Bonus Filter Design Project Report

Course: CSE 3313 - 001 Spring 2025

Student Name: Galib Raid

Submission Date: April 28, 2025

1. Introduction

In this project, a Butterworth lowpass filter was designed and implemented to remove high-frequency noise from a noisy audio sample. The goal was to preserve speech frequencies (approximately 100 Hz to 2 kHz) while attenuating noise between 2.5 kHz and 5.5 kHz. MATLAB was utilized for signal analysis, filter design, audio processing, and evaluation.

2. Methodology

Step 1: Frequency Analysis

- The noisy audio file 'noisyaudio-1.wav' was read into MATLAB.

- DFT of the audio was calculated and plotted.

Step 2: Filter Design

- Passband: 2000 Hz

- Stopband: 2500 Hz

- Passband Ripple: 1 dB

- Stopband Attenuation: 30 dB

- Filter Order: 14

- Normalized Cutoff Frequency: 0.3779

Step 3: Filter Implementation

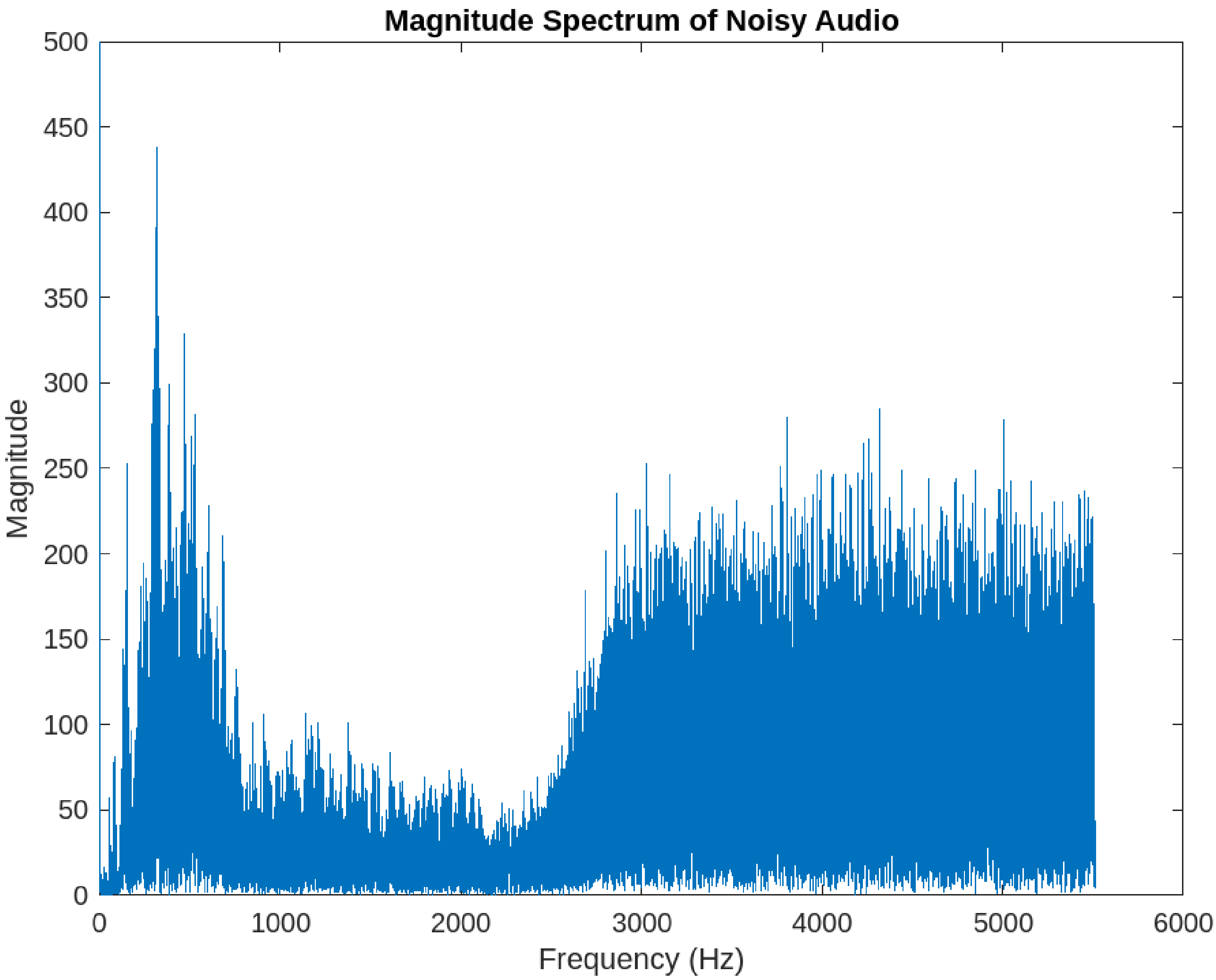
Bonus Filter Design Project Report

- The filter was applied to the noisy audio.

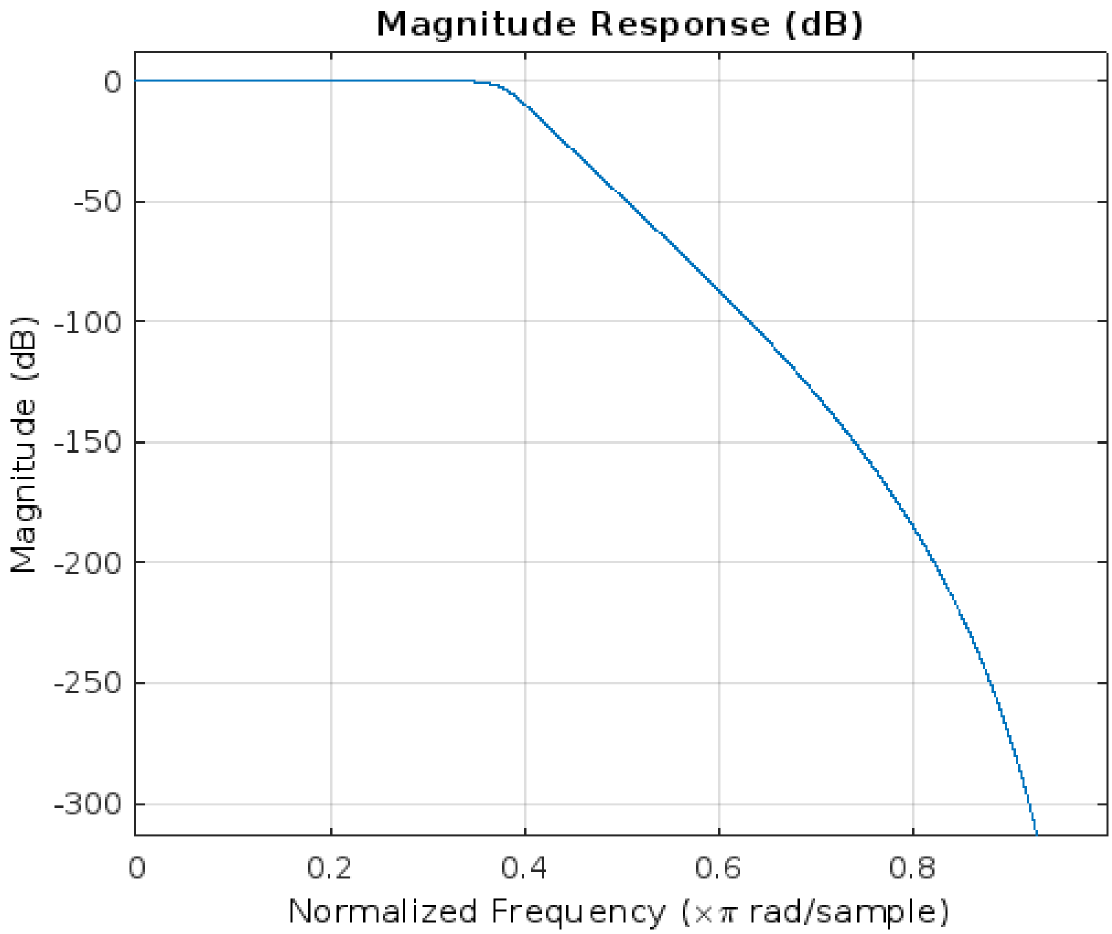
- Filtered output saved as 'filteredaudio.wav'.

- DFT was recalculated and compared.

DFT of Noisy Audio



Magnitude Response of Designed Butterworth Filter

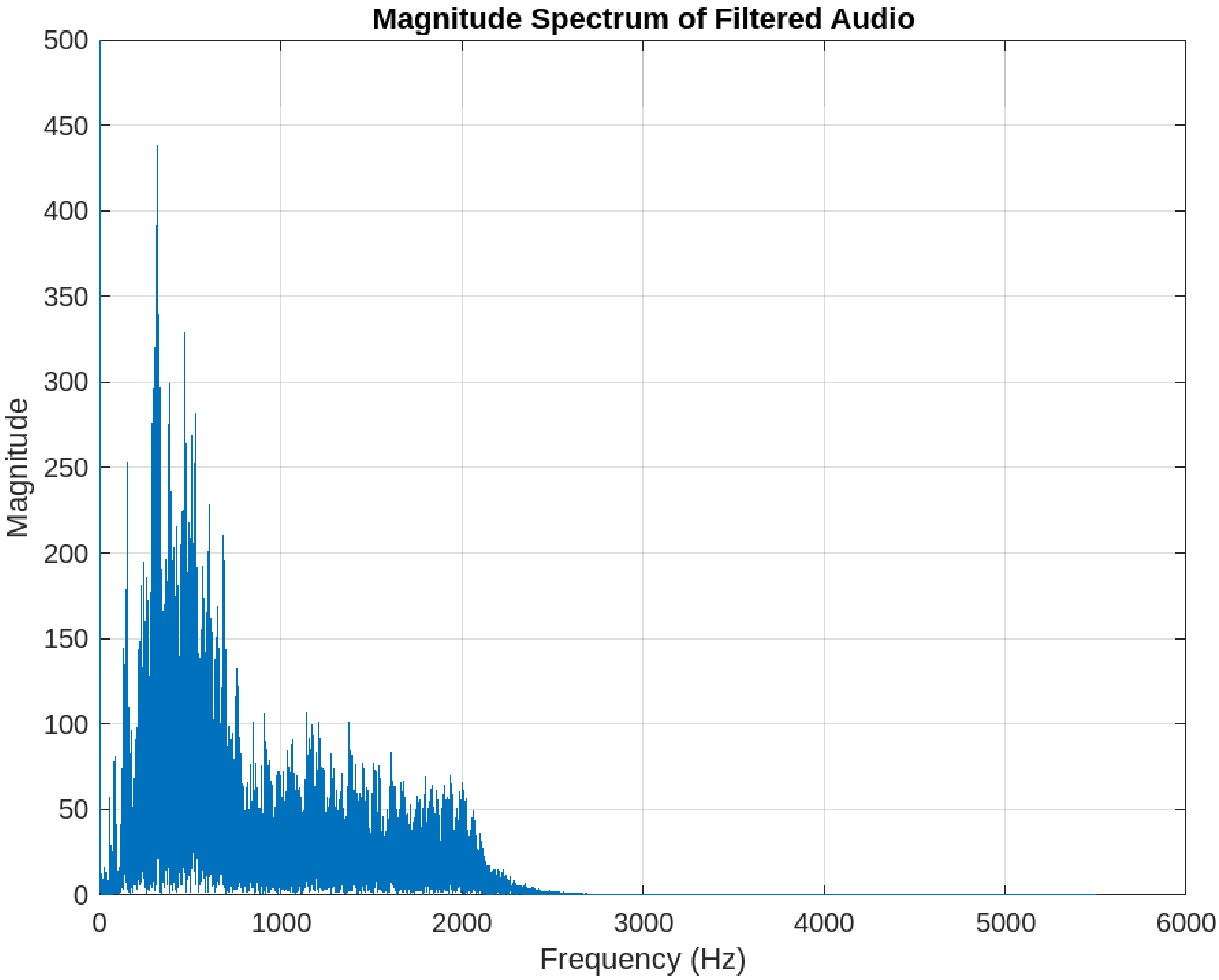


3. Results

The Butterworth lowpass filter significantly attenuated the high-frequency noise while preserving important speech frequencies. Listening tests confirmed a noticeable improvement in audio clarity.

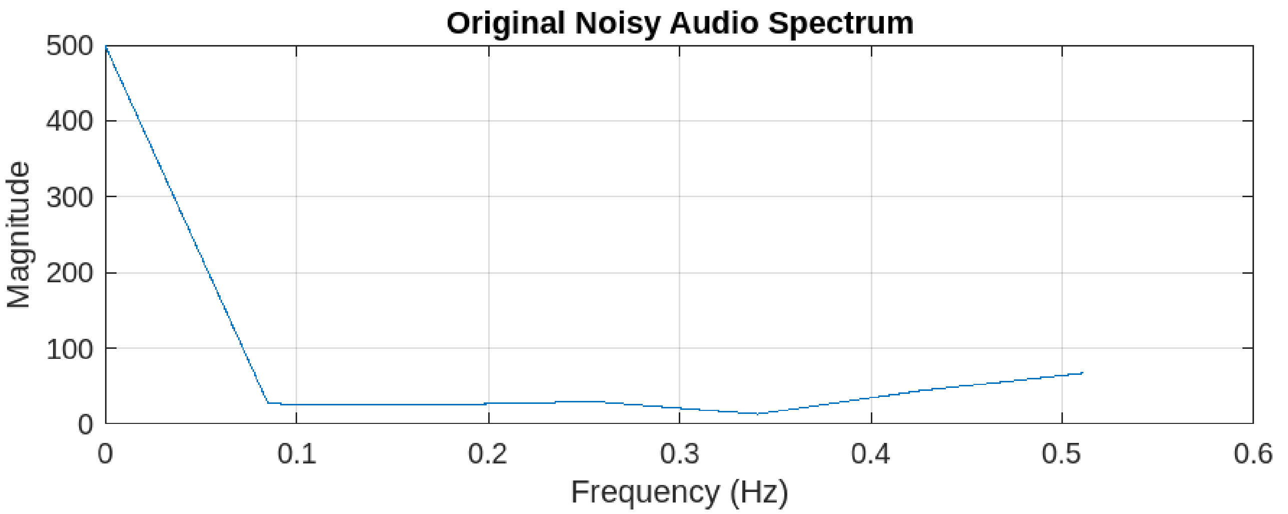
DFT of Filtered Audio

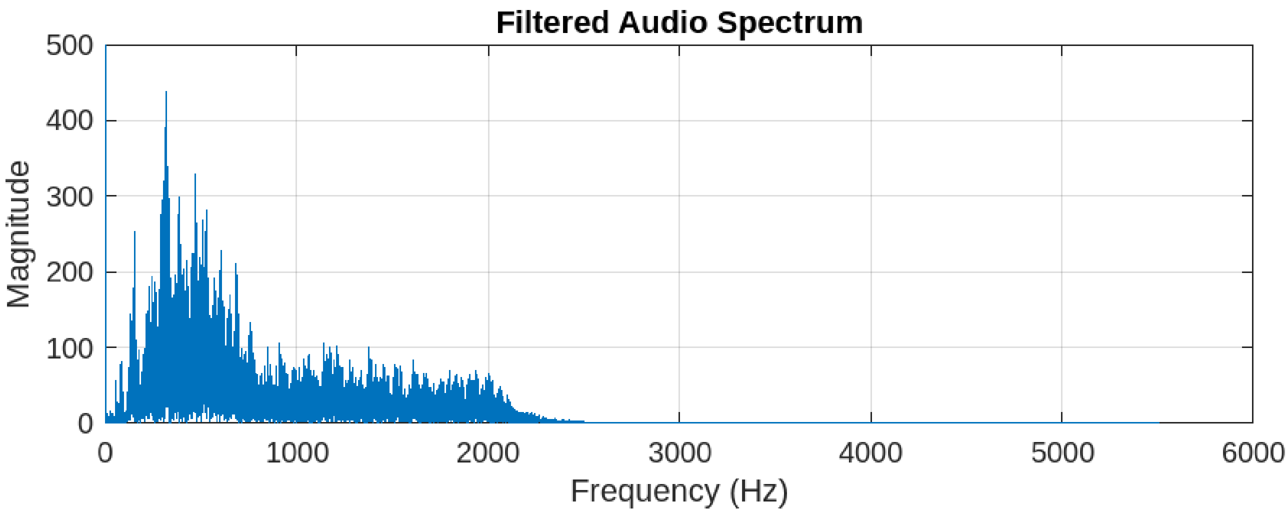
Bonus Filter Design Project Report



Comparison: Original vs Filtered Spectrum

Bonus Filter Design Project Report





4. Conclusion

The project successfully demonstrated the use of Butterworth lowpass filtering to clean noisy audio. MATLAB's capabilities allowed efficient design, application, and evaluation of the filter.

5. Appendix: MATLAB Code

```
y, Fs = audioread('noisyaudio-1.wav');

N = length(y);

Y = abs(fft(y));

f = (0:N-1)*(Fs/N);
```

Bonus Filter Design Project Report

```
wp = 2000; ws = 2500; Ap = 1; As = 30;

wp_norm = wp / (Fs/2);

ws_norm = ws / (Fs/2);

[N, Wn] = buttord(wp_norm, ws_norm, Ap, As);

[b, a] = butter(N, Wn);


y_filtered = filter(b, a, y);

audiowrite('filteredaudio.wav', y_filtered, Fs);

N_filtered = length(y_filtered);

Y_filtered = abs(fft(y_filtered));

f_filtered = (0:N_filtered-1)*(Fs/N_filtered);
```

6. Bonus: Manual Difference Equation Implementation

To verify the Butterworth filter manually, a custom difference equation was implemented using the b and a coefficients. The recursive equation was solved with the initial condition $y[n] = 0$ for $n < 0$. The magnitude spectrum of the manually filtered audio matched the spectrum obtained using MATLAB's filter() function, confirming the correctness of the manual implementation.

Magnitude Spectrum of Manually Filtered Audio

Bonus Filter Design Project Report