**DEPARTMENT OF**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**Faculty of Engineering and Technology SRM Institute of Science and Technology**

MINI PROJECT REPORT

ODD Semester, 2021-2022

Lab code & Sub Name: DIGITAL SIGNAL PROCESSING & 18ECC204J

Year & Semester: 3rd Year & 5th Semester

Project Title: Detection and Identification of frequency(value) in an audio

Lab Supervisor : Ms. U SNEHALATHA

Team Members (with Reg No.s) : PALLA SAI ANEESH – RA1911004010666

POLA YASWANTH REDDY – RA1911004010658

|  |  |  |
| --- | --- | --- |
| Reg. No | RA1911004010666  P.SAI ANEESH | RA19110040 10658  P YASWANTH |
| Mark split up |
| Novelty in the project work  (2 marks) |  |  |
| Level of understanding of the design formula (4 marks) |  |  |
| Contribution to the project  (2 Marks) |  |  |
| Report writing (2 Marks) |  |  |
| **Total (10 Marks)** |  |  |

## OBJECTIVE:

To find the frequency components in a given mixed frequency signal where the mixed signal was recorded by playing 2 tuning forks simultaneously.

The instantaneous frequency is computed from the received audio based on the discovery of each peak. This is done by maintaining a “last peak” sample number, subtracting that value from the sample number of the current peak, and multiplying by the sample time. For example a 1 kHz tone sampled at 8 kHz precisely at its peaks, will result in 8 samples between each peak. So if a current peak is at sample number 8088, and the previous peak was sampled at sample number 8080, and since the sample time of 8000 Hz is 0.000125 sec, then:

(8088 – 8080) \* 0.000125 = 0.001, and

1 / 0.001 = 1000 Hz.

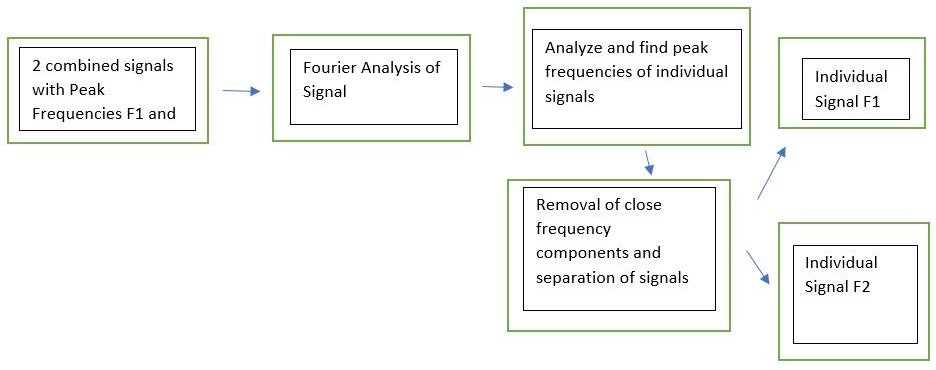
## ABSTRACT:

This is based on the process of finding Fourier transform of a singular channel signal. The signal is a singular channel wav file that is fed into the program and the audio signal is read and the signal is distinguished and the components separated.

This program finds the frequency components in a given mixed frequency signal. The given code in Sci-Lab runs a given audio file and it identifies the separate frequency components. In the process of execution, it displays the different values of frequency in the audio and displays it in the end. The magnitude of the Fourier transform of the signal is calculated and plotted, after the entire program completes execution the different frequency components are displayed, i.e., the value of the frequency. This is also applicable for singular frequency audios. Hence, through this program you can identify the different frequency components in an audio and find out its respective values.

There can be many advantages of using such a mechanism, the signal can be differentiated and analysed. The signals frequency components can be found and displayed. All the noise can be cleared and a pure tone can be obtained.

**BLOCK DIAGRAM:**



***EXPLANATION***

***I need the frequency values of a sound file (.wav) for analysis. I know a lot of programs will give a visual graph (spectrogram) of the values but I need to raw data. I know this can be done with FFT and should be fairly easily scriptable in python but not sure how to do it exactly. So let's say that a signal in a file is .4s long then I would like multiple measurements giving an output as an array for each timepoint the program measures and what value (frequency) it found (and possibly power (dB) too).***

***The complicated thing is that I want to analyse bird songs, and they often have harmonics or the signal is over a range of frequency (e.g. 1000-2000 Hz). I would like the program to output this information as well, since this is important for the analysis I would like to do with the data :)***

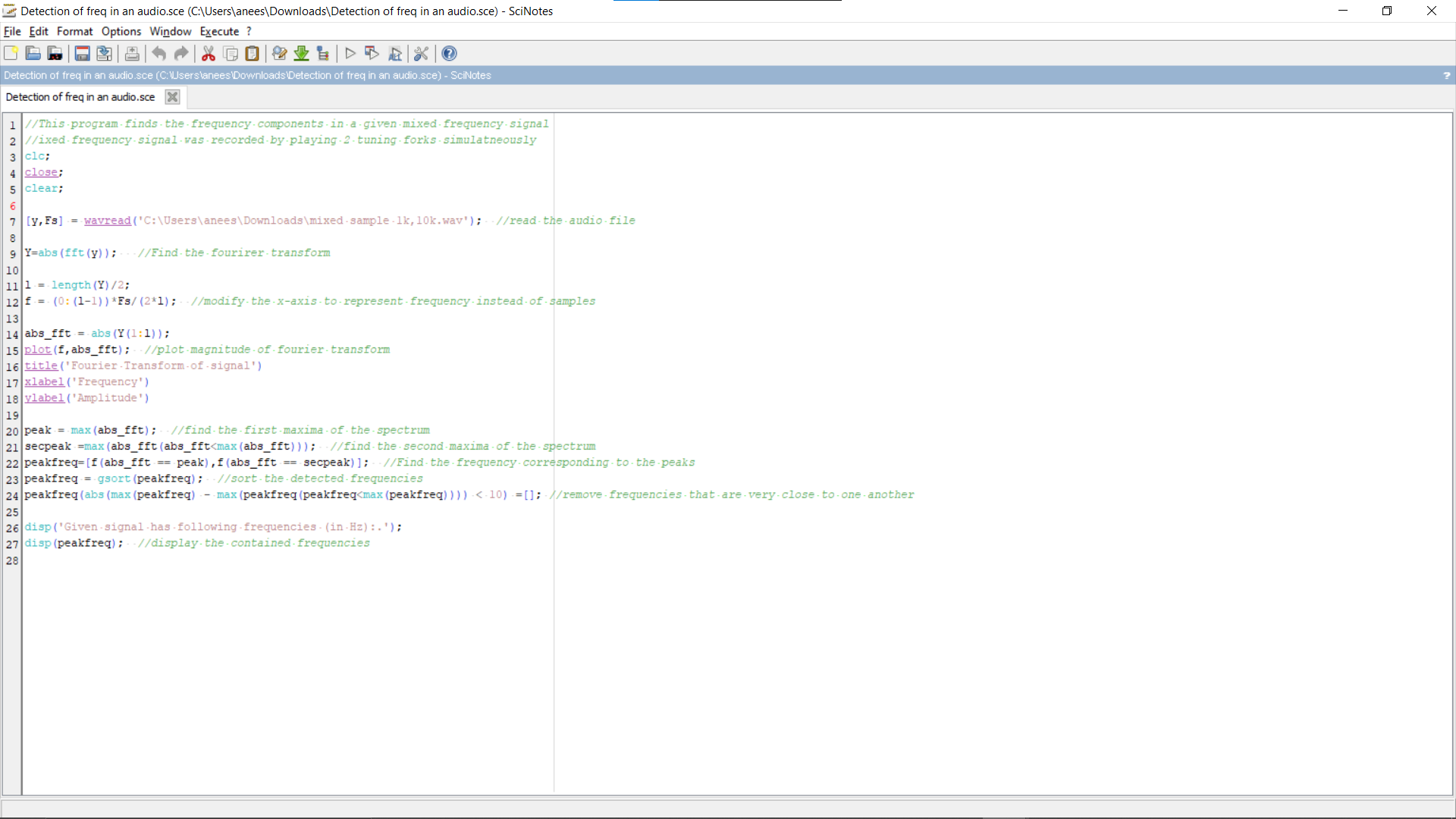
## HARDWARE/SOFTWARE REQUIREMENTS:

Sci-Lab Software, Audio File (.wav format)

# METHODOLOGY:

We try to detect the individual frequencies of the waves by finding the fast Fourier transform (FFT) of the given audio file. Then, we plot a frequency response graph and find the individual peak frequencies of the given signals, which can be noticed in the graph, with 2 individual peak frequencies shown, which are the peak frequency values of the two tuning forks. Now, these signal frequency values are sorted and values close to each other are removed, thereby allowing us to get a pure signal without noise.

## WORKING CODE:

****

**RAW CODE :**

*//This program finds the frequency components in a given mixed frequency signal*

*//ixed frequency signal was recorded by playing 2 tuning forks simulatneously*

clc;

close;

clear;

[y,Fs] = wavread('C:\Users\anees\Downloads\mixed sample 1k,10k.wav'); *//read the audio file*

Y=abs(fft(y)); *//Find the fourirer transform*

l = length(Y)/2;

f = (0:(l-1))\*Fs/(2\*l); *//modify the x-axis to represent frequency instead of samples*

abs\_fft = abs(Y(1:l));

plot(f,abs\_fft); *//plot magnitude of fourier transform*

title('Fourier Transform of signal')

xlabel('Frequency')

ylabel('Amplitude')

peak = max(abs\_fft); *//find the first maxima of the spectrum*

secpeak =max(abs\_fft(abs\_fft<max(abs\_fft))); *//find the second maxima of the spectrum*

peakfreq=[f(abs\_fft == peak),f(abs\_fft == secpeak)]; *//Find the frequency corresponding to the peaks*

peakfreq = gsort(peakfreq); *//sort the detected frequencies*

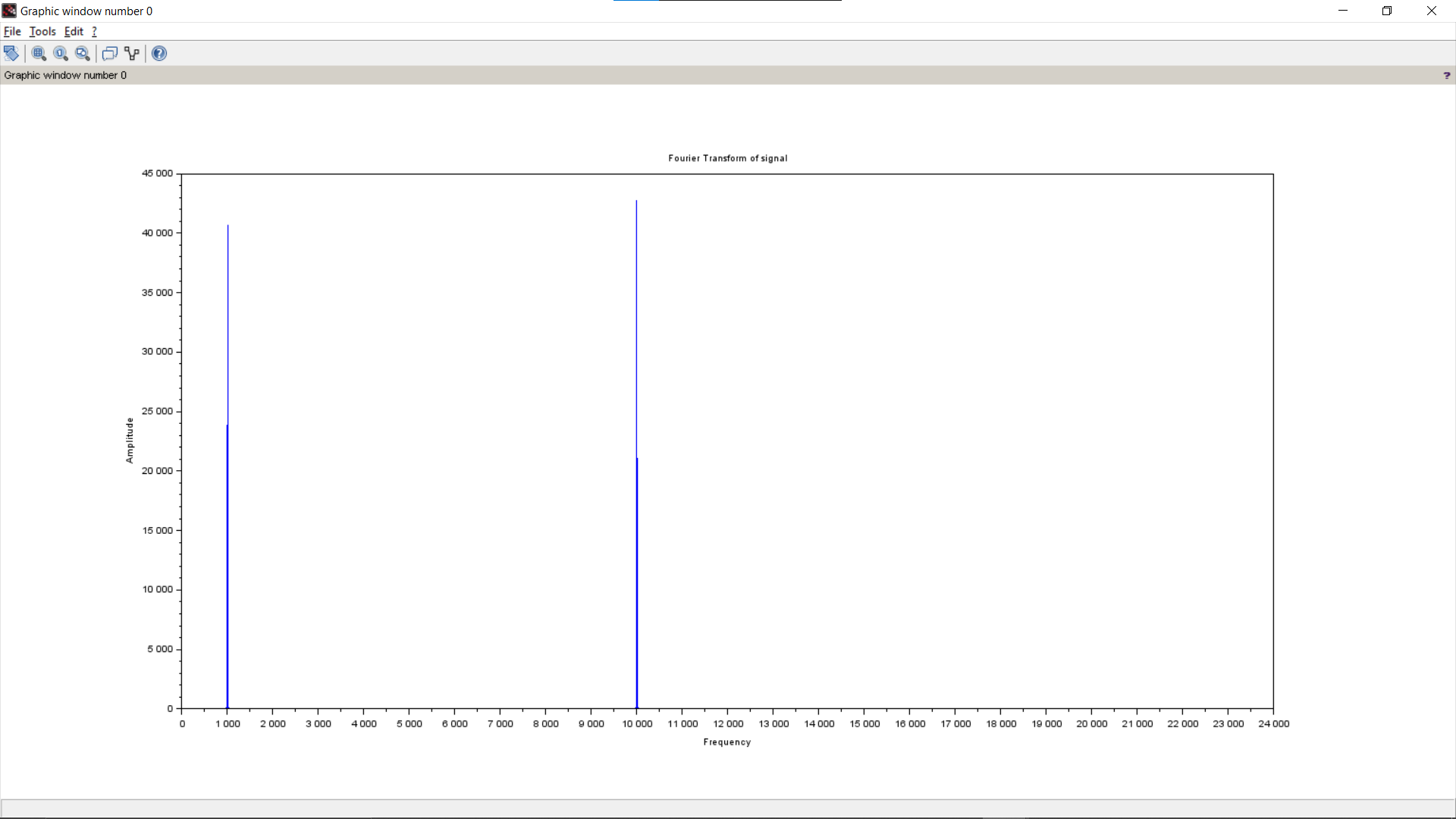
peakfreq(abs(max(peakfreq) - max(peakfreq(peakfreq<max(peakfreq)))) < 10) =[]; *//remove frequencies that are very close to one another*

disp('Given signal has following frequencies (in Hz):.');

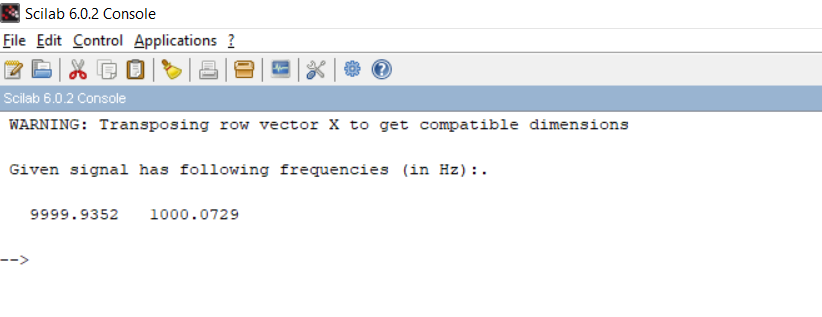
disp(peakfreq); *//display the contained frequencies*

**OUTPUT:**

**Graphical Output for audio file :**

****

**Numerical Output for above audio files respectively:**

****

## CONCLUSION:

The frequency components of the given mixed frequency signals were obtained and their graphs were plotted successfully.

# C:\Users\anees\AppData\Local\Microsoft\Windows\INetCache\IE\ETKEUWB0\Thank-You[1].jpg

# References:

[https://www.researchgate.net/publication/328041291\_A\_New\_Frequency\_Detection\_Met](https://www.researchgate.net/publication/328041291_A_New_Frequency_Detection_Method_Based_on_FFT_in_the_Application_of_SAW_Resonator_Sensor) [hod\_Based\_on\_FFT\_in\_the\_Application\_of\_SAW\_Resonator\_Sensor](https://www.researchgate.net/publication/328041291_A_New_Frequency_Detection_Method_Based_on_FFT_in_the_Application_of_SAW_Resonator_Sensor)

