

Laboratory Journal
of
DIGITAL SIGNAL PROCESSING

*For completion of term work of 6th semester
curriculum program*

Bachelor of Technology

In

ELECTRONICS AND TELECOMMUNICATION ENGINEERING



DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION
ENGINEERING

Dr. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY

Lonere-402 103, Tal. Mangaon, Dist. Raigad (MS)

INDIA

List of Practical

Exp. No.	Title
1.	To study Fourier transform of given sequence.
2.	To study sampling theorem
3.	To study design of various types of filter.
4.	To study impulse response of LTI system.
5.	To study IIR low pass filter.
6.	To study circular shifting property of sequence.
7.	Study of discrete Fourier transform.
8.	To implement DFT-FFT algorithm using matlab.
9	To study pole-zero plot of linear phase transfer function.
10	To implement comb filter
11	To illustrate step and impulse response of the system
12	To illustrate zero state and zero input response for first and second order system

EXPERIMENT NO.1

Aim: To study Fourier transform of given sequence.

Theory: Fourier transform convert the time domain to frequency domain. It is of two types.

- i. **CTFT:** Continuous Time Fourier Transform
- ii. **DTFT:** Discrete Time Fourier Transform

The DTFT of sequence $x[n]$ is given by

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

Where $x[n]$ should be absolutely summable. i.e. $\sum |x(n)| < \infty$

1. Fourier transform is periodic with period 2π .

2. Magnitude of Fourier transform is given as

$$|X(e^{j\omega})| = \sum_{n=-\infty}^{\infty} |x(n)|$$

3. Phase is given as $\angle X(e^{j\omega}) = \omega$

4. If $x[n]$ is given sequence , then

FT of $x[n]=X(\omega)$ symmetric

Real part of $x[n]=X_R(\omega)$ symmetric

Imaginary Part $x[n]=X_i(\omega)$ antisymmetric

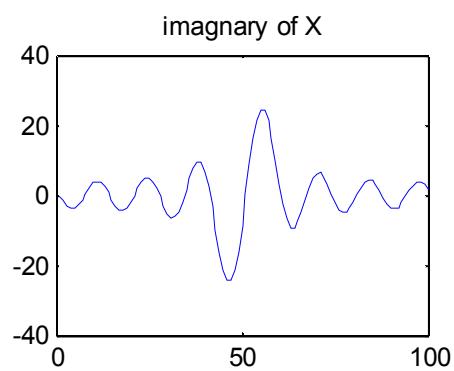
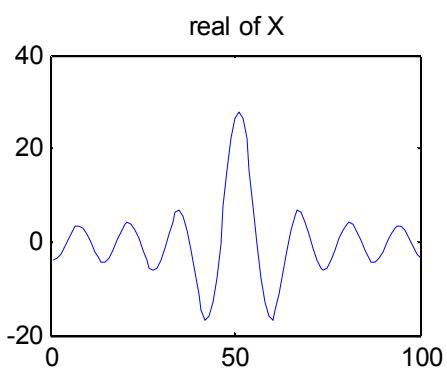
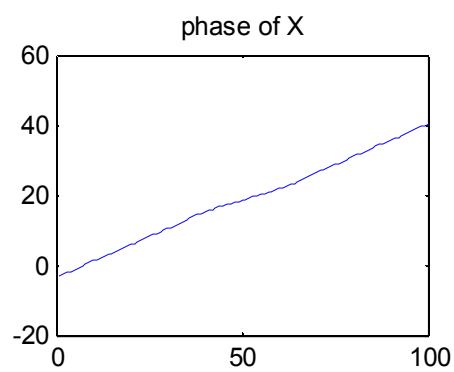
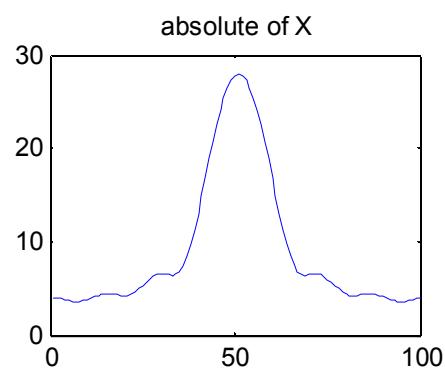
Phase of $x[n]=\angle X(\omega)$ antisymmetric

Conclusion: Thus we have studied Fourier transform.

```
%Fourier Transform%
x=input('enter the signal X:')
N=length(x)
w=-pi
for i=1:1:100
    n=1:1:N
    e=exp(-j*w*n)
    xft(i)=x*e'
    w=w+(2*pi)/100
end
subplot(2,2,1)
plot(abs(xft))
title('absolute of X')
subplot(2,2,2)
plot(phase(xft))
title('phase of X')
subplot(2,2,3)
plot(real(xft))
title('real of X')
subplot(2,2,4)
plot(imag(xft))
title('imaginary of X')
```

Output:

enter the signal x:[1 2 3 4 5 6 7]



EXPERIMENT NO.2

Aim: To study sampling theorem.

Theory:

Definition: It states that sampling frequency must be greater than or equal to twice that of maximum signal frequency.

$$f_s \geq 2f_m$$

$$X[n] = x[nT] = x[t] \mid t=nT$$

Where T=sampling period = $\frac{1}{f}$ =sampling frequency (f_s)

$$f_s \geq 2f_m$$

f_m =maximum frequency of signal

Conclusion: Hence we have studied the sampling theorem.

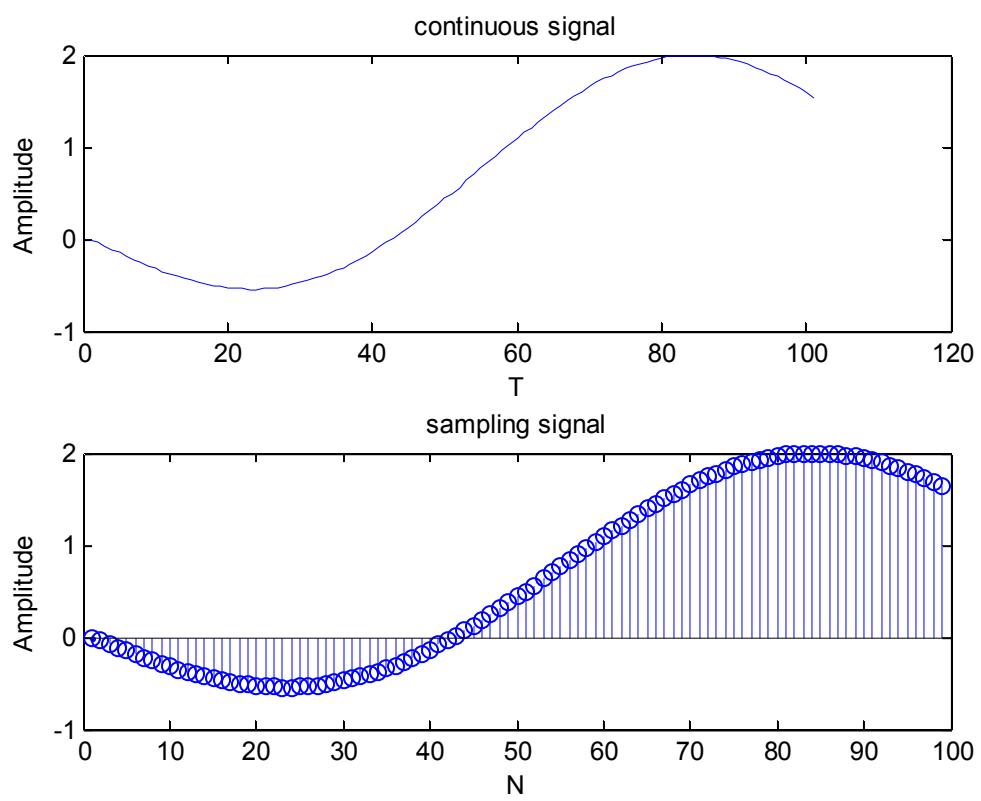
```

%sampling theorem%
f1=input('enter first frequency:')
f2=input('enter second frequency:')
N=input('enter no. of samples required:')
t=0:0.01:1
T=1
y=sin(2*pi*f1*t)-sin(2*pi*f2*t)
for n=1:1:N
    x(n)=y(n*T)
end
subplot (2,1,1)
plot(y)
xlabel('T')
ylabel('Amplitude')
title('continuous signal')
subplot(2,1,2)
stem(x)
xlabel('N')
ylabel('Amplitude')
title('sampling signal')

```

Output:

Enter the first frequency:0.3
 Enter the second frequency:0.9
 Enter no. of samples required:99



EXPERIMENT NO.3

Aim: To study design of various types of filter.

Theory: There are four types of filters

- a. Low pass filter
- b. Band pass filter
- c. Band stop filter
- d. High pass filter

- a. **Low pass filter:** It allows low frequencies to pass through it. For low-pass filter transfer function is

$$H(z) = \left(\frac{1+\alpha}{2}\right) \left(\frac{1+z^{-1}}{1+\alpha z^{-1}}\right)$$
$$\omega=0 \quad z=e^{j0}=1$$
$$\omega=\pi \quad z=e^{j\pi}=-1$$

Syntax in matlab: B1=FIR(N, ω_n , 'low');

- b. **High pass filter:** It allows high frequencies to pass through it. For high-pass filter transfer function is

$$H(z) = \left(\frac{1+\alpha}{2}\right) \left(\frac{1-z^{-1}}{1+\alpha z^{-1}}\right)$$
$$\omega=0 \quad z=e^{j0}=1$$
$$\omega=\pi \quad z=e^{j\pi}=-1$$

Syntax in matlab: B2=FIR(N, ω_n , 'high');

- c. **Band pass filter:** It allows certain frequencies of band to pass through it.

Syntax in matlab: B3=FIR(N, ω , 'band-pass');

- d. **Band stop filter:** It attenuates certain band of frequencies.

Syntax in matlab: B4=FIR(N, ω , 'band-stop');

Conclusion: Hence we have studied design of all types of filter.

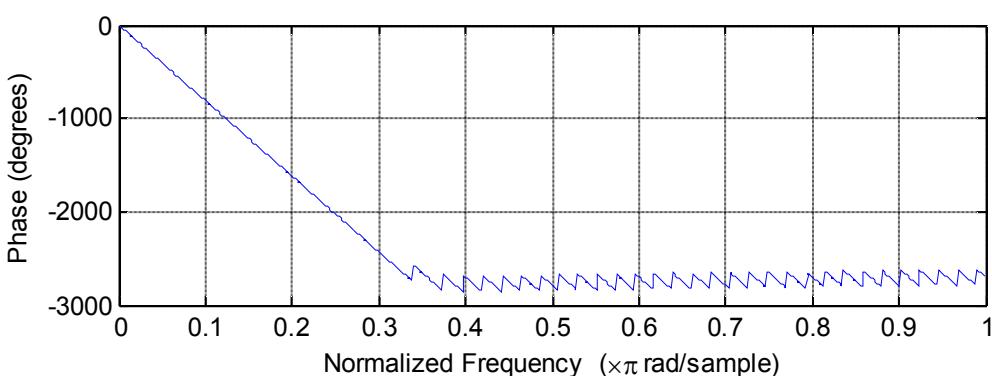
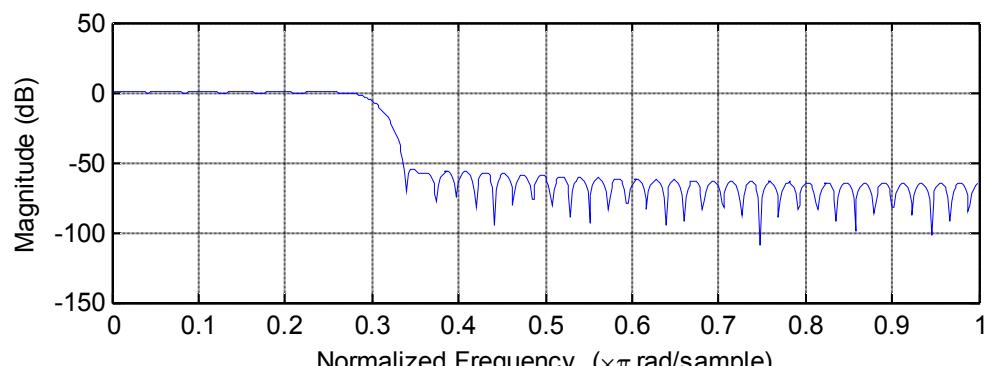
```

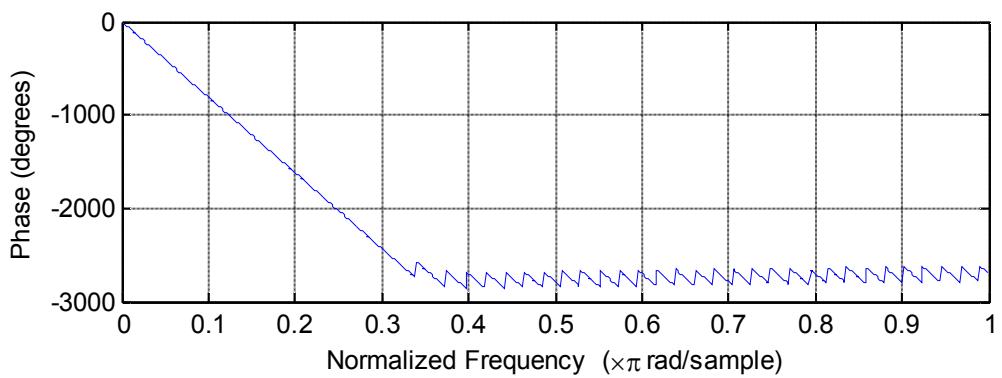
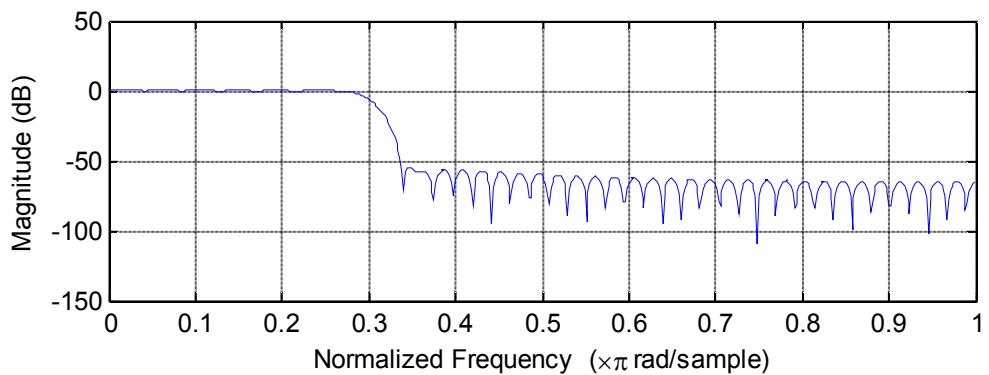
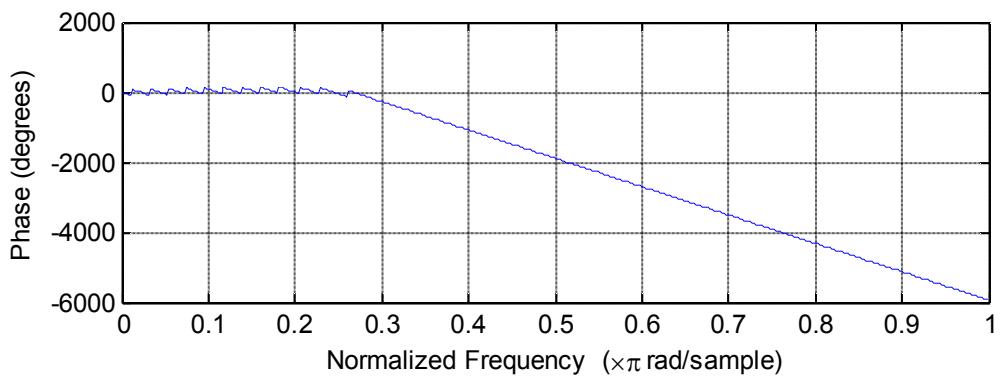
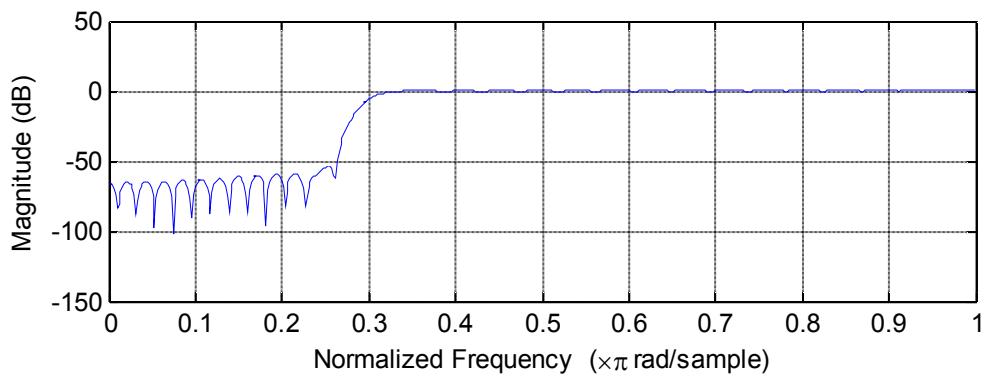
%design of filters%
N=input('enter the order of fir filter')
wn=input('enter the cutoff frequency')
w1=input('enter the first cutoff frequency')
w2=input('enter the second cutoff frequency')
W=[w1 w2]
B1=FIR1(N,wn,'low')
B2=FIR1(N,wn,'high')
B3=FIR1(N,w,'bandpass')
B1=FIR1(N,w,'stop')
FREQZ(B1)
figure
FREQZ(B2)
figure
FREQZ(B3)
figure
FREQZ(B4)
Figure

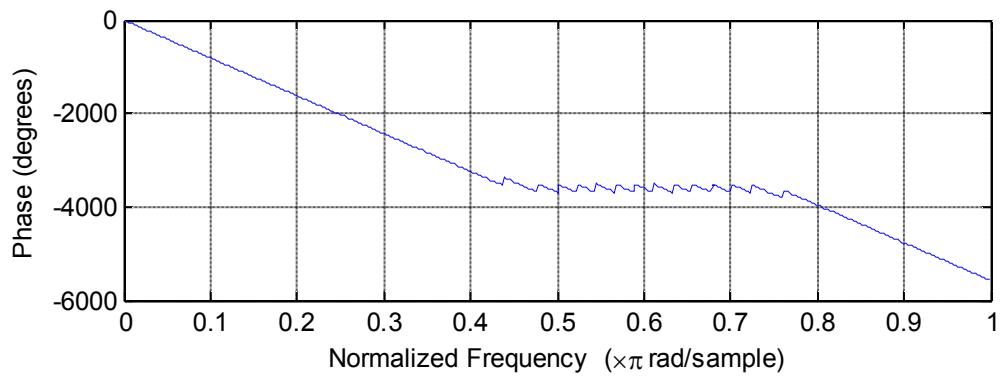
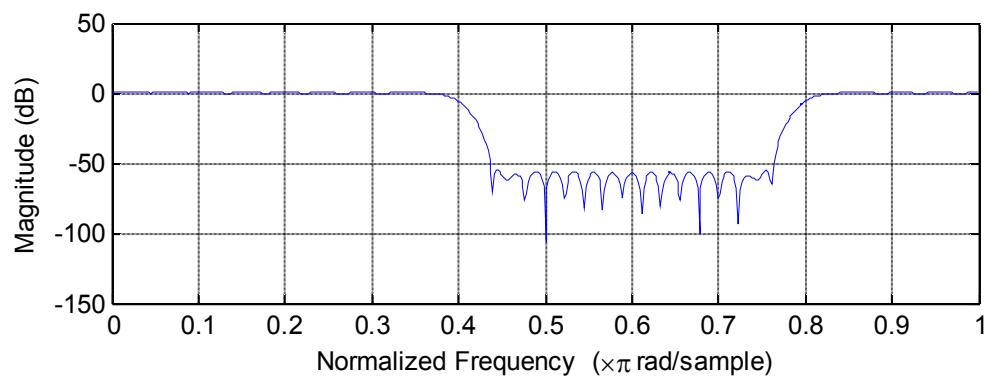
```

Output:

Enter the order of FIR filter:90
 Enter the cut off frequency:0.30
 Enter the first cut off frequency:0.4
 Enter the second cut off frequency:0.80







EXPERIMENT NO.4

Aim: To study impulse response of LTI system.

Theory: When $x(n)$ and $h(n)$ are two causal sequence are given as input to the LTI system then output is

$$y[n] = \sum_{k=0}^{p-1} h[k]x[n - k]$$

Then using above equation we can find the values of $h[n]$ as follows

$$y[0] = \sum_{k=0}^{p-1} h[k]x[0 - k]$$

$$y[0] = h[0] x[0]$$

$$h[0] = \frac{y[0]}{x[0]} \quad \dots \quad (1)$$

$$y[1] = \sum_{k=0}^{p-1} h[k]x[1 - k]$$

$$y[1] = h[0] x[1] + h[1] x[0]$$

$$h[1] = \frac{y[1] - h[0]x[1]}{x[0]} \quad \dots \quad (2)$$

$$y[2] = \sum_{k=0}^{p-1} h[k]x[2 - k]$$

$$h[2] = \frac{y[2] - h[0]x[2] - h[1]x[1]}{x[0]} \quad \dots \quad (3)$$

By observing equation (1), (2), (3)

We can write the general formula for $h[n]$ as follows

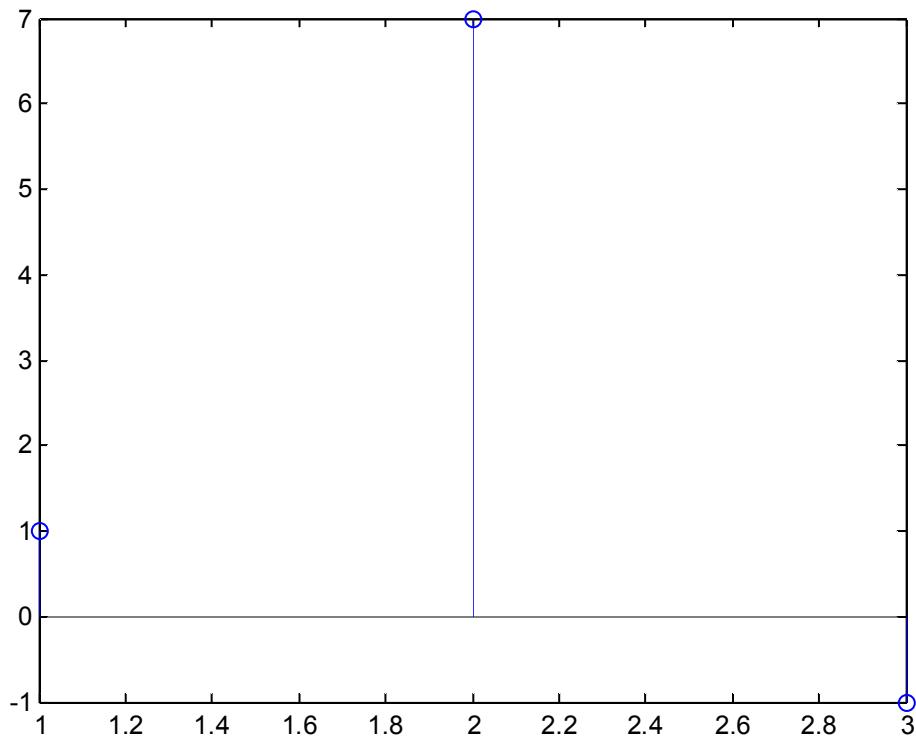
$$h[n] = \frac{y[n] - \sum_{k=0}^{p-1} h(k)x(n-k)}{x[0]}$$

Conclusion: We can find the impulse response input if we know that the input sequence and total output of LTI system.

```
%impulse response of LTI system%
clear all
x=input('enter the sequence')
y=input('enter the response')
h(1)=y(1)/x(1)
n1=length(x)
n2=length(y)
N=n2-n1+1
for n=2:1:N
    s=0
    for k=1:1:n-1
        s=s+h(k)*x(n-k+1)
    end
    h(n)=(y(n)-s)/x(1)
end
disp(h)
stem(h)
```

Output:

enter the sequence[1 1 1]
 enter the response [1 8 7 6 5]



EXPERIMENT NO.5

Aim: To study IIR low pass filter.

Theory: Simple IIR filter pole to be exist not zero.

The transfer function of IIR low pass filter is given as

$$H(z) = \left(\frac{1+\alpha}{2}\right) \left(\frac{1+z^{-1}}{1+\alpha z^{-1}}\right)$$

$$\omega=0, z=1$$

$$\omega=\pi, z=-1$$

The cut-off frequency of the filter is given as

$$\omega_c = \cos^{-1} \left(\frac{2\alpha}{1+\alpha^2} \right)$$

$$\alpha = \frac{-1 \pm \sin \omega_c}{\cos \omega_c}$$

Conclusion: Thus we studied low pass filter by observing its magnitude and frequency response.

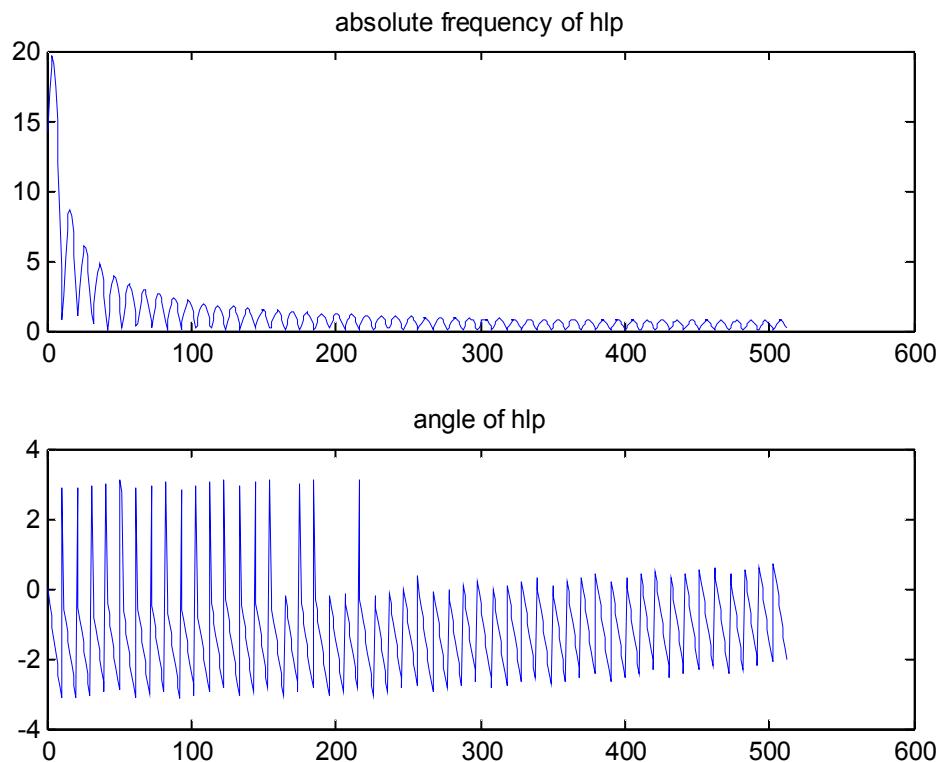
```

%IIR low pass filter%
wc=input('enter the frequency:')
a1=(1+sin(wc))/cos(wc)
a2=(1-sin(wc))/cos(wc)
A=min(a1,a2)
for i=1:1:100
    b=exp(-j*wc)
    hlp(i)=((1-A)/2)*(1+b)/(1-A*b)
    wc=wc+2*pi/100
end
subplot(2,1,1)
plot(abs(freqz(hlp)))
title('absolute frequency of hlp')
subplot(2,1,2)
plot(angle(freqz(hlp)))
title('angle of hlp')

```

Output:

Enter the frequency: 0.33



EXPERIMENT NO.6

Aim: To study circular shifting property of sequence.

Theory: If $x(n)$ is given sequence of length N , l is circular shift factor then $x(n)$ can be circularly shifted as follows

Eg. $x[n]=\{1,2,3,4\}$ and $l=2$

$x_1[n]=\{3,4,1,2\}$

It can be shifted by writing a program using if as follows

$x_1[n]=x[n-l]$, when $l < n \leq N-1$

$x_1[n]=x[n-l+N]$, when $0 < n \leq l$

Conclusion : Hence we have studied the circular shift property of a given sequence.

```
%circular shifting property%
clc
x=input('enter the sequence')
N=length(x)
l=input ('enter the shift factor' )
for n=1:1:N
    if (n>l)
        x1(n)=x(n-l)
    else
        x1(n)=x(n-l+N)
    end
end
disp(x1)
```

Output:

enter the sequence[6 5 4 3 2 1]

Output:

x =
6 5 4 3 2 1

N =
6

enter the shift factor3

l =
3
x1 =
3 6 1 2 3 4
x1 =
3 2 1 2 3 4
x1 =
3 2 1 2 3 4
x1 =
3 2 1 6 3 4
x1 =
3 2 1 6 5 4
x1 =
3 2 1 6 5 4
3 2 1 6 5 4

EXPERIMENT NO.7

Aim : Study of discrete Fourier transform.

Theory:

It is sampled from a continuous time Fourier transform

$x(n)$ is sequence of length N

$$X(\omega) = \sum_{n=0}^{N-1} x(n)e^{-j\omega n}$$

$$x(k) = X(\omega)|_{\omega=\frac{2\pi k}{N}}$$

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j(\frac{2\pi}{N})nk}$$

$$k=0,1,\dots,N-1$$

DFT is periodic with period N

$$\text{i.e. } X[k+N] = X[k]$$

$$X^*[n-k] = X[k]$$

For each value of k , N times complex multiplications are needed while $N-1$ times additions are needed.

While for N point DFT we require

N^2 =complex multiplication

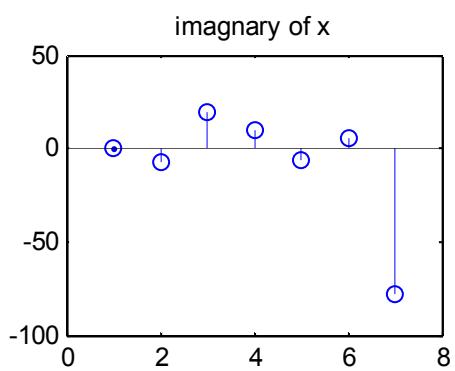
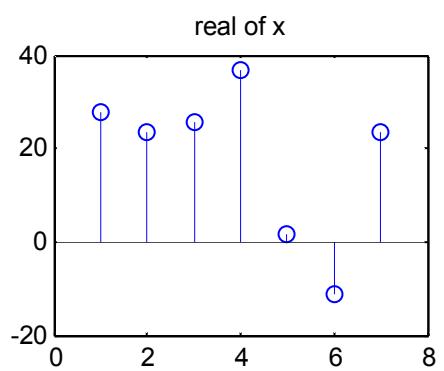
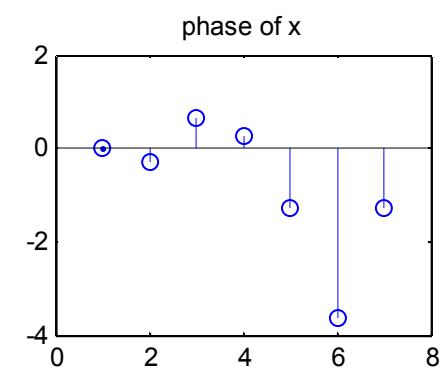
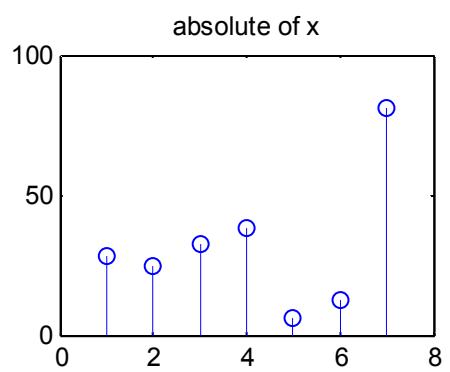
$N(N-1)$ =complex addition

Conclusion: Hence we studied discrete Fourier transform of sequence.

```
%discrete Fourier transform%
x=input('enter the discrete signal x')
N=length(x)
for k=0:1:N-1
    n=0:1:N-1
    e=exp(-j*2*pi*n*k/N)
    x(k+1)=x*e'
end
subplot(2,2,1)
stem(abs(x))
title('absolute of x')
subplot(2,2,2)
stem(phase(x))
title('phase of x')
subplot(2,2,3)
stem(real(x))
title('real of x')
subplot(2,2,4)
stem(imag(x))
title('imaginary of x')
```

Output:

Enter the discrete signal X:[1 2 3 4 5 6 7]



EXPERIMENT NO.8

Aim: To implement DFT-FFT algorithm using matlab.

Software: Matlab

Theory: Fast Fourier Transform (FFT). The FFT is an algorithm that efficiently computes the DFT.

The DFT of a sequence $x[n]$ of length N is given by complex value sequence, $X(k)$

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi n k}{N}} \quad 0 < k \leq N-1$$

Let ω_n be the complex value phase factor which is root of unity.

$$\omega_n = e^{-j\frac{2\pi}{N}}$$

$$X(k) = \sum_{n=0}^{N-1} x[n] W_N^{nk} \quad 0 < k \leq N-1$$

$$\text{Similarly, } x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{nk} \quad 0 < n \leq N-1$$

DFT is primarily inefficient as it does not expect the symmetry and periodicity. An efficient algorithm for DFT is FFT algorithm.

Symmetric property $W_N^k = -W_N^k$

Periodicity property $W_N^{k+N} = W_N^k$

Decimation in FFT

Let us assume that $x(n)$ represents sequence of N values where N is integer of power 2 i.e. $N=2^m$

The given sequence is decimated into two $N/2$ point segments consisting of even and odd segment.

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x[n] W_N^{nk} \\ &= \sum_{n=0}^{N/2-1} x[2n] W_N^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1] W_N^{2nk+k} \\ &= \sum_{n=0}^{N/2-1} x[2n] W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x[2n+1] W_{N/2}^{nk} \end{aligned}$$

From signal flow graph

$$g_0(0)=x(0)+x(4); \quad h_0(0)=x(1)+x(5);$$

$$g_0(1)=x(0)-x(4); \quad h_0(1)=x(1)-x(5);$$

$$g_1(0)=x(2)+x(6); \quad h_1(0)=x(3)+x(7);$$

$$g_1(1)=x(2)-x(6); \quad h_1(1)=x(3)-x(7);$$

for 4-point DFT

$$G(0)=g_0(0)+w_4^0 g_1(0); \quad H(0)=h_0(0)+w_4^0 h_1(0)$$

$$\begin{array}{ll}
 G(1) = g_0(1) + w_4^1 g_1(1); & H(1) = h_0(1) + w_4^1 h_1(1) \\
 G(2) = g_0(0) + w_4^2 g_1(0); & H(2) = h_0(0) + w_4^2 h_1(0) \\
 G(3) = g_0(1) + w_4^3 g_1(1); & H(3) = h_0(1) + w_4^3 h_1(1)
 \end{array}$$

For 8-point DFT

$$\begin{aligned}
 X(0) &= G(0) + w_8^0 H(0) \\
 X(1) &= G(1) + w_8^1 H(1) \\
 X(2) &= G(2) + w_8^2 H(2) \\
 X(3) &= G(3) + w_8^3 H(3) \\
 X(4) &= G(0) + w_8^4 H(0) \\
 X(5) &= G(1) + w_8^5 H(1) \\
 X(6) &= G(2) + w_8^6 H(2) \\
 X(7) &= G(3) + w_8^7 H(3)
 \end{aligned}$$

Conclusion : Hence we have implemented DFT FFT algorithm and concluded that we can reduce the multiplier by using DFT FFT.

```

%algorithm for fft%
x=input('enter the length and sequence')
for i=1:1:4
    g(i)=x(i)+x(i+4)
    g(i+4)=x(i)-x(i+4)
end
for i= 1:1:2
    h(i)=g(i)+g(i+2)
    h(i+2)=g(i)-g(i+2)
end
for i=5:1:6
    h(i)=g(i)-(j*(g(i+2)))
    h(i+2)=g(i)+(j*(g(i+2)))
end
y(i)=bitreorder(h(i))
for i=1:1:4
    X(i)=y(i)+exp(-j*2*pi*(i-1)/8)*y(i+4)
    X(i+4)=y(i)+exp(-j*2*pi*(i+3)/8)*y(i+4)
end
z=fft(x)
disp('X=')
disp (X)
disp('z')
disp(z)

```

Output:

Enter the length and sequence [1 2 3 4 5 6 7 8]

X=

36.0000	-4.0000+9.6569i	-4.0000+4.0000i	-4.0000+1.6569i
4.0000	-4.0000-1.6569i	-4.0000-4.0000i	-4.0000-9.6569i

Z=

36.0000	-4.0000+9.6569i	-4.0000+4.0000i	-4.0000+1.6569i
4.0000	-4.0000-1.6569i	-4.0000-4.0000i	-4.0000-9.6569i

EXPERIMENT NO.9

Aim:- To study pole-zero plot of linear phase transfer function.

Theory:- Linear phase transfer function

For function to be linear phase, phase response should be such that

$$\text{Phase response } \Phi(\omega) = k\omega + c$$

Type 1:

Symmetric impulse response of odd length

$$h(n) = h(l-1-n)$$

if length = 9;

$$h(n) = h(8-n)$$

$$h[1] = h[7]$$

$$h[2] = h[6]$$

$$h[3] = h[5]$$

$$h[4] = h[4]$$

$$h[n] = \{h_0, h_1, h_2, h_3, h_4, h_3, h_2, h_1, h_0\}$$

$$\text{For this IR, } H(\omega) = e^{jN\omega/2} \left[\sum_{i=0}^{N/2-1} 2 h_i \cos\left(\frac{N}{2} - i\right)\omega + \frac{h_0}{2} \right]$$

$$\begin{aligned} \text{And } H(\omega) &= \begin{cases} \frac{N}{2} \omega & \text{if } H_R(\omega) > 0 \\ \frac{N}{2} \omega + \pi & \text{if } H_R(\omega) < 0 \end{cases} \end{aligned}$$

Type2:

Symmetric impulse response of even length

Let L=6

$$h[n] = \{h_0, h_1, h_2, h_2, h_1, h_0\}$$

Type3:

Anti-symmetric impulse response of odd length

Let L=7

$$h[n] = \{h_0, h_1, h_2, 0, -h_2, -h_1, -h_0\}$$

Type4:

Anti-symmetric impulse response of even length

Let L=6

$$h[n] = \{h_0, h_1, h_2, -h_2, -h_1, -h_0\}$$

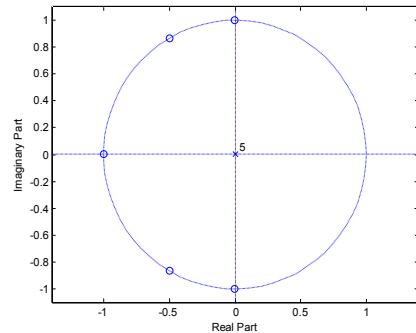
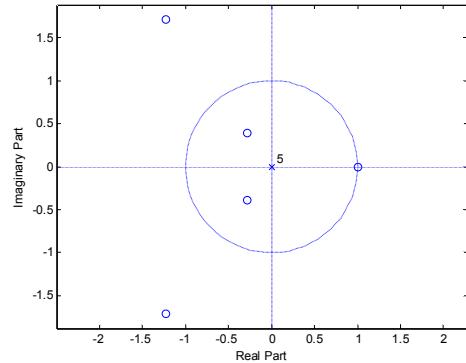
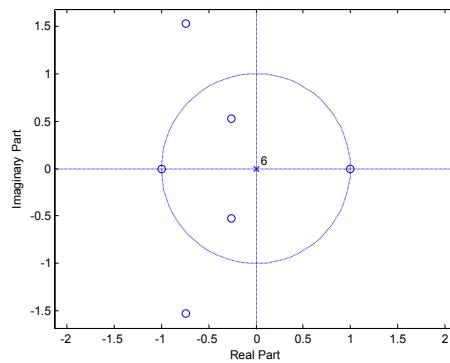
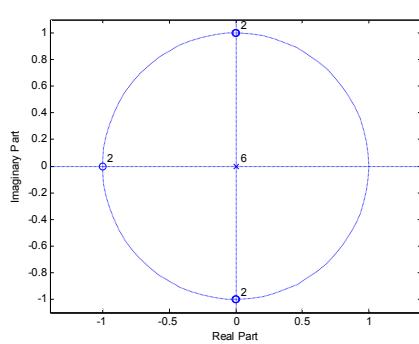
Z-plane: Z-plane zero plot of zeros Z and pole P with unit circle for reference. Each zero is represented with 'O' and each pole with 'X'. Multiple zeros and poles are indicated by multiplicity number shown to upper right of zero or pole.

Conclusion: Thus we have pole-zero plot of linear phase transfer function.

```
%to find pole zero plot of linear phase transfer function%
w=input('enter the first sequence')
x=input('enter the second sequence')
y=input('enter the third sequence')
z=input('enter the forth sequence')
zplane(w)
figure
zplane(x)
figure
zplane(y)
figure
zplane(z)
figure
```

Output

Enter the first sequence [1 2 3 4 3 2 1]
 Enter the second sequence [1 2 3 0 -3 -2 -1]
 Enter the third sequence [1 2 3 -3 -2 -1]
 Enter the forth sequence [1 2 3 3 2 1]



EXPERIMENT NO.10

Aim:-To implement comb filter

Theory:-

- a. Transfer function of comb filter with low pass filter

$$G(z) = H_{LP}(z)^L = (1+z^{-L})/2$$

$$\begin{aligned} H_{LP}(e^{jw})^L &= (1+e^{-jwL})/2 \\ &= e^{-jwL/2}(e^{jwL/2} + e^{-jwL/2})/2 \\ &= e^{-jwL/2}\cos(wL/2) \end{aligned}$$

$$\begin{aligned} |G(w)| &= |\cos(wL/2)| \\ |G(w)| \text{ will be zero when } wL/2 &= (2k+1)\pi/2 \\ \text{i.e. } w &= (2k+1)\pi/L \text{ for } k=0,1,2,3,\dots \end{aligned}$$

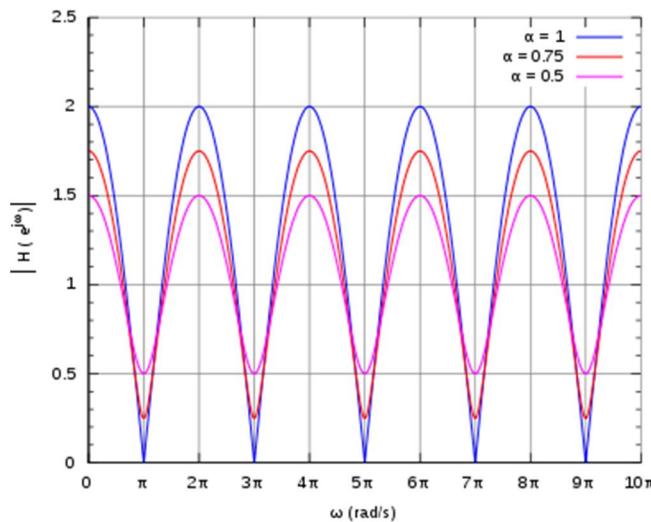


Fig.a. Comb Filter with low pass filter.

- b. Transfer function of comb filter with high pass filter

$$H_{HP}(z) = (1-z^{-L})/2$$

$$\begin{aligned} G(w) = H_{HP}(z^L) &= (1-z^{-L})/2 \\ &= (1-e^{-jwL})/2 \\ &= e^{jwL/2}(e^{jwL/2} - e^{-jwL/2})/2 \\ &= j e^{-jwL/2}\sin(wL/2) \end{aligned}$$

$$\begin{aligned} |G(w)| &= |\sin(wL/2)| \\ |G(w)| = 0 &\text{ when } wL/2 = k\pi \\ \text{i.e. } w &= 2\pi k/L \text{ for } k=0,1,2,3,\dots \end{aligned}$$

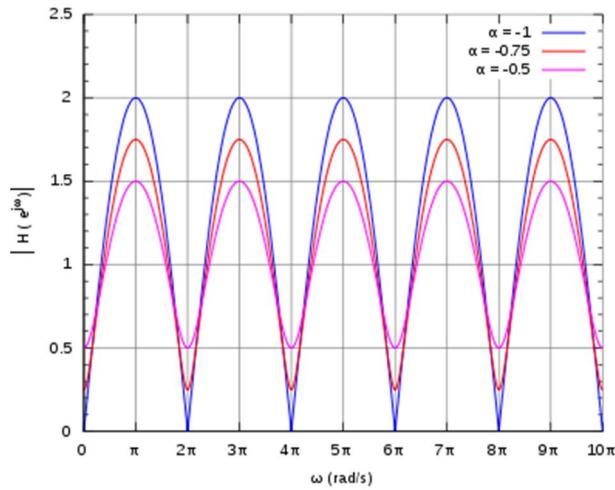


Fig.b. Comb Filter with high pass filter.

Conclusion:-Thus we have studied implementation of comb filter.

```
%to implement comb filter%
clc
L=input('enter the length of filter')
n=input('enter the response of filter')
w=0
for l=1:1:100
    e=exp(-j*w*L/2)
    y(l)=e*cos(w*L/2)
    w=w+(2*pi/100)
end
plot(abs(y))
```

Output:

enter the length of filter6

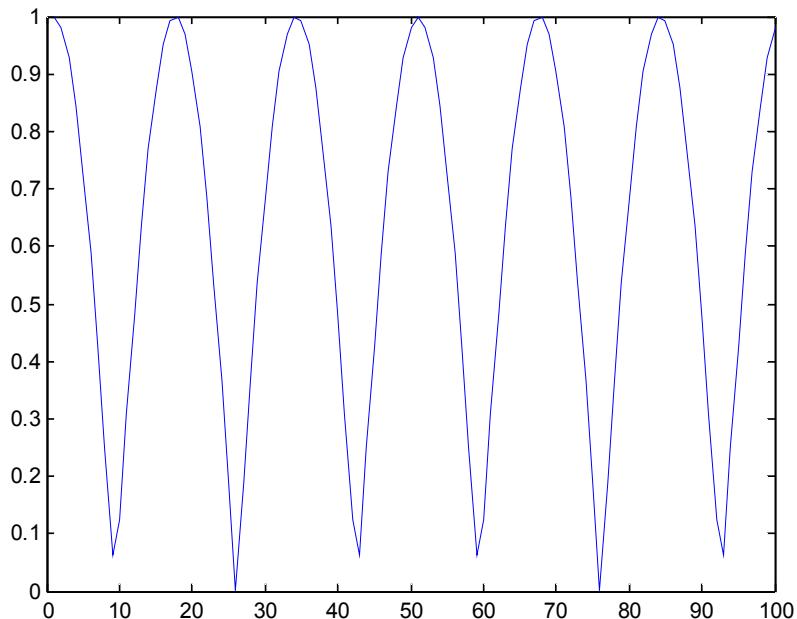
L =

6

enter the response of filter6

n =

6



EXPERIMENT NO.11

Aim:

To illustrate step and impulse response of the system

Software:

MATLAB 7.3

Theory:

Step response: It is defined as the response or output of the system when an input applied to the system is unit step signal.

Impulse response: It is defined as response or output of the system when input applied is impulse signal.

Program:

```
%step and impulse reasponse  
num=[2];  
den=[2 1 2];  
printsys(num,den)  
a = tf(num,den);  
subplot(211)  
impulse(a);  
subplot(212)  
step(a)
```

Conclusion:

Thus, we have illustrated the step and impulse response of a system using matlab.

Sample output

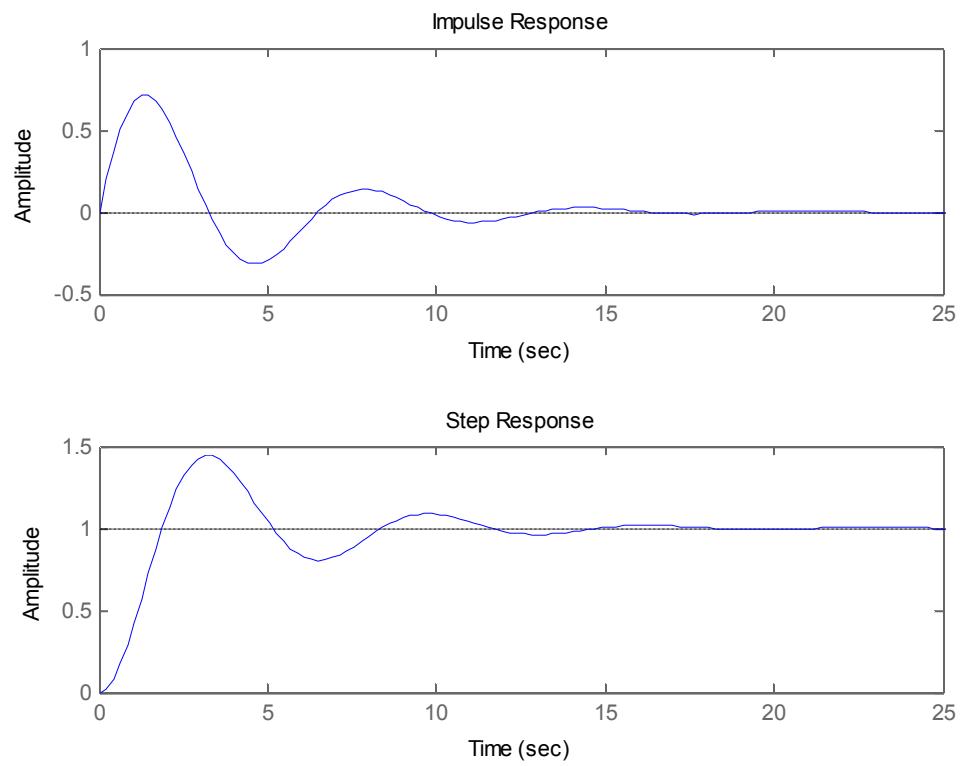
```
*****
```

num/den =

2

2 s^2 + s + 2

>>



EXPERIMENT NO. 12

Aim:

To illustrate zero state and zero input response for first and second order system

Software:

MATLAB 7.3

Theory:

Zero state response: It is the response of the system when all initial conditions are assumed to zero and considering only the input applied to the system.

Zero input response: It is the response of the system when input is zero and initial conditions are considered. It gives state of the system before any input is applied.

Program:

```
%FIRST ORDER FUNCTION
function [ydot]=myfun1(t,y)
ydot=-4*y+2*cos(2*t).*(t >= 0);
%END OF FUNCTION
%matlab program to solve first order differential Eqution
tspan=[0:0.01:15];
y0=[2];
[t,y]=ode23('myfun1',tspan,y0);
%solve ODE using ode23
plot(t,y,'k');
grid on;
xlabel('time');
ylabel('output response y(t)');
title('total response=ZIR+ZSR:MATLAB Result verification for First order Differential
equation');
figure;
y=1.6*exp(-4*t)+0.2*sin(2*t)+0.4*cos(2*t);
plot(t,y,'k');
grid on;
xlabel('time');
```

```

ylabel('output Response y(t)');
title('Total Response=ZIR+ZSR:Analytical Result Verification for first order Differential
Equation');
*****
%SECOND ORDER DIFFERENTIAL EQUATION
function [ydot]=myfunct2(t,y)
%usage:ydot=myfunc2(t,y)
ydot(1,1)=-5*y(1)-4*y(2)+3*cos(t)*(t>=0);
ydot(2,1)=y(1);
%end of function
Clear all;
tspan=[0:0.02:20];
y0=[-5;2];
[t,y]=ode23('myfunc2',tspan,y0)
plot(t,y(:,2),'k')
grid on;
xlabel('time');
ylabel('output Response y(t)');
title('Total Response=ZIR+ZSR:for second order Differential Equation');
figure;
y=(1/2)*exp(-t)+(21/17)*exp(-4*t)+(9/34)*cos(t)+(15/34)*sin(t);
plot(t,y,'k')
grid on;
xlabel('time');
ylabel('output Response y(t)');
title('Total Response=ZIR+ZSR:Analytical Result Verification for second order
Differential Equation');

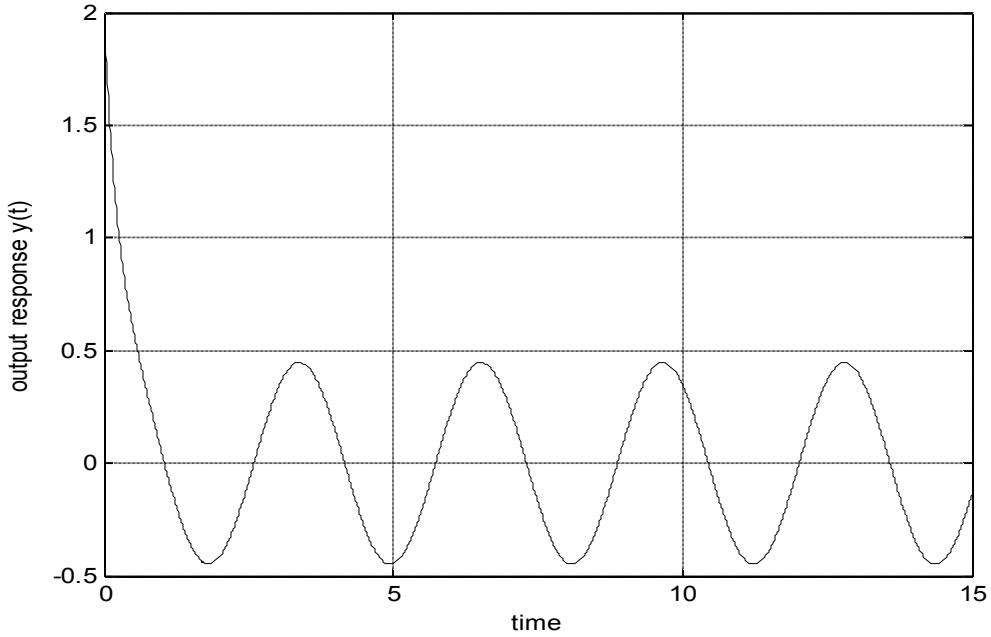
```

Conclusion:

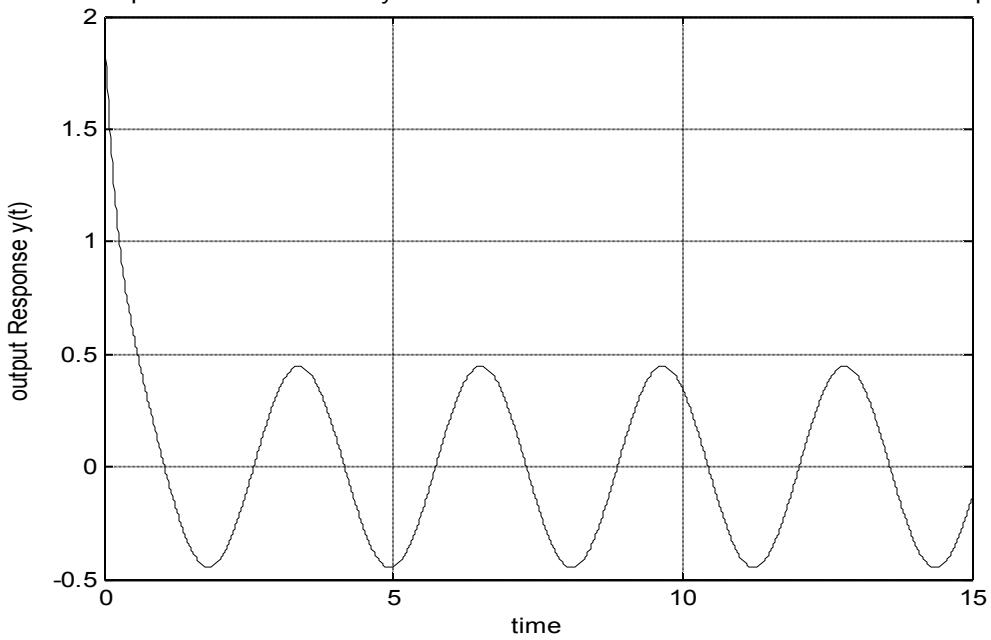
Thus, ZIR and ZSR for first order and second order differential equations are verified using matlab.

SAMPLE OUTPUT

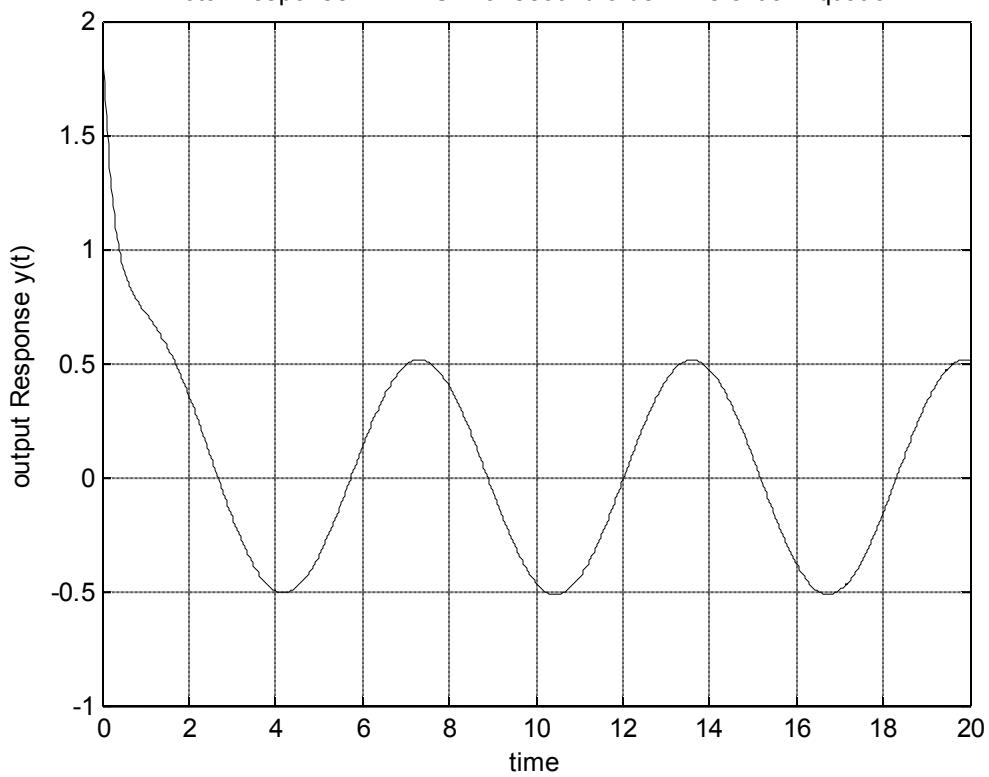
total response=ZIR+ZSR:MATLAB Result verification for First order Differential equation



Total Response=ZIR+ZSR:Analytical Result Verification for first order Differential Equation



Total Response=ZIR+ZSR:for second order Differential Equation



Total Response=ZIR+ZSR:Analytical Result Verification for second order Differential Equation

