Contents

1. Introduction:-

In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range.

The following blockdiagram illustrates the basic idea -

Fig(1)

There are two main kinds of filter, analog and digital. An analog filter uses analog electronic circuits made up from components such as resistors, capacitors and op amps to produce the required filtering effect, the signal being filtered is an electrical voltage or current which is the direct analogue of the physical quantity (e.g. a sound or video signal or transducer output) involved.

A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialized DSP chip. The analog input signal must first be sampled and digitized using an analog to digital converter. The processor, carries out numerical calculations on sampled data. These calculations typically involve multiplying the input values by constants and adding the products together. If necessary, the results of these calculations, which now represent sampled values of the filtered signal, are output through a DAC (digital to analog converter) to convert the signal back to analog form.

Note that in a digital filter, the signal is represented by a sequence of numbers, rather than a voltage or current.

- **Advantages of using digital filters**

The following list gives some of the main advantages of digital over analog filters.

1. A digital filter is programmable, i.e. its operation is determined by a program stored in the processor's memory. This means the digital filter can easily be changed without affecting the hardware. An analog filter can only be changed by redesigning the filter circuit.
2. Digital filters are easily designed, tested and implemented on a general-purpose computer or workstation.
3. The characteristics of analog filter circuits are subject to drift and are dependent on temperature. Digital filters do not suffer from these problems, and so are extremely stable with respect both to time and temperature.
4. Unlike their analog counterparts, digital filters can handle low frequency signals accurately. As the speed of DSP technology increases, digital filters are being applied to high frequency signals in the RF (radio frequency) domain.

There are four primary types of filters which 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).

- A low-pass filter is a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency.
- A high-pass filter is an electronic filter that passes signals with a frequency higher than a certain cutoff frequency and attenuates signals with frequencies lower than the cutoff frequency
- A band-pass filter or bandpass filter is a device that passes frequencies within a certain range and rejects frequencies outside that range.
- Band-stop filter or band-rejection filter is a filter that passes most frequencies unaltered, but attenuates those in a specific range to very low levels.

2. Objectives:-

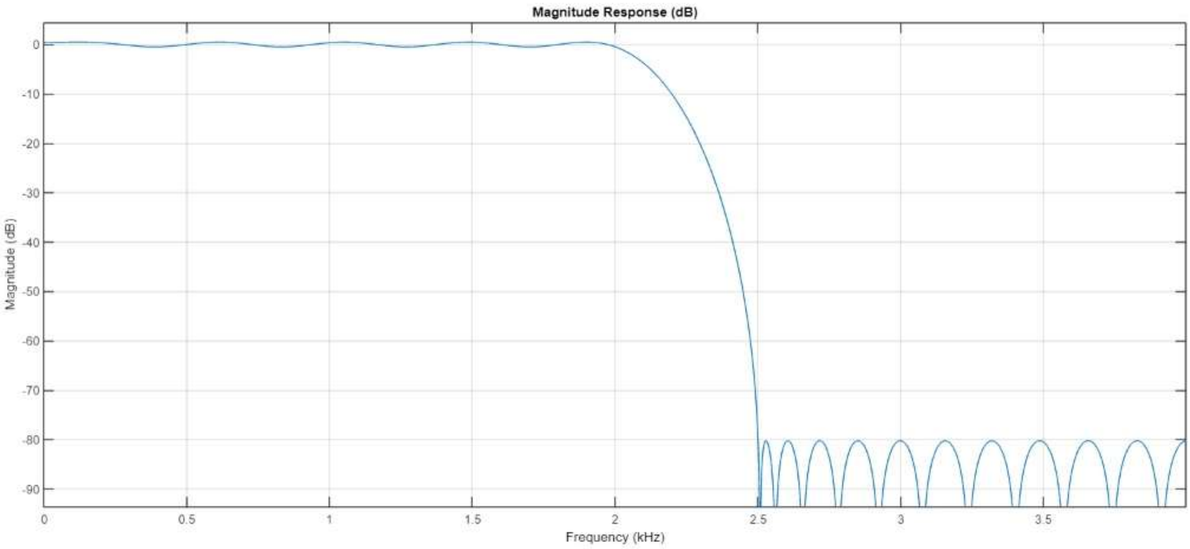
a) **To design a low pass filter.**
A low pass filter is a circuit that only passes signals below its cutoff frequency while attenuating all signals above it. It is the complement of high-pass filter which passes signal above its cutoff frequency.

b) **To remove noise from the audio signal using low pass filter.**
We have designed a low pass filter using filter designer. We have passed an audio frequency signal with noise added to it through our filter. It will filter the noise and give the original signal back. After the filtration, we are plotting its frequency domain graph.

3. Theoretical Background:-

- **LOW PASS FILTER**

A low-pass filter is a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency. The exact frequency response of the filter depends on the filter design. The filter is sometimes called a high-cut filter in audio applications.
The gain of the low pass filter is inversely proportional to the frequency. If the frequency of an input signal increases, the gain of the circuit decreases and also becomes zero at the transition band end-stage.



Fig(2)

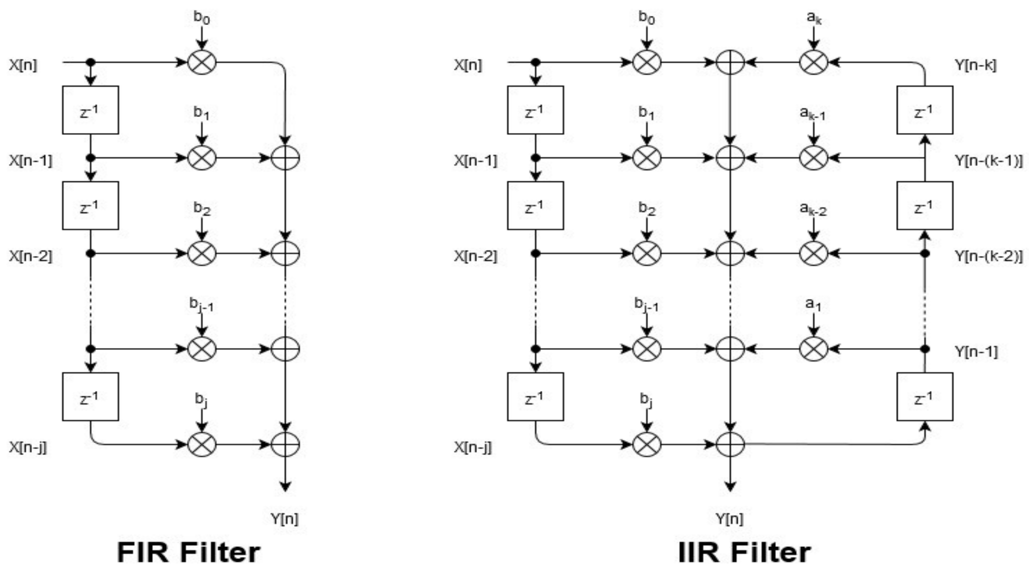
- Variables used in code :-
  - L,N - Length of signal- The duration of the signal in the unit of samples is the same as the size() of the signal.

    N = size(x,1)
  - f – Frequency Range- Is a value that expresses how much of a cycle of a sinusoidal wave is represented by a sample.

    f = Fs*(0:N-1)*N;
  - T – Sampling Time- In signal processing, sampling is the reduction of a continuous-time signal to a discrete-time signal. A common example is the conversion of a sound wave to a sequence of "samples".

    T = 1/Fs

- **Additive white Gaussian noise (AWGN)** - is a basic noise model used in information theory to mimic the effect of many random processes that occur in nature. The modifiers denote specific characteristics: Additive because it is added to any noise that might be intrinsic to the information system.
- **Signal to Noise Ratio(SNR)** - In analog and digital communications, a signal-to-noise ratio, often written S/N or SNR, is a measure of the strength of the desired signal relative to background noise (undesired signal).
- **IIR filters** – They are one of two primary types of digital filters used in Digital Signal Processing (DSP) applications (the other type being FIR). "IIR" means "Infinite Impulse Response." The impulse response is "infinite" because there is feedback in the filter; if you put in an impulse (a single "1" sample followed by many "0" samples), an infinite number of non-zero values will come out (theoretically.)
- **Disadvantages of IIR filters** - They are more susceptible to problems of finite-length arithmetic, such as noise generated by calculations, and limit cycles. (This is a direct consequence of feedback: when the output isn't computed perfectly and is fed back, the imperfection can compound.) They are harder (slower) to implement using fixed-point arithmetic. They don't offer the computational advantages of FIR filters for multirate (decimation and interpolation) applications.
- **FIR (Finite Impulse Response) filter** – It is a finite-length unit impulse response filter, also known as a non-recursive filter, which is the most basic element in a digital signal processing system. It can guarantee arbitrary amplitude-frequency characteristics while having strict linear phase-frequency characteristics, and its unit sampling response is finite, so the filter is a stable system.
- **Advantages of FIR filters** - The advantage of FIR is that the accuracy can be increased indefinitely (under the premise of sufficient computing power), and there is no phase accuracy problem for the IIR filter, which is a relatively high-end solution.



Fig(3)

- **Fast Fourier transform (FFT) -** It is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT). Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa.

Discrete Fourier Transform

$$\hat{F}\left(\tfrac{m}{L}\right) \equiv F(m) = \sum_{n=0}^{N-1} f(n) e^{-i2\pi \frac{nm}{2BL}}$$

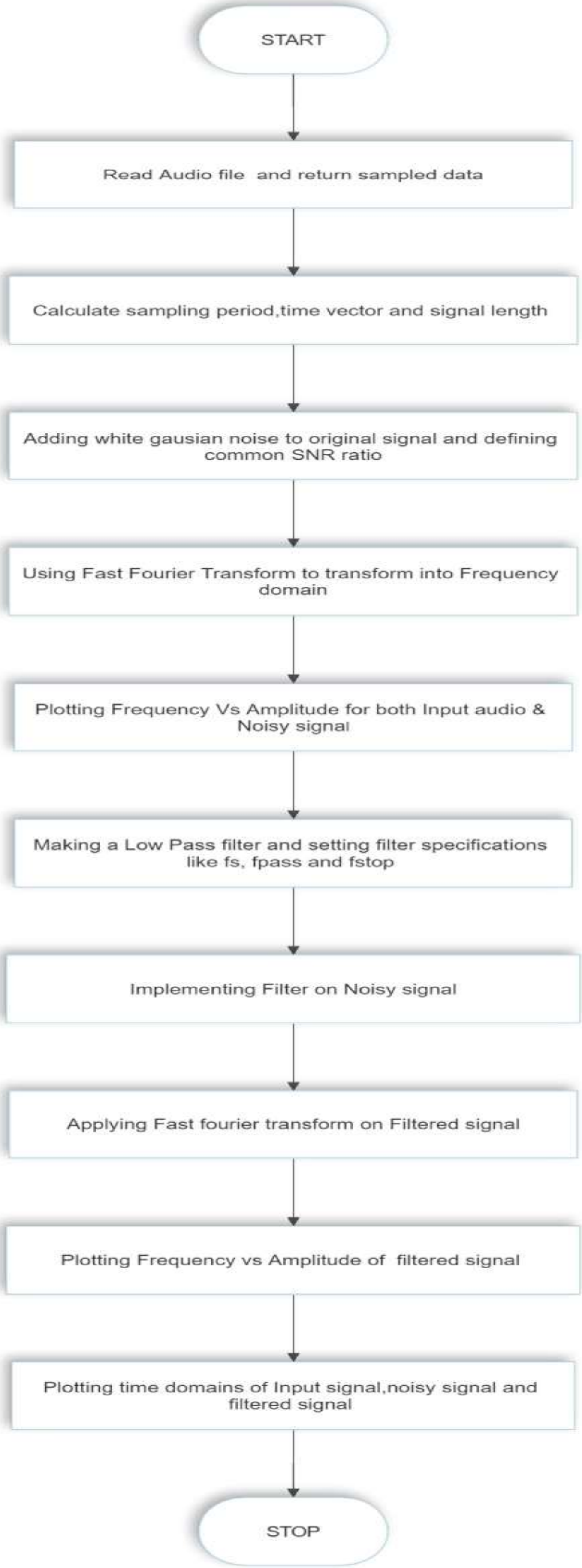
$2BL = N =$ number of samples

Discrete Fourier Transform (DFT):

$$F(m) = \sum_{n=0}^{N-1} f(n) e^{-i2\pi \frac{nm}{2BL}} = \sum_{n=0}^{N-1} f(n) e^{-i2\pi \frac{nm}{N}}$$

Fig(4)

4. Flow chart:-


START
Read Audio file and return sampled data
Calculate sampling period,time vector and signal length
Adding white gausian noise to original signal and defining common SNR ratio
Using Fast Fourier Transform to transform into Frequency domain
Plotting Frequency Vs Amplitude for both Input audio & Noisy signal
Making a Low Pass filter and setting filter specifications like fs, fpass and fstop
Implementing Filter on Noisy signal
Applying Fast fourier transform on Filtered signal
Plotting Frequency vs Amplitude of filtered signal
Plotting time domains of Input signal,noisy signal and filtered signal
STOP

5. Code :-

```
%%
clearvars;
clc
close all;
disp('Section-1 Complete')
%%
[x,Fs] = audioread('my_voice_new.wav');
N = size(x,1);
f = Fs*(0:N-1)*N;
T = 1/Fs;
L = N;
t = (0:L-1)*T;
A = 0.02;
disp('Section-2 Complete');
```

Section_1 - Clear the command window.
Section_2 - Inserting input audio file and then assigning all the variables.

```
%%
f_noisy = awgn(x,45);
audiowrite('f_noisy.wav',f_noisy,Fs);
disp('Section-3 Complete');

%%
xn = x + f_noisy;
audiowrite('Noisy-Signal.wav',xn,Fs);
disp('Section-4 Complete');
```

Section_3 – Creating White Guassion Noise with fixed SNR = 45.
Section_ 4 – Adding White Guassion Noise to the input audio signal.

```
%%
[audio_in,audio_freq_sampl]=audioread('my_voice_new.wav');
Length_audio=length(audio_in);
df=audio_freq_sampl/Length_audio;
frequency_audio=-audio_freq_sampl/2:df:audio_freq_sampl/2-df;

figure
FFT_audio_in=fftshift(fft(audio_in))/length(fft(audio_in));
plot(frequency_audio,abs(FFT_audio_in));
title('FFT of Input Audio');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
disp('Section-5 Complete');
```

Section_5 - Applying Fast Fourier Transform (FFT) to input audio signal and plotting the frequency spectrum graph for the same.

```
%%
[audio_in,audio_freq_sampl]=audioread('f_noisy.wav');
Length_audio=length(audio_in);
df=audio_freq_sampl/Length_audio;
frequency_audio=-audio_freq_sampl/2:df:audio_freq_sampl/2-df;

figure
FFT_audio_in=fftshift(fft(audio_in))/length(fft(audio_in));
plot(frequency_audio,abs(FFT_audio_in));
title('FFT of noisy signal');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
disp('Section-6 Complete');
%%
%filterDesigner
load matlab.mat
disp('Section-7 Complete');
```

Section_6 - Applying Fast Fourier Transform (FFT) to noisy audio signal and plotting the frequency spectrum graph for the same.

Section_7 – Designing Low-Pass Filter using filterDesinger tool and saving the coefficients of the filter in workspace and fetching our filter from there.

```
%%
filtered_signal = filter(Hd,f_noisy);
disp('Section-8 Complete');
%%
audiowrite('Output.wav',filtered_signal,Fs);
disp('Section-9 Complete');
%%
[audio_in,audio_freq_sampl]=audioread('Output.wav');
Length_audio=length(audio_in);
df=audio_freq_sampl/Length_audio;
frequency_audio=-audio_freq_sampl/2:df:audio_freq_sampl/2-df;

figure
FFT_audio_in=fftshift(fft(audio_in))/length(fft(audio_in));
plot(frequency_audio,abs(FFT_audio_in));
title('FFT of Output Audio');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
disp('Section-10 Complete');
```

Secton_8 – Using Low-Pass Filter we are filtering the noisy audio signal.

Section_9 - writing a matrix of filtered audio data , with sample rate Fs(8000Hz) to a file called ‘Output.wav’.

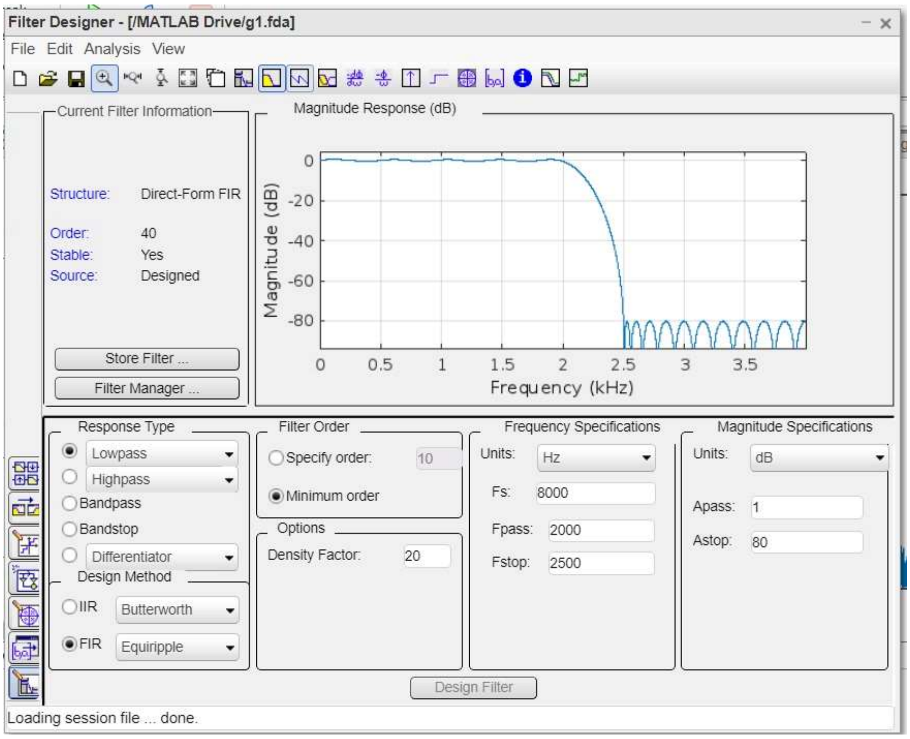
Section_10 - Applying Fast Fourier Transform (FFT) to the filtered signal and plotting the frequency spectrum graph for the same.

```
%%
figure
subplot(3,1,1)
stem(t, x);title('Original: Time-domain'); xlabel('time(seconds)');
subplot(3,1,2)
stem(t , f_noisy , 'r');title('Noisy: Time-domain'); xlabel('time(seconds)');
subplot(3,1,3)
stem(t , filtered_signal , 'g');title('After processing: Time-domain'); xlabel('time(seconds)');

disp('Section-11 Complete')
%%
%sound(x,Fs)
%sound(f_noisy,Fs)
%sound(filtered_signal,Fs)
```

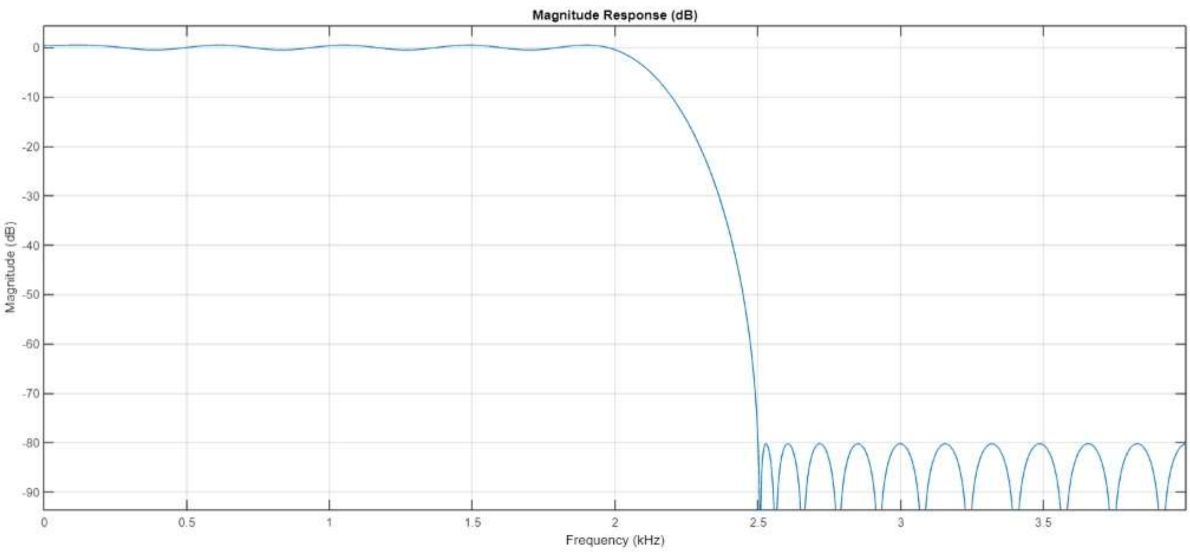
Section_11- Plotting the Time Domain graph for the input audio signal, the noisy signal and the filtered signal for comparison.

6. Results:-



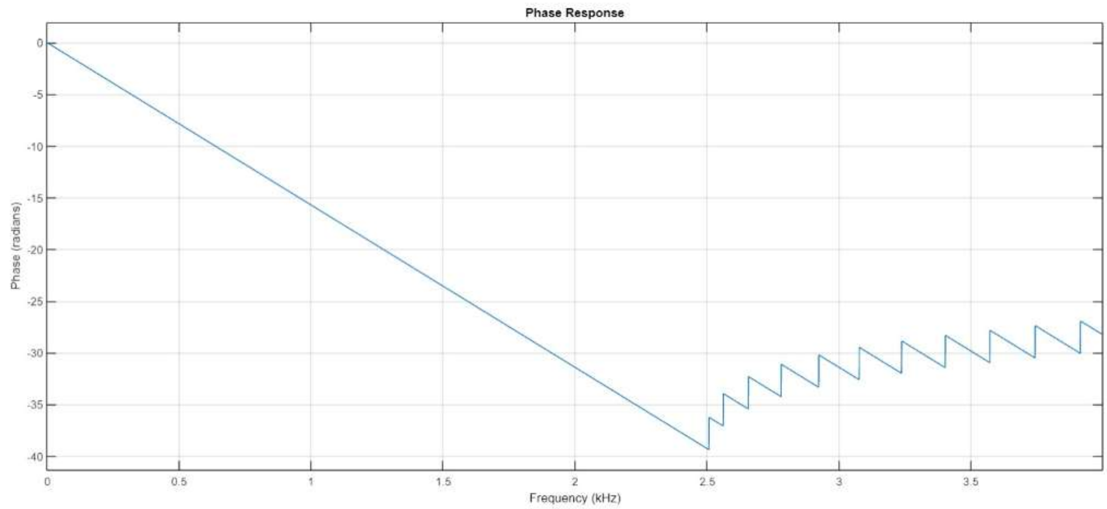
Fig(5)

In the above figure we have designed a Low-pass filter using filterDesigner tool by setting the respective frequencies as Fs= 8000Hz, stop band =2500Hz and pass band=2000Hz . After designing the filter we get its order=40 , stability ,structure, magnitude response and phase response graph.



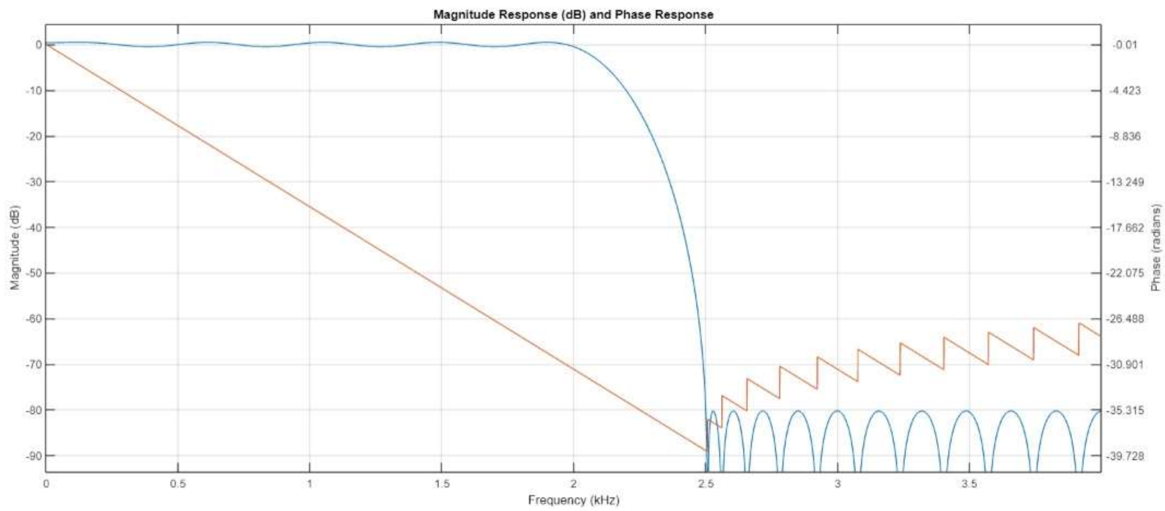
Fig(6)

This is the magnitude response plot of the filtered signal using Low-pass filter design.



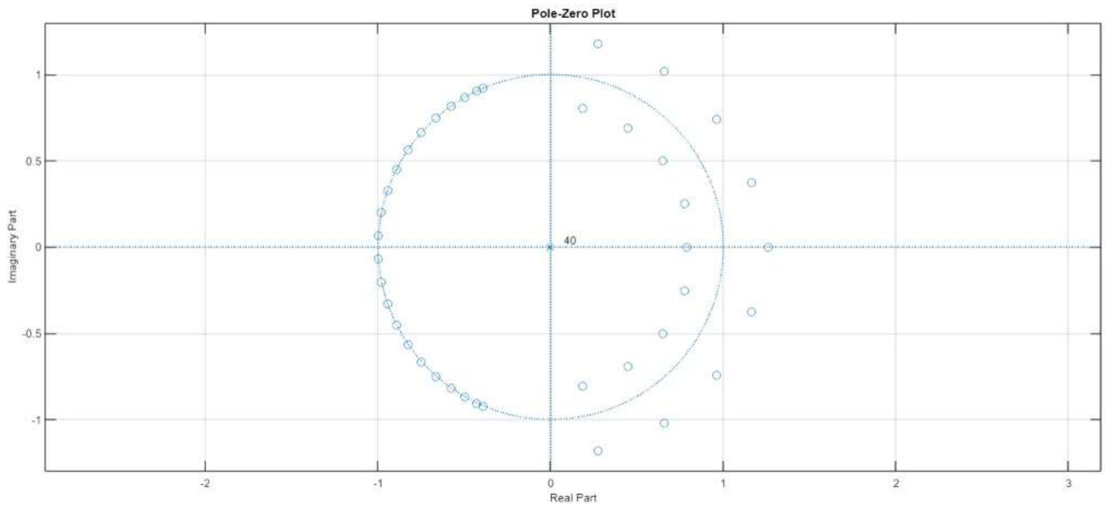
Fig(7)

Above is the phase response plot of the filtered signal using Low-pass filter design.



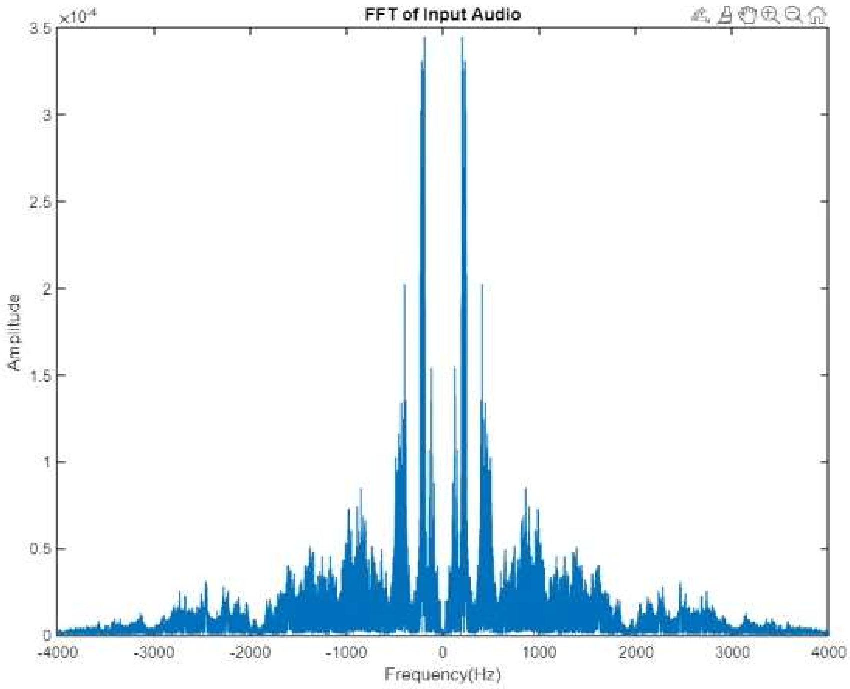
Fig(8)

Above fig. is the magnitude and phase response of the signal is shown after the filtering operation is carried out . This plot helps us to understand at which frequency is cutoff from the spectrum after using the Low-pass filter.



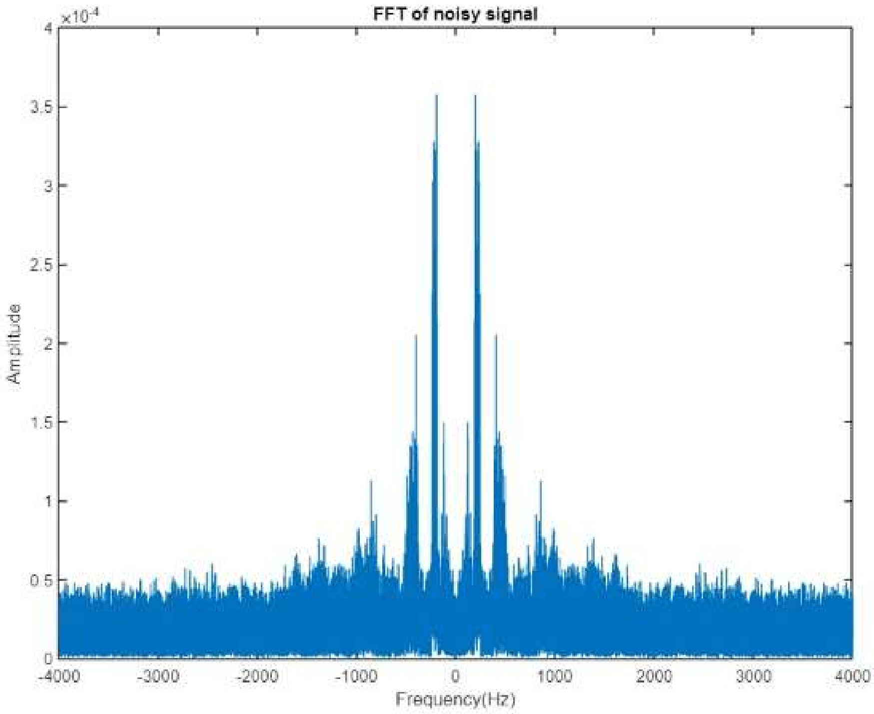
Fig(9)

Above image is the pole-zero plot of the designed filter.



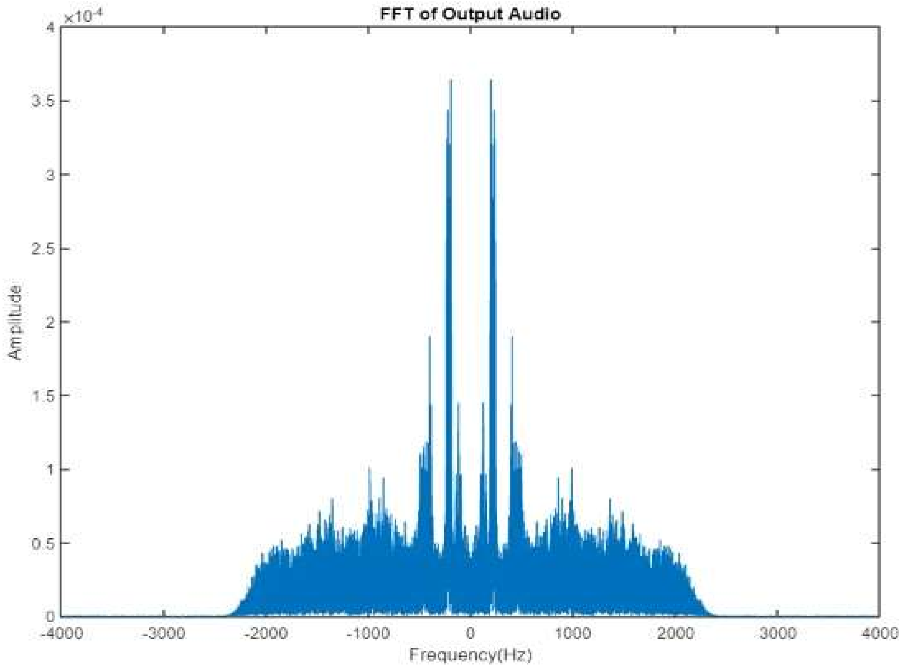
Fig(10)

This is the frequency spectrum graph of FFT (fast Fourier transform) of the input signal where x axis indicates frequency in hertz(Hz) and y axis indicates amplitude of the signal.



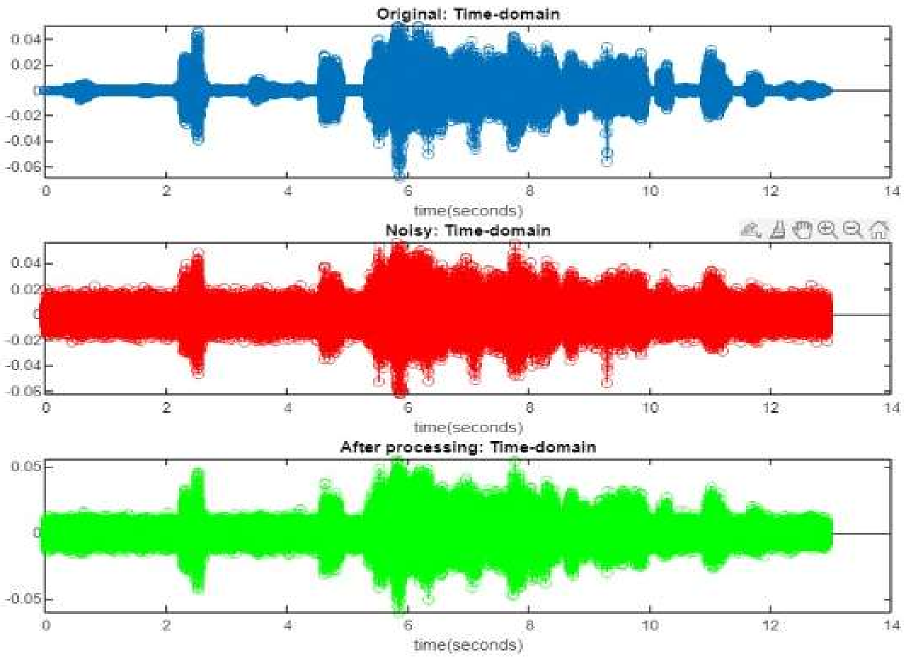
Fig(11)

This is the frequency spectrum graph of FFT (fast Fourier transform) of the noisy signal where x axis indicates frequency in hertz(Hz) and y axis indicates amplitude of the signal . As we can see that the White Guassian noise is added and is present in complete input signal.



Fig(12)

This is the frequency spectrum graph of FFT (fast Fourier transform) of the output signal where x axis indicates frequency in hertz(Hz) and y axis indicates amplitude of the signal. This graphs indicates that noise gets cutoff after stop band frequency (i.e.2500Hz).



Fig(13)

This graph gives the Time Domain Analysis of the Input signal, noisy signal and output signal after filtering.

Conclusion- In this project we have designed a Low-pass filter using filterDesigner tool with sample frequency (Fs)=8000Hz , pass band frequency = 2000Hz and cut-off frequency = 2500Hz. Then we have added White Guassian noise to the input audio creating noisy signal which we filtered and got our output signal with minimum noisy signal in it.

Hence, we have successfully designed a low pass filter which removes high frequency noisy signal.

7. Applications: -

The low pass filter applications include the following..

- Used in audio applications as it removes the high frequency signal from the original signal and give us much clear sound.
- The filters employed in biomedical system are used for sensing bioelectrical signals which, typically, are in the range of 1V-100mV while the frequencies are below 100 Hz. At the input, a low-pass filter (LPF) is usually employed in order to limit the frequency band.
- Electronic low-pass filters are used on inputs to subwoofers and other types of loudspeakers, to block high pitches that they cannot efficiently reproduce. Radio transmitter use low-pass filters to block harmonic emissions that might interfere with other communications.
- Used in wave analyzers, audio amplifiers, and equalizers.

8. Limitations and Future scope:-

A. Limitations:-

In this filter we have attenuated all the samples with frequencies higher than our cut-off frequency along with the white gaussian noise that we have added in our input audio signal. But as we know with low pass filter, we can only remove noise with frequencies higher than our cut-off frequency therefore, it will not be able to remove low frequency noise from audio signal.Therefore, for better result we have to increase the order of the filter which results in the increase of hardware thus, the cost increases.

B. Future scope:-

The low pass filter that we have designed with the help of MATLAB can easily remove the high frequency noise from our input signal ,the remaining noise that we were not able to remove single-handedly only by the use of the low pass filter ,can be removed by cascading the low pass and high pass filter.

9. References:

[1] https://in.mathworks.com/help/signal/ref/lowpass.html
[2] https://www.eecs.tufts.edu/~dsculley/tutorial/rc/rc4.html
[3] https://epdf.tips/digital-filters.html
[4] http://www.dave40.co.uk/1/VwSlAr.php?id=15
[5] https://en.wikipedia.org/wiki/Low-pass_filter#Higher_order_passive_filters
[6] https://www.sciencedirect.com/topics/engineering/lowpass-filter
[7] http://www.learningaboutelectronics.com/Articles/Low-pass-filter.php