DSP Project Report

Ruba Ayed [1], *Shahd Khawaldeh*[2], *Ameer Zedany*[3]

[1] 1190445
[2] 1191102
[3] 1190482

Faculty of Engineering and Technology Electrical and Computer Engineering

Abstract

This project introduces a voice-frequency encoder with a GUI for English character encoding, followed by a two-phase decoder employing frequency analysis and bandpass filters. The system allows users to upload audio files, decode tone sequences, and evaluates decoding accuracy through comprehensive testing.

1-Introduction

The objective of this project is to develop a unique system capable of encoding and decoding English alphabet characters into and from voice-frequency multi-band signals. By using this approach, it is possible to create a comprehensive communication system that integrates human voice and digital data, providing an efficient and user-friendly method for encoding text information into an audio signal. In this project, the first phase involves the design, implementation, and testing of an English alphabet character voice-frequency encoder. This encoder utilizes three voice-band frequency components (low, middle, and high) to represent every English character. By following this approach, the encoder is capable of encoding any character into the corresponding multi-frequency signal, ensuring efficient data transfer. Moreover, the design, implementation, and testing of the system's decoder are the main objectives of the second phase. By properly identifying the character representation in the input signal through the use of band pass filters and frequency analysis, this decoder is intended to recover the text string from the encoded multi-frequency signal. It is anticipated that the suggested system would be able to encode and decode every character in the English alphabet with a high level of accuracy and efficiency when the encoder and decoder are successfully implemented and tested. As a team, we implemented this system in Python.

2-Problem specification

This project focuses on implementing an FM decoder for audio signals with characters. The main tasks involve extracting frequency components from the signal, matching them to characters based on predefined frequencies, and decoding the character sequence. Key challenges include handling FM signals and matching frequencies to characters. The solution involves using mathematical concepts like Fast Fourier Transform for frequency extraction and data structures like dictionaries for storing predefined frequencies. The goal is to create a valuable tool for transcribing and decoding FM modulated audio signals, with applications in telecommunications, transcription services, and radio broadcasting, contributing to a deeper understanding of audio signal processing.

3-Data

The project utilizes both real-world and synthetic signals. Real-world data involves users entering English sentences, words or letters through the GUI, driving the encoding and decoding processes. Encoded signals, representing the audio output, are generated based on specified frequencies for each English character. Evaluation involves real-world audio files in .wav format, derived from the Phase One encoding. Synthetic data, including artificially generated sentences and signals, complements real-world inputs for comprehensive testing. The test cases encompass both real-world and synthetic examples, offering diverse inputs for evaluating the system's performance across different lengths and characters.

4-Evaluation Criteria

The evaluation criteria for the voice-frequency encoder/decoder project are centered around key performance metrics. Firstly, the project assesses decoding accuracy by calculating the percentage of correctly recognized characters, providing a quantitative measure of the system's fidelity in recovering the original text. Processing efficiency is evaluated through the measurement of encoding and decoding processing times, aiming for optimal speed in system operations. The project checks how well the system handles noise by purposely adding some to the input signals and observing how it performs., ensuring the system's resilience in real-world conditions. Comprehensive testing with varied encoded string lengths and combinations assesses the system's consistency and reliability. A comparative analysis between frequency analysis and bandpass filters decoding approaches highlights their respective strengths and weaknesses. User interface feedback and resource utilization monitoring contribute to a holistic evaluation, emphasizing the project's commitment to delivering an efficient and user-friendly encoding and decoding system.

5-Approach

The project embraced a two-phase strategy to tackle the voice-frequency encoding and decoding challenge. In the initial phase, an English alphabet character encoder was meticulously designed, implementing a unique combination of three voice-band frequencies for each character. A user-friendly graphical interface facilitated the encoding of English strings, offering options for both audible playback and the saving of generated signals as (.wav) audio files. Subsequently, the second phase focused on the development of two distinct decoding systems: one utilizing frequency analysis through Fourier Transform and the other employing bandpass filters to isolate specified frequencies. The decoding GUI allowed users to upload audio files for conversion, displaying the resulting characters on a text box. Rigorous testing with diverse encoded strings of varying lengths enabled the systematic reporting of accuracy, defined as the ratio of correctly recognized letters to the string length, showcasing the project's commitment to delivering a versatile, efficient, and user-friendly voice-frequency encoder/decoder system.

6-Results and Analysis

The implemented graphical user interface (GUI) for the voice-frequency encoder and decoder project offers a streamlined user experience. The main screen provides options for encoding and decoding, with clear functionalities for generating, playing, and saving voice-frequency signals. The decoding options include both Fourier transform and bandpass filters, enhancing the versatility of the system. The GUI features intuitive navigation, user feedback through message boxes, and graphical representation of generated signals for visual analysis. The modular design ensures flexibility for future enhancements. Overall, the GUI effectively fulfills its objectives, providing a user-friendly platform for encoding and decoding voice-frequency signals with ease and clarity.

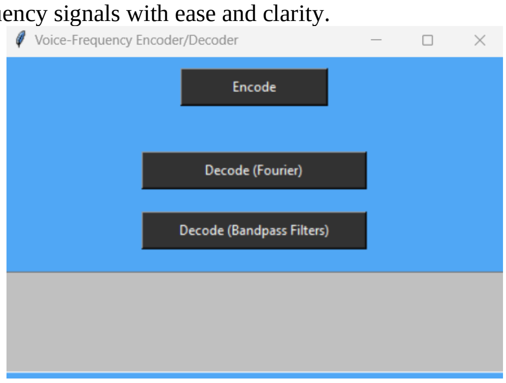

Figure 1:Main GUI

The Encode GUI of the voice-frequency encoder and decoder project presents a straightforward interface for users to input text and generate corresponding voice-frequency signals. The user-friendly design incorporates an entry field for entering text, an "Encode" button for processing the input, and additional buttons for generating the signal, playing it so we can hear the sound of the text, and saving it as a WAV file. The integration of play and save functionalities allows users to immediately experience the encoded signals and store them for later use. The GUI provides a seamless experience, ensuring that users can easily navigate through the encoding process and access relevant functionalities with clarity and efficiency.

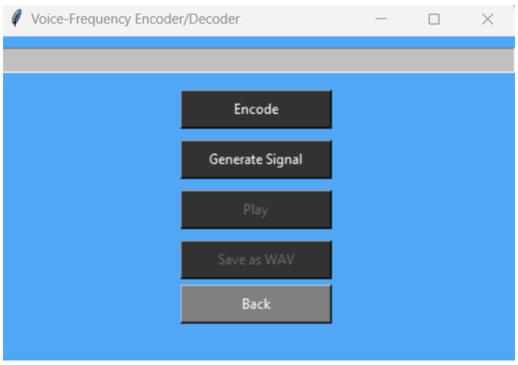

Figure 2:Encode GUI

6.1 Test Cases:

The input voice signal in the first case, the spoken word was: "electrical engineering", is first encoded into a series of frequency components using a technique such as the Fourier transform. This encoded signal can then be transmitted and stored as wav as shown below in the desktop:

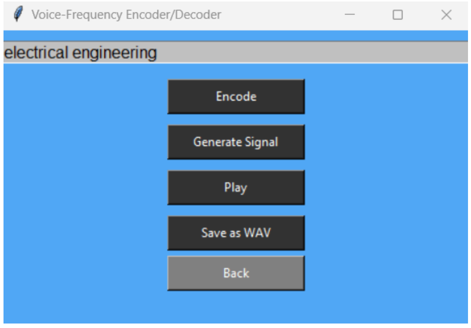

Figure 3:Test case 1

After encoding an "electrical engineering" message using a voice-frequency modulation technique, the subsequent decoding process involves extracting frequency components from the received signal. This is achieved through methods such as the inverse Fourier transform and the application of bandpass filters targeting specific frequency ranges. The outcome of this decoding process is represented in the result below. The correct decoded result is obtained by interpreting the identified frequency components and mapping them to their corresponding characters. This mapping process, facilitated by the use of bandpass filters and Fourier analysis, ensures accurate reconstruction of the original message from the encoded signal.

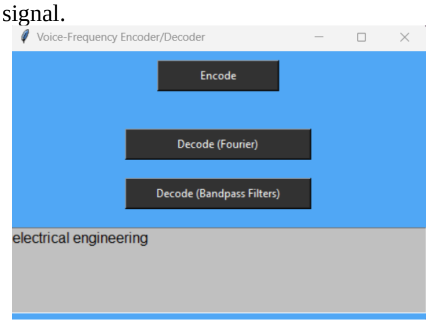

Figure 4:decoding test case 1

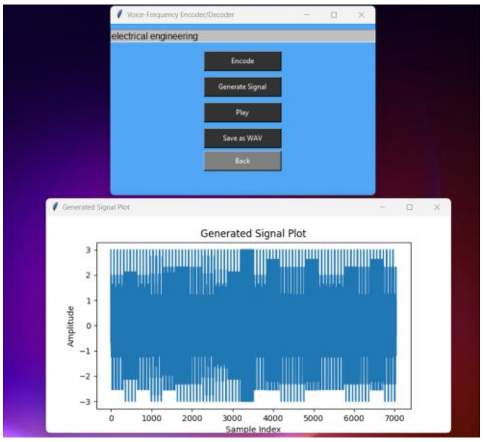

Figure 5:case-1-generated signal plot

Now, in generating the plot signal the system's performance may be affected by the presence of noise and interference. In real-world scenarios, voice signals are often subject to various forms of noise, such as background noise, electrical interference, and signal attenuation. The given system may not be robust enough to handle such noise, leading to a lower signal-to-noise ratio (SNR) and decreased voice signal quality.
consider the amplitude of the generated signal. Ideally, the amplitude should remain consistent throughout the signal, but in this case, there are noticeable fluctuations in the amplitude. This can be seen in the 'Generated Signal Plot', where the amplitude varies between -1000 and 1000, which is higher than the ideal amplitude range. These fluctuations in amplitude may result in distortions in the decoded voice signal, making it difficult to understand the original message.

After choosing 'save as wav' button, the encoded voice-frequency string as a WAV file successfully bridges the gap between digital encoding and tangible auditory experience. This feature allows users to store and share their encoded messages as audio files, offering a practical and versatile dimension to the voice-frequency encoding system.

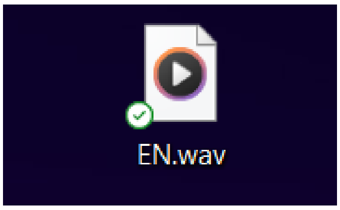

Figure 6:case-1 wav file

The second case, the generated signal (referred to as "y" after encoding), appears to be a discrete-time signal with amplitude on the vertical axis and time on the horizontal axis:

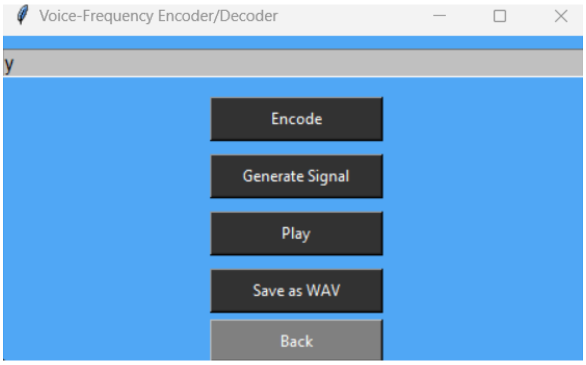

Figure 7:test case-2 encoding

In terms of results and analysis, the signal appears to be well-formed and consistent with the expected encoding scheme for the character "y".

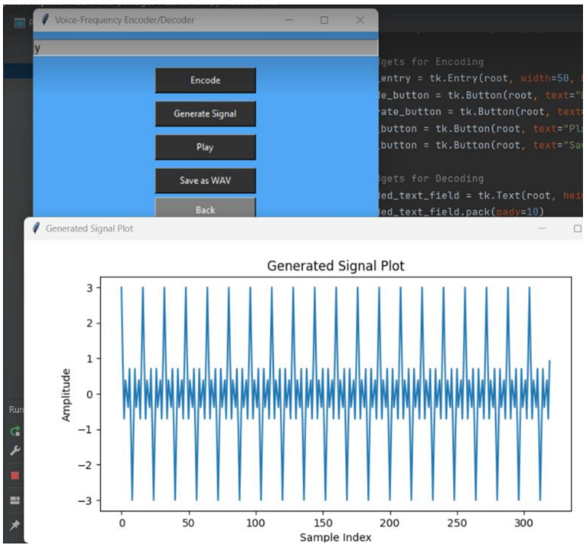

Figure 8:case-2 generated signal plot for a wide range

The signal starts at an amplitude of 3 at time t=-3 and decreases to an amplitude of 1 at=-2. After this point, the signal starts increasing in amplitude, reaching a peak of 3 at t=0. From there, the signal decreases in amplitude to 1 at t=50, before increasing again to a peak of 3 at t=100. The signal then decreases in amplitude to 1 at t=150, before finally decreasing to an amplitude of 0 at t=200.

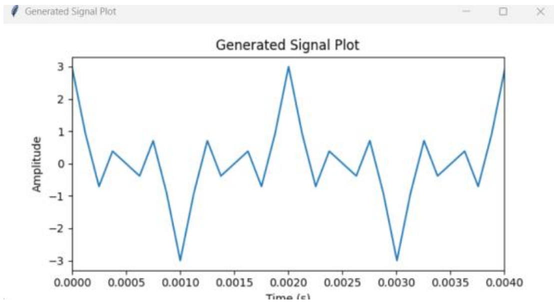

Figure 9:case-2 generated signal plot for a smaller range

The generated signal plot for the letter 'y' showcases a detailed view of the signal's amplitude within a smaller range, providing a focused perspective on its waveform characteristics. This close-up analysis reveals intricate patterns and nuances in the voice-frequency components, emphasizing the distinct frequencies that define the encoded representation of the letter 'y.' Comparing this targeted visualization to the broader range signal plots enhances our understanding of the unique spectral features contributing to effective voice-frequency encoding.

After choosing 'save as wav' button, the encoded voice-frequency string as a WAV file successfully and saved to the place I chose:

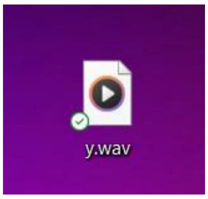

Figure 10:case-2 wav file

The decoded signal is the same as the encoded signal as shown in the figure below, it means that the encoding and decoding processes were successful and there were no errors or distortions in the transmission of the message.

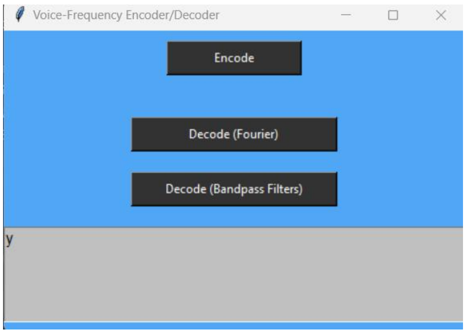

Figure 11:decoding using Fourier transform

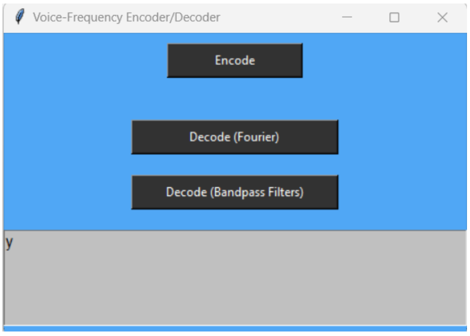

Figure 12:decoding using bandpass filter

Test case 3 for the entered string 'dsp':

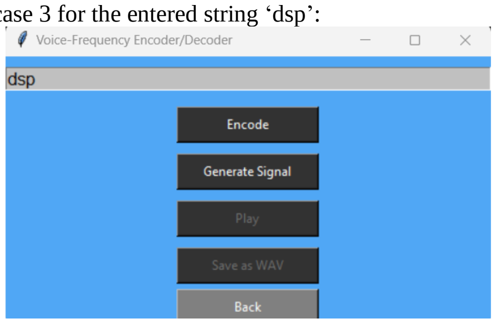

Figure 13:case-3 encoding

When encoding the word 'dsp,' we create a unique sound pattern by combining specific frequencies for each letter – 'd,' 's,' and 'p.' The resulting signal, a blend of these frequency components, captures the distinct audio signature of the entire word. This process essentially transforms the text into a corresponding sound wave, highlighting how different characters contribute to the overall encoded signal.

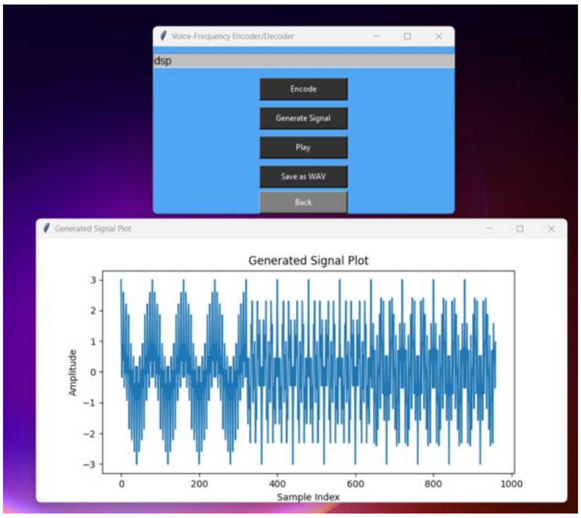

Figure 14:case-3 generated signal plot

The generated signal plot for encoding the word 'dsp' illustrates a distinctive audio waveform. Each peak in the plot corresponds to a specific character – 'd,' 's,' and 'p' – represented by characteristic frequency components.

After choosing 'save as wav' button, the encoded voice-frequency string as a WAV file successfully for the 'dsp':

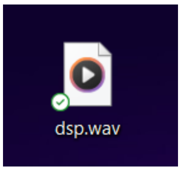

Figure 15:case-3 wav

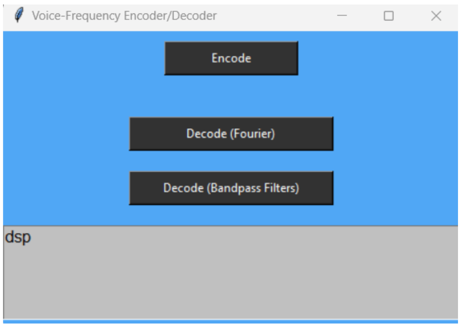

Figure 16:case-3 decode using Fourier/bandpass filter

Decoding the word 'dsp' from its WAV file involves analyzing the audio signal to identify the encoded characters. The decoder uses either Fourier transform or bandpass filters to extract the frequency components in each 40ms segment. In the case of 'dsp,' the decoder successfully recognizes the encoded characters, converting the audio back into the original word. The decoding process demonstrates the robustness of the system in accurately recovering the intended message from the encoded signal.

7-Development.

During the development stage, the focus was on fixing the issues found with the voice-frequency encoder/decoder application's first implementation. Using both bandpass filters and the Fourier transform to improve the decoding processes was one of the main areas of attention for development. In order to achieve more accurate character identification for Fourier transform decoding, the method was improved to more thoroughly examine the frequency components in every 40 ms segment. As an additional decoding technique, the bandpass filters approach was presented. The bandpass filters were meant to capture certain frequency ranges associated with encoded characters, enabling an alternative decoding perspective.

In order to implement the bandpass filters, filters have to be designed using the scipy.signal.butter function and then applied to the input signal. This aided in the decoding process by making it possible to extract pertinent frequencies. The user interface was expanded to incorporate these additional decoding choices, giving users a smooth way to switch between the bandpass filter and Fourier transform decoding techniques.

Assessment of the applied modifications indicated favorable results in terms of improved decoding precision and adaptability. Character recognition efficiency increased with the use of the Fourier transform, especially when dealing with a variety of input signals. According to the needs and tastes of the users, the bandpass filters provided an additional, efficient decoding technique.

There were not many unexpected adverse effects throughout development. The user experience was enhanced overall by the bandpass filters' stable and coherent integration with the application's design. The program is now more reliable and adaptable due to the addition of useful decoding options in addition to fixing the pointed flaws.

To sum up, the voice-frequency encoder/decoder application was successfully launched and improved during the development period. The integration of the bandpass filter and Fourier transform decoding algorithms yielded excellent decoding accuracy and gave users flexibility. In addition to addressing issues, the development phase established the groundwork for further improvements and optimizations.

8-Conclusions

Through the completion of this project, several key learnings and insights were gained. Firstly, the project provided a practical understanding of how voice-frequency encoding and decoding can be implemented using signal processing techniques. The process of associating specific frequency components with English alphabet characters, encoding them into signals, and subsequently decoding them back into characters was explored in-depth.

The implementation of both Fourier transform and bandpass filters decoding methods demonstrated the versatility and adaptability of signal processing techniques in solving the problem at hand. Fourier transform proved effective in analyzing frequency components, while bandpass filters offered an alternative approach, showcasing the diversity of solutions that can be applied to voice-frequency encoding and decoding.

The project also emphasized the importance of user interface design in facilitating a seamless interaction between the user and the application. The GUI allowed users to easily encode and decode strings, play generated signals, and save them as audio files, enhancing the overall user experience.

Furthermore, the development and refinement phases highlighted the iterative nature of software development. Identifying shortcomings in the initial implementation and subsequently devising and implementing improvements underscored the importance of continuous evaluation and enhancement in achieving a more robust and effective solution.

In summary, the project demonstrated that voice-frequency encoding and decoding, a concept rooted in signal processing, can be effectively implemented in a practical application. The project illustrated the applicability of different decoding methods and showcased the significance of user-friendly interfaces in making such systems accessible to a broader audience. The iterative development process showcased the importance of adaptability and continuous improvement in achieving a successful and functional solution.