DSP PROJECT: Multi-Frequency Encoding and Decoding System

Dr. Ashraf Al-Rimawi, Dr. Abualseoud Hanani

[1] Abdalkarim Eiss 1200015, sec 1
[2] Razi Atyani 1200028, sec 2
[3] Saleh Zhour 1201941, sec 1

Abstract

The primary objective of this project revolves around the implementation and design of both a string encoder and its corresponding graphical user interface (GUI). Additionally, the project focuses on the implementation and design of a string decoder aimed at recovering the original string as individual characters, complemented by its respective GUI. Furthermore, the project aims to assess the accuracy of the implemented systems by different approaches. The evaluation will gauge the effectiveness of both encoding and decoding processes, shedding light on their performance and overall efficiency.

1. Introduction

The project entails the design and implementation of an English alphabet character voice-frequency encoder and decoder system, divided into two distinct phases. Initially, in the encoding phase, we will develop an encoder capable of representing English characters through specific combinations of low, middle, and high frequencies. This encoder will generate a signal corresponding to a given character based on predefined frequency components. Subsequently, in the decoding phase, we will design two distinct decoding systems. The first decoding system will utilize frequency analysis, such as Fourier transform, to decipher the encoded multi-frequency signals. The second decoding system will employ bandpass filters to recover the text from the encoded signals. These decoding methods aim to reconstruct the original text from the multi-frequency representations generated by the encoder, providing comprehensive insights into the efficacy of each approach.

2. Problem Specification

The technical problem at hand involves designing a system for encoding and decoding English characters through multi-frequency signals. Specifically, the goal is to represent each English character by a combination of three voice-band frequency components (low, middle, and high). The encoding process maps characters to specific frequency components, creating a corresponding signal with a duration of approximately 40ms. The challenge extends to creating a user-friendly graphical interface for users to input English strings, encode them into corresponding signals, and choose between playing the generated signal or saving it as a .wav audio file. In the decoding phase, the task is to recover the original text from the encoded multi-frequency signal. Two decoding approaches are proposed: one utilizing frequency analysis, such as Fourier transform, and the other employing bandpass filters to isolate relevant frequencies. The overall technical goal is to develop robust encoding and decoding systems, ensuring accurate representation and recovery of English characters through the manipulation of voice-band frequency components.

3. Data

The primary data used in this project is English string that contains just characters which we need to connect them to the set of voice-band frequency components assigned to each English alphabet character. Additionally, audio signals in the form of .wav files are used for testing and evaluating the decoding systems.

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

Figure 1:Encoding frequencies for each English character

4. Evaluation Criteria

The project's success will be measured based on the accuracy of the decoding systems. The accuracy is defined as the number of correctly recognized letters divided by the string length multiply with 100%. Objective and quantitative measures will be employed to assess the performance of the frequency analysis and bandpass filter decoding methods.

The accuracy of our two systems DFT and BPF almost equal 100% as we tested it with different paragraphs with different lengths. For example, we tested it with paragraph that contains 20 words it gives us all of them right.

5. Approach

The project follows a two-phase approach. In the first phase, an encoder is implemented based on the provided frequency components for each character. A GUI is built to facilitate user input and signal generation. In the second phase, two decoding systems are designed using frequency analysis and bandpass filters. A GUI is created for users to upload audio files, choose a decoding method, and view the decoded results.

(We used NumPy library[3])

First decode system is Frequency analysis (DFT): we take the generated signal and cut the signal to intervals each interval is 40ms then take the highest three peaks for each character.[2]

```python
def frequency_analysis(self, chunk, plot=True):
    # Analyze frequencies in the input chunk
    spectrum = np.fft.fft(chunk)
    freqs = np.fft.fftfreq(len(chunk), 1 / self.sample_rate)
    positive_freqs = freqs[freqs > 0]
    positive_spectrum = spectrum[freqs > 0]
    peaks, _ = find_peaks(np.abs(positive_spectrum), height=0)
    sorted_peaks = sorted(peaks, key=lambda
        x: np.abs(positive_spectrum[x]), reverse=True)
    highest_peaks = sorted_peaks[:3]
    pulse_frequencies = positive_freqs[highest_peaks]

    return pulse_frequencies
```

Figure 2:code of DFT

Second decode system is Band Pass Filter (BPF): we take the generated signal and we have made 3 filters for each char (low, middle, high) entered it to BPF. Low and High is taken based on the frequencies.[1]

```python
for char, char_frequencies in self.character_frequencies.items():
    if isinstance(char_frequencies, (list, tuple)) and len(char_frequencies) == 3:
        lowcut, middle, highcut = char_frequencies
    else:
        raise ValueError(f"Invalid format for char_frequencies: {char_frequencies}")

    nyquist = 0.5 * self.sample_rate

    # Design lowpass filter
    low = (lowcut - bandwidth/2) / nyquist
    high = (lowcut + bandwidth/2) / nyquist
    b_low, a_low = butter(5, [low, high], btype='band')
    filters[char] = {'b_low': b_low, 'a_low': a_low}

    # Plot lowpass filter
   # self.plot_filter_response(b_low, a_low, char, 'Lowpass Filter')

    # Design bandpass filter for middle frequency
    low = (middle - bandwidth/2) / nyquist
    high = (middle + bandwidth/2) / nyquist
    b_middle, a_middle = butter(5, [low, high], btype='band')
    filters[char]['b_middle'] = b_middle
    filters[char]['a_middle'] = a_middle

    # Plot bandpass filter for middle frequency
    #self.plot_filter_response(b_middle, a_middle, char, 'Bandpass Filter (Middle)')

    # Design highpass filter
    low = (highcut - bandwidth/2) / nyquist
    high = (highcut + bandwidth/2) / nyquist
    b_high, a_high = butter(5, [low, high], btype='band')
    filters[char]['b_high'] = b_high
    filters[char]['a_high'] = a_high

    # Plot highpass filter
    #self.plot_filter_response(b_high, a_high, char, 'Highpass Filter')
```

Figure 3:Code of BPF

6. Result and Analysis

Upon evaluating the system using the specified criteria, several observations and shortcomings became apparent. Firstly, during testing, it was observed that the accuracy of the decoding systems varied based on the complexity and length of the encoded strings. The frequency analysis-based decoder demonstrated challenges in accurately identifying characters in instances where there was interference or overlapping frequencies. This was particularly evident when decoding longer sequences, leading to a decline in overall accuracy.

The filter-based decoder, while generally effective, faced issues with signal distortion and loss of information due to the inherent limitations of bandpass filters. Certain frequency components crucial for accurate decoding were occasionally filtered out, leading to errors in the reconstructed text. This highlighted the sensitivity of the filter-based approach to variations in input signal quality and underscored the need for robust filtering techniques.

Additionally, the GUI, while functional, lacked advanced features for signal visualization and user feedback during the encoding and decoding processes. Incorporating more comprehensive visualizations could enhance user understanding and troubleshoot issues related to signal processing.

In summary, the evaluation revealed that both decoding systems had their strengths and weaknesses. The frequency analysis approach struggled with interference, while the filter-based method faced challenges in preserving essential frequency components. Improvements in signal visualization and user feedback mechanisms would be essential for a more user-friendly and effective system. This analysis underscores the intricate nature of signal processing concepts and the need for nuanced approaches to address the complexities inherent in encoding and decoding multi-frequency signals.

7. Development

To address identified decoding shortcomings, adjustments were made to both frequency analysis and bandpass filter methods. In frequency analysis, parameters were fine-tuned to enhance accuracy, resulting in improved decoding. However, increased computational complexity was observed. For bandpass filters, parameter refinement reduced cross-talk but necessitated a delicate balance between selectivity and noise. While improvements were realized, ongoing optimization is essential, considering trade-offs and unexpected side effects. Future efforts will focus on algorithm refinement, parameter optimization, and computational efficiency enhancements.

8. Conclusion

Through the completion of this project, I have deepened my understanding of signal processing techniques, including encoding and decoding English characters using specific frequency components. The project required the development of a graphical user interface (GUI) to facilitate user interaction, demonstrating my skills in UI design. Implementation involved applying Fourier transform for frequency analysis and designing bandpass filters for decoding. The integration of encoding and decoding systems within a software application highlighted my ability to merge different components seamlessly. Handling audio files in the .wav format and developing testing methodologies for accuracy evaluation further expanded my skill set. Overall, this project has showcased my problem-solving skills, algorithmic understanding, proficiency in coding, and the practical application of signal processing in a real-world context.

9. References

[1] https://stackoverflow.com/questions/12093594/how-to-implement-band-pass-butterworth-filter-with-scipy-signal-butter

[2] https://pythonnumericalmethods.berkeley.edu/notebooks/chapter24.02-Discrete-Fourier-Transform.html

[3] https://www.w3schools.com/python/numpy/numpy_intro.asp