A Mini Project Report on

‘Sound Signal Analysis’

Submitted By

Durgesh Kolte SEE021

Yash Pamnani SEE012

A Project report submitted as a partial fulfillment towards Practical of Applied Mathematics for semester III of S. Y. B. Tech. (Electronics Engineering)

2018-19

Under the guidance of

Usha Verma

School of Electrical Engineering

MIT Academy of Engineering, Alandi (D), Pune-412 105

C E R T I F I C A T E

*This is to certify that **Durgesh Kolte (S186003) and Yash Pamnani (S186012)** of MIT Academy of Engineering, Alandi (D), Pune have submitted MATLAB Project report on "**Sound Signal Analysis**" as a partial fulfillment of Semester-III S. Y. B. Tech. for completion of Applied Mathematics Practical work during the academic year 2018-19.*

Usha Verma
(Guide)

Mrs. Prabha Kasliwal
(Dean, SHES)

External Examiner

ACKNOWLEDGEMENT

I take this opportunity to record my profound gratitude and indebtedness to Usha Verma, Assistant Professor, School of Electrical Engineering for their inspiring guidance, valuable advices, constant encouragement and untiring supervision throughout my project work.

I express my deep sense of gratitude to Mrs. Prabha Kasliwal Dean, School of Humanities and Engineering Sciences, for her continuous inspiration and encouragement.

Finally, I would like to acknowledge and express my special thanks to my family, friends and classmates for their patience, encouragement, support they have made during the period of this work.

Durgesh Kolte (SEE021)

Yash Pamnani (SEE012)

INDEX

List of Figures

List of Tables

ABSTRACT

From an audio sample retrieved from a concert, we aim to identify different instruments and human voice from the audio clip. We know that different instrument or voice has different range of frequencies which differentiates it from other sound. With the use of MATLAB and transform methods we need to achieve our purpose.

CHAPTER 1: INTRODUCTION

1.1 Problem Definition

From 2-minute audio sample retrieved from a concert, we aim to identify different instruments and human voice from the audio clip. We know that different instrument or voice has different range of frequencies which differentiates it from other sound. With the use of MATLAB and transform methods we need to achieve our purpose.

We need to use transforms to convert amplitude v/s coefficient graph into amplitude v/s frequency graph. After obtaining this graph we need to look into amplitude v/s frequency graph to analyze the different frequencies. This sample would have been consisting of human voice along with instruments each with different frequencies. So we need to separate those. We know that different instruments have different frequency, so by looking into graph we can predict human voice and difference instruments present in the sample by looking into the peak of the amplitude with respect to frequency at that particular peak, this frequency is nothing but the required frequency of that instrument or voice.

For example, if the frequency of guitar is from 80 Hz to 1200 Hz. So, the peak of amplitude obtained on amplitude v/s frequency graph is observed in this range of frequency.

1.2 Theoretical Background

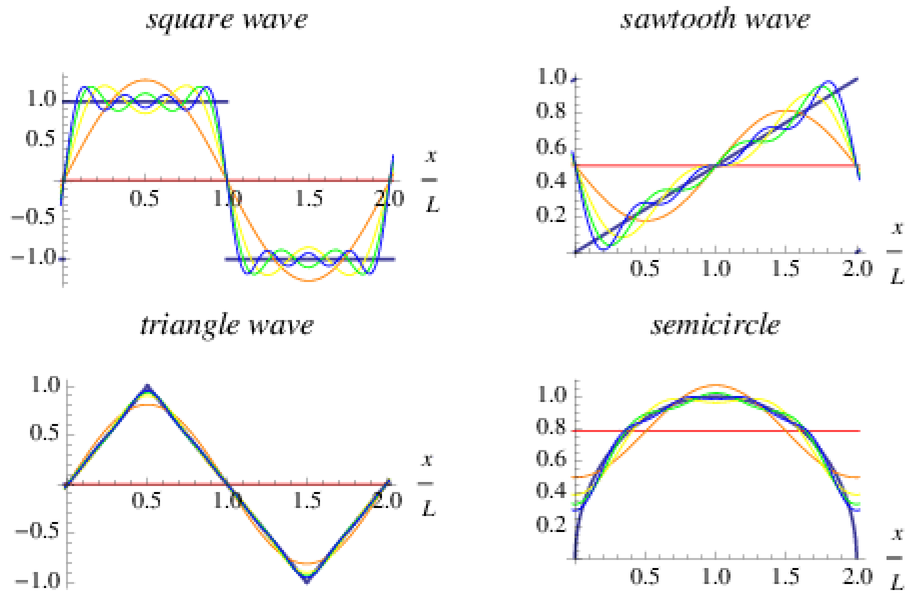

Fig No. 1 Different Waveforms

A **Fourier series** is an expansion of a periodic function f(x) in terms of an infinite sum of sines and cosines. Fourier series make use of the orthogonality relationships of the sine and cosine functions.

Y = fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm. [8]

Audio signal processing is a subfield of signal processing that is concerned with the electronic manipulation of audio signals. As audio signals may be represented in either digital or analog format, processing may occur in either domain. Analog processors operate directly on the electrical signal, while digital processors operate mathematically on the digital representation of that signal. [10]

CHAPTER 2: METHODOLOGY

The table shown below consists of frequencies of various musical instruments, the range of the frequency and the fundamental frequency.

Table No. 1 Frequencies of different instruments [11]

SOUND	FREQUENCY RANGE		FUNDAMENTAL FREQUENCY
Kick Drum	60	250	155
Toms	60	210	135
Snare	120	250	185
Cymbal/Hi-hat	3000	5000	4000
Electric Guitar	82	1397	739.50
Bass Guitar	41	262	151.50
Acoustic Guitar	82	1397	739.50
Mandolin	136	1320	728
Tenor Sax	104	659	381.50
Alto Sax	150	800	475
Harmonica Various	180	3100	1640
Vocal (Baritone)	87	349	218
Vocal (Tenor)	130	523	326.50
Vocal (Alto)	180	700	440
Vocal (Soprano)	250	1300	775
Violin	196	4186	2191
Viola	315	1175	745
Cello	65	988	526.50
Double Bass	41	247	144
Piccolo	523	3951	2237
Flute	250	2500	1375
Oboe	225	1500	862.50
Clarinet	165	1568	866.50
Accordion	180	1000	590
Bassoon	60	620	340
Trumpet	165	988	576.50
Trombone	60	500	280
French Horn	110	880	495
Tuba (Bass)	44	349	196.50
Harp	30	7000	3515
Harpsichord	40	1500	770
Piano	28	4186	2107
Pipe Organ	16	7040	3528
Keyboard / Synth	20	4000	2010
Female Voice	250	1000	625
Male Voice	100	800	450
Sub Bass	16	60	38
Concert Flute	262	1976	1119
Xylophone	700	3500	2100
Timpani (Drum)	90	180	135
Contra Bass	41	330	185.50
Bass	33	330	181.50
Baritone Sax	65	440	252.50
Soprano	262	1047	654.50
Mezzo Soprano	110	880	495
Contra Alto	175	698	436.50

The figure shown below is of random noise signal.

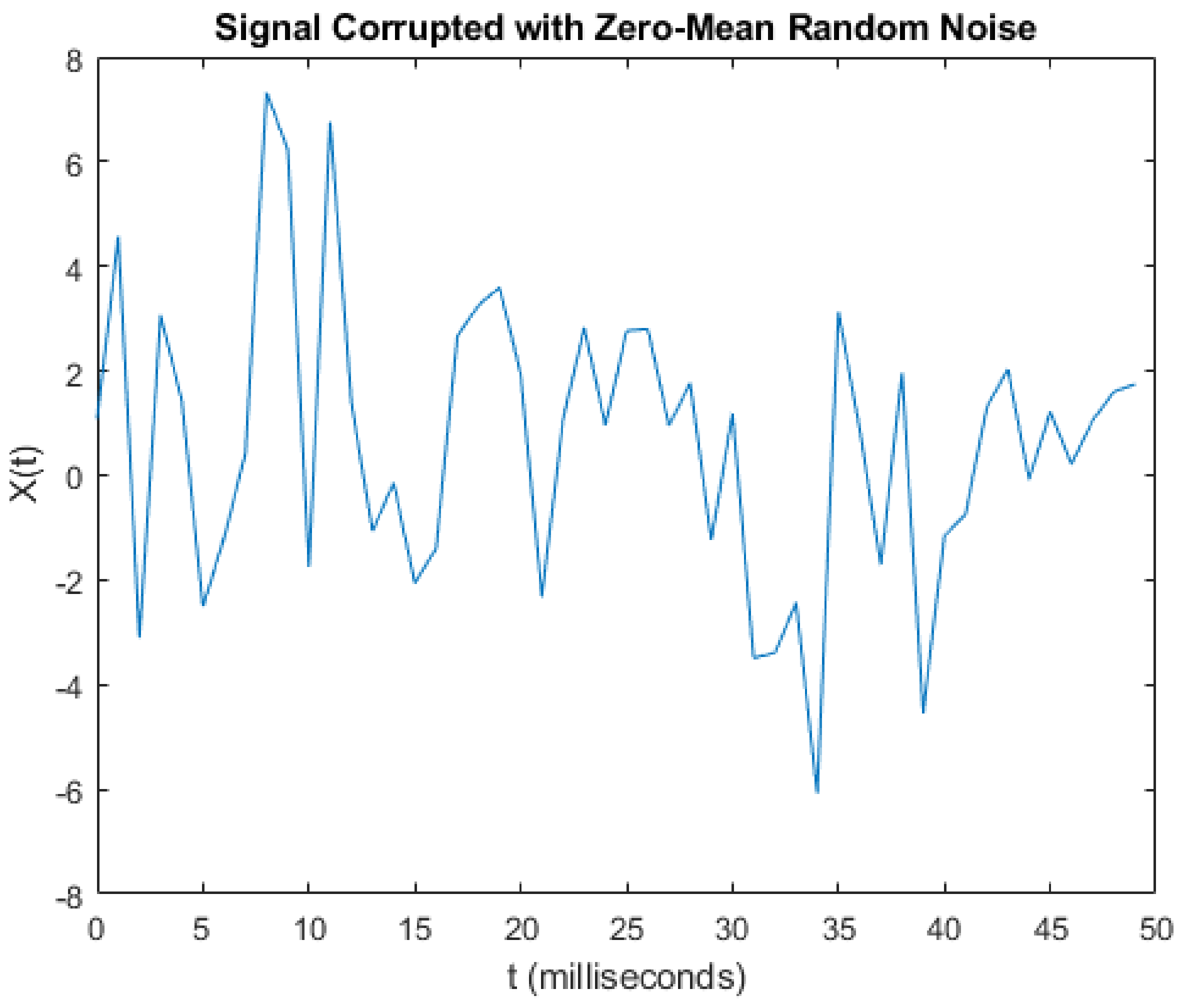

Figure No. 2 [12]

The following figure shown below is filtration of above noise signal. This is done by taking Fourier transform of the above function.

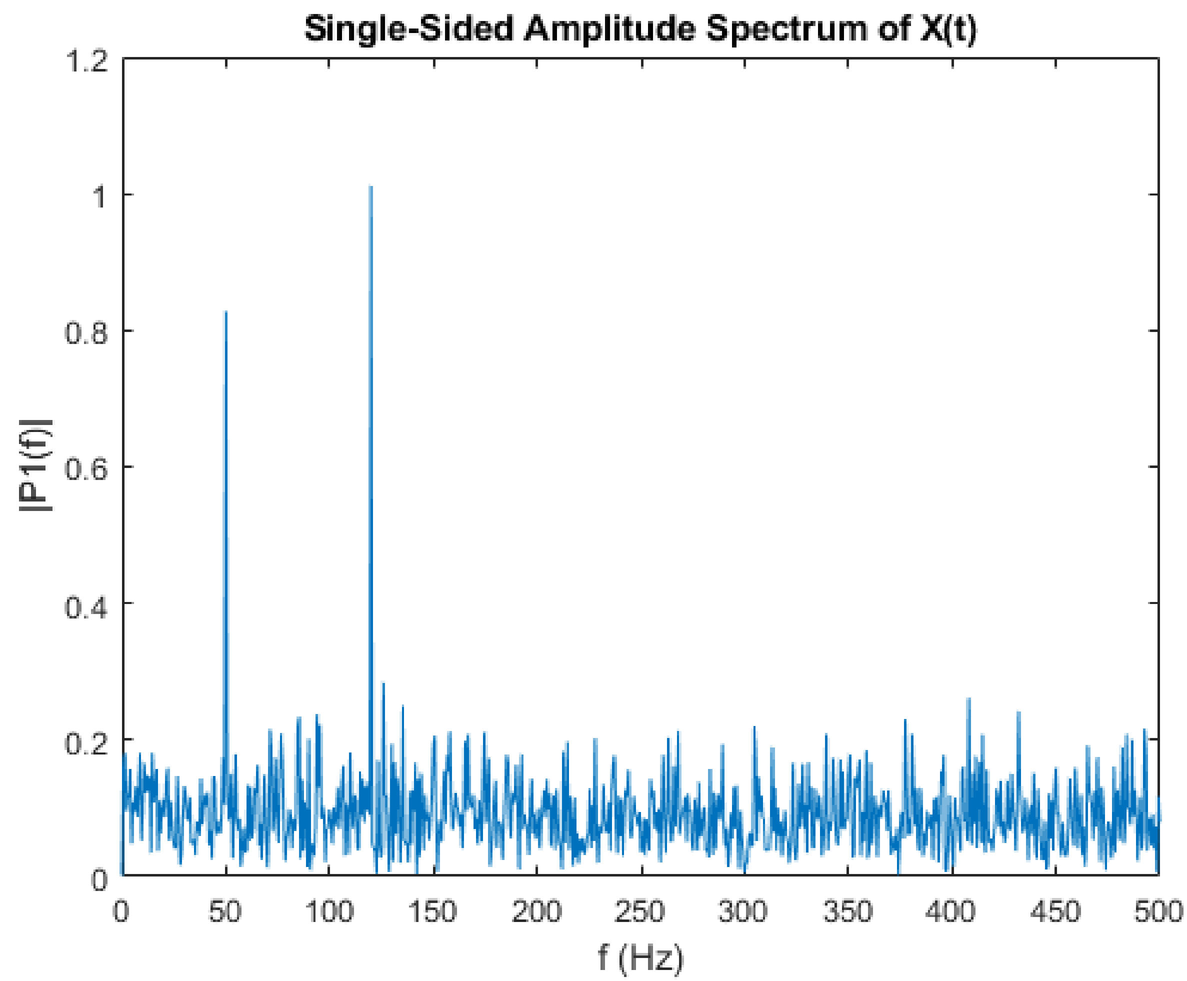

Figure No. 3

The following figure shown below is the zoomed view of the above figure.

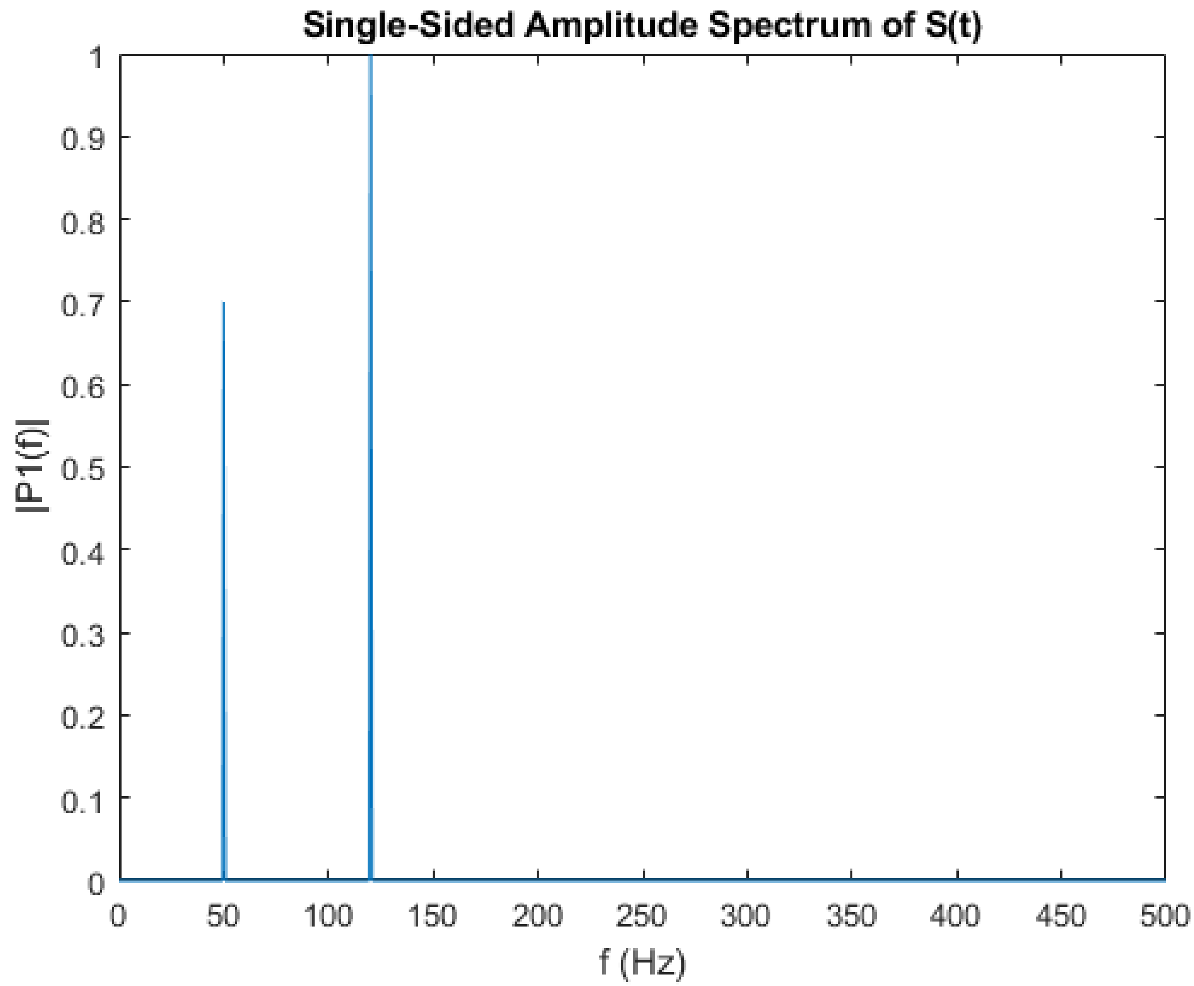

Figure No. 4

CHAPTER 3: ALGORITHM

1. Use Fourier transforms to find the frequency components of a signal buried in noise.
2. Specify the parameters of a signal with a sampling frequency of 1 kHz and a signal duration of 1.5 seconds.
3. Plot the noisy signal in the time domain. It is difficult to identify the frequency components by looking at the signal X(t).
4. Compute the Fourier transform of the signal.
5. Compute the two-sided spectrum P2. Then compute the single-sided spectrum P1 based on P2 and the even-valued signal length L.
6. Define the frequency domain f and plot the single-sided amplitude spectrum P1. The amplitudes are not exactly at 0.7 and 1, as expected, because of the added noise. On average, longer signals produce better frequency approximations.
7. Now, take the Fourier transform of the original, uncorrupted signal and retrieve the exact amplitudes, 0.7 and 1.0.

CHAPTER 4: IMPLEMENTATION IN MATLAB

The following code is implemented using MATLAB. It is the code of sound signal analysis using Fourier transform. We encountered with many errors shown below.

```
clc;close all;clear all;
[X,Fs]=audioread('guitar.wav');
T = 1/Fs;                       % Sampling period
L = length(X);                  % Length of signal

Y = fft(X);                     %taking fast Fourier transform

P2 = abs(Y/L);                  %transforming into frequency domain
P1 = P2(1:L/2+1);               %making graph single sided
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;

plot(f,P1)
title('Single-Sided Amplitude Spectrum of X(t)')
xlabel('f (Hz)')
ylabel('|P1(f)|')
```

ERRORS and COMMANDS:

Following are the errors which were faced while coding in MATLAB.

```
>> help audioread

 audioread Read audio files

    [Y, FS]=audioread(FILENAME) reads an audio file specified by the

    string FILE, returning the sampled data in Y and the sample rate

    FS, in Hertz.

    [Y, FS]=audioread(FILENAME, [START END]) returns only samples START

    through END from each channel in the file.

    [Y, FS]=audioread(FILENAME, DATATYPE) specifies the data type format of

    Y used to represent samples read from the file.

    If DATATYPE='double', Y contains double-precision normalized samples.

    If DATATYPE='native', Y contains samples in the native data type

    found in the file.  Interpretation of DATATYPE is case-insensitive and

    partial matching is supported.

    If omitted, DATATYPE='double'.

    [Y, FS]=audioread(FILENAME, [START END], DATATYPE);

    Output Data Ranges

    Y is returned as an m-by-n matrix, where m is the number of audio

    samples read and n is the number of audio channels in the file.

    If you do not specify DATATYPE, or dataType is 'double', then Y is of type double,
```

```
    and matrix elements are normalized values between -1.0 and 1.0.


    If DATATYPE is 'native', then Y may be one of several MATLAB

    data types, depending on the file format and the BitsPerSample

    of the input file:

    Call audioinfo to learn the BitsPerSample of the file.

    Note that where Y is single or double and the BitsPerSample is

    32 or 64, values in Y might exceed +1.0 or -1.0.

    See also audioinfo, audiowrite


>> Example

Error: File: Example.m Line: 1 Column: 23

Unexpected MATLAB operator

**file format extension missing

Problem finding getMapfileName in com.mathworks.mlwidgets.help.HelpUtils: null

**wrong location specified


>> Example

Error: File: Example.m Line: 1 Column: 30

Unexpected MATLAB expression.

**single quotes missing
```

CHAPTER 5: RESULTS

The following figure shown below is of amplitude v/s frequency. This is done by taking the Fourier transform of the sound sample.

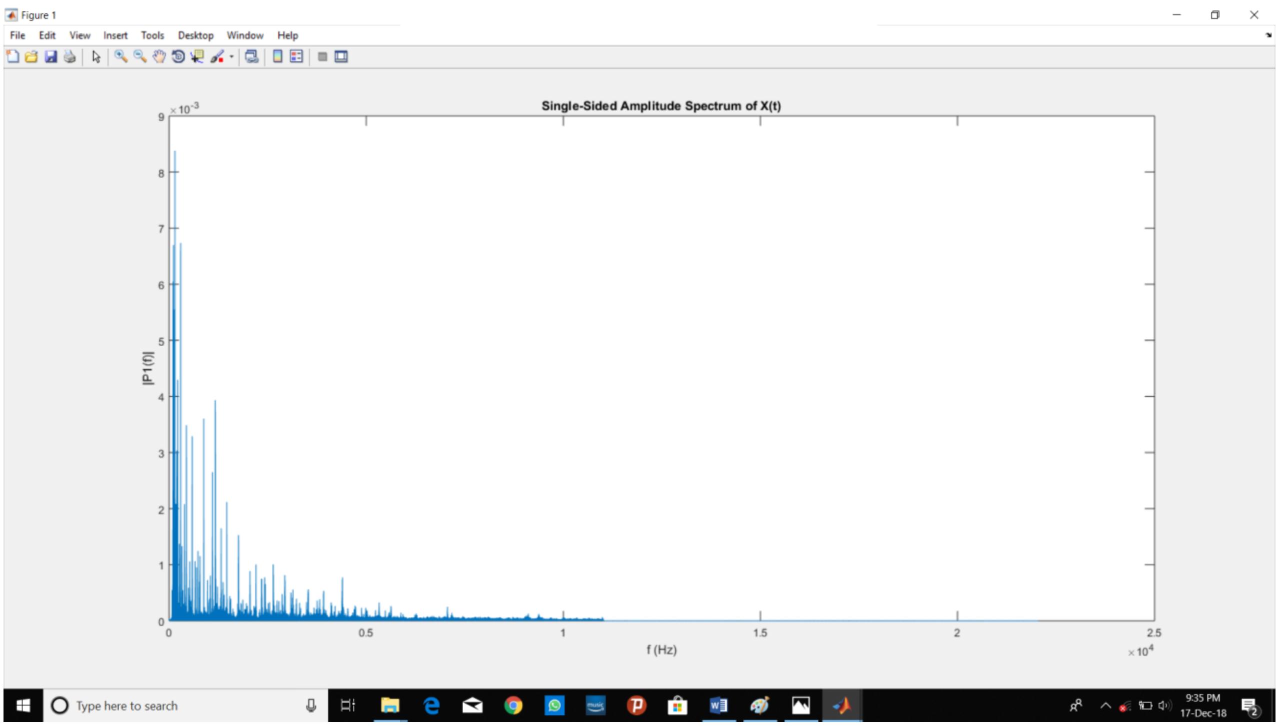

Figure No. 5: Amplitude v/s Frequency Graph

The following figure shown below is the zoomed view of above waveform. The highest peak and the corresponding frequency are obtained. This is the frequency which lies in the range shown in the table no.1.

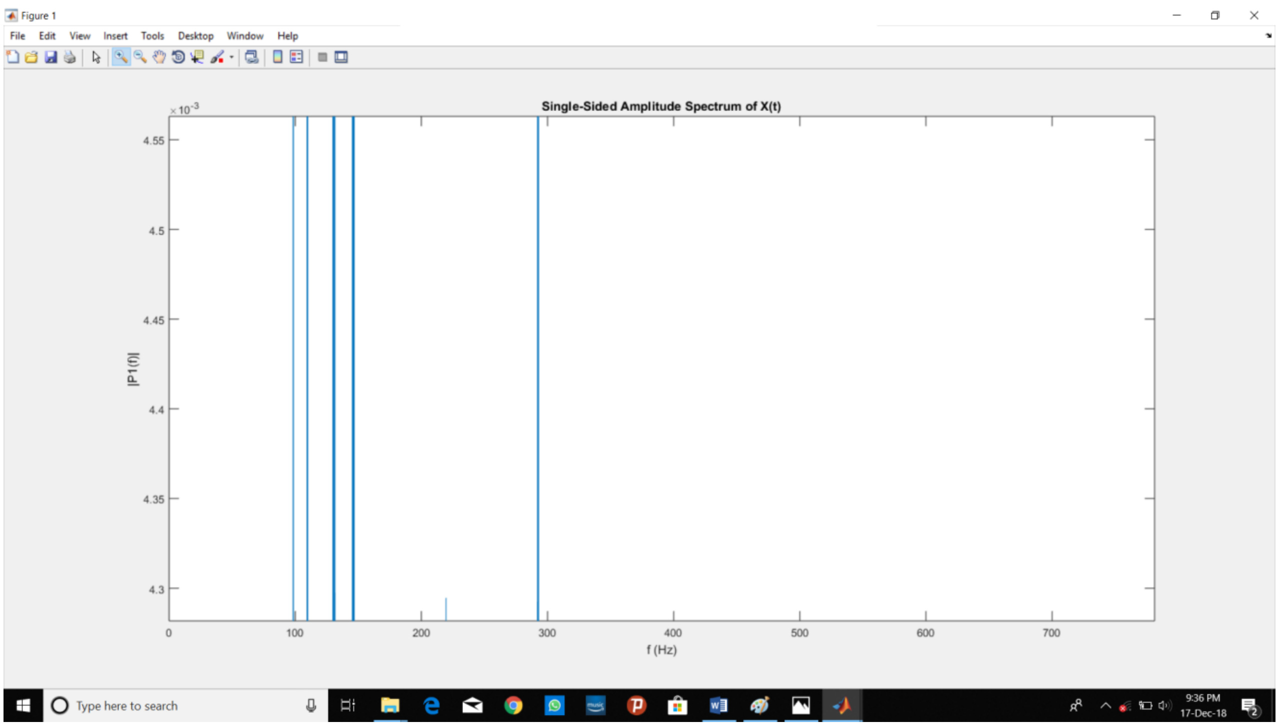

Figure No. 6: Zoom In part of the Amplitude v/s Frequency Graph

CHAPTER 6: CONCLUSION

Every audio consists of noise (unwanted sound) which needs to be rejected. So, it is very important that audio file is analyzed first for the range of frequencies and known noise frequency is rejected. This may enhance the sound quality. So, it can be concluded to reject this unwanted sound it is very important that first we analyze this audio file for various range of frequencies and manipulate this according to our requirement on frequency domain. So, we simply compare the frequency values that we got from the graph to the Table No. 1 and estimate that which Instruments might be playing in the sound sample.

Applications of the project :

1. We can use our technique to identify the noise in a music audio clip and remove unwanted part of it.

2. This method can be used in Crime solving Department where we can use it on audio clip of call records or anything like that to separate and identify the audio finely and map out scenario by audio processing.

REFERENCES

1. https://in.mathworks.com/matlabcentral/fileexchange/38837-sound-analysis-with-matlab-implementation

2. https://www.youtube.com/watch?v=HR8nzrjuS-s

3. https://www.youtube.com/watch?v=d7BwSz3rqtU

4. https://www.youtube.com/watch?v=W38sHna0QNk&t=0s&index=6&list=WL

5. https://www.youtube.com/watch?v=CCmw0O_CwWk

6. https://www.youtube.com/watch?v=9uv3-m8jkVg&t=0s&index=2&list=WL

7. http://mathworld.wolfram.com/FourierSeries.html

8. https://in.mathworks.com/help/matlab/ref/fft.html

9. http://www.fftw.org/

10. https://en.wikipedia.org/wiki/Audio_signal_processing

11. http://www.zytrax.com/tech/audio/audio.html

12. http://mathworld.wolfram.com/images/eps-gif/FourierSeriesExamples_1200.gif

Book: Frigo, M., and S. G. Johnson. "FFTW: An Adaptive Software Architecture for the FFT." Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Vol. 3, 1998, pp. 1381-1384.