Melody Transcription
EC304 Signal Processing: Project
Project Report

Anirith Pampati IMT2018516

Hruthik ch IMT2018507

Rohan Paladugu IMT2018515

Pavan Peruru IMT2018517

Contents:

"Welcome to the future. Now computers are able to hear, process and predict"
-Dave Waters

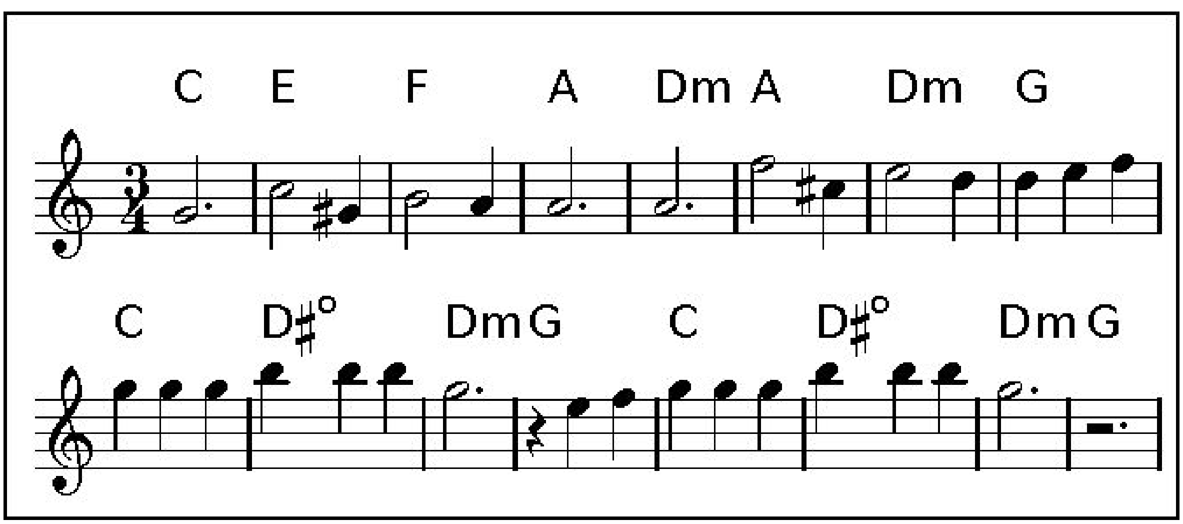
Introduction

The main aim of this project is to create a code which can identify all the notes present in the music audio sample. To explain the aim briefly, we can say when a trained musician is listening to a piano music sample, he/she can identify the note which is being played. So our code in this project must similarly print the notes being played in the audio.

"Signal Processing is mainly about interpreting the data"

So we use a 5 step process for the method of interpreting the music audio to predict the notes present in the music sample.
The process we followed for this is:
Step-1 : Filtering
Step-2 : Segmentation
Step-3 : Transcription
Step-4 : Evaluation
Step-5 : Generalisation

The process of melody transcription which is used widely in music industry(practically) is denoting each note with a special symbol as shown in the figure:

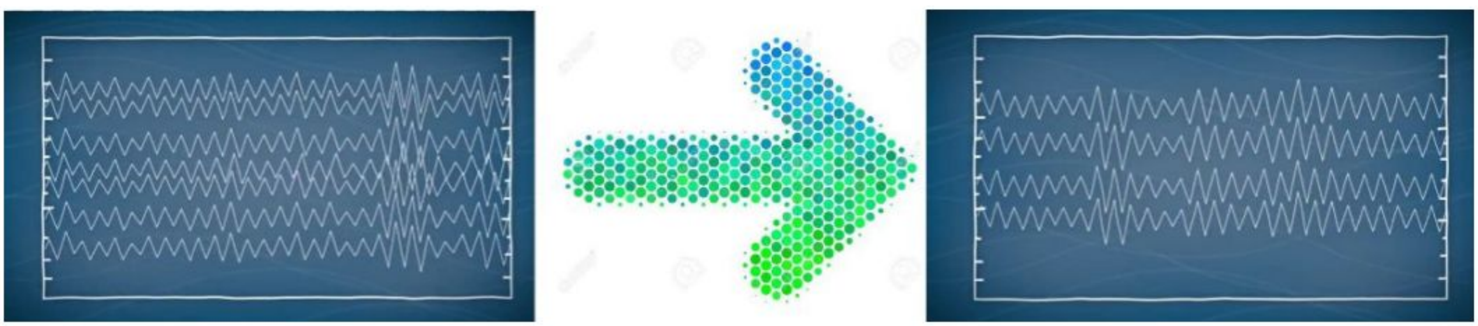


Filtering

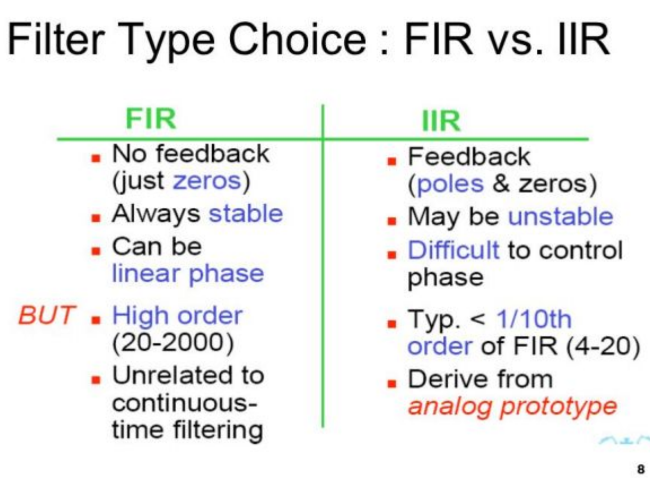
Now the aim of the first step, which is filtering is to remove the unwanted noise present in the sample. We want the note prediction to be as optimised as possible. Looking from a practical point of view, when you are listening to music, if you hear a horn sound in between, your mind automatically processes to pay attention to the music only. Similarly, here also we remove the unwanted noise in the sample and process only the required part of the sample.

A filter is an element which is capable of passing a certain limit of frequencies while attenuating or completely stopping the unwanted frequencies. In our design, we are using filters in our project to eliminate the unwanted frequencies and to obtain a certain frequency range.

For you to visualise the process of filtering, look at the figure below.



As you see in the image we have a large signal with many notes. So, our next step is to break the signal into some parts such that each part has only one note, which can be identified easily.
We have two filter types to think of. i) FIR filter and ii)IIR filter. Both the filters have their own advantages and disadvantages as well. We have to pick the type of filter which serves our purpose well.

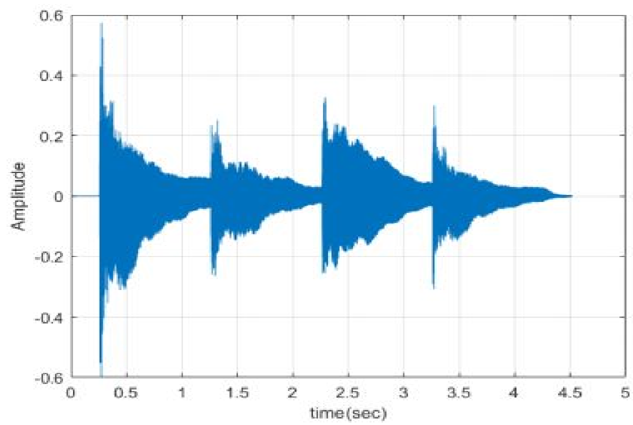


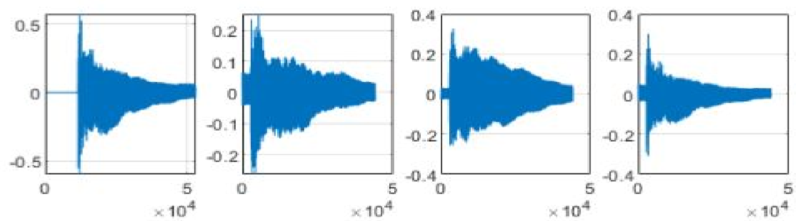
From above we say, FIR filters offer more stability to the system despite we have to perform operations at higher order. So an FIR filter can be used to apply a bandpass filter for our audio signal.

Segmentation

In our attempt to recognize the notes played in the music from a musical instrument, we have applied filters to remove the unwanted frequencies. The next step is to segment(dividing into parts) the individual notes perfectly with respect to time and then to perform the successive steps to recognize the individual note.

Each musical note has its own time duration which can't be specified as a constant. The figure below helps us to visualise the process of segmentation.



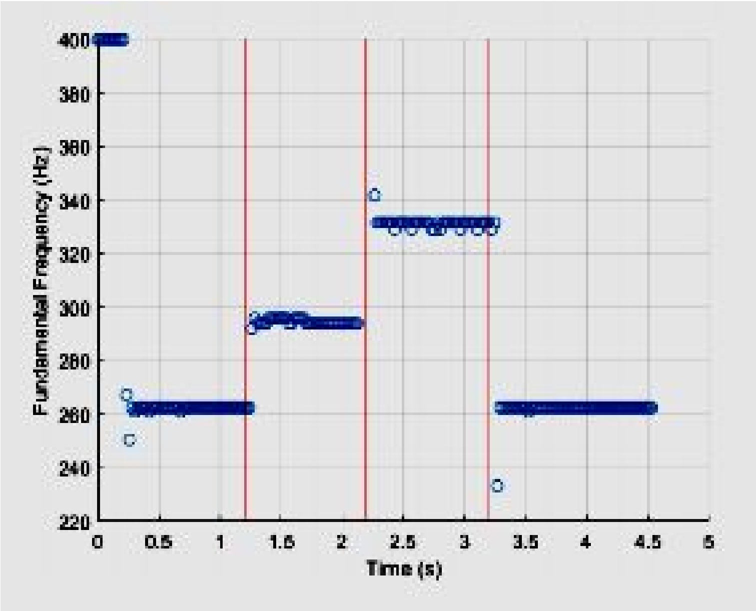


For this segmentation to divide the notes correctly, we have to identify the point of instant of note change. For this we used the difference between frequency as time proceeds.

"The main and challenging part of the whole project is the segmentation part, Because we need to find the correct instant automatically"

Explanation:

we have found the change in fundamental frequency with respect to time, which in turn helps us to identify the notes and in separating them. So we can say when the difference between y values of 2 successive points on the x-axis is greater than a threshold, Then we can break the audio at those points.So we wrote a code which can find these points by iterating through x axis and comparing the y values. For convenience we took the threshold difference to be 10Hz.

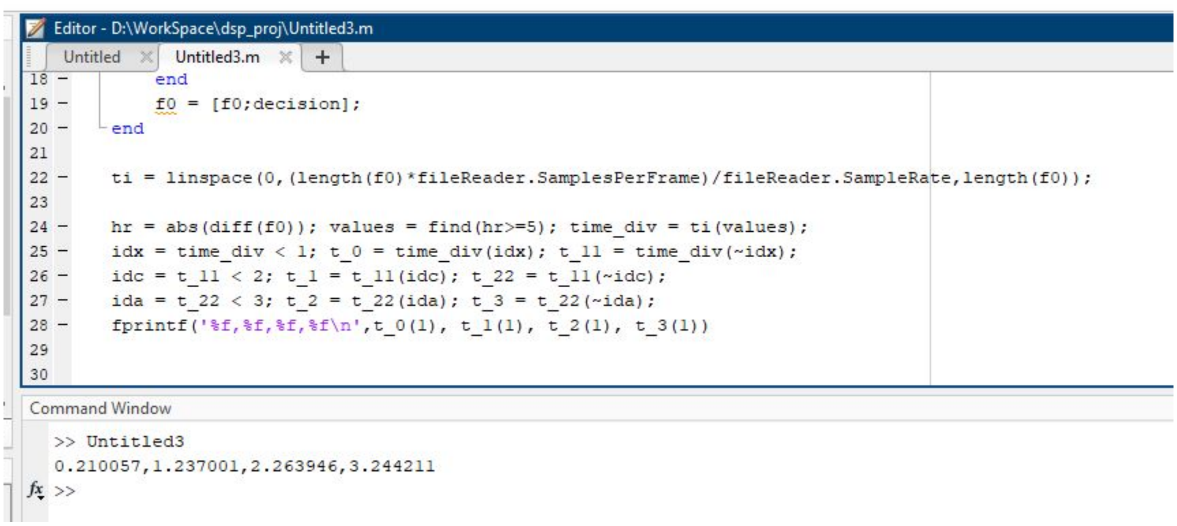


Observations:
We can see the 4 points where the difference is significant, they are : 0.2, 1.2, 2.2, 3.2 sec.

So we need those points to be printed. Which is done by our automatic segmentation code.

“The reason we do this segmentation is, we use mode function to find fundamental frequency. So, we need only one note at a time or else it gets disturbed by values of others”

We get the output of segmentation process as follows:



As you can clearly see the points 0.2, 1.2, 2.2, 3.2 are printed directly in the 4 variables. Similarly we did for 8 samples in the youtube video and also a video which is recorded by using virtual guitar online. We got all the points as expected.

Conclusion: We found out the samples which have major change in the frequency and then we segmented the plot along the lines where there is change in the fundamental frequency of the signal.In this way, we were able to segment the audio signal to get the individual notes.

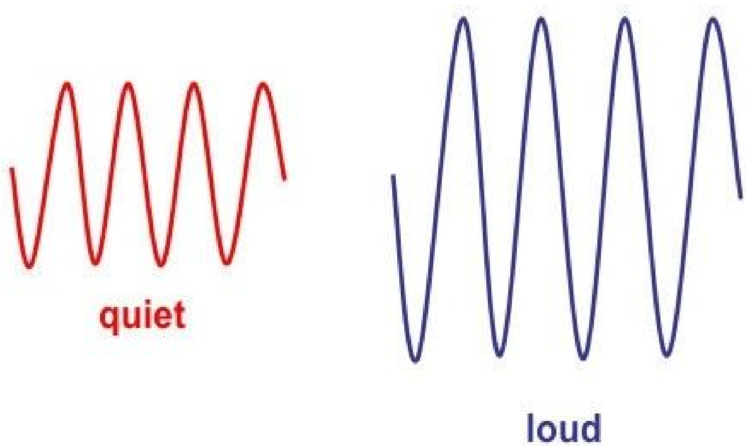
Transcription

Once we segmented the individual notes properly, the process of the transcription is much easier. In this step, all we need to do is to identify the note(note recognition). For this we need to look at the characteristics of a musical note so that we can differentiate and transcribe them based on their characteristics. The different characteristics of a musical note are: loudness, pitch and timbre.

"There are various aspects of music transcription such as pitch analysis, rhythm analysis, percussion transcription, source separation, instrument recognition, and music structure analysis."

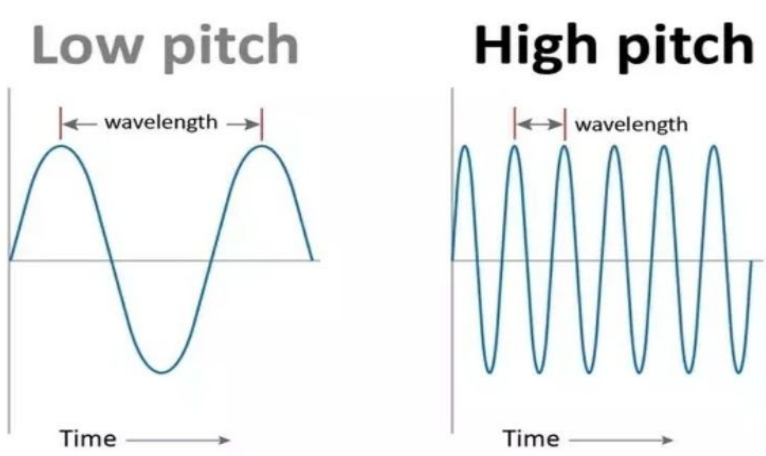
Now we will differentiate these characteristics from a musician's point of view:

Loudness:



The loudness of the note is measured by the magnitude of the changes in air pressure. This is controlled by how hard a piano key is pressed or how hard one blows on the mouthpiece of a saxophone. From the image we can deduce that an increase in amplitude will result in an increase in the Loudness.

Pitch:



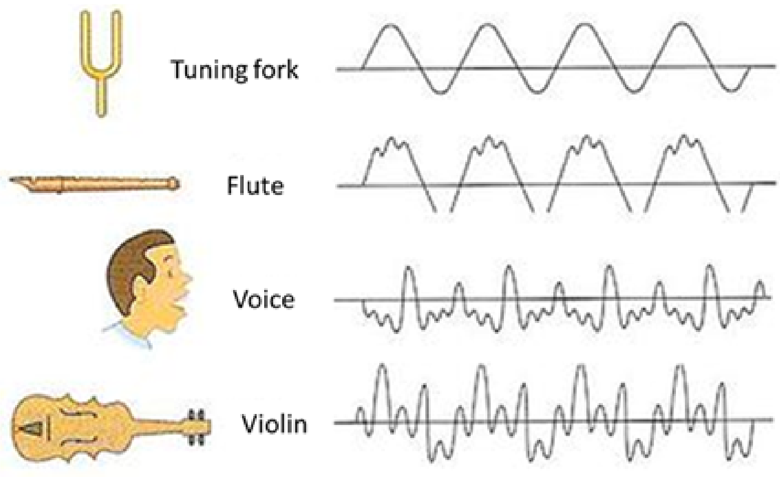
The pitch of the note is the frequency of repetition of the basic pressure pattern. More precisely, the frequency is the number of times the basic pattern is repeated per unit of time. The frequencies of interest to us will be measured in cycles per second -- one cycle per second is called a hertz in honor of Heinrich Hertz. So, for example, a note with pitch 440 hertz has a pressure function that repeats itself 440 times per second, i.e. with period 1/440 seconds. Human hearing is confined to frequencies that range roughly from 20 to 18,000 hertz.

Timbre:

The timbre of the note includes those characteristics that enable us to tell a piano note from a violin note with the same loudness and pitch.

The figure beside depicts that different sounds have different timbres(Shapes).

This figure depicts the different timbres of sounds from various instruments.



A listener can distinguish between 2 sounds of the same pitch and loudness by using the concept of Timbre. Timbre is a multidimensional concept: unlike pitch or loudness, which can be considered unidimensional, there is no single scale along which we can compare or order the timbres of different sounds.

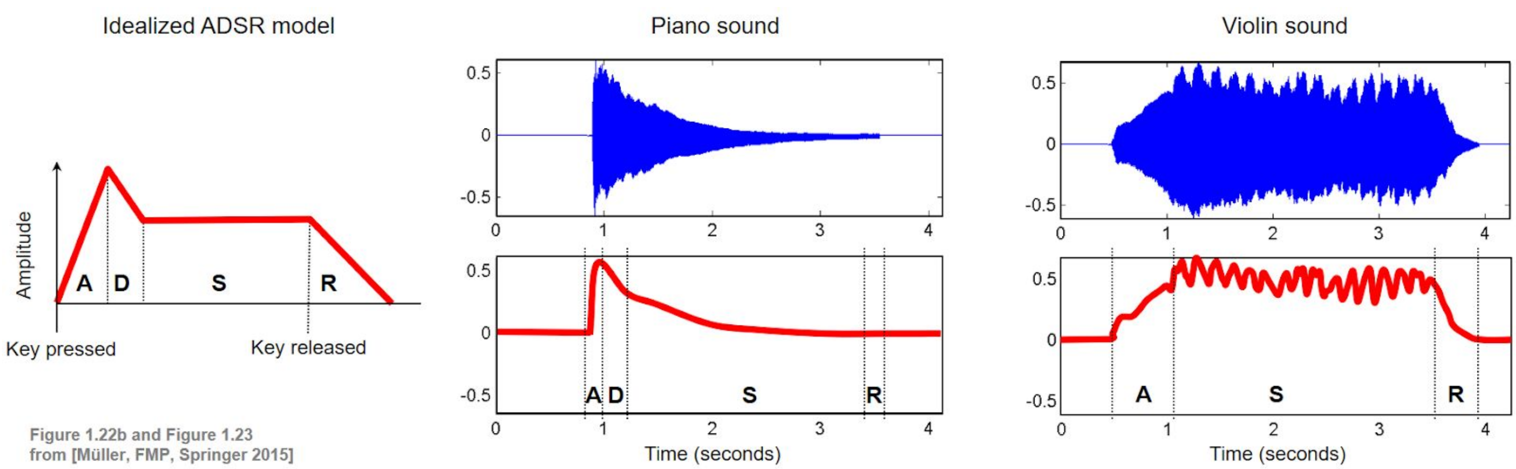


Figure 1.22b and Figure 1.23
from [Müller, FMP, Springer 2015]

There are few parameters in time and frequency domain which play a major role in determining the timbre, such as: Attack(A), Decay(D), Sustain(S), Release(R).

Typically, sound produced by an instrument has four portions:

a. Attack: duration where the sound signal goes from silence to the maximum amplitude.

b. Decay: the amplitude decays to the sustain level.

c. Sustain: the amplitude stays constant.

d. Release: duration from the moment the stimulation was stopped (key released, in the case of a piano), to the amplitude dropping down to zero.

As you can see in the image above the piano note has a very less attack time. We can say it is quicker than the violin note. Because the time in which the hammer hits the string is the only time which is considered as attack time, which is obviously very less. The decay is rapid as the string settles into harmonic equilibrium. Since the stimulation stopped the moment the hammer hit, there is usually no sustain. However, a sustain pedal can be used to have a non-zero sustain period. If there's no sustain, the decay is followed by a slow release.

"Timbre can be considered as a characteristic used for differentiating between the instruments(theoretically)"

When we hear a musical instrument sound a note, we have a general sense of its pitch. For example, we know that the piccolo sounds relatively high frequency notes and the tuba sounds relatively low frequency notes. The names associated with these notes -- "A," "C", etc. -- are determined solely by the pitch.

E_3	164.81	209.33
F_3	174.61	197.58
$F^{\#}_3/G^b_3$	185.00	186.49
G_3	196.00	176.02
$G^{\#}_3/A^b_3$	207.65	166.14
A_3	220.00	156.82
$A^{\#}_3/B^b_3$	233.08	148.02
B_3	246.94	139.71
C_4	261.63	131.87
$C^{\#}_4/D^b_4$	277.18	124.47
D_4	293.66	117.48
$D^{\#}_4/E^b_4$	311.13	110.89
E_4	329.63	104.66
F_4	349.23	98.79
$F^{\#}_4/G^b_4$	369.99	93.24
G_4	392.00	88.01
$G^{\#}_4/A^b_4$	415.30	83.07
A_4	440.00	78.41
$A^{\#}_4/B^b_4$	466.16	74.01
B_4	493.88	69.85
C_5	523.25	65.93
$C^{\#}_5/D^b_5$	554.37	62.23
D_5	587.33	58.74
$D^{\#}_5/E^b_5$	622.25	55.44

So we want to recognize the note by using frequencies. We found the fundamental frequency of each individual note and mapped them to the alphabet which should have that fundamental frequency.

This figure shows the values of frequency we used in our program. We took it from google. The second column is the frequency of the note given in the first column in Hz. The third column is the wavelength of the note given in the first column in cm.

We have kept a margin of 2 Hz to map them because these values may differ from the practical values. So we just mapped the values of frequency we found with a 2Hz range of the frequency from this table.

In mathematical language we took:
If
$(\text{Fund. freq})_{\text{X note}} - 2 < (\text{Fund. freq})_{\text{ex}} < (\text{Fund. freq})_{\text{X note}}\ 2$
Then ex note = X note.

Evaluation

This is the final step towards our goal. So we have fundamental frequencies present in the sample and the table of frequencies. So the thing we do here is just map the values using the steps already as described above(2Hz range). The procedure we followed here was we created 3 arrays as described below.

"Mode of the pitches of the frequencies in the segmented samples is the fundamental frequency of that sample"

First array: Fundamental Frequencies present in the sample.
freq = [f0_seg1 f0_seg2 f0_seg3 f0_seg4];

Second array: Names of notes in a particular order.
Notes=['C','C','D','D','E','F','F','G','G','A','A','B','C','C','D','D','E','F','F','G','G','A','A','B','C','C','D','D','E','F','F','G','G','A','A','B','C','C','D','D','E','F','F','G','G','A','A','B','C','C','D','D','E','F','F','G','G','A','A','B','C','C','D','D','E','F','F','G','G','A','A','B','C','C','D','D','E','F','F','G','G','A','A','B','C','C','D','D','E','F','F','G','G','A','A','B','C','C','D','D','E','F','F','G','G','A','A','B'];

Third array: Frequencies of the notes in an order which is used in the second array.
f_absval = [16.35, 17.32, 18.35, 19.45, 20.60, 21.83, 23.12, 24.50, 25.96, 27.50, 29.14, 30.87, 32.70, 34.65, 36.71, 38.89, 41.20, 43.65, 46.25, 49.00, 51.91, 55.00, 58.27,61.74, 65.41, 69.30, 73.42, 77.78, 82.41, 87.31, 92.50, 98.00, 103.83,110, 116.54, 123.47, 130.81, 138.59, 146.83, 155.56, 164.81, 174.61, 185.00, 196, 207.65, 220, 233.08,246.94, 261.63, 277.18, 293.66, 311.13, 329.63, 349.23, 369.99, 392.00, 415.30, 440, 466.16, 493.88, 523.25, 554.37, 587.33, 622.25, 659.25, 698.46, 739.99, 783.99, 830.61, 880, 932.33, 987.77, 1046.5, 1108.73, 1174.66, 1244.51, 1318.51, 1396.91, 1479.98, 1567.98, 1661.22, 1760, 1864.66, 1975.33, 2093, 2217.46, 2349.32, 2489.02, 2637.02, 2793.83, 2959.96, 3135.96, 3322.44, 3520, 3729.31, 3951.07, 4186.01, 4434.92, 4698.63, 4978.03, 5274.04, 5587.65, 5919.91,6271.93, 6644.88, 7040, 7458, 7902.13];

The process we used for mapping can be easily described in programming language as follows:

```
for i = 1:4
   for j = 1:length(f_absval)
      if abs(freq(i)-f_absval(j))<=2
         fprintf('%s\n',notes(j))
      end
   end
end
```

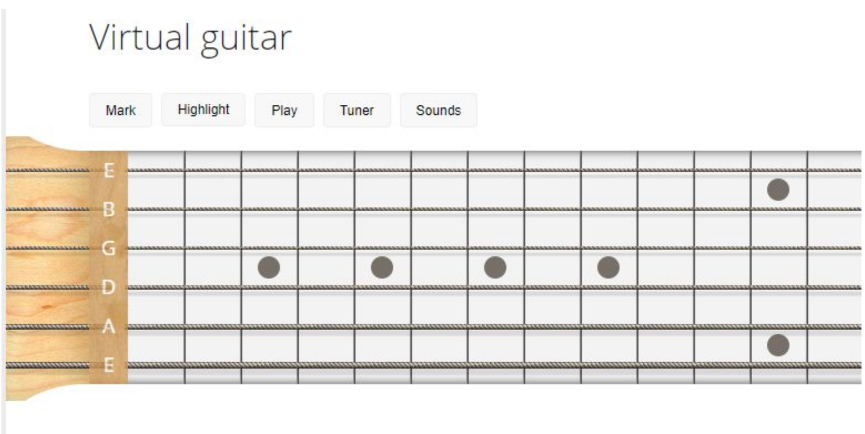
Basically we found a value of j where the notes in sample and original have an absolute difference of 2Hz and then printed the note at the j^{th} position.

Generalisation

A generalization is a form of abstraction whereby common properties of specific instances are formulated as general concepts or claims. Generalizations posit the existence of a domain or set of elements, as well as one or more common characteristics shared by those elements.

Now it is time to generalise our program. So, we checked the evaluation step for 8 samples present in the youtube video and found our answers where matching. But all the sounds in that video are from piano, So we thought the generalisation step must be used to check a different instrument.

"Any program is not called successful until it is generalised,because a program working for few conditions is not our goal it must work under any circumstances which is taken care of by the Generalisation step."



We found an interesting website called virtual guitar, where we have different strings as shown in the figure, we can make our computer make sounds like guitar, So we recorded the sound by playing a few notes and then applied our program to that sound. We found that the results were absolutely correct.

Process to use this virtual guitar : When we click on the alphabets present on the brown part of the guitar, the computer produces the sound corresponding to the note we clicked. We used our mobile phones to record the sound, and then converted the recorded .ogg file into .wav file. Finally we used the .wav file to evaluate the notes.

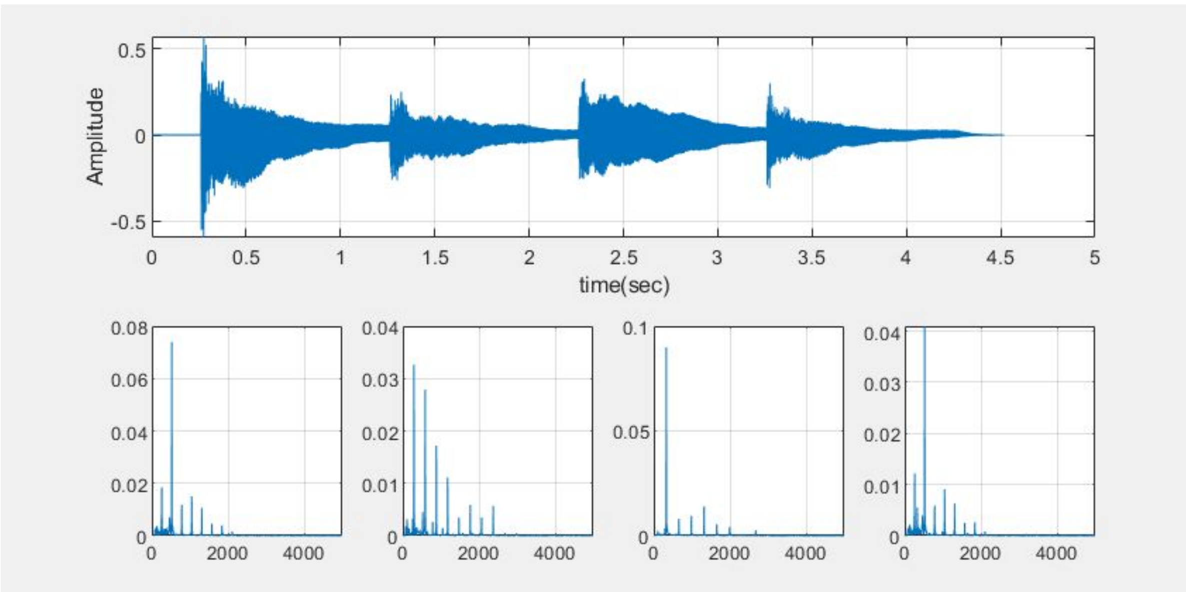
<u>Conclusion</u>:

All the results of 8 samples and samples from the virtual guitar are provided at last. So now we proved our program is working for 8 samples of youtube video and also guitar samples taken from this virtual guitar. So, we conclude our explanation, by saying that our program can be used to predict any sample given like a professional musician.

Results

In this part we submitted all our graphs of the samples with their respective notes.In all the figures you can see the time domain graphs of the samples before segmentation and frequency domain graphs after segmentation. The result printed in the command window are the notes present in the sample.

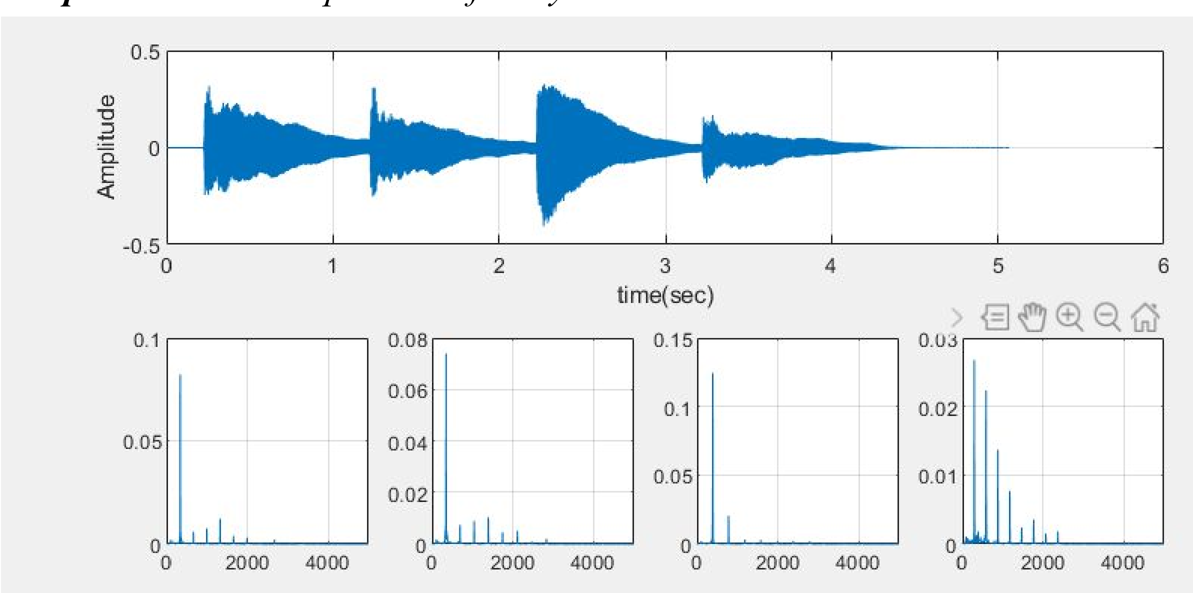
Sample-1:*Audio Sample taken from youtube video.*



Command Window

```
The notes in the audio:- C D E C
fx >>
```

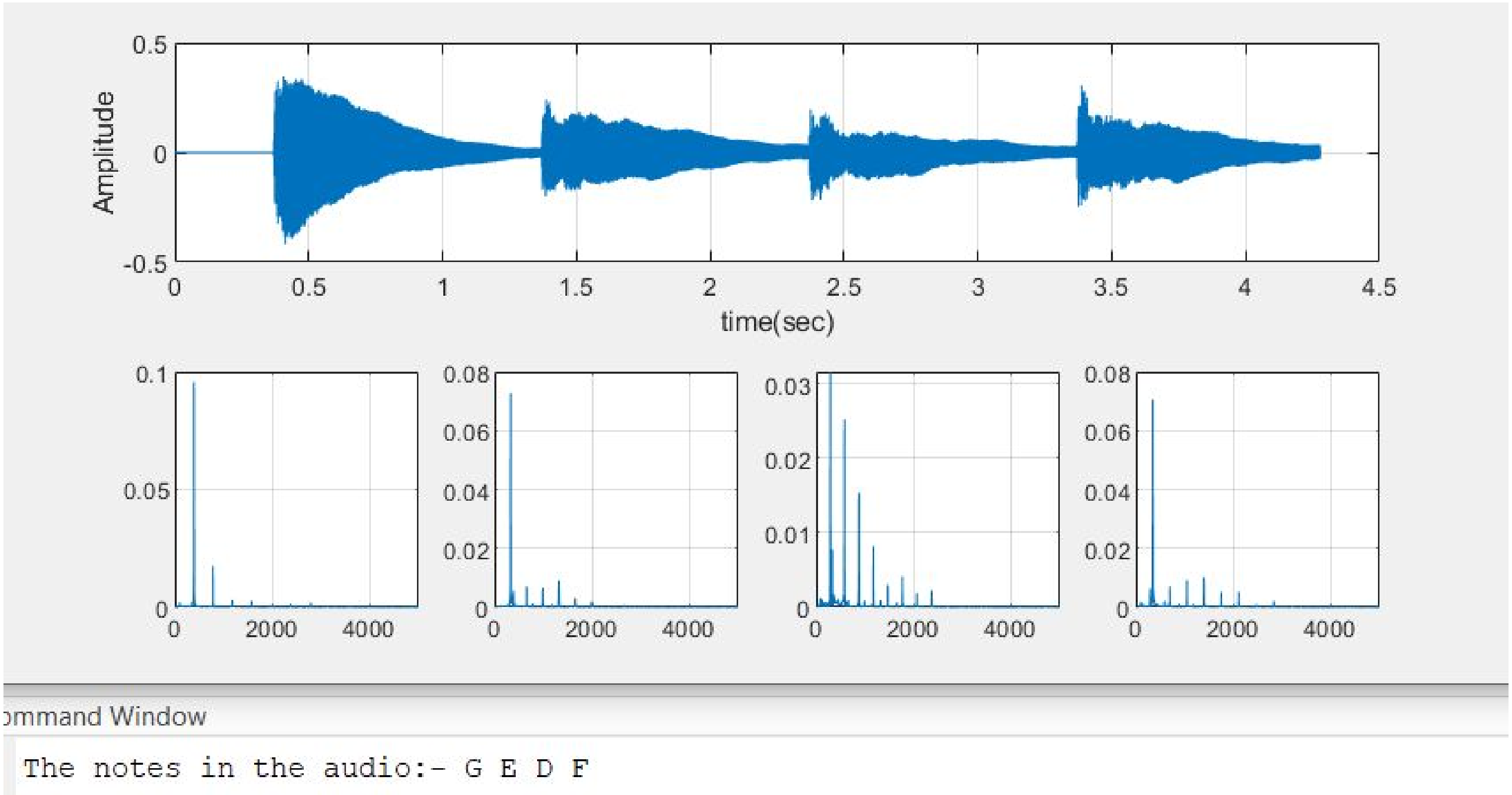
Sample-2: *Audio Sample taken from youtube video.*



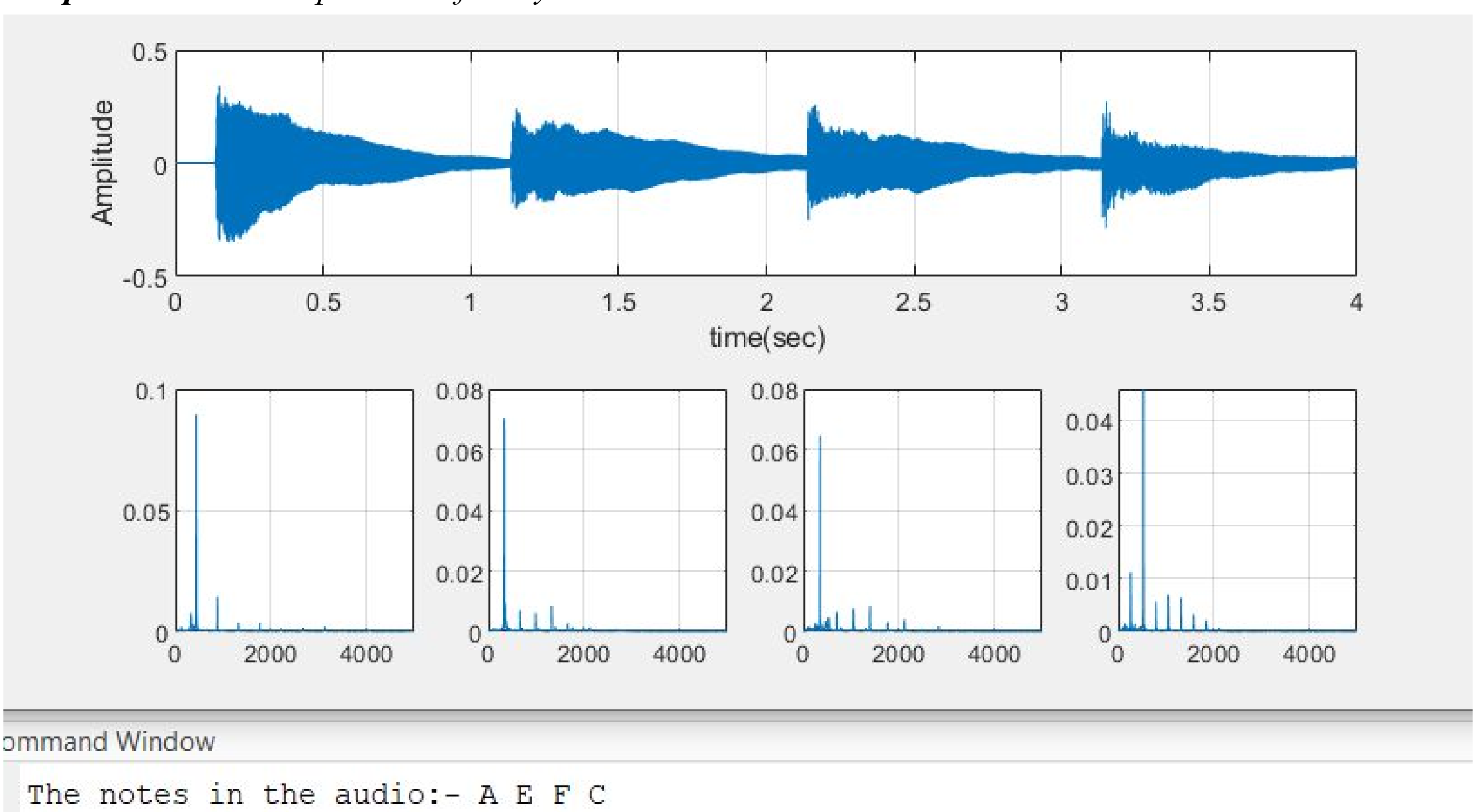
Command Window

```
The notes in the audio:- E F G D
```

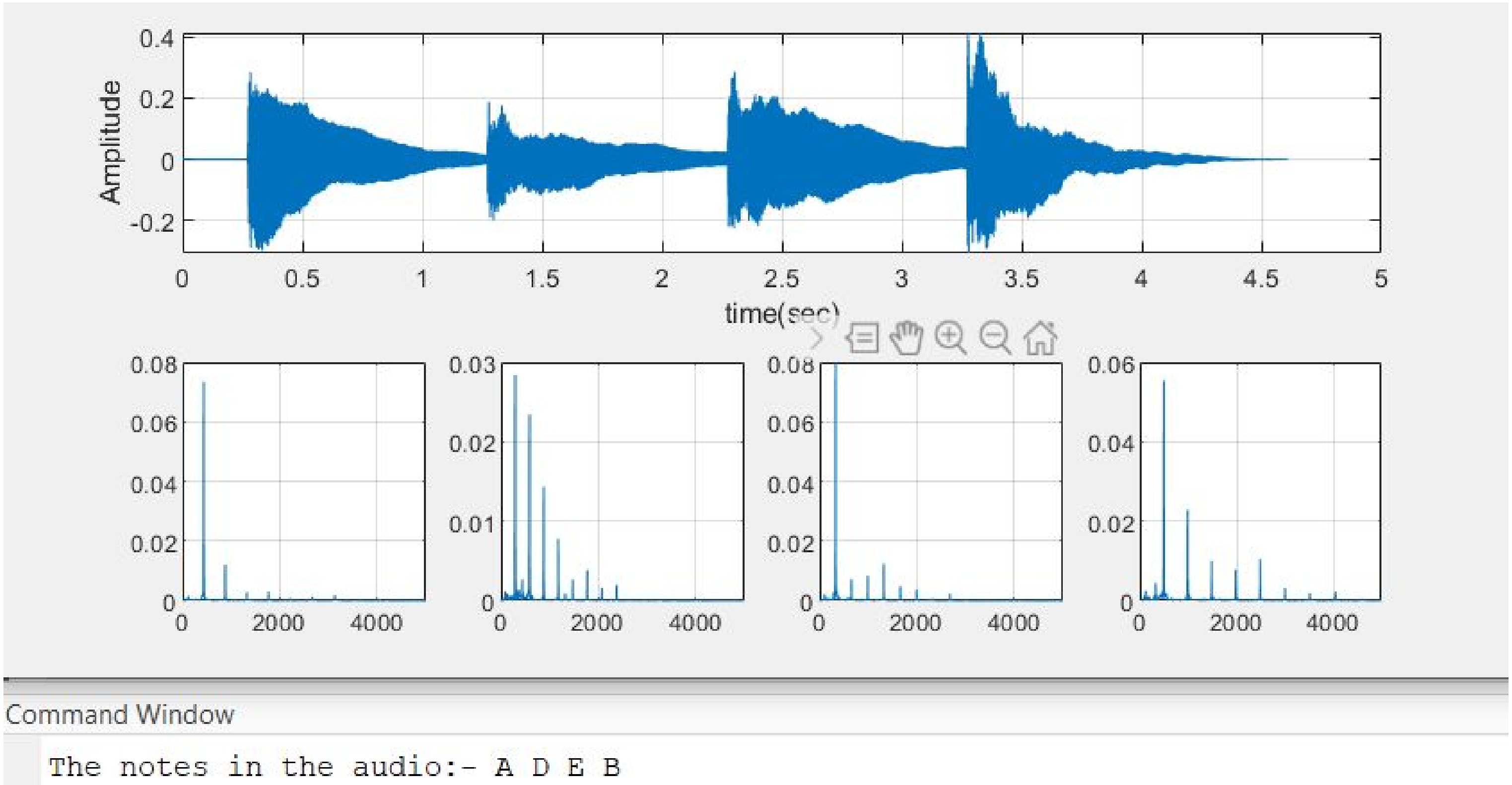
Sample-3: *Audio Sample taken from youtube video.*



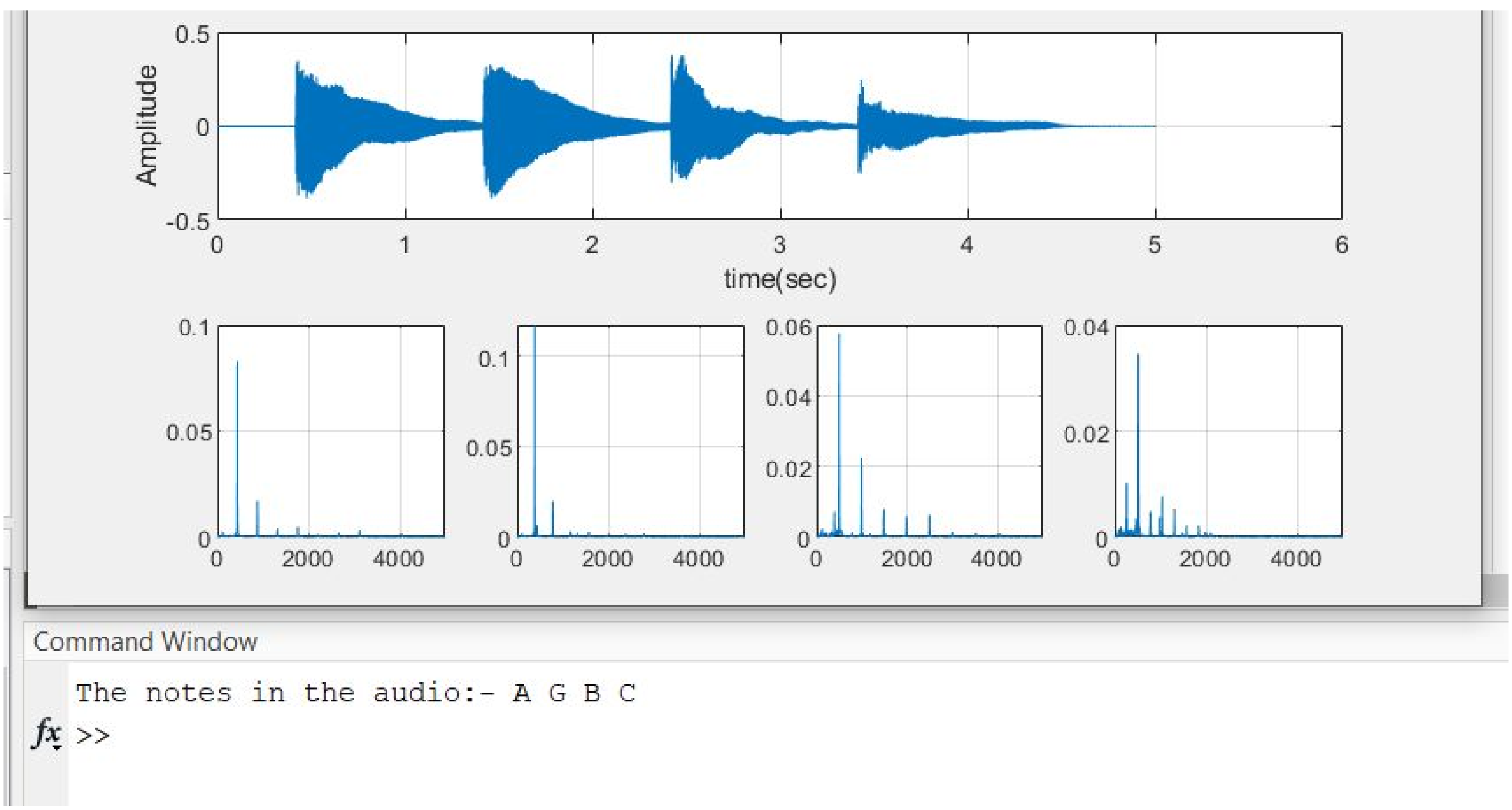
Sample-4: *Audio Sample taken from youtube video.*



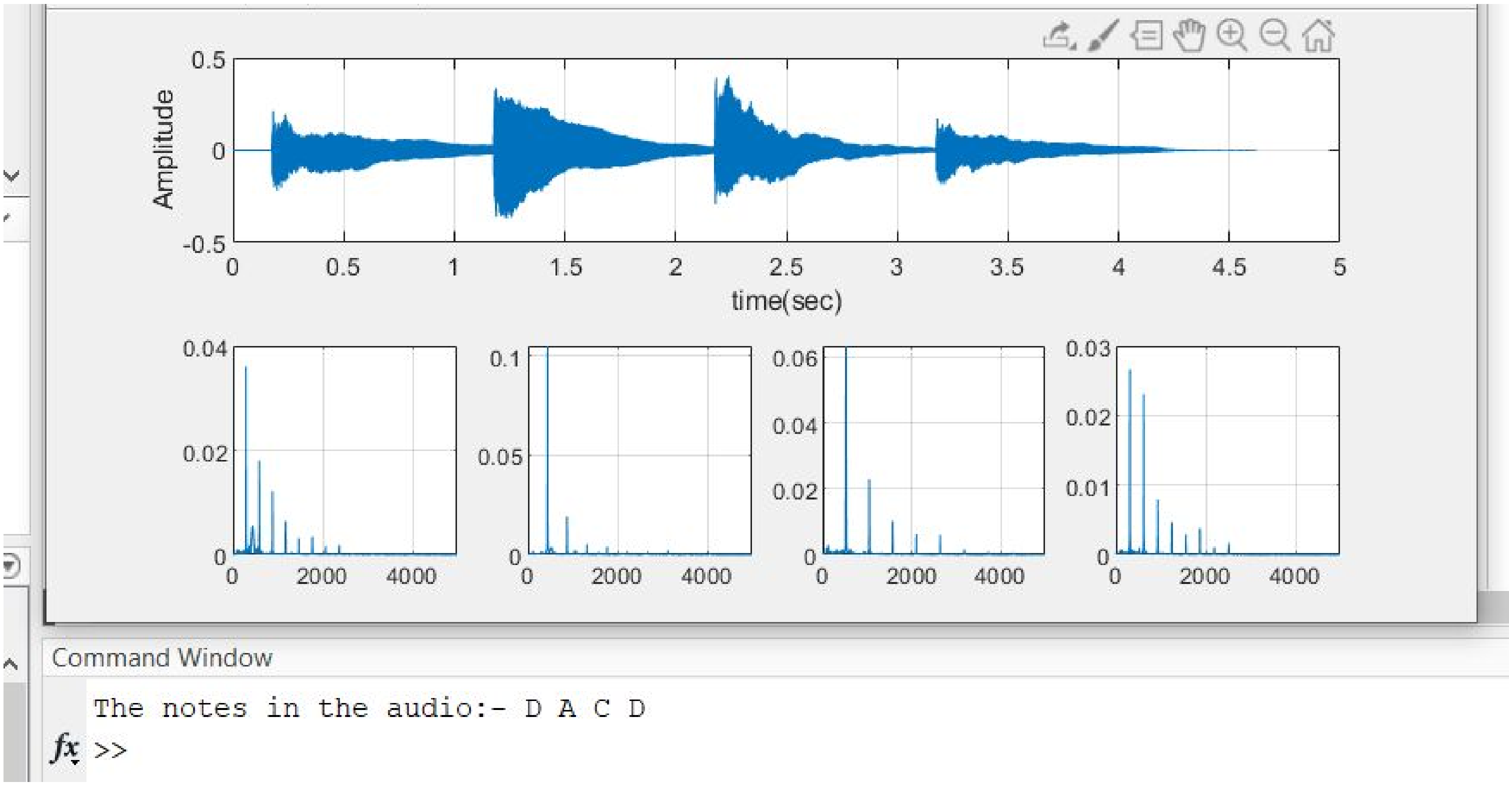
Sample-5: *Audio Sample taken from youtube video.*



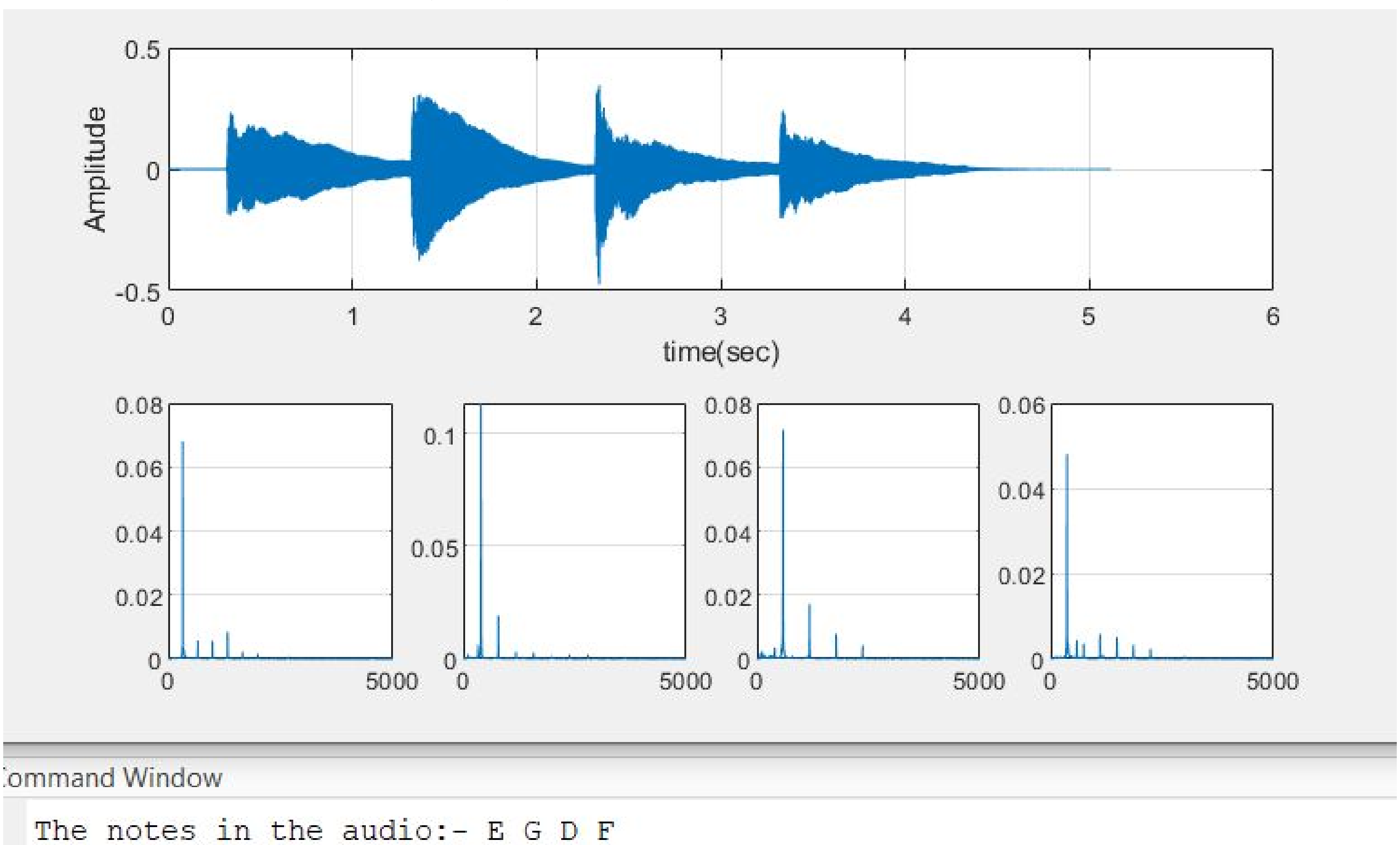
Sample-6: *Audio Sample taken from youtube video.*



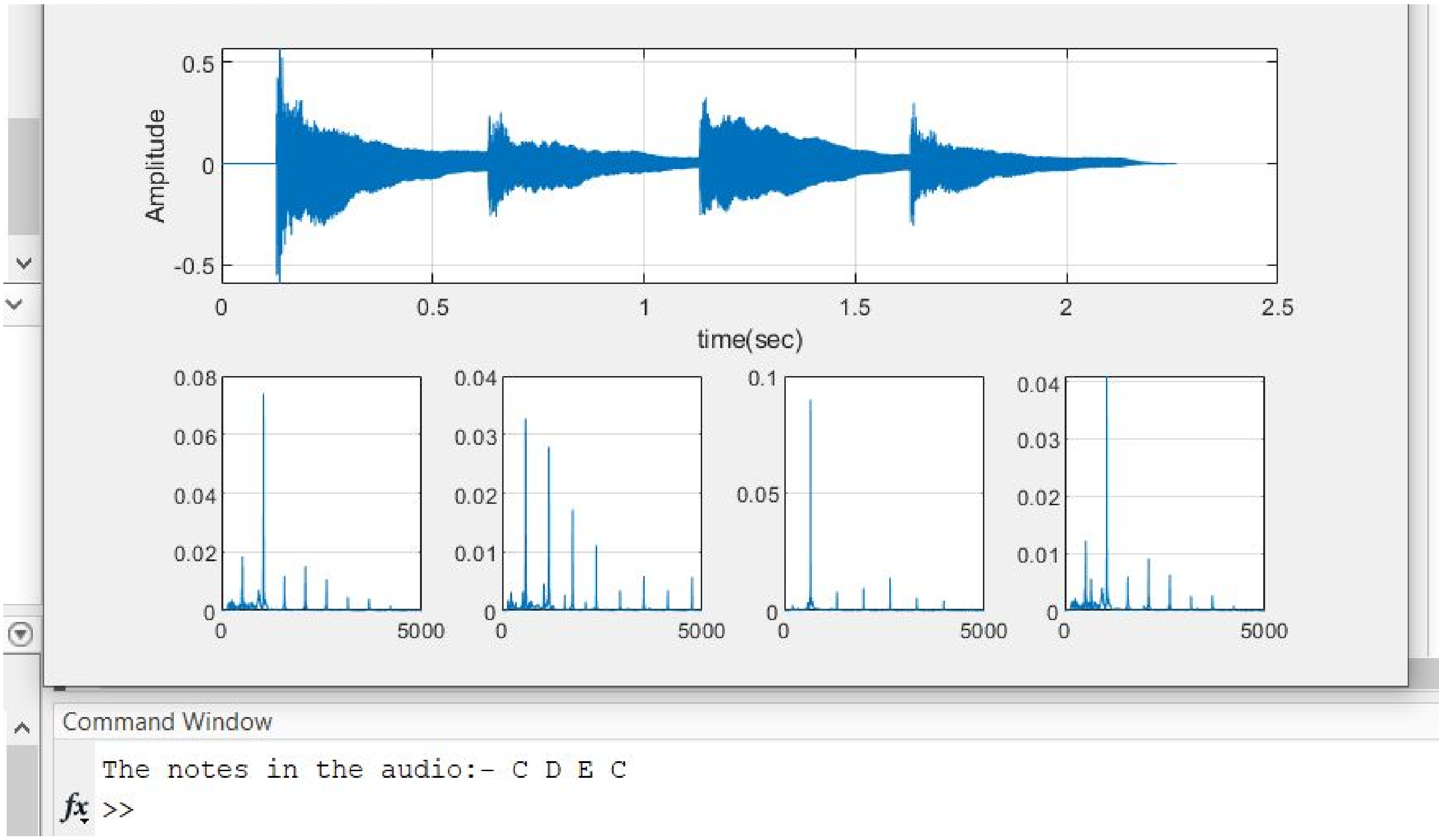
Sample-7: *Audio Sample taken from youtube video.*



Sample-8: *Audio Sample taken from youtube video.*

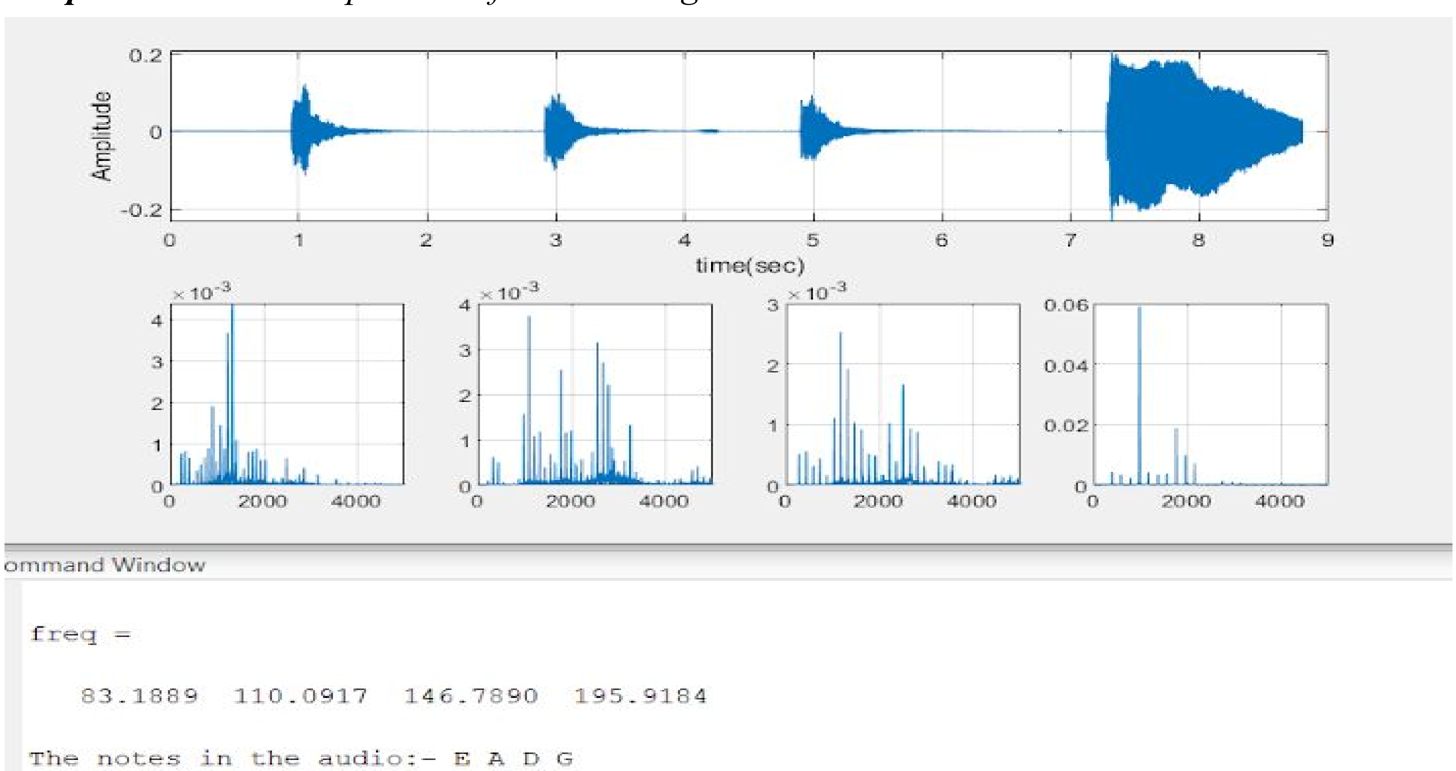


Sample-9: *Audio Sample taken from youtube video but speeded it by 2x so as to prove our program doesn't depend on speed(Taken Sample-1 only(can check from above)).*



Sample 1 has 4.5 sec time here it is 2.25(2x speed).

Sample-10: *Audio Sample taken from virtual guitar.*



References

1. Physics of Musical Notes:
https://pages.mtu.edu/~suits/notefreqs.html

2. Characteristics of Musical Notes:
https://www.phys.uconn.edu/~gibson/Notes/Section2_1/Sec2_1.htm

3. FIR and IIR Filters:
https://zone.ni.com/reference/en-XX/help/370858P-01/genmaths/genmaths/calc_filterfir_iir/

4. Virtual Guitar:
https://www.musicca.com/guitar?notes=&highlighted=&inverted=

5. Transcription:
https://www.springer.com/gp/book/9780387306674

6. Pitch Estimation:
https://in.mathworks.com/help/audio/ref/pitch.html

7. Musical Notes Identification using DSP by Jay K. Patela, E.S.Gopi. available at :
www.sciencedirect.com

8. Real Time Automated Transcription of Live Music into Sheet Music using Common Music Notation by Kevin Chan