Voice-Frequency Encoder and Decoder for English Alphabet.

Razan Abdalrahman[1]
1200531
Electrical and Computer Enginnering Department
Birzeit University
1200531@student.birzeit.edu

Osaid Hamza[2]
1200875
Electrical and Computer Enginnering Department
Birzeit University
1200875@student.birzeit.edu

Maisam Alaa[3]
1200650
Electrical and Computer Enginnering Department
Birzeit University
1200650@student.birzeit.edu

Abstract

This paper presents a two-phase project focused on designing and implementing an English alphabet character voice-frequency encoder and decoder system. In Phase One, an encoder is developed to represent characters using three voice-band frequency components. A user-friendly GUI facilitates encoding of English strings, with options to play the generated signal or save it as a .wav audio file. Phase Two introduces two decoding systems based on frequency analysis and bandpass filters, enabling recovery of text strings from encoded multi-frequency signals. The GUI allows users to upload audio files, convert tones into characters, and displays results. The accuracy of the systems is evaluated through testing with various encoded strings, providing a versatile solution for secure communication.

Keywords— Multi-frequency signal, Bandpass filters, Encoder, Decoder, Fourier transform, Accuracy evaluation.

1. Introduction

This project revolves about the design and implementation of a user-friendly interface for a voice-frequency encoding and decoding system for English characters. The motivation behind this project lies in the exploration of innovative methods to encode English characters each with a unique combination of low, middle, and high frequencies into unique voice-frequency signatures and subsequently decode them with high accuracy using two distinct methods: frequency analysis and bandpass filters. Such a system combines elements of signal processing, graphical user interface development, and algorithmic design and it can be found in secure communication, audio data transmission, and voice recognition technologies applications.

1.1. Voice-frequency encoder

Or a vocoder is a category of speech coding that analyzes and synthesizes the human voice signal for audio data compression, multiplexing, voice encryption or voice transformation [1].

1.2. Voice-frequency decoder

System designed to interpret or decode voice signals that have been encoded or modulated using specific frequency components. This term is commonly associated with telecommunication and signal processing [2].

2. Problem Specification

The system is divided into two phases: Phase One focuses on the creation of an English alphabet character voice-frequency encoder, while Phase Two involves designing a decoder system to recover text strings from the encoded multi-frequency signals.

2.1. Phase One: Multi-Frequency Encoder Design

2.1.1. Objective:

Design and implement an English alphabet character voice-frequency encoder that represents each character through a combination of three voice-band frequency components (low, middle, and high).

2.1.2. Specifications:

- Utilize the provided table (Table 2.1) with frequencies ranging from 100Hz to 3500Hz for each English character.
- Assume a character signal duration of approximately 40ms.
- Implement a graphical user interface (GUI) allowing users to encode English strings and generate corresponding signals.
- Provide options to play the generated signal or save it as a (.wav) audio file.

Table 2-1 Encoding frequencies for each English character.

Character	Low frequency component	Middle frequency component	High frequency component
a	100	1100	2500
b	100	1100	3000
c	100	1100	3500
d	100	1300	2500
e	100	1300	3000
f	100	1300	3500
g	100	1500	2500
h	100	1500	3000
i	100	1500	3500
j	300	1100	2500
k	300	1100	3000
l	300	1100	3500
m	300	1300	2500
n	300	1300	3000
o	300	1300	3500
p	300	1500	2500
q	300	1500	3000
r	300	1500	3500
s	500	1100	2500
t	500	1100	3000
u	500	1100	3500
v	500	1300	2500
w	500	1300	3000
x	500	1300	3500
y	500	1500	2500
z	500	1500	3000
space	500	1500	3500

2.2. Phase Two: Multi-Frequency Decoder Design

2.2.1. Objective:

Design, implement, and test a decoder capable of recovering text strings from the encoded multi-frequency signals generated in Phase One. Two decoding approaches must be implemented: frequency analysis using Fourier transform and bandpass filters.

2.2.2. Specifications:

- Implement two decoding systems, one using frequency analysis and the other using bandpass filters.
- Design a GUI enabling users to upload audio files and initiate the conversion of encoded tones into corresponding characters.
- Conduct extensive testing with various encoded strings of different lengths to report the accuracy of both decoding systems.
- The accuracy is defined as the number of correctly recognized letters divided by the string length.

3. Data

The successful execution of the project relies primarily on two key types of input data: User-supplied strings and the.wav file generated as an output in Phase One These files are essential since they facilitate Phase Two's input, which seeks to decode the strings from them.

3.1. Phase One

3.1.1. Input data.

User provide sequences of English characters "Strings" to be encoded.

3.1.2. Usage.

The user inputs an English string into the graphical user interface (GUI) to trigger the encoding process.

3.1.3. Output data.

The system generates a multi-frequency signal corresponding to the input English string in the form of a .wav file.

3.2. Phase Two

3.2.1. Input data.

The .wav file generated in Phase One, containing the encoded multi-frequency signals.

3.2.2. Usage.

The user uploads the .wav file through the GUI to initiate the decoding process.

3.2.3. Output data.

The system retrieves and displays the original text string from the encoded multi-frequency signals using two ways frequency and filter.

4. Evaluation Criteria

To evaluate the performance of this project, we will employ the following objective, quantitative, and discriminatory criteria.

4.1. Encoding Accuracy

To measure the accuracy of the voice frequency encoder we need to compare the encoded characters with the known frequency combinations, the percentage of the correct encoded characters is the accuracy.

4.2. Decoding Accuracy

To measure the accuracy of the decoding we need to evaluate the accuracy of both decoding systems frequency analysis and bandpass filters, by comparing the decoded text with the original input.

4.3. GUI Usability

To measure the usability of the GUI we need to evaluate the clearness of the instructions, how easy the interaction was and the overall user satisfaction.

4.4. Real-time Performance

To measure the system's real-time performance we need to evaluate the time taken to process different lengths of input strings and assess the system's responsiveness during encoding and decoding tasks.

4.5. Adaptability

To measure the system's adaptability we need to evaluate the system's behavior to varying input conditions, such as different signal durations and levels of background noise.

5. Approach

5.1. Phase one approach

- Character encoding.

This approach combines two functions: one to generate individual audio signals for each character in English using assigned frequencies, and another to combine those signals into an entire representation of a sentence including pauses.

- Signal play back and saving.

This approach integrates playback in real time with the help of a "sounddevice" library, which makes it possible to listen immediately generated audio signals. It also gives you the freedom to save your work as wav files within any current project directory for easy sharing and future access.

5.2. Phase two approach

- Frequency-Based Decoder.

Through Fourier Transform, analyze short audio segments and determine the frequencies that dominate them before mapping these back to their respective English characters. The decoded text is immediately shown within the user interface.

- Filter-Based Decoder.

Customized bandpass filters specific to each character allow the input signal through, thus exposing their unique frequencies. We decode and map these frequencies to letters by analyzing short bursts of the filtered sound, thus revealing the hidden speech content in user interface.

6. Results and Analysis

6.1. Phase One (Encoding) analysis

- The first phase is a voice-frequency encoder implemented using the Tkinter library for the graphical user interface (GUI).
- It uses a set of predefined frequencies for each English character to generate a signal.
- The signal is generated by combining cosine waves with frequencies corresponding to the chosen character.
- The encoded signal can be saved as a WAV file, played, and its time and frequency domains can be visualized.

6.2. Phase Two (Decoding) analysis

- The second phase is a voice-frequency decoder implemented using Tkinter as well.
- It provides two decoding methods: one using the raw frequencies and the other using bandpass filters.
- The decoder reads a WAV file, processes it in 40ms segments, and decodes the frequencies or filters to identify the characters.
- The decoded text is displayed in a text box.

6.3. Test case

Below is an example showcasing to assess the program's functionality in both the encoding and decoding stages:

1. Phase One "Encoder":

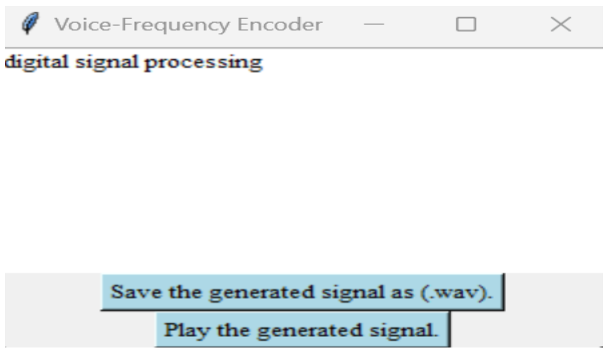

Figure 6-1 Input strings.

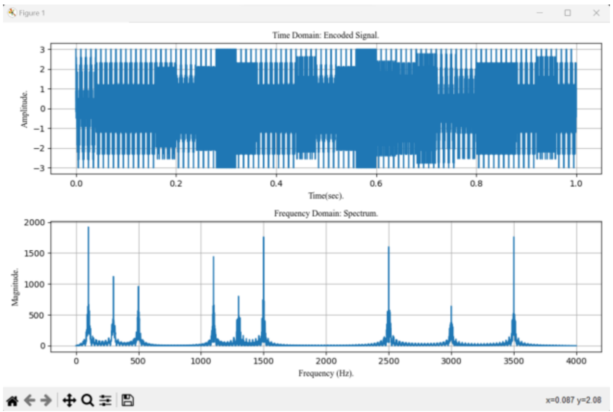

Figure 6-2 The signal in time and frequency domains.

2. Phase Two "Decoder":

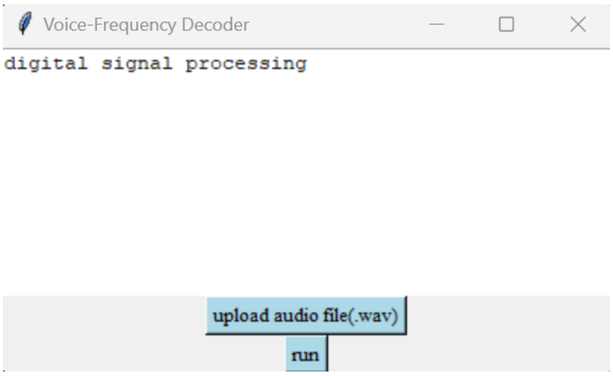

Figure 6-3 Decode using filter result.

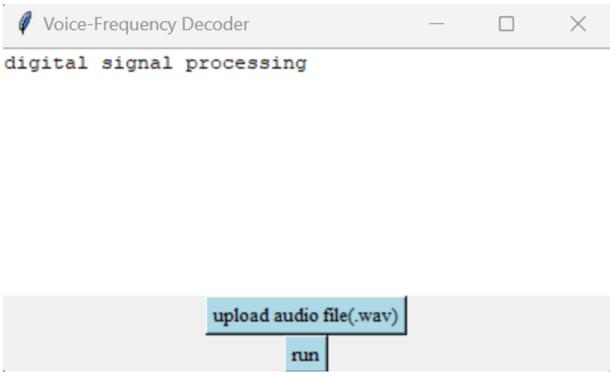

Figure 6-4 Decode using frequency result.

7. Development

The encoding and decoding processes were improved to make them more robust. Special characters and digits within the input string are now handled with care, ensuring smooth handling of various types of content. This enhancement greatly enhances the reliability and versatility of the system.

7.1. Encoding improvement

The encoding process has been improved to include special characters and digits in the input string, extending its range beyond just English letters. This upgrade guarantees a more extensive representation by cleverly assigning special characters and digits to distinct frequency combinations, thus expanding the system's ability to handle a wide variety of input content.

And that appear in the following code:

```
# Function to encode a string into a signal.
def encode_string_to_signal(input_string):
    signal = []
    for char in input_string:
        if char in FREQUENCIES: # if the character is in the dictionary.
            signal = np.concatenate( (signal,
            generate_character_signal(FREQUENCIES[char]) ) , axis=None )
        else:
            # If the character is not in the dictionary
            # add a some frequencies to the signal not in the dictionary.
            signal = np.concatenate( (signal,
            generate_character_signal([600, 1600, 2600]) ) , axis=None )
    return signal
```

Figure 7-1 Handling special characters in encoding stage

7.2. Decoding improvement

The decoding system has been enhanced to effectively handle special characters and digits. If these characters are found in the encoded signal, the decoder will display a "?" in the output. This serves as a visual clue that these characters are not included in the original English alphabet encoding scheme, helping with accurate decoding.

And that appear in the following code:

```
# Create a reverse mapping from frequencies to characters.
REVERSE_FREQUENCIES = {tuple(sorted(values)): key for key, values in FREQUENCIES.items()}

def decode_frequencies(frequencies):
    frequencies = tuple(sorted(frequencies))
    # Return '?' for unknown frequencies.
    return REVERSE_FREQUENCIES.get(frequencies, '?')
```

Figure 7-2 Handling special characters in decoding stage.

8. Conclusions

In conclusion, this project has provided valuable experience in signal processing, GUI design, and algorithm design, leaving us with a deeper understanding of communication's hidden melodies, to address the challenge of voice-frequency encoding and decoding for English characters. The encoder phase has shown that it's an efficient method to represent characters through unique frequency signatures, while the decoding phase demonstrated the effectiveness of both frequency analysis and bandpass filters in accurately recovering the original text from encoded signals. The project not only solved the problem but also provided a good understanding of how different approaches can lead to innovative solutions in the realm of secure communication and data processing.

9. References

[1] [Online]: https://en.wikipedia.org/wiki/Vocoder
[Accessed 22 1 2024].

[2] [Online]:https://www.zegocloud.com/blog/audio-encoding
[Accessed 22 1 2024].