Contents

1.Introduction:

Digital Signal Processing involves various techniques like filtering , equalization etc for improving the accuracy and reliability of digital communications.

Noise is undesired modifications introduced in a source signal that occur during the capture , storage , transmission or processing of its information. Noise could be added due to temperature , Electromagnetic or Radio Frequency Interference etc . Filtering , averaging , smoothing , adaptive filtering and signal processing are some common technique employed for noise removal. Noise is of four types based on the frequency range – continuous , intermittent , impulsive and low frequency noise.

Gaussian noise also known as white noise is a type of statistical noise which follows gaussian distribution, is characterised by random nature . It has a probability distribution which is bell shaped curve when plotted .
Filters are devices that supress unwanted components from a signal . They are either analog (hardware) or digital(software). The most basic types of filters are Low pass filter , High Pass Filter , Band Pass filter and Band Stop filter.

Advantages of using digital filter which make preferred for wide usage are:
i. These are programmable.
ii. Easy to design, test and implement.
iii. Digital filters are unaffected by hardware problems or shortcomings and are extremely stable.
iv. Unlike analog filters, they can handle low frequency accurately.

Low pass filters allow signals with a frequency lower than certain cutoff frequency to pass through while reducing or attenuating the amplitudes of signals with frequency higher than cutoff frequency.

High pass filter allows signals with a frequency higher than a certain frequency to pass through while attenuating lower frequency signals . It performs the opposite functions of the low pass filter.

Band Pass Filter allows signal within certain frequency range to pass through. This range is called the passband .The frequency other than the passband are attenuated.

Band Stop Filter also called notch filter allows signal with most frequencies to pass through except a specific frequency range which is blocked. This range is called the stopband.

2.Objectives:

a) To design a lowpass filter:

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

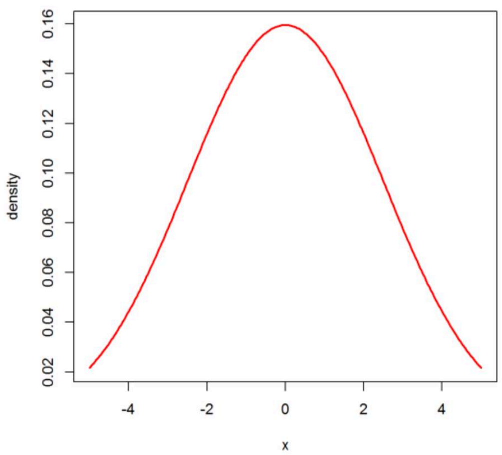
b) To remove noise from audio signal using Lowpass filter:

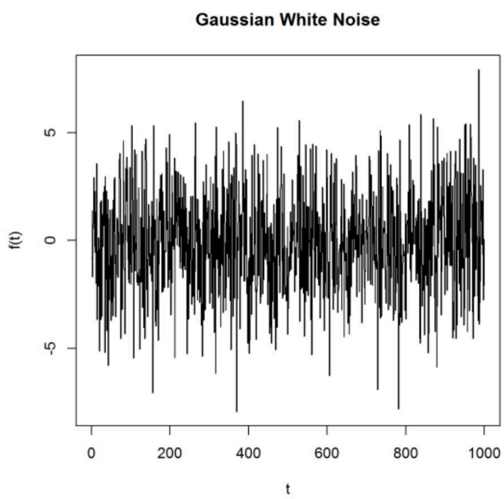
We have designed a low pass filter using filter designer. We have passed an audio frequency signal with noise added to it through our filter. It will filter the noise and give the original signal back. After the filtration, we are plotting its frequency domain graph.

3.Theoretical Background:

In signal processing, **White Noise** is a random signal having equal intensity at different frequencies, giving it a constant power spectral density.

Gaussian noise is inherently random, each noise sample is independent of the other .The probability distribution of gaussian noise forms bell shaped curve and majority of values concentrated around the mean . This type of noise is smooth , lacking abrupt transitions between values. It is common in natural and man made sound .



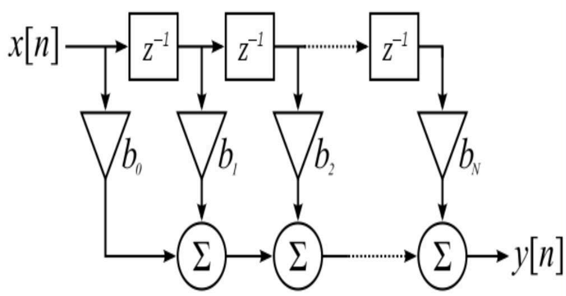


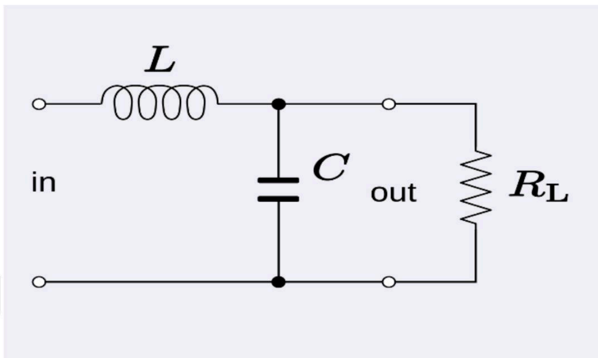
The white gaussian noise can be mimicked in MATLAB using the ‘awgn’ function which takes SNR and vector signal as input and creates a new vector signal consisting of the white gaussian noise.

SNR is an important parameter that affects the performance and quality of systems that process or transmit signals. A high SNR means that the signal is clear and easy to detect or interpret, while a low SNR means that the signal is corrupted or obscured by noise and may be difficult to distinguish or recover. SNR can be improved by various methods, such as increasing the signal strength, reducing the noise level, filtering out unwanted noise, or using error correction techniques.

IIR Filters incorporate feedback in their design allowing them to have infinite impulse response. They have high computational efficiency but may become instable .

FIR Filters are used to overcome the stability problem.
The advantages of FIR filters are :
i. They are always stable.
ii. The design methods are generally linear.
iii. They can be realised efficiently in hardware.



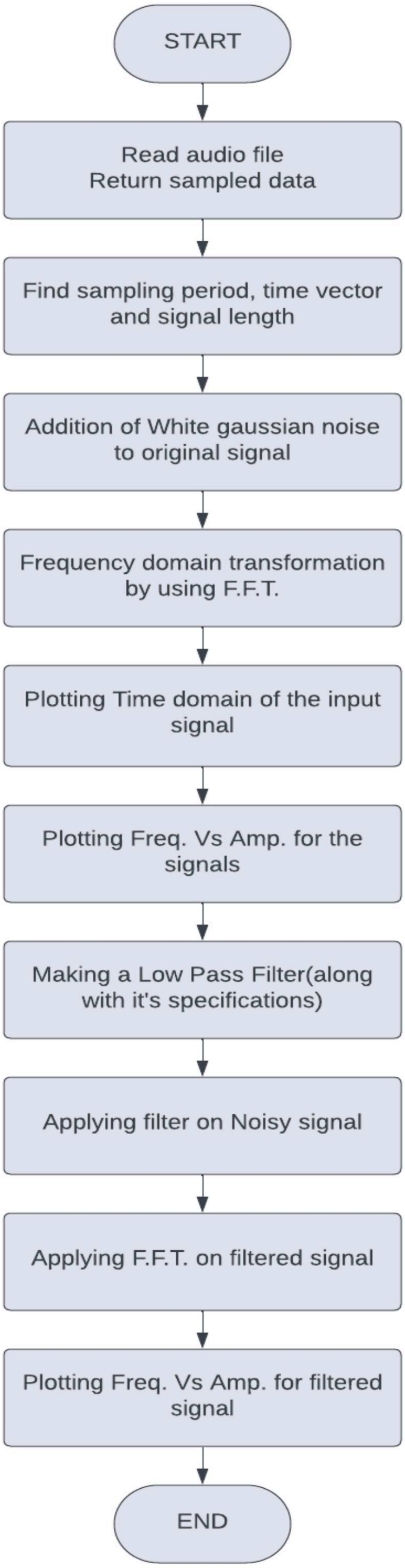


Low Pass Filter passes the signal with frequency lower than the band pass frequency set and stops or supresses the signals with frequency above the band pass frequency. Low-pass filters have applications such as anti-aliasing, reconstruction, and speech processing, and can be used in audio amplifiers, equalizers, and speakers.
Low-pass filters can also be used in conjunction with high-pass filters to form bandpass, band-stop, and notch filters.

Order of filter affects the transition band as higher the order of the filter will be, the narrower will the transition band will be thus increasing the efficiency of the filter and accuracy of the output.

The **Fast Fourier Transform** (FFT) is an important measurement method in the science of audio and acoustics measurement. It converts a signal into individual spectral components and thereby provides frequency information about the signal. FFT is an optimized algorithm for the implementation of the "Discrete Fourier Transformation" (DFT). A signal is sampled over a period of time and divided into its frequency components. In signal processing it forms the basis of frequency domain analysis (spectral analysis) and is used for signal filtering, spectral estimation, data compression, and other applications.

4.Flow chart:


START
Read audio file
Return sampled data
Find sampling period, time vector and signal length
Addition of White gaussian noise to original signal
Frequency domain transformation by using F.F.T.
Plotting Time domain of the input signal
Plotting Freq. Vs Amp. for the signals
Making a Low Pass Filter(along with it's specifications)
Applying filter on Noisy signal
Applying F.F.T. on filtered signal
Plotting Freq. Vs Amp. for filtered signal
END

5.Code:

```
%%
clearvars;
clc
close all;
disp('Section-1 Complete')
%figure,plot(x)
%%
[x,Fs] = audioread("C:\Users\seema\Downloads\Gaussian-Noise-Removal-Using-Lowpass-
Filter-main\Gaussian-Noise-Removal-Using-Lowpass-Filter-main\dsp+new.wav");
%sound(x,Fs);
N = size(x,1);% length of signal
f = Fs*(0:N-1)*N;% frequency range
T = 1/Fs;% sampling time
L = N; % length of signal
t = (0:L-1)*T; %time vector

disp('Section-2 Complete');
%%
noisy1 = awgn(x,30);
noisy2 = awgn(x,25);
noisy3 = awgn(x,75);
audiowrite('noisy1.wav',noisy1,Fs);
audiowrite('noisy2.wav',noisy2,Fs);
audiowrite('noisy3.wav',noisy3,Fs);
disp('Section-3 Complete');
%%
[audio,audio_freqs]=audioread("C:\Users\seema\Downloads\Gaussian-Noise-Removal-
Using-Lowpass-Filter-main\Gaussian-Noise-Removal-Using-Lowpass-Filter-
main\dsp+new.wav");
Length=length(audio);
df=audio_freqs/Length;
freq_audio=0:df:audio_freqs/2-df;

disp('Section-4 Complete');
%%
%sound(noisy1, Fs);
xn1 = x + noisy1;
audiowrite('Noisy-Signal1.wav',xn1,Fs);
%sound(xn1, Fs);

[audio1,audio_freqs1]=audioread('noisy1.wav');
Length1=length(audio1);
df=audio_freqs1/Length1;
freq_audio1=0:df:audio_freqs1/2-df;
```

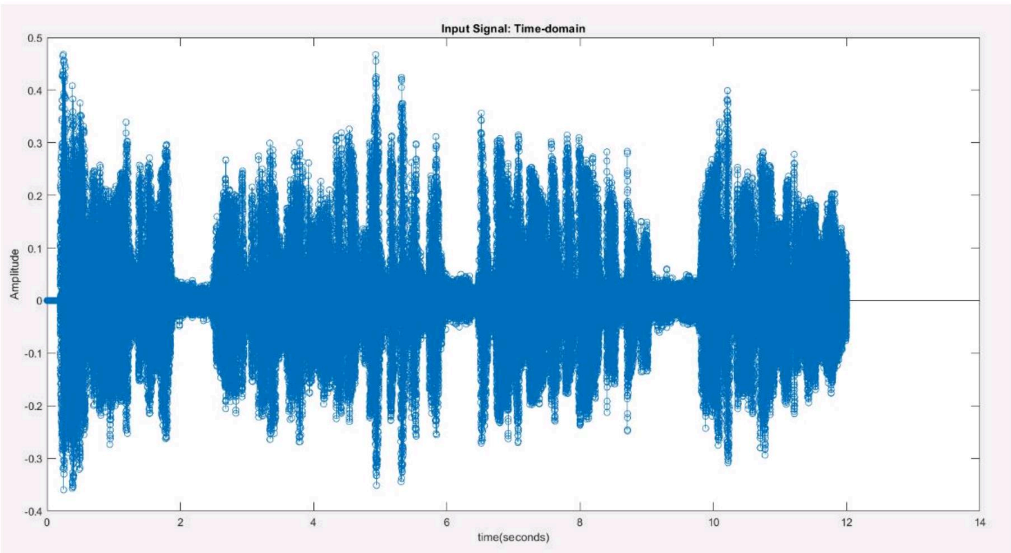
```
%sound(noisy2, Fs);
xn2 = x + noisy2;
audiowrite('Noisy-Signal2.wav',xn2,Fs);
%sound(xn2, Fs);

[audio2,audio_freqs2]=audioread('noisy2.wav');
Length2=length(audio2);
df=audio_freqs2/Length2;
freq_audio2=0:df:audio_freqs2/2-df;
%
xn3 = x + noisy3;
audiowrite('Noisy-Signal3.wav',xn3,Fs);

[audio3,audio_freqs3]=audioread('noisy3.wav');
Length3=length(audio3);
df=audio_freqs3/Length3;
freq_audio3=0:df:audio_freqs3/2-df;
%
figure,
stem(t , x);
title('Input Signal: Time-domain'); xlabel('time(seconds)');ylabel('Amplitude');
%
figure,
FFT_audio=(fft(audio))/length(fft(audio));
subplot(2,2,1);
plot(freq_audio,abs(FFT_audio(1:length(freq_audio))));
title('FFT of Input Audio');
xlabel('Frequency(Hz)');
ylabel('Amplitude');

FFT_audio1=(fft(audio1))/length(fft(audio1));
subplot(2,2,3);
plot(freq_audio1,abs(FFT_audio1(1:length(freq_audio1))));
title('FFT of noisy signal 1(30 SNR)');
xlabel('Frequency(Hz)');
ylabel('Amplitude');

FFT_audio2=(fft(audio2))/length(fft(audio2));
subplot(2,2,2);
plot(freq_audio2,abs(FFT_audio2(1:length(freq_audio2))));
title('FFT of noisy signal 2(25 SNR)');
xlabel('Frequency(Hz)');
ylabel('Amplitude');

FFT_audio3=(fft(audio3))/length(fft(audio3));
subplot(2,2,4);
plot(freq_audio3,abs(FFT_audio3(1:length(freq_audio3))));
title('FFT of noisy signal 3(75 SNR)');
xlabel('Frequency(Hz)');
ylabel('Amplitude');
```

```
disp('Section-5 Complete');
%%
load("C:\Users\seema\Downloads\Gaussian-Noise-Removal-Using-Lowpass-Filter-main\Gaussian-Noise-Removal-Using-Lowpass-Filter-main\workspace.mat");
filtered_signal1 = filter(Hd1,xn1);
filtered_signal2 = filter(Hd1,xn2);
filtered_signal3 = filter(Hd1,xn3);
filtered_signal4 = filter(Hd1,x);
disp('Section-6 Complete');

%%
audiowrite('Output1.wav',filtered_signal1,Fs);
audiowrite('Output2.wav',filtered_signal2,Fs);
audiowrite('Output3.wav',filtered_signal3,Fs);
audiowrite('Output.wav',filtered_signal4,Fs);

disp('Section-7 Complete');
%%
[audio1,audio_freqs1]=audioread('Output1.wav');
Length1=length(audio1);
df=audio_freqs1/Length1;
freq_audio1=0:df:audio_freqs1/2-df;

[audio2,audio_freqs2]=audioread('Output2.wav');
Length2=length(audio2);
df=audio_freqs2/Length2;
freq_audio2=0:df:audio_freqs2/2-df;

[audio3,audio_freqs3]=audioread('Output3.wav');
Length3=length(audio3);
df=audio_freqs3/Length3;
freq_audio3=0:df:audio_freqs3/2-df;

[audio,audio_freqs]=audioread('Output.wav');
Length=length(audio);
df=audio_freqs/Length;
freq_audio=0:df:audio_freqs/2-df;

figure,
FFT_audio=(fft(audio))/length(fft(audio));
subplot(2,2,1);
plot(freq_audio,abs(FFT_audio(1:length(freq_audio))));
title('Filtered Input Audio');
xlabel('Frequency(Hz)');
ylabel('Amplitude');

FFT_audio1=(fft(audio1))/length(fft(audio1));
subplot(2,2,3);
plot(freq_audio1,abs(FFT_audio1(1:length(freq_audio1))));
title('Filtered Noisy Signal 1 (30 SNR)');
```

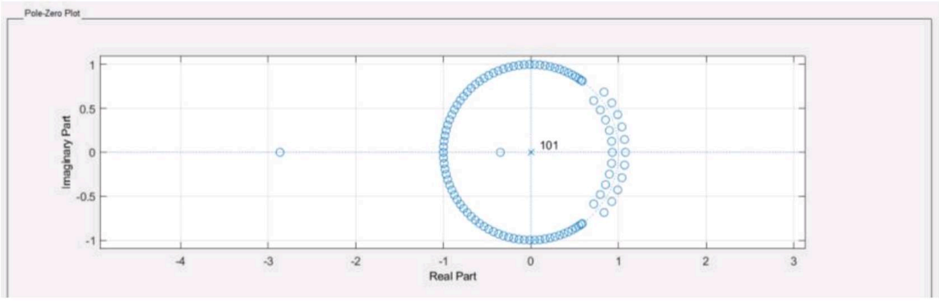
```
xlabel('Frequency(Hz)');
ylabel('Amplitude');

FFT_audio2=(fft(audio2))/length(fft(audio2));
subplot(2,2,2);
plot(freq_audio2,abs(FFT_audio2(1:length(freq_audio2))));
title('Filtered Noisy Signal 2 (25 SNR)');
xlabel('Frequency(Hz)');
ylabel('Amplitude');

FFT_audio3=(fft(audio3))/length(fft(audio3));
subplot(2,2,4);
plot(freq_audio3,abs(FFT_audio3(1:length(freq_audio3))));
title('Filtered Noisy Signal 3 (75 SNR)');
xlabel('Frequency(Hz)');
ylabel('Amplitude');

disp('Section-8 Complete');
```

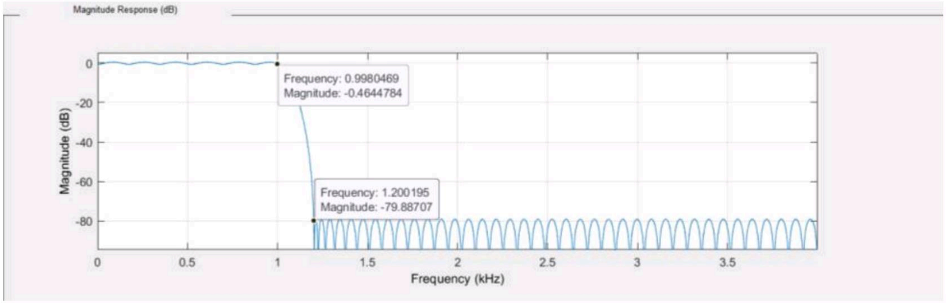
6.Results:

Input signal is a normal recorded audio. The maximum amplitude peaks lie within the frequency range of 0 to 1100 Hz. The specifications of the input audio are :
i. Sampling Frequency = 8000 Hz
ii. Duration = 12 seconds



Filter used here FIR Equiripple LPF Filter of specifications:
i. Order : 101
ii. Band pass Frequency:1000 Hz
iii. Band Stop Frequency : 1200 Hz
iv. Band Pass Gain: 1dB
v. Band Stop Gain: 80 dB

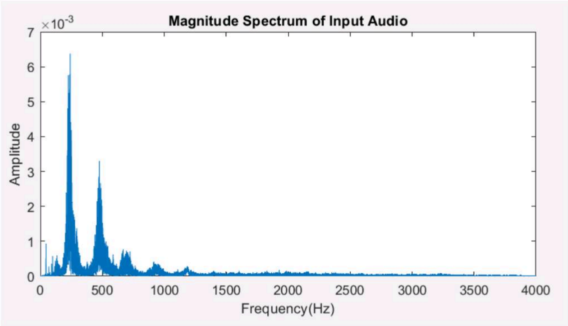


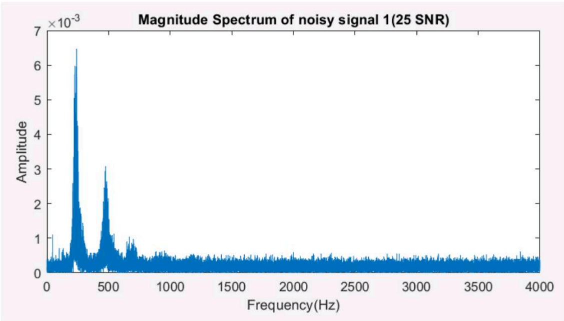


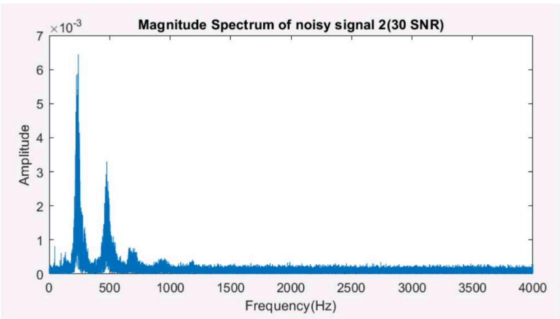
The noise used is White Gaussian noise that is added to the original audio with three different values of SNR : 25 , 30 and 75 and thus creating three different noisy audios.
The number of samples in all the audios(input and noisy) is 96004.

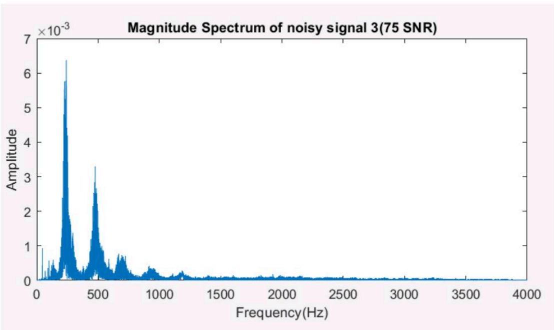
After passing the input signal and the noisy signals through the low pass filter of the above specifications the below given magnitude spectrums are obtained.
It is observed that after filtering, the signals with frequency above stopband frequency(1200 Hz) are attenuated and the filtered signals are obtained adhering with standard Low Pass Filter characteristics .

INPUT

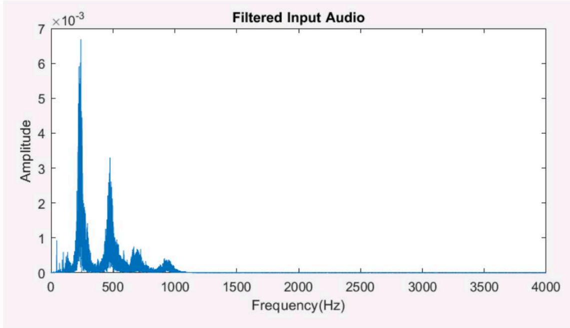


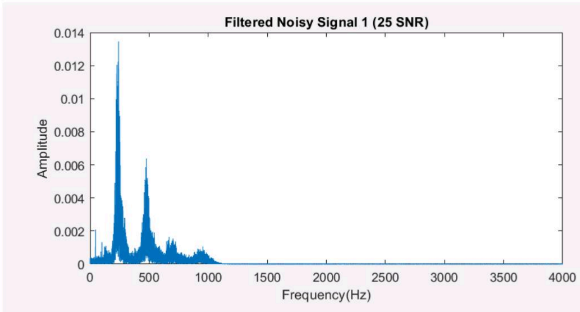


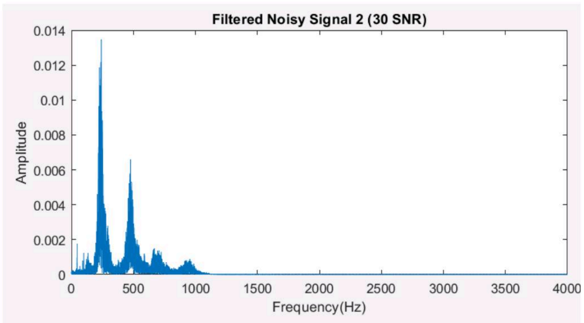


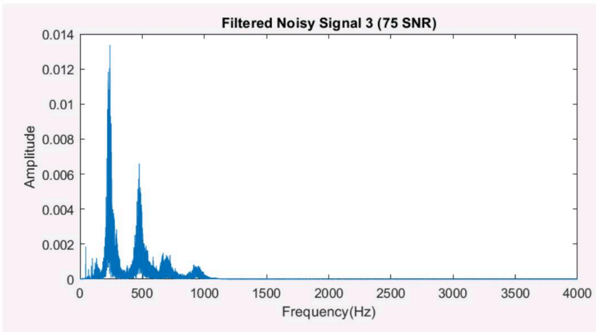


OUTPUT









7.Applications:

1. **Medical Imaging:**
In medical imaging, such as MRI or CT scans, images may be corrupted by Gaussian noise. Low-pass filtering can help improve the clarity of these images by reducing noise while preserving important details, making it easier for doctors to interpret the results.

2. Communication Systems:
In wireless communication systems, signals can be affected by noise during transmission. Applying low-pass filters helps mitigate high-frequency noise, improving the signal quality and reducing errors in communication.

3. Signal Processing in IoT Devices:
In Internet of Things (IoT) devices, sensor data may be affected by noise during acquisition. Low-pass filtering can be applied to sensor readings to improve the accuracy of data used for decision-making in applications like environmental monitoring or industrial automation.

8.Limitations and Future scope:

Limitations:

1. The Low Pass Filter employed for signal filtering attenuates parts of input or original signal lying above the stop band frequency.
2. Noise of frequency lying below Band Stop frequency is not removed and persists in the filtered audio.
3. It is incapable of real time audio filtering as it is rigid .
4. These filters might have trouble with time-varying non-stationary noise, and their linear design might not be sufficient for nonlinear signal-noise interactions.

Future Scope:
1. Future work might concentrate on creating adaptive filters that react dynamically to changing noise levels and incorporate machine learning methods for improved noise pattern identification.
2. It is anticipated that real-time computing power will become more and more important, especially for applications like virtual reality and live broadcasting. Investigating nonlinear filtering techniques and working with specialised hardware could improve the effectiveness of noise reduction even more.

9.References:

1. https://github.com/
2. https://in.mathworks.com/
3. https://en.wikipedia.org/wiki/e