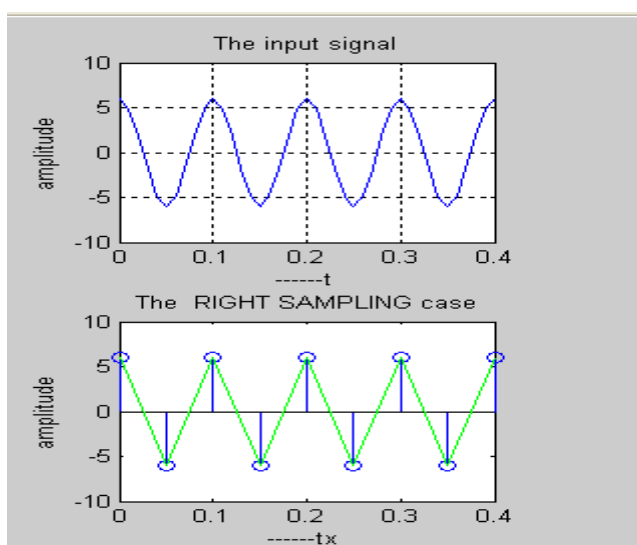
```
clear all;
close all;% clears the variables declared
k=input ('Enter the no of cycles = ');
a=input ('Enter the input signal amplitude= ');
fm=input ('Enter the input frequency = ');
t=0:1/(fm*fm):k/fm;
y=a*cos(2*pi*fm*t)
figure;
subplot(2,2,1)
plot(t,y)
grid on;
xlabel(' ----- t');ylabel('amplitude');
title('The input signal');

fnq=2*fm

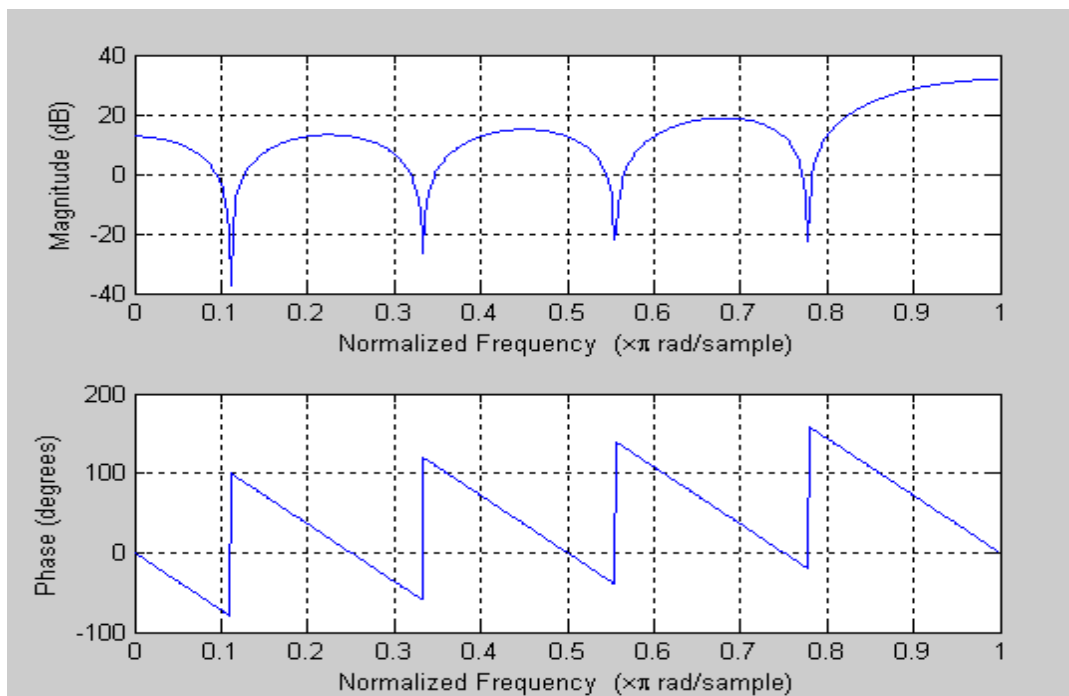
% Nyquist sampling
fs=fnq;
tx=0:1/fs:k/fm;
ys=a*cos(2*pi*fm*tx)
subplot(2,2,3);
stem(tx,ys);
hold on;
plot(tx,ys,'g');xlabel(' ----- tx');ylabel('amplitude');
title('The RIGHT SAMPLING case');
p1=(k/fm)+1
figure
freqz(ys,p1)
```

Result:

NYQUIST SAMPLING(Time domain)



NYQUIST SAMPLING (Frequency domain)



(c) OVERSAMPLING

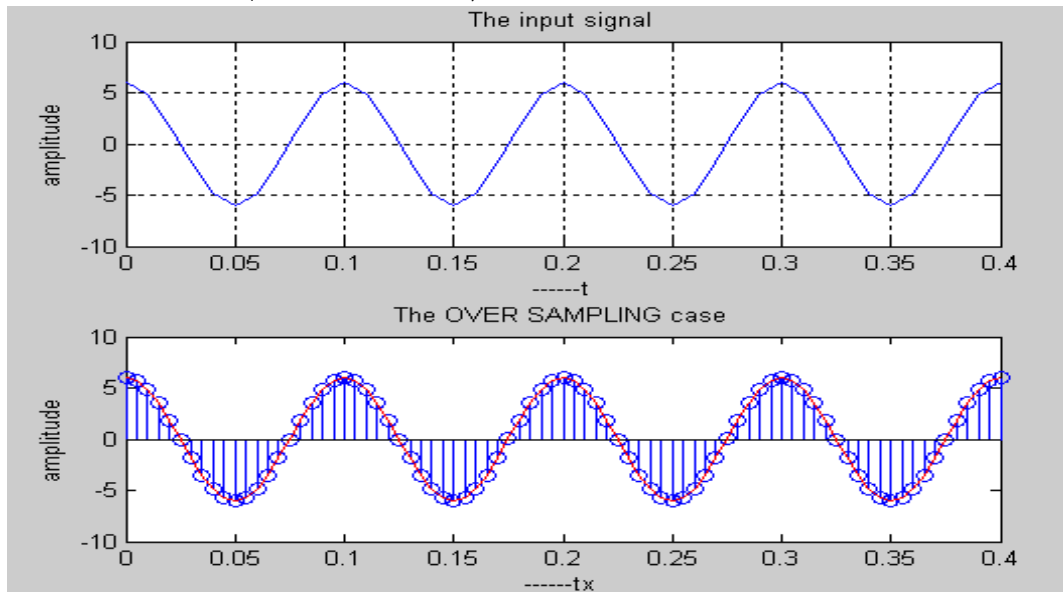
```

clear all;
close all;% clears the variables declared
k=input ('Enter the no of cycles = ');
a=input ('Enter the input signal amplitude= ');
fm=input ('Enter the input frequency = ');
t=0:1/(fm*fm):k/fm;
y=a*cos(2*pi*fm*t)
figure;
subplot(2,1,1)
plot(t,y)
grid on;
xlabel(' ----t');ylabel('amplitude');
title('The input signal');
fnq=2*fm
fs=10*fnq;
tx=0:1/fs:k/fm;
ys=a*cos(2*pi*fm*tx)
subplot(2,1,2);
stem(tx,ys);
hold on;
plot(tx,ys,'r');xlabel(' ----tx');ylabel('amplitude');
title('The OVER SAMPLING case');
p=(k/fm)+1
figure;
freqz(ys,p)

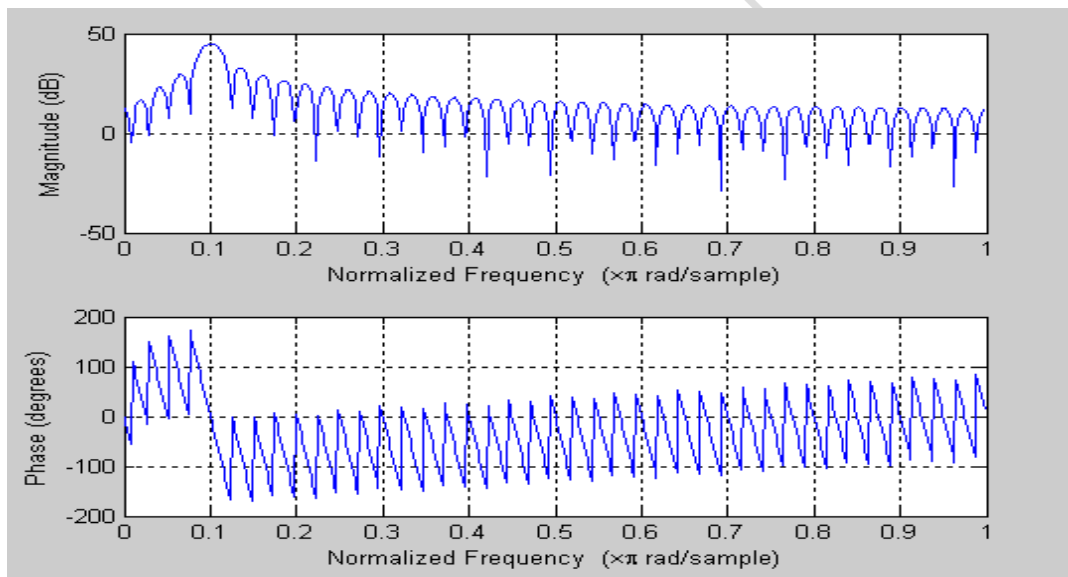
```

Result

OVERSAMPLING (Time domain)



OVERSAMPLING (Frequency domain)



2. IMPULSE RESPONSE OF THE SYSTEM

(a) Find the impulse response of a system given by the following transfer function.

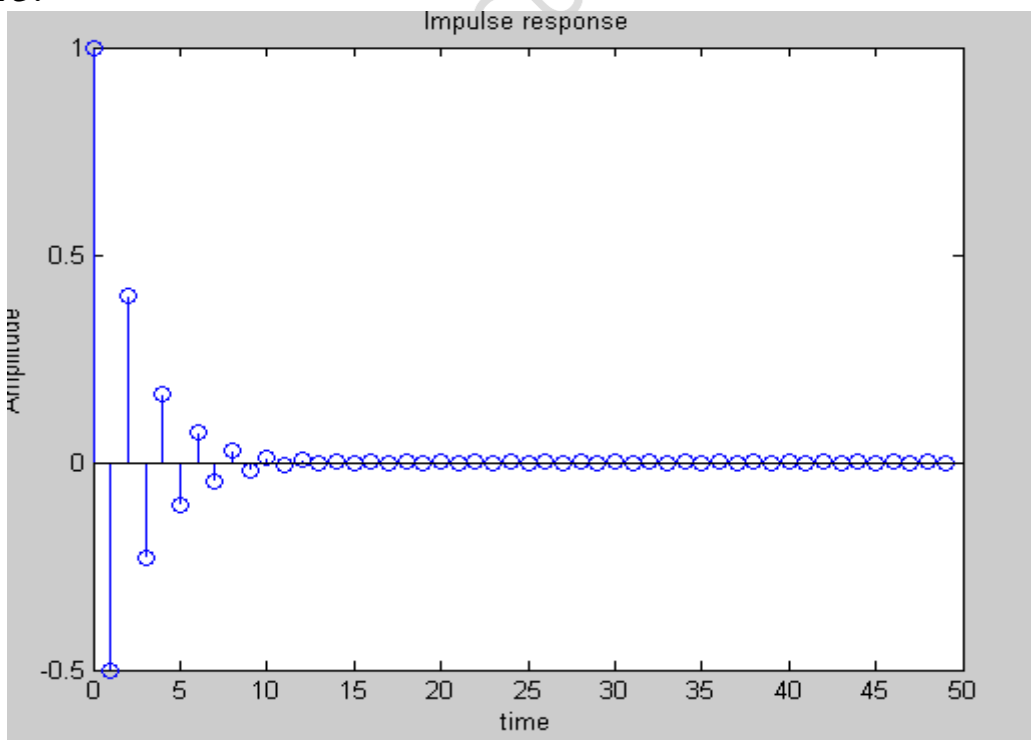
$$H(z) = \frac{1 - 0.3z^{-1}}{1 + 0.2z^{-1} - 0.3z^{-2}}$$

Plot first 50 samples. Find out if the system is IIR or FIR. Illustrate from the transfer function.

Program:

```
clear all;
close all;
clc;
num=[1 -0.3];
den=[1 0.2 -0.3];
[h t]=impz(num,den,50);
stem(t,h);
title('Impulse response');
xlabel('time');
ylabel('Amplitude');
```

Result:



b) Solve $H(z) = \frac{1 - 0.6z^{-1}}{1 + 0.4z^{-1} - 0.6z^{-2}}$

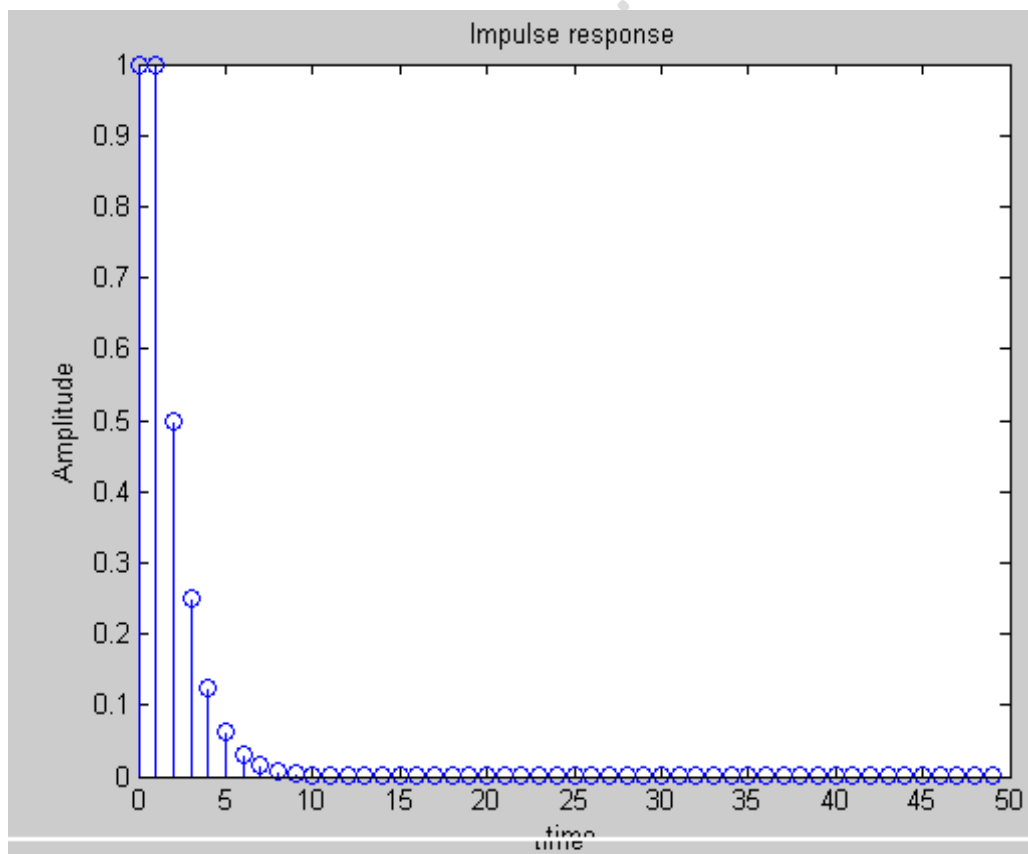
(b) From the following difference equation representation find the impulse response.

$$a) \quad y(n] - \frac{1}{2}y[n-1] = x[n] + \frac{1}{2}x[n-1]$$

Program:

```
clear all;
close all;
clc;
b=[1 0.5];
a=[1 -0.5];
[h t]=impz(b,a,50);
stem(t,h);
title('Impulse response');
xlabel('time');
ylabel('Amplitude');
```

Result:



b) Solve

$$y[n] - \frac{3}{4}y[n-1] = x[n] + \frac{3}{2}x[n-1]$$

3. CONVOLUTION

3 Aim: To find the Linear Convolution of given two sequences

Program:

```

clc;
clear all;
close all;

x=input('Enter the first input sequence x[n]= ');
xsi=input('Enter the starting index of x[n]= ');

h=input('Enter the first input sequence h[n]= ');
hsi=input('Enter the starting index of h[n]= ');

%To calculate the convolution of x[n] and h[n]%
y = conv(x,h)

%To generate the index arrays%
n1 = xsi:1:length(x)+xsi-1
n2 = hsi:1:length(h)+hsi-1
n3 = xsi+hsi:1:length(x)+length(h)+xsi+hsi-2

%To sketch the result%
subplot(311),stem(n1,x);
xlabel('n1----->'),ylabel('Input Sequence----->')
title ('Input Sequence x(n)')
subplot(312),stem(n2,h);
xlabel('n2----->'),ylabel('Impulse Response----->')
title ('Input Sequence h(n)')
subplot(313),stem(n3,y);
xlabel('n3----->'),ylabel('Output Sequence----->')
title ('Convolution of two sequences')

```

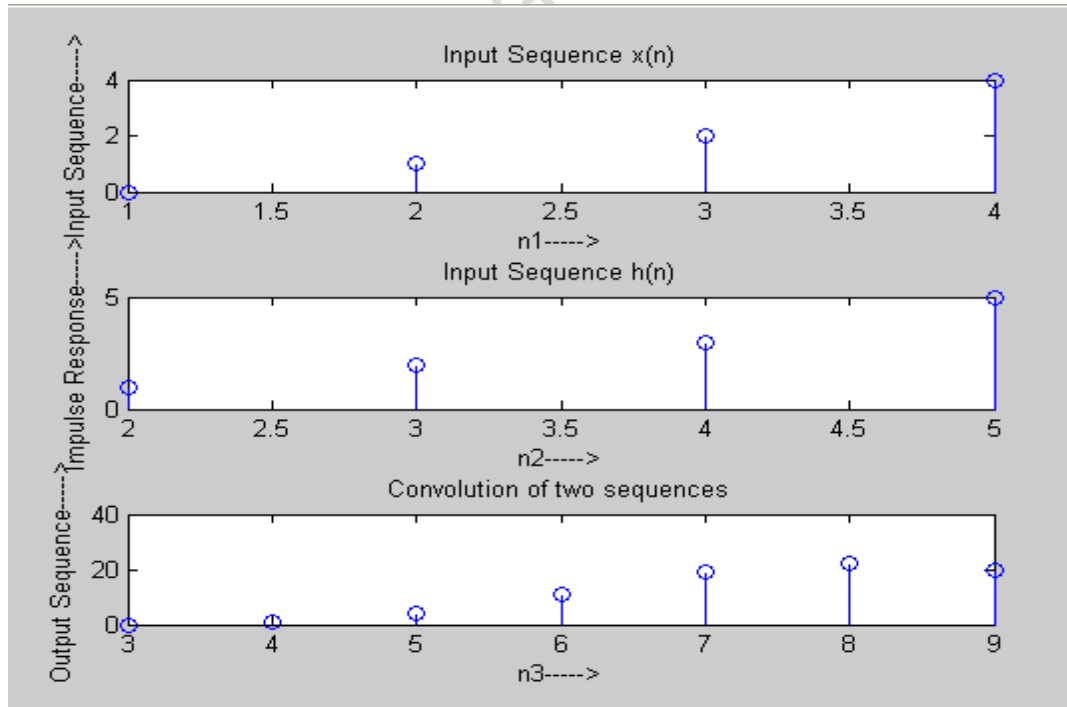
Result:

Enter the first input sequence x[n]= [0 1 2 4]

Enter the starting index of x[n]= 1

```

xsi = 1
Enter the first input sequence h[n]= [1 2 3 5]
Enter the starting index of h[n]= 2
hsi = 2
y =
    0    1    4   11   19   22   20
n1 =
    1    2    3    4
n2 =
    2    3    4    5
n3 =
    3    4    5    6    7    8    9
    
```



b) Find the Linear Convolution of given two sequences

$x[n] = [2 \ 3 \ 4 \ 5]$ $h[n] = [1 \ 2 \ 3 \ 0]$

4. CIRCULAR CONVOLUTION

4a. To find the Circular Convolution of given two sequences

Program:

```

clc;
clear all;
close all;

x=input('Enter the first sequence x[n]= ');
h=input('Enter the second sequence h[n]=');

N = max(length(x),length(h));
Y=cconv(x,h,N)      %y=x⊗Nh%

stem(Y);
xlabel('N----->'),ylabel('Amplitude ----->')
title('Plot of circularly convoluted sequence');

```

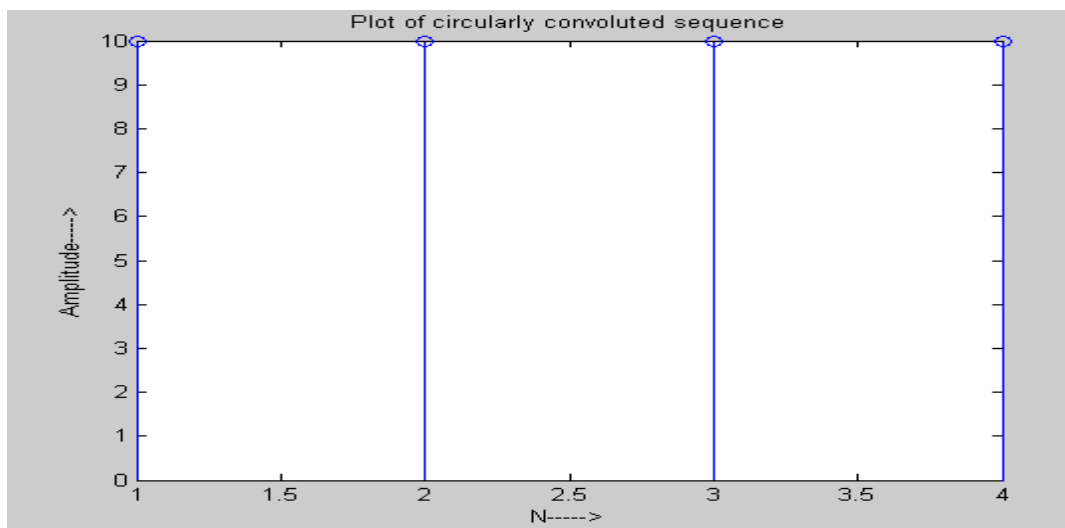
Result:

Enter the first sequence x[n]= [1 2 3 4]

Enter the second sequence h[n]=[1 1 1 1]

Y =

10 10 10 10



b) Find the Circular Convolution of $x[n]=[2\ 3\ 4\ 5]$ and $h[n]=[1\ 1\ 1\ 1]$

4b. To find the Circular Convolution of given two sequences using

matrix method.**Program:**

```
clc;
clear all;
close all;
x=input('Enter the first sequence x[n]= ');
h=input('Enter the second sequence h[n]=');
N = length(x);
for i=1:N
y(i)=0;
for j=1:N
    k=i-j+1;
if(k<=0)
        k=N+k;
end;
y(i)=y(i)+x(j)*h(k)
end;
end;
stem(y);
title('Plot of circularly convoluted sequence');
```

4c.To find the Circular Convolution of given two sequences by using zero padding method

```
clc;
clear all;
close all;
x=[2 1 2 1];
Y=[1 2 3];
Clin=conv(x,y);
xpad=[x zeros(1,6-length(x))];
hpad= [y zeros(1,6-length(y))];
```

```
ccirc = ifft(fft(xpad).*fft(ypad));
    subplot(2,1,1)
    stem(cclin,'filled')
    ylin([0 1 1])
    title('Linear convolution of x and y')
    subplot(2,1,2)
    stem(ccirc, 'filled')
    ylin([0 1 1])
    title('circular convolution of xpad and ypad')
```

5. DISCRETE FOURIER TRANSFORMATION

Aim: To compute N point DFT of a given sequence x[n] and to plot magnitude and phase spectrum.

Program:

```
clc;
clear all;
close all;
x = input('enter the input seq=')
N = length(x);
X = zeros(N,1)
for k = 0:N-1
    for n = 0:N-1
        X(k+1) = X(k+1) + x(n+1)*exp(-j*pi*2*n*k/N)
    end
end
mag=abs(X)
phase=angle(X)*180/pi
t = 0:N-1
subplot(311)
stem(t,x);
xlabel('Time (s)');ylabel('Amplitude');
title('Time domain - Input sequence')
subplot(312)
stem(t,mag)
xlabel('Frequency');ylabel('|X(k)|');
title('Frequency domain - Magnitude response')
subplot(313)
stem(t,phase)
xlabel('Frequency');ylabel('phase(X(k))');
title('Frequency domain - phase response')
```

with inbuilt function

```
clc;
clear all;
close all;
x=input('enter the first sequence=')
N=length(x);
X=fft(x,N)
mag=abs(X)
phase=angle(X)*180/pi

t = 0:N-1
subplot(311)
stem(t,x);
xlabel('Time (s)');
ylabel('Amplitude');
title('Time domain - Input sequence');

subplot(312)
stem(t,X)
xlabel('Frequency');
ylabel('|X(k)|');
title('Frequency domain - Magnitude response');

subplot(313)
stem(t,angle(X))
ylabel('phase(X(k))');
title('Frequency domain - phase response');
```

Result:

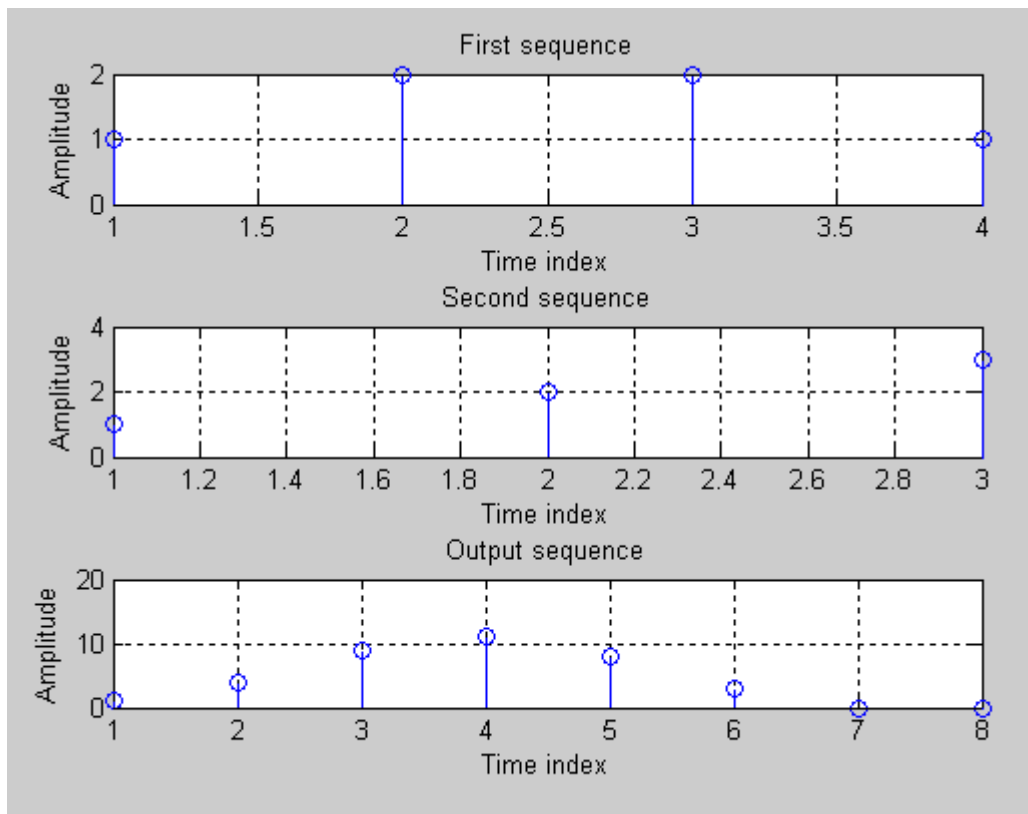
6. DISCRETE FOURIER TRANSFORMATION & INVERSE DISCRETE FOURIER TRANSFORMATION

6a. MATLAB program to verify the linear convolution property of DFT & IDFT

Program:

```
clear all;
close all;
clc;
x=[1 2 2 1];
h=[1 2 3];
%h=[h 0];
X=fft(x,8);
H=fft(h,8);
Y=X.*H;
y=ifft(Y,8);
subplot(311);
stem(1:4,x);
title('First sequence');
xlabel('Time index');
ylabel('Amplitude');
grid on;
subplot(312);
stem(1:3,h);
title('Second sequence');
xlabel('Time index');
ylabel('Amplitude');
grid on;
```

```
subplot(313);  
stem(1:8,y);  
title('Output sequence');  
xlabel('Time index');  
ylabel('Amplitude');  
grid on;
```

Result:

6b. MATLAB program to verify the circular convolution property of DFT

Circular convolution property: $x(n) \otimes y(n) = X(k)Y(k)$

% program to prove circular convolution property of DFT

```

clc;
clear all;
close all;
x1=input('enter the 1st input seq=')
x2=input('enter the 2nd input seq=')
N= input('enter the value for N=')
y1=cconv(x1,x2,N)
X1=fft(x1,N)
X2=fft(x2,N)
H=X1.*X2
y2=ifft(H,N)
    if y1 == y2
        disp('circular convolution property is proved');
    else
        disp('circular convolution property is not proved');
    end

subplot(2,1,1)
stem(y1);
xlabel('N ----->'),ylabel('amplitude')

subplot(2,1,2)
stem(y2);
xlabel('N ----->'),ylabel('amplitude')

```

Result:

enter the 1st input seq=[1 2 3 4]

x1 =

1 2 3 4

enter the 2nd input seq=[1 1 1 1]

x2 =

1 1 1 1

enter the value for N=4

N =

4

y1 =

10 10 10 10

X1 =

10.0000 -2.0000 + 2.0000i -2.0000 -

2.0000 - 2.0000i

X2 =

4 0 0 0

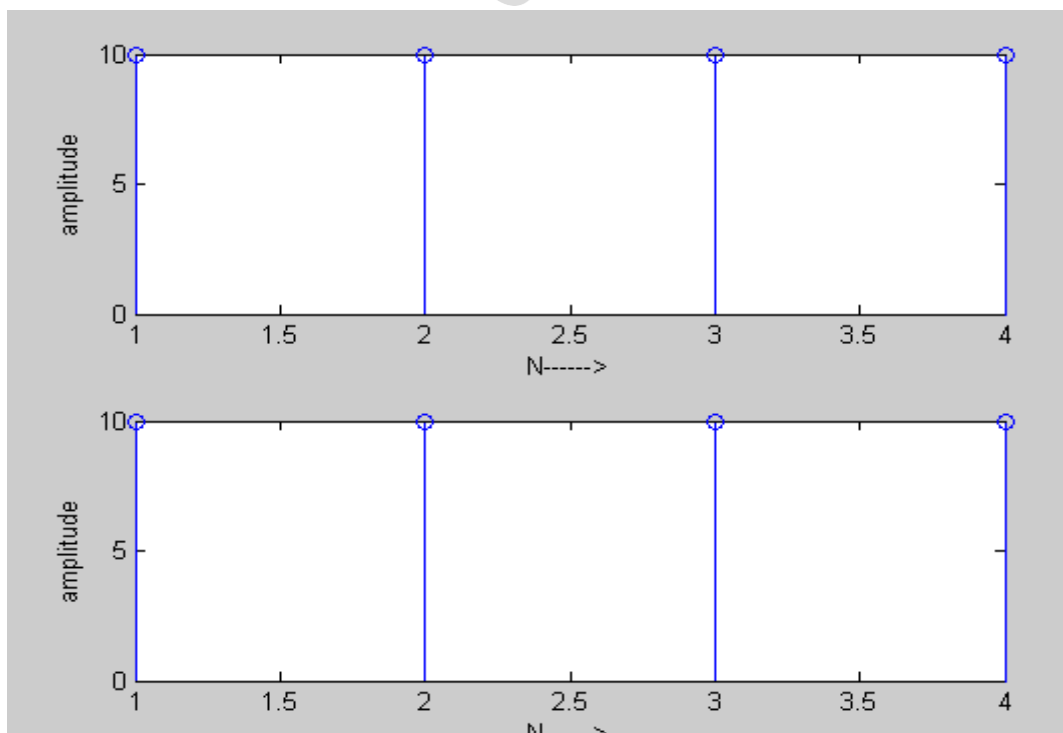
H =

40 0 0 0

y2 =

10 10 10 10

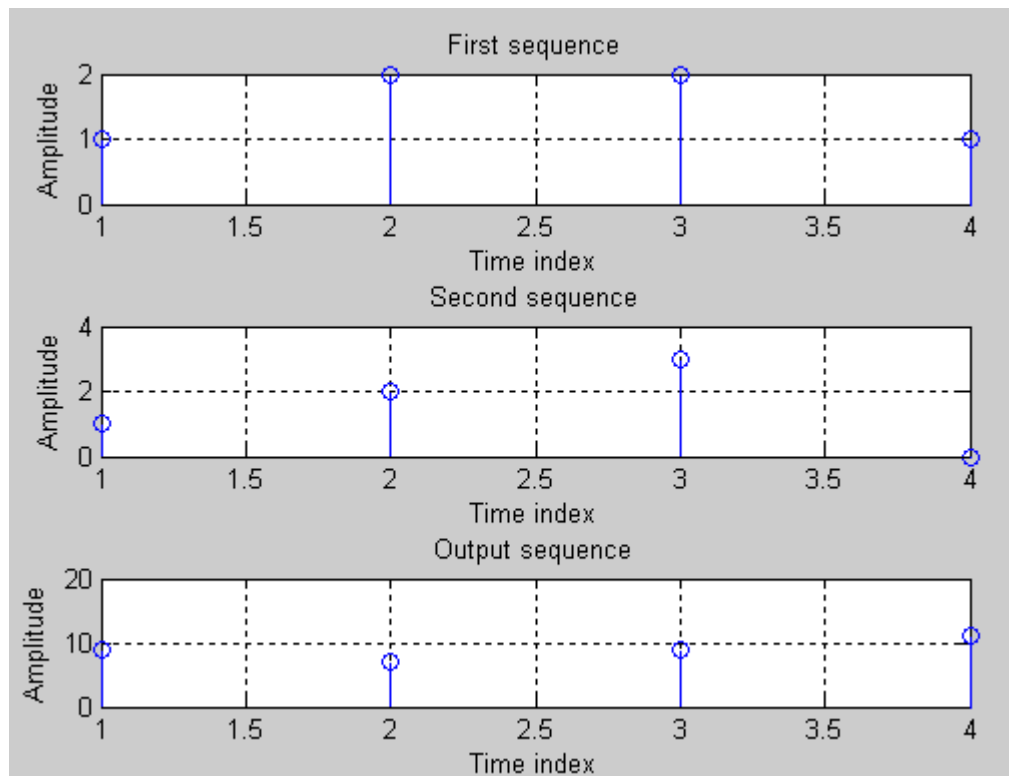
circular convolution property is proved



6b. MATLAB program to verify the circular convolution property of**IDFT**

```
Clc;
Clear all;
Close all;
x=[1 2 2 1];
h=[1 2 3];
h=[h,0];
X=FFT(x,4);
H=FFT(h,4);
Y=X.*H;
y=ifft(y,4);
subplot(311);
stem(1:4,x);
title('First sequence');
xlabel('Time index');
xlabel('Amplitude');
grid on;
subplot(312);
stem(1:4,h);
title('second sequence');
xlabel('Time index');
xlabel('Amplitude');
grid on;
subplot(313);
stem(1:4,y);
title('output sequence')
xlabel('Time index');
xlabel('Amplitude');
grid on;
xlabel('Time index');
xlabel('Amplitude');
grid on;
```

RESULT:



7. SOLUTION TO DIFFERENCE EQUATION

Aim:To solve the given difference equation (DE) with initial conditions.

Program:

```
clc;
clear all;
close all;
a=input('Enter the coefficient of y')
b=input('Enter the coefficient of x')
xi=input('Enter the initial values x[-1],x[-2],...= ')
yi=input('Enter the initial values y[-1],y[-2],....= ')
x=input ('Enter the input sequence x= ');
zi=filtic(b,a,yi,xi);
disp('The response of the given system is,');
y = filter(b,a,x,zi)
n= 0:1:length(y)-1;
stem(n,y);
xlabel('n---->'),ylabel('amplitude ---->');
title ('solution to a given differential equation with initial
conditions')
```

8. CALCULATION OF DFT AND IDFT BY FFT

```
clc;
clear all;
close all;
x=input('enter the first sequence=')
N=length(x);
X=fft(x,N)
mag=abs(X)
phase=angle(X)*180/pi

t = 0:N-1
subplot(311)
stem(t,x);
xlabel('Time (s)');
ylabel('Amplitude');
title('Time domain - Input sequence');

subplot(312)
stem(t,X)
xlabel('Frequency');
ylabel('|X(k)|');
title('Frequency domain - Magnitude response');

subplot(313)
stem(t,angle(X))
ylabel('phase(X(k))');
title('Frequency domain - phase response');
```


9. IIR LOW PASS FILTER

9 a) Aim: To Design and implement Digital IIR low pass filter

Design specifications:

- a) k_P = Pass band attenuation(dB) =
- b) k_S = Stop band attenuation(dB) =
- c) f_P = Pass band edge(Hz) =
- d) f_S = Stop band edge(Hz) =
- e) $F_S = \frac{1}{T_S}$ = Sampling rate(samples/sec) =

Design steps:

I. Obtain the equivalent digital filter specifications:

- a) $\omega_P = 2 * \pi * f_P * T_S$ rad.
- b) $\omega_S = 2 * \pi * f_S * T_S$ rad.

II. Prewarp the frequencies ω_P, ω_S :

$$\text{a) } \Omega_P = \frac{2}{T_S} * \tan\left(\frac{\omega_P}{2}\right)$$

$$\text{b) } \Omega_S = \frac{2}{T_S} * \tan\left(\frac{\omega_S}{2}\right)$$

III. Design an analog Butterworth filter with specifications $k_P, k_S, \Omega_P, \Omega_S$:

a) Order of the filter is,

$$N = \frac{1}{2} \left(\frac{\log \left[\frac{10^{0.1k_S} - 1}{10^{0.1k_P} - 1} \right]}{\log \left(\frac{\Omega_S}{\Omega_P} \right)} \right)$$

b) Cut-off frequency, $\Omega_c = \frac{1}{2} \left\{ \frac{\Omega_P}{(10^{0.1k_P} - 1)^{\frac{1}{2N}}} + \frac{\Omega_S}{(10^{0.1k_S} - 1)^{\frac{1}{2N}}} \right\}$

- c) From Butterworth polynomial table write down the transfer function of the normalized analog Butterworth filter $H_N(s)$.

Table.1 Normalized Denominator Polynomials in Factored Form.

N	Normalized Denominator Polynomials in Factored Form.
1	$(1+s)$
2	$(1+1.414s+s^2)$
3	$(1+s)(1+s+s^2)$
4	$(1+0.765s+s^2)(1+1.848s+s^2)$
5	$(1+s)(1+0.618s+s^2)(1+1.618s+s^2)$
6	$(1+0.518s+s^2)(1+1.414s+s^2)(1+1.932s+s^2)$
7	$(1+s)(1+0.445s+s^2)(1+1.247s+s^2)(1+1.802s+s^2)$
8	$(1+0.390s+s^2)(1+1.111s+s^2)(1+1.663s+s^2)(1+1.962s+s^2)$
9	$(1+s)(1+0.347s+s^2)(1+s+s^2)(1+1.532s+s^2)(1+1.879s+s^2)$
10	$(1+0.313s+s^2)(1+0.908s+s^2)(1+1.414s+s^2)(1+1.782s+s^2)$ $(1+1.975s+s^2)$

d) Un normalize the filter using substitution $H(s) = H_N(s)$ with $s = \frac{S}{\Omega_c}$

IV. Using BLT obtain equivalent digital filter system function:

$$H(z) = H(s), \text{ where } s = \frac{2}{T_s} * \left[\frac{1 - z^{-1}}{1 + z^{-1}} \right]$$

V. Verify the frequency response as to whether the filter satisfies the given specifications (ω_P , ω_S , k_P , k_S).

Note:

1) That is pass band edge after prewarping in $\text{Hz} = \frac{\Omega_P}{2\pi}$ & stop band edge after

prewarping in $\text{Hz} = \frac{\Omega_S}{2\pi}$.

2) For digital filter frequency response, x-axis is in rad & it is normalized. That

is pass band edge = $\frac{\omega_P}{\pi}$ and stop band edge = $\frac{\omega_S}{\pi}$

Design example:**Specifications:**

- a) Passband attenuation (dB) = -1dB
 b) Stop band attenuation (dB) = -3 dB
 c) Pass band edge (Hz) = 1500Hz
 d) Stop band edge (Hz) = 2000Hz
 e) $F_s = \frac{1}{T_s} = \text{Sampling rate (samples/sec)} = 8000\text{Hz}$

Solution:**I. Equivalent digital filter specifications:**

- a) $\omega_P = 1.178 \text{ rad.}$
 b) $\omega_S = 1.5708 \text{ rad.}$

II. Prewarp the frequencies ω_P , ω_S :

- a) $\Omega_P = 10689.73 \text{ rad/sec.}$
 b) $\Omega_S = 16000 \text{ rad/sec.}$

III. Design of analog filter:

- a) Order of the filter is, $N \approx 2$
 b) Cut-off frequency, $\Omega_c = 16000 \text{ rad/sec.}$

$$c) H(s) = \frac{1}{s^2 + 1.414s + 1}$$

$$d) H(s) = H_N(s) = \frac{256 * 10^6}{s^2 + 22654.29s + 256 * 10^6}$$

IV. Using BLT equivalent digital filter system function:

$$H(z) = \frac{0.2932Z^2 + 0.586Z + 0.2932}{Z^2 + 0.0014Z + 0.1716}$$

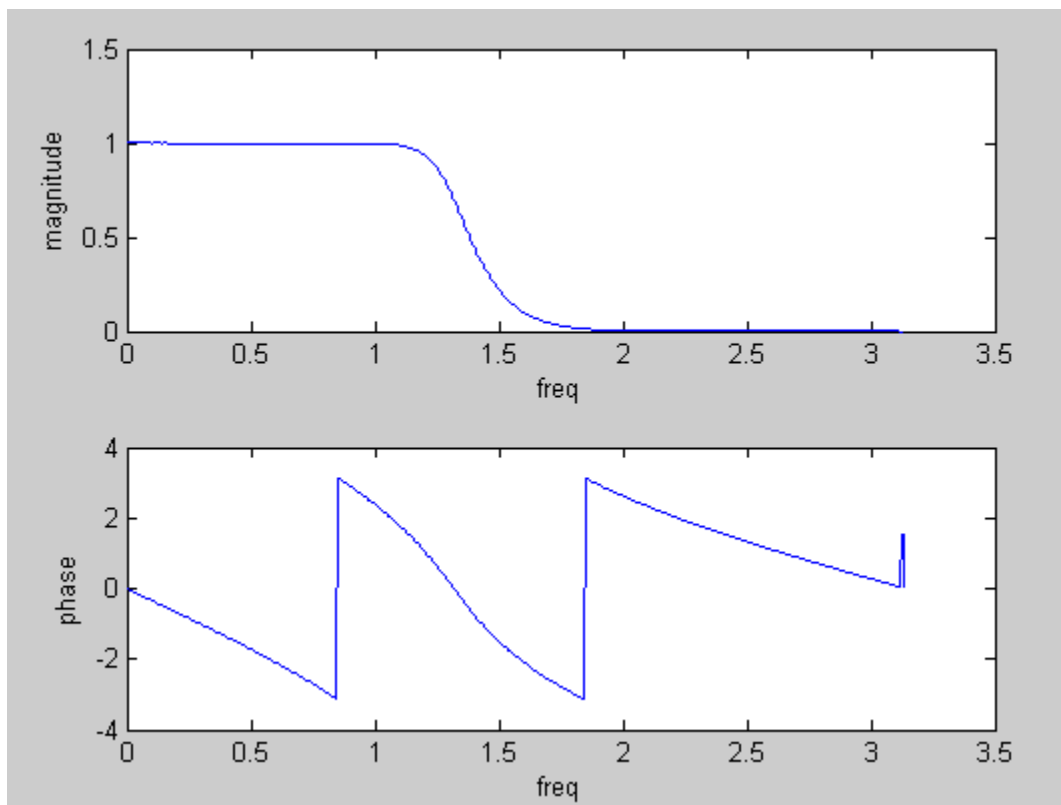
NOTE: Specifications may be given in radians or rad/sec or Hz.

9a) Aim: To Design and implement Digital IIR low pass filter

Design an IIR high pass digital filter using Butterworth approximation. The pass band ripple is less than 3db, Stop band attenuation is not more than 40db, and pass band corner frequency is 200 Hz, Stop band edge frequency is 100 Hz, Sampling frequency 1KHz. Plot Magnitude and Phase response of such a filter.

Program:

```
clear all;
close all;
clc;
Rp=3;
Rs=40
fp=200;
fs=300;
F=1000
Wp=2*fp/F;
Ws=2*fs/F;
[n,Wn]=buttord(Wp,Ws,Rp,Rs);
[b,a]=butter(n,Wn);
[h,w]=freqz(b,a,512);
subplot(211);
plot(w,abs(h));
xlabel('freq');
ylabel('magnitude');
subplot(212);
plot(w,angle(h));
xlabel('freq');
ylabel('phase');
```

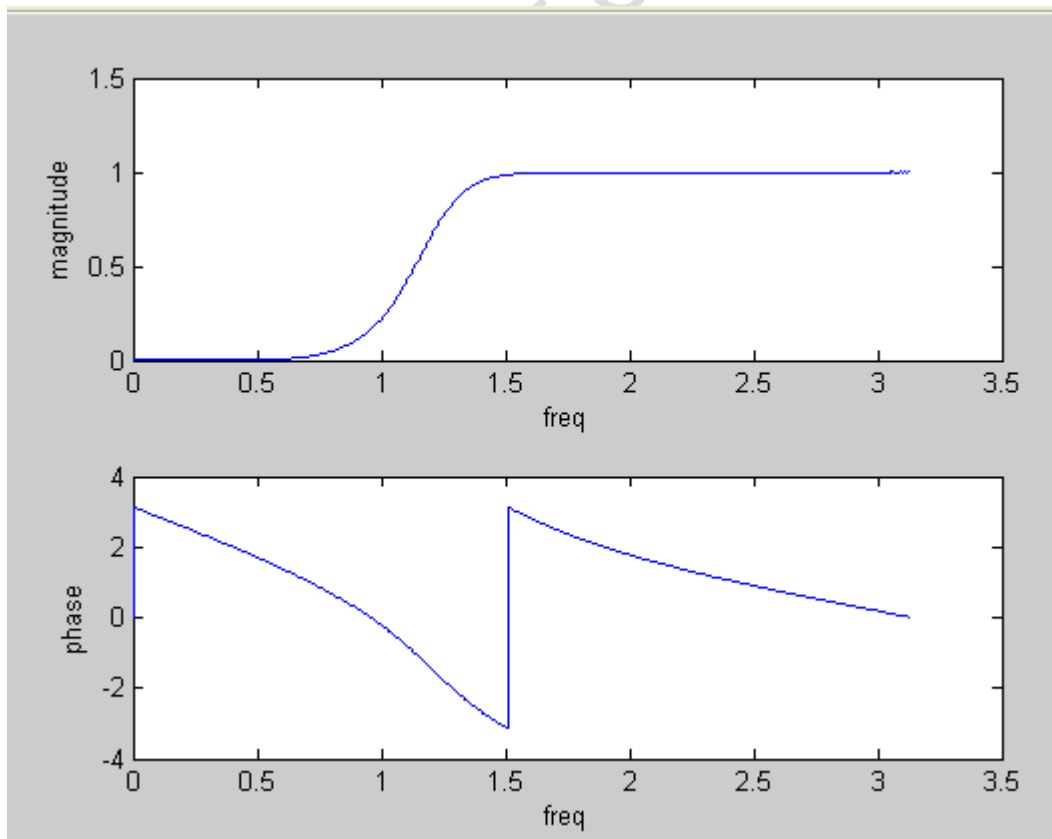
Result:**9b) Aim: To Design and implement Digital IIR high pass filter**

Design an IIR high pass digital filter using Butterworth approximation. The pass band ripple is less than 3db, Stop band attenuation is not more than 40db, and pass band corner frequency is 200 Hz, Stop band edge frequency is 100 Hz, Sampling frequency 1KHz. Plot Magnitude and Phase response of such a filter

Program:

```
clear all;
close all;
clc;
Rp=3;
Rs=40
```

```
fp=200;  
fs=100;  
F=1000  
Wp=2*fp/F;  
Ws=2*fs/F;  
  
[n,Wn]=buttord(Wp,Ws,Rp,Rs);  
[b,a]=butter(n,Wn,'high');  
  
[h,w]=freqz(b,a,512);  
  
subplot(211);  
plot(w,abs(h));  
xlabel('freq');  
ylabel('magnitude');  
subplot(212);  
plot(w,angle(h));  
xlabel('freq');  
ylabel('phase');
```

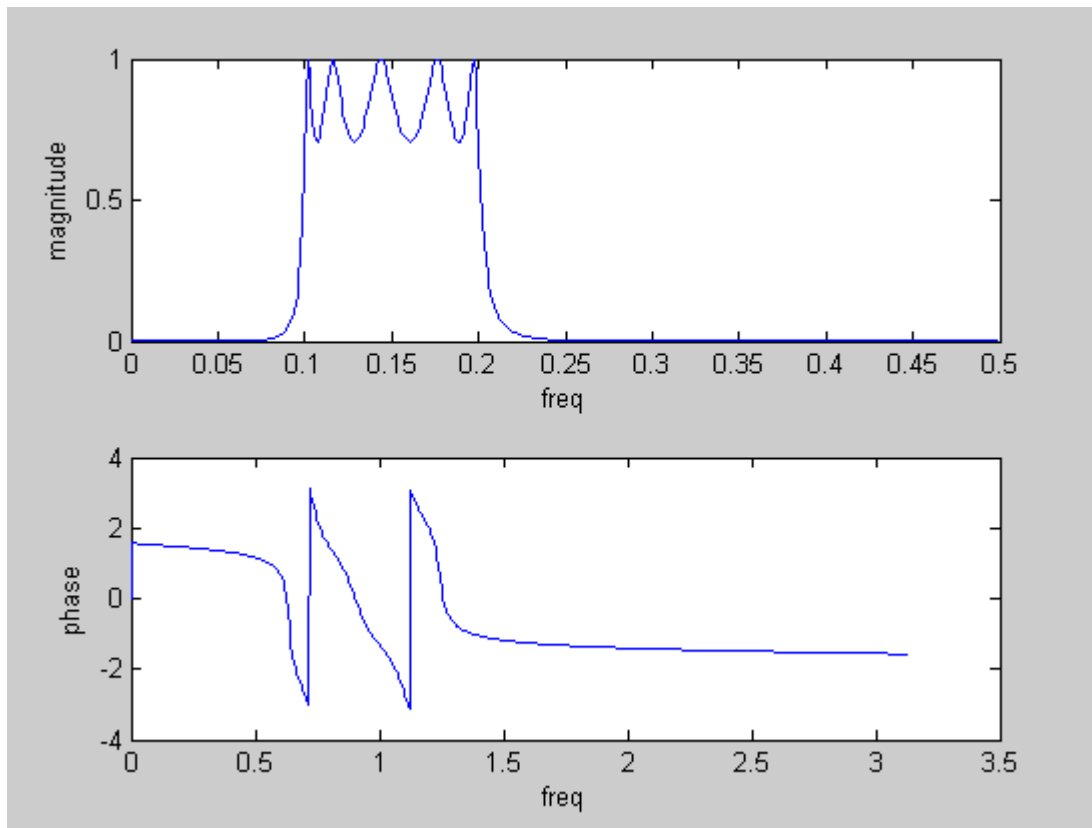


9c) Aim: To Design and implement Digital IIR band pass filter

Qn: Design an IIR band pass digital filter using Chebyshev approximation. The pass band ripple is less than 3db, Stop band attenuation is not more than 40db, and pass band corner frequency are 100Hz and 200 Hz, Stop band edge frequency are 50Hz and 300 Hz, Sampling frequency 1KHz. Plot Magnitude and Phase response of such a filter.

Program:

```
clear all;
close all;
clc;
Rp=3;
Rs=40
fp=[100 200];
fs=[50 250];
F=1000
Wp=2*fp/F;
Ws=2*fs/F;
[n,Wn]=cheblord(Wp,Ws,Rp,Rs);
[b,a]=cheby1(n,Rp,Wn,'bandpass');
[h,w]=freqz(b,a,512);
subplot(211);
plot(w/(2*pi),abs(h));
xlabel('freq');
ylabel('magnitude');
subplot(212);
plot(w,angle(h));
xlabel('freq');
ylabel('phase');
```


RESULT**9d) Aim: To Design and implement Digital IIR band reject filter**

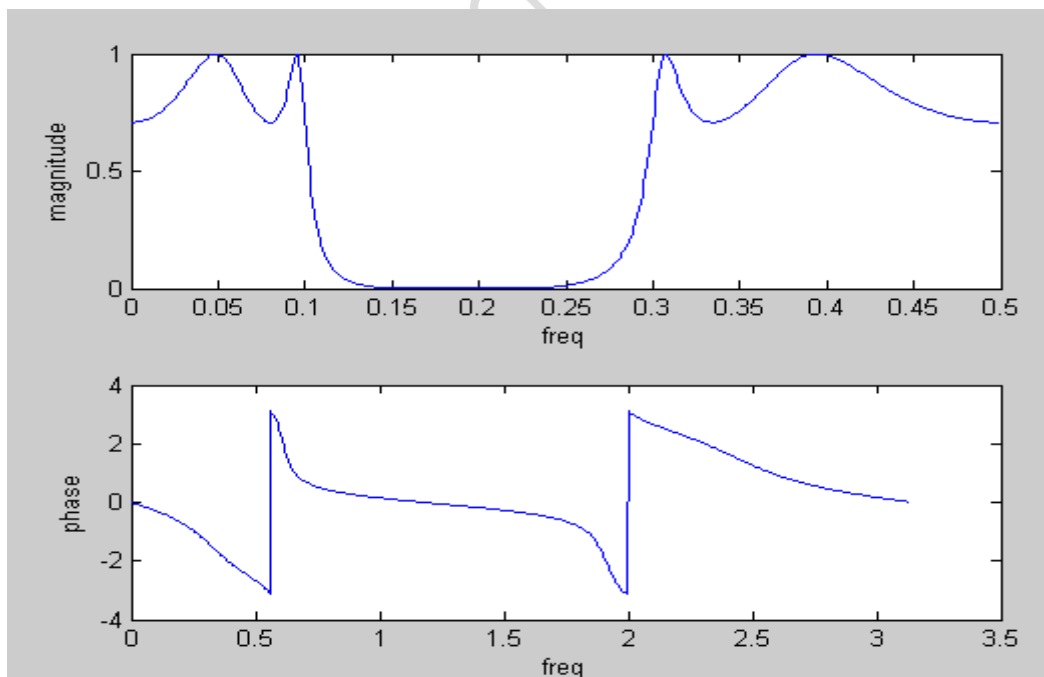
Design an IIR band stop digital filter using Chebyshev approximation. The pass band ripple is less than 3db, Stop band attenuation is not more than 40db, and pass band corner frequency are 100Hz and 300 Hz, Stop band edge frequency are 150Hz and 250 Hz, Sampling frequency 1KHz. Plot Magnitude and Phase response of such a filter.

Program:

```
clear all;
close all;
clc;
Rp=3;
Rs=40
fp=[100 300];
```

```
fs=[150 250];  
F=1000  
Wp=2*fp/F;  
Ws=2*fs/F;  
[n,Wn]=cheblord(Wp,Ws,Rp,Rs);  
[b,a]=cheby1(n,Rp,Wn,'stop');  
[h,w]=freqz(b,a,512);  
subplot(211);  
plot(w/(2*pi),abs(h));  
xlabel('freq');  
ylabel('magnitude');  
subplot(212);  
plot(w,angle(h));  
xlabel('freq');  
ylabel('phase');
```

RESULT



10. FIR FILTER DESIGN USING WINDOWING TECHNIQUE

Aim :10a.:To Design a digital FIR low pass filter with following specifications.

- a) Pass band cut-off frequency : $\omega_P = \underline{\hspace{2cm}}$ rad
 b) Stop band cut-off frequency : $\omega_S = \underline{\hspace{2cm}}$ rad
 c) Stop band attenuation : $k_S = \underline{\hspace{2cm}}$ dB

Choose an appropriate window function and determine impulse response and provide a plot of frequency response of the designed filter.

Design steps:

1. Choose an appropriate window function from the stop band attenuation using the following table.

Sl. No.	Window function $w[n]$	Stop band attenuation	K
1	Rectangular(Boxcar) window	21 dB	2
2	Triangular (Bartlett) window	27 dB	4
3	Hanning window	44 dB	4
4	Hamming window	53 dB	4
5	Blackman window	75 dB	6

2. Determine the order of the FIR filter (number of coefficients) using the relation;

$$N \geq \frac{2\pi k}{(\omega_p - \omega_s)}$$

Note: Choose N as odd. If N is even select next odd integer.

3. Choose the cut-off frequency of the filter as $\omega_C = \omega_P + \frac{\Delta\omega}{2}$

4. Choose $\alpha = \frac{(N-1)}{2}$ so that linear phase is ensured.

5. The impulse response of the filter is given by

$$h(n) = \begin{cases} \frac{\sin(\omega_c(n-\alpha))}{\pi(n-\alpha)} * w(n); & \text{for } 0 < n < N-1 \text{ \& } n \neq \alpha \\ \frac{\omega_c}{\pi} * w(n); & \text{for } n = \alpha \end{cases}$$

where $w(n)$ is window function.

$$H(\omega) = \left(e^{-j\omega\left(\frac{N-1}{2}\right)} \right) * \left[h\left(\frac{N-1}{2}\right) + \sum_{n=0}^{\left(\frac{N-3}{2}\right)} 2h(n) \cos\left[\omega\left(n - \left(\frac{N-1}{2}\right)\right)\right] \right]$$

Design Example

Design a digital FIR low pass filter with following specifications.

- Pass band cut-off frequency : 0.3π rad
- Stop band cut-off frequency : 0.45π rad
- Stop band attenuation : 50 dB

I step: Choose an appropriate window function from the stop band attenuation specification.

As $K_s = 50$ dB select Hamming window.

(Note: Though Blackman window can be used it results in higher transition bandwidth)

II step: Compute the order of the FIR filter (number of coefficients) using the relation,

$$N \geq \frac{2\pi k}{(\omega_p - \omega_s)} \quad [K=4 \text{ for Hamming window}]$$

$$N \geq 53.5 \quad (\text{choose } N=55)$$

III step: Choose the cut-off frequency of the filter as $\omega_c = \omega_p + \frac{\Delta\omega}{2}$ rad.

IV step: Choose $\alpha = \left(\frac{N-1}{2}\right)$ so that linear phase is ensured.

V step: The impulse response of the filter is given by,

$$h(n) = \frac{\sin(\omega_c(n-\alpha))}{\pi(n-\alpha)} * w(n); \text{ for } 0 < n < N-1 \text{ \& } n \neq \alpha.$$

where $w(n)$ is window function.

$$h(n) = \frac{\sin(0.3\pi(n-27))}{\pi(n-27)} * \left[0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)\right]; 0 < n < N-1 \text{ \& } n \neq \alpha$$

For ex : The impulse response coefficients are,

$$h(0) = 0.00029114 = h(54)$$

$$h(1) = -5.9806 * (10^{-4}) = h(53) \dots\dots$$

Note: when $n = \alpha$

$$\text{i.e. } h(27) = \frac{\omega_c}{\pi} * w(27) = 0.3 * 1 = 0.3$$

VI step: Frequency response can be obtained using equation,

$$H(\omega) = \left(e^{-j\omega\left(\frac{N-1}{2}\right)} \right) * \left[h\left(\frac{N-1}{2}\right) + \sum_{n=0}^{\left(\frac{N-3}{2}\right)} 2h(n) \cos\left[\omega\left(n - \left(\frac{N-1}{2}\right)\right)\right] \right]$$

To calculate the Phase, $\theta(\omega)$ of the FIR filter:

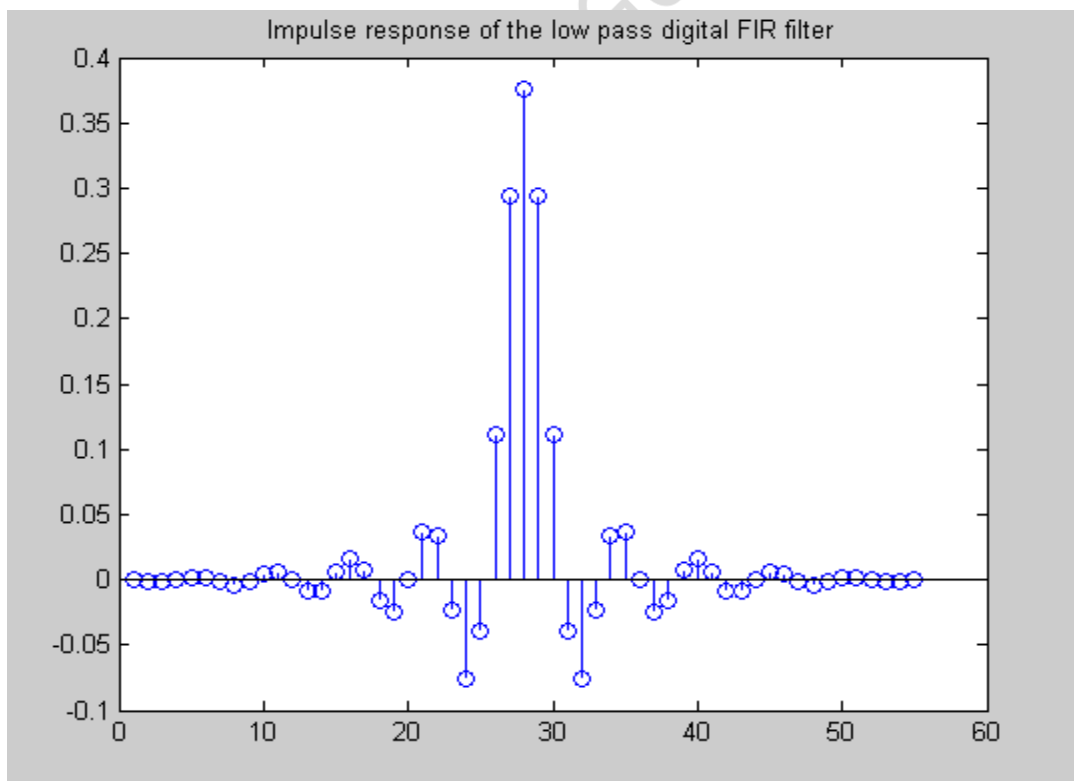
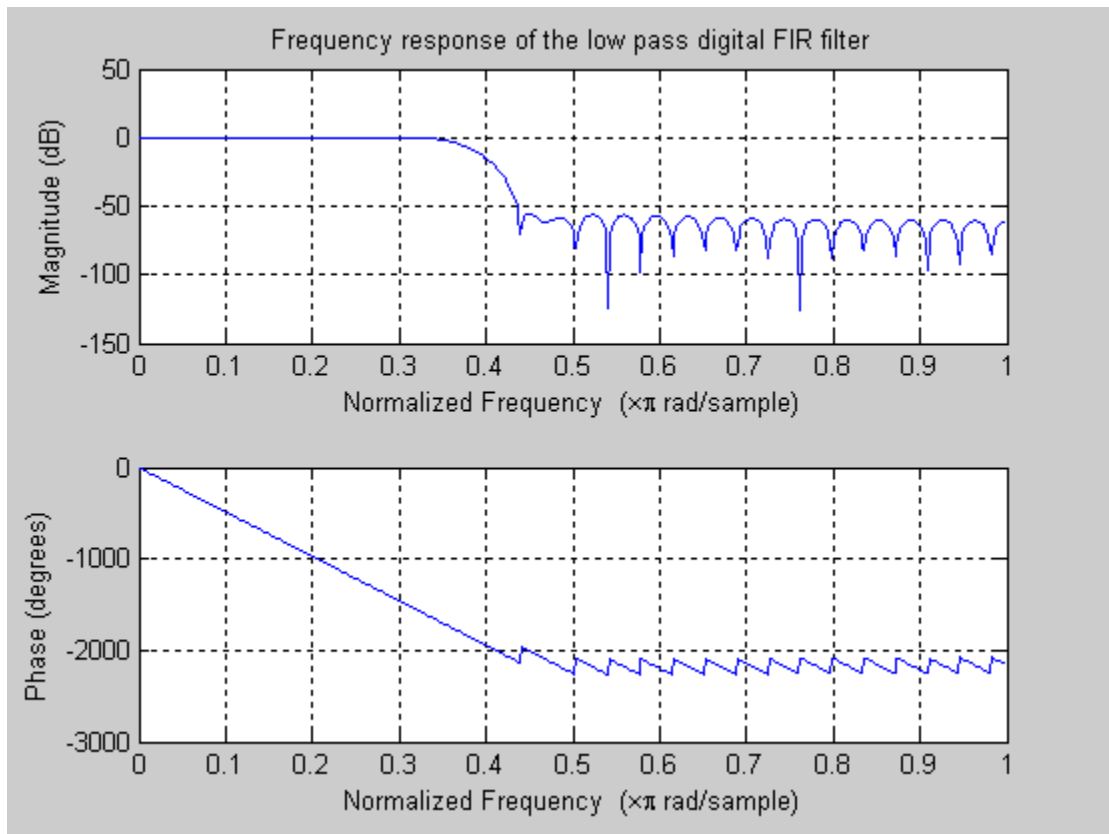
$$\theta(\omega) = \frac{\pi}{2} - \omega\left(\frac{N-1}{2}\right); H(\omega) > 0$$

$$\theta(\omega) = \frac{3\pi}{2} - \omega\left(\frac{N-1}{2}\right); H(\omega) < 0$$

Program:

```
clc;
clear all;
close all;
wp= input('Enter the pass band edge (rad)= ');
ws= input('Enter the stop band edge (rad)= ');
ks= input('Enter the stop band attenuation (dB)= ');
%If 43<Ks<54 choose hamming window.
%To select N,order of filter.
N= (2*pi*4) ./ (ws-wp); % k=4 for Hamming window.
N= ceil(N); %To round-off N to the next integer.
r = rem(N,2); %Choose odd N.
if(r==0)
N=N+1;
end
wc=(wp+(ws-wp)/2) ./pi
% To obtain h(n)
h= fir1(N-1,wc,hamming(N))
% Frequency response
freqz(h,1); % 1 is the normalized frequency
title('Frequency response of the low pass digital FIR filter')
figure(2), stem(h);
title('Impulse response of the low pass digital FIR filter')
```

RESULT



10 a) Designing of a LPF Using Rectangular and Hamming windows**%To design a 25-tap low pass filter with cutoff frequency****% .5pi radians using Rectangular and Hamming windows****%To plot their frequency response.**

```
clear all;
wc= .5*pi
N=25;
alpha = (N-1)/2;eps = .001;
n= 0:1:N-1;
hd = sin(wc*(n-alpha+eps))./(pi*(n-alpha+eps));
wr=boxcar(N);
hn = hd.*wr';
w=0:.1:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h));
hold on
wh = hamming(N);
hn = hd.*wh';
w=0:01:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h,'-.'));grid;
ylabel('magnitude');
xlabel('normalised frequency');
hold off
```


10 b) Designing of a HPF Using Rectangular and Blackman window**%To design a 25-tap high pass filter with cutoff frequency****% .5pi radians using Rectangular and Blackman window****%To plot their frequency response.**

```
clear all;
wc= .5*pi;
N=25;
alpha = (N-1)/2;eps = .001;
n= 0:1:N-1;
hd = (sin(pi*(n-alpha+eps))-sin(wc*(n-alpha+eps)))/(pi*(n-
alpha+eps));
wr=boxcar(N);
hn = hd.*wr';
w=0.1:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h));
hold on
wb = blackman(N);
hn = hd.*wb';
w=0:.01:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h),'-.');
ylabel('magnitude');
xlabel('normalised frequency');
hold off
```

10 c) Designing of a BPF Using Rectangular and Blackman window
%To design a 25-tap Band pass filter for the given specifications.

```
clear all;
wc1= .25*pi; wc2= .75*pi;
N=25;
alpha = (N-1)/2;eps = .001;
n= 0:1:N-1;
hd = (sin(wc2*(n-alpha+eps))-sin(wc1*(n-alpha+eps)))/(pi*(n-
alpha+eps));
wr=boxcar(N);
hn = hd.*wr';
w=0:0.1:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h));
hold on
wh = hamming(N);
hn = hd.*wh';
w=0:0.1:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h),'-.');
ylabel('magnitude');
xlabel('normalised frequency');
hold off
```

10 d) Designing of a BSF Using Rectangular and Blackman window
%To design a 25-tap band reject filter for the given specifications.

```
clear all;
wc1= .25*pi; wc2= .75*pi;
N=25;
a = (N-1)/2;eps = .001;
n= 0:1:N-1;
hd = (sin(wc1*(n-a+eps))-sin(wc2*(n-a+eps))+sin(pi*(n-
a+eps)))/(pi*(n-a+eps));
wr=boxcar(N);
hn = hd.*wr';
w=0:.01:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h));grid;
hold on
wh = hamming(N);
hn = hd.*wh';
w=0:.01:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h),'-.');grid;
hold on
wb= blackman (N);
hn= hd.*wb';
w=0:.01:pi;
h = freqz(hn,1,w);
plot(w/pi,abs(h));grid;
ylabel('magnitude');
xlabel('normalised frequency');
```

Aim :10a.:To Design a digital FIR high pass filter with following specifications.

Program:

```
fs=8000;

n=50;

w=1200/ (fs/2);

b=fir1(n,w,'high');

freqz(b,1,128,8000);

% this function plots the phase(degree)and magnitude in db

subplot(2,1,2)

figure(2)

[h,w]=freqz(b,1,128,8000);

plot(w,abs(h));

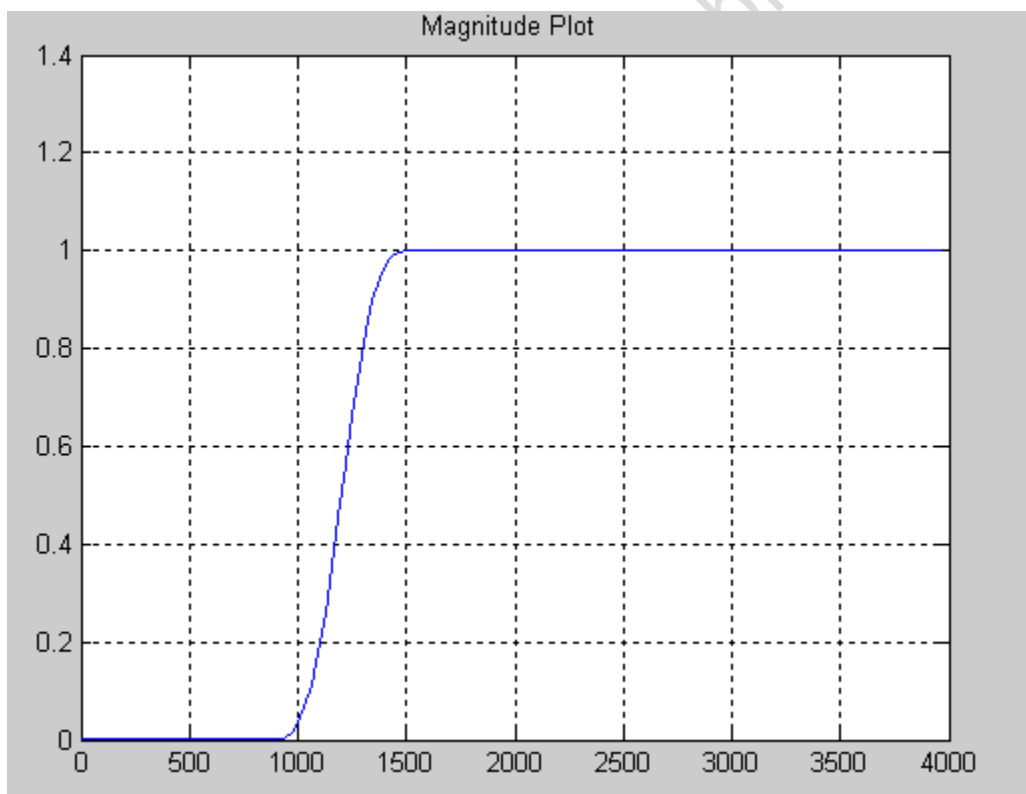
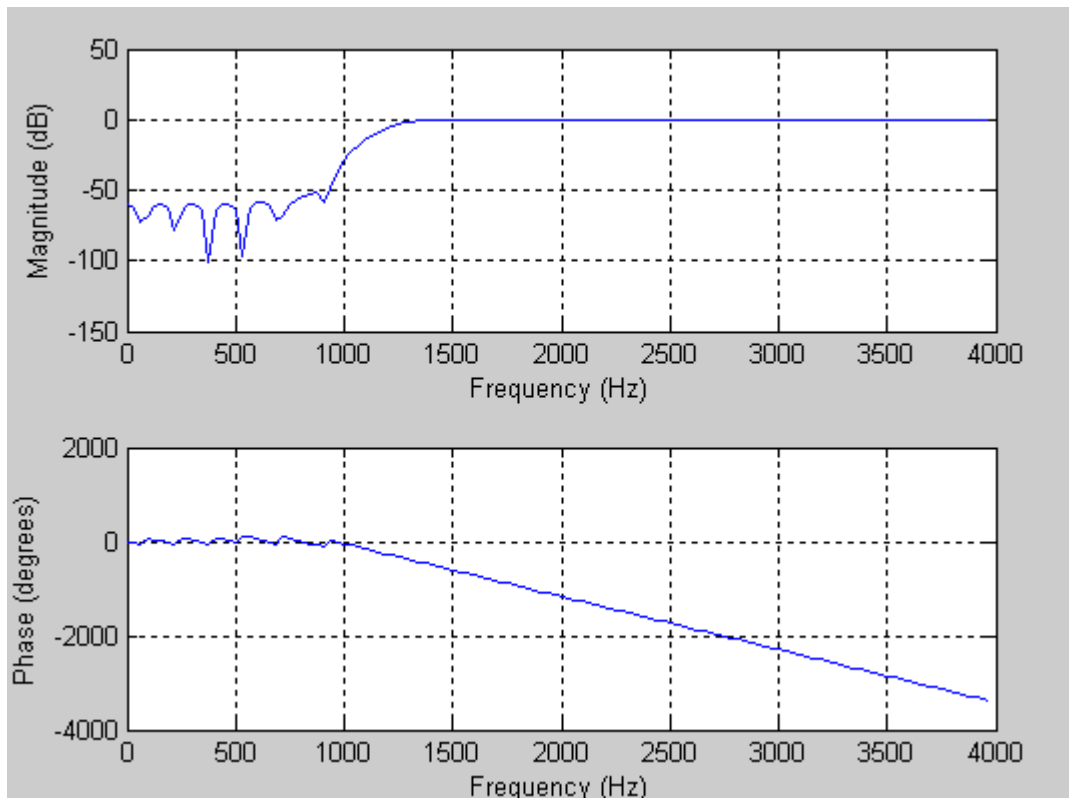
% Normalized Magnitude Plot

title('Magnitude Plot ');

grid on

figure(3)

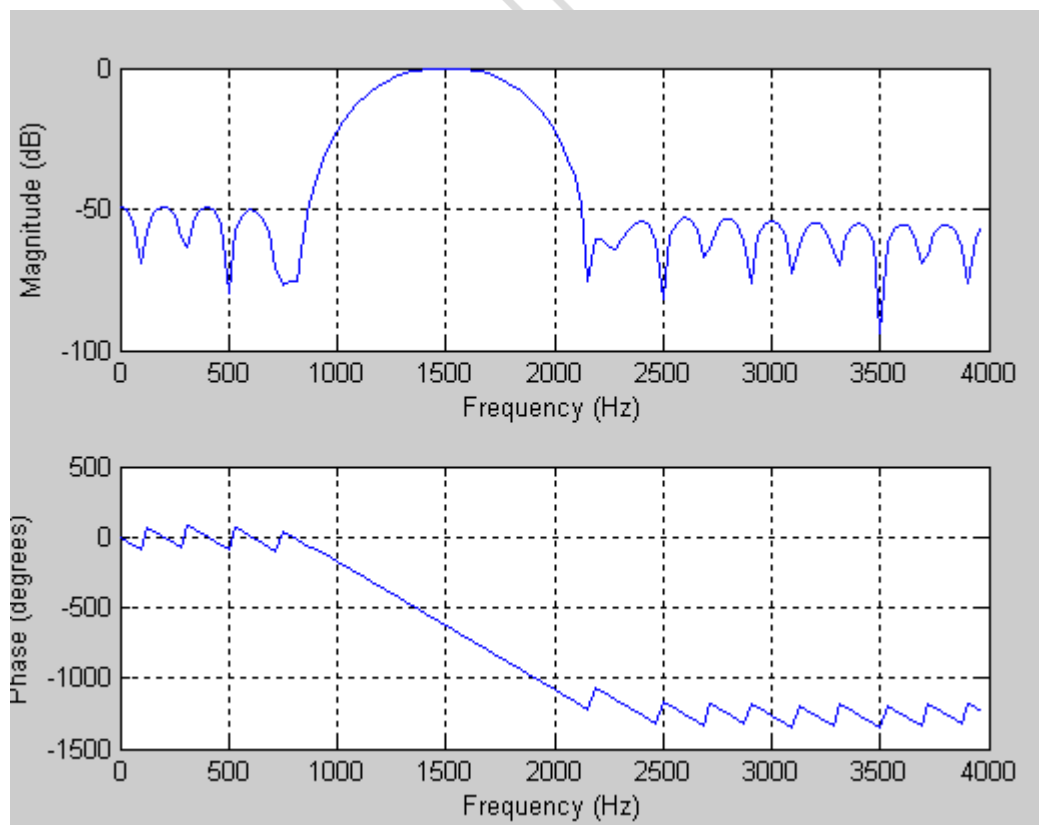
zplane(b,1); % this function plots fiq in zplane
```

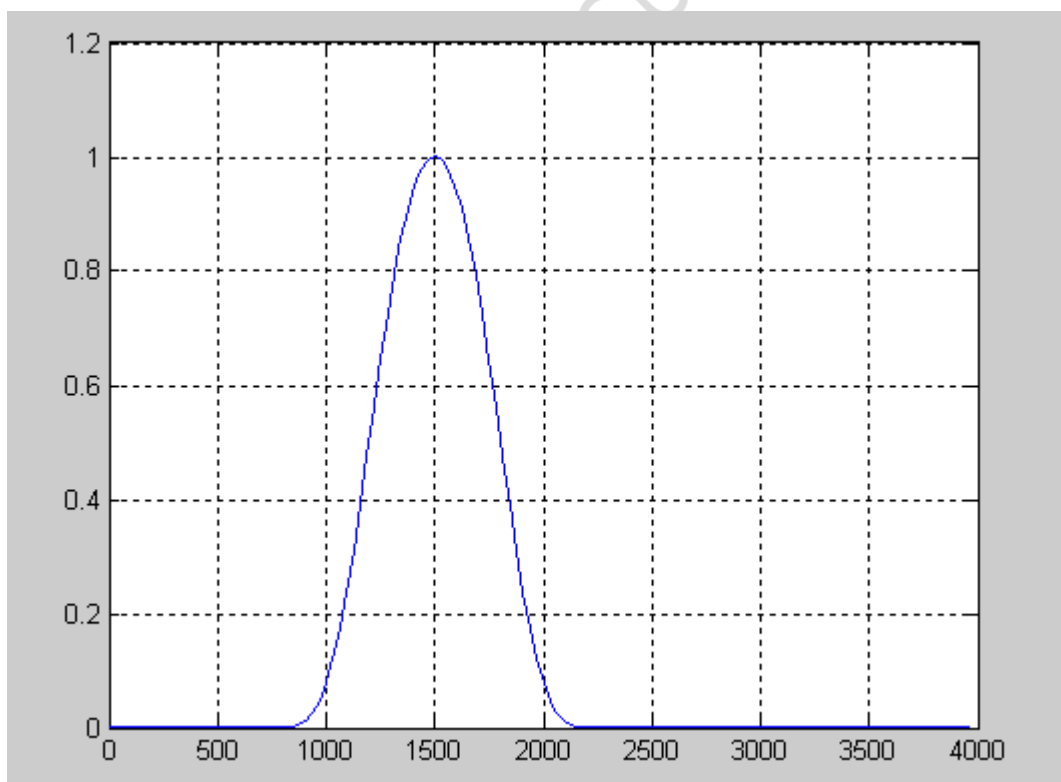
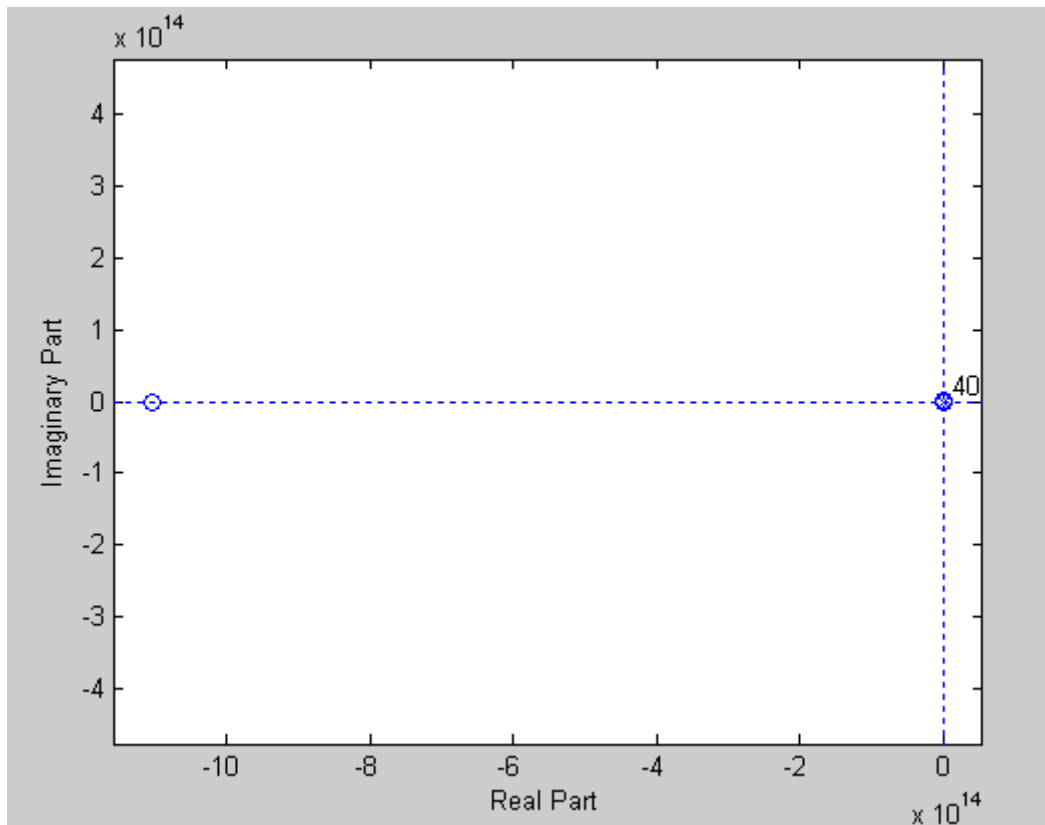


Aim :10b.:To Design a digital FIR band pass filter with following specifications.

Program:

```
fs=8000;  
n=40;  
b=fir1(n,[1200/4000 1800/4000],'bandpass');  
freqz(b,1,128,8000)  
figure(2)  
[h,w]=freqz(b,1,128,8000);  
plot(w,abs(h));  
% Normalized Magnitude Plot  
Grid on  
figure(3)  
zplane(b,1);
```

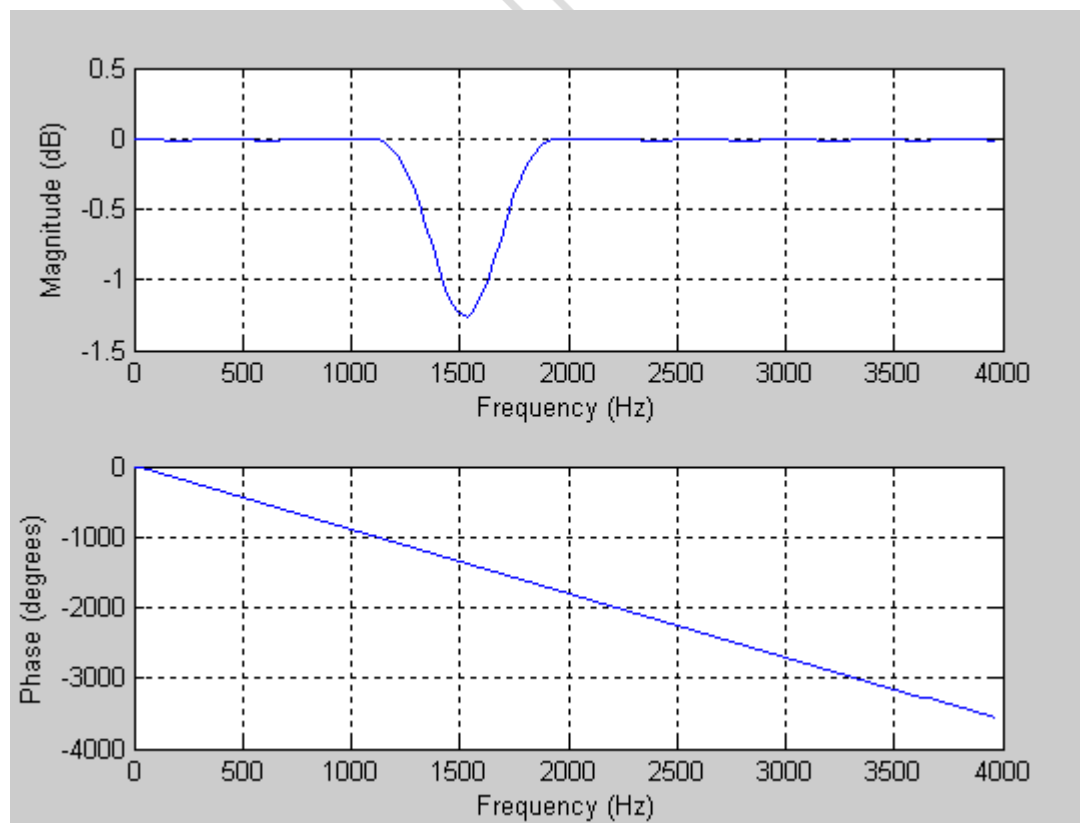


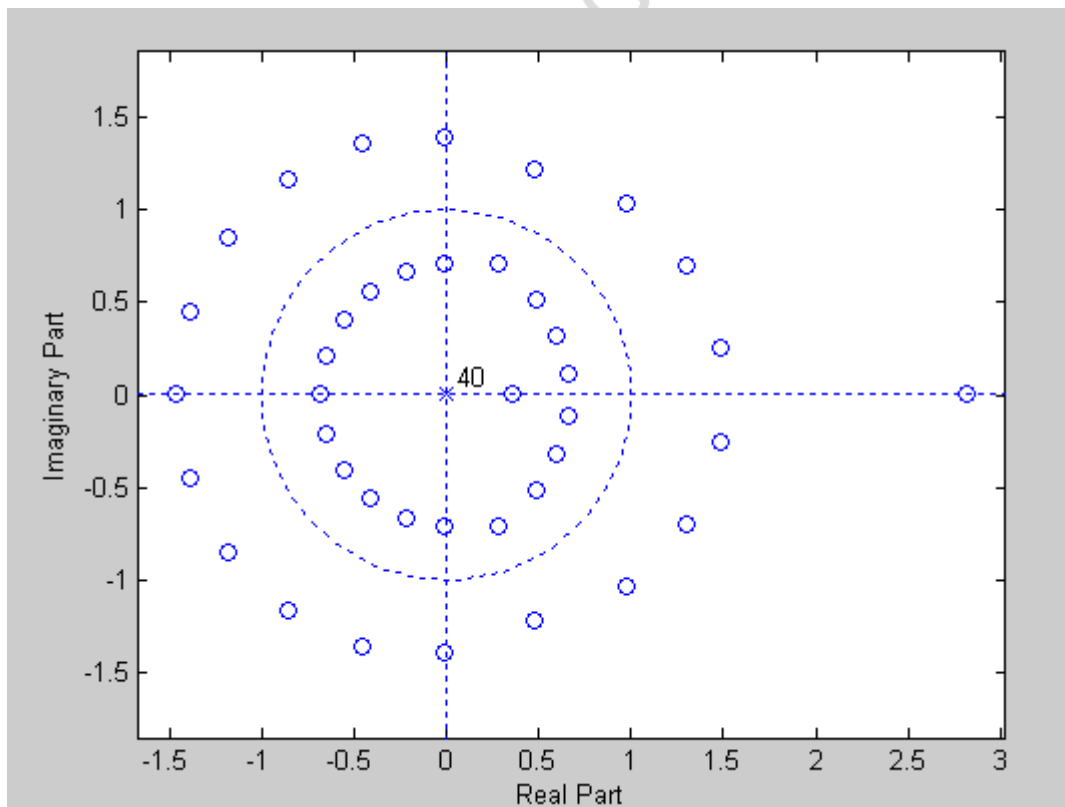
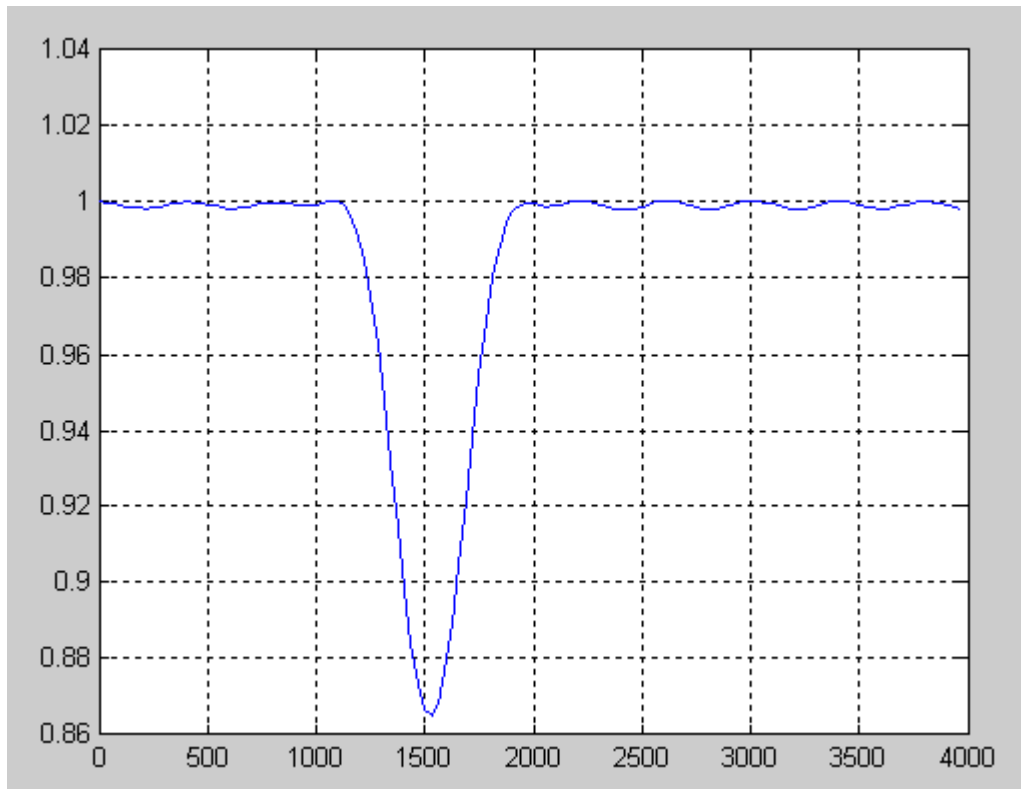


Aim :10c.:To Design a digital FIR band reject filter with following specifications.

Program:

```
fs=8000;  
  
n=40;  
  
b=fir1(n,[1500/4000 1550/4000],'stop');  
  
freqz(b,1,128,8000)  
  
figure(2)  
  
[h,w]=freqz(b,1,128,8000);  
  
plot(w,abs(h));  
  
% Normalized Magnitude Plot  
  
grid on  
  
figure(3)  
  
zplane(b,1);
```



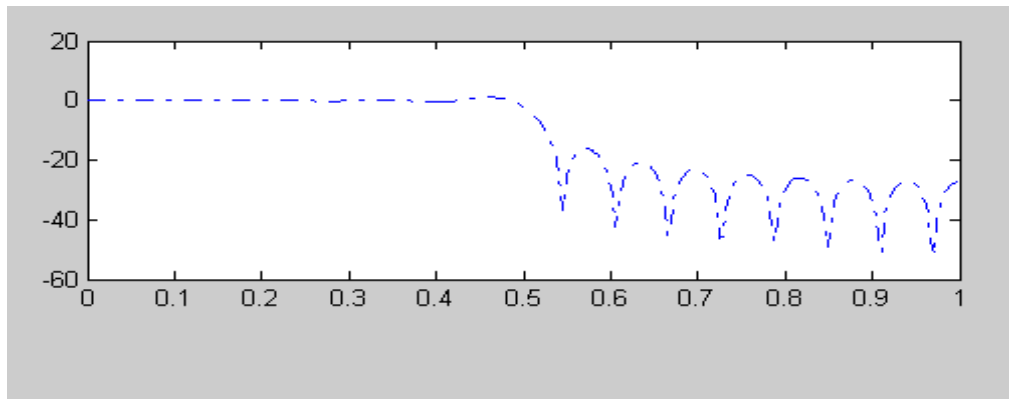


11. FIR FILTER DESIGN USING FREQUENCY SAMPLING TECHNIQUE

```

clear all;
clc;
N=33;
alpha=(N-1)/2;
Hrk=[ones(1,9), zeros(1,16), ones(1,8)];
k1=0:(N-1)/2;
k2=(N+1)/2:N-1;
theetak=[(-alpha*(2*pi)/N)*k1, (alpha*(2*pi)/N)*(N-k2)];
Hk=Hrk.*(exp(i*theetak));
hn=real(iff(Hk,N));
w=0:0.01:pi;
H=freqz(hn,1,w);
q=20*log10(abs(H))
subplot(211);plot(w/pi,q,'-.');
hold on;
hrk=[ones(1,9),0.5,zeros(1,14),0.5,ones(1,8)];
k1=0:(N-1)/2;
k2=(N+1)/2:N-1;
theethak=[-alpha*(2*pi)/N*k1, alpha*(2*pi)/N*(N-k2)];
Hk=Hrk.*(exp(i*theetak));
Hn=real(iff(Hk,N));
w=0:0.01:pi;
H=freqz(hn,1,w);
subplot(212);plot(w/pi,20*log10(abs(H)));
xlabel('normalized frequency');
ylabel('magnitude in db');
hold off

```

**Prog 11.a**

```
function xn=ifft(Xk,N)
%computation of twiddle factor
for k=0:1:N-1;
    for n=0:1:N-1;
        p=exp(i*2*pi*n*k/n);
        x2(k+1, n+1)=p;
    end
end
xn=(Xk*x2.')./N;
```

Prog 11.b

```
Xk = [4,1+i,0,1-i,0,1+i,0,1-i]

N=length(Xk);
xn=ifft(Xk,N);
```

Chebyshev Low-pass

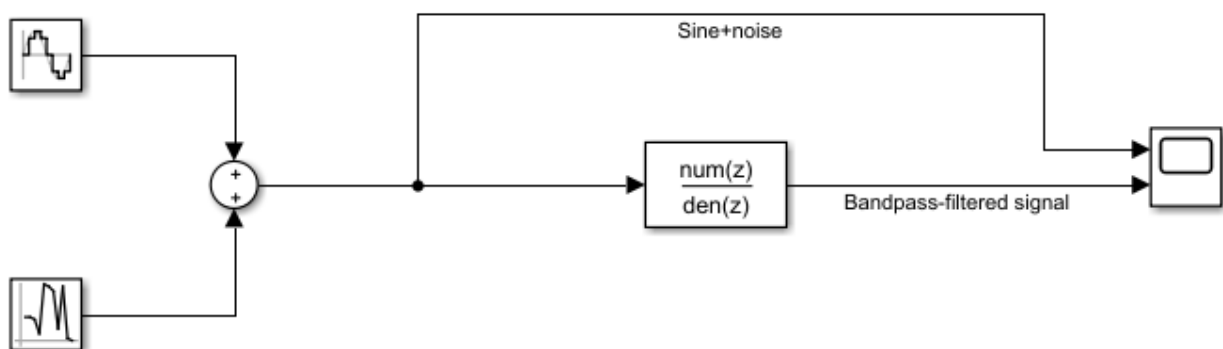
```
clear all;
alphap=1;
alphas=15;
wp=.2*pi;
ws=.3*pi
% order and cut off
[n, wn]=cheblord(wp/pi,ws/pi,alphap,alphas);
%system function
[b,a]=cheby1(n,alphap,wn);
w=0:.01:pi;
[h,ph]=freqz(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1); plot(ph/pi,m);grid;
ylabel('gain in db');
subplot(2,1,2); plot(ph/pi,an);grid;
ylabel('phase');
[b,a]=cheby1(n,alphap,wn)
```

12. REALIZATION OF IIR AND FIR FILTERS

Aim :To implement IIR and FIR filters.

PROCEDURE:

- Open a new Simulink model.
- Implement using the following circuit.
- Change parameters as per requirement.
- Run the program.
- Observe the output in connected oscilloscope.



REFERENCES

- [1]. Digital signal processing using MATLAB –SanjeetMitra, TMH,2001.
- [2]. Digital signal processing using MATLAB - J. G. Proakis&Ingale, MGH, 2000.
- [3]. Digital Signal Processors, B. Venkataramani and Bhaskar, TMH,2002.
- [4]. A.Gilat, *MATLAB: An Introduction with Applications*, John Wiley & Sons, Inc. 2004.
- [5]. R. Pratap, *Getting Started with MATLAB 7: A Quick Introduction for Scientists and Engineers*, Oxford University Press, 2009.
- [6]. D. Hanselman and B. Little_eld, *Mastering MATLAB 5: A Comprehensive Tutorial and Reference*, Prentice Hall, 1998.
- [7]. B. R. Hunt, R. L. Lipsman, and J. M. Rosenberg (with K. R. Coombes, J. E. Osborn, and G. J. Stuck), *A Guide to MATLAB: for beginners and experienced users*, Cambridge University Press 2001.
- [8] RulphChassaing, *“DSP Applications Using C and the TMS320C6X DSK”*, John Wiley, New York, 2002

VIVA

- 1) What is DSP? Why do we need DSP?
- 2) List the merits and demerits of DSP
- 3) What are the applications of DSP?
- 4) What is sampling theorem?
- 5) What is aliasing effect?
- 6) Define Quantization.
- 7) Explain the principle of operation of analog to digital conversion of the signal with a neat diagram.
- 8) Explain the significance of Nyquist rate and aliasing during the sampling of continuous time signals.
- 9) What do you mean by process of reconstruction?
- 10) What are techniques of reconstructions?
- 11) What do you mean Aliasing? What is the condition to avoid aliasing for sampling?
- 12) Write the conditions of sampling. How many types of sampling there?
- 13) Explain the statement: $t = 0:0.000005:0.05$
In the above example what does colon (:) and semicolon (;) denotes.
- 14) What is a) Undersampling b) nyquist plot c) Oversampling.
- 15) What is MATLAB? What are the applications of MATLAB?
- 16) What is the use of command 'legend'?
- 17) Distinguish between 'plot' and 'stem' functions.
- 18) Explain the function 'subplot'?
- 19) What is convolution? What are the properties of convolution?
- 20) What is linear convolution?
- 21) Distinguish between Linear convolution and circular convolution.
- 22) How will you obtain linear convolution from circular convolution?
- 23) Explain how convolution syntax built in function works.
- 24) What is the total output length of linear convolution sum?
- 25) What is an LTI system?
- 26) Describe impulse response of a function.
- 27) What is the difference between convolution and filter?
- 28) Where to use command 'filter' or 'impz', and what is the difference between these two?
- 29) What is the use of function command 'deconv'?

- 30) What is the difference between linear and circular convolution?
- 31) What do you mean by statement subplot (3,3,1).
- 32) What do you mean by command "mod" and where it is used?
- 33) How to calculate output of DFT using MATLAB?
- 34) Distinguish between DFT and DTFT.
- 35) Where DFT is used?
- 36) What is the difference between DFT and IDFT?
- 37) What do you mean by built in function 'abs' and where it is used?
- 38) What do you mean by phase spectrum and magnitude spectrum/give comparison.
- 39) How to compute maximum length N for a circular convolution using DFT and IDFT. (What is the command?).
- 40) Explain the statement- $y=x1.*x2$
- 41) What is FFT?
- 42) Why FFT is needed?
- 43) What is meant by Radix-2 FFT?
- 44) What is decimation-in-time algorithm?
- 45) What is decimation in frequency algorithm?
- 46) What are the differences and similarities between DIF and DIT algorithm?
- 47) How many stages are there for 8-point DFT?
- 48) How many multiplication terms are required for doing DFT by expressional method and FFT method?
- 49) What is the main advantage of FFT?
- 50) What are the applications of FFT algorithms?
- 51) What do you mean by 'filtic' command, explain.
- 52) How to calculate output length of the linear and circular convolution?
- 53) What do you mean by built in function 'fliplr' and where we need to use this?
- 54) What is steady state response?
- 55) Which built in function is used to solve a given difference equation?
- 56) Explain the concept of difference equation.
- 57) What is filter?
- 58) What are the types of digital filter according to their impulse response?
- 59) What is FIR Filter?
- 60) Write the procedure for designing FIR filters

- 61) Write the characteristics of FIR filter.
- 62) What are the advantages and disadvantages and applications of FIR FILTER?
- 63) What is GIBBS phenomenon?
- 64) Write the desirable characteristics of frequency response of window functions.
- 65) Write the characteristics features of rectangular window.
- 66) List merits and demerits of rectangular window.
- 67) What do you understand by linear phase response?
- 68) What are the advantages of Kaiser Window?
- 69) What is the need for employing window technique for FIR filter design?
- 70) What is windowing and why it is necessary?
- 71) What is the necessary and sufficient condition for linear phase characteristic in FIR filter?
- 72) Define IIR filter.
- 73) What are the methods available for designing analog IIR filter?
- 74) Mention the importance of IIR filter.
- 75) What is warping effect?
- 76) Compare analog and digital filter.
- 77) What is FIR and IIR filter define, and distinguish between these two.
- 78) Explain the command - $N = \text{ceil}(6.6 * \pi / \text{tb})$
- 79) What is the matlab command for Hamming window? Explain.
- 80) What do you mean by cut-off frequency?
- 81) What are the differences between FIR and IIR filters?
- 82) What are the differences between Butterworth and Chebyshev filters?
- 83) What do you mean by command 'butter' and 'cheby1'?
- 84) Explain the command in detail- $[N, \text{wc}] = \text{buttord}(2 * \text{fp} / \text{fs}, 2 * \text{fstp} / \text{fs}, \text{rp}, \text{As})$

DIGITAL SIGNAL PROCESSING LABORATORY

Question Bank

Programs based on MATLAB implementation:

- 1) By using MATLAB implementation sample a band limited continuous time signal band limited to $f_m = \dots\dots\dots$ Hz i) $f_s < 2f_m$ ii) $f_s = 2f_m$ iii) $f_s \leq 2f_m$
- 2) Using MATLAB Compute the linear convolution of two given finite length of sequences. Given: $x(n)=[\dots\dots\dots]$ and $h(n)=[\dots\dots\dots]$ using matrix method.
- 3) Using MATLAB Compute circular convolution of two given finite length of sequences. Given: $x(n)=[\dots\dots\dots]$ and $h(n)=[\dots\dots\dots]$ using matrix method.
- 4) For a given sequence $x(n)=[\dots\dots\dots]$. Compute the N-point DFT using MATLAB. Plot magnitude and phase spectrum. (with equation method and verify using built in function)
- 5) Obtain the complete response of the system described by the following difference equation $\dots\dots\dots$ for the input $x(n)=[\dots\dots\dots]$ With initial conditions $\dots\dots\dots$. Use MATLAB as a tool.
- 6) Using MATLAB design a low-pass FIR filter with the following Specifications:
 - a) Pass band cut-off frequency : $\omega_p = \dots\dots\dots$ rad
 - b) Stop band cut-off frequency : $\omega_s = \dots\dots\dots$ rad
 - c) Stop band attenuation : $k_s = \dots\dots\dots$ dB
 Plot the Frequency, phase and impulse response of the designed filter.
- 7) Using MATLAB, design a low-pass IIR Butterworth filter with the following specifications.

$k_p =$ Pass band attenuation(dB) =

$k_s =$ Stop band attenuation(dB) =

$f_p =$ Pass band edge(Hz) =

$f_s =$ Stop band edge(Hz) =

$$F_s = \frac{1}{T} = \text{Sampling rate (samples/sec)} =$$

Plot the Frequency, phase and impulse response of the designed filter.

- 8) Using MATLAB, design a low-pass IIR Chebyshev type 1 filter with the following specifications.

k_P = Pass band attenuation(dB) =

k_S = Stop band attenuation(dB) =

f_P = Pass band edge(Hz) =

f_S = Stop band edge(Hz) =

$F_s = \frac{1}{T_s}$ = Sampling rate (samples/sec) =

