ATME College of Engineering

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

LABORATORY MANUAL

SIGNALS & DIGITAL SIGNAL PROCESSING LABORATORY

(IPCC) BEE502

ACADEMIC YEAR 2024-25

SEMESTER: V

Prepared by	**Verified by**	**HoD**
Ms. Kavyashree S	Ms. Swapna H	
(Lab Incharge)		

INSTITUTIONAL VISION AND MISSION

VISION:

Development of academically excellent, culturally vibrant, socially responsible and globally competent human resources.

MISSION:

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional and moral foundations and shine as torchbearers of tomorrow's society.
- To strive to attain ever-higher benchmarks of educational excellence

DEPARTMENT VISION AND MISSION

VISION:

To create Electrical and Electronics Engineers who excel to be technically competent and fulfill the cultural and social aspirations of the society.

MISSION:

- To provide knowledge to students that builds a strong foundation in the basic principles of electrical engineering, problem solving abilities, analytical skills, soft skills and communication skills for their overall development.
- To offer outcome based technical education.
- To encourage faculty in training & development and to offer consultancy through research & industry interaction.

PROGRAMME OUTCOMES:

Engineering Graduates will be able to:

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of EXPERIMENTs, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

At the end of graduation, the student will be able,

PSO1: Apply the concepts of Electrical & Electronics Engineering to evaluate the performance of power systems and also to control Industrial drives using power electronics.

PSO2: Demonstrate the concepts of process control for Industrial Automation, design models for environmental and social concerns and also exhibit continuous self- learning.

Program Educational Objectives (PEOs)

PEO1: To produce competent and Ethical Electrical and Electronics Engineers who will exhibit the necessary technical and managerial skills to perform their duties in society.

PEO2: To make students continuously acquire and enhance their technical and socio-economic skills.

PEO3: To aspire students on R&D activities leading to offering solutions and excel in various career paths.

PEO4: To produce quality engineers who have the capability to work in teams and contribute to real time projects.

SIGNALS & DIGITAL SIGNAL PROCESSING LABORATORY

Course Code	**:**	**BEE502(IPCC)**	**CIE**	**:**	**50**
Hours/Week	**:**	**02**	**Exam Hours**	**:**	**03**
Total Hours	**:**	**10 Slots**	**SEE**	**:**	

LIST OF EXPERIMENTS

Exp no	Name of the Experiment	
1.	Generation of different signals in both continuous and discrete time domains.	CO1
2.	Verification of Sampling Theorem both in time and frequency domains	CO1
3.	To perform basic operations on given sequences- Signal folding, evaluation of even and odd components.	CO1
4.	Evaluation of impulse response of a system	CO1
5.	Solution of a given difference equation.	CO1
6.	Evaluation of linear convolution and circular convolution of given sequences.	CO2
7.	Computation of N- point DFT and IDFT of a given sequence by use of (a) Defining equation; (b) FFT method.	CO2,3
8.	Evaluation of circular convolution of two sequences using DFT and IDFT approach.	CO2
9.	Design and implementation of IIR filters to meet given specification (Low pass, high pass, band pass and band reject filters).	CO4
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.	CO5
11.	Design and implementation of FIR filters to meet given specifications (Low pass, high pass, band pass and band reject filters) using frequency sampling technique.	CO5
12.	Realization of IIR and FIR filters.	CO4,5

REFERNCE BOOKS:

1. Introduction to Digital Signal Processing, Jhonny R. Jhonson, Pearson, 1 st Edition, 2016.
2. . Digital Signal Processing – Principles, Algorithms, and Applications, Jhon G. ProakisDimitris G. Manolakis, Pearson 4th Edition, 2007.
3. Digital Signal Processing, A.NagoorKani, McGraw Hill, 2nd Edition, 2012.

COURSE OUTCOMES

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

CO1:Perform elementary signal operations, **apply** convolution for both continuous & discrete time signals and to understand sampling theorem. [L3]

CO2:**Evaluate** Discrete Fourier Transform of a sequence, to understand the various Properties of DFT and Signal segmentation using overlap save and add method. [L4]

CO3:**Evaluate** Discrete Fourier Transform of a sequence using decimation in time and decimation in frequency methods. [L4]

CO4:**Design** Butter worth and Chebyshev IIR digital filters and to represent the IIR filters using different methods. [L4]

CO5: **Design** FIR filters using window method and frequency sampling method and to represent FIR filters using direct method and lattice method. [L4]

Cycle of Experiments

Exp no	Cycle-I	Delivery Plan
	Lab Orientation and Introduction to Software Tool	Sep (Week-2)
1.	Generation of different signals in both continuous and discrete time domains.	Sep (Week-3)
2.	Verification of Sampling Theorem both in time and frequency domains	Sep (Week-4)
3.	To perform basic operations on given sequences- Signal folding, evaluation of even and odd components.	Sep (Week-5)
4.	Evaluation of impulse response of a system	Oct (Week-1)
5.	Solution of a given difference equation.	Oct (Week-2)
6.	Evaluation of linear convolution and circular convolution of given sequences.	Oct (Week-4)
	Cycle -II	
7.	Computation of N- point DFT and IDFT of a given sequence by use of (a) Defining equation; (b) FFT method.	Oct (Week-5)
8.	Evaluation of circular convolution of two sequences using DFT and IDFT approach.	Nov (Week-2)
9.	Design and implementation of IIR filters to meet given specification (Low pass, high pass, band pass and band reject filters).	Nov (Week-3)
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.	Nov (Week-4)
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.	Nov (Week-5)
12.	Realization of IIR and FIR filters.	Dec (Week-1)

Table of Contents

Experiment no. 1

Generation of different signals in both continuous and discrete time domains

Objective: To learn the generation of various signals such as unit impulse, unit step, ramp and sine signal in continuous and discrete time domains

Theory:

1. Unit impulse signal in continuous time domain: $\delta(t)$ is defined as an infinitely tall and narrow pulse at $t=0$, with an area of 1.
   Discrete time domain: $\delta[n]$ is a sequence where $\delta[n]=1$at $n=0$ and $\delta[n]=0$ elsewhere.
2. Unit step signal in continuous time domain: $u(t)$ is a function that equals 0 for $t<0$ and 1 for $t\geq 0$
   Discrete time domain: $u[n]$ is a sequence where $u[n]=0$ for $n<0$ and $u[n]=1$ for $n\geq 0$.
3. Ramp signal in continuous time domain: $r(t)r(t)$ is a function increasing linearly with time, given by $r(t)=t\cdot u(t)$.
   Discrete time domain: $r[n]$ is a sequence increasing linearly with time, given by $r[n]=n\cdot u[n]$
4. Sine signal in continuous time domain: $x(t)=A\sin(\omega t+\phi)$ where A is amplitude, ω is angular frequency, and $\phi$is phase angle.
   Discrete time domain: $x[n]=A\sin(\omega n+\phi)$, where A, ω, and ϕ have the same meanings as in continuous time.

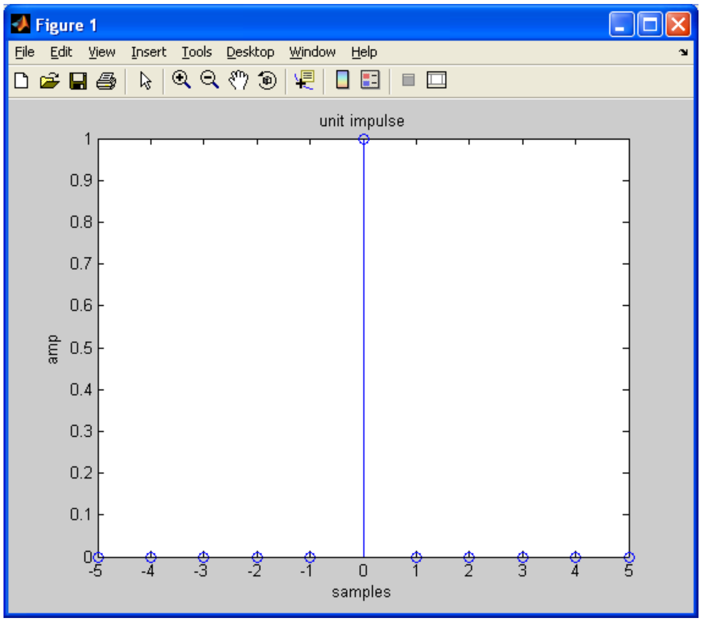
Program:

1. **Unit Impulse**

```
clear all;
n=input ('enter the value of n');
x= -n:1:n; %Time duration
y= [zeros (1, n),ones(1,1),zeros(1,n)];
stem(x,y);
xlabel('time');
ylabel('amplitude');
title('the unit impulse signal is');
```

OUTPUT:

enter the no 5

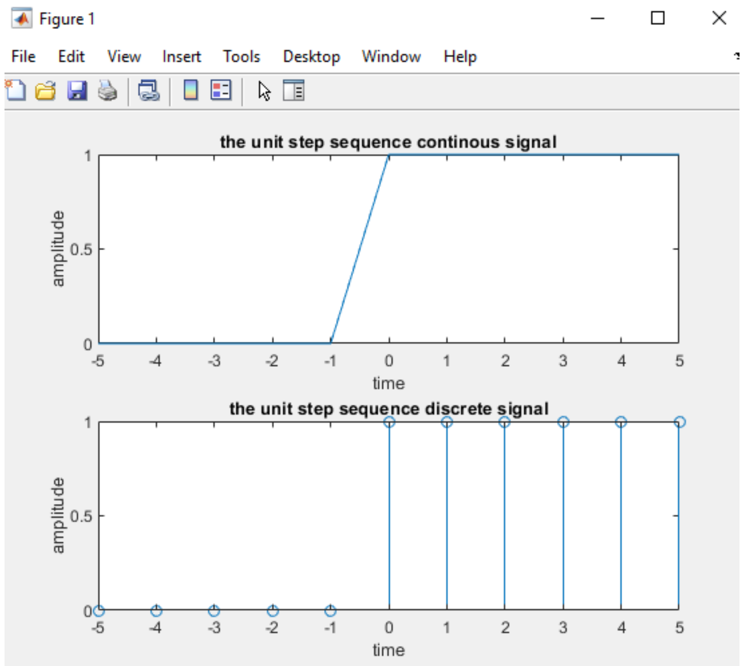


2. UNIT STEP:

```
clc;
n=input ('Enter the number n:');
t=-n:1:n;
y=[zeros(1,n),ones(1,n+1)];
figure();
subplot(2,1,1);
plot(t,y);
xlabel('time');
ylabel('amplitude');
title('the unit step sequence continous signal');
subplot(2,1,2);
stem (t,y);
xlabel('time');
ylabel('amplitude');
title('the unit step sequence discrete signal');
```

OUTPUT:

Enter the number n: 5

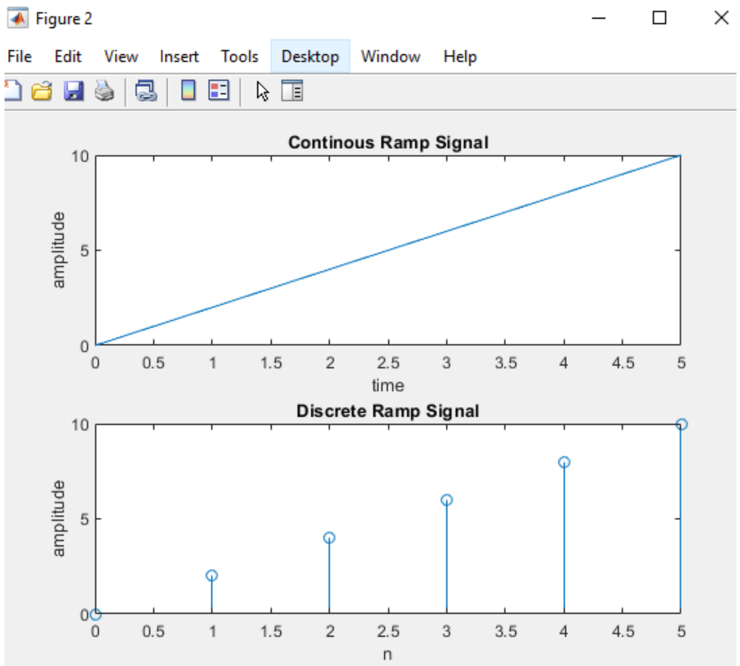


3. Ramp Signal

```
clear all;
n=input('Enter the number of sample n: ');
a=input('Enter the coefficient: ');
t=0:1:n;
y=a*t;
figure();
subplot(2,1,1);
plot(t,y);
xlabel('time');
ylabel('amplitude');
title('Continous Ramp Signal');
subplot(2,1,2);
stem(t,y);
xlabel('n');
ylabel('amplitude');
title('Discrete Ramp Signal');
```

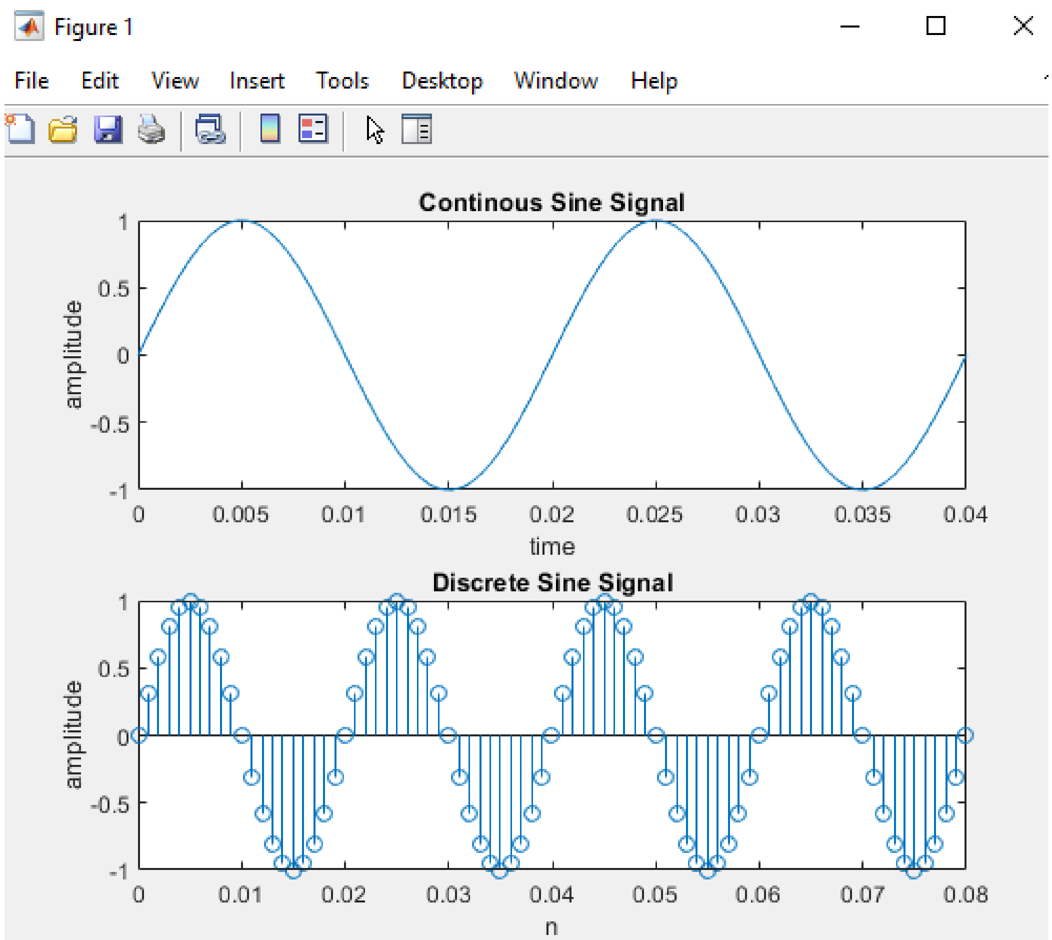
OUTPUT:

Enter the number of sample n: 5
Enter the coefficient: 2



4. Sine Signal

```
f=50;
subplot(2,1,1);
t1=0:0.00001:0.04
y1=sin(2*pi*f*t1);
plot(t1,y1);
xlabel('time');
ylabel('amplitude');
title('Continous Sine Signal');
subplot(2,1,2);
t2=0:0.001:0.08
y2=sin(2*pi*f*t2);
stem(t2,y2);
xlabel('n');
ylabel('amplitude');
title('Discrete Sine Signal');
```



Result: Generation of ________________________________ signals have been done in __________________________ and ____________________ Domains.

Experiment no. 2

Verification of Sampling Theorem both in time and frequency Domains

Objective:

- To learn the physical interpretation of sampling theorem.
- To Verify sampling theorem both in time and frequency domains.

Theory: In digital signal processing, the **sampling theorem** is the basic bridge between continuous-time signals and discrete-time signals.

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

$$f_S \geq 2f_m$$

Program:

```
clc;
  clear;
     f=50
         t=0:0.00001:0.04
           y=sin(2*pi*f*t);
subplot(3,3,1);
plot(t,y);

% for sampling frequency=supply frequency (fs=f)

fs1=50;

   n=0:1;

     y1=sin(2*pi*f*n/fs1);

subplot(3,3,2);

       stem(n,y1);

hold on

subplot(3,3,3)
```

```
plot(n,y1)

% for sampling frequency ≥ 2*supply frequency (fs>2*f)
fs2=500;
     n=0:20;
        y2=sin(2*pi*f*n/fs2);
subplot(3,3,4);
       stem(n,y2);
hold on
subplot(3,3,5)
plot(n,y2)

% for sampling frequency >≥ 2*supply frequency (fs >>2*f)

  fs3=1000;
       n=0:40;
          y3=sin(2*pi*f*n/fs3);
subplot(3,3,6);
         stem(n,y3);
hold on
subplot(3,3,7);
plot(n,y3);
```

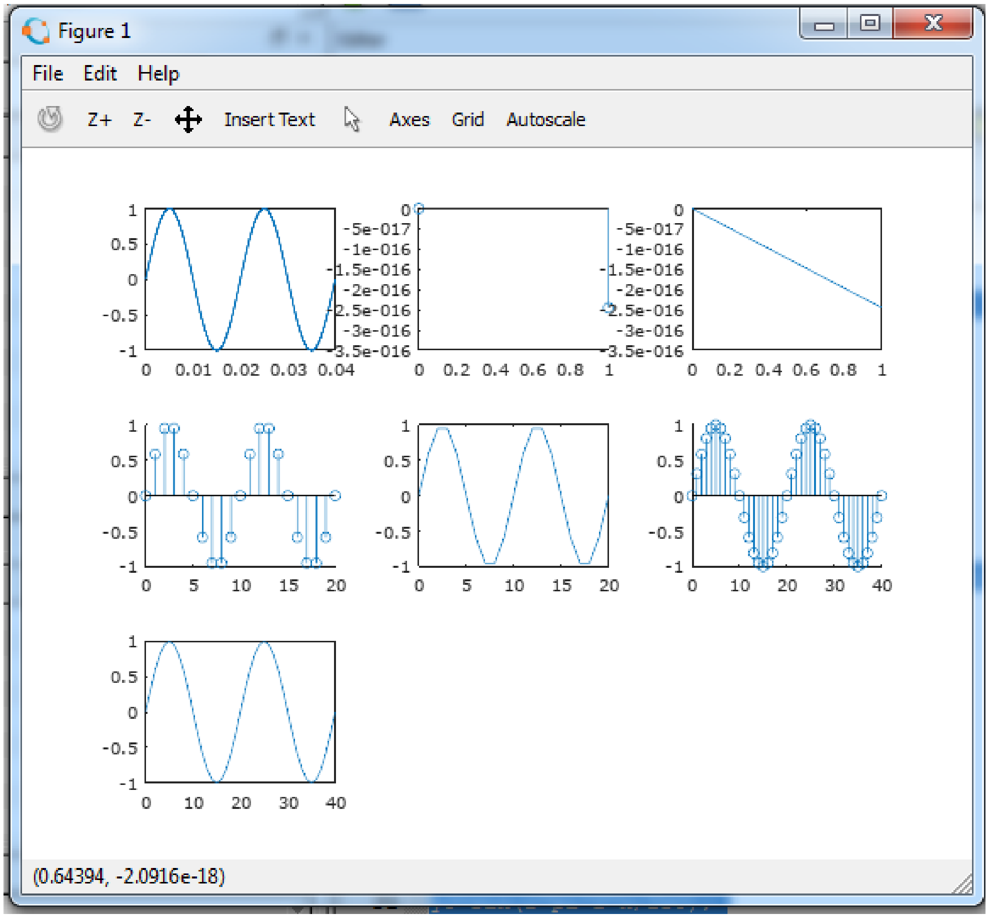


Fig. 2.1: waveforms for sampling frequency = 50Hz, 500Hz and 1000 Hz.

Result: Sampling theorem is verified for the sampling frequency of ___________ Hz

Waveform is ______________ for the sampling frequency of ___________ Hz

Waveform is ______________ for the sampling frequency of ___________ Hz

Experiment no. 3

To perform basic operations on given sequences- Signal folding, evaluation of even and odd components

Objective:

- To learn the basic operation of signal folding on the given sequence
- Verify the computed values of even and odd components with the simulated values.

Theory: In digital signal processing, signal folding occurs when the sampling frequency is insufficient to accurately represent the signal, causing higher frequencies to fold back into the spectrum. Even and odd components of signals describe their symmetry properties, which can be useful in signal analysis and processing.

Example:

Let x(t) = (1, 2, 3, 4, 5)

x(-t) = (5,4,3,2,1)

Even Component, $x_e(t) = \frac{1}{2} * [(x(t) + x(-t)])$

Odd Component, $x_o(t) = \frac{1}{2} * [(x(t) - x(-t)])$

Therefore, $x_e(t)$ = 0.5* (6,6,6,6,6)

= (3,3,3,3,3)

$x_o(t)$ = 0.5 * (-4, -2, 0, 2, 4)

= (-2, -1, 0, 1, 2)

Program:

```
clc;
clear all;

% Signal Folding and Evaluation of Even and Odd Components

% Define the sequence
x = [1 2 3 4 5];
t=0:4;

% Signal Folding
x_flipped = fliplr(x); % Flip the sequence
```

```
% Even and Odd Components
x_even = 0.5 * (x + x_flipped); % Even component
x_odd = 0.5 * (x - x_flipped); % Odd component

% Display Results
figure();
disp('Original Sequence:');
disp(x);
subplot(2,2,1)
stem(t,x);
xlabel('time');
ylabel('amplitude');
title('Original Sequence:');

disp('Flipped Sequence:');
disp(x_flipped);
subplot(2,2,2)
stem(t,x_flipped);
xlabel('time');
ylabel('amplitude');
title('Flipped Sequence:');

disp('Even Component:');
disp(x_even);
subplot(2,2,3)
stem(t,x_even)
xlabel('time');
ylabel('amplitude');
title('Even Component');

disp('Odd Component:');
disp(x_odd);
subplot(2,2,4)
stem(t,x_odd);
xlabel('time');
ylabel('amplitude');
title('Odd Component');
```

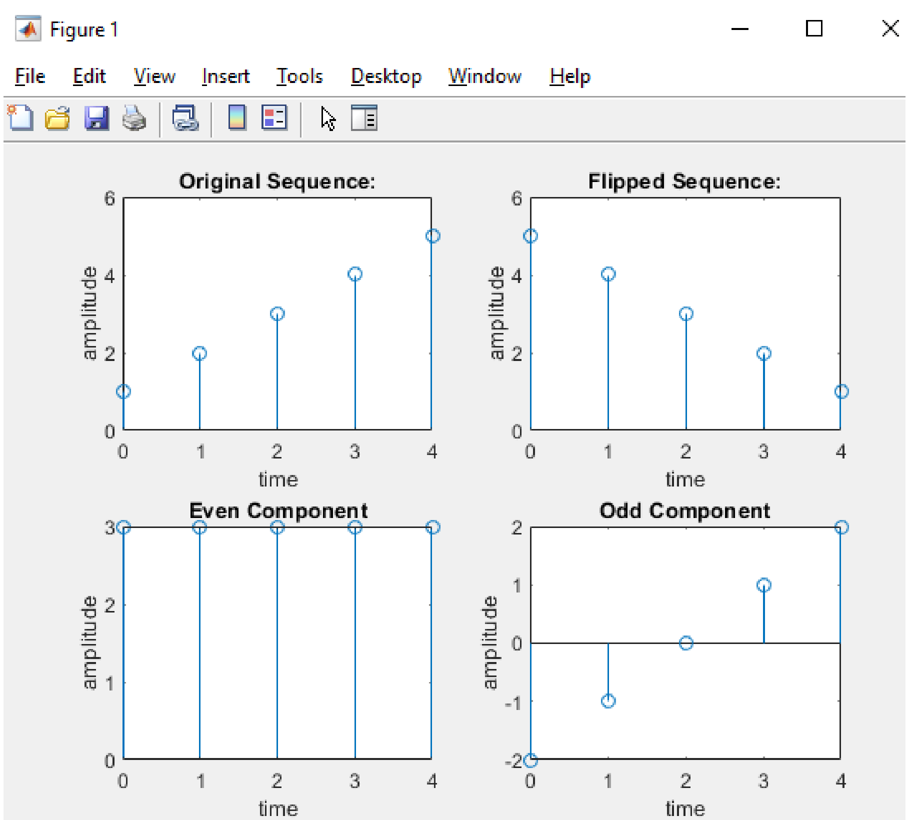
Output Windows

Original Sequence:
1 2 3 4 5

Flipped Sequence:
5 4 3 2 1

Even Component:
3 3 3 3 3

Odd Component:
-2 -1 0 1 2



Result: Coefficients of Even and Odd Component values are ____________

Simulated and calculated values are ___________

Experiment no. 4

Evaluation of impulse response of a system

Objectives:

- To arrive at TF of a system.
- To understand the response of a system for an impulse input.

Theory: The impulse response of a system is the output of the system when the input signal is an impulse function.

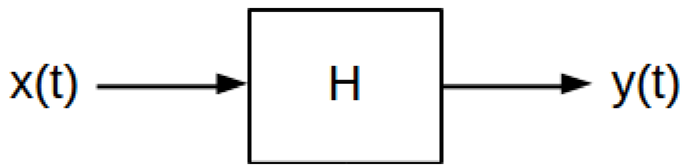


Fig. 4.1: generalized system with input and output

The impulse response and frequency response are two attributes that are useful for characterizing linear time-invariant (LTI) systems.

Example:

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

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

Taking Z transform on both sides,

$$Y(Z) - Z^{-1}Y(Z) - Z^{-2}Y(Z) = X(Z) + 0.5Z^{-1}X(Z) + 0.85Z^{-2}X(Z)$$

$$Y(Z)[1 - Z^{-1} - Z^{-2}] = X(Z)[1 + 0.5Z^{-1} + 0.85Z^{-2}]$$

$$\text{T F, } H(Z) = \frac{Y(Z)}{X(Z)}$$

$$H(Z) = \frac{Y(Z}{X(Z)} = \frac{[1 + 0.5Z^{-1} + 0.85Z^{-2}]}{[1 - Z^{-1} - Z^{-2}]}$$

By power series method

$H(Z) = 1 + 1.5\ Z^{-1} + 3.35\ Z^{-2} + 4.85\ Z^{-3}$

h(n) = [1 1.5 3.35 4.85]

Program:

```
clc;
   clear all;
        b=input ('enter the coefficients of x(n), x(n-1)-----');
        a=input ('enter the coefficients of y(n), y(n-1) ----');
        N=input ('enter the number of samples of impulse response ');
       [h, t]=impz(b,a,N);
disp(h);
stem(t, h);
```

Output Windows

```
enter the coefficients of x(n),x(n-1)-----[1 0.5 0.85]
enter the coefficients of y(n),y(n-1)----[1 -1 -1]
enter the number of samples of impulse response 4
   1.0000   1.5000   3.3500   4.8500
```

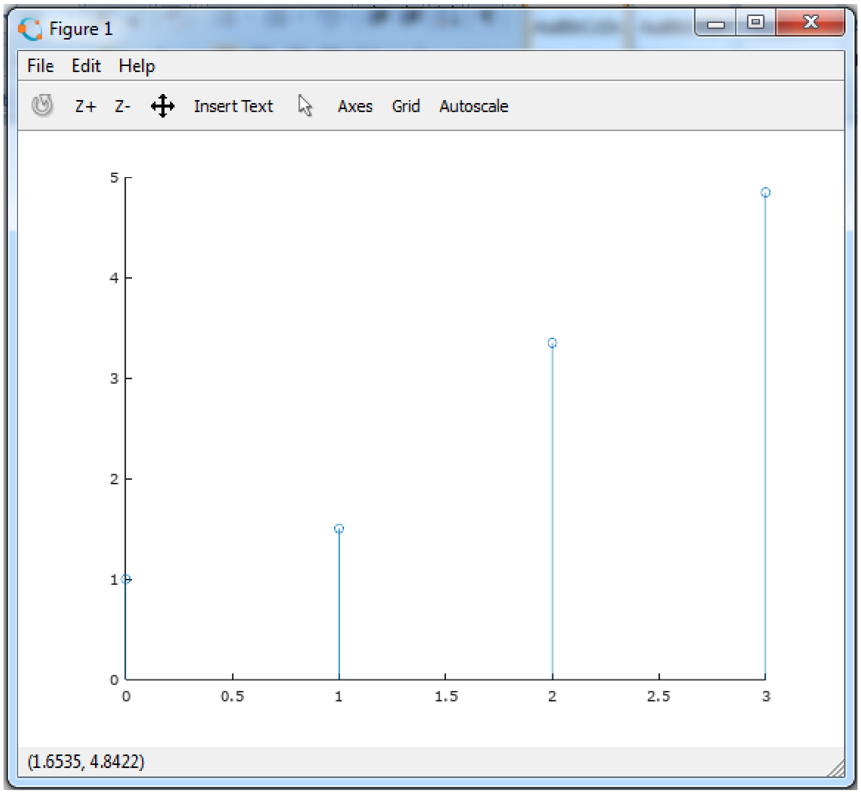


Fig. 4.2: output waveform of the impulse response of the given system

Result: Coefficients of the given system are ____________

Simulated and calculated values are ___________

Experiment 5

Solution of a given difference equation

Objectives:

- To find solution for difference equation.
- To verify the solution of difference equation using simulation tool.

Theory: An LTI discrete system can also be described by a linear constant coefficient difference equation of the form.

$$\sum_{K=0}^{N} a_k y(n-k) = \sum_{m=0}^{M} b_m x(n-m) \quad \forall n$$

Where x(n) is input and y(n) is output

Example:

$$\text{let } y(n) - \tfrac{3}{2} y(n-1) + \tfrac{1}{2} y(n-2) = x(n)$$

$$\text{Given } x(n) = \left(\tfrac{1}{4}\right)^n u(n)$$

Initial conditions are y(-1)=4 y(-2)=10

$$x(n) = \left(\tfrac{1}{4}\right)^n u(n)$$

For n=0 x(0)=1
n=1 x(1)=0.25
n=2 x(2)=0.0625
n=3 x(3)=0.0156
n=4 x(4)=0.0039

$$y(n) - \tfrac{3}{2} y(n-1) + \tfrac{1}{2} y(n-2) = x(n)$$

$$y(n) = \tfrac{3}{2} y(n-1) - \tfrac{1}{2} y(n-2) = x(n)$$

For n=0 y(0)=1.5*y(0-1)-0.5*y(0-2)+x(0)=1.5*4-0.5*10+1=2
n=1 y(1)=1.5*y(0)-0.5*y(-1)+x(1)=1.25
n=2 y(2)=0.9375
n=3 y(3)=0.7969
n=4 y(4)=0.7365

Program:

```
clc;
clear;
b=input('enter the co-effencient of x(n):');
a=input('enter the co-effencient of y(n):');
y=input('enter the initial conditions:');
```

```
n1=input('enter the value of n:');
z=filtic(b,a,y);
n=0:n1-1;
x=(1/4).^n;
yo=filter(b,a,x,z);
disp(yo);
stem(n,yo);
```

```
enter the co-effencient of x(n):[1]
enter the co-effencient of y(n):[1 -3/2 1/2]
enter the initial conditions:[4 10]
enter the value of n:5
  2.00000   1.25000   0.93750   0.79688   0.73047
```

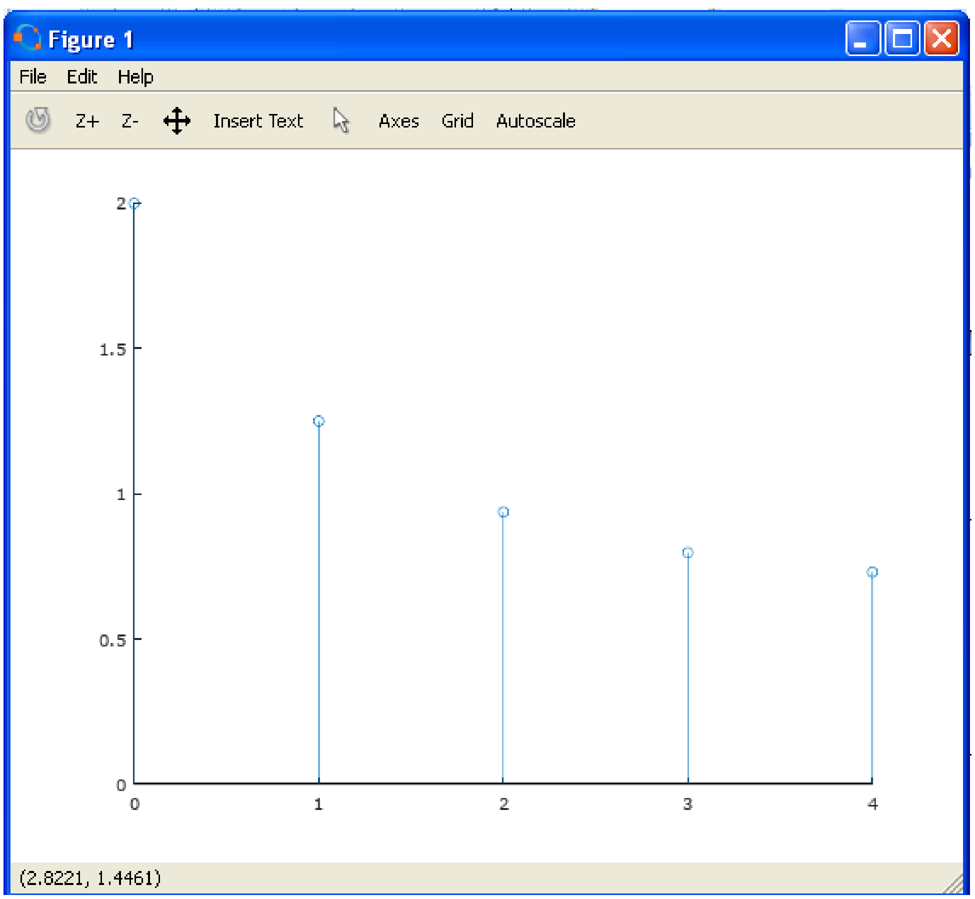


Fig. 7.1: Waveform for the solution of the given difference equation

Result: solution of difference equation is evaluated.

Experiment 6

Evaluation of linear convolution and circular convolution of given sequences

6(a) To Perform Linear Convolution of given Sequences

Objectives:

- To perform linear convolution of the given two sequences
- To verify the linear convolution using simulation tool.

Theory: convolution relates the input signal and impulse response of the system to the output of the system. Convolution is a mathematical operation which is used to express the input-output relationship of an LTI system.

Linear convolution is the basic operation to calculate the output for any linear time invariant system given its input and its impulse response. Linear convolution is given by

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

Example: x(n) = (1 2 3 5 7 8 9) and h(n) = (1 2 3)

Solution:

	1	2	3	5	7	8	9
1	1	2	3	5	7	8	9
2	2	4	6	10	14	16	18
3	3	6	9	15	21	24	27

y(n) = (1 4 10 17 26 37 46 42 27)

Program

```
clc;
  clear;
      x=input('enter the first sequence x(n) ');
        subplot(3,1,1);
           stem(x);
             title('plot of the first sequence');
                h = input('enter 2nd sequence h(n) ');
                  subplot(3,1,2);
                    stem(h);
```

```
title('plot of 2nd sequence');
y=conv2(x,h);
disp('output of linear convolution is');
disp(y);
xlabel('time index n');
ylabel('amplitude');
subplot(3,1,3);
stem(y);
title('linear conv of sequence');
```

Output Windows

Enter the first sequence x(n) [1 2 3 5 7 8 9]
Enter 2nd sequence h(n) [1 2 3]
Output of linear convolution is [1 4 10 17 26 37 46 42 27]

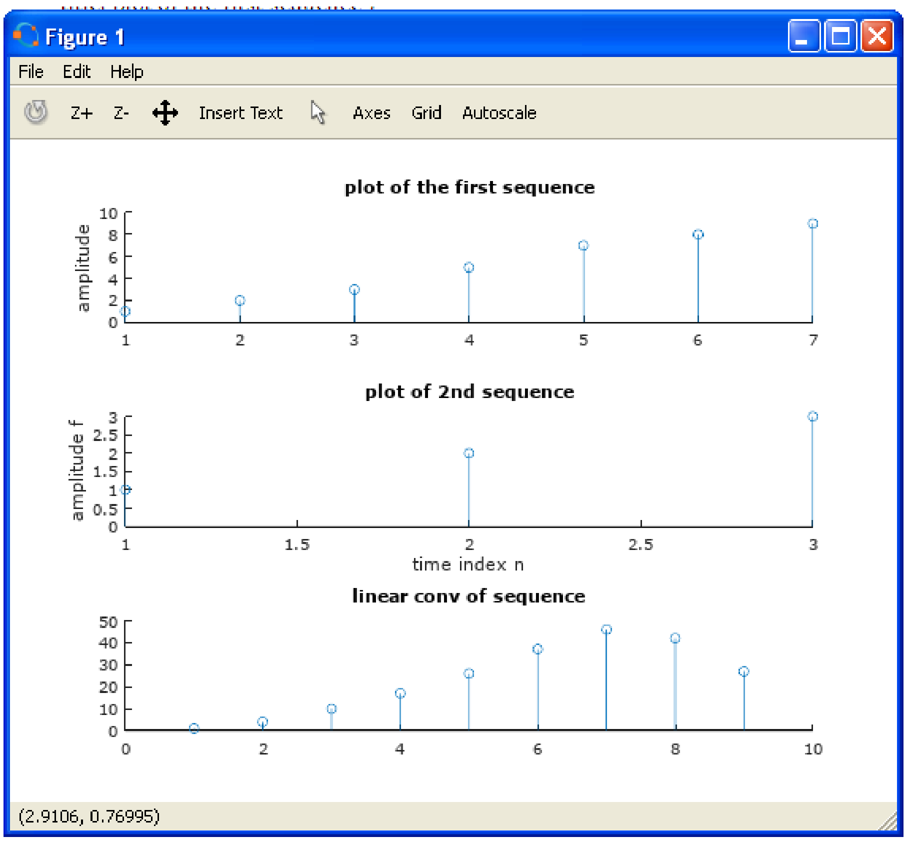


Fig. 6.1: input and output waveform of the given sequence.

Result:

Linear convolution for the given sequences are _______

Simulated and calculated values are found to be ___________

6 (b) Circular Convolution of the Given two Sequences

Objectives:

To perform circular convolution of given sequences using
(a) The convolution summation formula.
(b) The matrix method.

i. The convolution summation formula:

Program:

```
clc;
  clear;
    x1=input('enter x1(n):');
      n1=length(x1);
        x2=input('enter x2(n):');
          n2=length(x2);
            N=max(n1,n2);
              T=1:N;
               x1=[x1 zeros(1,N-n1)];
                 x2=[x2 zeros(1,N-n2)];
                   y=zeros(1,N);
                    for m=1:N
                        for n=1:N
                            i=m-n+1;
                             if (i<=0)
                              i=N+i;
                             end
                          y(m)=y(m)+x1(n)*x2(i);
                        end
                    end
              disp(y);
        stem(y);
```

Output Windows

enter x1(n):[0 1 2 3]
enter x2(n):[2 1 2 1]
circular convolution of the given sequences is
8 10 8 10

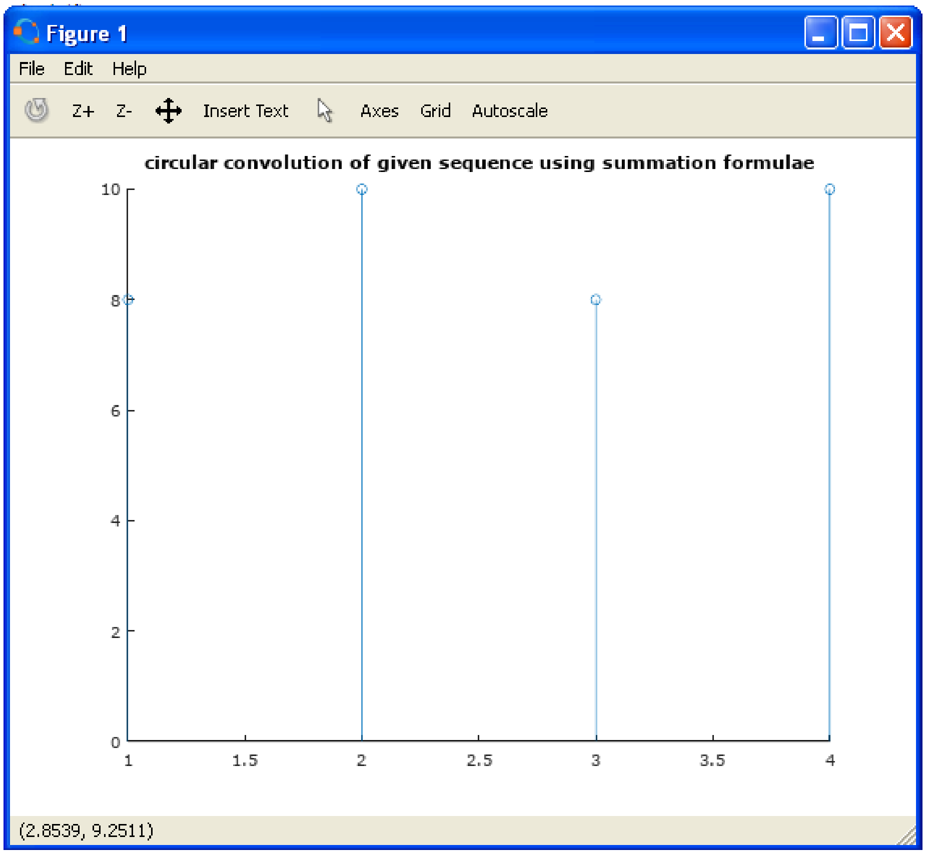


Fig. 6.2: Output waveform for the given sequence using convolution summation formula.

Result: Circular convolution for the given sequences and for the method________ is _______

Simulated and calculated values are found to be ___________

ii. To perform circular convolution of given sequences using matrix method

Program:

```
clc;
  clear;
    a=input('Enter the first sequence x1(n):');
      b=input('Enter the first sequence x2(n):');
        c=circshift(a,1);
          d=circshift(a,2);
            e=circshift(a,3);
             m=[a c d e];
               y=m*b;
             disp(m);
          disp('Circular Convolution of x1(n) & x2(n) is:');
      disp(y);
    n=0:3;
  stem(n,y);
title('circulation convolution using matrix method');
```

Output Windows

```
Enter the first sequence x1(n):[2;1;2;1]
Enter the first sequence x2(n):[1;2;3;4]
  2   1   2   1
  1   2   1   2
  2   1   2   1
  1   2   1   2
Circular Convolution of x1(n) & x2(n) is:
  14
  16
  14
  16
```

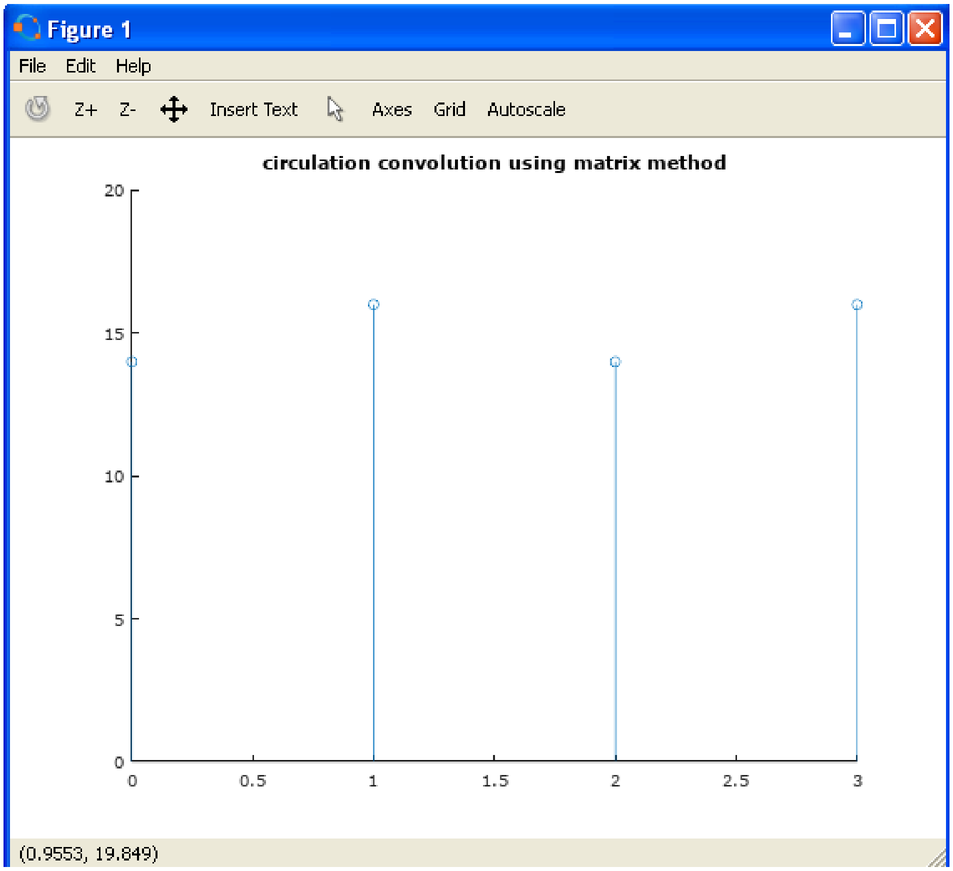


Fig. 6.3: Output waveform for the given sequence using matrix method

Result: Circular convolution for the given sequences and for the method________ is _______

Simulated and calculated values are found to be ___________

Experiment 7:

Computation of N- point DFT and IDFT of a given sequence by use of (a) Defining equation (b) FFT method.

7(a) Computation of N-point DFT of a given sequence by use of Defining equation

Objectives:

- To perform N-point DFT computation and plot the magnitude and phase spectrum.

Theory: DFT is the process of conversation of time domain continuous signal into frequency domain discrete signal. DFT of an N-point sequence is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}$$

Example: **Find the DFT of 4-point sequence x(n)= (0,1,2,3)**

Solution:

$$X(k) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ -2+j2 \\ -2 \\ -2-j2 \end{bmatrix}$$

X(k) = (6,-2+j2,-2,-2-j2)

Magnitude spectrum: $|X(k)| = (6, 2.8284, 2, 2.8284)$

Phase angle spectrum: $\angle X(k) = (0, 2.356, -3.14, -2.356)$ rad

Program:

```
clc;
 close all;
  clear all;
   xn = input('enter the sequence x(n) =');
     xn=xn';
       N=length(xn);
        n=0:N-1;
         for k=0:N-1
            Xk(k+1)=exp(-j*2*pi*k*n/N)*xn;
         end
```

```
disp('DFT of the given sequence is:');
disp(Xk);
k = 0:N-1;
disp('magnitude of the sequence is:')
disp(abs(Xk))
subplot(2,1,1)
stem(k,abs(Xk));
xlabel('k');
ylabel('|X(k)|');
title('Magnitude spectrum');
subplot(2,1,2);
stem(k,angle(Xk));
disp('phase angle of the sequence in radians is:');
disp(angle(Xk));
xlabel('k');
ylabel('<X(k)');
title('Phase spectrum');
```

Output Window:

enter the sequence x(n) =[0 1 2 3]

DFT of the given sequence is:

6.0000 + 0.0000i -2.0000 + 2.0000i -2.0000 - 0.0000i -2.0000 - 2.0000i

magnitude of the sequence is:

6.0000   2.8284   2.0000   2.8284

phase angle of the sequence in radians is:

0.00000   2.35619 -3.14159 -2.35619

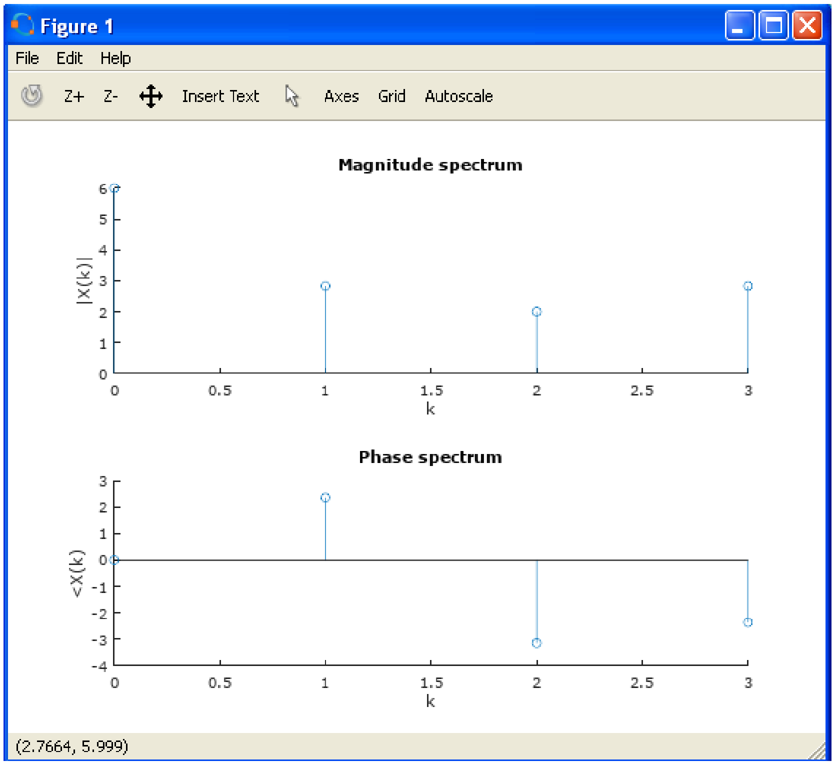


Fig. 5.1: Magnitude and phase angle spectrum.

Result:

a) _____ point DFT for the given sequence is ____

b) The Simulated values of magnitude and phase spectrum is __________

c) Simulated and calculated values are found to be ___________

Experiment 7(b):

Computation of N- point DFT and IDFT of a given sequence by FFT Method

Objectives:

- To find DFT & IDFT using FFT.
- Evaluating the DFT and IDFT using simulation tool.

Theory: as discussed earlier DFT is the process of conversation of time domain continuous signal into frequency domain discrete signal. DFT of an N-point sequence is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}$$

IDFT is the process of conversation of frequency domain discrete signal into time domain continuous signal. IDFT of an N-point sequence is given by

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi nk}{N}}$$

i. **DFT of a N-point sequence**

Example: find the DFT of x(n)=(1,2,3,4,4,3,2,1)

```
N=input('How many point DFT do you want?');
x =input('Enter the first sequence=');
n = length(x)
x = [x zeros(1,N-n)]
X = fft(x,N)
X_mag = abs(X)
subplot(2,1,1)
stem(X_mag)
title('Magnitude plot')
X_ang = angle(X)
subplot(2,1,2)
stem(X_ang)
title('Angle plot')
```

```
How many point DFT do you want?8
Enter the first sequence=[1 2 3 4 4 3 2 1]
n =  8
x =
   1   2   3   4   4   3   2   1
X =
 Columns 1 through 3:
   20.00000 +  0.00000i   -5.82843 -  2.41421i    0.00000 +  0.00000i
 Columns 4 through 6:
   -0.17157 -  0.41421i    0.00000 +  0.00000i   -0.17157 +  0.41421i
```

```
 Columns 7 and 8:
   0.00000 -  0.00000i   -5.82843 +  2.41421i
X_mag =   20.00000    6.30864    0.00000    0.44834    0.00000    0.44834    0.00000    6.30864
```

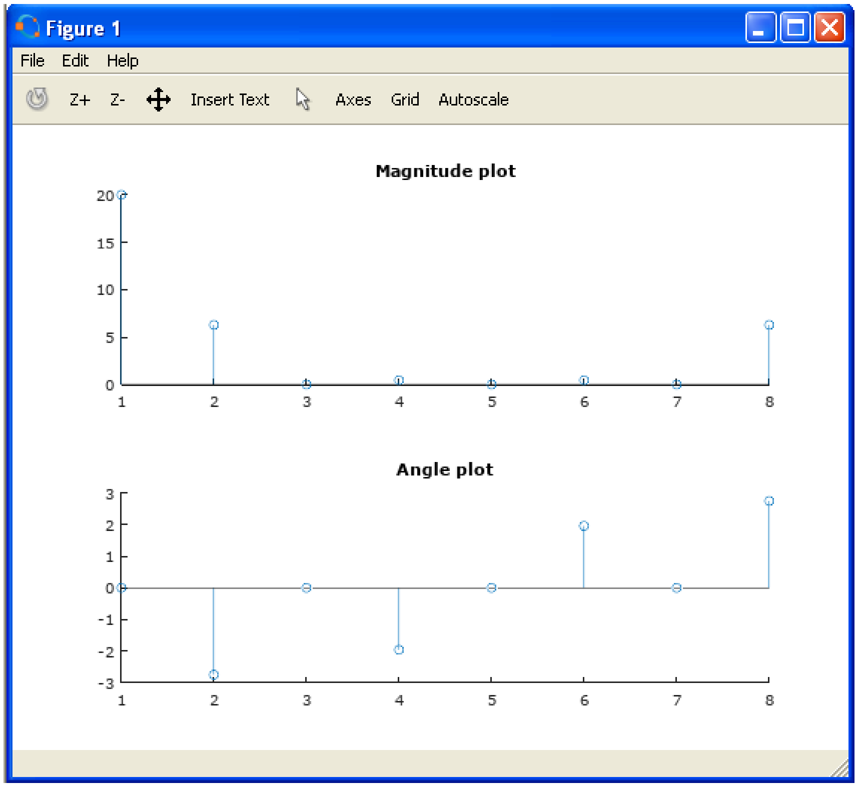


Fig. 8.1: Magnitude and phase angle plot for 8-point sequence.

ii. Calculation of IDFT by FFT

Program:

```
clc;
clear all;
N=input('Enter the value of N: ');
X =input('Enter the first sequence=');
x = ifft(X,N);
disp('IDFT of the given sequence is');
disp(x);
stem(x);
title('IDFT of given sequence');
```

Enter the value of N: 4
Enter the first sequence=[6 -2+2i -2 -2-2i]
IDFT of the given sequence is
0 1 2 3

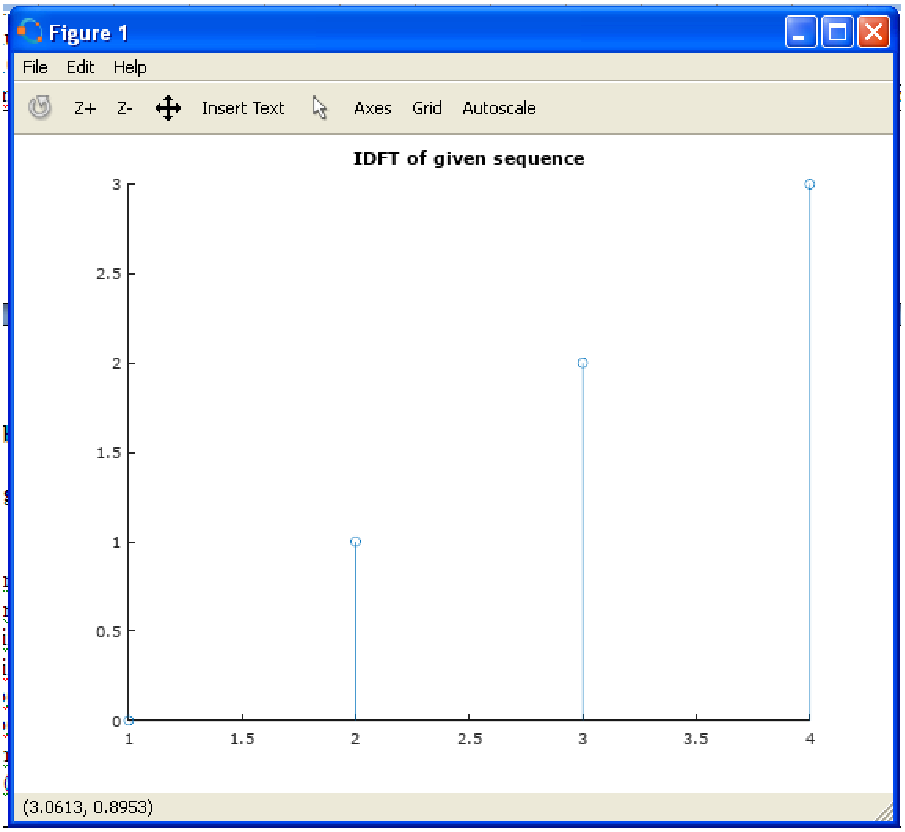


Fig. 8.2: output waveform of the IDFT for the given sequence

Result: DFT of the N-point sequence is determined.

Experiment 8:

Evaluation of circular convolution of two sequences using DFT and IDFT approach

Objective:

- To verify the Linear convolution by DFT & IDFT method using simulation tool
- To verify the circular convolution by DFT & IDFT method using simulation tool

Theory: DFT and IDFT method is also known as stockham's method used to find the Circular convolution of the given sequences.

$y(n) = x(n) \otimes_N h(n)$

Applying DFT for x(n) and h(n)

X(k)=DFT(x(n)) and H(k)=DFT(h(n))

Y(K) = X(K)*H(K)

Apply IDFT

y(n) =IDFT(Y(K))

Program: Circular Convolution using DFT and IDFT

```
clc;
  clear all;
    close all;
      xn = input('enter the first sequence x(n) = ');
          hn = input('enter the second sequence h(n) = ');
              N = max(length(xn),length(hn));
                Xk = fft(xn,N);
                    Hk = fft(hn,N);
                        Yk = Xk.*Hk;
                           yn = ifft(Yk,N);
                             disp('Circular convolution of x(n) and h(n) =');
                               disp(yn);
                                  subplot(2,2,1);
```

```
stem(xn);
xlabel('n');
ylabel('x(n)');
title('plot of x(n)');
subplot(2,2,2);
stem(hn);
xlabel('n');
ylabel('h(n)');
title('plot of h(n)');
subplot(2,2,3);
stem(yn);
xlabel('n');
ylabel('y(n)');
title('Circular convolution Output');
```

Output Screen

Enter the first sequence x(n) = [1 2 3 4]
Enter the second sequence h(n) = [4 5 6 1]
Circular convolution of x(n) and h(n) =   44   40   32   44

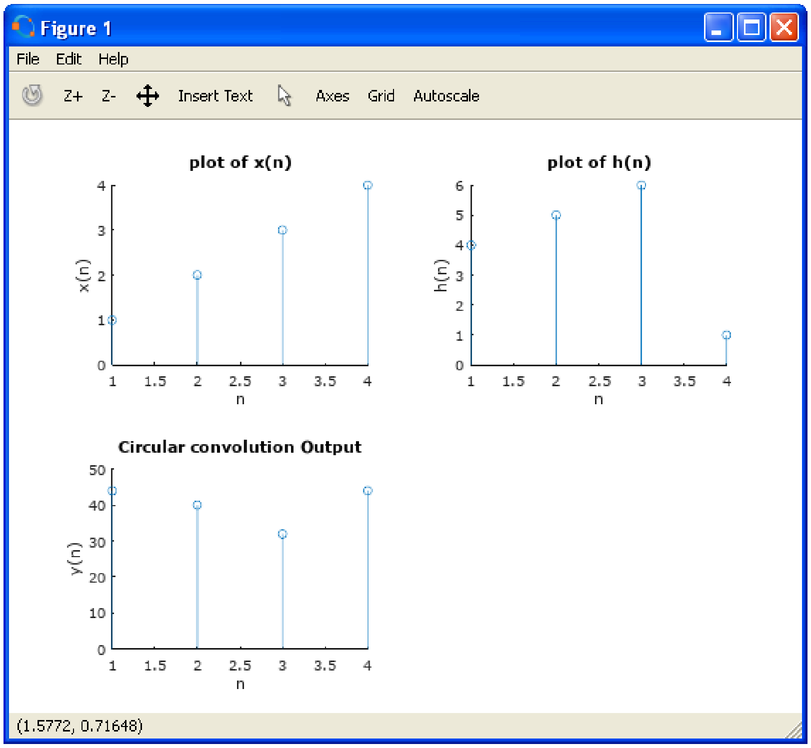


Fig. 8.1: Input and output waveforms of the given sequence using DFT and IDFT method.

Result:

a) _____ point DFT & IDFT for the given sequence is ____

b) Simulated and calculated values are found to be ___________

Experiment 9:

Design and implementation of IIR filters to meet given specification

Objective:

- To Design and implement IIR filter for given specification
- To verify the IIR filter design using simulation tool.

Theory: filer enhances a wanted signal and eliminates to unwanted signals. IIR filters are digital filters with infinite impulse response. They have the feedback and are known as recursive digital filters therefore. IIR filters have much better frequency response than FIR filters of the same order. There are mainly two types of IIR filter, Butterworth and Chebyshev filter.

The frequency response of the various filters is shown below.

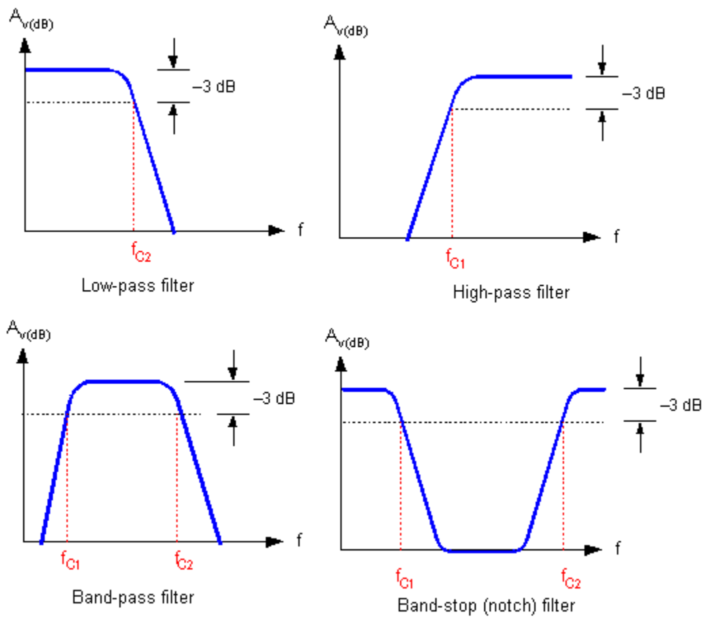


Fig. 9.1: Frequency response curve

Design procedure:

Finding the value of N:

$$N = \frac{\log\left[\dfrac{10^{\frac{-K_P}{10}} - 1}{10^{\frac{-K_S}{10}} - 1}\right]}{2\log\left[\dfrac{\Omega_P}{\Omega_S}\right]}$$

Finding the poles of the transfer function

$S_K = 1\angle\theta_K$ Where k=0,1,2,3…2N-1

$$\theta_K = \frac{\pi k}{N} + \frac{\pi}{2N} + \frac{\pi}{2}$$

Finding Transfer function:

$$H_N(s) = \frac{1}{\sum_{\text{LHP only}} (s - s_k)}$$

Example: Design a low pass Butterworth filter to satisfy the following specifications;
Pass band cutoff = 500 Hz; Pass band ripple = 3.01 db;
Stop band cutoff = 750 Hz; Stop band ripple = 15 db;

Program:

```
clc; % clear screen
clear all; % clear screen
close all; % close all figure windows
fp = input('Enter the Pass band frequency in Hz = ' ); % input specifications
fs = input('Enter the Stop band frequency in Hz = ');
Fs = input('Enter the Sampling frequency in Hz = ');
Ap = input('Enter the Pass band ripple in db:');
As = input('Enter theStop band ripple in db:');
wp=2*fp/Fs; % Analog frequency
ws=2*fs/Fs;
Up = 2*tan(wp/2);% Prewrapped frequency
Us = 2*tan(ws/2);
[n,wn]= buttord (Up,Us,Ap,As,'s'); %Calculate order and cutoff freq
disp('order of the filter N =');
disp(n);
 [num, den] = butter(n,wn,'s'); % analog filter transfer
[b,a] = bilinear(num, den,1); % conversion of analog filter to digital filter
Freqz(b,a,512,Fs); % frequency response of the filter
Printsys(b,a,'z'); % print the H(z) equation obtained on screen
```

Output Window

```
Enter the Pass band frequency in Hz = 500
Enter the Stop band frequency in Hz = 750
Enter the Sampling frequency in Hz = 2000
Enter the Pass band ripple in db:3.01
Enter theStop band ripple in db:15
order of the filter N =
    4
num/den =
    0.0022316 z^4 + 0.0089263 z^3 + 0.013389 z^2 + 0.0089263 z + 0.0022316
   ----------------------------------------------------------------------
           z^4 - 2.6932 z^3 + 2.8684 z^2 - 1.4042 z + 0.26461
```

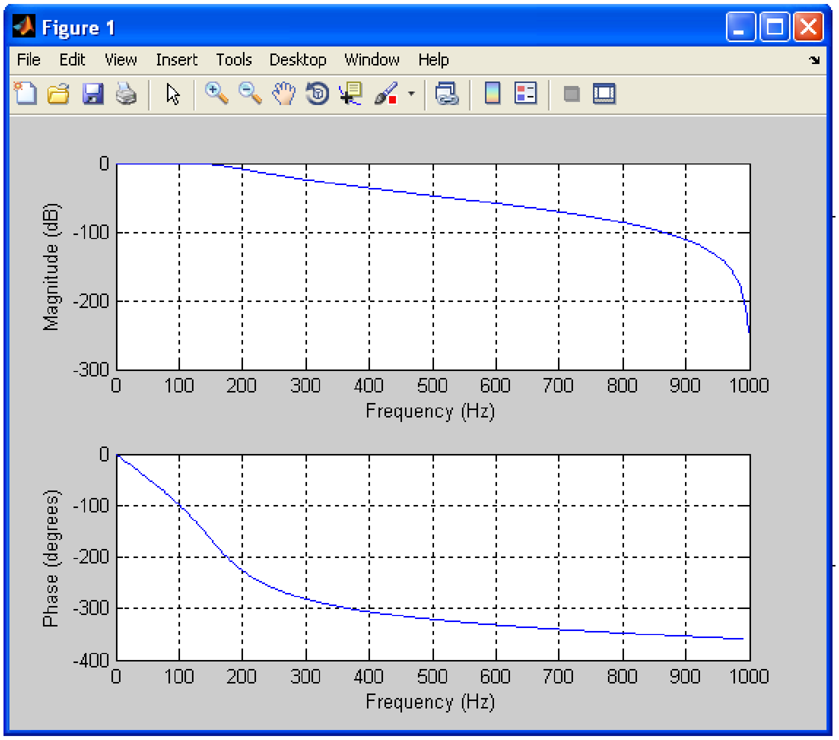


Fig. 9.2: Magnitude and phase angle curve of low pass filter

Example: Design a high pass Butterworth filter to satisfy the following specifications;
Pass band cutoff = 32 Hz; Pass band ripple = 2 db;
Stop band cutoff = 16 Hz; Stop band ripple = 20 db;

```
clc;
clear all;
close all;
rp=input('Enter the pass band ripple:rp=');
rs=input('Enter the stop band ripple:rs=');
fp=input('Enter the pass band freq:wp=');
fs=input('Enter the stop band freq:ws=');
Fs=input('Enter the sampling freq fs=');
wp=2*fp/Fs;
ws=2*fs/Fs;
Up = 2*tan(wp/2);% Prewrapped frequency
Us = 2*tan(ws/2);
[N,wn]=buttord(Up,Us,rp,rs,'s');
Disp(' Order of the filter is: ');
disp(N);
disp('freq resp of iir high pass filter is:');
[b,a]=butter(N,wn,'high','s');
disp('Normalized cut off frequency = ');
disp(wn);
[num, den] = butter(N,wn,'high','s'); % analog filter transfer
[b,a] = bilinear(num, den,1); % conversion of analog filter to digital filter
```

```
Freqz(b,a,512,Fs); % frequency response of the filter
Printsys(b,a,'z'); % print the H(z) equation obtained on screen.
```

```
Enter the sepecifications of iir filter
Enter the pass band ripple:rp=2
Enter the stop band ripple:rs=20
Enter the pass band freq:wp=32
Enter the stop band freq:ws=16
Enter the sampling freq fs=2000
Order of the filter is:
 4
num/den =
    0.96355 z^4 - 3.8542 z^3 + 5.7813 z^2 - 3.8542 z + 0.96355
   ----------------------------------------------------------
        z^4 - 3.9257 z^3 + 5.78 z^2 - 3.7827 z + 0.92843
```

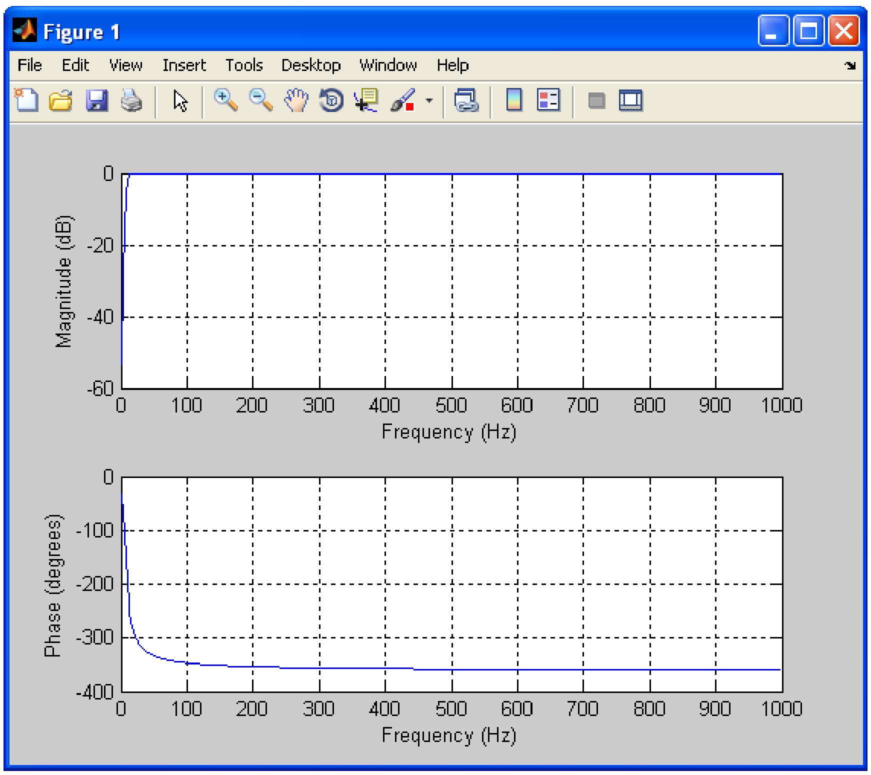


Fig. 9.3: Magnitude and phase angle curve of high pass filter

Result: IIR filter was designed was verified using simulation tool.

Experiment 10:

Design and Implementation of FIR Filters Using Different Window Functions

Objective: To Design and implement FIR filter using various window techniques.

Theory:FIR Filter Design

Digital filters with finite-duration impulse response (all-zero, or FIR filters) have both advantages and disadvantages compared to infinite-duration impulse response (IIR) filters.
FIR filters have the following primary advantages:

- They can have exactly linear phase.
- They are always stable.
- The design methods are generally linear.
- They can be realized efficiently in hardware.
- The filter startup transients have finite duration.

The primary disadvantage of FIR filters is that they often require a much higher filter order than IIR filters to achieve a given level of performance. Correspondingly, the delay of these filters is often much greater than for an equal performance IIR filters.

Problem:

Design FIR lowpass filter to meet the following specifications:
Passband attenuation = 0.01
Stopband attenuation = 0.02
Passband frequency = 1000 Hz
Stopband frequency = 1500 Hz
Sampling frequency = 8000 Hz

Program:

```
clc;
clear all;
Rp=input('Enter the passband attenuation in dB: ');
Rs=input('Enter the stopband attenuation in dB: ');
fp=input('Enter the passband frequency in Hz: ');
fs=input('Enter the stopband frequency in Hz: ');
Fs=input('Enter sampling frequency in Hz: ');
Wp=2*pi*fp/Fs;
Ws=2*pi*fs/Fs;
%Rectangular Window
if Rs<=21
M = ceil(4*pi/(Ws-Wp))+1
y = rectwin(M);
figure, subplot(2,1,1),stem(y);
title('Rectangular Window');
end
```

```
%Bartlett Window
if Rs<=25 &Rs>21
M =ceil(8*pi/(Ws-Wp)) +1
y=bartlett(M);
figure, subplot(2,1,1),stem(y);
title('Bartlett Window');
end
%Hanning Window
if Rs<=44 &Rs>25
M =ceil(8*pi/(Ws-Wp)) +1
y=hanning(M);
figure, subplot(2,1,1),stem(y);
title('Hanning Window');
end
%Hamming Window
if Rs<=53 &Rs>44
M = ceil(8*pi/(Ws-Wp)) +1
y=hamming(M);
figure, subplot(2,1,1),stem(y);
title('Hamming Window');
end
% Blackman Window
if Rs<=74 &Rs>53
M = ceil(12*pi/(Ws-Wp))+1
y=blackman(M);
figure, subplot(2,1,1),stem(y);
title('Blackman Window');
end
wc = (Ws +Wp)/2
%Ideal lowpass filter
alpha = (M-1)/2
n = [0 : 1 : (M-1)];
m = n - alpha + eps;
hd = sin(wc*m)./(pi*m);
h = hd.*y';
subplot(2,1,2), stem(h);
title('Impulse response of FIR filter');
%w = logspace(-1,1);
figure, freqz(h)
title('Frequency response of FIR lowpass filter')
```

Output Window:

Enter the passband attenuation in dB: 0.01
Enter the stopband attenuation in dB: 0.02
Enter the passband frequency in Hz: 1000
Enter the stopband frequency in Hz: 1500
Enter sampling frequency in Hz: 8000

```
M =  33
wc =  0.98175
alpha =  16
```

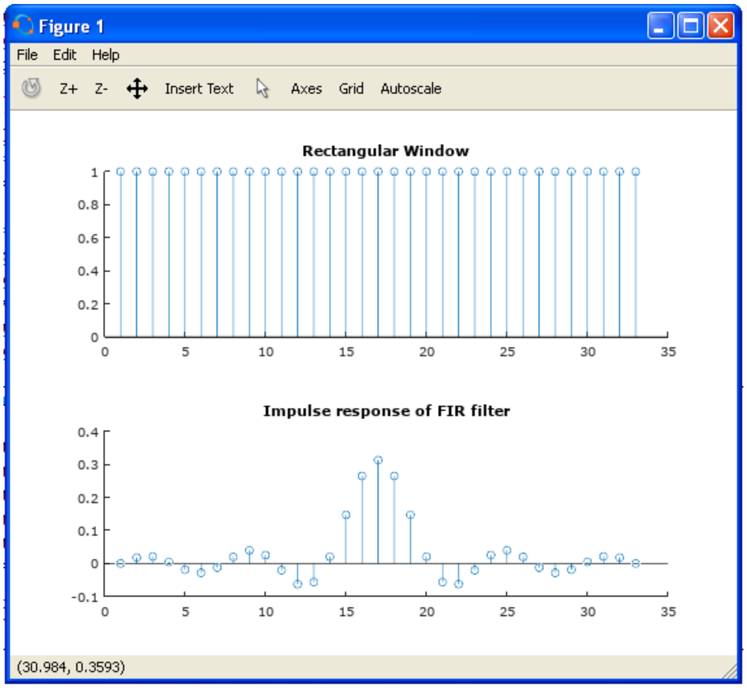


Fig. 10.1: Rectangular window and Impulse response curve of FIR filter

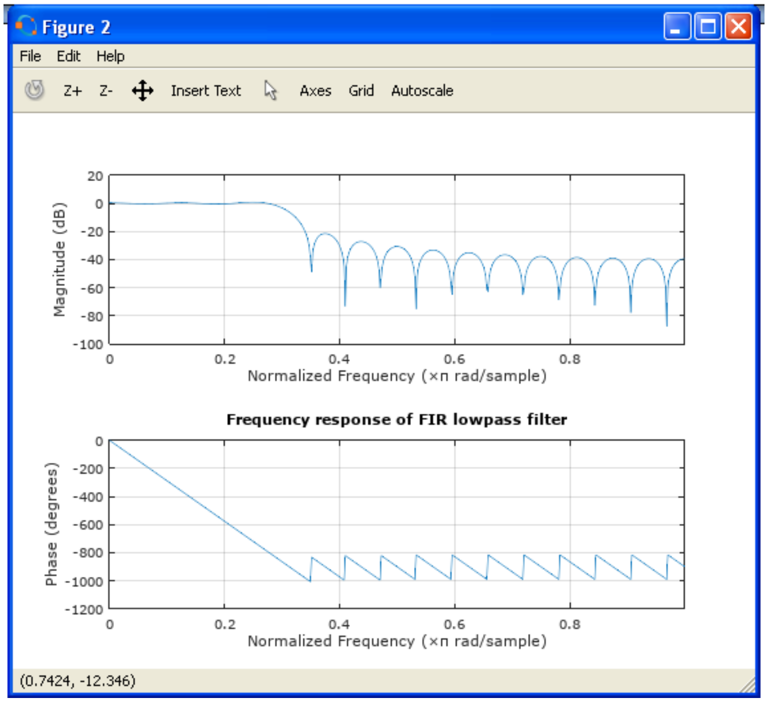


Fig. 10.2: Magnitude and phase angle curve of lowpass FIR filter

Problem:
Design a bandpass filter with the given specifications:
Lower stopband edge = 0.2π
Lower Passband edge = 0.35π
Upper passband edge = 0.65π
Upper stopband edge = 0.8π
Passband attenuation = 1db
Stopband attenuation = 60db

Program:

```
clc;
clear all;
Rp=input('Enter the passband attenuation in dB: ');
Rs=input('Enter the stopband attenuation in dB: ');
Wp=input('Enter the passband frequency in pi rad: ');
Ws=input('Enter the stopband frequency in pi rad: ');
tr_width = min((Wp(1)-Ws(1)),(Ws(2)-Wp(2)))
%Rectangular Window
if Rs<=21
M = ceil(4*pi/tr_width)+1
y = rectwin(M);
figure, subplot(2,1,1),stem(y);
title('Rectangular Window');
end
%Bartlett Window
if Rs<=25 &Rs>21
M =ceil(8*pi/tr_width) +1
y=bartlett(M);
figure, subplot(2,1,1),stem(y);
title('Bartlett Window');
end
%Hanning Window
if Rs<=44 &Rs>25
M =ceil(8*pi/tr_width) +1
y=hanning(M);
figure, subplot(2,1,1),stem(y);
title('Hanning Window');
end
%Hamming Window
if Rs<=53 &Rs>44
M = ceil(8*pi/tr_width) +1
y=hamming(M);
figure, subplot(2,1,1),stem(y);
title('Hamming Window');
end
```

```
% Blackman Window
if Rs<=74 &Rs>53
M = ceil(12*pi/tr_width)+1
y=blackman(M);
figure, subplot(2,1,1),stem(y);
title('Blackman Window');
end
wc1 = (Ws(1) +Wp(1))/2
wc2 = (Ws(2)+Wp(2))/2
%Ideal bandpass filter
alpha = (M-1)/2
n = [0 : 1 : (M-1)];
m = n - alpha + eps;
hd = (sin(wc2*m)./(pi*m))- (sin(wc1*m)./(pi*m));
h = hd.*y';
subplot(2,1,2), stem(h);
title('Impulse response of FIR filter');
%w = logspace(-1,1);
figure, freqz(h)
title('Frequency response of FIR Bandpass filter')
```

Output window

```
Enter the passband attenuation in dB: 1
Enter the stopband attenuation in dB: 60
Enter the passband frequency in pi rad: [0.35*pi 0.65*pi]
Enter the stopband frequency in pi rad: [0.2*pi 0.8*pi]
tr_width =  0.47124
M =  81
wc1 =  0.86394
wc2 =  2.2777
alpha =  40
```

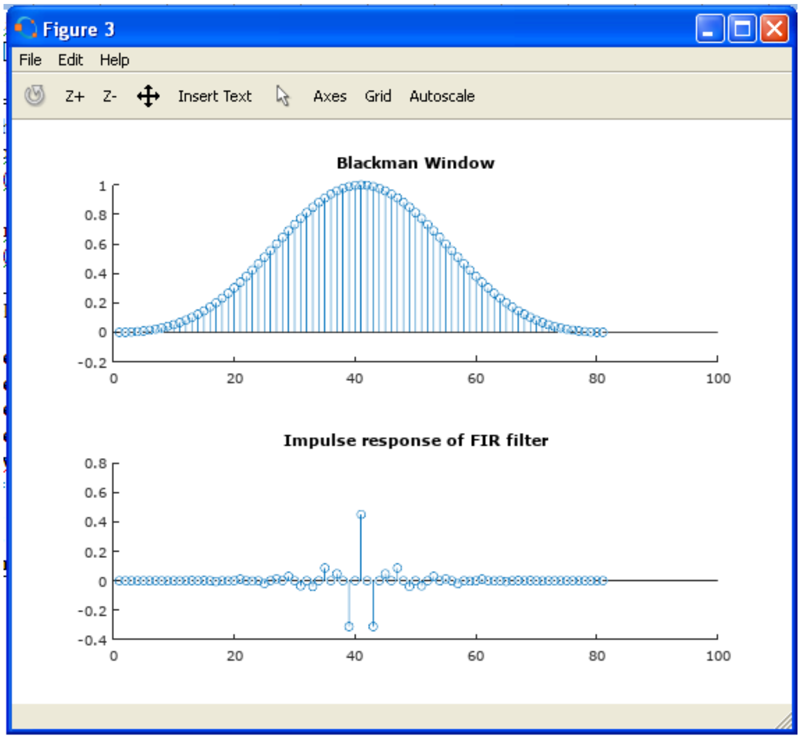


Fig. 10.3: Blackman window and Impulse response curve of FIR filter

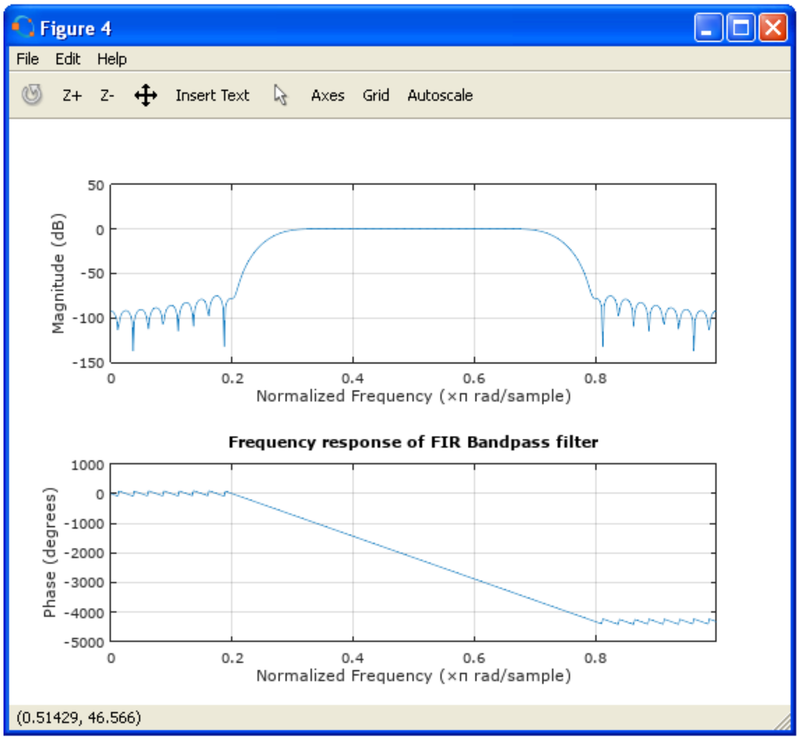


Fig. 10.4: Magnitude and phase angle curve of bandpass FIR filter

Note:

1. For highpass filter change the **hd** line in lowpass filter program to line specified below
   hd = sin(pi*m)./(pi*m)- sin(wc*m)./(pi*m)
2. For bandstop filter change the **hd** line in bandpass filter program to line specified below
   hd = (sin(wc1*m)./(pi*m)) + (sin(pi*m)./(pi*m)) - (sin(wc2*m)./(pi*m))

Result: FIR Filter design was verified using different windows.

Experiment 11:

Design and Implementation of FIR Filters using Frequency Sampling Technique

Objective: To design and implement FIR filters using frequency sampling technique to meet given specifications

Problem: Design an FIR low-pass filter for the given specification using frequency sampling approach
Wp = 0.28π; Pass band attenuation = 0.25db;
Ws = 0.57π; stop band attenuation = 20db;

```
M = 7;
alpha = (M-1)/2;
l = 0:M-1
wl = (2*pi/M)*l
%Hrs = [1,1,1,zeros(1,15),1,1]; %Ideal Amp Res sampled
Hrs = [1,1,0,0,0,0,1]
Hdr = [1,1,0,0];%Ploting ideal response i;e 1 in passband and 0 in stopband
wdl = [0,0.42,0.42,1]; %Ideal Amp Res for plotting
k1 = 0:floor((M-1)/2)
k2 = floor((M-1)/2)+1:M-1
angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)]
H = Hrs.*exp(j*angH)
h = real(ifft(H,M))
subplot(2,2,1)
plot(wl(1:4)/pi,Hrs(1:4),'o',wdl,Hdr)
title(„Frequency samples M=7)
fvtool(h)
```

Output:
l = [0 1 2 3 4 5 6]
wl = [0 0.8976 1.7952 2.6928 3.5904 4.4880 5.3856]
Hrs = [1 1 0 0 0 0 1]
k1 = [0 1 2 3]
k2 = [4 5 6]
angH = [0 -2.6928 -5.3856 -8.0784 8.0784 5.3856 2.6928]
H =
[1.0000 + 0.0000i -0.9010 - 0.4339i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i -0.9010 + 0.4339i]
h = [-0.1146 0.0793 0.3210 0.4286 0.3210 0.0793 -0.1146]

Output Waveform

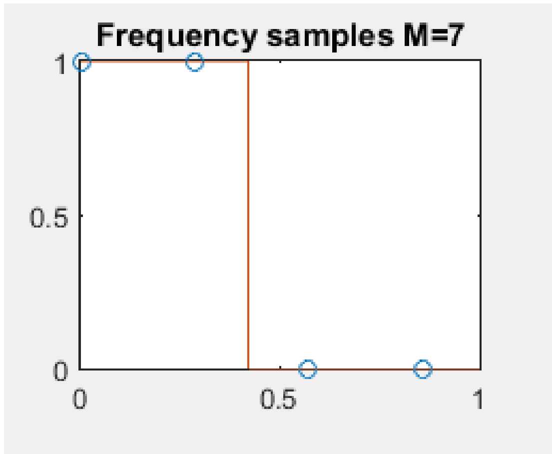


Fig. 11.1: frequency samples for M=7

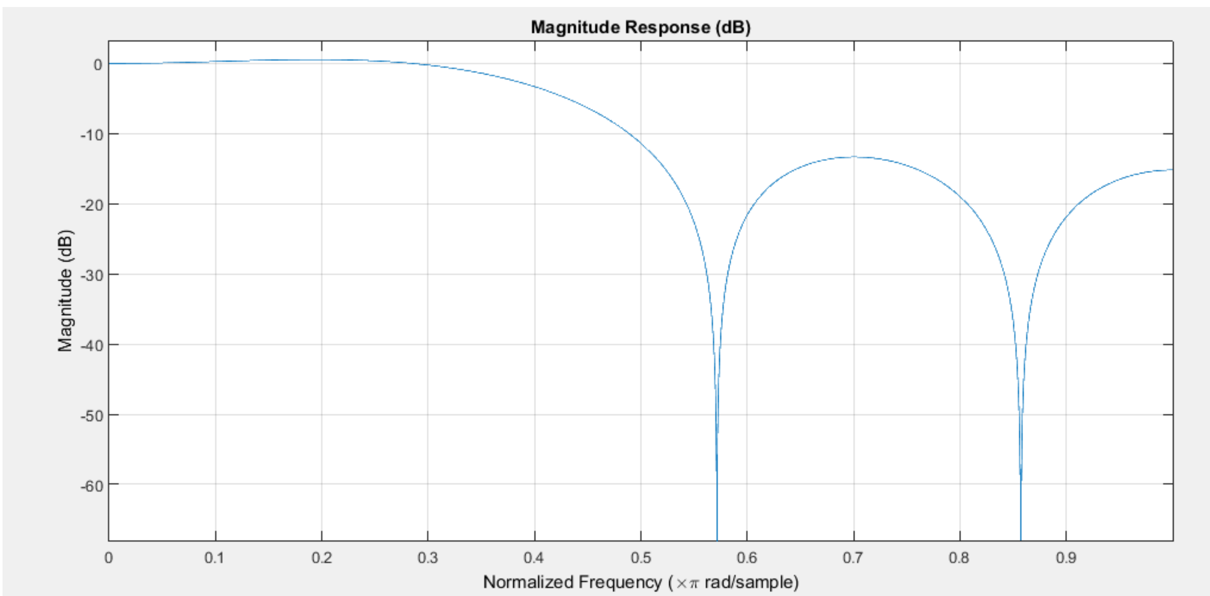


Fig. 11.2: magnitude response curve of lowpass filter

Design the FIR high-pass filter with following specifications frequency sampling technique.
Stop band edge: ws = 0.6π, As = 50db;
Pass band edge: wp = 0.8π, Rp = 1db;

```
M = 10;
alpha = (M-1)/2;
l = 0:M-1
wl = (2*pi/M)*l
%Hrs = [1,1,1,zeros(1,15),1,1]; %Ideal Amp Res sampled
Hrs = [0,0,0,0,1,1,1,0,0,0]
Hdr = [0,0,1,1];%Ploting ideal response i;e 1 in passband and 0 in stopband
wdl = [0,0.6,0.8,1]; %Ideal Amp Res for plotting
```

```
k1 = 0:floor((M-1)/2)
k2 = floor((M-1)/2)+1:M-1
angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)]
H = Hrs.*exp(j*angH)
h = real(ifft(H,M))
subplot(2,2,1)
plot(wl(1:6)/pi,Hrs(1:6),'o',wdl,Hdr)
title('Frequency samples M=10')
fvtool(h)
```

Output:
l = [0 1 2 3 4 5 6 7 8 9]
k1 = [0 1 2 3 4]
k2 = [5 6 7 8 9]
angH = [0 -2.8274 -5.6549 -8.4823 -11.3097 14.1372 11.3097 8.4823 5.6549 2.8274]
H = Columns 1 through 7
[0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.3090 + 0.9511i 0.0000 + 1.0000i 0.3090 - 0.9511i]
Columns 8 through 10
[0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i]
h = [0.0618 -0.1618 0.2000 -0.1618 0.0618 0.0618 -0.1618 0.2000 -0.1618 0.0618]

Output Waveform:

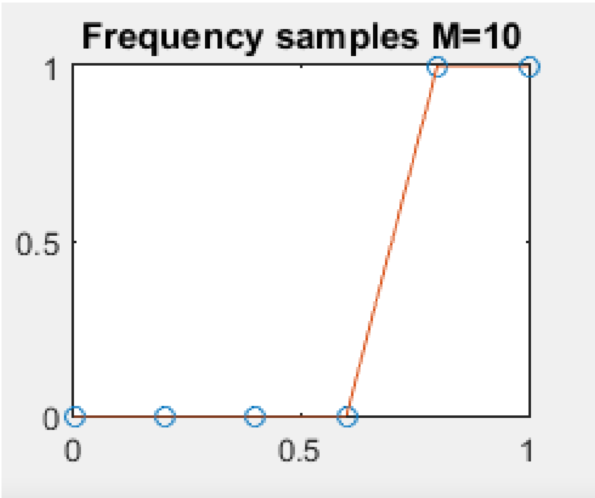


Fig. 11.3: frequency samples for M=10

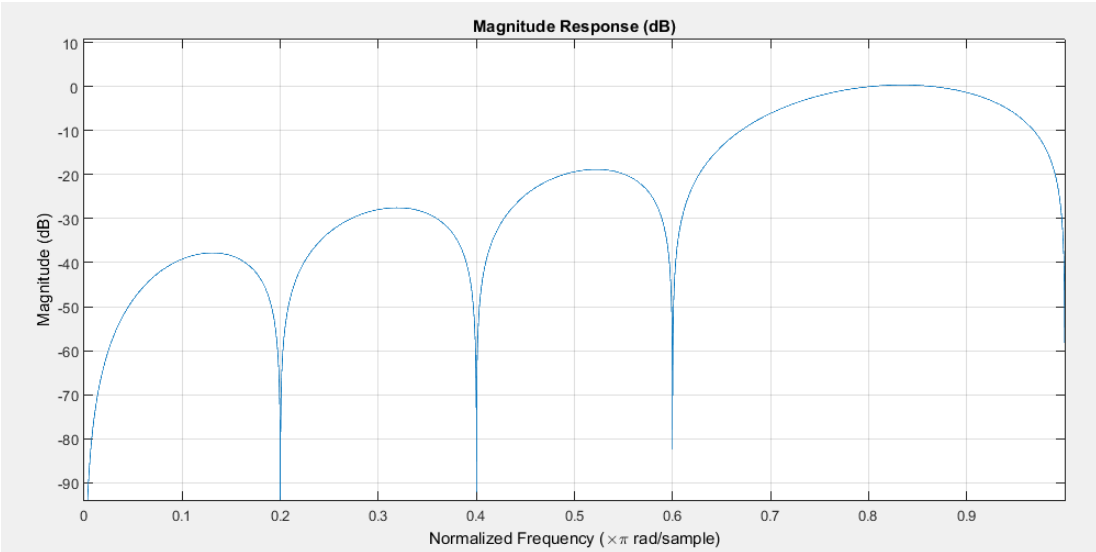


Fig. 11.4: magnitude response curve of highpass filter

Result: FIR Filter design was verified using frequency sampling technique.

Experiment 12:

Realization of IIR and FIR Filters

Objective:

- To realize the IIR and FIR filters..
- To verify the digital filter realization using simulation tool.

Theory:

The realization of a discrete filter can be done in hardware or in software. In either case, the implementation of the transfer function H(z) of a discrete filter requires delays, adders, and constant multipliers as actual hardware or as symbolic components. Figure depicts the operation of each of these components as block diagrams.

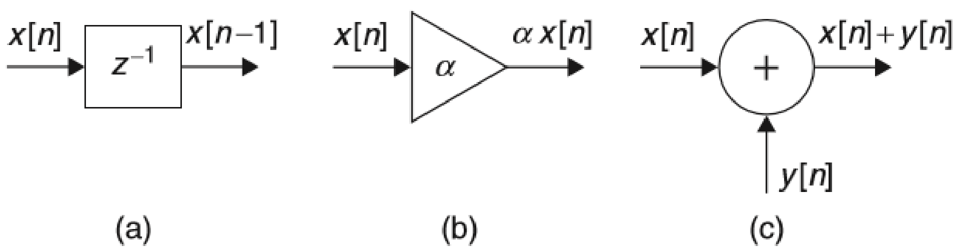


Fig 12.1: a) delay block b) multiplication block c) addition block

Realization of IIR Filters:

The structures commonly used to realize IIR filters are:

□□Direct forms I and II

□□Cascade

□□Parallel

FIR realization using Lattice structure:

Lattice Structure For a discrete Nth order all-pole or all-zero filter described by the polynomial coefficients a(n), n = 1, 2, …,N+1, there are N corresponding lattice structure coefficients k(n), n = 1, 2, …,N. The parameters k(n) are also called the reflection coefficients of the filter. Given these reflection coefficients, you can implement a discrete filter as shown below.

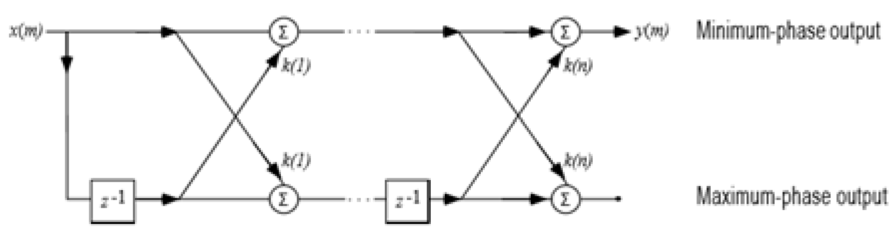


Fig. 12.2: FIR filter lattice structure

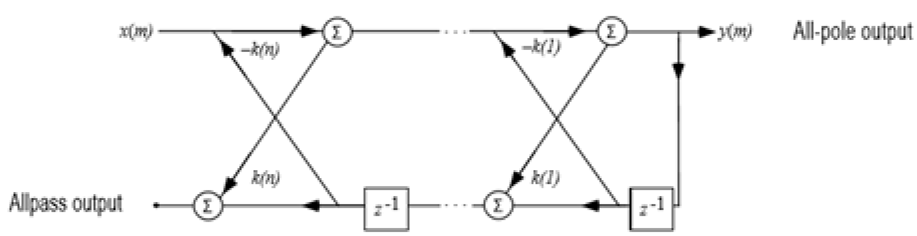


Fig. 12.3: IIR filter lattice structure

Problem: A filter is described by the following difference equations:
16y(n)+12y(n-2)+2y(n-2)-4y(n-3)-y(n-4)=x(n)-3x(n-1)+11x(n-2)-27x(n-3)+18x(n-4)
Determine its cascaded form structure and verify by converting to direct form.

Program:

```
function [b0,B,A] = dir2cas(b,a)
%compute gain coefficients
b0 = b(1);
b = b/b0;
a0 = a(1);
a = a/a0;
b0 = b0/a0;
M = length(b);
N = length(a);
if N > M
b = [b zeros(1,N-M)]
elseif M > N
a = [a zeros(1,M-N)]
N= M;
else
NM = 0;
end
K = floor(N/2)
B = zeros(K,3)
A = zeros(K,3)
if K*2 == N
b = [b 0]
a = [a 0]
end
broots = cplxpair(roots(b))
aroots = cplxpair(roots(a))
fori = 1:2: 2*K
Brow = broots(i : 1 : i+1,:)
Brow = real(poly(Brow))
B(fix((i+1)/2),:) = Brow
Arow = aroots(i : 1 : i+1,:)
Arow = real(poly(Arow))
A(fix((i+1)/2),:) = Arow
```

```
end
%Realization of IIR filters direct to cascade structure
b = [1 -3 11 -27 18]
a = [16 12 2 -4 -1]
[b0,B,A] = dir2cas(b,a)
%Realization of IIR filters cascade to direct structure
[b,a] = cas2dir(b0,B,A)
```

Output:

```
%direct to cascade realization
b0 =[0.0625]
B = [1.0000 0.0000 9.0000
1.0000 -3.0000 2.0000]
A = [1.0000 1.0000 0.5000
1.0000 -0.2500 -0.1250]
%cascade to direct realization
b = [0.0625 -0.1875 0.6875 -1.6875 1.1250]
a = [1.0000 0.7500 0.1250 -0.2500 -0.0625]
```

Problem;

Determine the coefficients k_m of the lattice filter corresponding to FIR filter described by the system function:

.$H(Z)=1+2Z^{-1}+0.33Z^{-2}$

Program:

```
b = [1 2 1/3]
k = tf2latc (b)
```

Output:

```
k =[1.5000  0.3333]
```

Result: Realization of filter was verified.

Viva questions:

1. Define signal and system
2. Differentiate between analog and digital signal
3. List the advantages and disadvantages of digital signal processing
4. What is DFT?
5. What are the different types of filters based on impulse response?
6. What are the different types of filters based on frequency response?
7. Distinguish between FIR and IIR filters.
8. What are the advantages and disadvantages of FIR filter
9. What are the techniques of designing FIR filters?
10. What are the advantages of Kaiser window?
11. What are the techniques used to analog to digital transformation?
12. What is prewarping?