Amrita Vishwa Vidyapeetham

Amritapuri Campus

DIGITAL COMMUNICATION LAB (15ECE385)

MINI PROJECT REPORT ON

REMOVING RANDOM NOISE FROM AUDIO SIGNAL

SUBMITTED BY

GROUP - 9

P.Narasimha (AM.EN.U4ECE18139)

K.Avinash Hegde (AM.EN.U4ECE18128)

V.M Varun Nivas (AM.EN.U4ECE18167)

N.Chaitanya Sai Kumar (AM.EN.U4ECE18136)

M.V.Tejendra Prasad (AM.EN.U4ECE18512)

GVR Yeshwanth (AM.EN.U4ECE18117)

UNDER THE GUIDANCE OF

Asst. Prof. Anuraj K

Asst. Prof. Poorna SS

OBJECTIVE:

To filter out random noise from a dual channel audio signal.

Abstract: When an audio signal has to be decoded or understood at the receiver end, it has to be free of noise. As noise less transmission in practically not possible, this project intends to propose a sequential procedure to make the received signal noise free to the maximum extent possible. Audio filtering plays a key role in many applications like speech recognition. This project work assumes the noise to be random in most of the cases. The proposed sequential method to filter out noise was verified by implementing in Matlab R2019a.

Flow of Program:

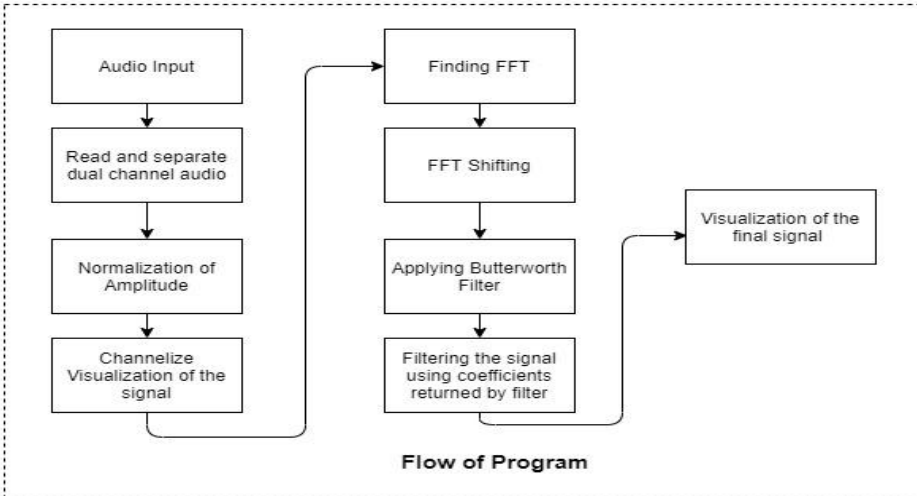


Figure1: Flow of the program used to remove noise from audio signal

Theory:

The corrupted signal (signal with noise) as shown in figure2 in page4, is random in nature. So initially the signal is transformed to frequency domain by means of fast Fourier transform. We chose Butterworth filter because of ripple less smooth transition as shown in figure2.

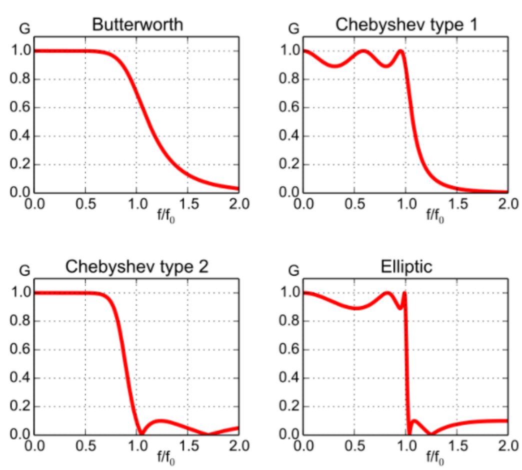


Figure2: Gain vs f/f0 curve for different filters

Butterworth Filter

To reduce the background noise and suppress the interfering signals by removing some frequencies is called filtering. There are various types of filters which are classified based on various criteria such as linearity-linear or nonlinear, time-time variant or time invariant, analog or digital, active or passive, and so on. In linear continuous time filters Butterworth filter is one of the prominent filters.

A Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the passband. Hence the Butterworth filter is also known as “maximally flat magnitude filter”. The frequency response of the Butterworth filter is flat in the passband (i.e. a band pass filter) and roll-offs towards zero in the stopband. The rate of roll-off response depends on the order of the filter. The number of reactive elements used in the filter circuit will decide the order of the filter. In the case of Butterworth filter only capacitors are used. So, the number of capacitors will decide the order of the filter.

Butterworth Filter Design

The corner frequency or cutoff frequency is given by the equation:

$$H_{(j\omega)} = \frac{1}{\sqrt{1+\varepsilon^2\left(\frac{\omega}{\omega_p}\right)^{2n}}}$$

The Butterworth filter has frequency response as flat as mathematically possible, hence it is also called as a maximally flat magnitude filter (from 0Hz to cut-off frequency at -3dB without any ripples). The quality factor for this type is just Q=0.707 and thus, all high frequencies above the cut-off point band rolls down to zero at 20dB per decade or 6dB per octave in the stop band. The Butterworth filter changes from pass band to stop-band by achieving pass band flatness at the expense of wide transition bands and it is considered as the main disadvantage of Butterworth filter. The low pass Butterworth filter standard approximations for various filter orders along with the ideal frequency response which is termed as a “brick wall”. The frequency response of the nth order Butterworth filter is given as

$$H_{(j\omega)} = \frac{1}{\sqrt{1+\varepsilon^2\left(\frac{\omega}{\omega_p}\right)^{2n}}}$$

By using the standard voltage transfer function, we can define the frequency response of Butterworth filter as

$$H(s) = \frac{Vout}{Vin} = \frac{1}{S^2 + S + 1}$$

Application of Butterworth Filter

- The Butterworth filter is typically used in data converter applications as an anti-aliasing filter because of its maximum flat pass band nature.
- The radar target track display can be designed using Butterworth filter.
- The Butterworth filters are frequently used in high quality audio applications

The audio file used for experiment can be found here: Audio file **(click on Audio file)**

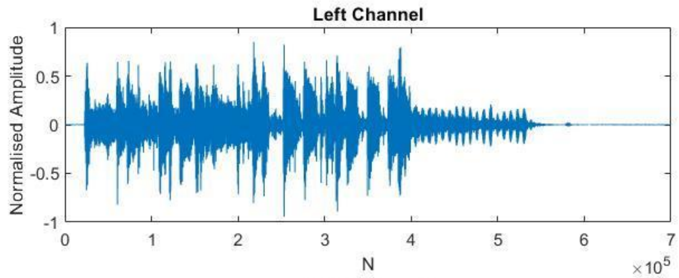
Experiment and Result:

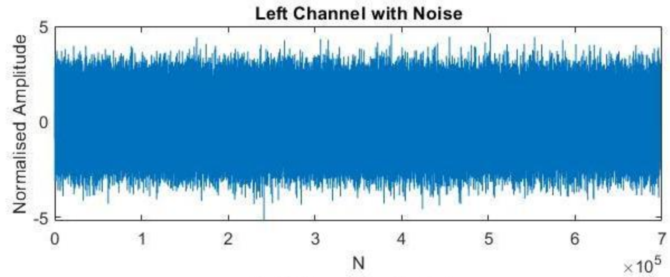
The code snippet that is given below has been used to test and evaluate using a sample dual channel audio file in "wav" format. As shown in the last two figure of the Output section, the proposed algorithm has delivered filtered output that is very close to the original expected output.

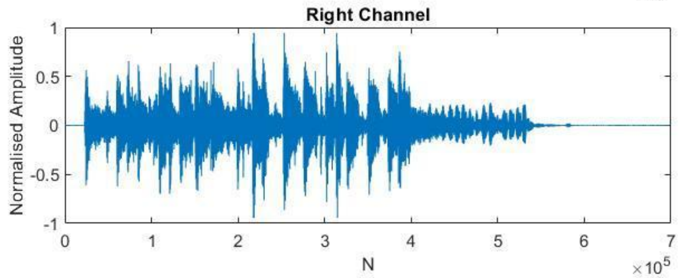
Code Snippet:

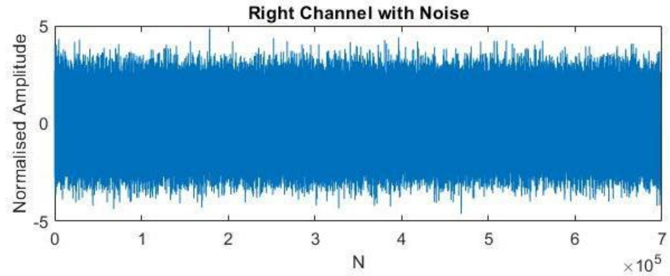
```
clear
close all
clc
[x,fs] =
audioread('triall.wav');
whos x;
N = size(x,1);
figure;
subplot(2,1,1);
plot(1:N, x(:,1));
title('Left Channel');
xlabel('N')
ylabel('Normalised Amplitude')
subplot(2,1,2);
plot(1:N, x(:,2));
title('Right Channel');
xlabel('N');
ylabel('Normalised Amplitude')

%Adding Noise in audio signal
[group-09]
y=x;
y = y + randn(size(y));
figure;
subplot(2,1,1);
plot(1:N, y(:,1));

title('Left Channel with
Noise');
xlabel('N');
ylabel('Normalised Amplitude');
subplot(2,1,2);
plot(1:N, y(:,2));
title('Right Channel with
Noise');
xlabel('N')
ylabel('Normalised Amplitude')
df = fs/N;
w = (-(N/2):(N/2)-1)*df;
a1=fft(y(:,1),N)/N;
a2=fftshift(a1);
y1= fft(y(:,1),N)/N;
y2 = fftshift(y1);
figure;
plot(w,abs(y2));
title('shifted FFT of Noise
mixed signal')
xlabel('Frequency')
ylabel('Absolute value of Y2')
n = 7;
beginFreq = 50 / (fs/2);
endFreq =   150  / (fs/2);
[b,a] = butter(n, [beginFreq,
endFreq], 'stop');
oout = filter(b,a,a2);
fout = filter(b,a,y2);
figure;
plot(b,a);
title('Filter response
Coefficients')
xlabel('b')
ylabel('a')
figure
stem(w,abs(fout))
title('Noise filtered output')
xlabel('Frequency');
ylabel('Absolute value of filter
output')
figure
stem(w,abs(oout))
title('output before adding
Noise')
xlabel('Frequency');
ylabel('Absolute valu

            e of filter output')
```

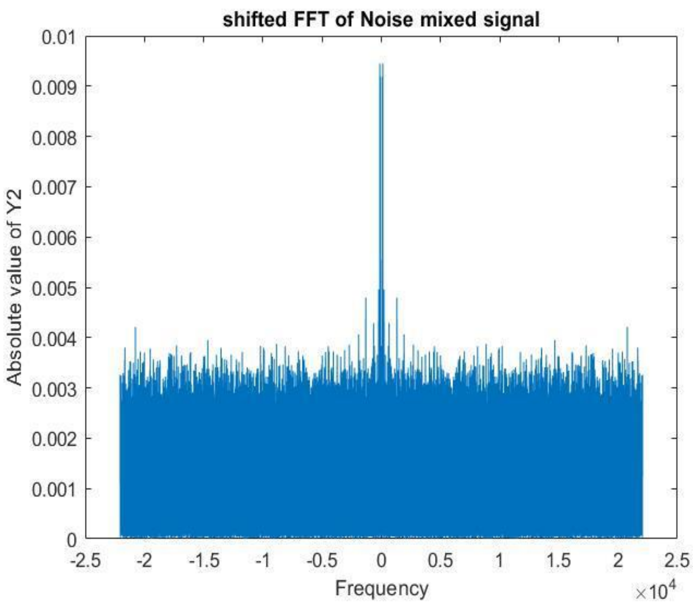
OUTPUT:-

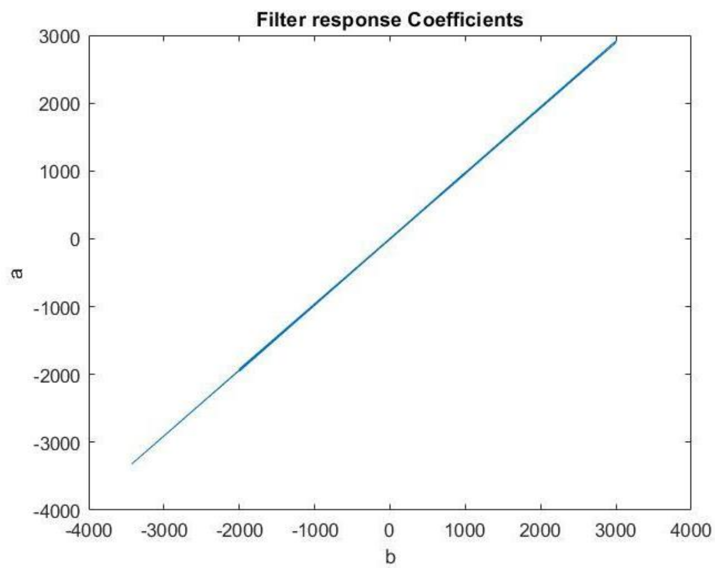


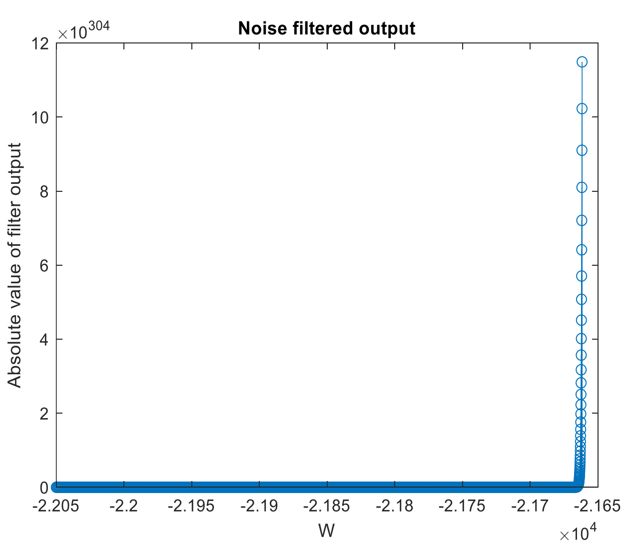




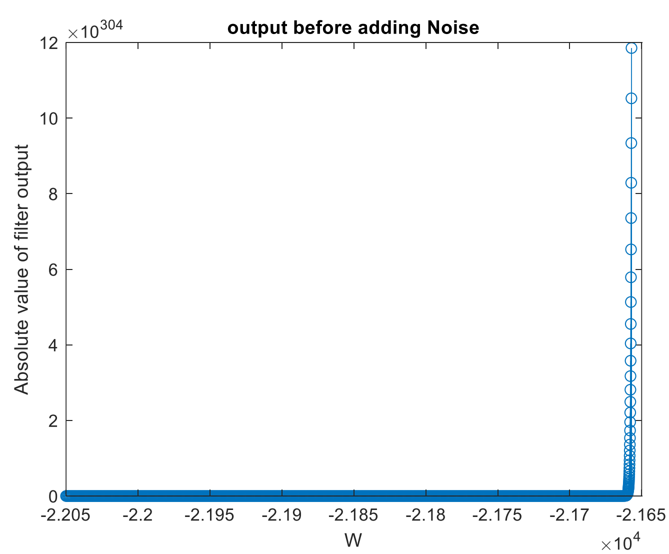








The Above figure is the output of proposed algorithm



The above figure is the expected output

References:

https://en.wikipedia.org/wiki/Butterworth_filter

https://www.youtube.com/watch?v=QC0Pl8sirXU

https://www.researchgate.net/publication/331302592_Removing_Noise_from_audio_signal_using_FFT