Experiment No. : 1

To Study and Implement Convolution and Correlation of Discrete Time signals

<u>NAME OF STUDENT</u>: Vedant Pandey

<u>BATCH</u>: B3

<u>ROLL.NO</u>: 19ET1049

Experiment No. 1

Vedant Pandey

19ETI049 B3

Experiment No. 1

Aim: To study and implement linear convolution, circular convolution and corelation of discrete time signals.

Software : GNU Active Software

Theory: i) Convolution

Convolution is an integral concentration of two signals. It has many applications in numerous areas of signal processing. The most popular applications in the determination of the output signal of a linear time - invarient system by convolving the input signal

ii) Linear Convolution

The output signal from a linear system is equal to the input signal convolved with systems impulse response. In discrete time, convolution of two signal is flipped and shifted. The response y[n] of an LTI systems for any arbitratry input x[n] is given by convolution of impulse response h[n] of the system and the arbitrary input x[n]

$$y[n] = x[n] * h[n] \Rightarrow \sum_{k=0}^{\infty} x[n]\, h[n-k]$$

$$= \sum_{k=0}^{\infty} h[k]\, x[n-k]$$

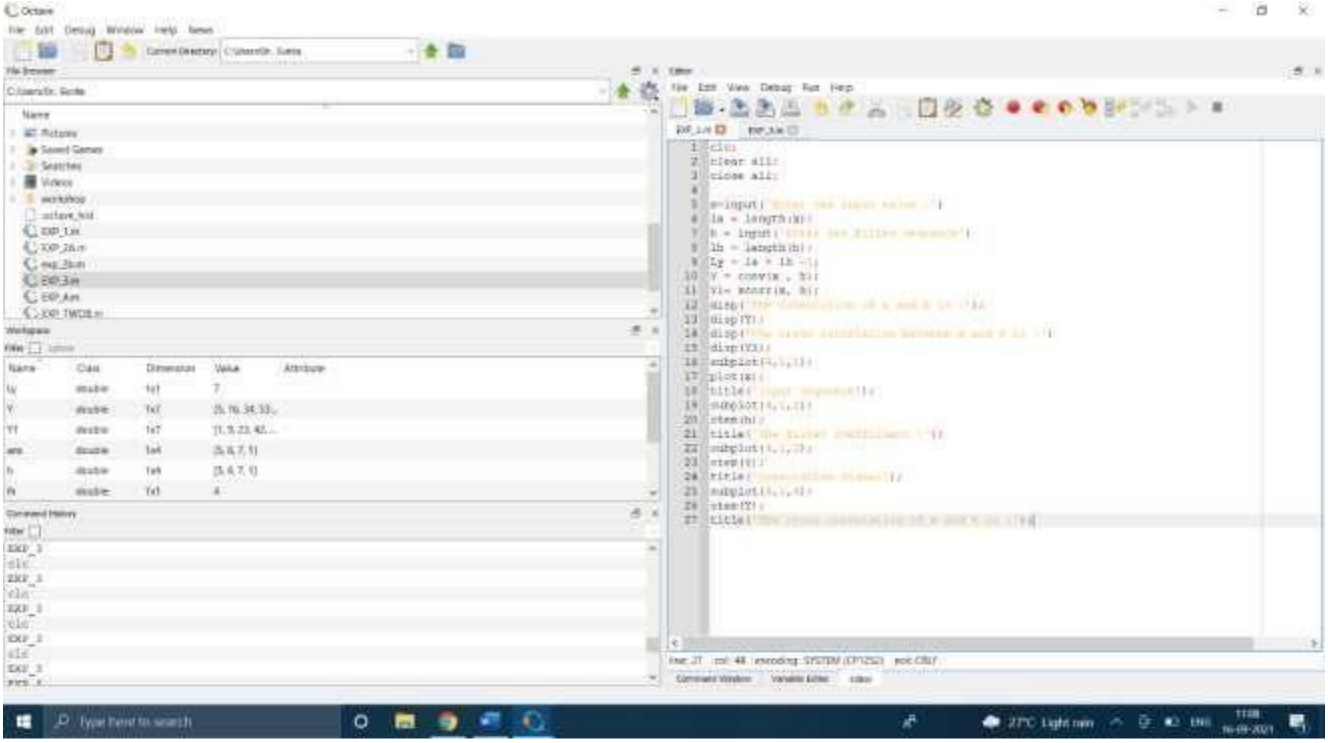
Conclusion : Program results are verified by manual calculation. The plot for corelation and convolution are developed are developed in octave. For convolution, conv command and for correlation xcore command is used.

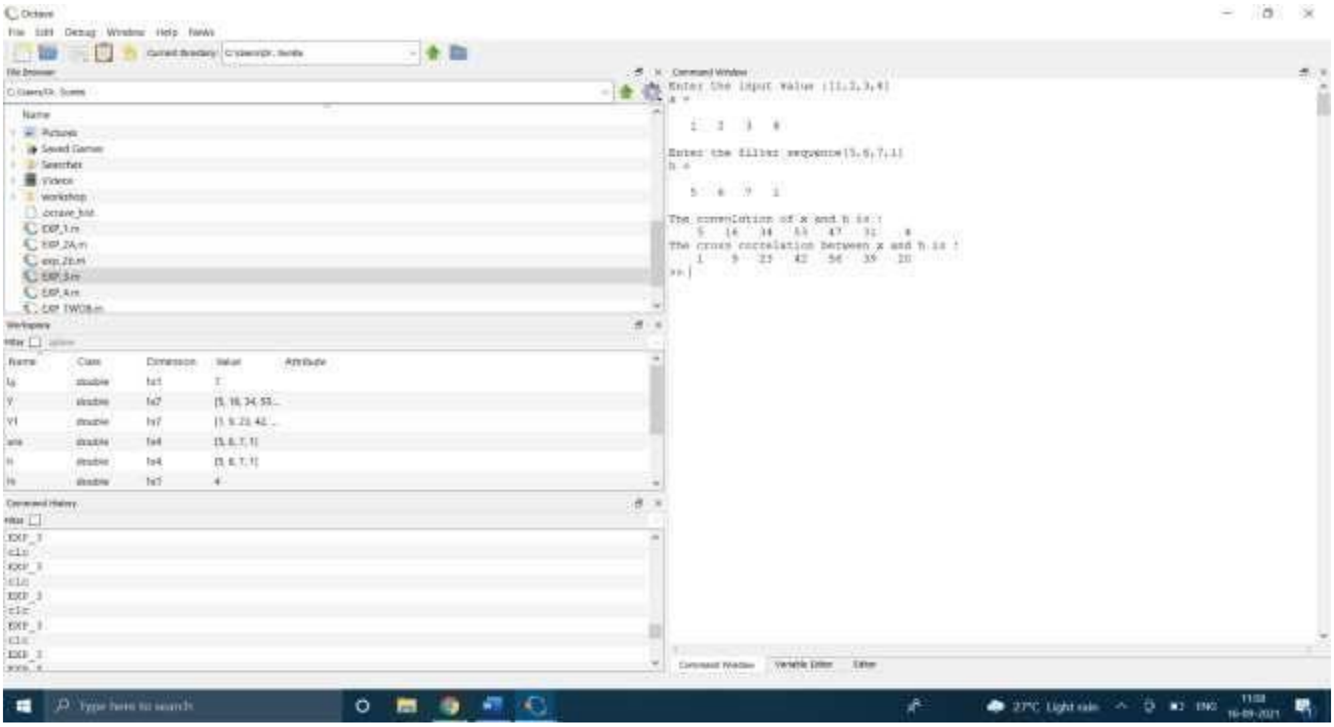
6 Simulation Result:

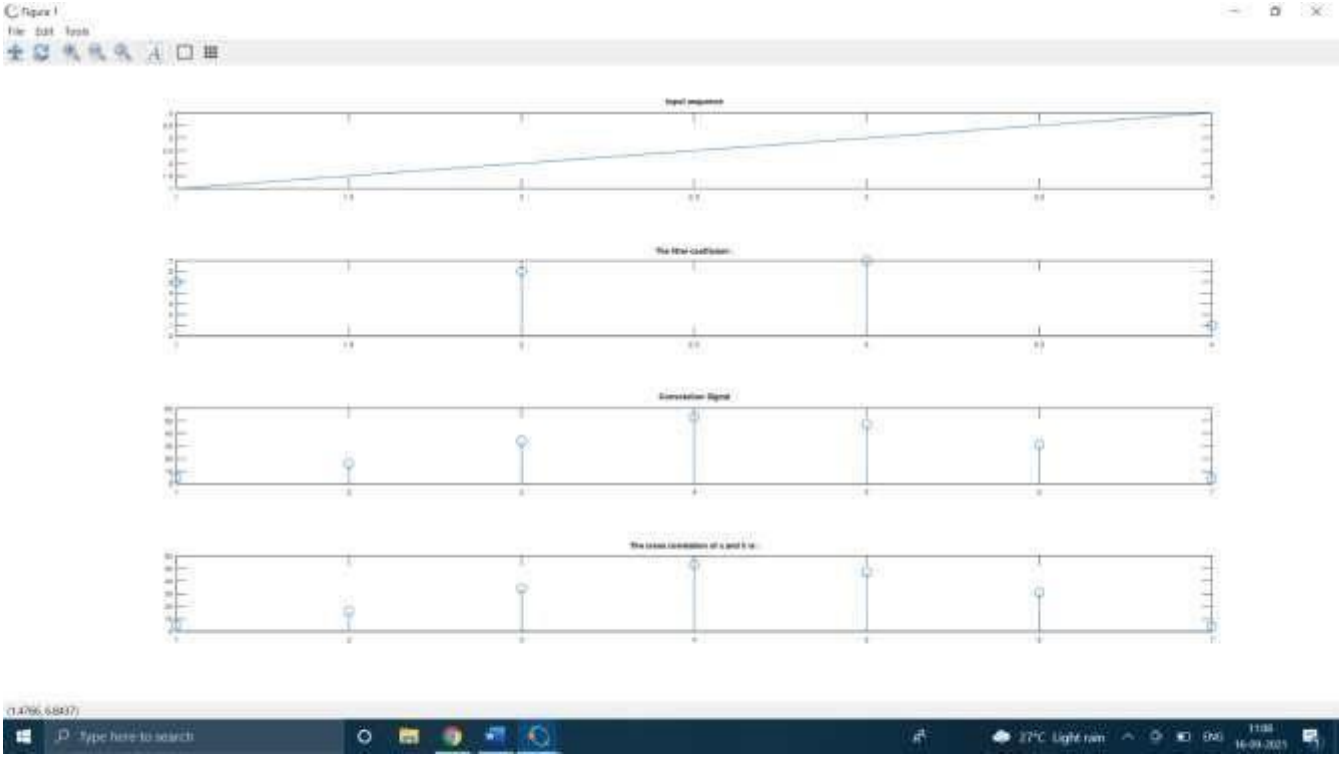
Code:

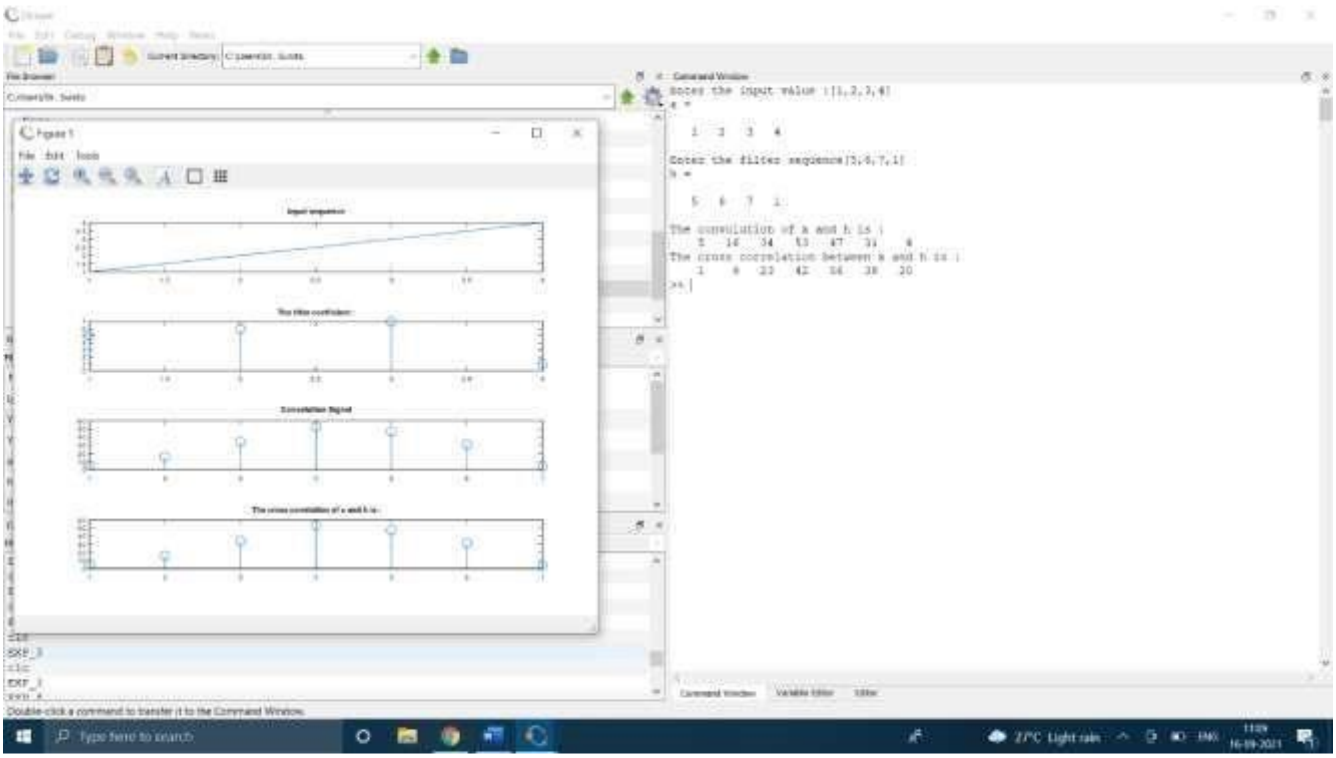
```
clc; clear
all; close
all;

x=input('Enter the input value :') lx
= length(x); h = input('Enter the
filter sequence') lh = length(h); Ly
= lx + lh -1;
Y = conv(x , h); Y1= xcorr(x, h); disp('The
convolution of x and h is :'); disp(Y); disp('The
cross correlation between x and h is :')
disp(Y1); subplot(4,1,1); plot(x); title('input
sequence'); subplot(4,1,2); stem(h); title('The
filter coefficient :'); subplot(4,1,3); stem(Y);
title('Convolution Signal'); subplot(4,1,4);
stem(Y);
title('The cross correlation of x and h is :');
```

Simulation on Editor, Command Window and on Figure Plotting:









Enter the input value :[1,2,3,4]
x =
1 2 3 4
Enter the filter sequence[5,6,7,1]
h =
5 6 7 1
The convolution of x and h is :
5 16 34 53 47 31 4
The cross correlation between x and h is :
1 9 23 42 56 38 20

6. QUIZ / Viva Questions:

1. What is convolution?
2. Explain linear convolution with the help of example.
3. What is the significance of zero padding?
4. What is the length of linear and circular convolutions if the two sequences are having the length n1 and n2?
5. Which library function used to execute convolution in MATLAB?
6. Explain circular convolution with example.
7. Explain library functions plot, subplot, grid and stem.
8. Can we calculate convolution using DTFT?
9. What is the difference between the arithmetic operators * and .* ?
10. What is cross correlation?
11. Which is better auto or cross correlation?

7. References:

1. R.E. Crochiere and A.V. Oppenheim. Analysis of linear digital networks. *Proc. IEEE*, 62:581–595, April 1975
2. E.S. Gopi - Algorithm Collections for Digital Signal Processing Applications usingMatlab
3. Vinay K. Ingle & John Proakis - Digital Signal Processing A MATLAB based Approach
4. Sanjit K. Mitra - Digital Signal Processing – Computer Based Approach
5. Digital Signal Processing includes MATLAB programs by S. Salivahanan, A. Vallavaraj

Experiment No. : 2A

To study discrete Fourier transform of a discrete-time signal.

NAME OF STUDENT: Vedant Pandey

BATCH: B3

ROLL.NO: 19ET1049

Experiment No. 2A

Vedant Pandey

19ET1049 B3

Experiment 2A

Aim: To study discrete Fourier Transform of discrete signals

Theory:

i) The discrete time fourier transform (DTFT) $x(e^{j\omega})$ of a sequence $x(n)$ is defined by

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

In general $x(e^{j\omega})$ is a complex function of the real variable and can be written as

$$x(e^{j\omega}) = X_{re}(e^{j\omega}) + j x_{im}(e^{j\omega})$$

where $x_{re}(e^{j\omega})$ & $x_{im}(e^{j\omega})$ are respectively the real and imaginary parts of $x(e^{j\omega})$

Alternately $X(e^{j\omega}) = |X(e^{j\omega})| e^{j\theta(\omega)}$

The quantity $|X(e^{j\omega})|$ is called magnitude function and $\theta(\omega)$ is called the phase function, with both being real functions of ω. In many applications, the fourier transfom is called the fourier spectrum

For a real sequence $x[n]$, the real part $x_{re}(e^{j\omega})$ of its DTFT and the magnitude function $|X(e^{j\omega})|$ are even functions of ω, whereas the imaginary part $x_{im}(e^{j\omega})$ and eve phase function θ (or as odd function of ω)

Conclusion: The discrete Fourier transform can be used to obtain the frequency domain contents of the signal. The magnitude and phase of the DFT is studied in this experiment

1. Algorithm:

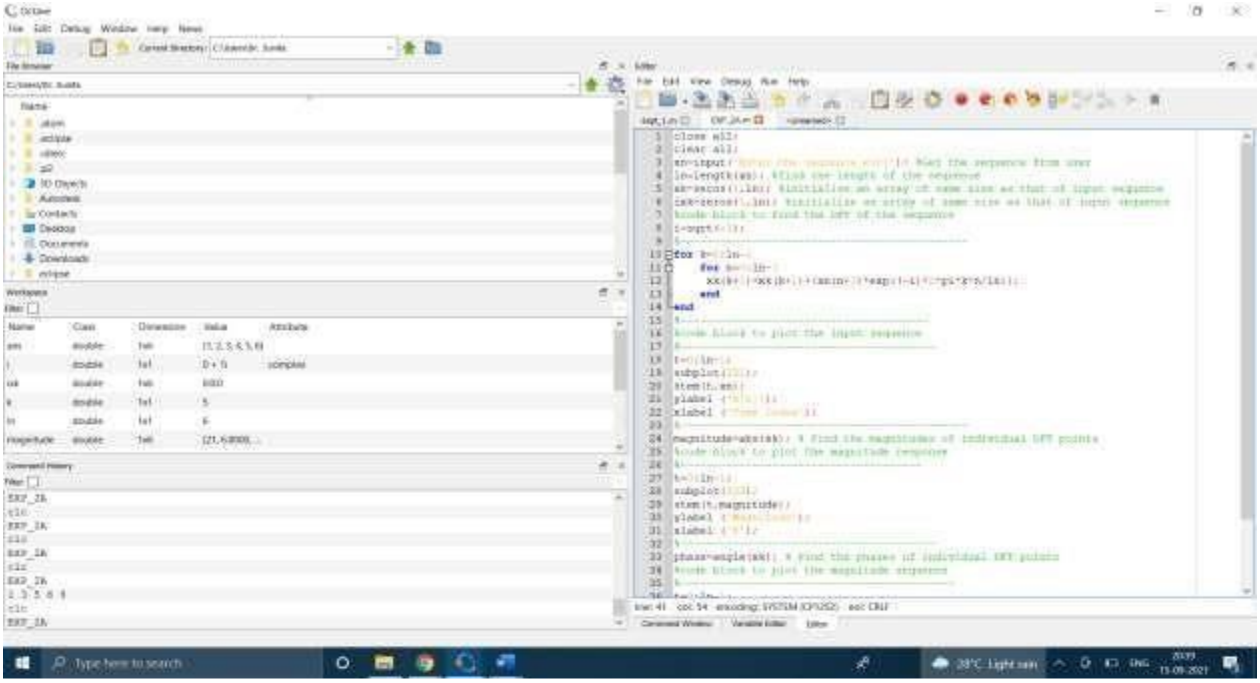
1. Enter the input sequence of discrete signal whose magnitude & phase (response) has to be plot.
2. Enter the length of the sequence.
3. Plot the input sequence.
4. Using the MATLAB command plot the magnitude response of given signal.
5. Using the MATLAB command plot the phase response of given signal.

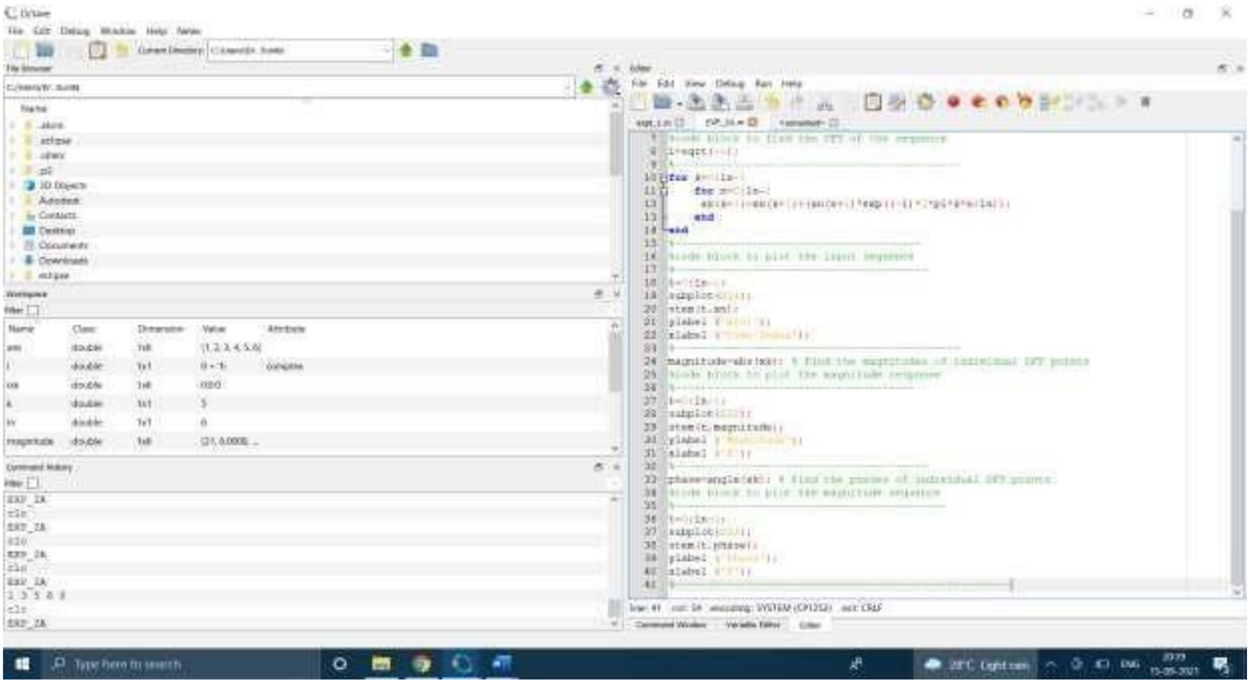
1. Simulation Result:

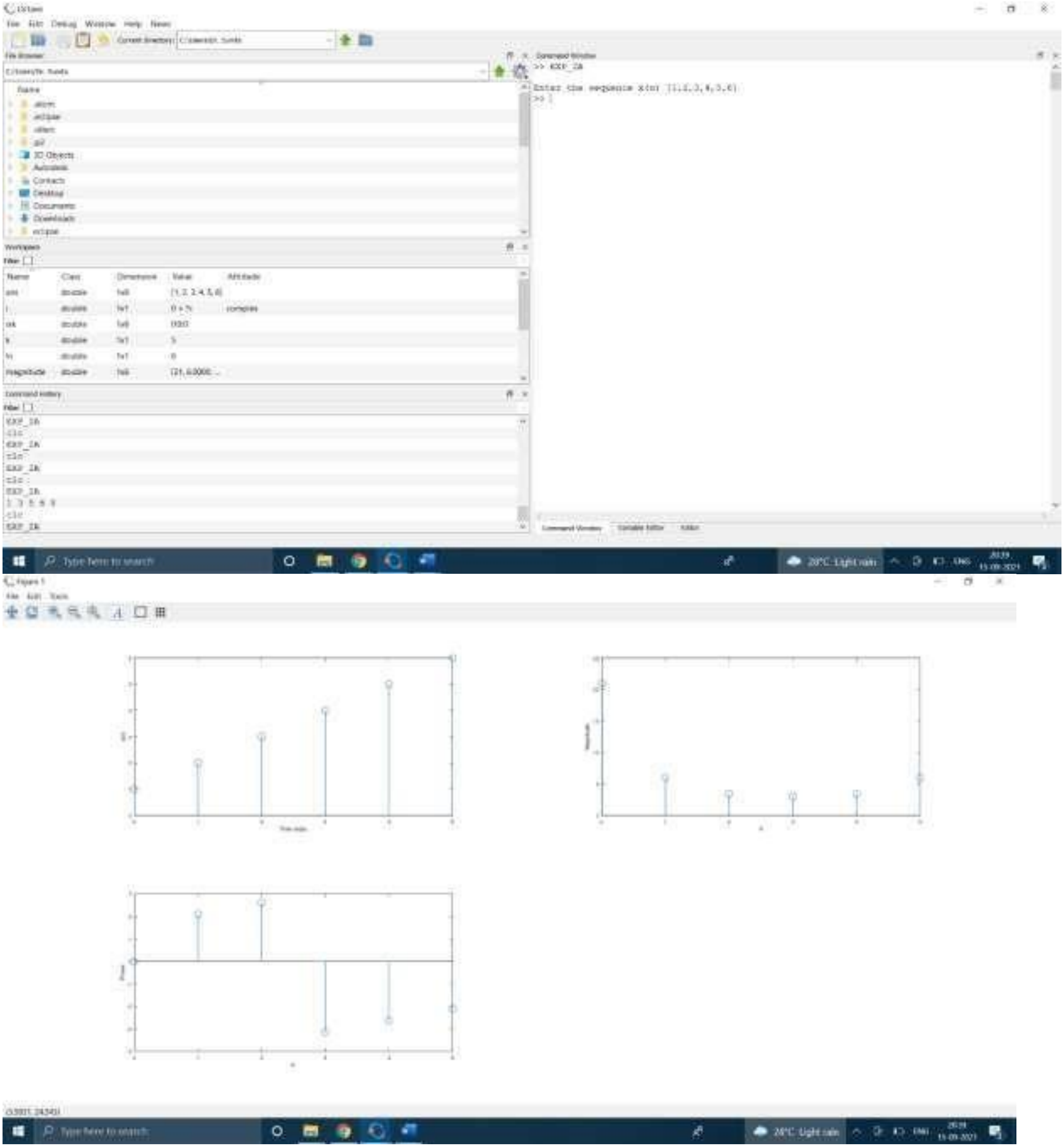
```
%program to find the DFT of a sequence
close all;clear all;
xn=input('Enter the sequence x(n)'); %Get the sequence
from user
ln=length(xn); %find the length of the sequence
xk=zeros(1,ln); %initialize an array of same size as
that of input sequence
ixk=zeros(1,ln); %initialize an array of same size as
that of input sequence

%code block to find the DFT of the sequence

i=sqrt(-1);
%------------------------------------------------------------
for k=0:ln-1     for n=0:ln-1
     xk(k+1)=xk(k+1)+(xn(n+1)*exp((-i)*2*pi*k*n/ln));
end
end
%------------------------------------------------------
%code block to plot the input sequence
%-------------------------------------------------------
t=0:ln-1; subplot(221); stem(t,xn);
ylabel ('x[n]'); xlabel ('Time Index');
```

```
%-----------------------------------------------------------


magnitude=abs(xk); % Find the magnitudes of individual
DFT points
%code block to plot the magnitude response
%--------------------------------------------------
t=0:ln-1; subplot(222); stem(t,magnitude);
ylabel ('Magnitude'); xlabel ('K');
%-----------------------------------------------------
phase=angle(xk); % Find the phases of individual DFT
points

%code block to plot the magnitude sequence
%--------------------------------------------------------
t=0:ln-1; subplot(223); stem(t,phase);
ylabel ('Phase'); xlabel ('K');
%---------------------------------------------------------------------
```

Result:







‘

The computed DFT is analyzed for it’s magnitude and phase. Figure 2.1 shows the signal magnitude and phase of the DFT.

.

1. QUIZ / Viva Questions:

1. What is DFT?
2. Define the Discrete Time Fourier Transform.
3. Discuss the steps involved in calculation of DFT.
4. In spectral analysis which command is used to find out magnitude?
5. In spectral analysis which command is used to find out phase?

Experiment No. : 2B
Study of Fast Fourier Transform (FFT)

<u>NAME OF STUDENT</u>: Vedant Pandey

<u>BATCH</u>: B3

<u>ROLL NO</u>: 19ET1049

Experiment No. 2B

1. Aim: To develop program for computing Fast Fourier Transform (FFT).

1. What you will learn by performing this experiment?

- To understand how to do efficient calculation than DFT algorithm.
- To learn what is Decimation In Time – Fast Fourier Transform (DIT-FFT).
- To learn what is Decimation In Frequency – Fast Fourier Transform (DIF-FFT).

1. **Software Required:** MATLAB/ OCTAVE/ PYTHON

Vedant Pandey

19ET1049 B_3

Experiment 2B

Aim: To develop program for computing Fast Fourier Transform

Theory: i) Efficient computation of the DFT

The problem:

Given signal sample : $x[n] \ldots\ldots x[N-1]$

develop a procedure to compute

$$x[n] = \sum_{N=0}^{N-1} x[n]\, W_8^{kn}$$

For $K = 0$ $N-1$ where $W_N = e^{\frac{-j2\pi}{N}}$

ii) Radix - 2 FFT

Useful when N is a power of 2, $N = 2^k$ for integer k. It is called the radix, which comes from the Latin Word meaning 'a root' and is the no. of stages When N is a power of $r = 2$, this is called radix - 2 and the natural "divide and conquer approach" is to split the sequence into two.

Conclusion: The discrete fourier transform can be computed using FFT. The index K is related to the frequency f of the analog signal as $f = xfs/N$. The magnitude and phase of the DFT is ~~checked~~ obtained using FFT. The FFT requires lesser number of computations then direct DFT computation and hence it is faster.

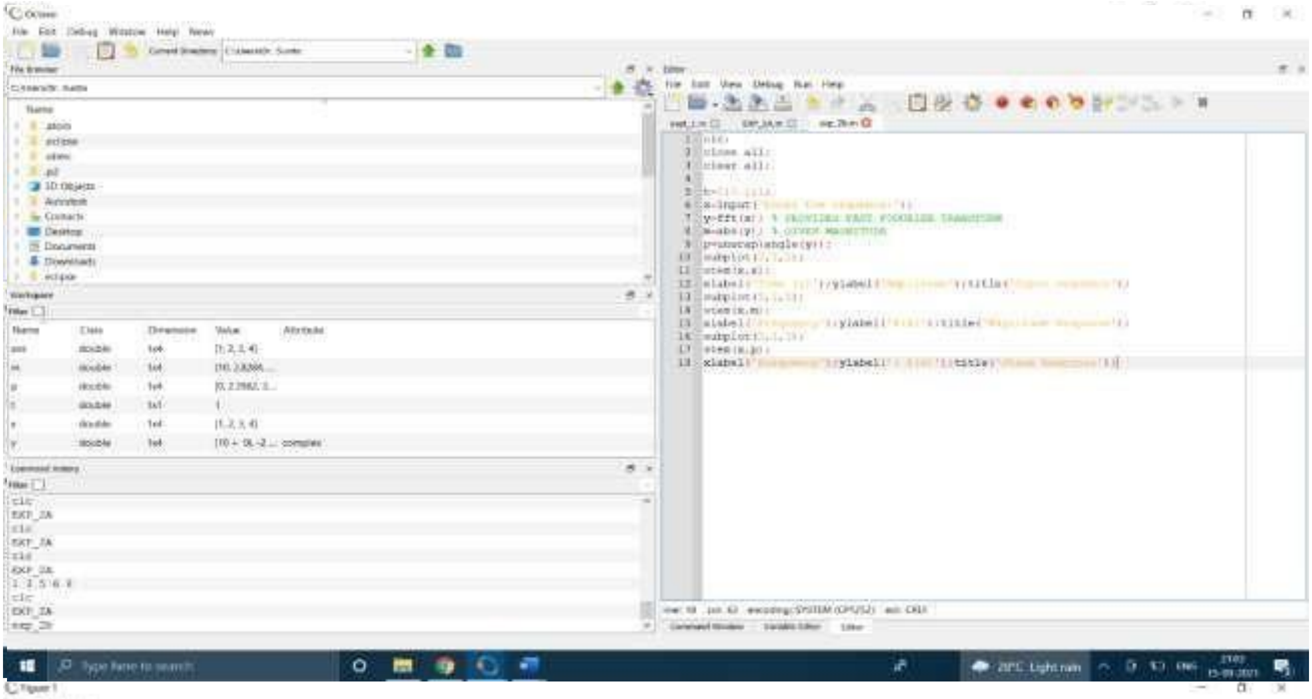
1. Algorithm:

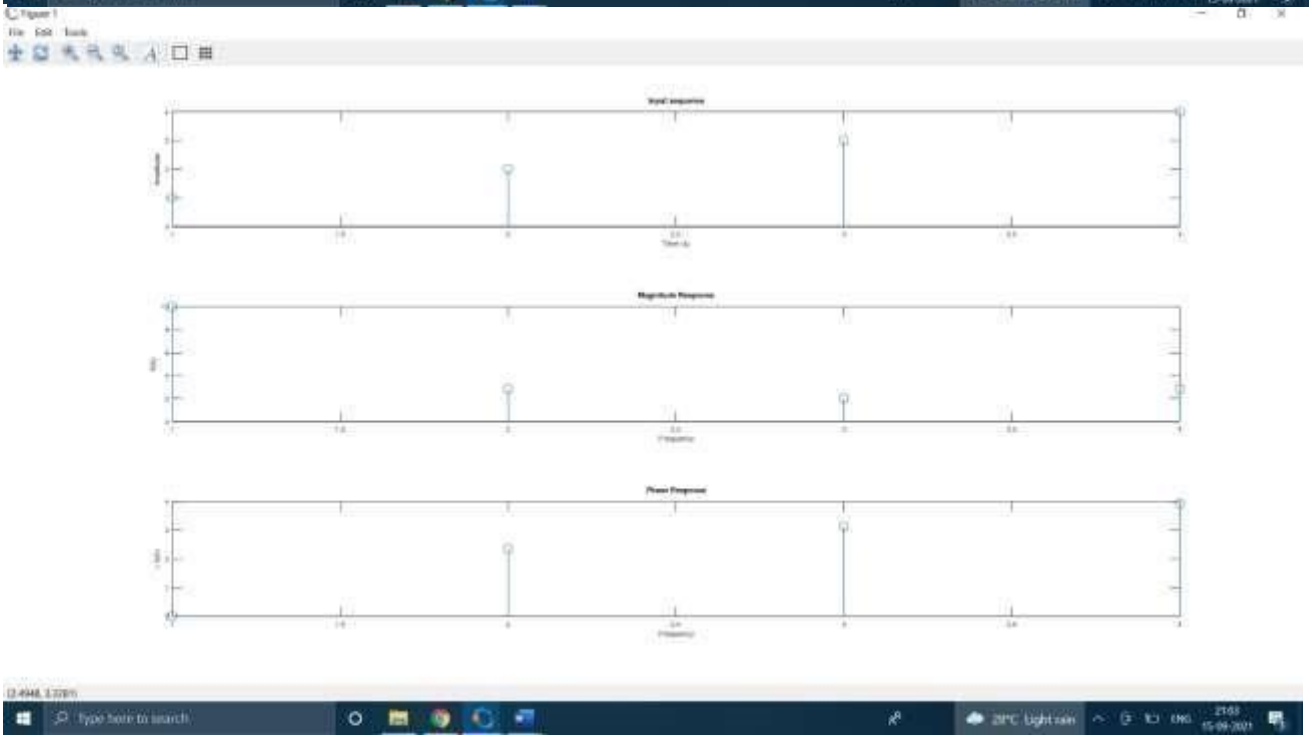
1. Enter the input sequence of discrete signal.
2. Find the length of input sequence.
3. If length is not equal to in terms of power of 2 then convert it into next higher power of 2.
4. Compute the DFT of the zero padded signal using fft() command.
5. Display the Magnitude and phase of the computed DFT.

1. Simulation Result:

```
clc; close all; clear all;

t=1:0.1:1;
x=input('Enter the sequence:'); y=fft(x); % PROVIDES FAST
FOOURIER TRANSFORM m=abs(y); % GIVES MAGNITUDE
p=unwrap(angle(y));  subplot(3,1,1); stem(x,x); xlabel('Time
(s)');ylabel('Amplitude');title('Input sequence');
subplot(3,1,2); stem(x,m);
xlabel('Frequency');ylabel('X(k)');title('Magnitude
Response'); subplot(3,1,3); stem(x,p);
xlabel('Frequency');ylabel('< X(k)');title('Phase
Response');
```

RESULTS:


clc;
close all;
clear all;



Experiment No. : 3

Minimum phase, Maximum phase and Mixed phase

<u>NAME OF STUDENT</u>: Vedant Pandey

<u>BATCH</u>: B3

<u>ROLL NO</u>: 19ET1049

Experiment No. 3

Vedant Pandey

19E1301? B3

Experiment 8

Aim: To identify whether given system is minimum phase, maximum phase or mixed phase

Theory: For FIR and IIR filters

1. When all zeroes ~~and~~ are inside the unit circle then its called a minimum phase
2. When all zeroes are outside the unit circle then its called a maximum phase
3. When some zeroes are inside the unit circle and remaining are outside the unit circle, then its called mix phase.

If H(z) is having minimum phase. Then its inverse system $H^{-1}(z)$ has minimum phase ~~difference~~ But mixed phase and minimum phase systems results in unstable inverse systems.

i) a) Since all zeroes are inside the unit circle, its called minimum ~~f~~ phasee.

b) All poles are not inside then given systems is not stable

c) Its an all pass filter

ii) a) Since all zeroes were outside circle is called maximum phase

b) All poles inside the circle is stable

c) Since some zeroes are outside one circle and same are outside is called mixed ~~plase~~ phase

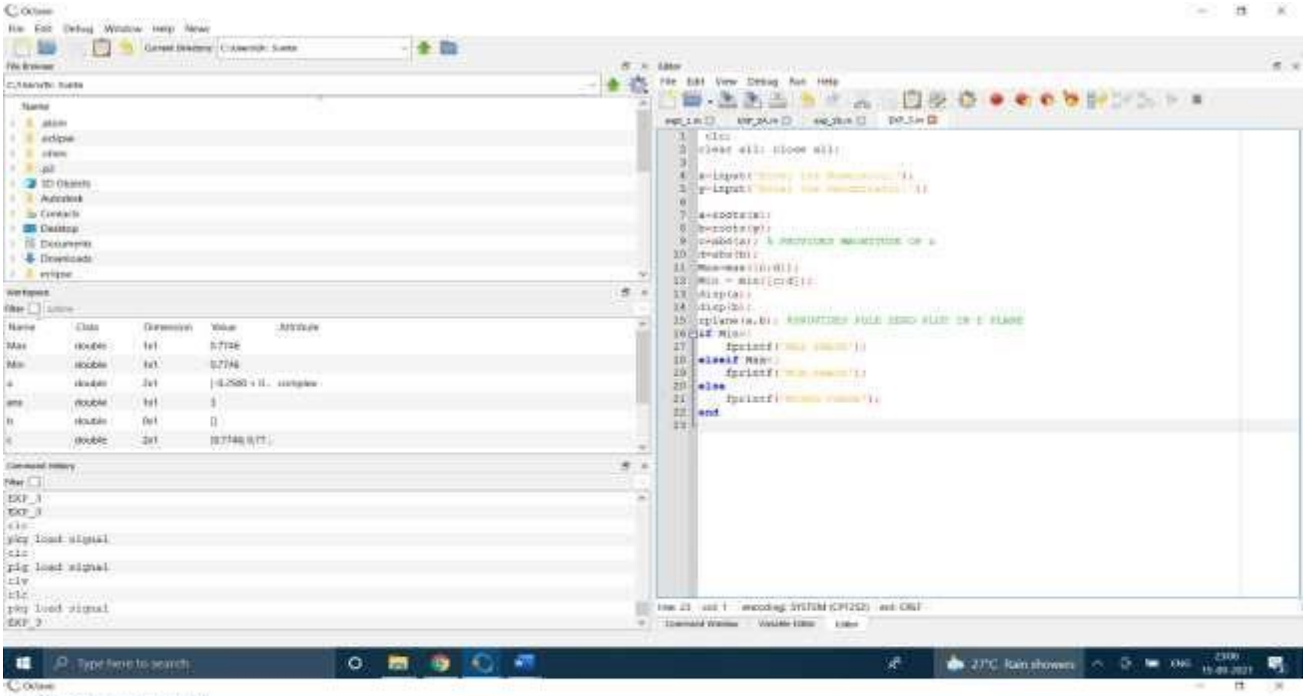
FOR EDUCATIONAL USE

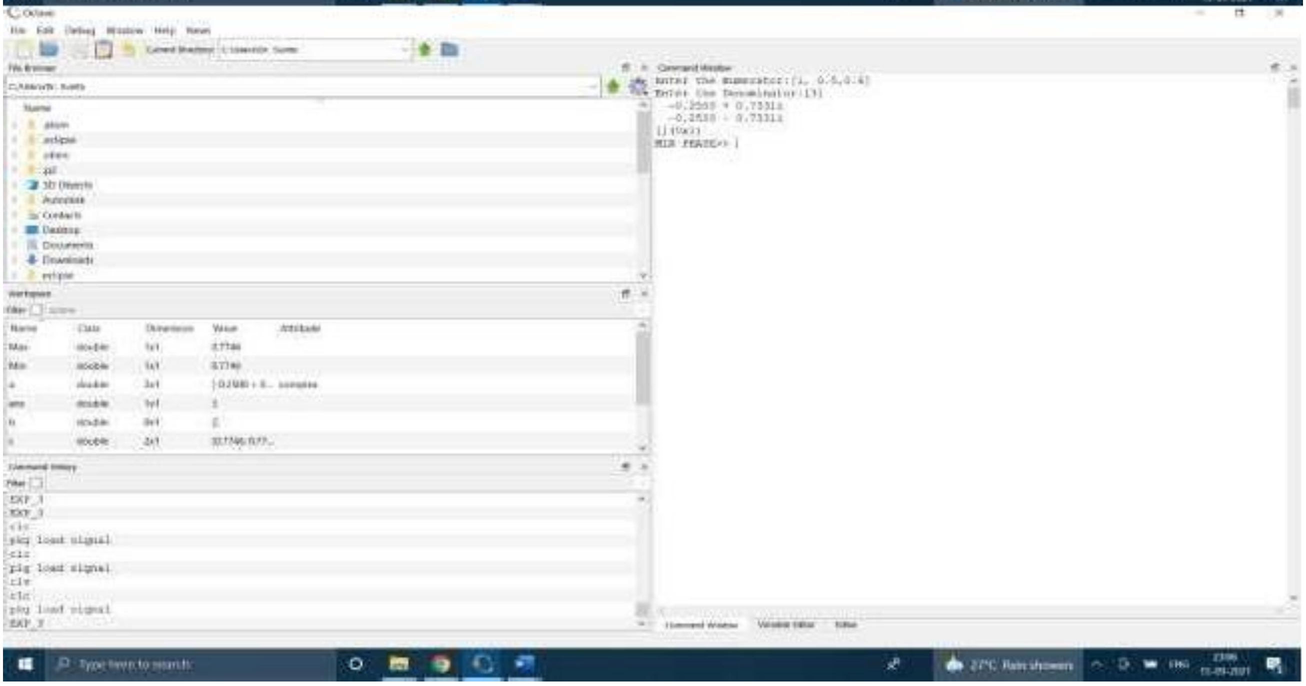
1. **Simulation:** `clc; clear all; closeall;`

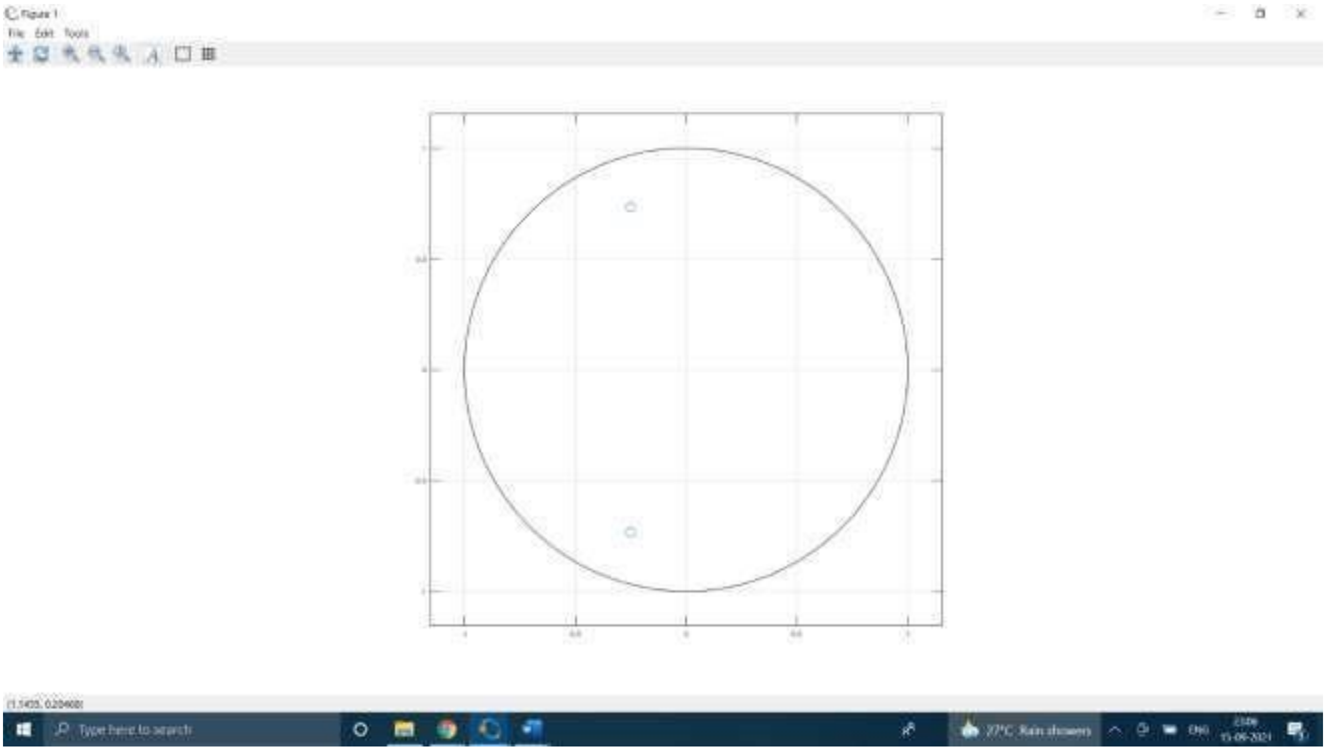
```
x=input('Enter the Numerator:');
y=input('Enter the Denominator:');

a=roots(x);b=roots(y);
c=abs(a); % PROVIDES MAGNITUDE OF a
d=abs(b); Max=max([c;d]); Min = min([c;d]);
disp(a);disp(b);zplane(a,b); %PROVIDES POLE
ZERO PLOT IN Z PLANE if Min>1
    fprintf('MAX PHASE');
elseif Max<1
    fprintf('MIN PHASE');
else     fprintf('MIXED
PHASE');
end
```

OUTPUT:

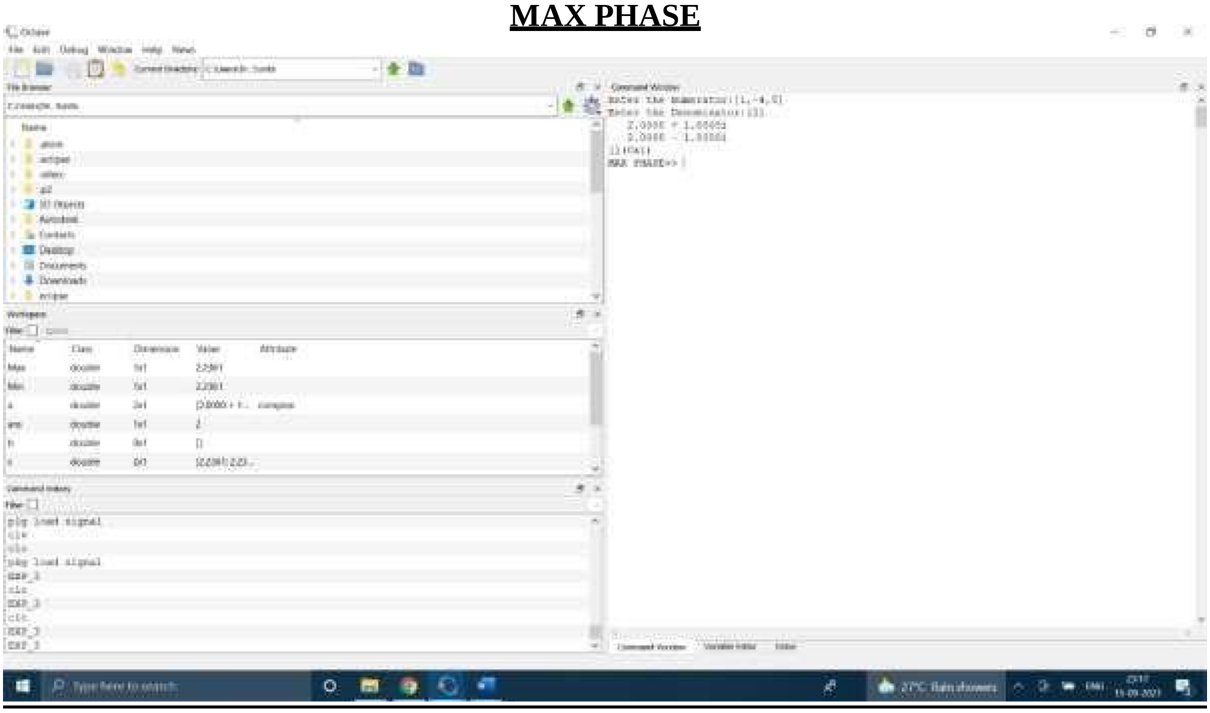
MIN PHASE

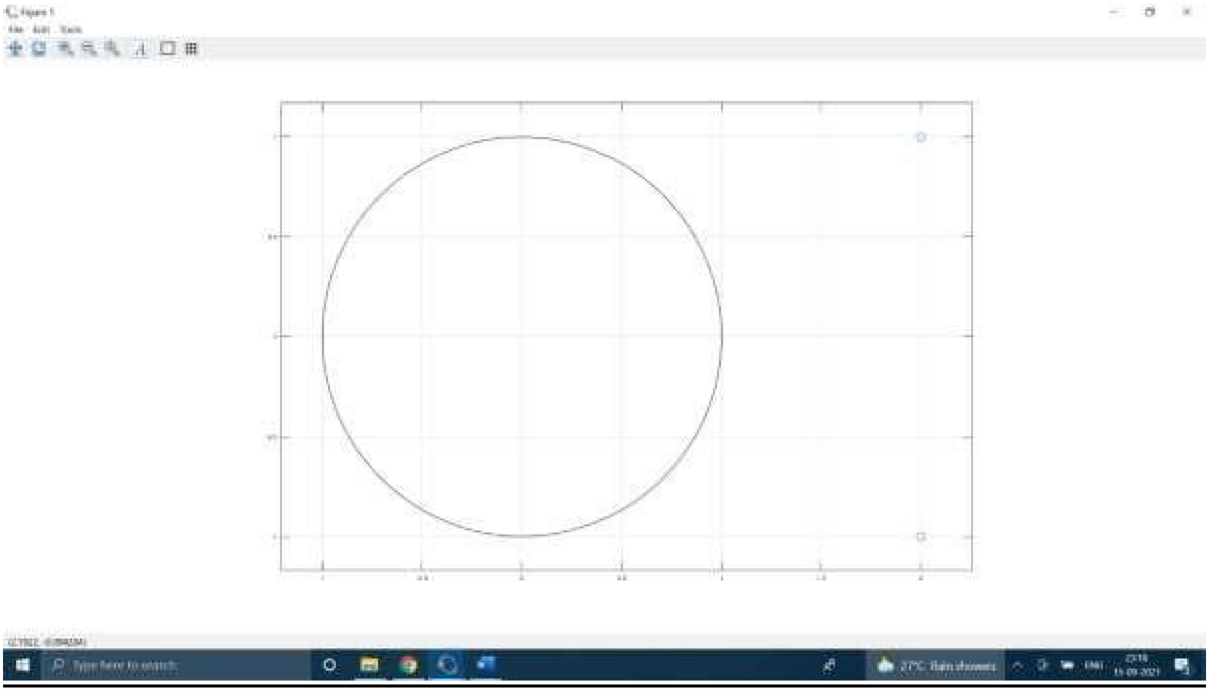




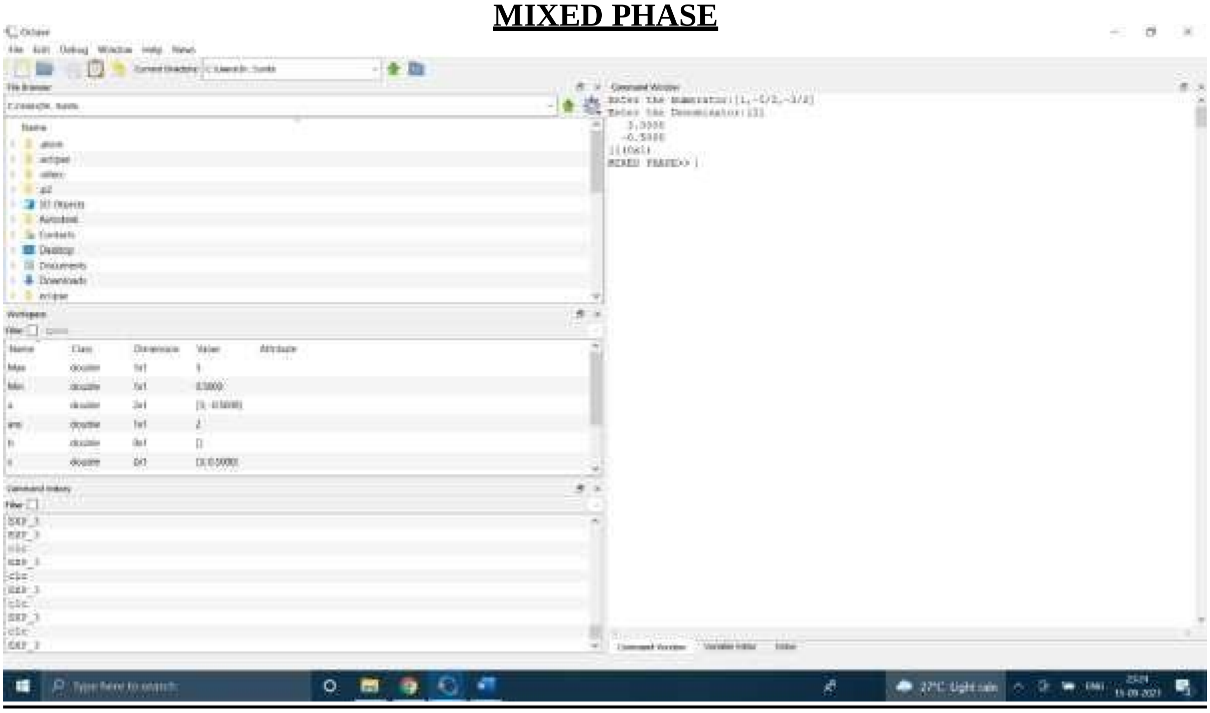


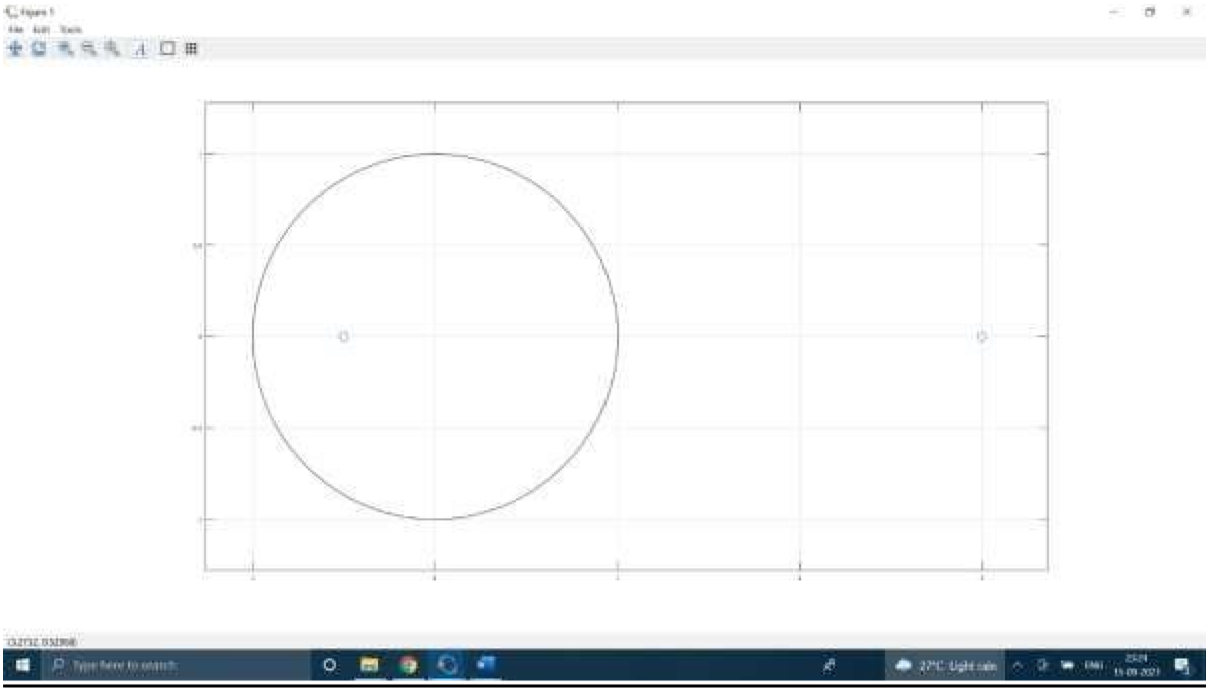
MAX PHASE





MIXED PHASE





1. QUIZ / Viva Questions:

- What is meaning of minimum phase system?
- What is meaning of maximum phase system?
- What is the need of pole zero plot?

Experiment No. : 4

To implement IIR Digital filter

NAME OF STUDENT: Vedant Pandey

BATCH: B3

ROLL NO.: 19ET1049

Experiment No. 4A

Rollno. : 19ETI049 B3

Experiment 4

Aim : To implement IIR Digital Butterworth filter Chebyshev - V

Theory : The process of deriving the transfer function G(c) whose frequency response $G(e^{i\omega})$ approximation the given frequency response specification, called digital filter design

The analog transfer function is usually one of the following types, Butterworth, Chebyshev and elliptic transfer function. The difference between these filter types can be explained by considering the analog low filter. The Butterworth low pass tranfer function. The difference between these filters types can be explained by considering the analog low pass filter. The butterworth low pass filter has maximally flat magnitude response at dc, that is, $\Omega = 0$ and a monotonically decreasing response at dc. The type 1 chebyshevs low pass filter has an equiripple magnitude response in the low pass filter. The type 2 chebyshevs low pass transfer function ha a monotonically decreasing magnitude in the step band with increasing freq. and can equiripple magnitude respons in the step band. Finally the elliptic low pass transfer function ha equiripple magnitude responses both in the pass band.

Conclusion : 1. Cutoff frequency obtained from the response is approximate equal to the designed theoritical value.

2. It can be seen that the db magnitude at R_c is 3db less than the maximum value

3. Magnitude response can be plotted with the help of octave.

Algorithm:

- Get the pass band and stop band gain from the specifications
- Get the pass band and stop band edge frequencies
- Calculate the order of the filter using MATLAB command.
- Find the filter coefficients, using MATLAB command.
- Plot the magnitude and phase response.

Program and Simulation Result:

```
clc
clear
close all
```

```
% Specifications:
% Passband 0 to 0.2pi rads, gain between 1 and 0.9
% Stopband 0.3pi to pi rads, atten 0.001 = -60 dB
%
% Compare Butterworth, Chebyshev I & II, and elliptic designs.

passband = 0.2*pi/pi; % convert to normalized frequency stopband
= 0.3*pi/pi;
passrip = -20*log10(0.9); % ripple in positive dB
stopatten = -20*log10(0.001); %stopband attenuation in positive dB

% Find order and natural frequency to meet these specs

[Nb, Wnb] = buttord(passband, stopband, passrip, stopatten); % Butterworth filter
[Nc1, Wnc1] = cheb1ord(passband, stopband, passrip, stopatten); % Cheby 1 filter

% Now use the order and frequencies just identified to design these filters
% by finding their difference equation coefficients

[Bb,Ab] = butter(Nb, Wnb);
[Bc1,Ac1] = cheby1(Nc1,passrip,Wnc1);

% Display frequency responses figure(1)
[Hb,W]= freqz(Bb,Ab); h =
plot(W/pi,20*log10(abs(Hb)));
title(['Butterworth Filter, Order =
',num2str(Nb)]) axis([0 1 -80, 5]) ylabel('Gain
(dB)')
xlabel('Normalized Frequency (\times \pi rad/sample)') grid
on

figure(2)
[Hc1,W]=   freqz(Bc1,Ac1);   h   =
plot(W/pi,20*log10(abs(Hc1)));
```

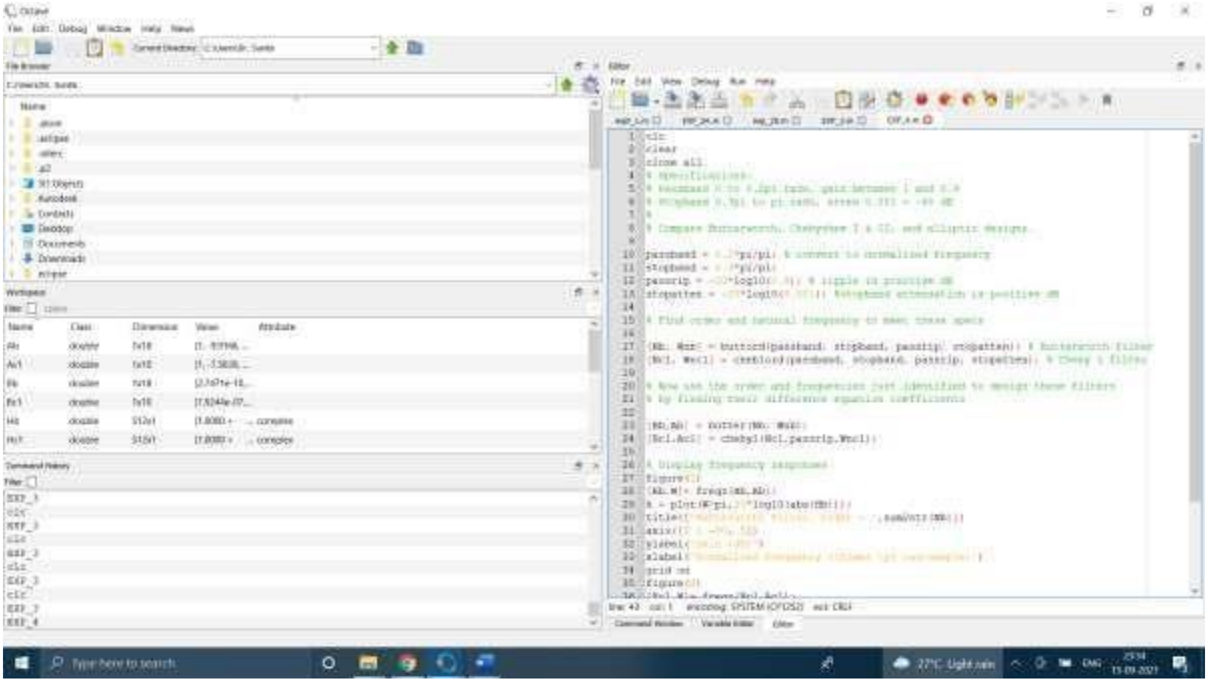
title(['Chebyshev Type 1 Filter, Order =

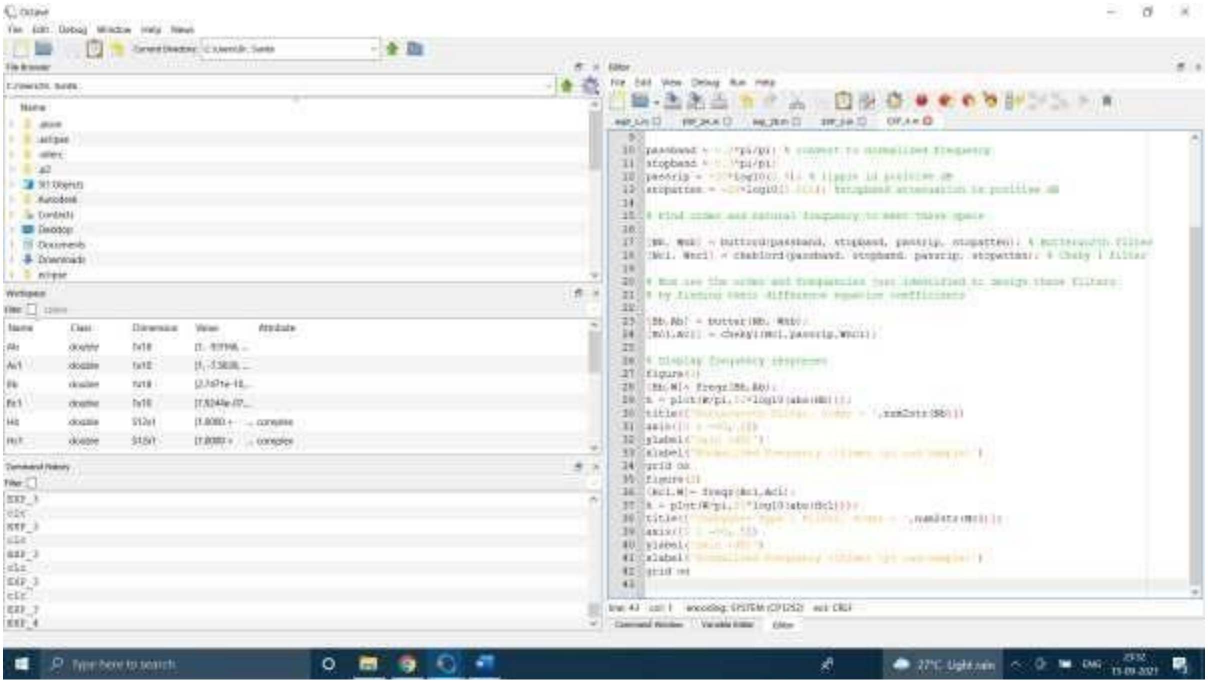
',num2str(Nc1)]) axis([0 1 -80, 5]) ylabel('Gain (dB)')

xlabel('Normalized Frequency (\times \pi rad/sample)') grid

on

OUTPUT





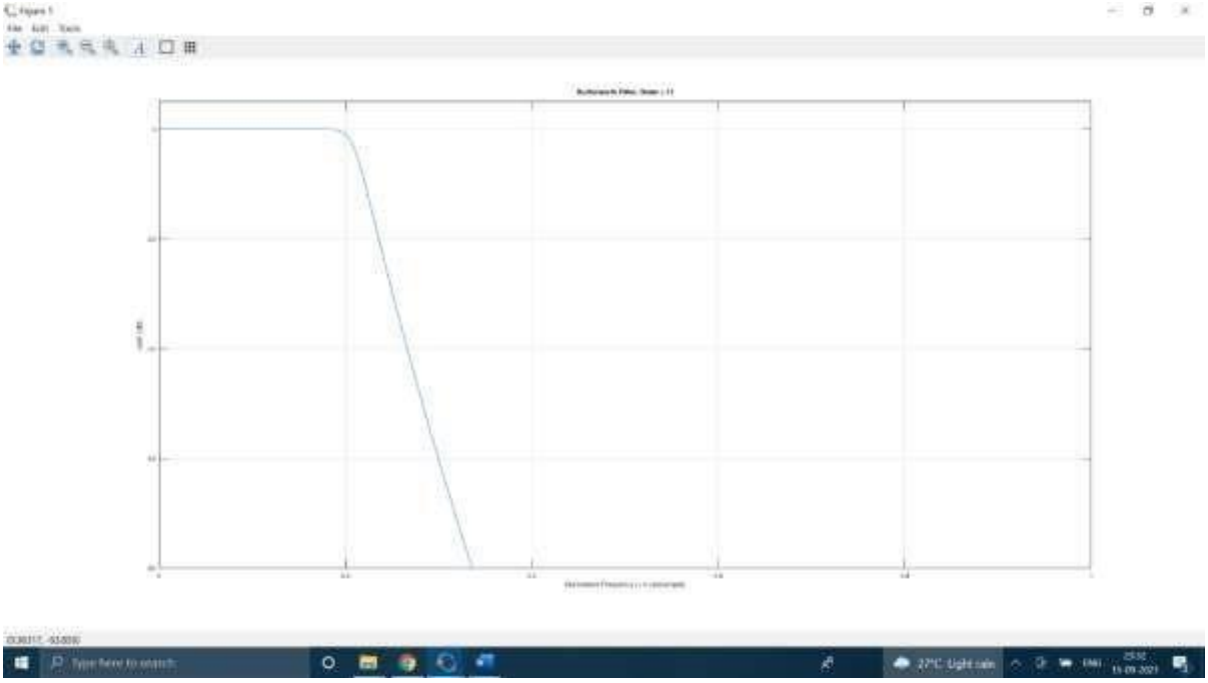


Figure 4.4: Magnitude Response for Butterworth filter

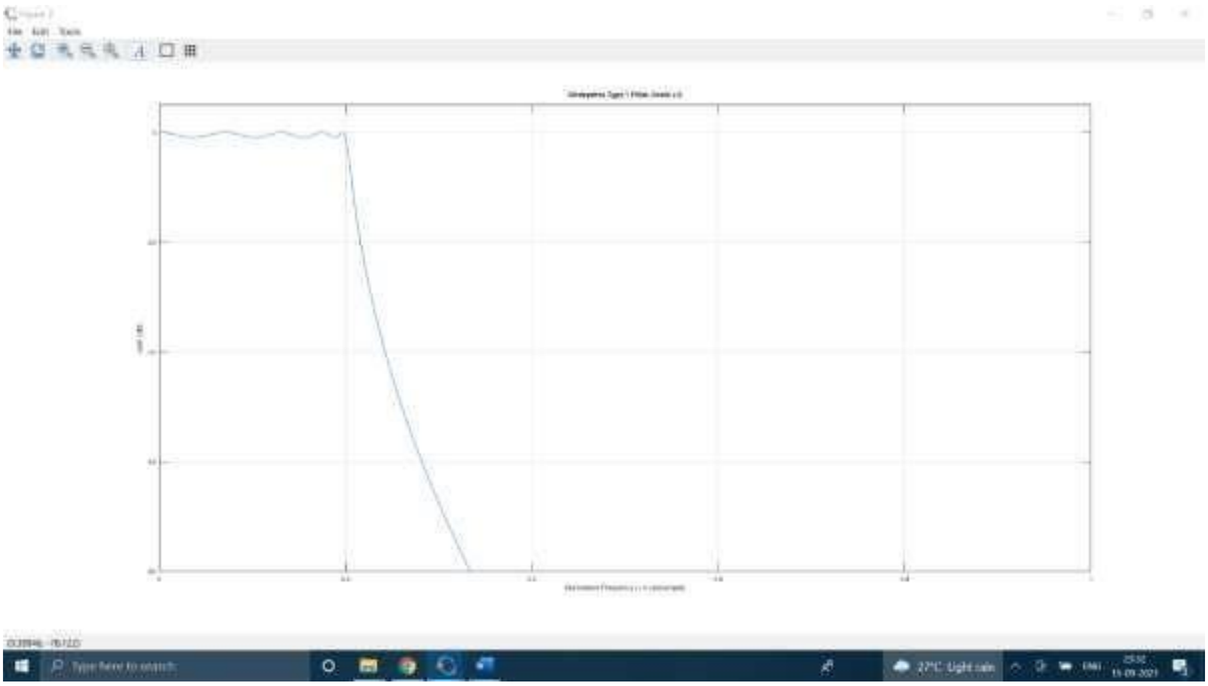
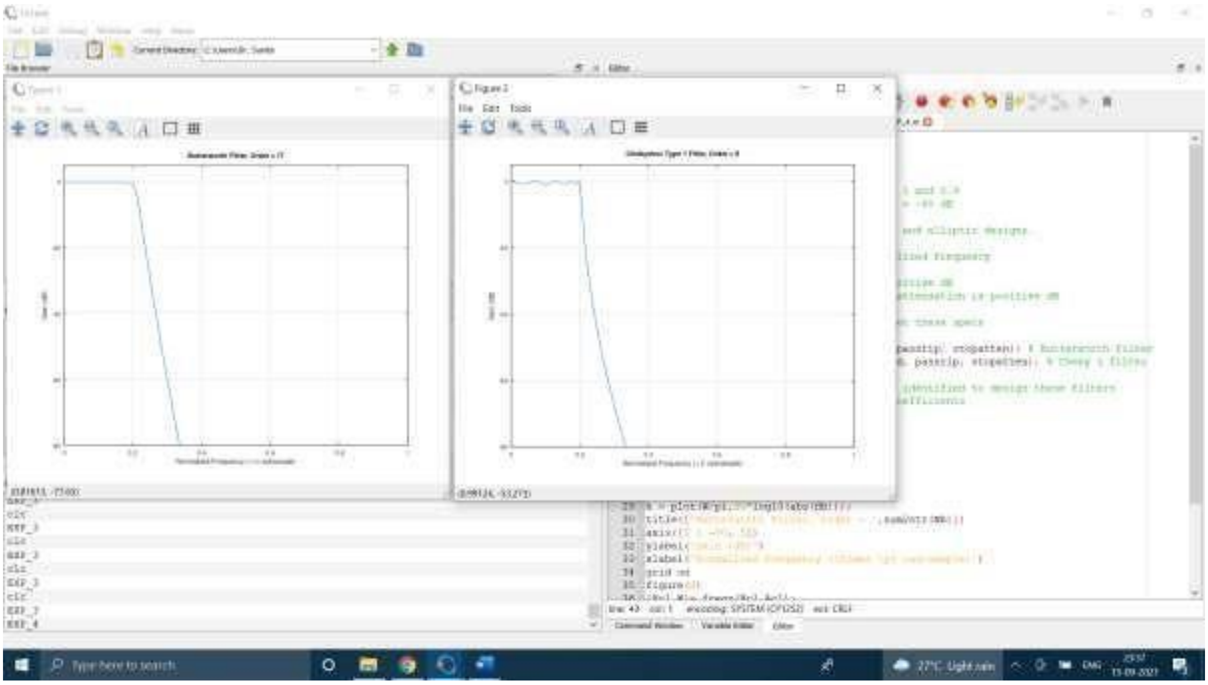


Figure 4.5: Magnitude Response for Chebyshev filter



QUIZ / Viva Questions:

1. What are the advantages of digital filters over analog filters?
2. What are methods used to convert analog to digital filter?
3. Why bilinear transformation is preferred over impulse invariant transformation?

Experiment No. : 5

To implement a FIR filter using Windowing Techniques.

<u>NAME OF STUDENT</u>: Vedant Pandey

<u>BATCH</u>: B3

<u>ROLL.NO</u>: 19ET1049

Experiment No. 5

Name : [illegible]

Roll no : [illegible]

Experiment 3

Aim: To implement FIR lowpass filter using various windows and plot the magnitude and phase response using MATLAB

Theory: The frequency of an N^{th} order causal FIR filter is obtained by ~~[illegible]~~ DTFT of the impulse response. The design of FIR filter involves finding the coefficients h(n) that results in a frequency response that satisfies a given specification. FIR filter have two ~~arbitrary~~ important advantages over IIR filter. First, they are guaranteed to be stable, even after the filter coefficients have been quantized. Second they can be easily combined to have linear phase. Because FIR are generally designed to have linear phase in the following we consider the design of linear phase filter

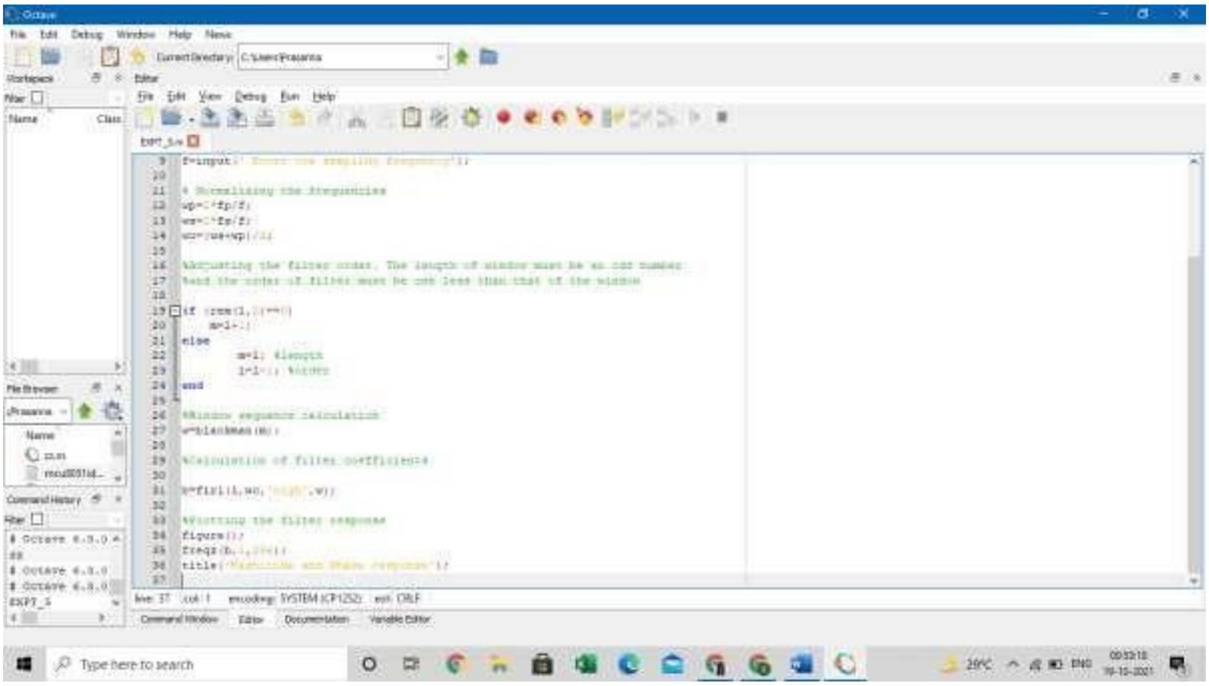
Conclusion

The result of the windowing techniques shown. We have studied the effect of different windows like blackman, hamming and rectangular window on response of a filter. Rectangular window gives larger side lobes

5. **Algorithm:**

- Get the order of the filter ● Get the cut off frequency
- use ‘ fir1 ’& ‘rectangular’ function to compute the filter coefficient
- use window functions as per the requirements
- Plot the magnitude and phase response ● Do the above steps for all windows.

6. **Simulation/Program**



Results

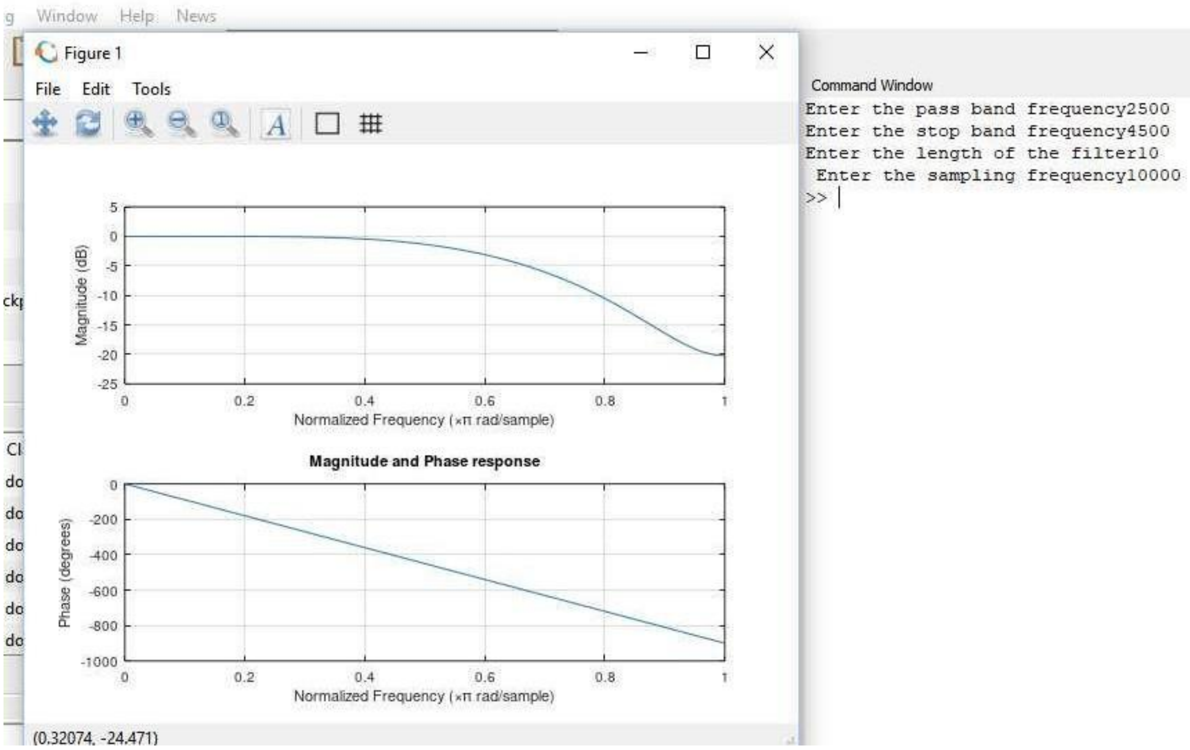


Figure 5.1: Magnitude response of LPF with length of filter M=10

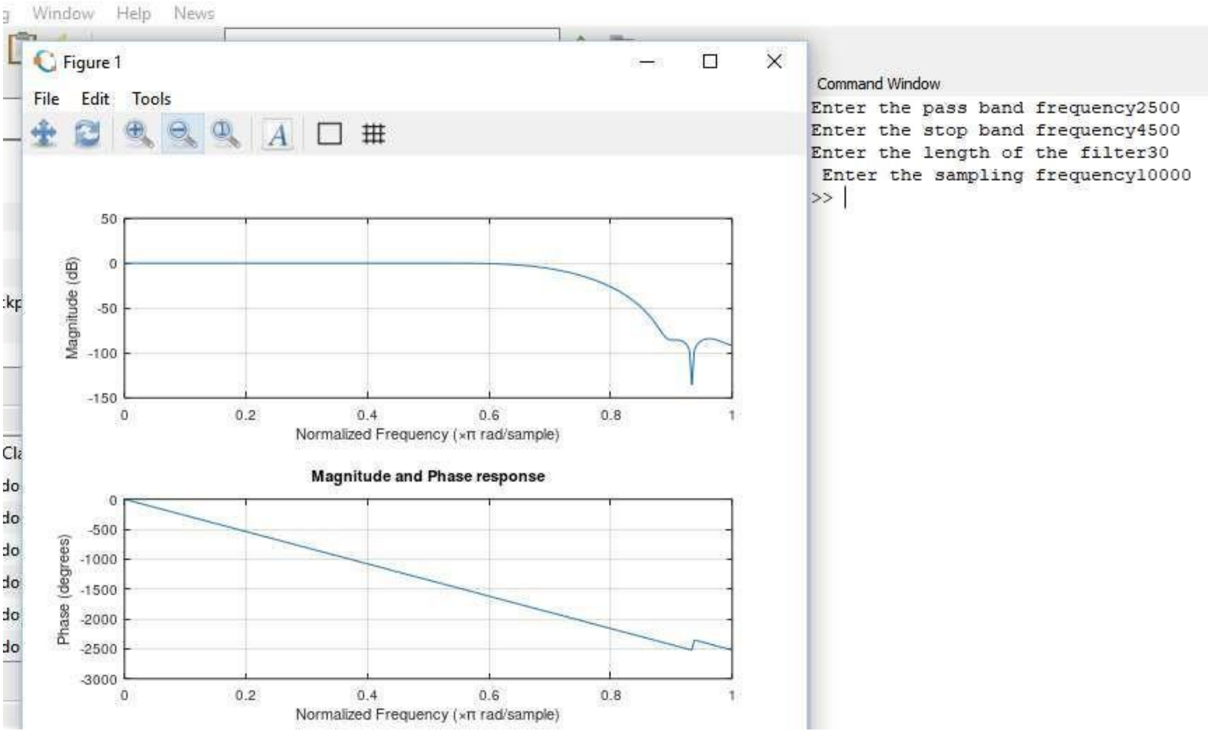


Figure 5.2: Magnitude response of LPF with length of filter M=30

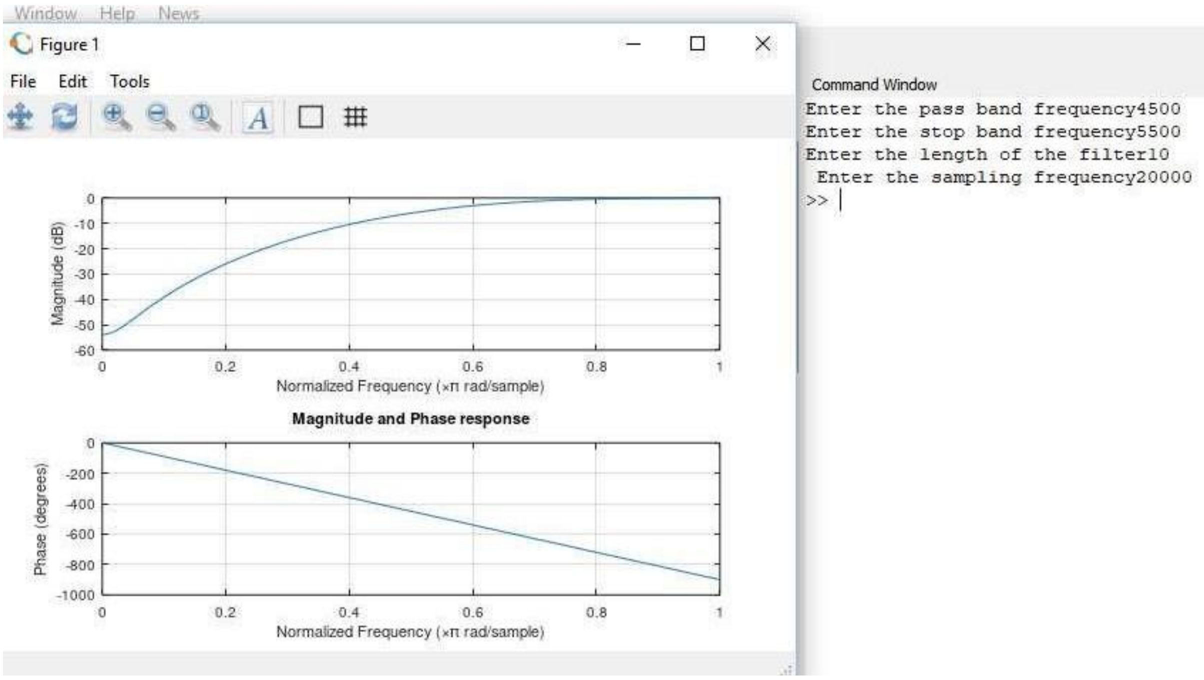


Figure 5.3: Magnitude response of HPF with length of filter M=10

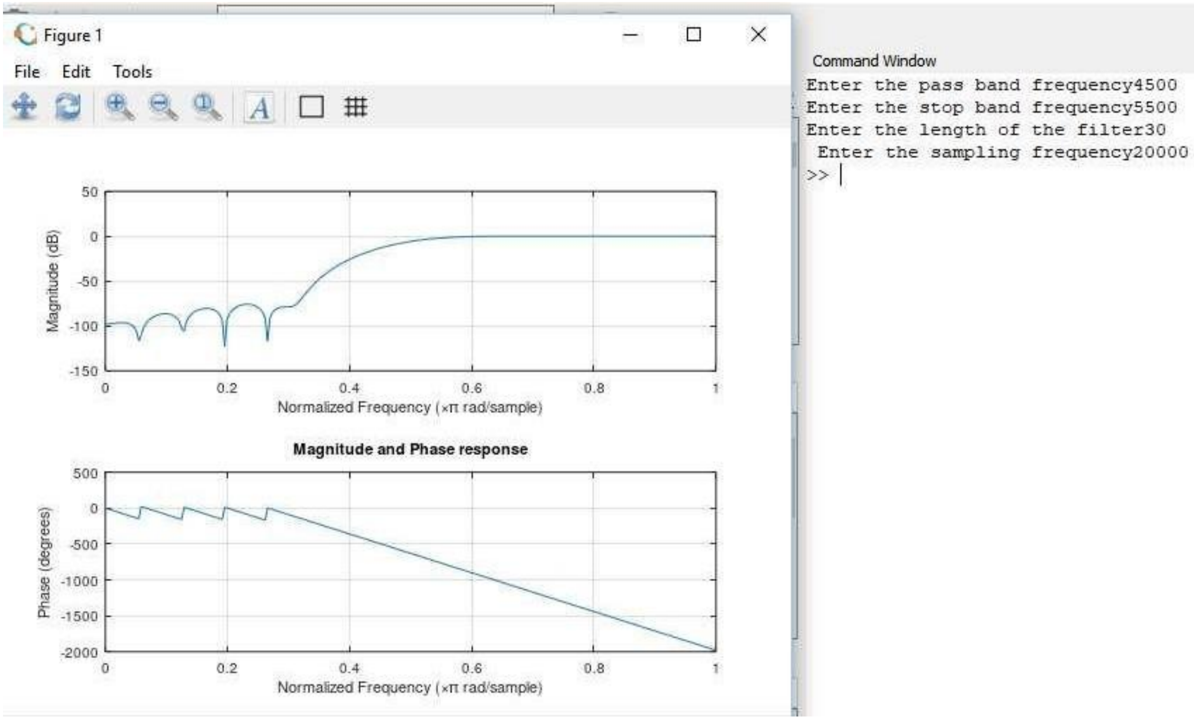


Figure 5.4: Magnitude response of HPF with length of filter M=30

Experiment No. : 6

Realization of IIR Filter

<u>NAME</u> : Vedant Pandey

<u>BATCH</u>: B3

<u>ROLL NO.</u>: 19ET1049

Experiment No. 6

Name : [illegible]

Roll no : [illegible]

Experiment 3

Aim: To implement FIR lowpass filter using various windows and plot the magnitude and phase response using MATLAB

Theory: The frequency of an N^{th} order causal FIR filter is obtained by ~~[illegible]~~ DTFT of the impulse response. The design of FIR filter involves finding the coefficients $h(n)$ that results in a frequency response that satisfies a given specification. FIR filters have two ~~[illegible]~~ important advantages over IIR filters. First, they are guaranteed to be stable, even after the filter coefficients have been quantised. Second, they can be easily constrained to have linear phase. Because FIR are generally designed to have linear phase in the following we consider the design of linear phase filter.

Conclusion

The result of the windowing techniques shown. We have studied the effect of different windows like blackman, hamming and rectangular window on response of a filter. Rectangular window gives larger side lobes.

Experiment No. : 7

Study of Finite Word Length Effects

NAME : Vedant Pandey

BATCH: B1

ROLL NO.: 19ET1058

Experiment No. 7

Name : Vedant Pandey

Roll no. : 19ET1049 B_3

Experiment 7

Aim : To study finite word length effects

Theory : Computers store numbers not with infinite precision but rather in some optimisation that can be packed into fixed numbers of bits or bytes. Almost all computers allow the programmers a choice among several different such representations on data types. Data types can differ from number of bits utilized but also in a more fundamental respect of whether the stored number is represented in fixed point or floating point

Fixed point

A number in fixed point representation is exact. Arithmatics between numbers in fixed point representation is also exact with conditions that (i) The answer is not outside the range of integers that can be represented and (ii) that division is interpreted as producing an integer result, throwing away any integer remainders. There are many formats to represent fixed point number

ii) Finite word length effects : Numerical quantization affects the implementation of linear time invarient discrete time signal In this experiment, we see some of them

Conclusion : In this experiment, different binary number representation schemes are studied. We conclude that the study of finite word length is very crucial in implementation of different digital filters.

Experiment No. : 8

Analysis of speech signal using spectrogram (STFT)

NAME : Vedant Pandey

BATCH: B3

ROLL NO.: 19ET1049

Experiment No. 8

Vedant Pandey

19EE10049 B3

Experiment 8

Aim To identify whether given system is minimum phase maximum phase or mixed phase

Theory : For FIR and IIR filters
1. When all zeroes ~~are~~ are inside the unit circle th its called a minimum phase
2. When all zeroes are outside the unit circle the called a maximum phase
3. When some zeroes are inside the unit circle and remaining are outside the unit circle, then its called mix phase.

If H(z) is having minimum phase. then its inverse system $H^{-1}(z)$ has minimum phase ~~difference~~ But m phase and minimum phase systems results in unstabl inverse systems.

i) a) Since all zeroes are inside the unit circle, its called minimum ~~f~~ phasee.
b) All poles are not inside then given systems is not stable
c) Its an all pass filter

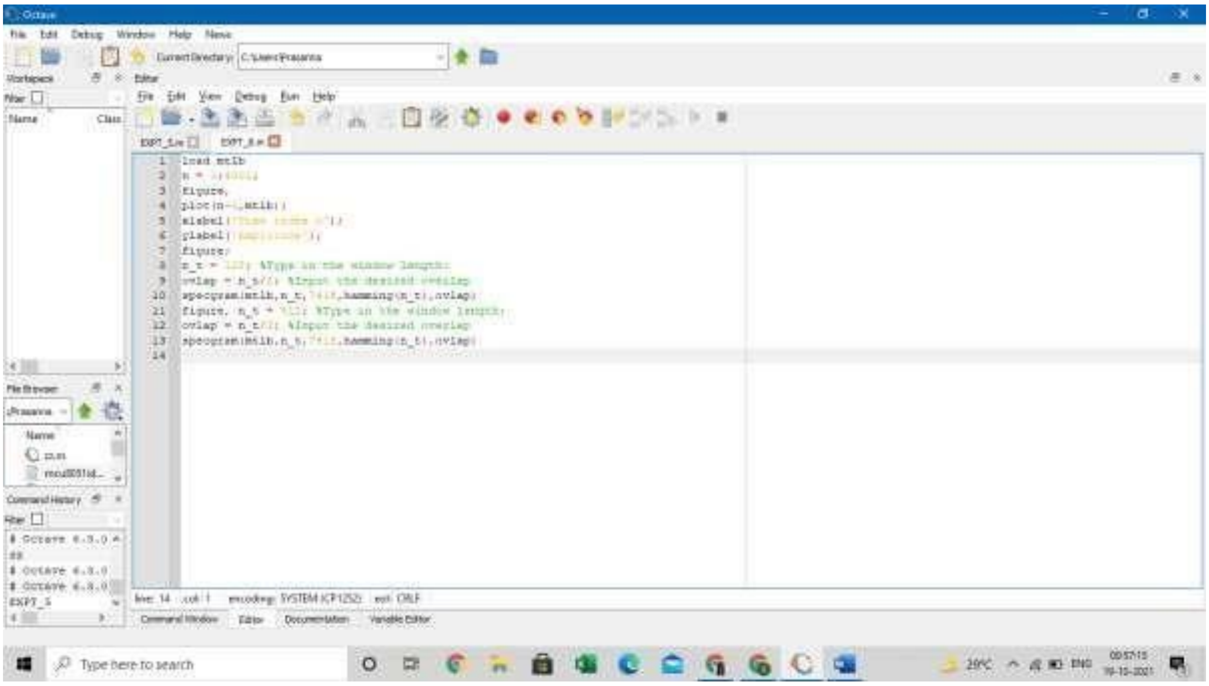
ii) a) Since all zeroes were outside circle is called maximum phase
b) All poles inside the circle is stable
c) Since some zeroes are outside one circle and sam are outside is called mixed ~~plase~~ phase

FOR EDUCATIONAL USE

Program:

The following MATLAB code illustrates the STFT analysis of a speech signal The Signal Processing Toolbox of MATLAB contains a speech signal of duration 4001 samples sampled at 7418 Hz. We compute its STFT using a Hamming window of length 256 with an overlap of 50 samples between consecutive windowed signals using following program.

-



1. Result

Speech signal waveform is shown in Figure 9.1. The narrowband spectrogram of this speech signal shown in Figure 9.2. It is evident from the figure that we get higher frequency resolution. The wideband spectrogram of this speech signal shown in Figure 9.3. It is clear from the figure that we get higher frequency resolution.

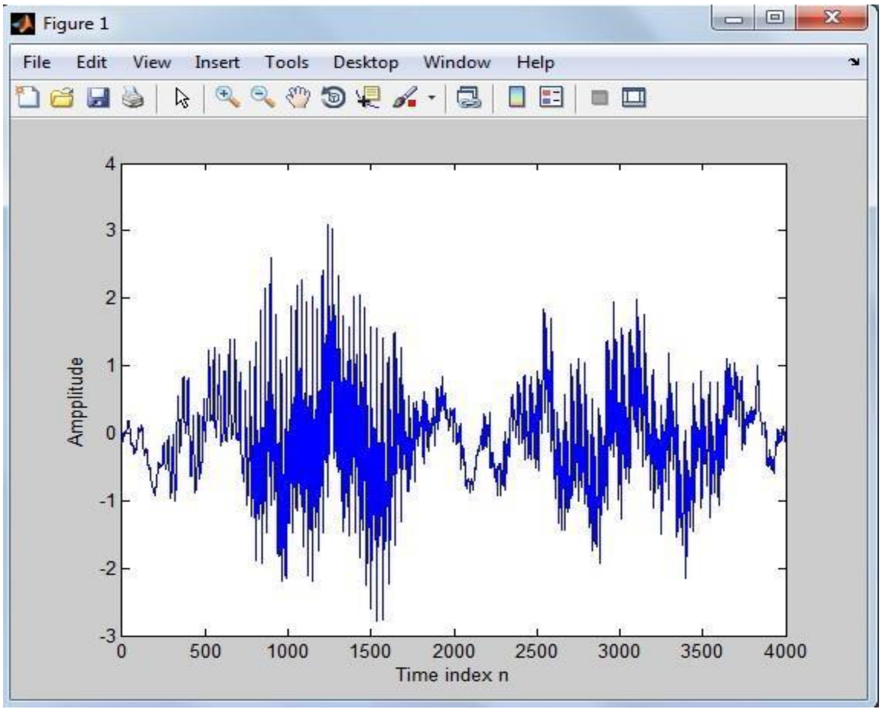


Figure 9.1: Original Speech Signal

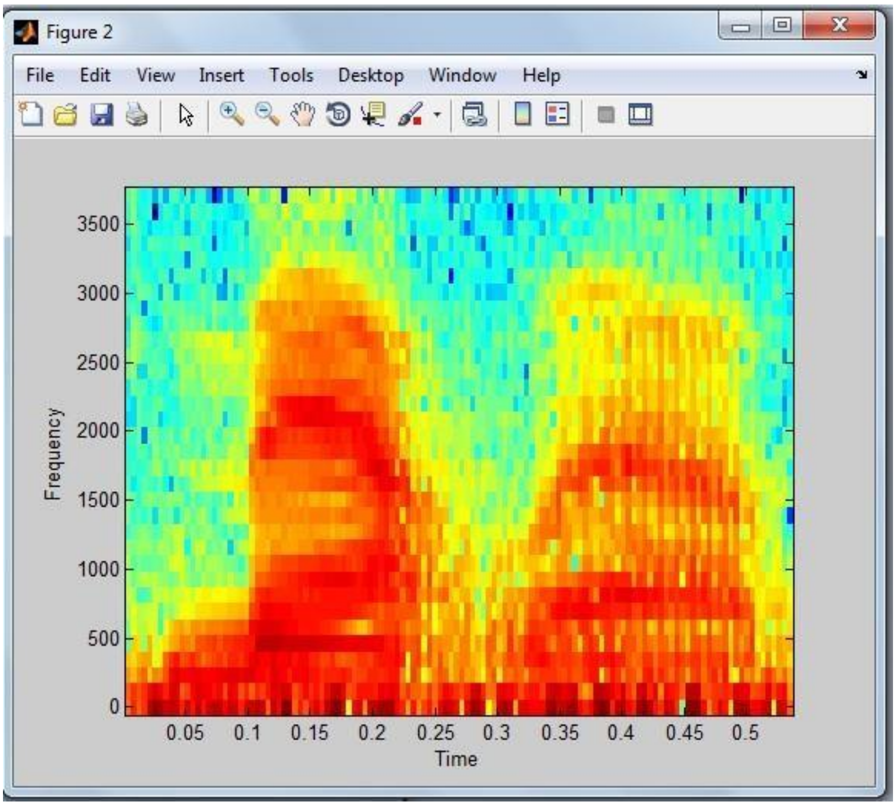


Figure 9.2: Wide band Spectrogram

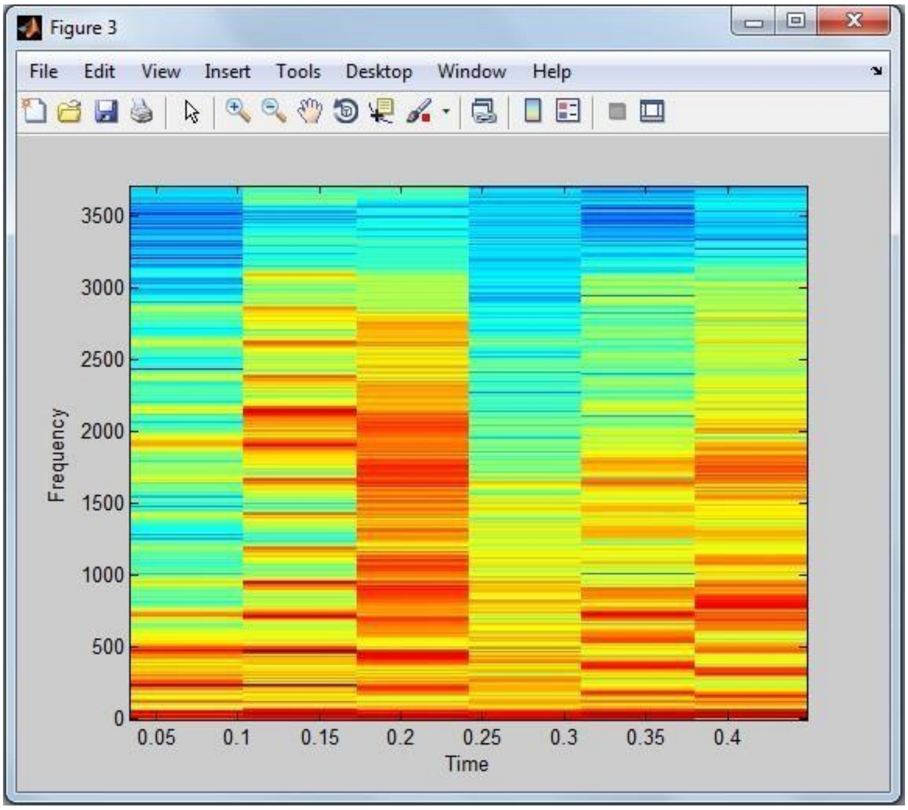


Figure 9.3: Narrowband spectrogram

2. Conclusion and discussion

Using STFT we can analyse signals with various time and frequency resolutions while in Fourier analysis we don't have choice of variable time resolution. Variable time resolution in STFT is achieved using variable lengths. If window length is large we get higher frequency resolution and spectrogram is called as narrowband spectrogram. The spectrogram with smaller window lengths is called as wide band spectrogram which gives the higher time resolution as compared to the narrowband spectrogram