AUDIO SPECTRUM ANALYSER USING PYTHON

B.Tech.2020.19CCE204 (Signal Processing)

TERM PROJECT REPORT

COMPUTER & COMMUNICATION ENGGINEERING – SEMESTER 3

Abstract :

This project focuses on developing an **Audio spectrum analyser** using Python taking the user's microphone as input. The spectrum analyser measures the amplitude, magnitude of the input signal. With respect to the frequency. It is mainly used to analyse the amplitude of different frequencies of signals. The analyser is able to sample the incoming spectrum in the time domain and convert the sampled information into frequency domain.

In this project the algorithm that we did will be used to convert the time domain signal to frequency domain called **Fast Fourier Transform (FFT)**. Several libraries such as NumPy, PyAudio, Struct, Matplotlib are used to generate the fft for the Python spectrum analyser, and to build the audio spectrum analyser we used Python (a multipurpose programming language) and this works out of the box with the microphone in windows.

System Description :

At first to develop an audio spectrum analyser, where we sample the audio from the microphone and display it in the time domain. We will be needing the following Python Libraries.

SOFTWARE USED : Visual Studio Code (IDe for Python 3.8) / Sypder Ide.

HARDWARE USED : User's System Microphone.

LIBRARIES INSTALLED FOR PYTHON :

Numpy

Numpy is a Python Library that adds support for large, multi-dimentional arrays and matrices.

PyAudio

PyAudio is another Python Library that can be easily used to play and record audio with Python on a variety of platforms.

MatPlotLib

Matplotlib is a library for the Python and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

VARIALBES : Then, we initialize some variables.

CHUNK is the number of samples that will be processed and displayed at any instance of time.

FORMAT is the data type that the PyAudio library will output.

CHANNELS is the number of channels our microphone has.

RATE is the sampling rate. We will choose the most common one, that is 44100Hz or 44.1KHz.

We will be using the **(FFT) Fast Fourier Transform** algorithm to calculate the frequency for the spectrum analyser. Now, FFT is an algorithm that computes the Discrete Fourier Transform.

In short, the DFT of a sequence of signal to represent it in frequency domain. There are many different FFT algorithms based on a wide range of published theories, from simple complex-number arithmetic to group theory and number theory. The one we will be using is the python fft algorithm from the library Numpy.

Now we will glance the methodology for this Audio Spectrum Analyser.

Methodology :

1. We create a PyAudio object from class PyAudio and store it in a variable ***p***.
2. Then, we use the ***open*** method on the object ***p*** and pass on the variables we initialized as parameters.
3. Then, we will initialise the plot with random values and set the limits for each of the two axes. (Two subplots)
4. Finally, we create an infinite loop and read the data from the microphone, convert the data into 16-bit numbers and further plot it using the matplotlib.pyplot function.
5. But, To convert Time domain to Frequency domain we convert that binary data into 16-bit integers and displayed them on a Plot using the MatPlotLib's PyPlot.
6. Next, The plotting function, we add the values from ***0*** uptill ***44100*** with number of parts equal to the size of ***CHUNK***. And the next line sets the ***X*** axis as semilog since the frequency representation is always done on semilog plots.

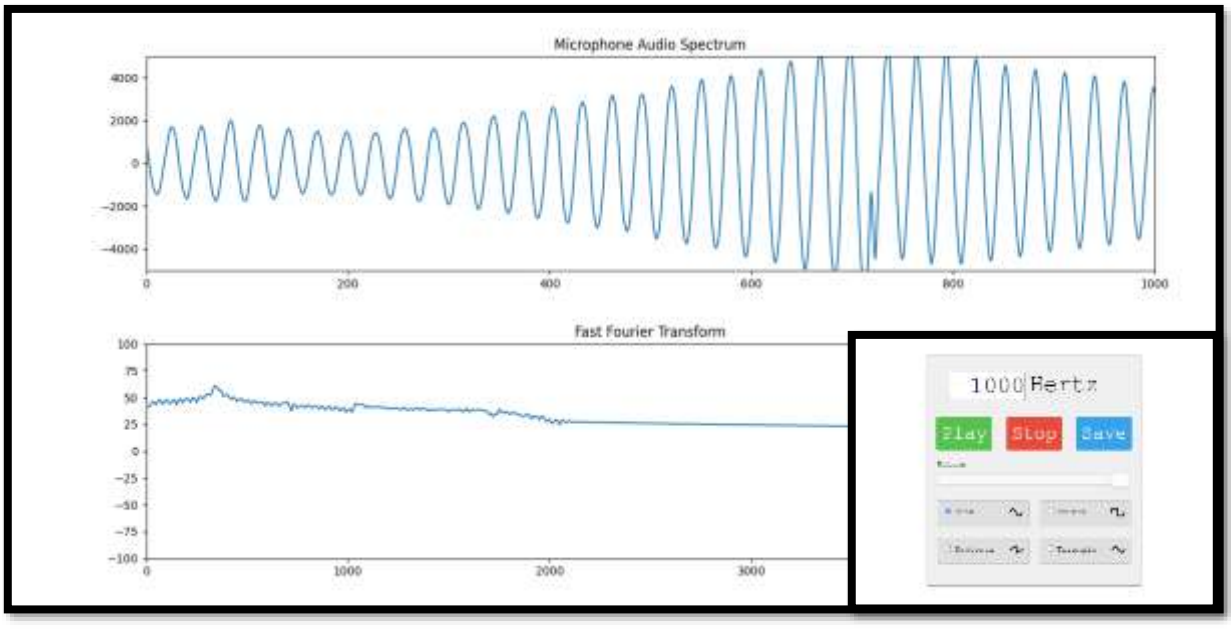
7. Since the output of the FFT computations are going to range from 0 to 1. We change the ***Y*** limits of the frequency plot. And also, since the computation produces a mirror of the spectrum after half the sampling rate. We don't need the part after half the sampling rate, that is after 22050. And hence we change the ***X*** limit also.
8. And then in the infinite loop, add the following lines to compute the FFT and plot it.

   Since the computed FFT contains both the real and the imaginary part. We take the absolute value of the returned FFT frequency in the Spectrum, multiply it by 2, and then divide it by 33000 times **CHUNK** to produce a plot with ***Y*** values ranging from 0 to 1.
9. Finally, RUN the code for the flawless Simulation.

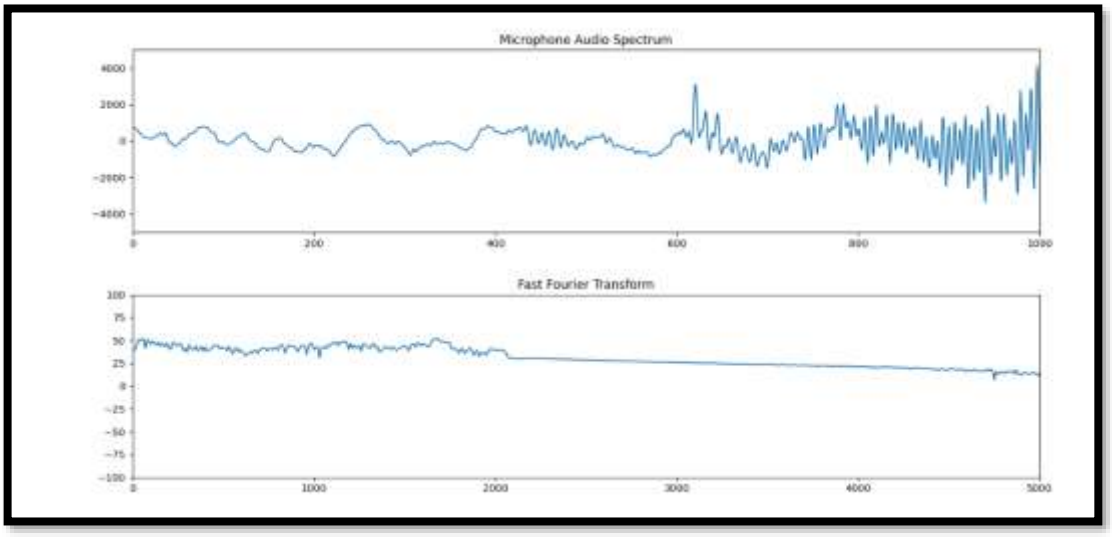
Simulation Results :

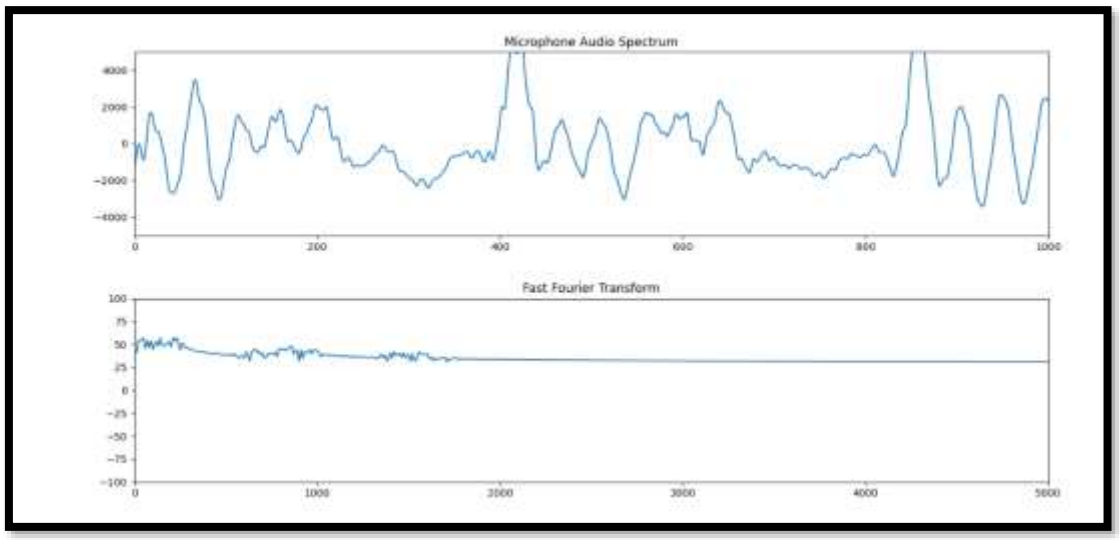
When we finally execute the code, a separate (figure 1) window opens to display the **Audio spectrum** and the **Fast Fourier transform spectrum** under two subplots with flawless animation.

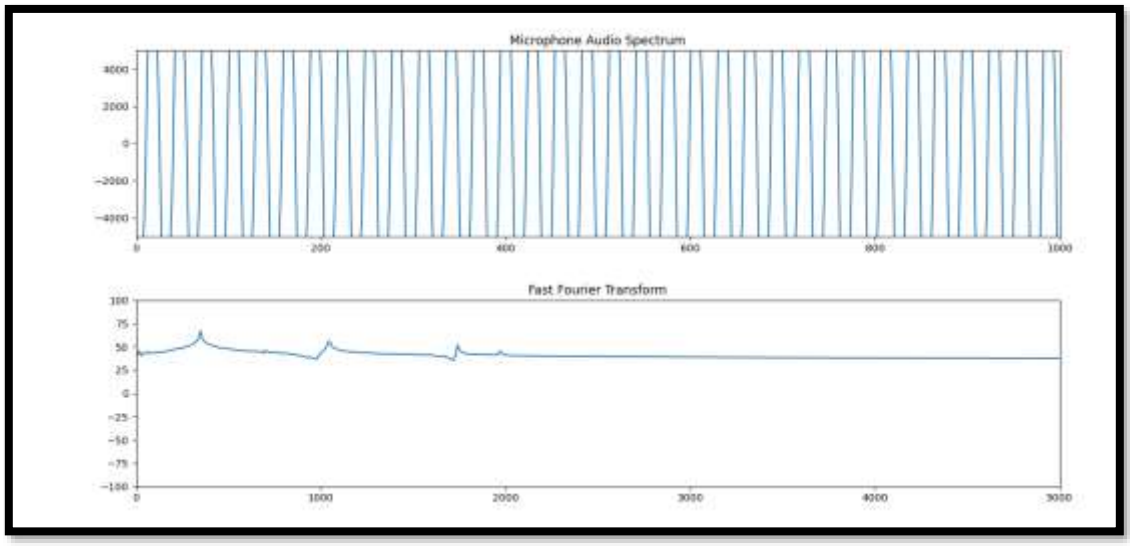
1. We use an online tone generator to generate sin and square waveforms and input those raw audio signal to the laptop's microphone and the following simulation is displayed.



2. Second simulation is Normal Human Speech input to the laptop's microphone and the following simulation is displayed.







Conclusion :

We conclude our project by building a **Real time audio spectrum analyser** which is used in various applications such as DJ spectrum analyzers and visualization display, RF recording and playback systems , EMC compliance testing and troubleshooting, spectrum monitoring. A Realtime spectrum analyser doesn't have any build time or lag. The analyser is able to sample the incoming spectrum in the time domain and convert the sampled information into frequency domain. And with the great usage of the algorithm of FFT, **Fast Fourier Transform** to calculate spectrum for the spectrum analyser, that computes the Discrete Fourier Transform. In short, the DFT of a sequence of signal to represent it in frequency domain.

Reference :

FFT: Fun with Fourier Transforms, By Tony DiCola, Analysing audio signals with Teensy 3.0.

AIM Audio spectrum analyser, https://analyticsindiamag.com/step-by-step-guide-to-audio-visualization-in-python/

https://medium.com/quick-code/python-audio-spectrum-analyser-6a3c54ad950

GitHub https://github.com/hnhaefliger.

Thank You 😊