**PROJECT**

**Aim:** **Study of time and frequency domain analysis of your recorded voice signal.**

**Theory:**

This project is a multifaceted exploration of recorded voice signals, delving into the intricacies of both their time and frequency domain representations. This project begins with the capture of a voice sample, typically spanning 10 to 15 seconds, followed by its conversion to a digital audio format (.wav). Through the application of MATLAB and digital signal processing techniques, it endeavors to accomplish several key objectives. In the time domain, it offers a visual analysis of phonetic characteristics, allowing researchers and enthusiasts to gain insights into the nuances of their own voice. Moreover, it investigates the impact of low-pass and high-pass filtering on the time-domain representation, shedding light on how these filters affect the signal's temporal dynamics. Another vital aspect of the project in the time domain is the segregation of the audio signal into voice and un-voice components, a process that enhances our understanding of phonetic variation. In the frequency domain, the project presents a visual analysis of the signal's spectral content, enabling the inspection of its frequency details. Subsequently, it examines the effects of low-pass and high-pass filtering in the frequency domain, shedding light on how these filters influence the signal's frequency characteristics. Lastly, just as in the time domain, the project seeks to segregate voice and un-voice segments, offering a unique perspective on the spectral attributes of the recorded voice.

# **Discussion**

The project provides a comprehensive analysis of a recorded voice signal. By examining the signal in both the time and frequency domains and applying filtering and segregation techniques, we gain insights into various aspects of the voice signal. These insights can be used for a wide range of applications, including speech recognition, voice quality assessment, and phonetic analysis.

The visual analysis of phonetics in the time domain offers a glimpse into the speech patterns and articulation of the recorded voice. It allows us to observe the temporal variations and transitions between different phonetic units.

Filtering in both the time and frequency domains demonstrates the effects of low-pass and high-pass filtering on the voice signal. This is valuable for tasks like noise reduction or emphasizing specific frequency components relevant to voice quality.

The segregation of voice and un-voice parts is a crucial step for speech analysis. It helps identify voiced speech intervals, silence, and unvoiced segments, which are essential in phonetic studies and speech processing applications.

In the frequency domain, we gain insights into the spectral characteristics of the voice signal. Understanding the frequency content is critical for distinguishing different phonemes and speech sounds.

# **Conclusions**

**Objective 1: Visual Analysis of Phonetics (Time Domain)**

The visual analysis of the recorded voice signal in the time domain reveals the intricate patterns of speech articulation and phonetics. By examining the waveform, we can identify the dynamic transitions between speech sounds, which are fundamental for speech analysis and understanding the spoken language.

**Objective 2: Effect of LPF Filtering in Time Domain**

Applying a low-pass filter to the voice signal in the time domain has a significant impact on the removal of high-frequency components. This filtering technique is effective in reducing noise and enhancing the clarity of the voice signal, making it valuable for noise reduction and improving voice quality.

**Objective 3: Effect of HPF Filtering in Time Domain**

The application of a high-pass filter in the time domain emphasizes high-frequency details in the voice signal. This technique is useful for accentuating certain speech characteristics and can be employed in scenarios where fine-grained analysis of high-frequency components is necessary.

**Objective 4: Segregate Voice and Un-voice Part (Time Domain)** Segregating voice and un-voice parts in the time domain allows us to distinguish voiced speech from unvoiced intervals and silence. This separation is crucial for phonetic analysis and speech recognition systems, enabling the identification of speech segments and non-speech portions.

**Objective A: Visual Analysis about the Frequency Details (Frequency Domain)**

The frequency domain analysis provides a comprehensive view of the spectral characteristics of the recorded voice signal. It offers insights into the distribution of frequencies, allowing us to identify key features and distinguish different phonemes based on their frequency patterns.

**Objective B: Effect of LPF Filtering in Frequency Domain**

Applying a low-pass filter in the frequency domain attenuates high-frequency components, resulting in a smoother frequency spectrum. This filtering technique is valuable for reducing spectral noise and emphasizing the fundamental frequency components of the voice signal.

**Objective C: Effect of HPF Filtering in Frequency Domain**

The high-pass filter in the frequency domain accentuates high-frequency components in the spectrum. It is instrumental in highlighting fine details and high-frequency harmonics, making it suitable for tasks that require a focus on high-frequency information.

**Objective D: Segregate Voice and Un-voice Part (Frequency Domain)** The frequency-based segregation of voice and un-voice parts enables us to isolate voiced speech segments from unvoiced and silent regions. This segregation enhances our ability to analyze and identify the spectral characteristics associated with voiced speech, contributing to phonetic and speech processing applications.

# **MATLAB Code**

clc; close all; clear;

% Objective 1: Visual Analysis of Phonetics (Time Domain)

[y, Fs] = audioread('Recording.wav');

t = (0:(length(y) - 1)) / Fs;

figure('Name', 'Time Domain Analysis: Visual Analysis of Phonetics');

plot(t, y);

xlabel('Time (s)');

ylabel('Amplitude');

title('Time Domain Representation of the Recorded Voice Signal');

% Objective 2: Effect of LPF Filtering in Time Domain

LPF\_cutoff = 300; % Low-pass filter cutoff frequency in Hz

[b\_lpf, a\_lpf] = butter(2, LPF\_cutoff / (Fs / 2), 'low');

y\_lpf = filter(b\_lpf, a\_lpf, y);

% player = audioplayer(y\_lpf, Fs);

% play(player);

figure('Name', 'Time Domain Analysis: Effect of Low-Pass Filtering');

plot(t, y\_lpf);

xlabel('Time (s)');

ylabel('Amplitude');

title('Time Domain Representation after Low-Pass Filtering');

% Objective 3: Effect of HPF Filtering in Time Domain

HPF\_cutoff = 7000; % High-pass filter cutoff frequency in Hz

[b\_hpf, a\_hpf] = butter(2, HPF\_cutoff / (Fs / 2), 'high');

y\_hpf = filter(b\_hpf, a\_hpf, y);

% player = audioplayer(y\_hpf, Fs);

% play(player);

figure('Name', 'Time Domain Analysis: Effect of High-Pass Filtering');

plot(t, y\_hpf);

xlabel('Time (s)');

ylabel('Amplitude');

title('Time Domain Representation after High-Pass Filtering');

% Objective 4: Segregate Voice and Un-voice Parts (Time Domain)

voice\_threshold = 0.0004;

voice\_signal = y\_lpf .\* (abs(y\_lpf) > voice\_threshold);

unvoice\_signal = y\_lpf .\* (abs(y\_lpf) <= voice\_threshold);

% player = audioplayer(voice\_signal, Fs);

% play(player);

figure('Name', 'Time Domain Analysis: Segregation of Voice and Un-Voice Parts');

subplot(2, 1, 1);

plot(t, voice\_signal);

xlabel('Time (s)');

ylabel('Amplitude');

title('Voice Part (Low-Pass Filtered)');

subplot(2, 1, 2);

plot(t, unvoice\_signal);

xlabel('Time (s)');

ylabel('Amplitude');

title('Un-Voice Part (Low-Pass Filtered)');

% Objective A: Visual Analysis about the Frequency Details (Frequency Domain)

N = length(y);

w = linspace(-pi, pi, N);

Yw = fftshift(fft(y, N));

figure('Name', 'Frequency Domain Analysis: Visual Analysis about Frequency Details');

plot(w / pi, abs(Yw));

xlabel('Normalized Frequency');

ylabel('Magnitude');

title('Frequency Domain Representation of the Recorded Voice Signal');

% Objective B: Effect of LPF Filtering in Frequency Domain

Yw\_lpf = fftshift(fft(y\_lpf, N));

figure('Name', 'Frequency Domain Analysis: Effect of Low-Pass Filtering');

plot(w / pi, abs(Yw\_lpf));

xlabel('Normalized Frequency');

ylabel('Magnitude');

title('Frequency Domain Representation after Low-Pass Filtering');

% Objective C: Effect of HPF Filtering in Frequency Domain

Yw\_hpf = fftshift(fft(y\_hpf, N));

figure('Name', 'Frequency Domain Analysis: Effect of High-Pass Filtering');

plot(w / pi, abs(Yw\_hpf));

xlabel('Normalized Frequency');

ylabel('Magnitude');

title('Frequency Domain Representation after High-Pass Filtering');

% Objective D: Segregate Voice and Un-voice Parts (Frequency Domain)

frequency\_threshold = 0.0004;

voice\_mask = abs(Yw) > frequency\_threshold \* max(abs(Yw)); % same method as time domain

unvoice\_mask = ~voice\_mask;

Yw\_voice = Yw\_lpf .\* voice\_mask;

Yw\_unvoice = Yw\_lpf .\* unvoice\_mask;

voice\_signal\_freq = ifft(ifftshift(Yw\_voice));

unvoice\_signal\_freq = ifft(ifftshift(Yw\_unvoice));

% player = audioplayer(voice\_signal\_freq, Fs);

% play(player);

figure('Name', 'Frequency Domain Analysis: Segregation of Voice and Un-Voice Parts');

subplot(2, 1, 1);

plot(w/pi, Yw\_voice);

xlabel('Time (s)');

ylabel('Amplitude');

title('Voice Part (Frequency Domain Segregation)');

subplot(2, 1, 2);

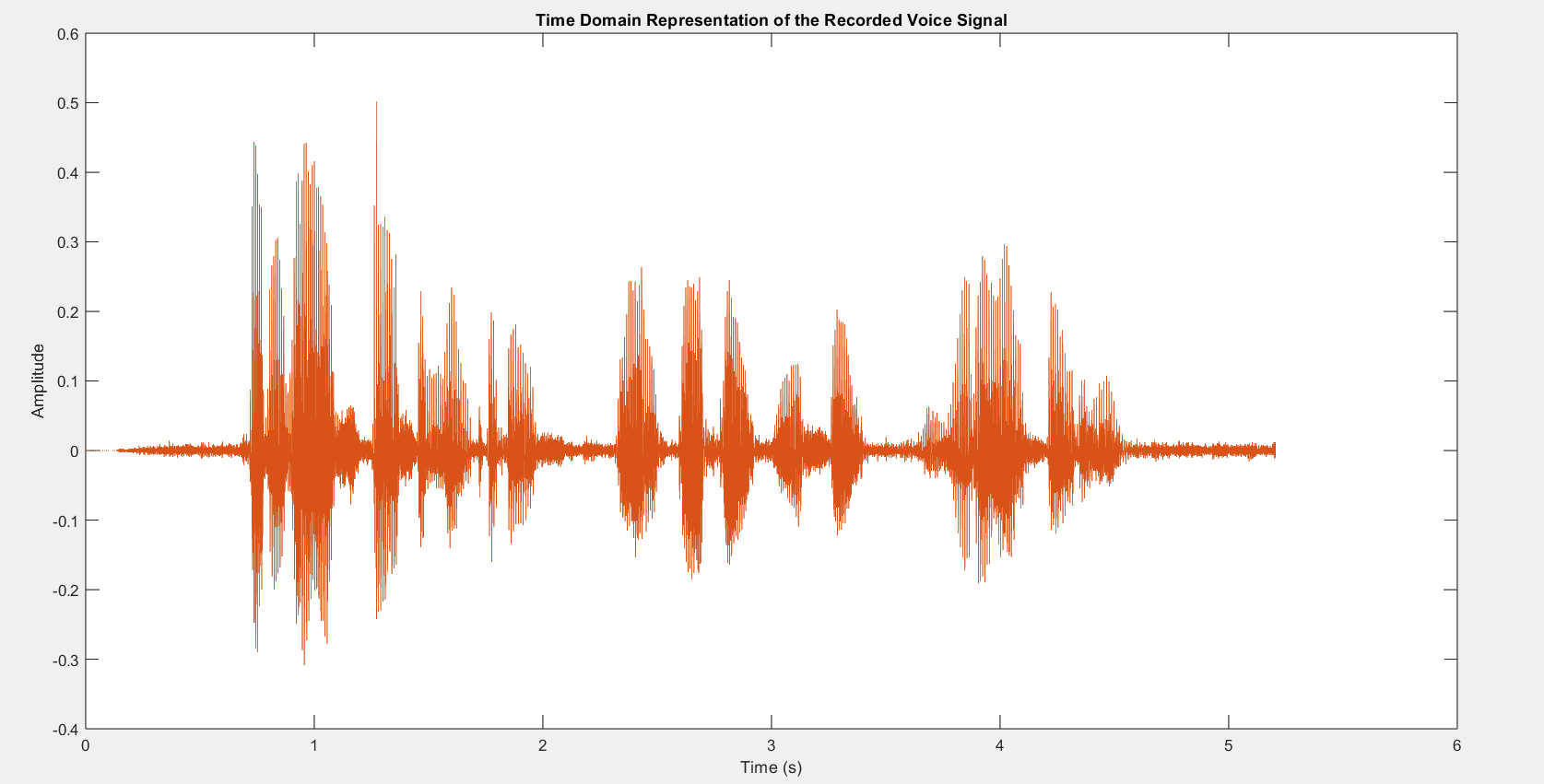
plot(w/pi, Yw\_unvoice);

xlabel('Time (s)');

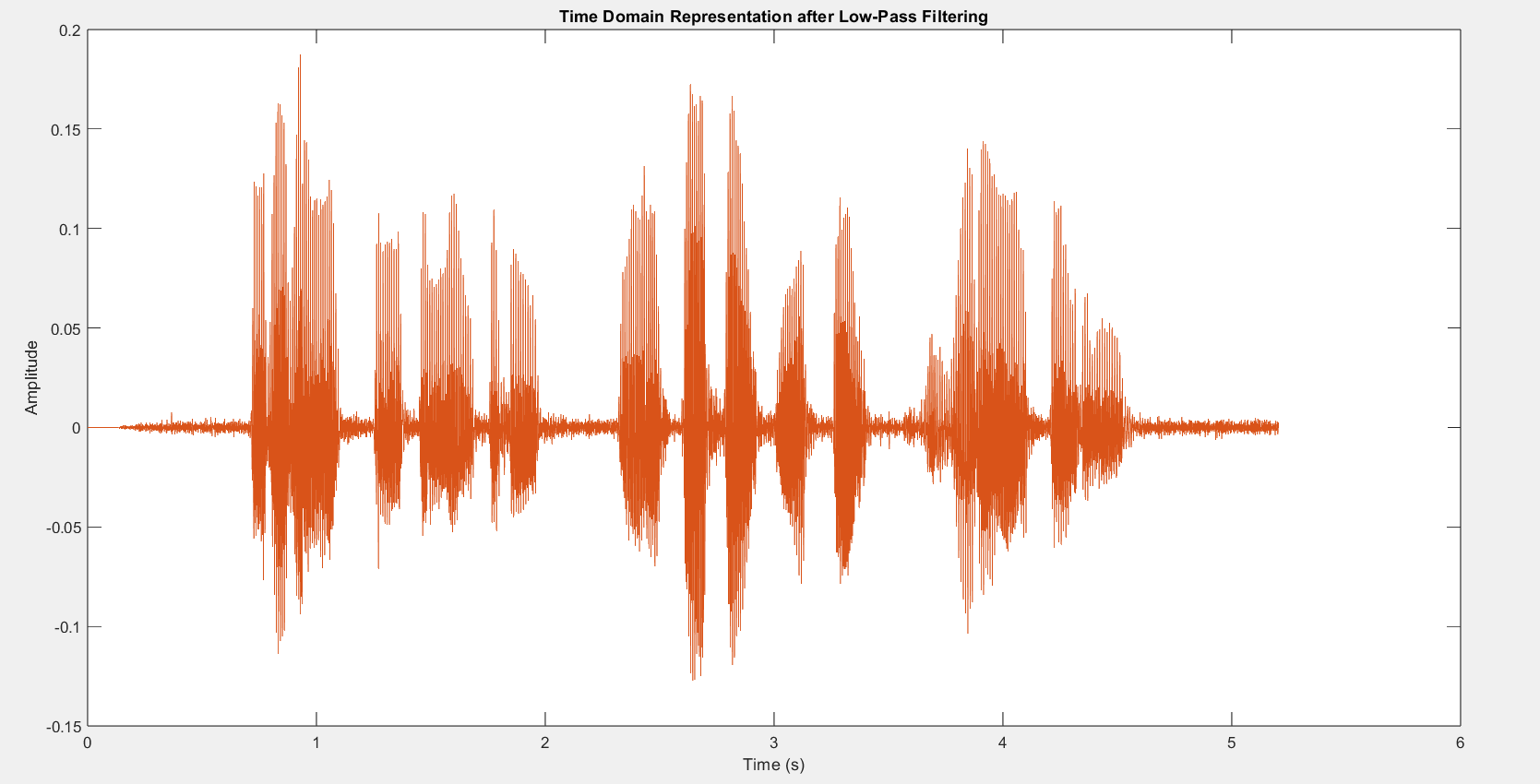
ylabel('Amplitude');

title('Un-Voice Part (Frequency Domain Segregation)');

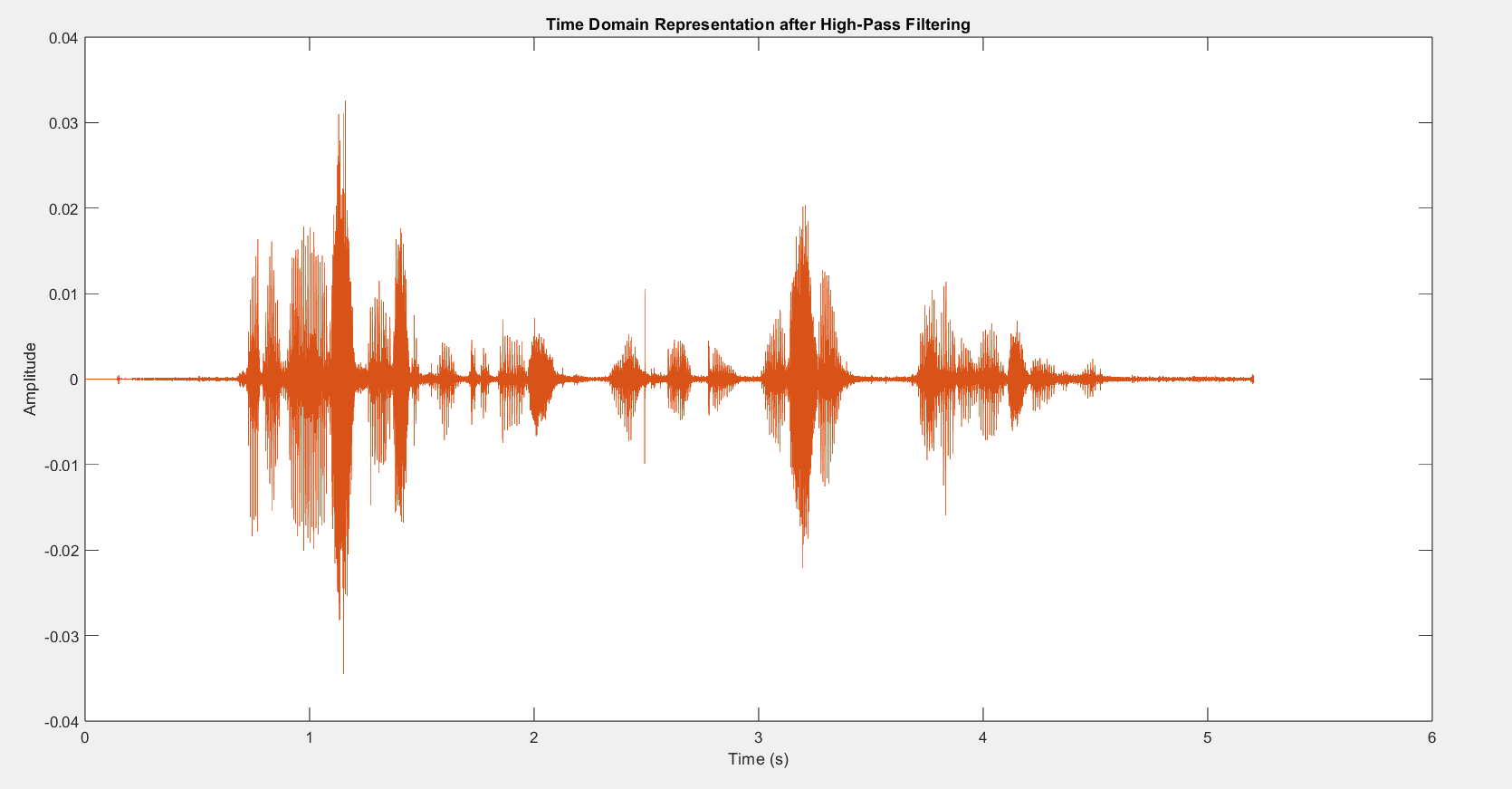
# **MATLAB Results**



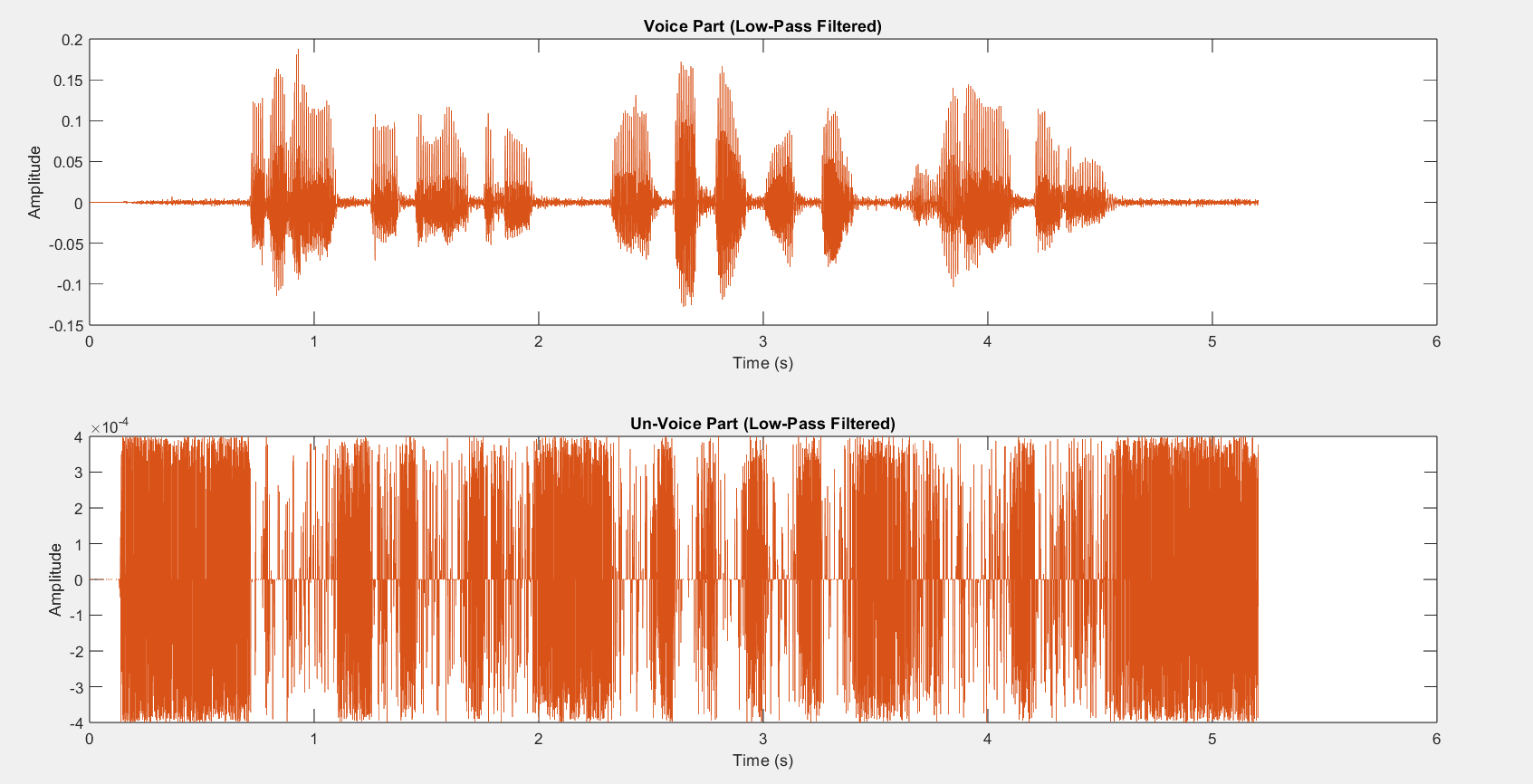
## ***Objective 1***



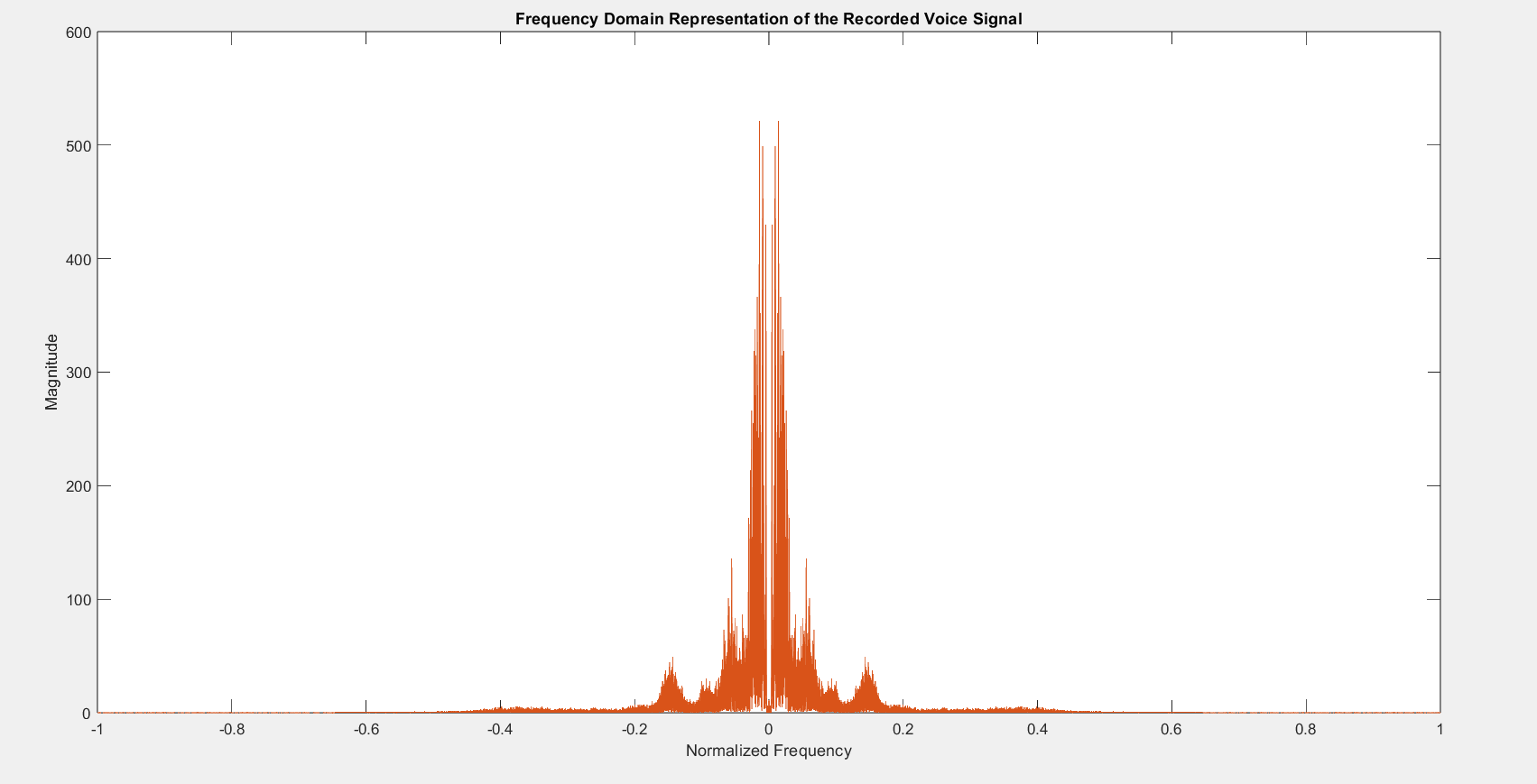
## ***Objective 2***



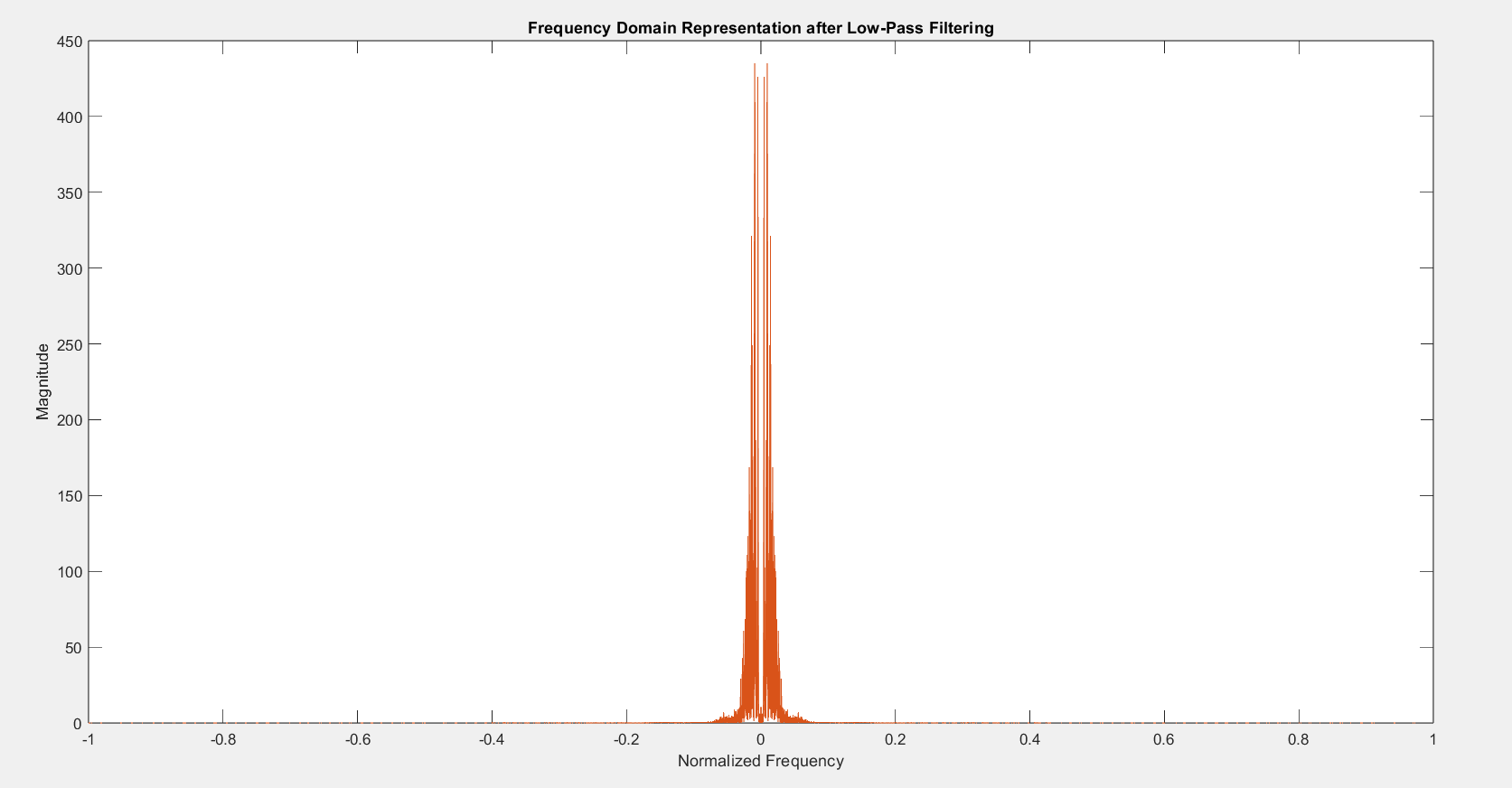
## ***Objective 3***



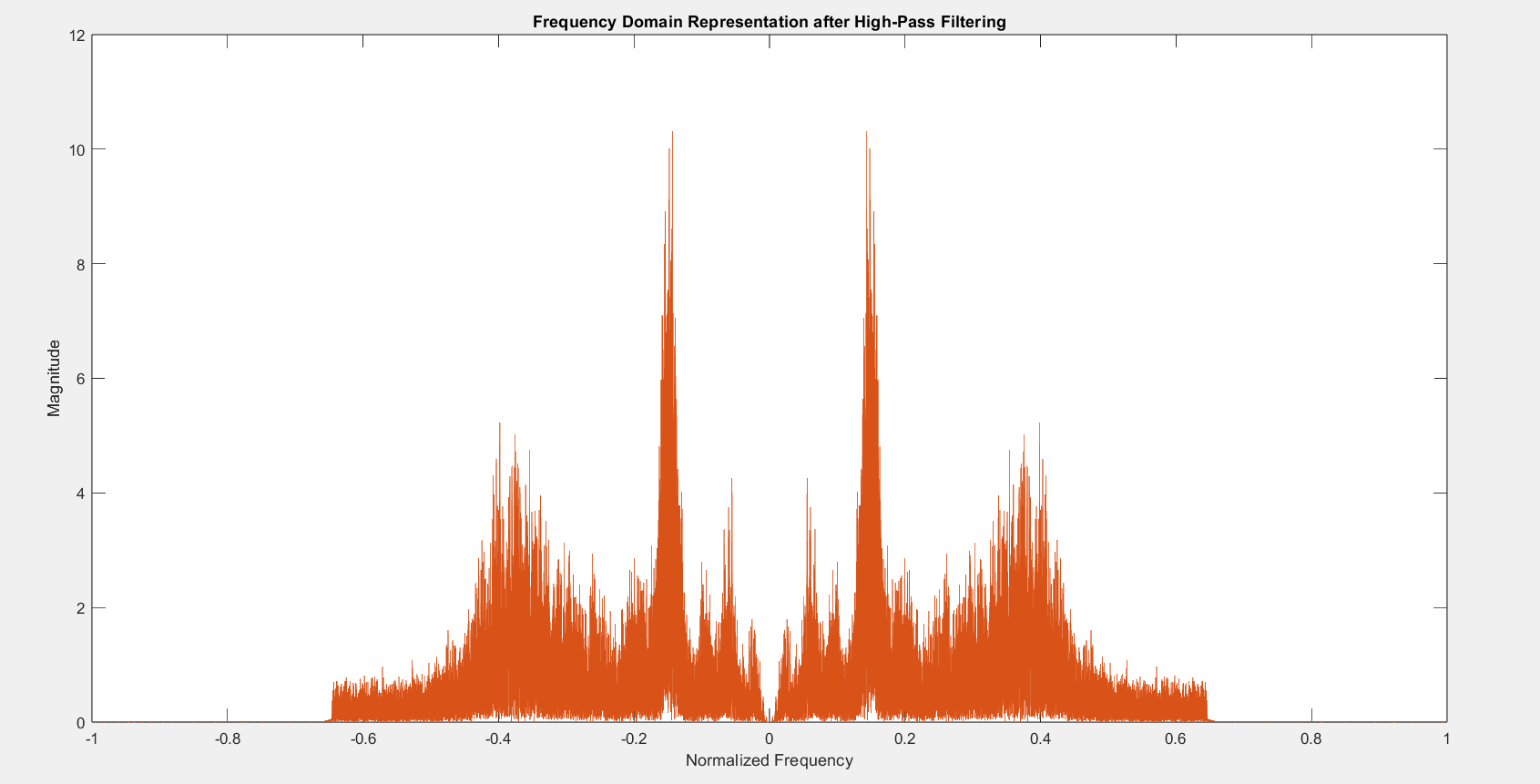
## ***Objective 4***



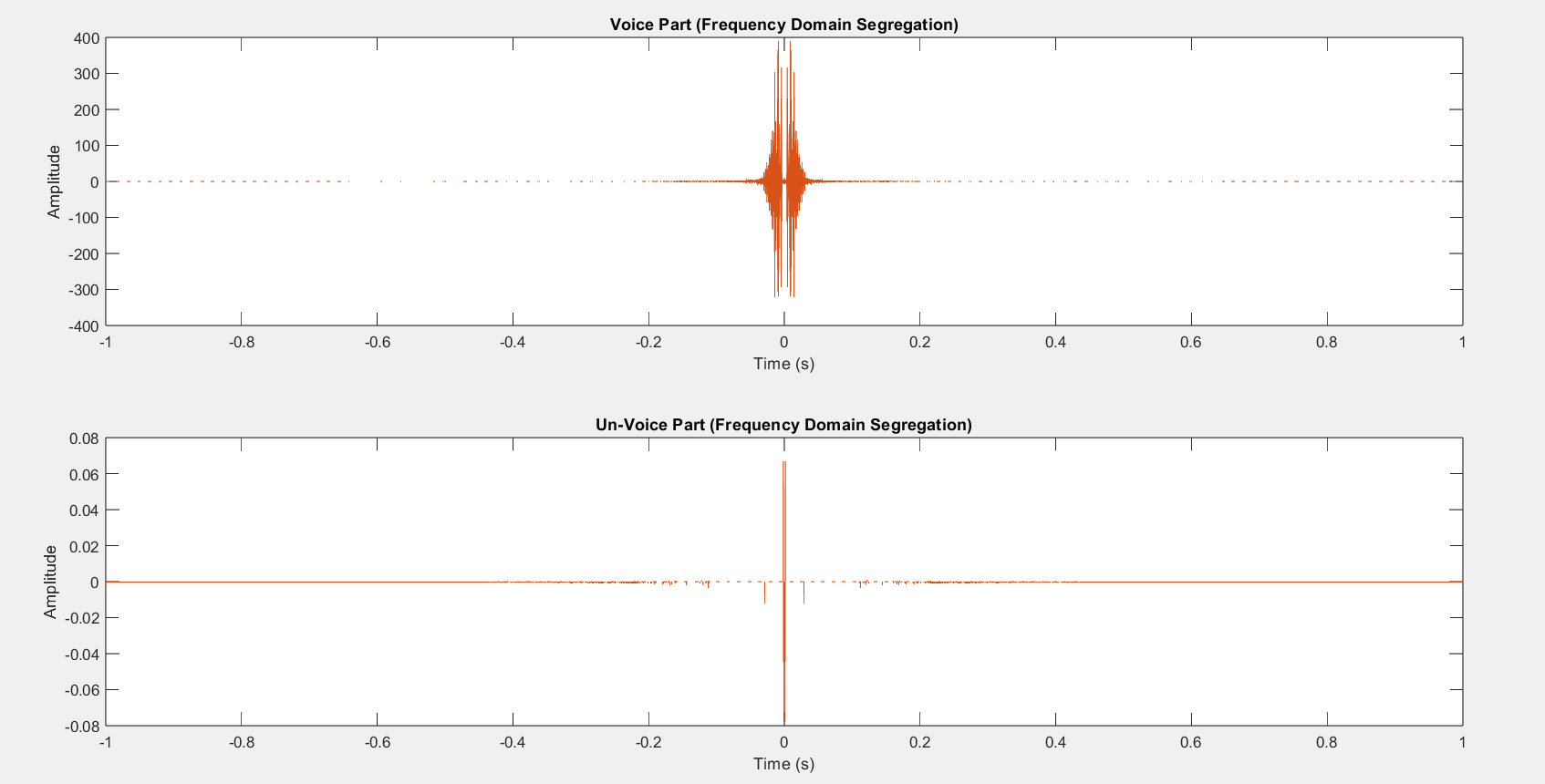
## ***Objective A***



## ***Objective B***



## ***Objective C***



## ***Objective D***