



SEEK WISDOM, ELEVATE YOUR INTELLECT AND SERVE HUMANITY !



ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Course name: Digital Signal Processing
Course Code: ECEG-3171

Project Title: Audio Spectrum Analyzer and Visualization

Submitted To: Rediet M.

Submitted By:

Group Members (Section 4A):

<u>Name</u>	<u>ID</u>
1. Solomon Asregdew	UGR/4372/12
2. Yosef Alemu	UGR/6607/12
3. Misgan Moges	UGR/5779/12
4. Nigist Dessalegn	UGR/5223/12
5. Seble Kassa	UGR/7817/12

Submitting Date: Friday, June 24, 2022

Table of Contents

	Page
Introduction	1
Problem Statement	2
Objective	3
Methodology	4
Simulation and Discussion	7
Conclusion	9
Reference	10

Introduction

An audio spectrum analyzer is a tool used to visualize and analyze the frequency components of an audio signal. It is often used by audio engineers and musicians to identify and adjust specific frequencies in a sound. In this project, we will cover the basics frequency analyzer and functionality of an audio spectrum analyzer.

What are the components of an audio spectrum analyzer?

There are different components in audio spectrum in the process of analyzing and representing audio signal. Some of the basic components are discussed hereunder.

- ✓ **Input device:** The input device captures the audio signal and sends it to the analyzer for processing.
- ✓ **ADC (analog to digital converter):** The ADC converts the analog audio signal into a digital signal that the analyzer can process.
- ✓ **Signal processing:** The signal processing component analyzes the digital audio signal to determine the frequency components of the audio.
- ✓ **FFT (Fast Fourier Transform):** The FFT is a mathematical algorithm used to transform the audio signal from the time domain to the frequency domain.
- ✓ **Visualization:** The visualization component displays the frequency components of the audio signal in a graphical format.

What are the types of visualizations?

- ✓ **Waveform:** A waveform visualization displays the amplitude of the audio signal over time. It is useful for visualizing the overall loudness of a sound and identifying specific peaks or troughs in the signal.
- ✓ **Spectrum:** A spectrum visualization displays the frequency components of the audio signal. It is useful for identifying specific frequencies in the sound and adjusting them as needed.
- ✓ **Spectrogram:** A spectrogram visualization displays the frequency components of the audio signal over time. It is useful for identifying changes in the frequency components of the sound over time, such as a sudden increase in volume or the appearance of a new frequency component.
- ✓ **Waterfall:** A waterfall visualization displays the frequency components of the audio signal over time in a three-dimensional format. It is useful for identifying changes in the frequency components of the sound over time and seeing how they interact with each other.

But in this project we only used the waveform and spectrum visualizations.

Problem Statement

Design an audio spectrum analyzer and visualization system that can take an audio input and provide a real-time display of the spectrum of the audio signal. The system should be able to analyze and visualize in both time and frequency domains, and provide a graphical representation of the audio signal's spectral content.

Audio spectrum analyzers and visualizations are commonly used to solve various problems related to audio signal processing, including:

Sound engineering: audio spectrum analyzer are used to analyze and visualize the frequency content of audio signals, which is critical in sound engineering for tasks such as equalization and mixing.

Noise reduction: audio spectrum analyzers can help identify the specific frequencies of unwanted noise in an audio signal, allowing for targeted noise reduction and removal.

Music production: audio spectrum analyzers can be used to analyze the frequency content of musical instruments and vocals, allowing for better mixing mastering of tracks.

Acoustic analysis: audio spectrum analyzers can be used to analyze the frequency response of acoustic spaces, such as concert halls or recording studios, to better understand their sound characteristics.

Speech recognition: audio spectrum analyzers can be used in speech recognition systems to extract features from speech signal, such as formant frequencies that can be used to recognize and transcribe spoken words.

Overall, audio spectrum analyzers and visualizations play an important role in many different fields that involve audio signal processing, allowing for better understanding and manipulation of sound and music.

Objective

The main objective of an audio spectrum analyzer and visualization is to provide a detailed analysis of the frequency components of an audio signal, in order to improve the quality of the sound.

Audio spectrum analyzers allow audio engineers and musicians to see the frequency content of an audio signal and identify specific frequencies that may need to be adjusted or removed. By breaking down an audio signal into its frequency components, an audio spectrum analyzer can help identify problem areas in the sound, such as unwanted noise or resonance, and allow for adjustments to be made to improve the quality of the sound.

Visualization is an important part of audio spectrum analysis, as it allows engineers and musicians to see the frequency content of the audio signal and identify specific frequencies that may need to be adjusted or removed. Different types of visualizations, such as spectrograms, waterfalls, and oscilloscopes, offer different views of the frequency content of the audio signal and can be used to identify different types of problems in the sound.

Overall, the main objective of an audio spectrum analyzer and visualization is to improve the quality of the sound by providing a detailed analysis of the frequency components of an audio signal and allowing for adjustments to be made to achieve the desired sound. By using these tools effectively, audio engineers and musicians can achieve the best possible sound for their projects, whether it's a live performance, a recording, or a film soundtrack.

Methodology

This project was done using Python programming language, which is one of the high level programming languages we have today. The methodology we followed in making this project is summarized below in step by step manner. The overall methodology followed is first to be able to get audio in through our microphone and into Python then process it and display it by the help of matplotlib.


I. Step – 1

We first tried to gather as much information we can collect as possible and made summarization on it. The standard sampling frequency is 44.1 kHz or 44,100 samples per second. 48 kHz is also used as audio sample rate commonly used for movies. We used the 44.1 kHz sampling frequency for our project. The commonly referenced frequency range of audio is 20 Hz to 20,000 Hz. So, our plot should consider ranges of values between 20 Hz to 20,000 Hz.


II. Step – 2

Next, we determined the types of modules which are necessary for writing the code in Python. The modules we used in this project are listed below.

1. Numpy: is a powerful package which is used for working with arrays, matrices and many more.

 See the documentation here: <https://numpy.org/doc/stable>


2. Pyaudio; is a non-built-in module in python which makes it easy for us to play and record audio on a variety of platforms.

 See the documentation here: <https://pypi.org/project/PyAudio>

3. Struct: struct is a built-in module. We used struct module to help us unpack the audio data into integers.

 See the documentation here: <https://docs.python.org/3/library/struct.html>

4. Matplotlib: this module is used to plot graphs and create interactive visualization in python. We used this module for plotting the audio waveform in time domain and for plotting the frequency spectrum as well.

 See the documentation here: <https://matplotlib.org>

III. Step – 3

After importing the necessary libraries, the remaining part of the code was written accordingly.

IV. Step – 4

Then finally, in order to assert that the code is right, we tested it by giving audio input with varying the frequency of the audio and observed the time domain plot and the frequency spectrum of the audio.

```
import struct
import pyaudio
import matplotlib.pyplot
import numpy as np

'''
Step - 2

Declaring variables
'''
# VARIABLES
CHUNK = 1024*2
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100 # sampling frequency in Hz
'''STEP - 3

Creating objects or instances
'''
# OBJECTS/INSTANCES
p = pyaudio.PyAudio()

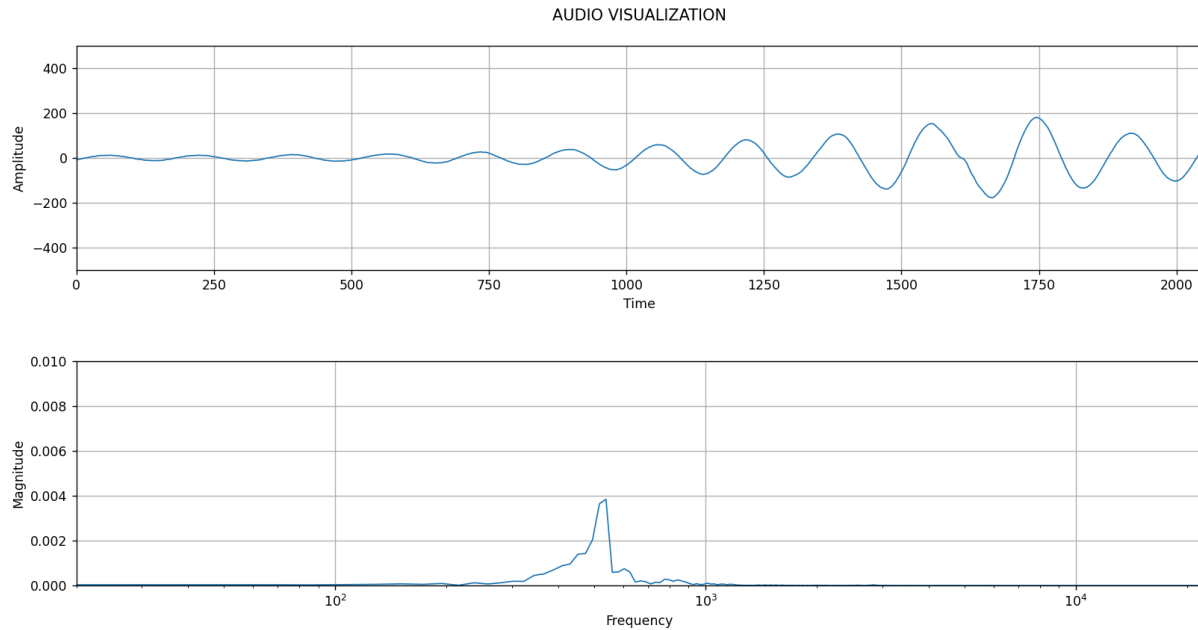
# opening stream
stream = p.open(
    format=FORMAT,
    channels=CHANNELS,
    rate=RATE,
    input=True,
    output=True,
    frames_per_buffer=CHUNK
)
'''
Step - 4
```

Finally, the plot can be made using matplotlib module for both the time domain and the frequency domain. but for the frequency domain plot, first the FFT has to be manipulated then it can be drawn.

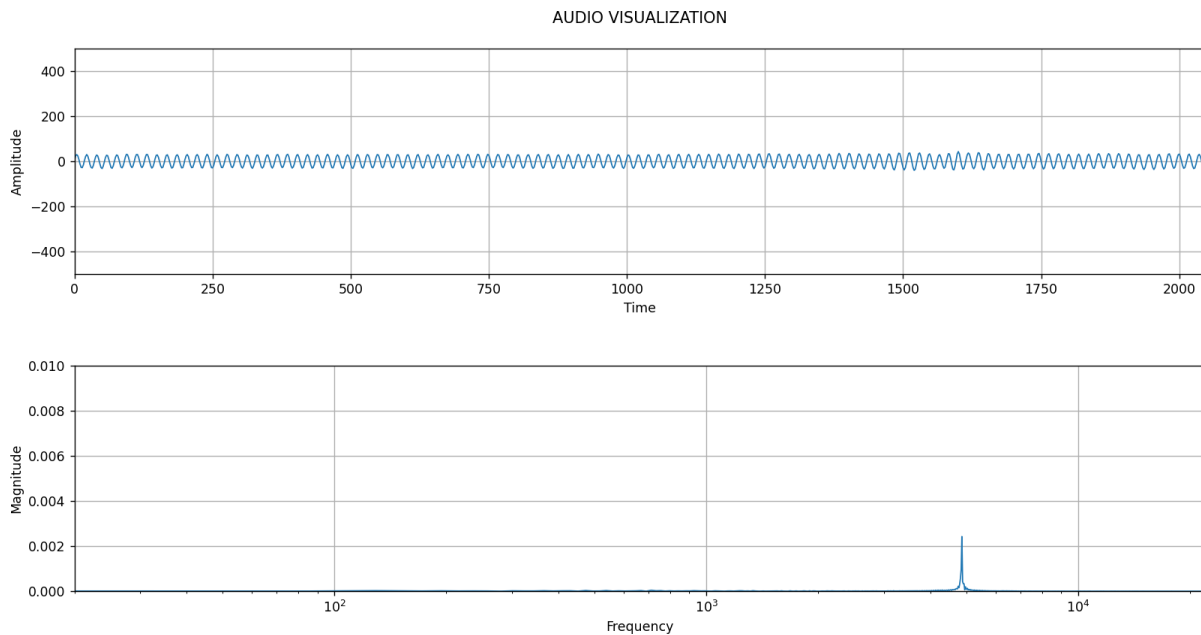
```
'''  
fig, (plot_one, plot_two) = matplotlib.pyplot.subplots(2, figsize=(5, 5))  
x = np.arange(0, 2*CHUNK, 2)  
  
#FFT computation  
x_fft = np.linspace(0, RATE, CHUNK)  
  
# line plots  
line, = plot_one.plot(x, np.random.rand(CHUNK), '-', lw=1)  
line_fft, = plot_two.semilogx(x_fft, (np.random.rand(CHUNK))), '-', lw=1)  
  
# time domain plot parameters' labels  
plot_one.set_title("AUDIO VISUALIZATION\n")  
plot_one.set_ylim(-500, 500)  
plot_one.set_xlim(0, CHUNK)  
plot_one.set_xlabel("Time")  
plot_one.set_ylabel("Amplitude")  
plot_one.grid()  
  
# frequency domain plot parameters' labels  
plot_two.set_xlim(20, RATE/2)  
plot_two.set_ylim(0, 0.01)  
plot_two.set_xlabel("Frequency")  
plot_two.set_ylabel("Magnitude")  
plot_two.grid()  
  
matplotlib.pyplot.tight_layout()  
fig.show()  
  
# loop for streaming audio  
while True:  
    data = stream.read(CHUNK)  
    dataInt = struct.unpack(str(CHUNK) + 'h', data)  
    line.set_ydata(dataInt)  
    line_fft.set_ydata(np.abs(np.fft.fft(dataInt))*2/(11000*CHUNK))  
    fig.canvas.draw()  
    fig.canvas.flush_events()
```


Simulation and Discussion

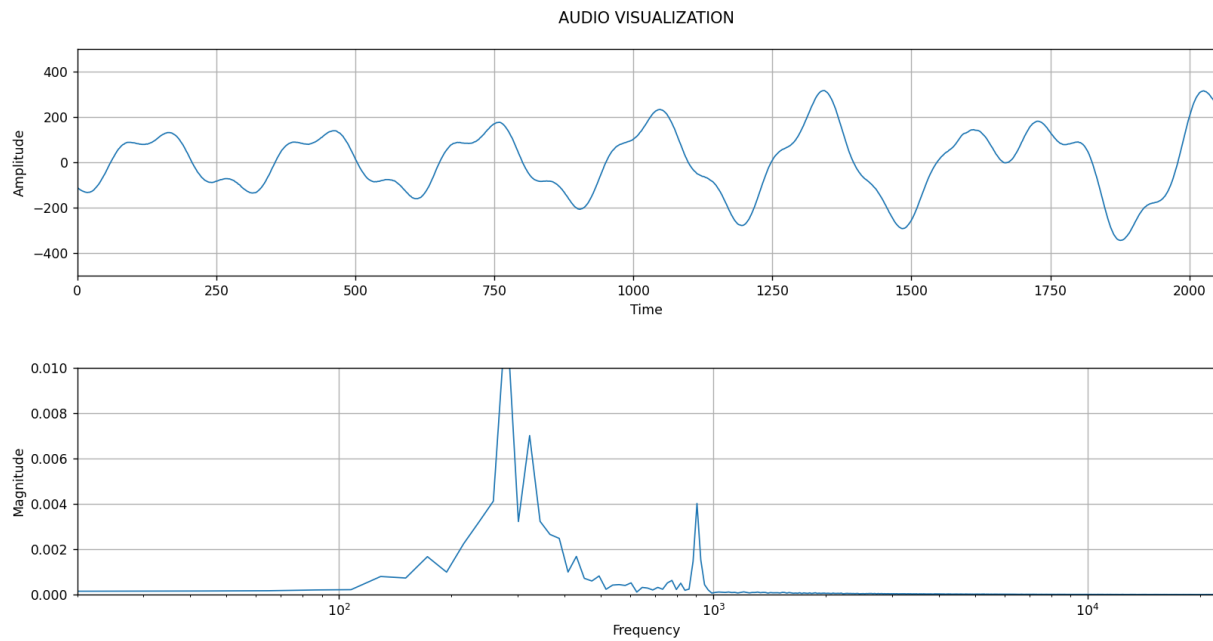
For low amplitude low frequency audio signals, the audio visualization in time domain and frequency spectrum is given in figure below.



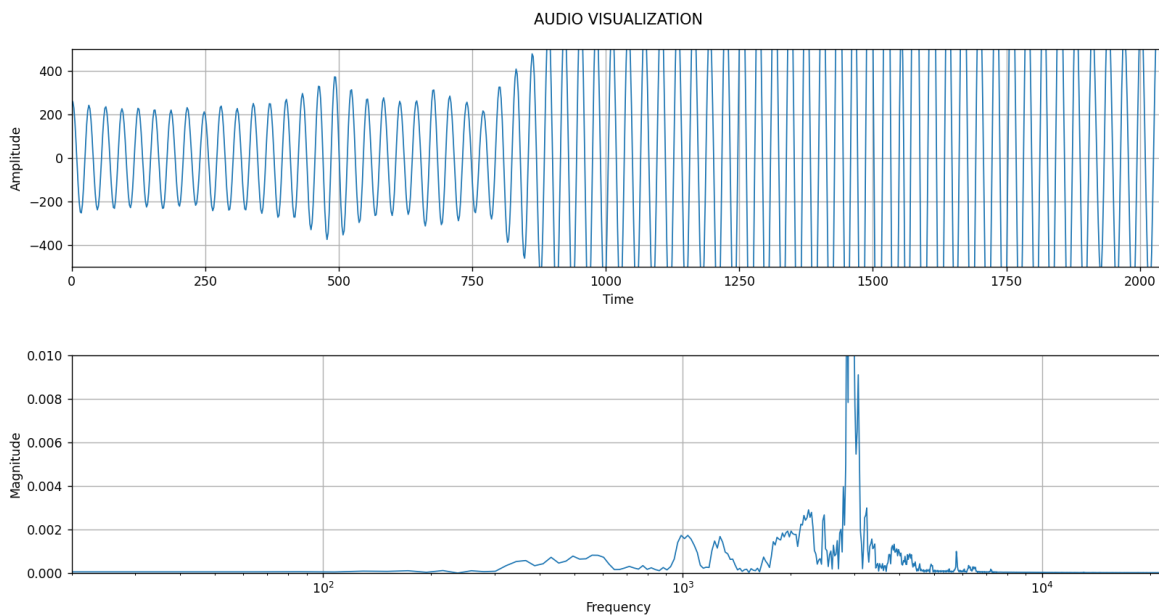
For low amplitude high frequency audio signals, the audio visualization in time domain and frequency spectrum is given in figure below.



For high amplitude low frequency audio signals, the audio visualization in time domain and frequency spectrum is given in figure below.



For high amplitude high frequency audio signals, the audio visualization in time domain and frequency spectrum is given in figure below.



Conclusion

An audio spectrum analyzer is a powerful tool for visualizing and analyzing the frequency components of an audio signal. With the ability to identify specific frequencies and adjust them as needed, it is an essential tool for audio engineers and musicians. By understanding the components and visualizations available, you can make the most of this powerful tool and achieve the best possible sound.

Python programming language can be used to make a simple audio analyzer and visualization tool. Python has many modules that can do so many amazing things that are so important in the contemporary science.

In summary, there are several methodologies for audio spectrum analysis and visualization, each with its own advantages and disadvantages. By understanding these methodologies and visualization techniques, you can choose the best approach for your specific needs and achieve the best possible sound.

Reference

Sites:

- www.python.org
- www.academo.org
- www.izotope.com

Books:

- Digital_Signal_Processing_-_Principles_Algorithms_Applications_3rd_Ed-[Proakis_Manolakis]
- K. Ingle and J. G. Proakis-Digital Signal Processing Using Matlab V.4
- [Steven_T._Karris,_Steven_Karris,_Steven_Karris]_S