Department of Computer Science & Engineering

Project Report

CSE-416 / Digital Signal Processing Lab

Submitted By	
Group No. 7	
Group Members :	
Student ID	: 221311113
Student ID	: 221311114
Student ID	: 221311115
Student ID	: 221311118
Student ID	: 221311120
Section	: C
Semester	: 8th
Name of the Program	: B.Sc. in CSE
Name of the Department	: Computer Science & Engineering

Submitted to:

Susmita Paul & **Rokaiya Tasnim**

Lecturer

Department Of CSE

Varendra University

Title: Design and implement of a lowpass filter for enhancing the quality of speech signal captured in noisy environment.

Theory: In signal processing, noise reduction is essential to enhance the quality and clarity of audio signals. A common method to remove high-frequency noise is by using **low-pass filters**, which allow low-frequency components of the signal to pass through while attenuating the high-frequency components.

Digital filtering techniques like Infinite Impulse Response (IIR) filters are commonly used for such purposes due to their efficiency and stability. The Fourier Transform (FFT) is also employed to analyze signals in the frequency domain and to identify noise components effectively.

Objective:

i. To reduce the noise from an audio signal using a lowpass filter in MATLAB.
ii. To analyze and compare the original and filtered signals in both time and frequency domains.To apply iii. basic DSP concepts practically using MATLAB.

Tools:

i. MATLAB R2020 or later.
ii. Audio signal: final_sample_noisy_sound.wav.
iii. Signal Processing Toolbox (for designfilt function).

Procedure:

Step 1: Load the Audio Signal

- The noisy sound file is loaded using the audioread function.

- The signal is visualized in both the time domain and frequency domain using FFT.

Step 2: Design the Filter

- A 64th-order **IIR lowpass filter** is designed using designfilt, with a cutoff frequency of 4600 Hz to eliminate high-frequency noise while preserving the main audio components.

Step 3: Apply the Filter to Audio

- The filter is applied using the filter() function to produce the clean (denoised) audio signal.
- Both time and frequency domain representations of the filtered signal are plotted.

Step 4: Save or Play Filtered Audio

- The clean audio is optionally saved or played back using the sound() function to compare with the noisy signal.

Visualization:

1. **Time Domain: Noisy Sound (Blue Plot)**
X-axis: Time (seconds), Y-axis: Amplitude
The signal is very dense and random, meaning there is a lot of noise present.

2. **Frequency Domain: Noisy Sound (Red Plot)**
X-axis: Frequency (Hz), Y-axis: Magnitude
There are many high-frequency components, which mostly represent noise.

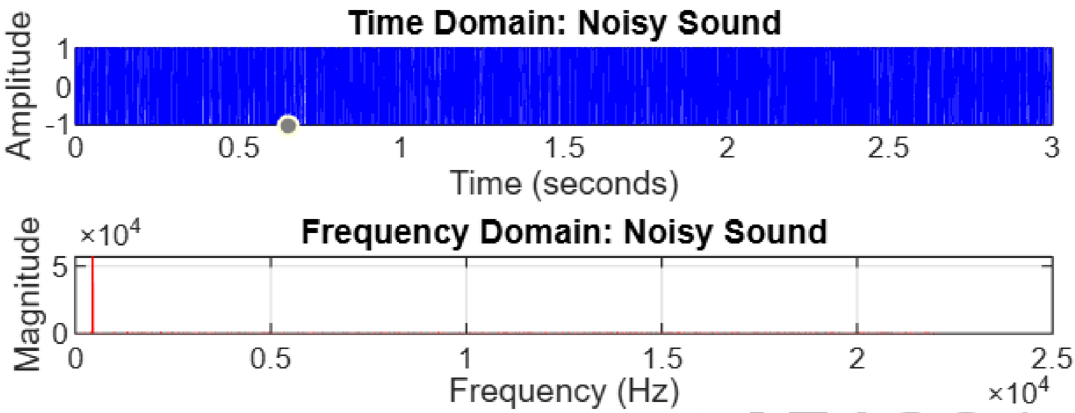

Figure: Noisy Sound

3. **Time Domain: Clean Sound (Green Plot)**
After filtering, the signal has become much smoother.
Most of the noise has been removed.

4. **Frequency Domain: Clean Sound (Magenta Plot)**
The high-frequency components have been cut off.
Now mostly low frequencies remain, meaning the main sound is preserved.

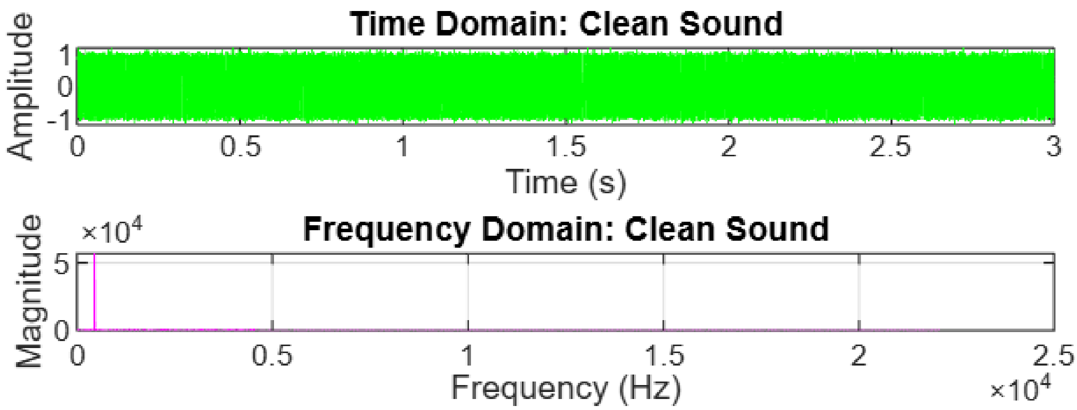

Figure: Clean Sound

Discussion: The original noisy signal contains high-frequency components that are clearly visible in the frequency domain plot. After applying the lowpass filter, these high-frequency components are significantly attenuated, resulting in a much cleaner signal.

In the time domain, the filtered signal appears smoother and free from abrupt noise spikes. The lowpass filter successfully retained the main audio structure (speech or music) while reducing unwanted noise.

This practical demonstration confirms the effectiveness of lowpass filtering for noise reduction, especially in environments with high-frequency disturbances.

Conclusion: This experiment successfully demonstrated how a lowpass filter can reduce high-frequency noise from an audio signal. Using MATLAB, we filtered out unwanted components while preserving the main sound, resulting in a clearer and more refined audio output.