English alphabets encoder and decoder

Submitted to: Dr. Abualseoud Hanani
Department of Electrical and Computer Engineering
Birzeit University

Group members:
Mohammad Saeed 1182500
Sondos Shahin 1200166
Shahd Qatri 1200431

Abstract

This project aims to design, implement and test an English alphabet character voice-frequency encoder, by representing each character with a combination of three voice-band frequency components. In addition to decoders that recovers the text from an encoded multi-frequency signal, by using the signal processing methods: Fourier transform and Filters.

1. Introduction

Digital signal processing (DSP) is the process of taking the real-word signals such as voice, audio, video, temperature…etc., that have been digitalized, and perform various operations on it such as "add", "subtract", "multiply" and many others, in high speed.
Digitalizing a signal include applying the signal to an analog to digital converter (A/D), so it can be represented by 0's and 1's, and therefore the computers can process it. Here comes the work of the DSP by taking the digitalized signal and processing it. Finally, the signal will be applied to a digital to analog converter (D/A) so it can be usable by the world. [1]
DSP also refers to the various techniques used for improving the accuracy and reliability of digital communications. It involves multiple mathematical operations, filtering, compression and many more. [2]
Digital signal processing with Python programming enables the use of Python's powerful libraries to analyze and process digital signals. One of the applications of using DSP with python is analyzing and processing audio signals, both can be used together to enhance the quality of audio recordings, reduce noise and reduce the audio files' sizes. [2]
In this project, we combined our knowledge of the digital signal processing with our skills in Python programming, in order to design and implement an encoder that takes a text string as input, and convert each character from the text to its corresponding frequencies; each character is represented by three different frequencies: high, middle and low. Then, the frequencies are transformed into an audio signal (.wav). For the other part of the project, digital signal processing techniques like: Fourier transform and narrowband filter were used to recover the encoded text in the first part from the audio file (.wav).

2. Problem specification

The idea of this project is to design and implement relatively accurate systems to both encode and decode a signal using digital signal processing methods.
The first phase aims to build a system that reads a sentence containing one or many words from the user, and encode each character into its corresponding signal. Each character is represented as a combination of three different frequencies: high, middle and low. After encoding, all signals are then combined as one signal and saved as an audio file (.wav).

The second phase of the project aims to build two systems, each one will decode the encoded signal saved as an audio file, and retrieve the original entered sentence in phase one. Each system will perform the decoding by using one of the digital signal processing techniques. The first system will decode using Fourier transform, and the other system will use narrowband filters.

3. Data

For the encoding system, the input is a string of one or many words. The characters in each word are the English alphabets in small letter, in addition to the space special character. The output of the encoder is an audio file in the format (.wav).

The two decoding system take an audio file (.wav) as an input, and their outputs are both a string of English small letter characters, in addition to the space special character.

4. Evaluation criteria

The evaluation of the work in this project is based on the accuracy of the results from the two decoding systems. The accuracy of our results was less than wanted at first. However, after having more understanding about the digital signal processing techniques, and learning more about the Python libraries and how to use them in implementing our encoder and decoders, we reached to a satisfyingly accurate results from our systems.

5. Approach

This project was built as three separate systems. Each one of them is different in the functionality and simplicity.
The first system is an English alphabet encoder, that reads a string from the user and transform it into an audio file. The system reads every character from the input string separately, and encode the character as a summation of three sinusoidal signals having different frequencies: high, middle and low. Every character in the input sting is encoded to its unique frequencies, and at the end, all the encoded characters are concatenated together to form one full signal that represents the whole string. This signal is then transformed into an audio signal (.wav). The simpleaudio and wave Python libraries were used to play and save the audio file.
Both the second and the third systems are decoders for the first system. Both systems take an encoded audio file as an input, and both systems should recover the encoded multi-frequency string entered for the first system. The difference between these two systems is the digital signal processing technique used.
The second system uses Fourier transform method. Since the audio signal represents the whole string, a loop was used to iterate through the samples corresponding for each character to analyze it alone, and by this dealing with each chunk of the signal alone. Then the Fourier transform was computed for each

chunk, and the corresponding frequencies were calculated, fft and fft.fftfreq functions from NumPy library were used for the calculations in this system. After finding all the frequencies for a chunk, the top three frequencies with highest amplitudes were found. Then, the closest match between the extracted frequencies and our predefined encoding frequencies were found, and by that the character corresponding to them was found. These steps were repeated for every character in the string.
The third system uses the bandpass filter technique. The spicy library was imported to make use of its functions in implementing the bandpass filter. A bandpass filter was designed based on the predefined frequencies for each character, and the signal was applied to these filters to attenuate all other unnecessary frequencies that may have resulted from errors or noise. Then the Fourier transform was computed for the filtered signal, and the corresponding frequencies and the highest amplitudes were found the same way as in the second system.

6. Result and analysis

Each system in the project was tested separately. Both decoders in our project resulted in 100% accuracy, but it was noticed that the decoder implemented using bandpass filters was slower than the decoder implemented using the Fourier transform alone. Each decoder was tested by decoding the encoded multi-frequency signal that resulted our encoding system, in addition to external test audio files that was generated from different encoding systems implemented by our colleagues. In both cases, the decoders still resulted in the same high accuracy output.

Test Case:

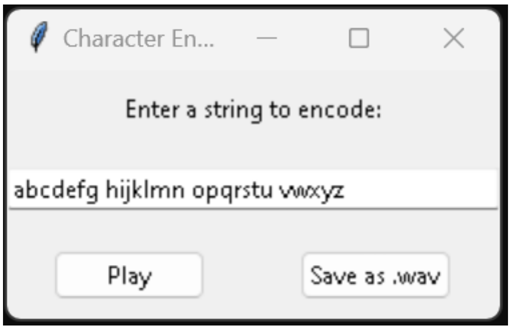

Figure 1: Encoding

Figure 2: Decoding using FT

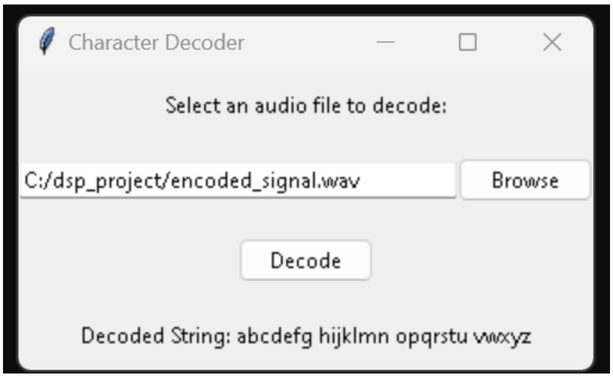

Figure 3: Decoding using bandpass filter

7. Conclusion

Working on this project (English alphabet voice-frequency encoder and decoder) have included working on many basic concepts of the digital signal processing (DSP) techniques. It resulted in better understanding of these concepts, which included discrete Fourier transform and bandpass filters. In addition to the new knowledge about the Python libraries and how to use them in implementing such projects.
This project has left us satisfied with our results, in addition to our new skills and knowledge about DSP.

8. References

[1] https://www.analog.com/en/design-center/landing-pages/001/beginners-guide-to-dsp.html
[2] https://www.techtarget.com/whatis/definition/digital-signal-processing-DSP