Scilab case study project on

Optimizing Audio Quality Using Noise Reduction Techniques

Kshitiz

Indian Institute of Information Technology, Manipur

Audio Processing

August 30, 2024

Abstract

Microphone-recorded voice samples often suffer from environmental noise, leading to distorted and unclear signals. This case study explores digital audio processing techniques to effectively reduce noise in such recordings, specifically focusing on a Navy officer's speech captured in a noisy environment. The goal is to minimize noise without compromising speech clarity, using various filters, including Low-Pass, High-Pass, Band-Pass, and Band-Stop filters. Additional audio enhancement methods are also applied to improve overall sound quality. This study demonstrates the importance of noise reduction in ensuring clear communication in critical situations, such as military operations, and provides a practical approach to enhancing the quality of voice recordings.

1. Introduction

In audio recording, achieving clear and intelligible voice signals is often challenging, particularly when using microphones in noisy environments. Even high-end recording devices like those in smartphones and laptops can capture significant background noise, which can distort the intended message and make it difficult to understand. This problem is especially critical in situations where clear communication is essential, such as military operations or emergency broadcasts.

This case study focuses on addressing the issue of noise in microphone-recorded voice samples by employing digital audio processing techniques. Specifically, the study examines a recorded speech sample of a Navy officer, where noticeable noise interferes with the clarity of the message. The objective is to suppress this noise as much as possible while preserving the integrity of the voice signal.

By utilizing various filtering techniques, including Low-Pass Filters (LPF), High-Pass Filters (HPF), Band-Pass Filters (BPF), and Band-Stop Filters (BSF), the study aims to remove unwanted frequency components and enhance the overall quality of the recording. In addition, methods to improve pitch and apply audio effects are explored to further refine the output.

Through this case study, we demonstrate the effectiveness of digital filtering and other audio enhancement techniques in improving the quality of voice recordings, making them more comprehensible in noisy environments. The findings have practical implications for enhancing communication clarity in critical scenarios, such as military and emergency operations.

2. Problem Statement

A voice sample of a Navy officer’s speech, along with other similar noisy signals, has been recorded using a microphone. The voice sample, which is 39 seconds long, contains noticeable noise introduced by the surrounding environment, making the message unclear. The task is to suppress this noise as much as possible without attenuating the essential information conveyed by the Navy officer. Various noise reduction techniques will be employed, with an aim to improve the pitch where possible, ensuring a clearer and more audible experience.

Additionally, the process will involve displaying time and frequency domain plots at each stage of noise suppression, alongside the original audio plot. This project is particularly designed for emergency situations, such as transmissions on an aircraft carrier where fighter planes are taking off. The recordings, made in various environments with different equipment, may exhibit attenuation from 20% to 40% due to the prevailing circumstances and the quality of the recording devices.

The primary objective is to ensure that the message in the audio sample is clearly identifiable, despite the noisy conditions.

3. Basic concepts related to the topic

I. Noise Removal Using FIR Filters

Noise in the audio sample can be effectively removed by discarding the high-frequency components of the recording.

II. FIR Filters

There are various types of filters used in digital signal processing, including:

1. Low-Pass Filter (LPF): Allows only low-frequency signals to pass through while eliminating high-frequency components. LPFs are particularly useful for controlling the highest range of frequencies in an audio signal.
2. High-Pass Filter (HPF): Opposite to LPF, this filter rejects frequency components below a certain threshold, allowing only high-frequency signals to pass through.
3. Band-Pass Filter (BPF): Allows signals within a specific frequency range to pass through, rejecting frequencies outside this range.
4. Band-Stop Filter (BSF): Rejects signals within a specific frequency range, allowing frequencies outside this range to pass through.

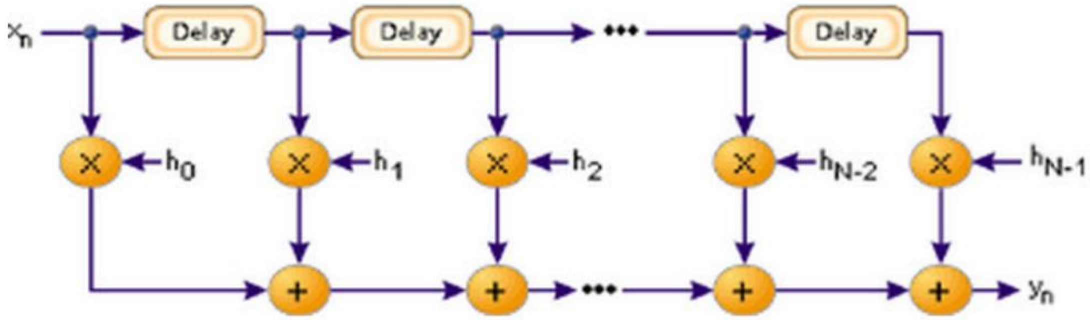
Logical Structure of FIR Filter

A Finite Impulse Response (FIR) filter is widely used to implement various digital frequency responses. Typically, FIR filters are designed with multipliers, adders, and a series of delays to generate the filter's output. The basic structure of an FIR filter includes:

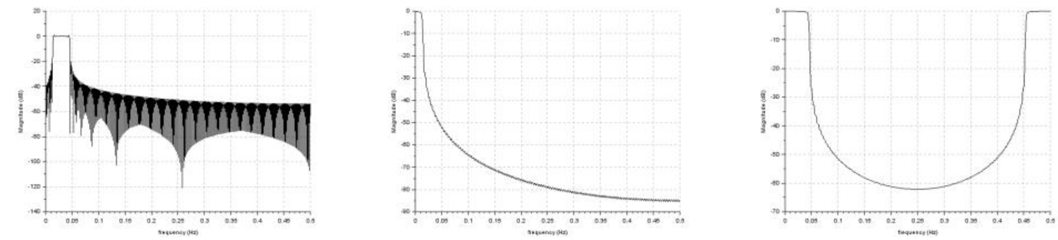
- Input Samples: The original signal to be filtered.
- Delay Elements: Used to store previous input samples.
- Multipliers: Coefficients hkh_khk are used to multiply each delayed sample.
- Adders: The summation of all delayed samples, each multiplied by the corresponding coefficient, produces the output of the filter.

The final output at any given time is the summation of these delayed and weighted samples, providing the desired filtered signal.

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{\sum_{k=0}^{N-1} a_k z^{-k}}$$



FILTERS USED IN THIS TASK:



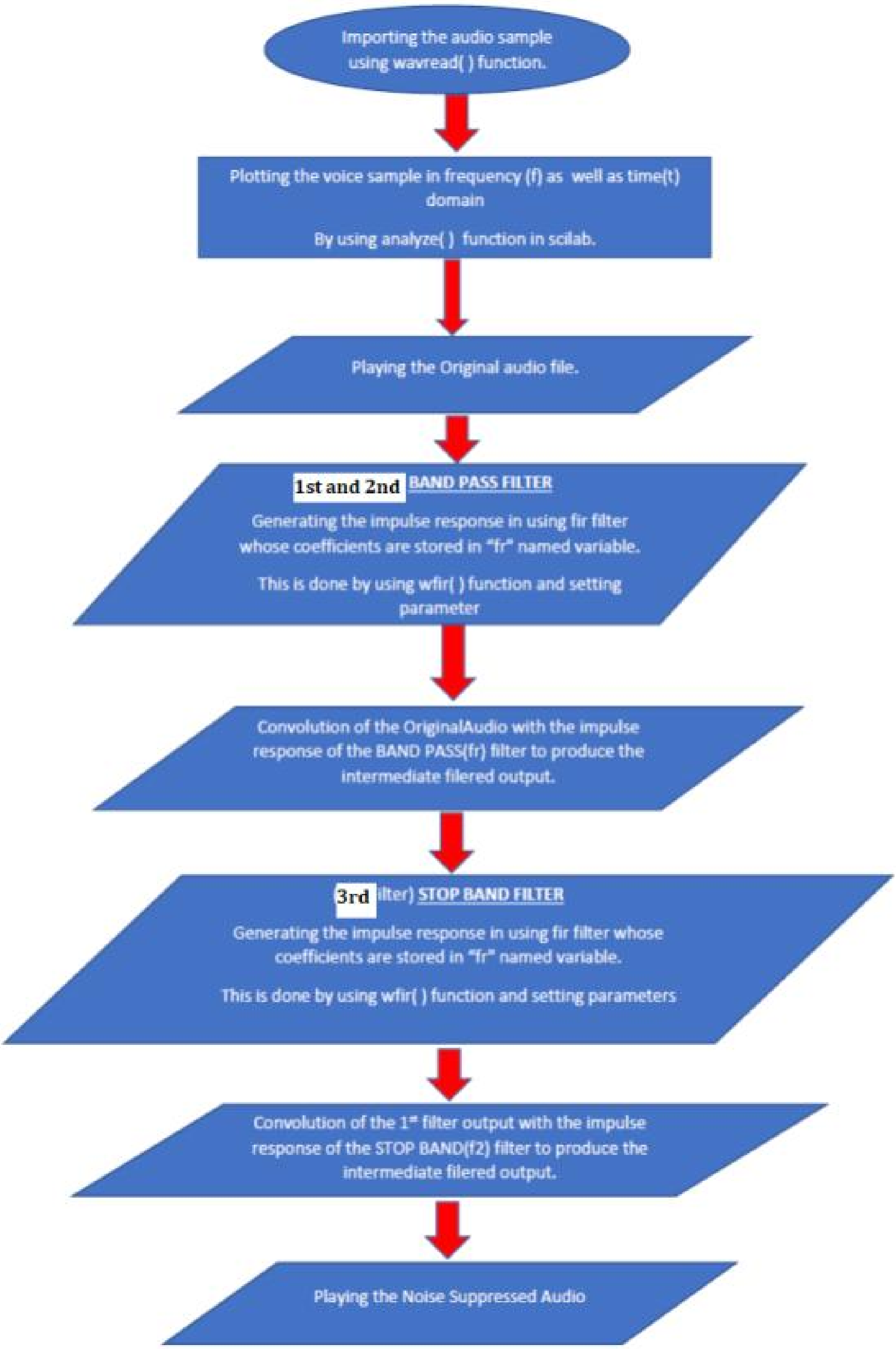
Filter at left is the Band Pass filter (600 Hz to 2000 Hz) normalized to [0-1] ,in center is the Band Pass filter (0Hz to 600Hz) normalized to [0-1] and at bright is the Stop Band filter (2000 Hz to 20000 Hz) with hamming window normalized to [0-1]

III. Convolution

In signal processing, multidimensional discrete convolution is a mathematical operation performed between two functions, fff and ggg, on an nnn-dimensional lattice. This operation produces a third function, also in nnn-dimensions. Multidimensional discrete convolution serves as the discrete analog of multidimensional convolution on Euclidean space. Additionally, it can be viewed as a special case of convolution on groups when the group is composed of nnn-tuples of integers.

The **convol2d** function uses the Fast Fourier Transform (FFT) to compute the full two-dimensional discrete convolution. The dimensions of the result, denoted as C, are given by size(A)+size(B)+1. The indices of the center element of B are defined as floor((size(B)+1)/2).

4. Flowchart


Importing the audio sample using wavread( ) function.
Plotting the voice sample in frequency (f) as well as time(t) domain
By using analyze( ) function in scilab.
Playing the Original audio file.
1st and 2nd BAND PASS FILTER
Generating the impulse response in using fir filter whose coefficients are stored in "fr" named variable.
This is done by using wfir( ) function and setting parameter
Convolution of the OriginalAudio with the impulse response of the BAND PASS(fr) filter to produce the intermediate filered output.
3rd ilter) STOP BAND FILTER
Generating the impulse response in using fir filter whose coefficients are stored in "fr" named variable.
This is done by using wfir( ) function and setting parameters
Convolution of the 1st filter output with the impulse response of the STOP BAND(f2) filter to produce the intermediate filered output.
Playing the Noise Suppressed Audio

5. Software/Hardware used

a. Windows 10 OS

b. Scilab 6.1.1

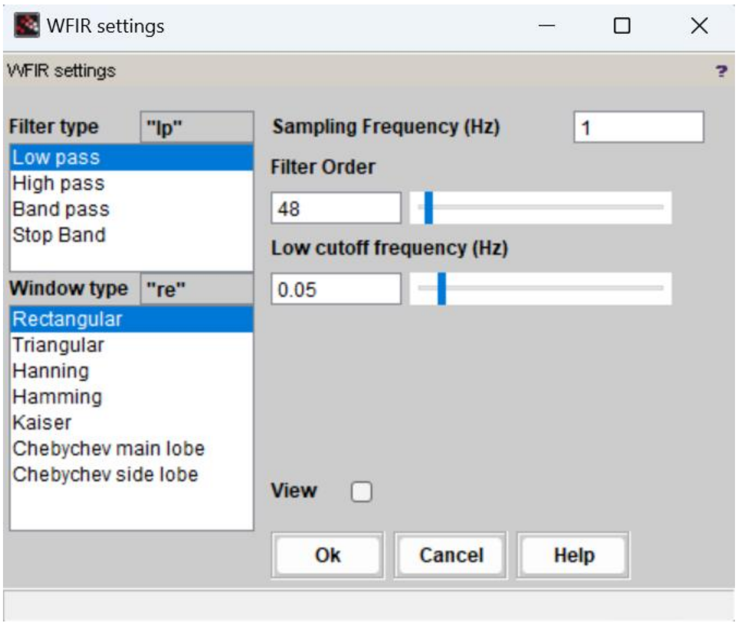
6. Procedure of execution

a. Change the current working directory to the folder containing the code.

b. Execute main.sce file.

c. Remove clc; from Case_study_dependent.sci while executing in Ubuntu

d. The first audio sample will play, and a WFIR settings window will appear. You can choose the filter type, window type, sampling frequency, and other parameters. If no values are specified, the default settings will be applied to filter the signal by clicking on OK.



e. After the first audio plays, frequency and time domain graphs will be displayed, along with comments printed in the console during execution.

f. Once the first audio is successfully filtered, the console will prompt you to press Enter to proceed with filtering the second audio file. This process will continue until all five audio files have been filtered.

NOTE: Three audio samples will be played after execution:

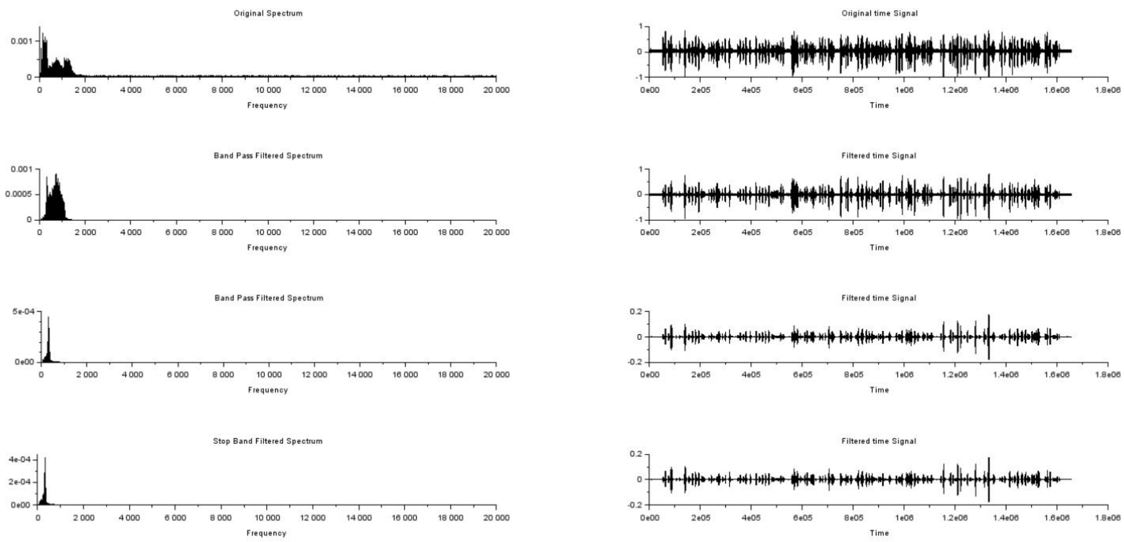
1. **Original Audio** - Full of noise
2. **Filtered Audio** - Slightly noise suppressed
3. **Filtered Audio** - Final output

Content of Folder: Case study1

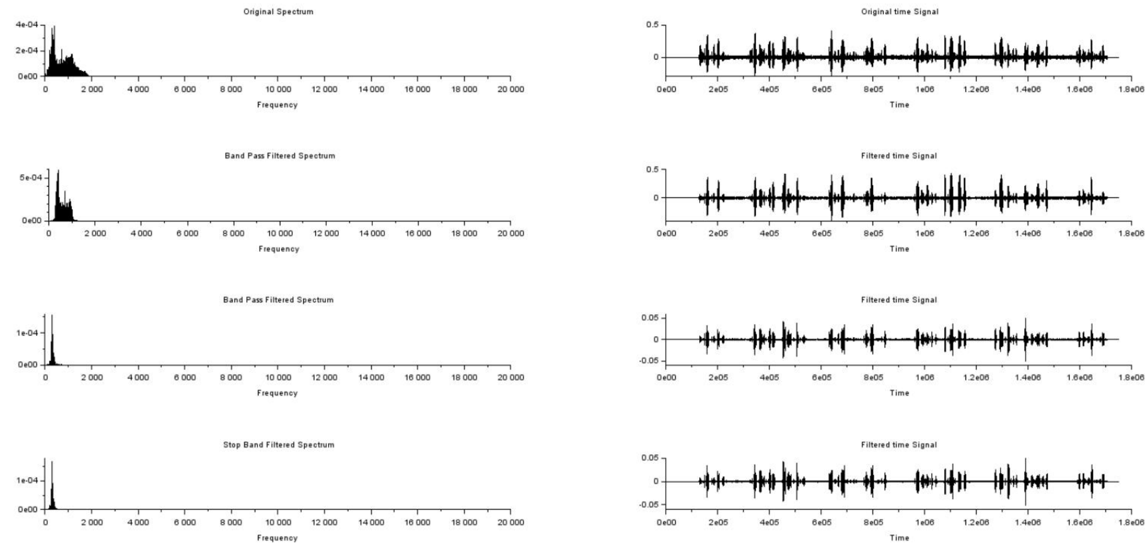
Sr No	File Name	Purpose
1	Case study1.sce	Main executable file
2	Case study dependent.sce	Dependent file
3	Case study dependent2.sci	Dependent file
4	Case study1.wav	Noise-filled audio file to be filtered
5	stock.wav	Noise-filled audio file to be filtered
6	hindi.wav	Noise-filled audio file to be filtered
7	story.wav	Noise-filled audio file to be filtered
8	noise.wav	Noise-filled audio file to be filtered

7. Result

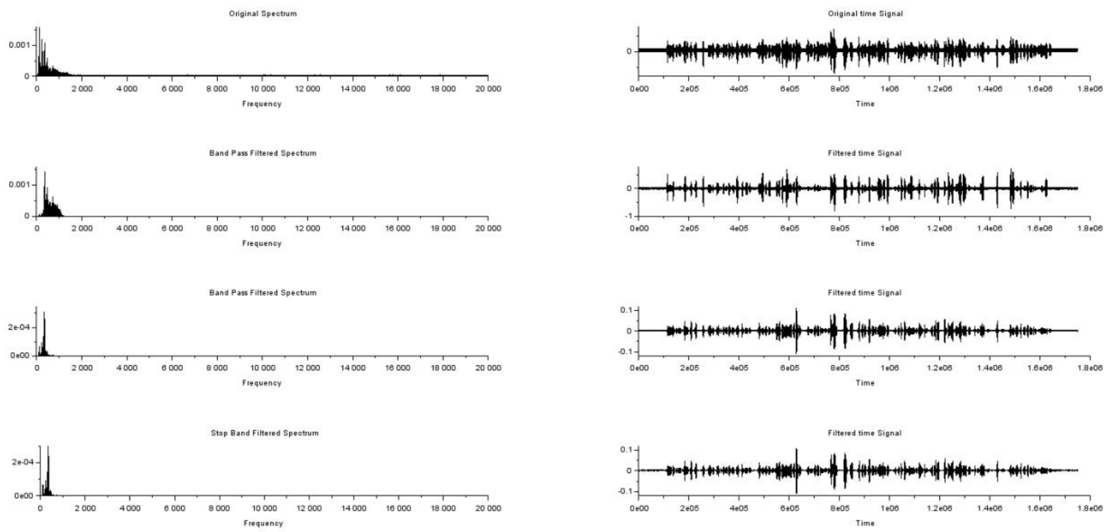
Output 1:Voice Sample1



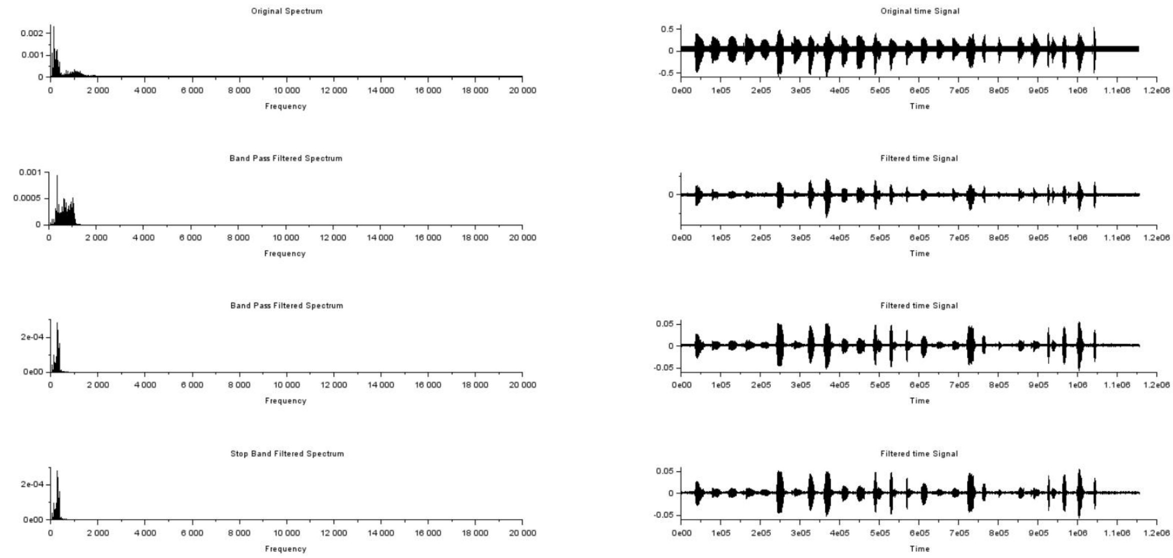
Output 2:Voice Sample2



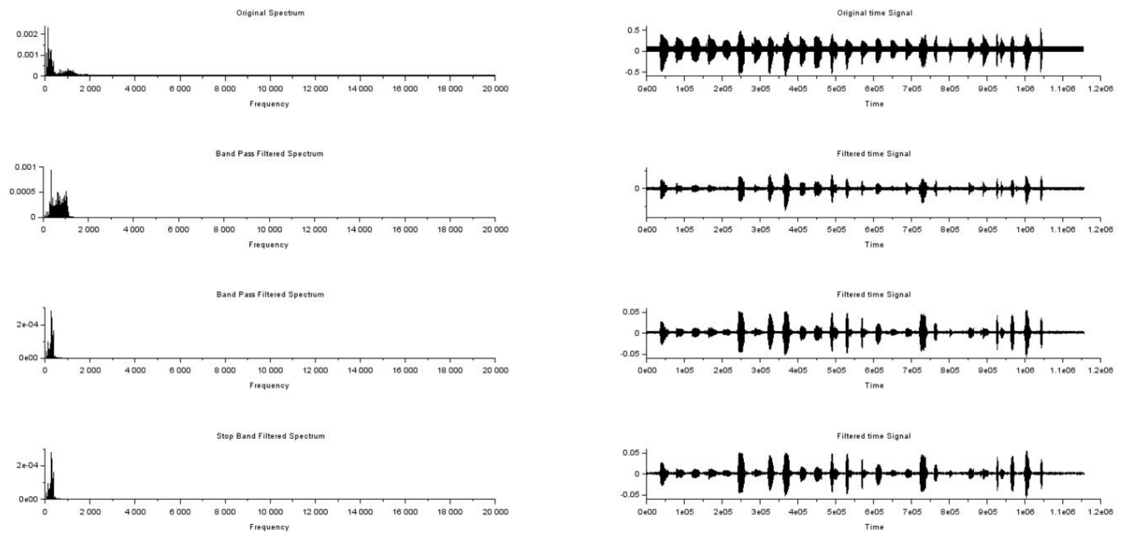
Output 3:Voice Sample3



Output 4:Voice Sample4



Output 5:Voice Sample5



Console Output

```
Scilab 6.1.1 Console

 "First Filter"

 "step 1: fiter type=Band Pass "

 "step 2 :Sampling frequency=44100"

 "step 3 :Window type=Triangular,Low cutoff frequency=600 Hz ,Upper cutoff frequency=2000,Filter

 "Original Track is being played"

 "2nd Filter"

 "step 1: fiter type=Band Pass"

 "step 2 :Sampling frequency=44100"

 "step 3 :Window type=Triangular,Low cutoff frequency=0 Hz,Upper cutoff frequency= 600 Hz,Filter

 "step 4: WAIT FOR SOME TIME FOR TIME AS IT WILL TAKE TIME TO PLAY NOISE SUPPRESED OUTPUT"

 "3rd Filter"

 "step 1: fiter type=Stop Band "

 "step 2 :Sampling frequency=44100"

 "step 3 :Window type=Triangular,Low cutoff frequency=2000 Hz,Upper cutoff frequency=20000 Hz,Fi
Press Enter to stop this audio and play another file
-->
```

8. References

[1] Dhar, P. K., Hee-Sung Jun, & Jong-Myon Kim. (2008). Design and Implementation of Digital Filters for Audio Signal Processing. Proceedings of the 2008 Third International Forum on Strategic Technologies.

[2] Ingle, Vinay K., & Proakis, John G. Digital Signal Processing Using MATLAB® (Third Edition). Northeastern University.