Smart Speech Analyzer GUI

Digital signal processing

COMM 520

Supervision by: Dr. Ahmed Zakaria

TAs: Eng. Gehad Mohy

Author by: Eng. Ali Hassan Ali Hassan Omara

Code: 210071

Author by: Eng. Abdullah Mohamed Mostafa

Code: 210062

M.H.I

2. Table of Contents

3. Abstract

This report presents the development of a MATLAB-based software tool named Smart Speech Analyzer, designed for educational and research purposes in the field of Digital Signal Processing (DSP). The software provides an interactive Graphical User Interface (GUI) that allows users to record or upload speech signals, apply digital audio filtering, and perform advanced analysis using Fast Fourier Transform (FFT) and time–frequency analysis techniques.

The main functionalities include:

- **Visualizing both time-domain and frequency-domain characteristics of speech.**
- **Computing RMS analysis to measure signal energy variations.**
- **Detecting silence and active speech durations.**
- **Estimating speech rate, dominant frequency, and mean frequency.**
- **Displaying spectrograms, RMS plots, and frequency spectra.**

To enhance audio clarity, a bandpass Butterworth filter (300–3400 Hz) is applied to isolate the human voice frequency range. The integration of FFT, RMS, and spectrogram analysis enables a comprehensive statistical summary of the speech signal.

This project demonstrates the application of theoretical DSP concepts to real-time signal processing, providing a practical and interactive learning platform with potential applications in speech enhancement, medical voice analysis, and communication systems.

4. Introduction

Digital Signal Processing (DSP) is a core area in modern engineering, widely used in communications, audio processing, biomedical engineering, and multimedia applications. One of the most prominent DSP domains is speech signal processing, which focuses on extracting meaningful information from human voice signals.

MATLAB is an ideal platform for DSP algorithm implementation due to its extensive built-in functions, GUI development capabilities, and suitability for real-time data handling. In this project, we developed a MATLAB-based GUI tool, Smart Speech Analyzer, enabling users to record, upload, filter, and analyze speech interactively.

The core analysis is driven by the Fast Fourier Transform (FFT), which provides frequency-domain insights, such as detecting dominant and average frequencies. Additional techniques include RMS analysis for energy tracking and time–frequency representations (spectrograms) for visualizing how frequencies evolve over time. The system also estimates silence duration, active speech duration, and speech rate (estimated word count).

By integrating multiple DSP techniques in one interface, this project bridges theoretical learning with practical implementation, making it an effective tool for both education and research in speech signal analysis.

5. Project Purpose

The primary purpose of this project is to analyze and process speech signals using DSP techniques. The system is capable of:

- **Recording or uploading audio files.**
- **Visualizing the signal in both time and frequency domains.**
- **Applying a Bandpass Filter (300–3400 Hz) to isolate the frequency range most relevant to human speech.**
- **Extracting key speech features such as dominant frequency, speech rate, and durations of silence versus active speech.**
- **Presenting multiple visual representations including FFT Spectrum, Spectrogram, and RMS Plot.**

6. Project Objectives

The primary objectives of this project are:

- **Develop an interactive MATLAB GUI capable of real-time recording and uploading of speech signals.**
- **Apply a bandpass Butterworth filter (300–3400 Hz) to enhance speech clarity by isolating the human voice frequency range.**
- **Perform FFT analysis to extract frequency-domain features such as dominant and mean frequencies.**
- **Conduct RMS analysis to evaluate energy variations in the signal.**
- **Perform time-domain analysis to calculate total duration, silence duration, and active speech duration.**
- **Estimate speech rate and the approximate number of spoken words.**
- **Visualize signal characteristics using time plots, frequency spectra, spectrograms, and RMS plots.**
- **Provide an educational DSP tool that integrates multiple analysis techniques for practical learning and experimentation.**

7. Applications of the Project

This project can be applied in a variety of fields, including:

1. **Speech Enhancement**
    - **Removing unwanted noise from audio recordings to improve clarity.**
2. **Speech Recognition Systems**
    - **Pre-processing audio for automatic speech recognition (ASR) to improve accuracy.**
3. **Medical Voice Analysis**
    - **Detecting and monitoring vocal disorders by analyzing speech characteristics.**
4. **Telecommunication Systems**
    - **Enhancing speech quality in phone calls, VoIP, and online meetings.**
5. **Linguistic Performance Analysis**
    - **Calculating speech rate, silence duration, and word count for education, training, or public speaking assessment.**
6. **Acoustic Research**
    - **Studying the acoustic properties of speech or other sounds for academic and industrial research.**

8. Theoretical Background

This section provides a summary of the fundamental DSP concepts used in the project.

8.1 Fast Fourier Transform (FFT)

FFT is an efficient algorithm for computing the Discrete Fourier Transform (DFT) of a signal. It converts a time-domain signal into its frequency-domain representation, allowing us to identify key frequency components present in the audio.

In this project, FFT is used to:

- **Identify the dominant frequency of speech,**
- **Calculate the mean frequency based on magnitude distribution,**
- **Plot the magnitude spectrum (|Y(f)|) for frequency analysis.**

8.2 Bandpass Filtering

Human speech typically lies between 300 Hz and 3400 Hz. To isolate this range and remove unwanted noise, we used a 4th-order Butterworth bandpass filter, implemented using MATLAB's butter and filtfilt functions. This step ensures better clarity and focuses analysis on the vocal range.

8.3 Time-Domain Analysis

In the time domain, the system performs the following:

- **Computes RMS (Root Mean Square) values per frame to detect energy,**
- **Determines silence frames based on a threshold RMS,**
- **Calculates speech duration by subtracting silence time from the total duration,**
- **Estimates syllables and words, assuming average speech patterns.**

8.4 Spectrogram

A spectrogram visualizes the signal's energy across time and frequency, showing how the spectrum changes dynamically. It helps in detecting voiced/unvoiced segments and understanding speech quality.

8.5 Speech Rate Estimation

The system assumes an average speech rate of 4 syllables per second, and 1.5 syllables per word, to estimate the number of words and the overall speech rate (words per second).

9. Project Background

Digital Signal Processing (DSP) is essential in audio and speech-based systems, enabling the transformation and analysis of signals for various applications. The human voice primarily lies within the 300–3400 Hz range, which is why telecommunication systems and voice analyzers often apply bandpass filters in this range.

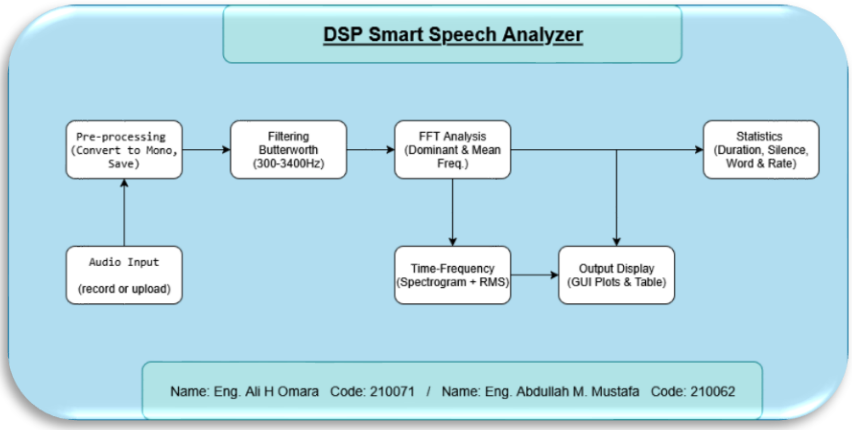
This project serves as a practical implementation of DSP concepts learned in the course, combining:

1. **Signal Acquisition – Recording or uploading audio files.**
2. **Filtering – Using a 4th-order Butterworth bandpass filter.**
3. **Frequency Analysis – Applying FFT to examine spectral properties.**
4. **Speech Metrics Calculation – Speech rate, silence detection, dominant frequency.**
5. **Visualization – Time plots, frequency spectra, spectrograms, and RMS graphs.**

By integrating these elements, the project simulates real-world audio processing systems and reinforces the connection between DSP theory and hands-on MATLAB implementation.

10. System Design & Methodology

The intelligent speech analyzer is designed to provide an interactive environment for processing speech signals using digital signal processing concepts. The system flow consists of several main stages, illustrated in the following block diagram:



10.1 System Architecture

The project is divided into six core functional blocks:

1. **Audio Input Stage**
   - Record audio using MATLAB's audiorecorder.
   - Upload existing audio files (.wav, .mp3, .opus).
2. **Pre-processing (Conversion to Mono, Save)**
   - The input audio is converted to a mono channel to simplify processing, then stored in a standard format for subsequent DSP operations.
3. **Bandpass Filtering (Butterworth 300–3400 Hz)**
   - A 4th-order Butterworth bandpass filter is applied to remove unwanted noise and retain only the human speech frequency range of 300–3400 Hz.
4. **FFT Analysis (Dominant & Mean Frequency)**
   - The filtered signal is transformed into the frequency domain using FFT to determine both the dominant frequency and the mean frequency of the speech.
5. **Time-Frequency Analysis (Spectrogram, RMS)**
   - A spectrogram is generated to visualize frequency variation over time, and the RMS (Root Mean Square) plot is used to analyze signal energy distribution.
6. **Statistics (Duration, Silence, Word Count & Speech Rate)**
   - The system calculates key time-domain metrics such as total duration, silence duration, active speech duration, estimated number of spoken words, and speech rate.
7. **Output Display (GUI Plot & Table)**
   - All processed results and visualizations are displayed in the MATLAB GUI, including plots and statistical tables for easy interpretation.

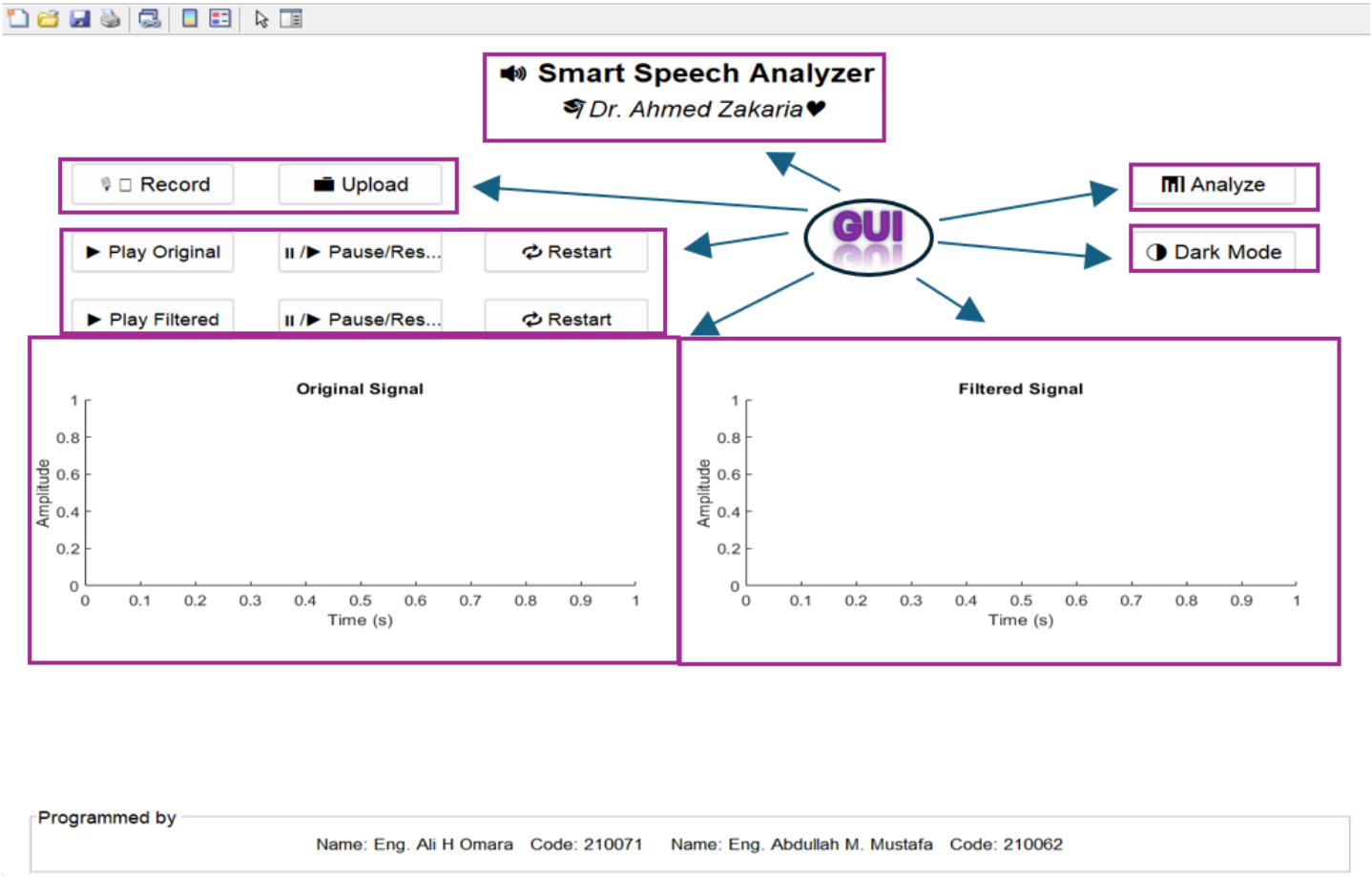
11. Software Implementation (MATLAB GUI)

The entire project is written in MATLAB, structured around a single GUI function:

```
function SmartSpeechAnalyzer_GUI()
```

11.1 GUI Layout

The main window contains:



- **Buttons to Record, Upload, Play, Pause, Restart signals.**
- **Buttons for Analysis and Dark Mode.**
- **Axes for plotting Original and Filtered signals.**
- **Developer info panel.**

12. DSP-Related Code Extracts & Explanations

```
[b, a] = butter(4, [300 3400]/(fs/2), 'bandpass');
```

- **Creates a 4th-order Butterworth bandpass filter to pass frequencies between 300 Hz and 3400 Hz (human speech range).**

```
filtered_y = filtfilt(b, a, y);
```

- **Applies zero-phase filtering to avoid distortion in the filtered signal.**

```
Y = fft(y);
```

- **Computes the Fast Fourier Transform (FFT) of the original signal to analyze frequency content.**

```
f_axis = (0:N-1)*(fs/N);
```

- **Generates the frequency axis for plotting the FFT results.**

```
mag = abs(Y);
```

- **Calculates the magnitude spectrum from FFT output.**

```
[~, maxIdx] = max(halfMag);
dominantFreq = halfFreq(maxIdx);
```

- **Finds the dominant frequency by locating the maximum magnitude.**

```
spectrogram(y,256,200,512,fs,'yaxis');
```

- **Generates a spectrogram showing how frequency components change over time.**

12.1 Core Features

a. Audio Recording & Uploading

- **Recording is done using:**

```
recObj = audiorecorder(fs, 16, 1);
recordblocking(recObj, 5);
```

- **Uploading is done via file dialog using:**

```
uigetfile({'*.wav;*.mp3;*.opus','Audio Files'});
```

b. Filtering

- **A Butterworth filter is applied using:**

```
[b, a] = butter(4, [300 3400]/(fs/2), 'bandpass');
filtered_y = filtfilt(b, a, y);
```

c. FFT Analysis

- **Frequency domain data is extracted using:**

```
Y = fft(y);
mag = abs(Y);
f_axis = (0:N-1)*(fs/N);
Dominant frequency is calculated as:
[~, maxIdx] = max(halfMag);
dominantFreq = halfFreq(maxIdx);
```

- **Mean frequency:**

```
meanFreq = sum(halfFreq .* halfMag') / sum(halfMag);
```

d. Speech & Silence Detection

- **The RMS of each frame is calculated to detect silent parts:**

```
rms(frame) < threshold
```

- **Speech duration and rate are estimated:**

```
speechDuration = totalDuration - silenceDuration;
estimatedSyllables = speechDuration * 4;
estimatedWords = estimatedSyllables / 1.5;
speechRate = estimatedWords / totalDuration;
```

e. Plotting

- **Time-domain and FFT plots are drawn using subplot and plot.**
- **Spectrogram is generated using MATLAB's spectrogram function.**

f. Dark Mode Toggle

- **GUI elements and axes colors are toggled based on darkMode boolean:**

```
set(allUI(i), 'BackgroundColor', bg);
set(allUI(i), 'ForegroundColor', fg);
```

13. Testing & Results

After building the full MATLAB GUI for Smart Speech Analyzer, the system was tested using several recorded and uploaded speech samples. The objective was to verify the performance of each feature—especially the FFT-based analysis, bandpass filtering, and speech metric calculations.

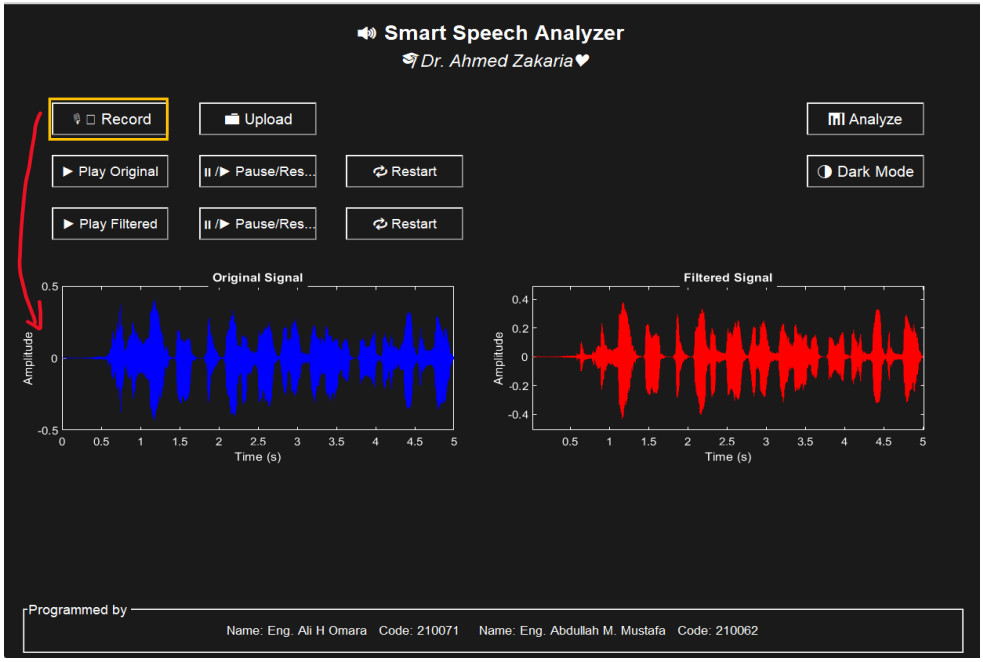
13.1 Test Environment

- **Sampling Rate: 44,100 Hz**
- **Recording Duration: 5 seconds (per session)**
- **Audio Types Tested:**
- **Normal speech in quiet room**
- **Background noisy speech**
- **Uploaded .wav and .mp3 files**
- **MATLAB Version: R2022b**

13.2 Output Results

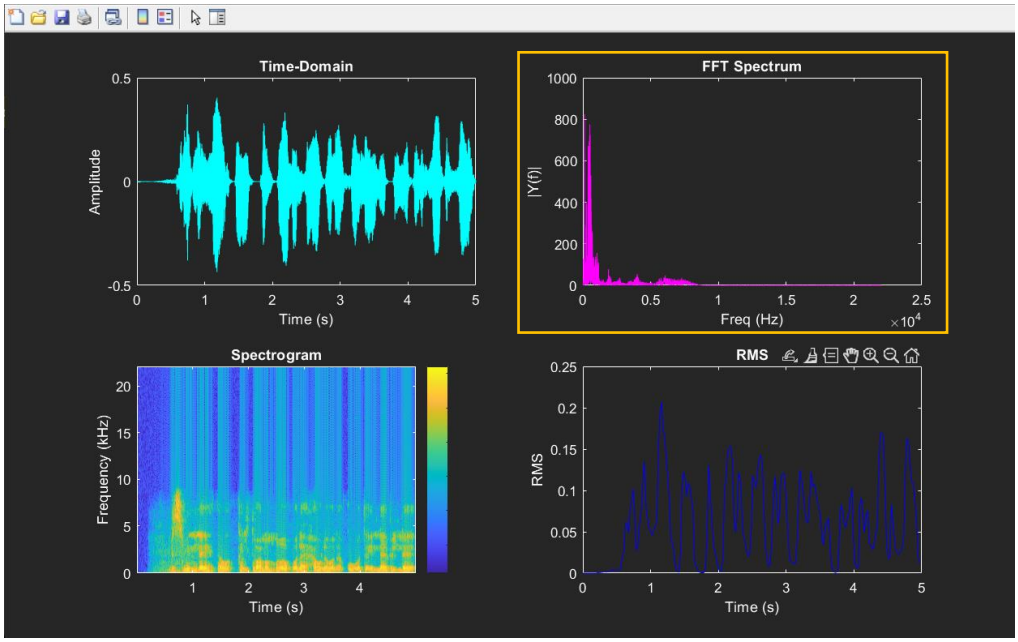
Each test produced the following results and visual outputs:

a. Time-Domain Plot (Original and Filtered)



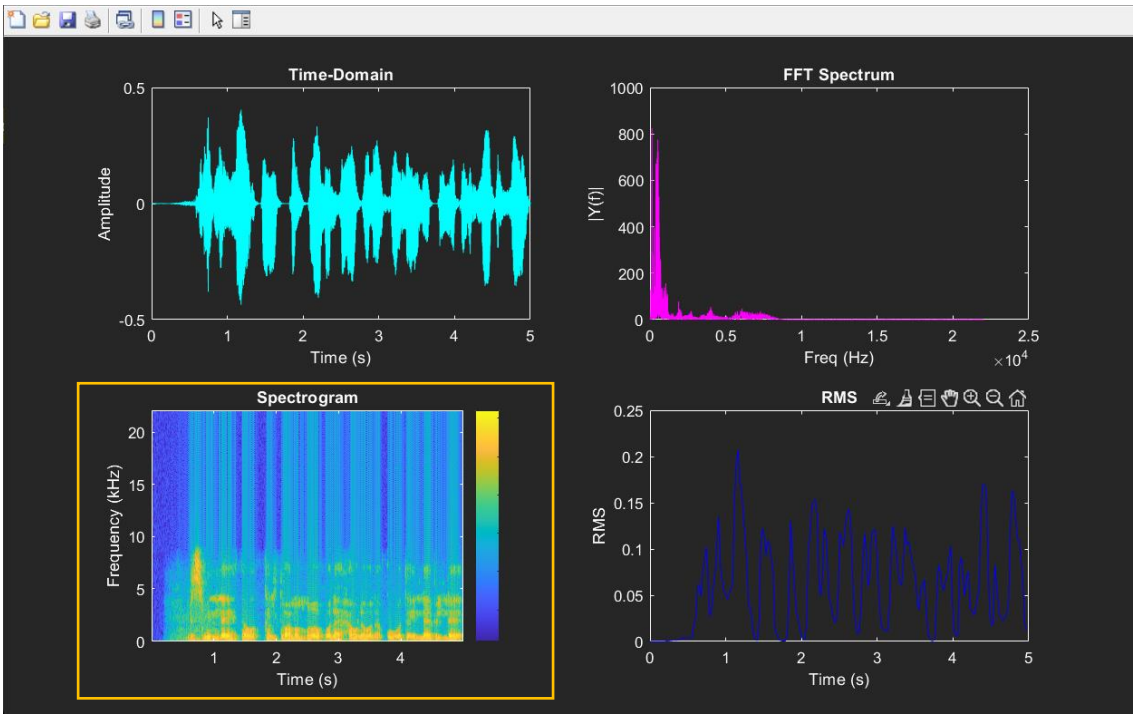
- **The original signal displayed raw amplitude fluctuations over time.**
- **The filtered signal (300–3400 Hz) showed a smoother waveform with removed background noise and high-frequency artifacts.**

b. FFT Spectrum



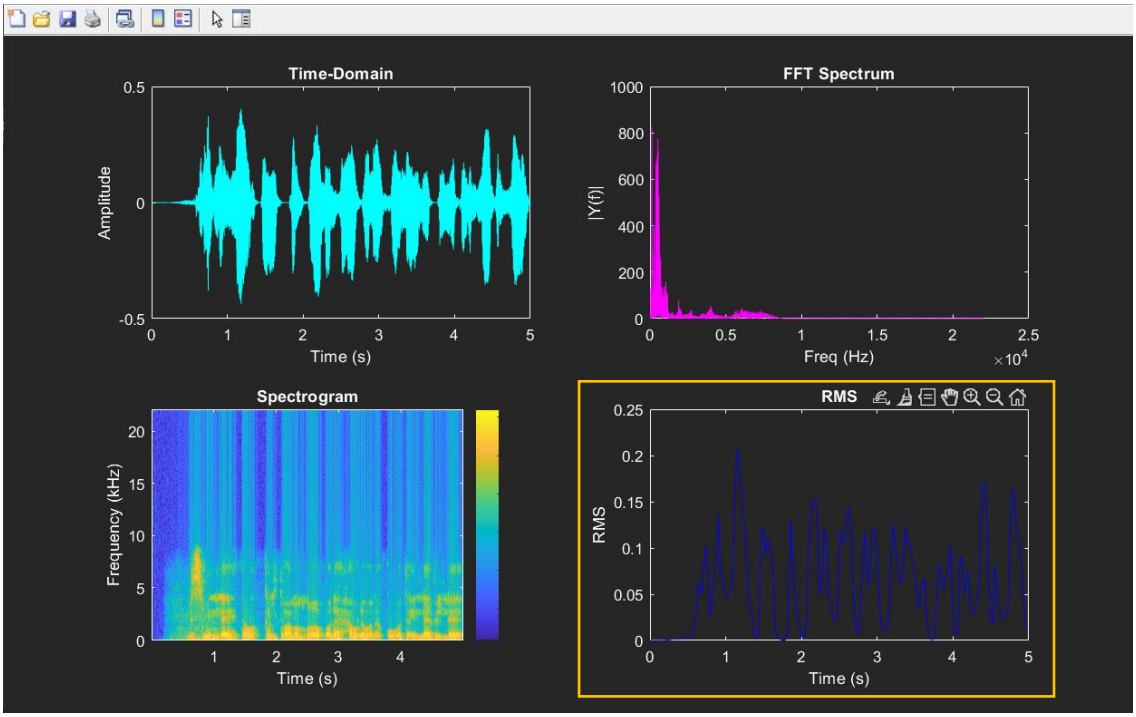
- **The FFT spectrum clearly identified the dominant frequency, usually between 100 Hz and 1000 Hz for male/female voices.**
- **The frequency resolution was sufficient to distinguish between voice types and detect background tones.**

c. Spectrogram



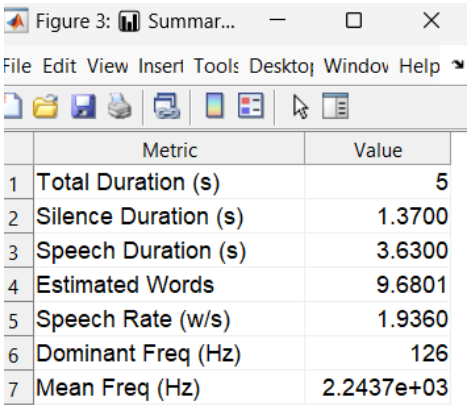
- **Displayed frequency content evolution over time.**
- **Showed voiced and unvoiced segments clearly, and silence segments were represented as darker regions.**

d. RMS Plot



- **RMS energy fluctuated with speech activity.**
- **Silent periods had low RMS values, allowing accurate speech segmentation.**

e. Summary Table Output Example



	Metric	Value
1	Total Duration (s)	5
2	Silence Duration (s)	1.3700
3	Speech Duration (s)	3.6300
4	Estimated Words	9.6801
5	Speech Rate (w/s)	1.9360
6	Dominant Freq (Hz)	126
7	Mean Freq (Hz)	2.2437e+03

14. Discussion

The project successfully achieved its intended goals and provided a rich environment for practical exploration of DSP principles using real-world signals. Below are key insights from the development and testing phases:

14.1 Significance of FFT in DSP

- **The Fast Fourier Transform (FFT) proved to be the most powerful tool for analyzing the speech signals in the frequency domain.**
- **It allowed for quick and accurate identification of dominant speech characteristics.**
- **The FFT output served as the backbone for features such as mean frequency calculation, speech classification, and spectral visualization.**

14.2 Filtering Performance

- **The Butterworth bandpass filter effectively cleaned up the input signal and isolated only the relevant voice range.**
- **The use of filtfilt ensured zero-phase distortion, which preserved the signal's structure while reducing noise.**

14.3 Practical Relevance

- **The speech rate and word estimation functions can be extended for real-time speech monitoring applications.**
- **The GUI design made the project accessible to users without coding knowledge, turning DSP concepts into interactive learning tools.**

14.4 Challenges

- **Detecting silence based purely on RMS sometimes misclassified low-volume speech.**
- **Background noise in uploaded files affected the accuracy of dominant frequency detection slightly.**
- **The project relies on fixed thresholds (e.g., RMS silence threshold = 0.02), which may need dynamic tuning for better generalization.**

14.5 Educational Value

This project provides a clear example of how MATLAB + DSP + GUI can be combined to turn theoretical signal processing into practical, visual, and interactive tools for both students and professionals.

15. Conclusion

The Smart Speech Analyzer - successfully demonstrates how fundamental Digital Signal Processing (DSP) techniques—especially the Fast Fourier Transform (FFT)—can be integrated into a real-world speech analysis tool using MATLAB. The software enables users to interactively explore, process, and analyze audio signals through a simple and intuitive graphical user interface (GUI).

Key outcomes of the project include:

- **Accurate frequency-domain analysis of speech signals using FFT.**
- **Enhanced audio clarity using bandpass filtering in the human voice range.**
- **Detailed time-domain and spectral visualization, including RMS and spectrograms.**
- **Estimation of speech metrics like silence duration, speech rate, and dominant frequency.**

The project not only fulfilled all technical objectives but also added value from an educational perspective by bridging theory with practical software development. It serves as a valuable learning resource for DSP students and practitioners alike.

16. Future Work

To further improve and expand this project, the following enhancements are proposed:

Functional Improvements:

- **Add automatic silence threshold adjustment based on background noise.**
- **Implement real-time recording visualization as the user speaks.**
- **Allow user-defined filter ranges beyond 300–3400 Hz for more flexibility.**
- **Include voice classification (male/female detection based on pitch).**

AI/ML Integration:

- **Use machine learning models to analyze speech emotion, language, or accent.**
- **Train models to auto-label spoken words or commands.**

Platform Expansion:

- **Convert the MATLAB GUI into a standalone desktop application.**
- **Develop a simplified mobile app version using MATLAB App Designer or cross-platform tools.**

These extensions would allow the project to grow from an educational tool into a semi-professional audio processing system.

17. References

1. **Oppenheim, A. V., & Schafer, R. W. (2010).** ***Discrete-Time Signal Processing*** **(3rd ed.). Pearson.**
2. **Proakis, J. G., & Manolakis, D. K. (2006).** ***Digital Signal Processing: Principles, Algorithms, and Applications*****. Pearson.**
3. **MathWorks Documentation. (n.d.).**
    - **FFT - MATLAB Function Reference**
    - **Butterworth Filter Design**
    - **GUI Development Environment**
4. **Rabiner, L. R., & Schafer, R. W. (1978).** ***Digital Processing of Speech Signals*****. Prentice Hall.**
5. **Smith, S. W. (1997).** ***The Scientist and Engineer's Guide to Digital Signal Processing*****. California Technical Publishing.**

18. Appendix

A.1 Comprehensive Project Model

https://github.com/aligedoo77/-Smart-Speech-Analyzer/blob/main/Speech_Analysis_Final_GUI.m

A.2 Project Info

Eng. Ali Hassan Omara

Code: 210071

Dr. Ahmed Zakaria

Eng. Abdullah M. Mustafa

Code: 210062

Keep You Happy

Thank You