Project Report

Group Members:	Muhammad Attiq (FA23 – BCE – 060) Muhammad Haroon (FA23 – BCE – 122)
Section:	BCE – 4A
Subject:	Signals & Systems
Instructor:	Dr. Bilal Qasim
Course Code:	CPE223

FIR Audio Noise Removal GUI in MATLAB

1. Introduction

In audio processing, noise is an unavoidable component that degrades signal quality. Removing noise without distorting the original signal is a crucial challenge, especially in real-time applications like voice communication, recording, and broadcasting. Finite Impulse Response (FIR) filters are widely used in digital signal processing due to their linear phase characteristics and stability. This project involves developing an interactive MATLAB GUI application for recording, loading, filtering, and saving audio using FIR filters.

The GUI provides options for different FIR filter types (Lowpass, Highpass, Bandpass, Bandstop) and allows users to specify cutoff frequencies and filter lengths. The tool also includes real-time audio visualization in both time and frequency domains.

2. Objectives

- To design a user-friendly graphical interface for FIR-based audio noise removal.
- To implement audio recording, loading, and saving features.
- To allow users to choose filter parameters and types dynamically.
- To visualize the effect of filtering in time and frequency domains.

3. Methodology

3.1 GUI Design

The MATLAB GUI was constructed using ‘uicontrol’, ‘uipanel’, and ‘axes’ components. The main interface is split into a sidebar for user controls and four axes panels to display:

- Original and filtered audio in time domain
- Frequency spectrum of the filtered signal
- FIR filter magnitude response
- FIR filter phase response

3.2 FIR Filtering Algorithm

- The filter is designed using MATLAB's `fir1` function.
- The filter order (length) and window type (Kaiser window) are adjustable.
- Preprocessing includes highpass filtering to remove DC offset and dynamic normalization.
- Depending on user selection, the GUI computes coefficients for lowpass, highpass, bandpass, or bandstop FIR filters.
- The filtered output is normalized and denoised by thresholding low-amplitude signals.

3.3 Phenomenon of Audio Filtering

Audio filtering involves altering or removing certain frequency components of an audio signal. FIR filters perform this operation by convolving the input signal with a set of filter coefficients. These coefficients determine which frequencies are passed and which are attenuated. For example:

- A lowpass filter passes low-frequency components while attenuating high-frequency noise.
- A highpass filter does the opposite, preserving high-frequency content and removing low-frequency hum or rumble.
- Bandpass and bandstop filters target specific ranges of frequencies to either isolate or remove them.

In the GUI, the user selects the filter type and cutoff frequencies. The FIR filter is then applied using the convolution principle.

Here, h[k] are the filter coefficients, x[n] is the input signal, and y[n] is the output (filtered) signal. This process ensures linear phase response, meaning all frequencies are delayed equally, preserving the waveform shape—critical for audio fidelity.

3.4 Audio Functions

- Recording: Uses `audiorecorder` to capture 5 seconds of audio.
- Loading: Audio files can be imported using `uigetfile` and `audioread`.
- Saving: Filtered audio can be exported using `audiowrite`.
- Playback: Users can preview both original and filtered audio via `sound`.

3.5 Visualization

- Time domain plots overlay raw and filtered audio.
- FFT with Gaussian smoothing is used for frequency domain visualization.
- Filter response is shown via `freqz` to provide insights into filtering behavior.

4. Results

The GUI performs efficiently on real-time audio recordings and imported files. It successfully removes background noise while preserving voice signals. The Kaiser window improves stopband attenuation without ripple distortion. Real-time visualization helps users understand the filter's behavior, making it an effective educational and practical tool.

Users can intuitively change cutoff frequencies and instantly view the effect of different filter configurations. The GUI also handles invalid input and gives proper status updates, enhancing usability.

5. Conclusion

The FIR Audio Noise Removal GUI offers a practical demonstration of digital filtering through a clean, interactive MATLAB interface. By combining visualization and parameter customization, it bridges the gap between theoretical FIR design and real-world signal processing. Future improvements could include:

IIR filter support.
Real-time streaming and adaptive filtering.
Automatic noise profiling and parameter tuning.

Lab Assessment		
Lab Task Evaluation	/6	/10
Lab Report	/4	
Instructor Signature and Comments		