UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

ELECTRICAL ENGINEERING DEPARTMENT

Digital Signal Processing
Lab Project
Complex Engineering Problem

For
7^{th} Semester
Electrical Engineering Department
(**B.Sc. Electrical Engineering**)

Lab Instructor: Zainab Shahid Awan
Course Instructor: Prof. Dr. Obaid Ullah

Submitted By

Name	**Registration Number**
Javeria Moazzam	18-EE-003
Azib Farooq	18-EE-043

Table of Contents

1. Problem Statement

The noisy audio signal is given in the form of a matrix data noisy_signal.mat. The uncorrupted audio signal was first recorded using MATLAB which was a single channel audio of 10 sec duration with 8-bit precision. Then an artificial noise with certain characteristics was created and added in the signal to produce corrupted audio signal data which was stored in .mat format with double precision in range from -1 to 1. The task is to filter the noisy audio signal and extract the audio part only using the appropriate IIR and FIR filters.

2. Aim and Objectives

The aim of this project is to design FIR and IIR filters to remove the noise content of the given audio signal. Main objectives include

- Understanding the types of FIR and IIR filters
- Understanding the nyquist criteria
- Understanding the use of window method in filter design
- Understanding the use of DFT to analyze the frequency content of a signal
- Implementing appropriate filters to reduce the noise content of audio signal

3. Literature Review

Signal analysis is one of the important areas of research in multimedia applications. All recording devices, analog or digital, have traits which make them susceptible to noise or some kind of background noise may be present in the signal. Noise can be random or white noise with no coherence, or coherent noise introduced by the device's mechanism or processing algorithms. Noise reduction is the process of removing noise from a signal.

The goal of this project is to remove the background noise found in a recording of a person's voice. Filtering can be done to remove noise. It is carried out through the filters which are devices or circuits and affects the frequencies pertaining to the signal aspect under consideration. We reviewed some techniques present in literature for filtering.

3.1. Digital Filters

The two major types of digital filters are finite impulse response digital filters (FIR filters) and infinite impulse response digital filters (IIR) when classified on the basis of impulse response. Digital filters can also be classified on the basis of what frequencies they pass and what frequencies they attenuate.

3.1.1. Types of Digital Filters

The four primary types of filters include the low-pass filter, the high-pass filter, the band-pass filter, and the notch filter (or the band-reject or band-stop filter). Take note, however, that the terms "low" and "high" do not refer to any absolute values of frequency, but rather they are relative values with respect to the cutoff frequency

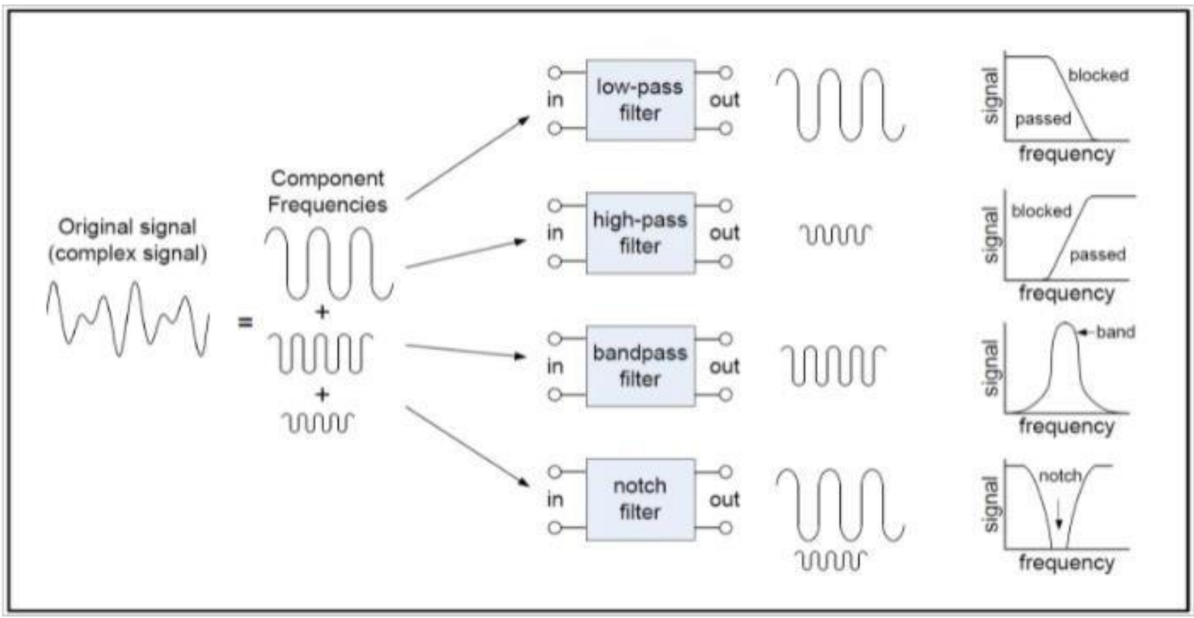


Figure 1: Types of Filters

A. Lowpass Filter

A low-pass filter, as the name says, attenuates frequencies higher than the set cutoff frequency and allows the frequencies lower than that to pass.

B. Highpass Filter

A high-pass filter is simply the opposite of a lowpass filter. It attenuates frequencies below a certain cutoff frequency and allows frequencies higher to pass. We use the term ‘stopband frequency’ to define the frequency in which the attenuation of the signal starts.

C. Bandpass Filter

A band-pass filter basically combines the likes of the two other filters as itfilters allow a certain range of signals with certain frequencies to pass, in its so-called passband, and attenuate all the other frequencies in its stopband.

D. Bandpass Filter

This this filter type is the inverse to a bandpass filter. Similar to the bandpass filter, a notch filter attenuates frequencies in a certain range only and allows the other frequencies to pass (in its passband). The main difference between a bandpass and notch filter is that a bandpass filter allows signals within a particular frequency range to pass and a notch filter will filter those signals out in that one frequency range.

E. Notch Filter

A notch filter is the same as a bandstop filter. It allows all other frequency components of the signal and blocks the specified narrow bandwidth.

3.1.2. Finite Impulse Response Filters (FIR)

The finite impulse response (FIR) filter is a non-recursive filter in which the output from the filter is computed using the current and previous inputs. FIR filters are characterized by the fact that they use only delayed versions of the input signal to filter the input to the output.

The basic FIR filter is characterized by the following equation.

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

The basic characteristics of Finite Impulse Response (FIR) filters are:

- Linear phase characteristic;
- High filter order (more complex circuits)
- Stability.

The main drawback of a digital FIR filter is the time that it takes to execute. Since the filter has no feedback, many more coefficients are needed in the system equation to meet the same requirements. For every extra coefficient, there are extra memory requirements. Hence, for a demanding system, the speed and memory requirements to implement an FIR system can make the system unfeasible.

Windows for FIR Filter Design

Windowing is a scheme to represent Finite Impulse Response (FIR) filters. The window method for digital filter design is fast, convenient, and robust, but generally suboptimal. The basic idea behind the window design is to choose a proper ideal frequency-selective filter (which always has a non-causal, infinite-duration impulse response) and then truncate (or window) its impulse response to obtain a linear-phase and casual FIR filter. Some of the windows are discussed below

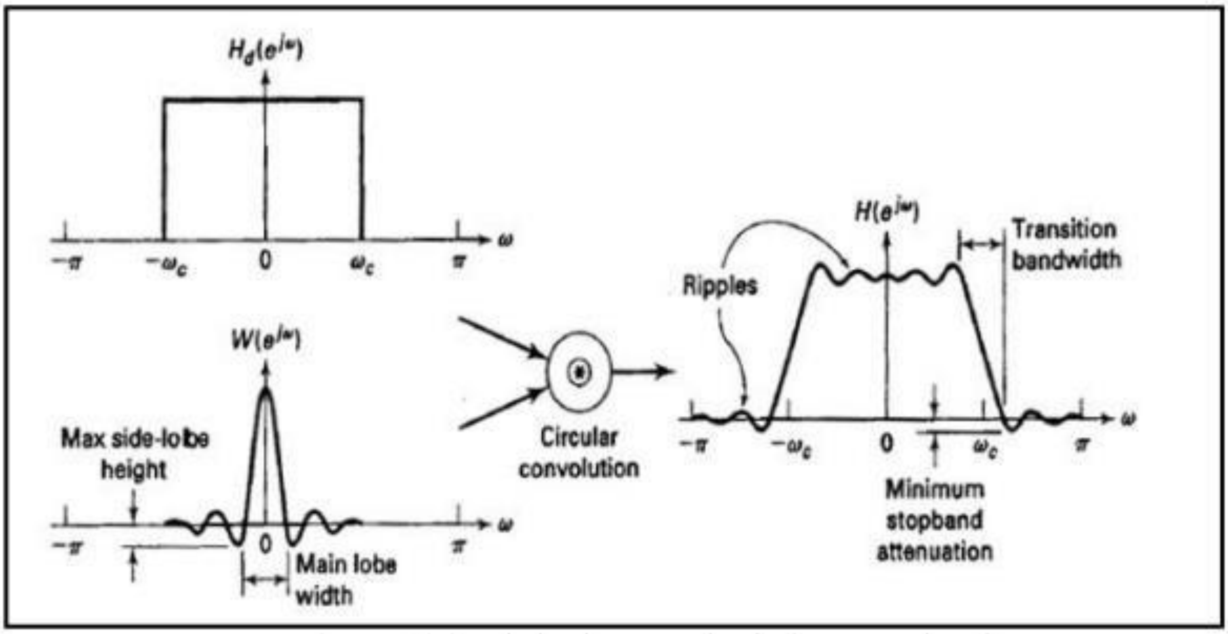


Figure 2: Windowing Method

A. Rectangular Window

The rectangular window (sometimes known as the boxcar or Dirichlet window) is the most straightforward, equivalent to replacing all but N values of a data sequence by zeros, making it appear as though the waveform suddenly turns on and off.

$$w[n] = \begin{cases} 1, & |n| \leq \dfrac{N-1}{2} \\ 0, & \text{otherwise} \end{cases}$$

This is a simple window operation in the time domain and an easy function to analyze in the frequency domain. However, there are two main problems. First, the minimum stopband attenuation of 21 dB is insufficient in practical applications. Second, the rectangular windowing suffers from the Gibbs phenomenon. The rectangular window is impractical in many applications.

B. Hanning Window

The Hann window of length M=N+1 used to perform Hann smoothing, named after the Austrian meteorologist Julius von Hann, is a window function given by

$$w[n] = \begin{cases} \cos^2 \dfrac{\pi n}{M}, & |n| \leq \dfrac{N+1}{2} \\ 0, & \text{otherwise} \end{cases}$$

The width of the main lobe is approximately 8π/N and the peak of the first side lobe is at -32dB.

C. Hamming Window

The window with these particular coefficients was proposed by Richard W. Hamming. Hamming window is optimized to minimize the maximum (nearest) side lobe, giving it a height of about one-fifth of the Hann window.

$$w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi}{N} n\right)$$

The width of the main lobe is approximately 8π/N and the peak of the first side lobe is at -43dB.The side roll off is 20 dB/decade.

D. Blackman Window

The Blackman window exhibits an even lower maximum stopband ripple (about 74 dB down) in the resulting FIR filter than the Hamming window. It is defined mathematically as,

$$w[n] = 0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \left(\frac{4\pi n}{N-1}\right)$$

There is an extra cosine term in Blackman function. This extra term in reduces the side lobes. Reducing side lobes means that the efficiency is increased. This means less power is lost

E. Kaiser Window

The Kaiser window is an approximation to the prolate spheroidal window, for which the ratio of the main-lobe energy to the side-lobe energy is maximized. For a Kaiser window of a particular length, the parameter β controls the relative side-lobe attenuation. For a given β, the relative side-lobe attenuation is fixed concerning the window length.

$$w[n] = \frac{I_o[\beta\sqrt{1-[(n-\alpha)/\alpha]^2}]}{I_o(\beta)}$$

3.1.3. IIR Filters

Infinite impulse response (IIR) filters are characterized by the fact that they use delayed versions of the input signal and fed-back and delayed versions of the output signal to filter the input to the output.

An IIR filter is characterized by the following equation

$$y = \sum_{i=0}^{n} \frac{A_i x[i]}{B_i y[i]}$$

The basic characteristics of Infinite Impulse Response (IIR) are:

- Non-linear phase characteristic;
- Low filter order (less complex circuits); and
- Resulting digital filter has the potential to become unstable.

IIR filters are more versatile and they are computationally easier to implement. They are cheaper too. However, they also have a higher chance of being affected by quantization operations like truncation and rounding. This is due to the feedback mechanism that introduces poles in the transfer function. On the contrary, FIR filter transfer functions do not have poles. Some of the IIR filters are discussed below.

A. Butterworth Filters

The classical method that is used to design analog filters is butterworth approximation. It is also referred as maximally flat filters. The calculations and other mathematical operations are simpler than the other filters. It also has poor phase characteristics. Higher the butterworth filter order, higher the number of cascaded stages required. Practically butterworth filter frequency response is unattainable because it produces excessive passband ripple.

B. Chebychev Filter

There are two types of chebyshev filter they are called chebyshev type1 and chebyshev type2. Generally type1 referred as regular filter and it is most common chebyshev filter. It has the steepest roll-off but it presents in band ripple. It has more rapid transition from passband to stopband rather than butterworth filter. Type1 has equiripple in passband whereas no ripple in stopband. Type2 filter is inversion of type1 filter because it has equi-ripple in stopband and no ripple in passband.

C. Eliptic Filter

Elliptic filter is has equalized ripple in both the bands. As the ripple in stopband is zero it becomes the chebyshev type 1filter and alternatively when ripples in passband are zero it becomes chebyshev type 2 filter. The amount of ripple in each band is adjustable independently. Transition bandwidth is minimum. It is difficult to design and not easy to analysis by basic tools.

To summarize, Filters can be classified in several different groups, depending on what criteria are used for classification. The two major types of digital filters are finite impulse response digital filters (FIR filters) and infinite impulse response digital filters (IIR). Both types have some advantages and disadvantages that should be carefully considered when designing a filter. It is necessary to take into account all fundamental characteristics of a signal to be filtered as these are very important when deciding which filter to use.

4. Methodology

We have designed IIR and FIR filters to get the denoised signal. The noisy signal was given which had artificial noise added to it. Below are the steps for filter design.

4.1. Determine the Sampling Frequency (Fs)

Sampling Frequency (Fs)

The given signal was of 10 sec duration and the total number of samples present in the signal was 80,000. To calculate the sampling frequency, we used the following formula

$$Sampling\ Frequency\ (Fs) = \frac{Number\ of\ Samples}{Duration}$$

$$Fs = \frac{80{,}000}{10}$$

$$Fs = 8000$$

Intuition behind the selection of the FS

The given signal is an audio signal. The maximum frequency of an audio signal is 4k Hz. According to nyquist criteria *the sampling frequency of a signal should be at least twice the maximum frequency of the signal to avoid aliasing.*

Fs > 2 fmax

So, the sampling frequency of an audio signal should be at least 8k Hz to avoid aliasing. The Fs of our signal is selected to be 8000 Hz which is sufficient to avoid aliasing according to the nyquist criteria.

Reason to store data in the double precision

The priority is the quality audio and as double precision provides better (perceptual) quality. Hence, double precision offers greater bit depth so it is used.

4.2. Analysis of Audio Signal in Time Domain

The audio was played using the ‘sound’ command in MATLAB. The audio was a speech signal with some pauses in between. The signal was then plotted in time domain in analog form. The noise is constant throughout the signal.

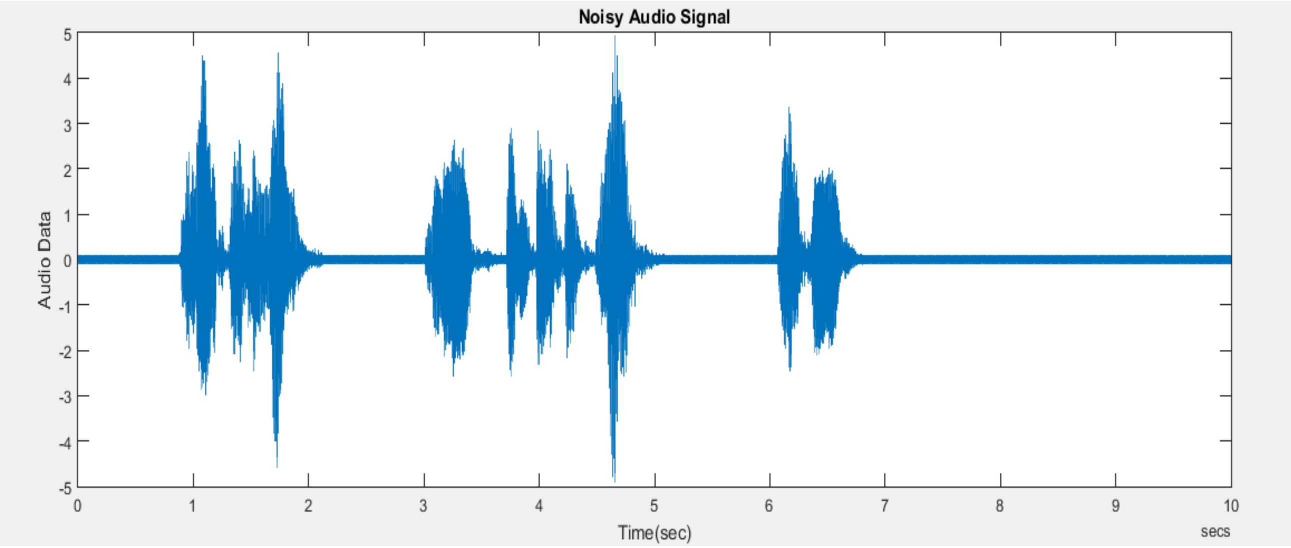


Figure 3: Noisy Audio Signal

4.3. Digitization of the Given Signal

Next, the audio signal was sampled and quantized to 15 quantization levels to convert it into digital form and the results were plotted.

4.4. Frequency Domain Analysis of the Noisy Signal

The frequency domain analysis of the audio signal was performed. The DFT of the noisy signal was computed using the fft command in MATLAB. It was observed that most of the signal content was contained in the low frequencies.

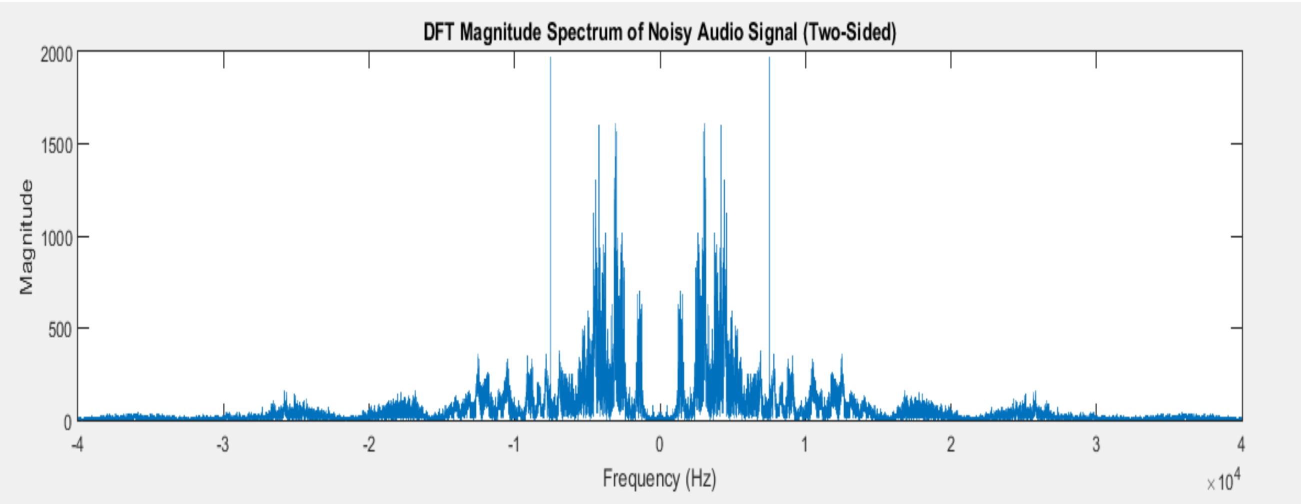


Figure 4: DFT of Audio Signal

Value of N chosen for DFT computation

For computing DFT, the value of N was chosen to be equal to the length of the given audio signal. The length of the signal was 80,000. So, the value of N was selected to be 80,000.

Implication of making N smaller or larger than the input signal data length

When length of the signal (L) is less than the number of DFT points (N) taken, then there will be no aliasing. The frequency resolution of the spectrum is increased. However, when length of the signal is greater than the number of DFT points then aliasing will occur and the original signal cannot be reconstructed back.

Reason for computing DFT through FFT algorithm in MATLAB

By using the FFT algorithm to calculate the DFT, convolution via the frequency domain can be faster than directly convolving the time domain signals. The final result is the same; only the number of calculations has been changed by a more efficient algorithm, resulting in faster computations

For example, the length of given signal is 80,000. For direct DFT computation,

$$\text{Number of Complex additions} = N^2 - N = 6,399,920,000$$
$$\text{Number of Complex Multiplications} = N^2 = 6,400,000,000$$

By using the radix 2 FFT algorithm, the number of operations required are reduced to:

$$\text{Number of Complex additions} = N\log_2 N = 1303016.99$$
$$\text{Number of Complex Multiplications} = (N/2)\log_2 N = 651200$$

4.5. Analysis of the Noisy Part

Next the nature of the noise was determined and analysis of the noisy part of signal was performed. DFT of that part of the signal was computed having only the noisy part.

Identification of the frequencies pertaining to the noise

The audio signal is a speech signal having some pauses in between. The pauses are the part of the signal where no useful information is present and only noise can be heard. The frequencies corresponding to the noise can be determined by extracting the chunks of the noisy signal where pauses are present and then taking the DFT of those chunks. The DFT gives the frequencies corresponding to the noise.

Inference about the Characteristics of the Noise Added in the Signal

After listening to the audio, it can be inferred that the noise in the audio signal is sinusoidal in nature. The audio signal has a beep tone in the background which is the characteristic of sinusoidal noise. By looking at the DFT of the noisy signal, it can be seen that two sinusoids of frequency 367 Hz and 656 Hz are added to the clean signal.

Below is the DFT of a chunk of the audio signal consisting only of the noisy part.

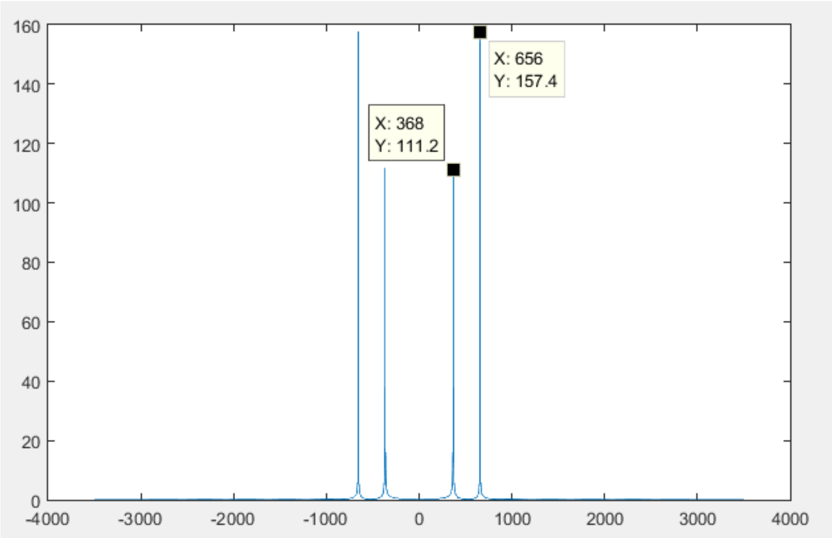


Figure 5: DFT of first 7000 samples of given signal (Consisting only of noisy part)

4.6. Type of Filter Required

For FIR filter, bandstop filter will be used to remove the noise content of the signal. For IIR filters, we will design two notch filters.

Intuition behind choice of type of the filter and design method

As we have observed that the noise signal pertaining the audio signal is sinusoidal consisting of two frequencies. We have to reject these two frequency components so we have used bandstop filter (FIR) because it allows a specific range of frequencies to not pass to the output, while allowing lower and higher frequencies to pass with little attenuation and notch filter (IIR) as it will remove only the two noise frequency components and allow rest of the frequencies to pass through. A notch-filter has the same working principle as of band-reject filter. It allows all other frequency components of the signal and blocks the specified narrow bandwidth.

The design method for FIR filters is based on the use of windows to truncate the impulse response h(n) and obtaining the desired spectral components. This method is used in order to minimize the side lobe level or ringing trails. By choosing the window carefully, we can manage various tradeoffs so as to maximize the filter-design quality in a given application. Hamming window is the best technique, usually preferred by many others.

We have designed the bandstop filter using the Hamming window for FIR filter as it is more preferred due to its relatively narrow main lobe width and good attenuation of the first few side lobes.

Type of the filter more suitable for the given CEP (FIR or IIR)

An FIR filter is a filter with no feedback. This can be an advantage because it makes an FIR filter inherently stable. Another advantage of FIR filters is the fact that they can produce linear phases. FIR filter can be used in cases where a speech signal must be extracted from a noisy environment like in our case. As the given speech signal is buried in a very noisy environment with many periodic frequency components lying in the same bandwidth as the speech signal.

However, since linear phase is not necessarily a requirement in speech signals, IIR filters are more suitable because IIR filters provide the same noise removal in the output signal but the advantage of using IIR is that the order and complexity of the designed filter will be much less than FIR filters. Hence, IIR filter is more suitable for the given CEP.

Advantages of FIR filters

- FIR filter are always stable
- They are simple
- FIR filters have linear phase response
- It is easy to optimize
- Both recursive, as well as nonrecursive filter, can be designed using FIR designing techniques
- For designing a filter having any arbitrary magnitude response; FIR designing techniques can be easily applied

Disadvantages of FIR Filters

- They require more memory and/or calculation to achieve a given filter response characteristic.
- Also, certain responses are not practical to implement with FIR filters
- For the implementation of FIR filter complex computational techniques are required
- Expensive due to large order
- Time-consuming process

Advantages of IIR filters

- IR filter is better than the FIR in that, it can produce the same response using some fewer delay blocks.
- This filter is useful only when some analog filter is bandlimited.
- They are more susceptible to the problem of line finite length arithmetic.
- Implementation of IIR filter involves fewer parameters, less memory requirement, and lower computational complexity.
- IIR filters usually require fewer coefficients to execute similar filtering operations, that IIR filters work faster, and require less memory space
- The implementation of an IIR filter involves fewer parameters needed..

Disadvantages of IIR filters

- There are more susceptible to the problem of finite length arithmetic, Such as the noise is generated by calculations, and limit cycle.
- They are harder to implement using fixed-point arithmetic.
- IIR filter becomes unstable.
- IIR filters usually have a non-linear response, while the FIR usually has a linear phase.
- The realization of the IIR filter is not very easy as compared to FIR filters.
- It is hard to optimize than the FIR filter.
- The disadvantage of IIR filters is the nonlinear phase response.
- Due to the presence of a feedback loop IIR filters are difficult to implement in a circuit

Which one is computationally efficient?

IIR filters are more computationally efficient than FIR filters. The computational efficiency of FIR filters is quite less than IIR filters. So, if liner phase is not a requirement, then IIR filters are used.

4.7. Order of FIR and IIR filter

Order of FIR Filter

As a general rule, the order of a filter is its length minus one.

As we have used hamming window design method and we have found the length of the designed filter using the formula

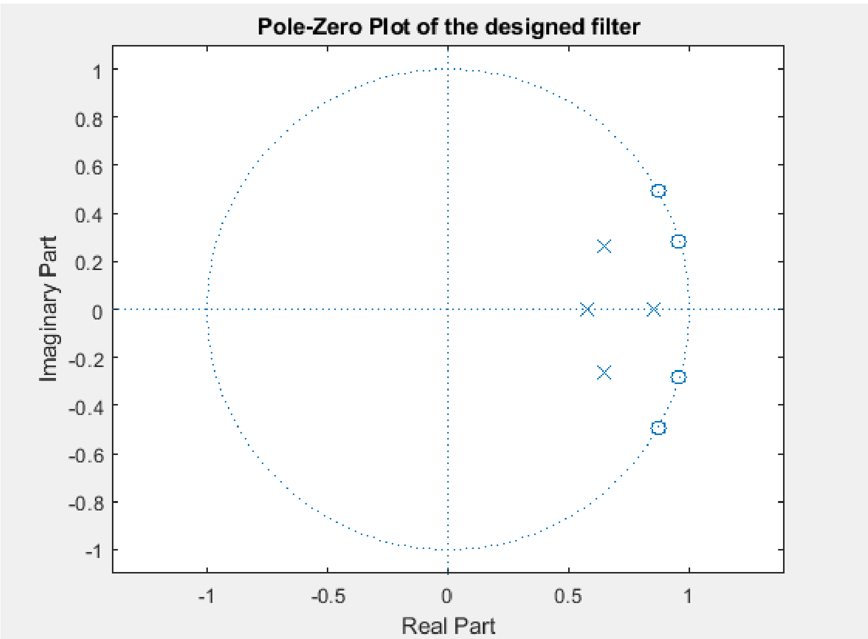
$$\boldsymbol{M} = \frac{\boldsymbol{6.6\pi}}{\Delta m}$$

As $\Delta\omega = 0.0346$ rad/sec
M=6.6*pi/ 0.0346;
M=600;
To make odd size, we use M+1 = **601**
Hence, order of the filter will be

Order of FIR filter = M-1 = 600

Order of IIR Filter

As the order of any filter is equal to the number of zeros in it, So, the order of the designed IIR notch filter can be found out by finding the zeros in the respective IIR notch filter.



We can clearly observe that there are four (4) zeros in the pole zero plot of the IIR filter. Hence, its order will be **four,** which is a lot less than the FIR filter.

Implications of changing the order of the filter

The order of a filter decides how better a filter performs a filtering action. Order determines how faster the filter cut-off is rolled off, that is, how steeper will be the transition from pass band to the stop band.

Higher order filters have the following advantages

- more attenuation between the pass band and stop band
- narrower transition band
- flatter pass band (less ripple)

On the other hand, there are some disadvantages and complications arises such as

- more compute intensive
- longer group delay
- high chance for instability in case of IIR filters
- higher dynamic range requiring more costly data format: single precision to double precision, integer to floating points etc.

4.8. FIR Filter Design

The passband and stopband frequencies were found using the noisy signal. The frequencies pertaining to noise were found and the passband and stopband frequencies were found accordingly. After that the cutoff frequencies were calculated using the formulas

$$\boldsymbol{wc1 = ws1 - wp1}$$

And

$$\boldsymbol{wc2 = wp2 - ws1}$$

Next the order of the filter was calculated the formula for hamming window, which was found to be 600.

$$\boldsymbol{M} = \frac{\boldsymbol{6.6\pi}}{\Delta m}$$

Where

$$\Delta m = \boldsymbol{min}(\boldsymbol{wc1}, \boldsymbol{wc2})$$

Finally the bandstop filter was obtained. The noiseless signal was obtained by convolving the noisy signal with the designed filter in time domain. The pole zero plot was plotted using the command ‘zplane’. Finally the noiseless digital audio signal was plotted.

4.9. IIR Filter Design

For the IIR filter we designed two notch filters that attenuated the frequencies at 367 Hz and 656 Hz respectively. The noisy signal was then filtered through both of these filters and the noiseless signal was obtained.

Symmetry in pole zero plots of the designed FIR and IIR filter

In case of FIR filter, there is no symmetry in pole zero plots has been observed but in case of IIR filters there is symmetry as pair of conjugate poles and zeros exist in it. one Quadruplet of zeros and one pair of complex poles and two real zeros can be observed.

5. Implementation

5.1. Filter Design Specifications (FIR)

Parameter	Value
Passband ripple	1 dB
Stopband ripple	-50 dB
Sampling frequency	8000 Hz
First Passband edge frequency	300 Hz
Second Passband edge frequency	800 Hz
First Stopband edge frequency	345 Hz
Second Stopband edge frequency	700 Hz
First Cutoff frequency	322 Hz
Second Cutoff Frequency	750 Hz

5.2. Filter Design Specifications (IIR)

Parameter	Value
Passband ripple	1 dB
Stopband ripple	-50 dB
Bandwidth	1500 Hz
Sampling Frequency	8000 Hz
Stopband Frequency of First Notch Filter	367 Hz
Stopband Frequency of Second Notch Filter	656 Hz

5.3. MATLAB Code for FIR Filter

```
%========================================================%
      %%%%     Digital Signal Processing Lab     %%%
      %%%%   Complex Engineering Problem (CEP)   %%%
%========================================================%
                     %GROUP MEMBERS
                   %ROLL NO:18-EE-03
                   %ROLL NO:18-EE-43

clear all;
close  all;
clc;
LW='linewidth';
load ('noisy_signal.mat')                          %loading the noisy signal

%sound(noisy_signal)                               %playing the noisy signal

no_of_samples=length(noisy_signal);
time_duration=10;
```

```
fs=no_of_samples/time_duration;                    %sampling freq (fs = No. of
samples of noisy signal/time duration)
t=0:seconds(1/fs):seconds(time_duration);          %Time Array
t=t(1:end-1);                                      %Time Index Adjustments

%sound (noisy_signal,fs);
%========================================%
%%%%     Noisy signal plot    %%%
%========================================%
figure(1)
plot(t,noisy_signal);
title('Noisy Audio Signal');
xlabel('Time(sec)');
ylabel('Audio Data');

%===========================================%
%%%%     Digitizing the Noisy signal     %%%
%===========================================%
%quantizing levels
n=100;
%calculating the extreme values of the input signal
y=noisy_signal;
y_max=max(y);
y_min=min(y);
yquan=y/y_max(1);
%calculating step size to accomodate 15 quantization levels
d=(y_max-y_min)/n;
d=d(1);
%quantization levels
q=d.*[0:n-1];
q=q-((n-1)/2)*d;
%quantized input audio signal
for i=1:n
    yquan(find((q(i)-
d/2<=yquan)&(yquan<=q(i)+d/2)))=q(i).*ones(1,length(find((q(i)-
d/2<=yquan)&(yquan<=q(i)+d/2))));
    bquan (find(yquan==q(i)))=(i-1).*ones(1,length(find(yquan==q(i))));
end
figure(2)
stem(yquan);
xlabel('Sample Number');
ylabel('Audio Data');
title('DIGITIZED AUDIO SIGNAL');

%==================================================%
%%%%     Frequency analysis of noise signal    %%%
%==================================================%
%%%%%%%%  Double Sided spectrum   %%%%%%%%%
fft_noisy_signal=abs(fftshift(fft(noisy_signal)));
L1=-no_of_samples/2:1:(no_of_samples/2)-1;
figure(3)
subplot(211);
plot(L1,fft_noisy_signal);
title('DFT Magnitude Spectrum of Noisy Audio Signal (Two-Sided)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

%%%%%%%%  Single Sided spectrum   %%%%%%%%%
x_mag=abs(fft(noisy_signal));
bin_val=[0 : no_of_samples-1];
Hz=bin_val*fs/no_of_samples;                   %conversion into Hertz
N_2=ceil(no_of_samples/2);
subplot(212);
```

```
plot(Hz(1:N_2),x_mag(1:N_2));
title('DFT Magnitude Spectrum of Noisy Audio Signal (Single-Sided)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
%===============================================================================
==%
%%%%     Frequency analysis of noise only (finding noise characteristics)    %%%
%===============================================================================
==%
noise=noisy_signal(1:7000,1);  %extracting the noise componenets

f2=length(noise);
fft_noise=abs(fftshift(fft(noise)));
f2=-f2/2:1:(f2/2)-1;
figure(4)
plot(f2,fft_noise);
title('Fourier Transform of Noise');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

%==================================================================%
%%%%     Magnitude and Phase Response of designed FIR filter    %%%
%==================================================================%
%passband and stopband edge frequencies specified by noise spectrum
wp1=0.075*pi;
ws1=0.086*pi;
ws2=0.2*pi;
wp2=0.3*pi;
Ap=1;As=50;

%========================================%
%%%%  Conversion to absolute values %%%%
%========================================%
deltap = (10^(Ap/20)-1)/(10^(Ap/20)+1);
deltas = (1+deltap)/(10^(As/20));
delta = min(deltap,deltas);
Deltaw1 = ws1-wp1;
Deltaw2 = wp2-ws2;
deltaw = min(Deltaw1,Deltaw2);

%========================================%
%%%%  CUT-OFF FREQUENCIES %%%%
%========================================%
wc1 = (ws1+wp1)/2;
wc2 = (ws2+wp2)/2;

%======================================================%
%%%% designed bandstop filter using hamming window %%%%
%======================================================%
M = ceil(6.6*pi/deltaw);                % Window Length
if mod(M,2) == 0; M = M+1; end
%
n = 0:M-1;
hd = ideal_lp(pi,M)-[ideal_lp(wc2,M)-ideal_lp(wc1,M)];  %bandstop filter
h = hd .* hamming(M);                                    %designed filter
figure(5)
w = linspace(0,1,1000)*pi;
H = freqz(h,1,w);
Hmag = abs(H);
Hdb = 20*log10(Hmag./max(Hmag));
subplot(211); plot(w/pi,Hdb,LW,2);
title('Mag. Response in dBs');
xlabel('Normalized Frequency (Hz)');
```

```
ylabel('Magnitude (dB)'); grid;
Hangle =  angle(H);
subplot(212); plot(w/pi,Hangle);
title('Phase Response');
xlabel('Normalized Frequency (Hz)');
ylabel('Phase (degrees)'); grid;
suptitle('Magnitude and Phase Response of designed filter');

%=======================================================%
%%%%   Pole-Zero Plot of the designed filter     %%%%
%=======================================================%
figure(6)
zplane(h);

%=======================================================%
%%%%              Filtered Audio Signal     %%%%
%=======================================================%
noiseless_signal=conv(noisy_signal,h);                  %applying the filter
on noisy signal
%sound(noiseless_signal);                               %playing back the
noiseless signal
figure(7)
stem(noiseless_signal);
title('Filtered Audio Signal');
xlabel('Sample Number');
ylabel('Audio Data');
```

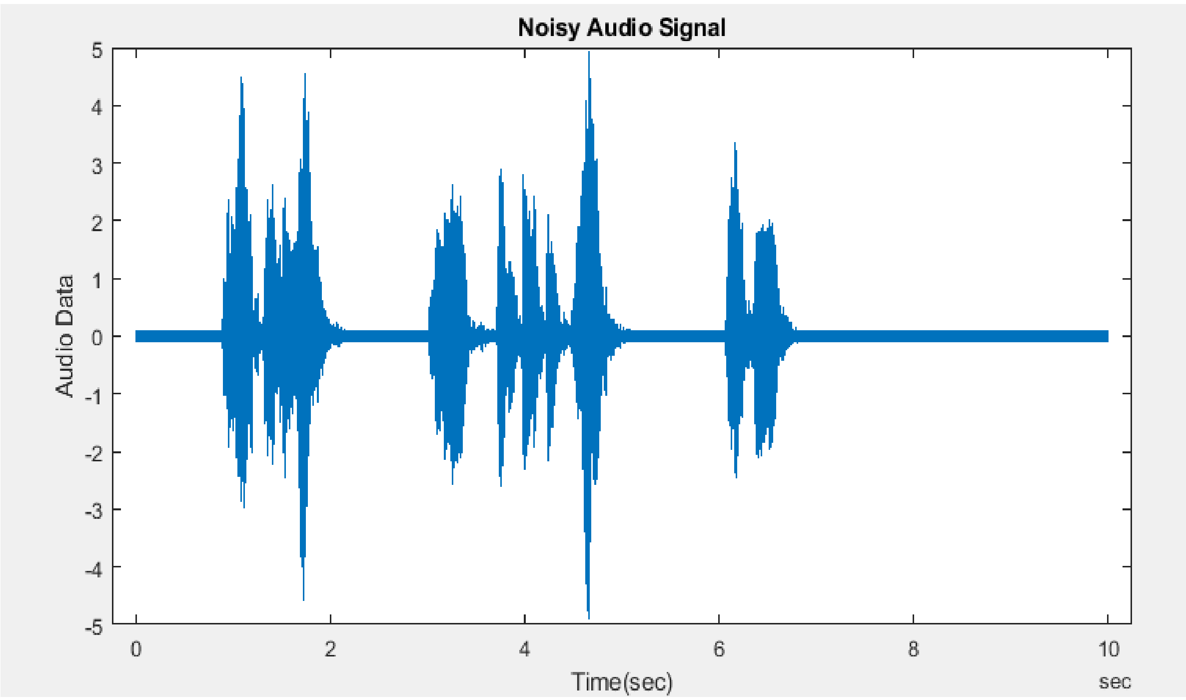
5.4. MATLAB Code for IIR Filter

```
            %=========================================================%
                 %%%%     Digital Signal Processing Lab     %%%
                 %%%%   Complex Engineering Problem (CEP)    %%%
            %=========================================================%
                                %GROUP MEMBERS
                              %ROLL NO:18-EE-03
                              %ROLL NO:18-EE-43

clear all;
close  all;
clc;
LW='linewidth';

load ('noisy_signal.mat')    %loading the noisy signal

%===========================================%
%%%%     Designing IIR filter    %%%
%===========================================%
Fs=8000;
Fnotch1=367;      % Notch Frequency
Fnotch2=656;
BW   = 1500;      % Bandwidth
Apass=1;          % Bandwidth Attenuation
[a0,b0] = iirnotch(Fnotch1/(Fs/2), BW/(Fs/2), Apass);   %%BW/(Fs/2)[[so that
respective bandwidth is bw 0 and 1]]
[a1,b1] =  iirnotch(Fnotch2/(Fs/2), BW/(Fs/2), Apass);

a=conv(a0,a1);   %numinator cofficients of IIR Notch filter
b=conv(b0,b1);   %denominator cofficients of IIR Notch filter
%==================================================================%
%%%%     Magnitude and Phase Response of designed IIR filter    %%%
%==================================================================%
```

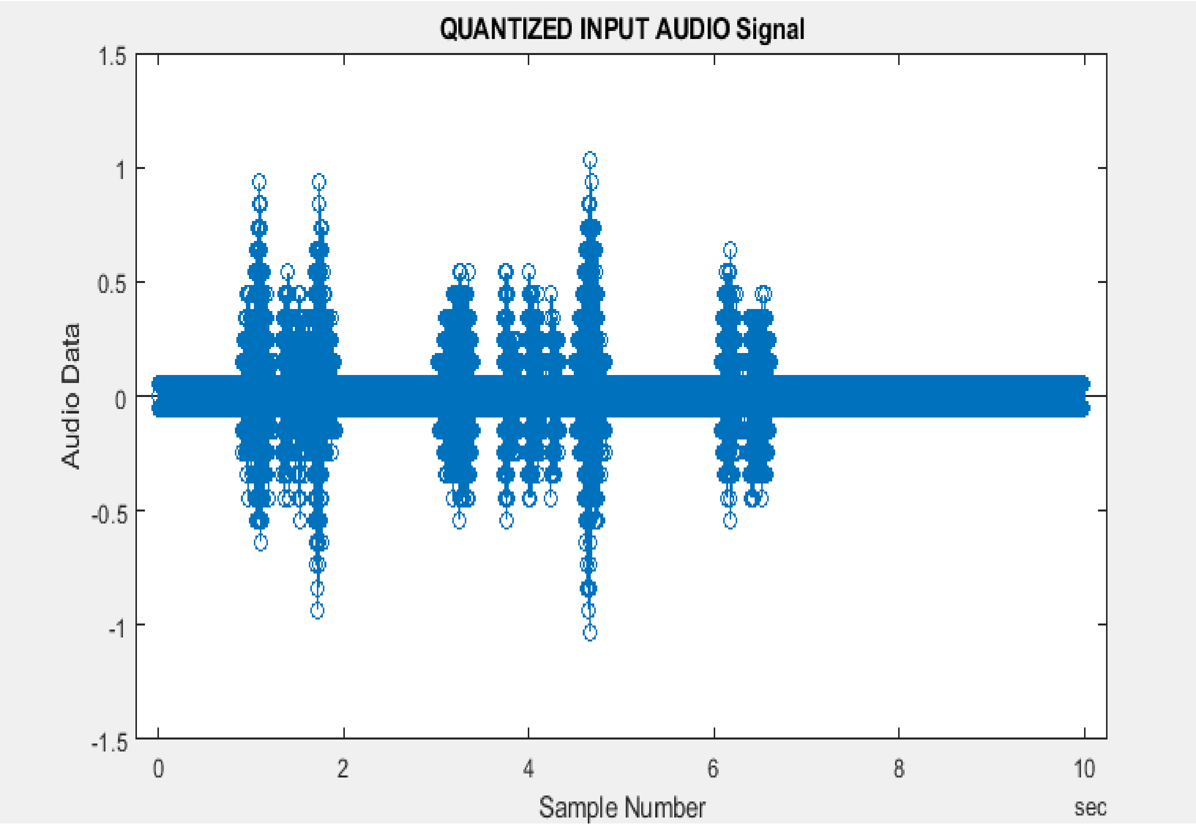
```
figure(1)
freqz(a,b);
suptitle('Magnitude and Phase Response of designed IIR filter');

%=======================================================%
%%%%   Pole-Zero Plot of the designed filter      %%%%
%=======================================================%

figure(2)
zplane(a,b);
title('Pole-Zero Plot of the designed filter');

%=======================================================%
%%%%              Filtered Audio Signal      %%%%
%=======================================================%

noiseless_signal=filtfilt(a,b,noisy_signal);%applying the filter on noisy signal
%sound(noiseless_signal);                    %playing back the noiseless signal

figure(3)

stem(noiseless_signal);
title('Filtered Audio Signal');
xlabel('Sample Number');
ylabel('Audio Data');
```

6. Simulation Results

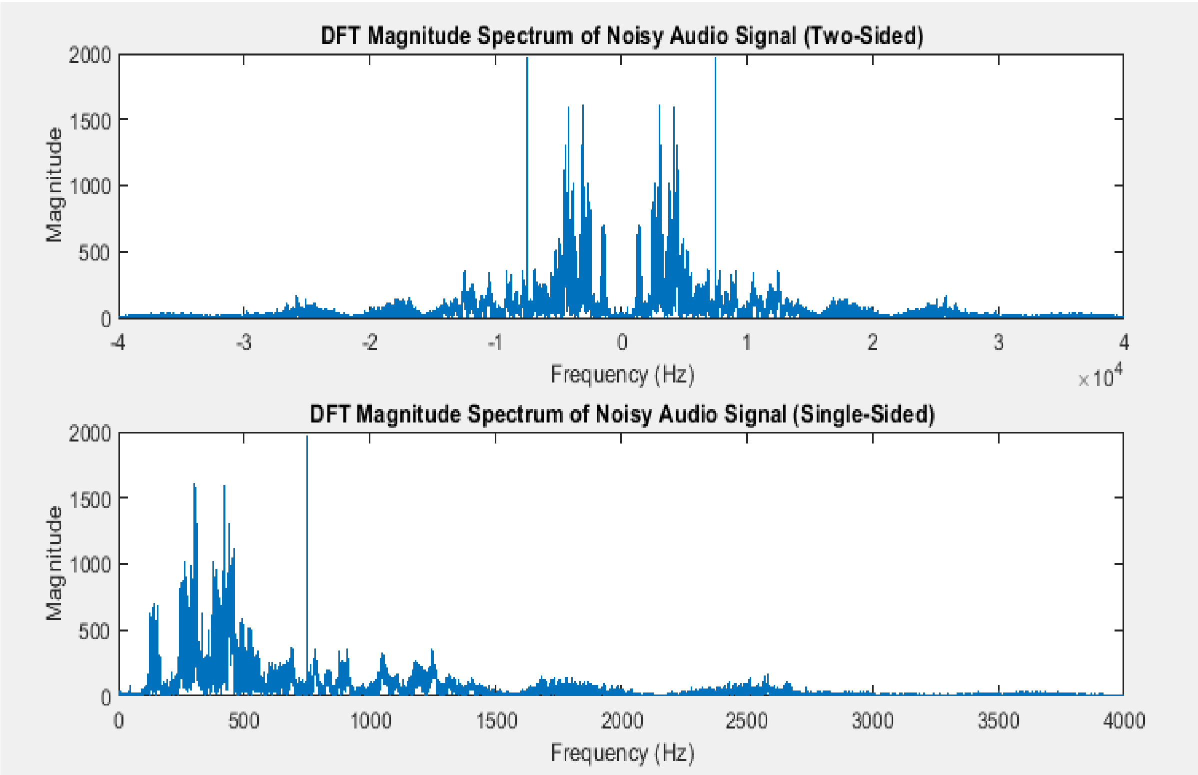
<u>NOISY AUDIO SIGNAL</u>



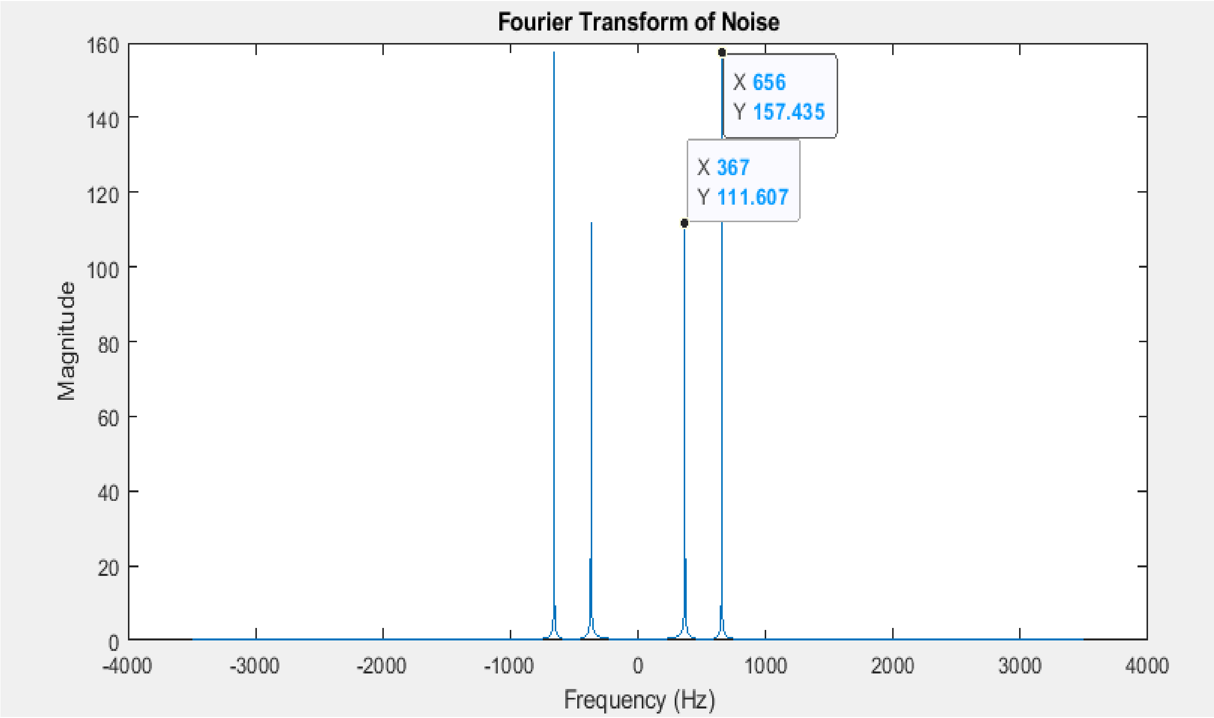
DIGITAL NOISY AUDIO SIGNAL



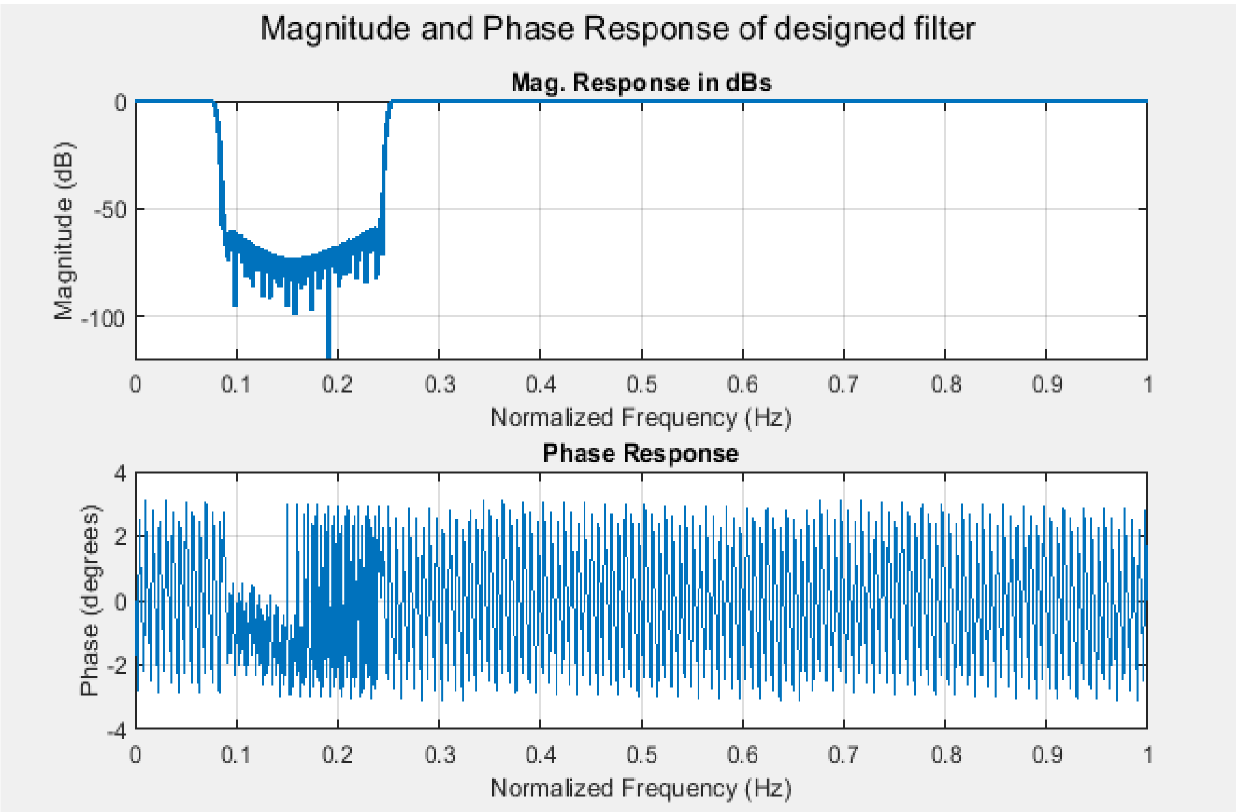
DFT MAGNITUDE SPECTRUM OF NOISY AUDIO SIGNAL



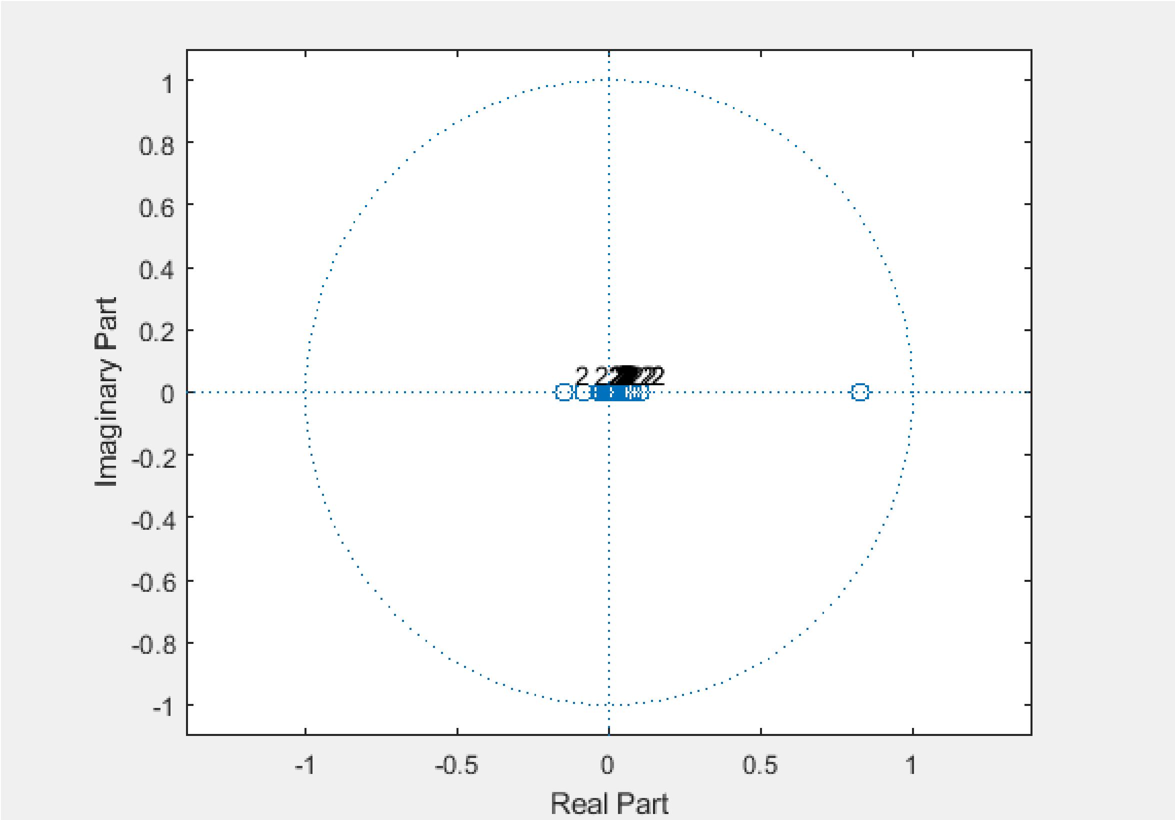
DFT OF NOISE (FIGURE TOIDENTIFY THE NOISE CHARACTERISTICS)



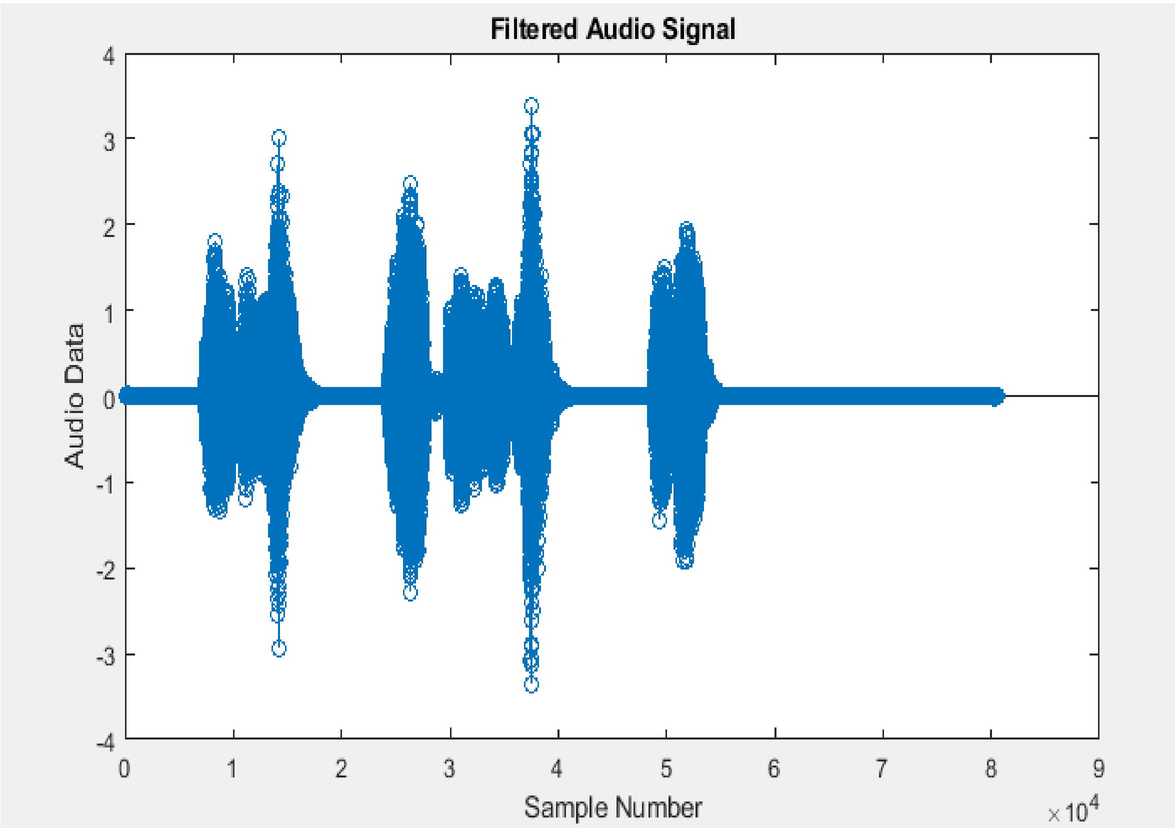
MAGNITUDE AND PHASE RESPONSE OF DESIGNED FILTER (FIR)



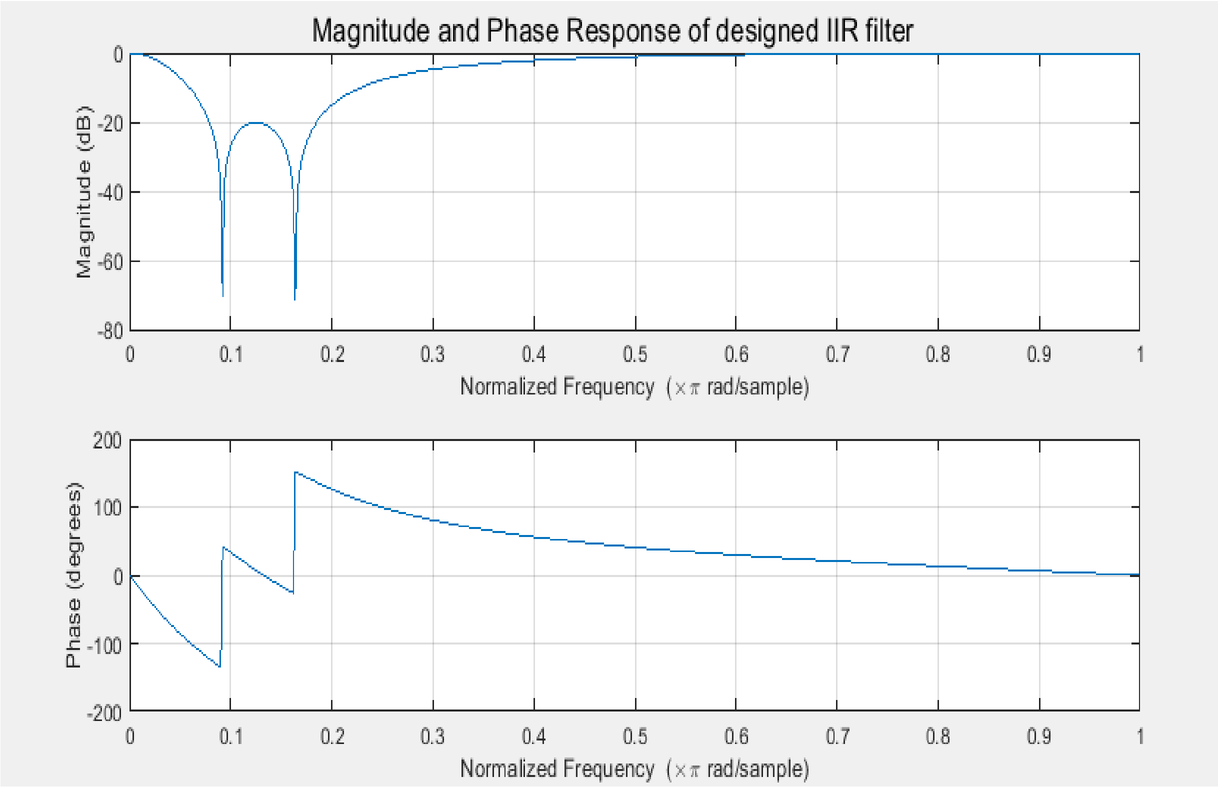
POLE-ZERO PLOT OF THE DESIGNED FIR FILTER



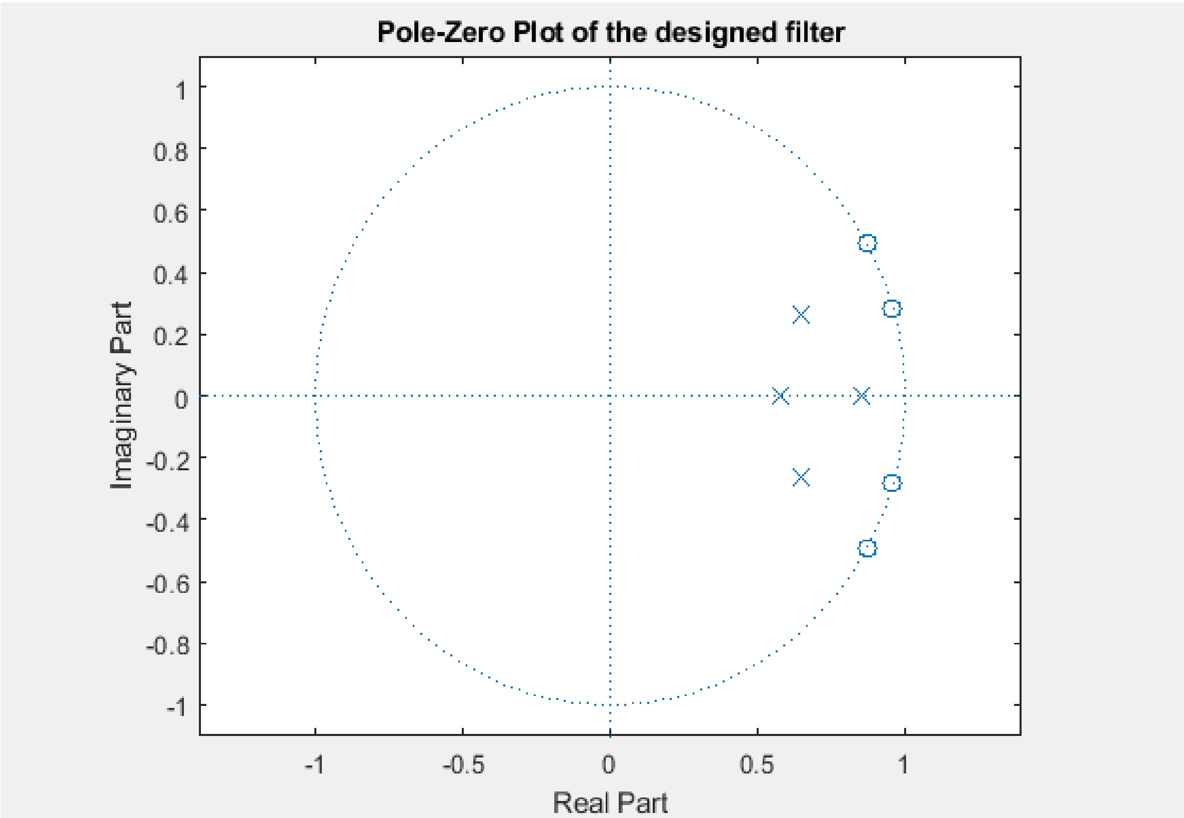
FILTERED AUDIO SIGNAL USING FIR FILTER



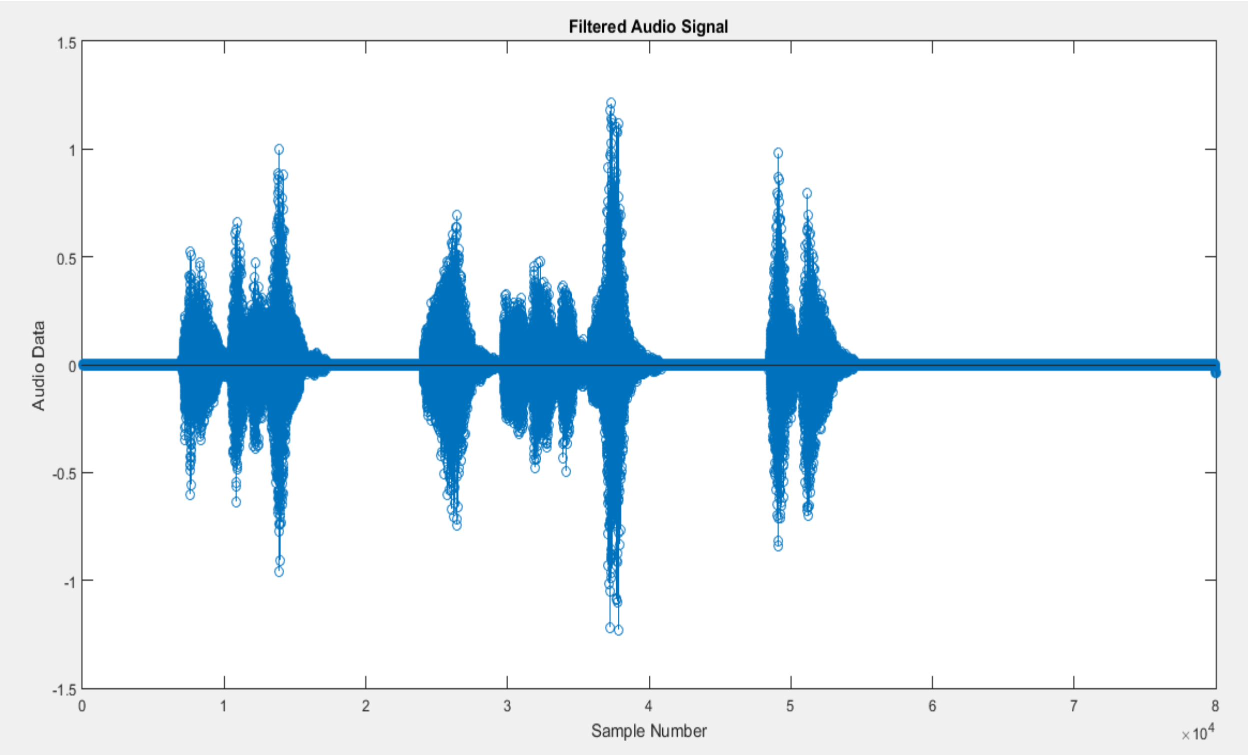
MAGNITUDE AND PHASE RESPONSE OF DESIGNED FILTER (IIR)



POLE-ZERO PLOT OF THE DESIGNED IIR FILTER



FILTERED AUDIO SIGNAL USING IIR FILTER



7. Conclusion

A noisy signal was given having background noise of sinusoidal nature. The signal has been successfully de-noised using FIR and IIR filters. The noise frequencies were identified by extracting a chunk of the signal containing only the noise and then taking its DFT. The noise frequencies were found to be at 367 Hz and 656 Hz. To eliminate the noise frequencies, a bandstop FIR filter was designed using hamming window and a notch IIR filter was designed. Both the filters successfully de-noised the noisy signal. The noise frequencies were rejected and all the other frequencies were allowed to pass through. The audio can be clearly heard after filtering. It can be observed that the order of the IIR filter is very less than that of the FIR filter. Also, the IIR and FIR filters de-noised the signal equally good. So, IIR filter is a better choice for this application.

8. References

[1] Nishan Singh, Dr. Vijay Laxmi, 'Audio Noise Reduction from Audio Signals and Speech Signals'

[2] Chakraborty Subhadeep,etl." Design of IIR Digital Highpass Butterworth Filter using Analog to Digital Mapping Technique" International Journal of Computer Applications (0975 – 8887) Volume 52 – No. 7, August 2012.

[3] http://site.iugaza.edu.ps/mbohisi/files/2017/02/lsb9-dsp.pdf

[4] Prajoy Podder Tanvir Zaman Khan Mamdudul Haque KhanComparative Performance Analysis of Hamming, Hanning and Blackman Window, International Journal of Computer Applications (0975 – 8887) Volume 96– No.18, June 2014

[5] Aniket Kumar, Mamta, 'Comparison of Different Types of IIR Filters', International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 5, Issue 2, February 2016

[6] Audio Processing with MatLab (iastate.edu)

[7] FIR Filter Basics

[8] Notch Filter | 5+ Important Applications and Types (lambdageeks.com)

[8] types-of-digital-filters - MikroElektronika