CIE 442 Course Project Report (Part II)

Rawan Eldalil
s-rawaneldalil@zewailcity.edu.eg

January 4, 2021

Introduction

In the 2^{nd} part of the course, more emphasis is on Spectrum Analyzers; they are devices which are used to view the frequency-domain characteristics of a given signal. For that purpose, any issues with the signal should be properly addressed to get more accurate information about the frequency content of the signal. Accordingly, different mathematical techniques are applied to signal to correct these issues; take the windowing technique as an example for eliminating/reducing the DFT leakage effect. Windowing is when one applies a window function to a signal. A window function is the mathematical function that is used to attenuate amplitudes outside the some chosen interval, while being symmetric around the middle of this interval. The mathematical expression of windowing a time-domain signal $x[n]$ is

$$x_{win}[n] = w[n]x[n]$$

for $0 < n < N-1$, where N is the length of the window. This part of the project is about building a spectral analyzer GUI-based program. Using Python with the aid of Tkinter GUI tool, one has created a spectrum analyzer that takes in audio signal, plot its the frequency- and time-domain representations, allow the user to choose different window functions with different lengths to examine their effect , calculate the average power of a portion of the signal and the RMS average of the signal and finally view the frequency-domain signal within two given frequencies.

The Spectrum Analyzer Program

Once the the Program Layout Cell is run, an 800*800 window pops up, and the user is asked to upload the audio signal by clicking the 'UPLOAD .WAV' button. Once the audio file is uploaded, the user can view the frequency- and time-domain plot of the audio signal by clicking on the button shown in **Figure 1**.

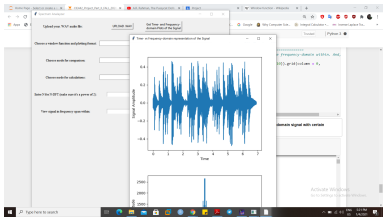

Figure 1: The frequency- and time-domain plots of the audio signal

The user is then allowed to choose a window function out of four options which are rectangular, triangular, Hann and hamming windows. Once the user chooses the window function, he/she is allowed to choose the length of the window. once the length is chosen, another window pops up displaying the frequency- and time-domain plots of the windowed signal, as shown in **Figure 2**.

Comparison Modes

The program allows the user to compare between the previously chosen window and either a different window function with the same length as before, as shown in **Figure 3**, or the same window with different length.

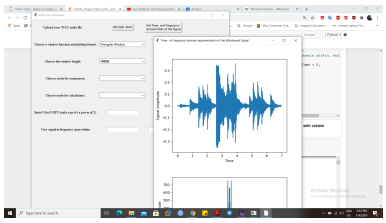

Figure 2: The frequency- and time-domain plots of the windowed audio signal

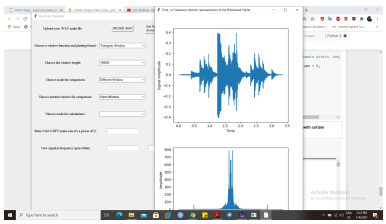

Figure 3: The frequency- and time-domain plots of the windowed audio signal in 'Different Window' Comparison Mode

Calculation Modes

The program allows the user to choose between two calculation option; either to get the average power of a potion of the frequency-domain signal by entering two frequencies, the start and end frequency shown in **Figure 4**, or get the RMS averaging of the frequency-domain signal as shown in **Figure 5**.

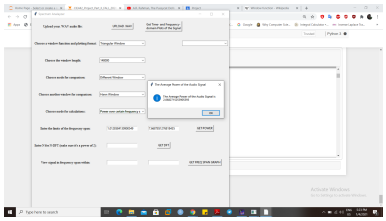

Figure 4: The Power Average Calculation Mode

N-DFT Choice for Frequency Resolution Examination

To examine the effect of the length of the signal on the frequency resolution in the frequency domain, the user is allowed to enter the a power-of-two N with maximum value 64,000 samples to apply the DFT on. Once the user enters the N and clicks the 'GET DFT' button, a frequency-domain plot pops up of the N-portion of the signal as shown in **Figure 6**.

Viewing the Signal within a Given Frequency Span

If the user wishes to view the signal within a given interval in the frequency domain, the program allows the user to enter the start and end frequencies of the to-be-viewed portion and clicking 'GET FREQ SPAN GRAPH' button as shown in **Figure 7** to get the plot of the desired portion.

Important Notes

When the user is asked to enter the frequencies in the program they must be equal to the frequencies generated while the program is running. So, for the to-be-spectrally-analyzed audio signal should be uploaded and button next to it is clicked, then close the program, and run cell like in **Figure 8**, and finally re-run the program.

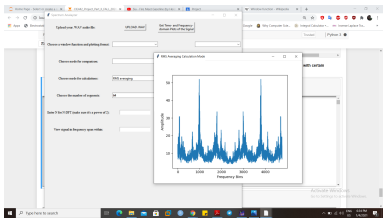

Figure 5: The RMS Averaging Calculation Mode

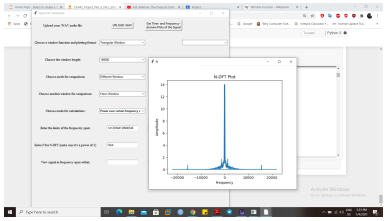

Figure 6: N-DFT Choice

Conclusion

Using Python with the aid of Tkinter GUI tool, one has created a spectrum analyzer that takes in audio signal, plot its the frequency- and time-domain representations, allow the user to choose different window functions with different lengths to examine their effect, calculate the average power of a portion of the signal and the RMS average of the signal and finally view the frequency-domain signal within two given frequencies.

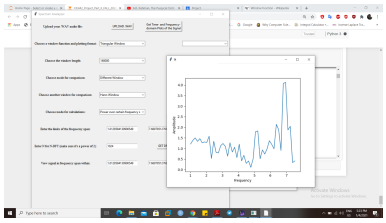

Figure 7: Viewing the Signal within a Given Frequency Span

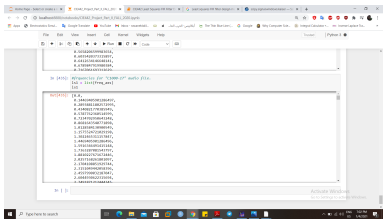

Figure 8: Frequencies Cell