Name: HAMID SHAHID
REG NO: 2019EE304

Designing of Guitar Tuner

Abstract: Our aim is to filter and remove noise from audio signal of each string of guitar, and in this, I used research papers to see standard frequencies of strings and other parameters as well. Audio files are saved in MATLAB and I used MATLAB to read sound and design band pass filter, then insert the audio signal with the noise signal into the filter and get an audio signal without noise. After that I took IFFT and got noiseless signal.[1]

Key words: Preprocessing and zero padding, FFT, FIR Filter, Hamming window, FFT, IFFT

INTRODUCTION

In order to play beautiful music, a musician needs to have perfectly tuned instruments. There is a method to design a guitar tuner. The tuner is designed using the filtering, measuring and analyzing capabilities of MATLAB. A guitar note is not made up of a single frequency. It consists of a number of harmonics as well. The difference in the harmonics is what makes a guitar and a violin sound different even while playing the same note. Once a sample note is played on the guitar, its fundamental frequency has to be calculated. Designing is implemented by different steps which I will explain you later in this report,

Step 01: Import Audio files in MATLAB using build-in function and perform zero-padding.
Step 02: Find the Frequency Response of audio of each string using FFT.
Step 03: Compare the Peak of fundamental Frequencies of strings with standard frequencies of string in research paper.
Step 04: Design a FIR Band-pass Filter using hamming Window technique and convolve the audio signals with designed filter to get noiseless audio.
Step 05: Again find the Frequency Response of audio (noiseless) of each string using FFT.
Step 06: Compare the Peak of fundamental Frequencies of strings with standard frequencies of string in research paper.
Step 07: Perform IFFT to get signal in matrix.[5]
Step 08: Finally restore the filtered audio signal of strings in audio form from matrix form.

I. IMPORT AUDIO FILES AND ZERO PADDING

First, Import audio files in MATLAB using build-in function and perform zero-padding to match length of matrix and preprocessing by adjusting Sampling rate of strings

II. FINDING THE FUNDAMENTAL FREQUENCY OF NOISY SIGNALS

The second step [3] is to find frequency response of each signal before passing through filter to get noiseless signal, using FFT and calculate the peaks of maximum frequency of frequency responses and compare them with the standard frequencies of strings. For Finding FFT, every string must have same matrix size so that's why trim and zero-padding is performed to get same size of six audio signal.

Code for FFT of Signal of E_Thin String:

```
%..FFT of Noisy signals..%
 y_1=fft(y1);
%..Ploting Of fft of SIGNALS...%
%Number of samples
N=362305
f_axis=[0:N-1]*fs_1/N;
 %plot(f_axis,abs(y_1));
%same is the procedure for other strings
```

FFT of Each String

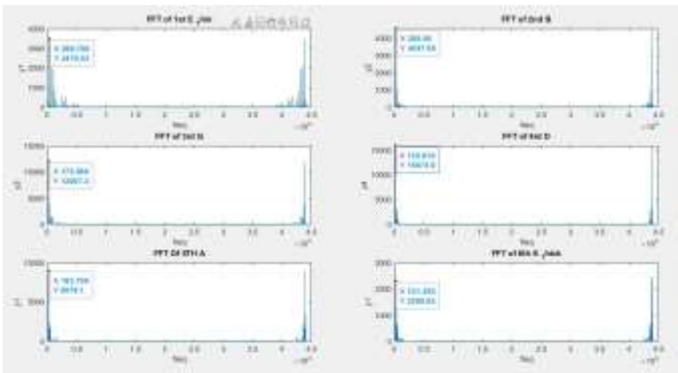


Figure 01: FFT of Signals

Noisy audios having Fundamental frequencies peak , not such close to standard frequencies of strings so we need to design FIR filter to filter the signal from noisy signal.

Name: HAMID SHAHID
REG NO: 2019EE304

METHODOLOGY

III FILTERING THE SIGNALS USING AN FIR FILTER

A. Designing a FIR Filter (using Hamming Window)

After Finding the peaks of fundamental frequencies of signals using FFT, We have to design FIR filter to filter the signal from noisy signal. For that purpose, we use hamming window having Rp=1, As=50 and band pass frequencies and band stop frequencies which are mentioned in following table I.[2]

TABLE I
Standard frequencies for guitar strings and frequencies intervals

Guitar String	Fundamental Frequency	Frequency Interval (Pass Band)
E2	82.4 Hz	60 Hz - 96.2 Hz
A2	110.0 Hz	96.2 Hz - 128.4 Hz
D3	146.8 Hz	128.4 Hz - 171.4 Hz
G3	196.0 Hz	171.4 Hz - 221.45 Hz
B3	246.9 Hz	221.45 Hz - 288.25 Hz
E4	329.6 Hz	288.25 Hz - 380 Hz

Code for Designing FIR Filter for E_thin string:

```
fpass1=247;
 fstop1=329.628;
 fsample=44100;
 w_p1=(fpass1/fsample)*2*pi;
 w_s1=(fstop1/fsample)*2*pi;
 fpass2=741;
 fstop2=988.884;
 w_p2=(fpass2/fsample)*2*pi;
 w_s2=(fstop2/fsample)*2*pi;
  tr_width_1 = w_s1 - w_p1;
  M = ceil(6.6*pi/tr_width_1) +1
 n=[0:1:M-1];
  wc_1 = (w_s1+w_p1)/2; % Ideal LPF cutoff frequency
  tr_width_2 = w_s2 - w_p2;
  M = ceil(6.6*pi/tr_width_2) +1
 n=[0:1:M-1];
  wc_2 = (w_s2+w_p2)/2; % Ideal LPF cutoff frequency
  hd = ideal_lp(wc_1, M) - ideal_lp(wc_2, M);
  w_ha = (hamming(M))';
  h = hd .* w_ha;
   [db,mag,pha,grd,w] = freqz_m(h,[1]);
  deltaw = 2*pi/1000;
  Rp=-(min(db(1:1:w_p1/ deltaw+1)));% Actual Passband Ripple
  As=-round(max(db(w_s1/deltaw+1:1:501)));%Min Stopband attenuation
  Rp2=-(min(db(1:1:w_p2/ deltaw+1)));% Actual Passband Ripple
  As2=-round(max(db(w_s2/deltaw+1:1:501)));
%Min Stopband attenuation
subplot(2,2,1); stem(n,hd); title('Ideal Impulse Response')
  axis([0 M-1 -0.1 0.3]); xlabel('n'); ylabel('hd(n)')
  subplot(2,2,2); stem(n,w_black);title('hamming Window')
  axis([0 M-1 0 1.1]); xlabel('n'); ylabel('w(n)')
  subplot(2,2,3); stem(n,h);title('Actual Impulse Response')
  axis([0 M-1 -0.1 0.3]); xlabel('n'); ylabel('h(n)')
  subplot(2,2,4); plot(w/pi,db);title('Magnitude Response in dB');grid;
 axis([0 1 -100 10]); xlabel('frequency in pi units');
 ylabel('Decibels')
%same is the procedure for design filter for other strings by just changing their fp1,fp2 and fs1,fs2
```

The maximum expected fundamental frequency from E_thin string of a guitar is about 329 Hz. The harmonics from the string will occur at multiples of the fundamental frequency. Our algorithm requires the use of at least 3 harmonics from the signal. Therefore we require the filter to pass frequencies up to approximately 329*3 = 741 Hz. The minimum expected frequency from the E_thin string of a guitar is about 329.628 Hz. After making FIR bandpass filter, Now perform convolution to find noiseless output signal. [1]

Output of Designed FIR Filter:

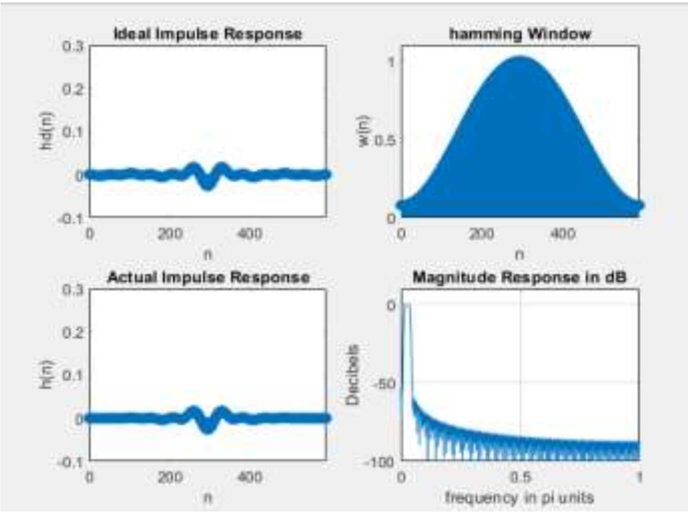


Figure 02: Design FIR Filter

The Figure 02 consists of ideal impulse, hamming window, actual and magnitude response of FIR filter

Name: HAMID SHAHID
REG NO: 2019EE304

B. Giving Input signal to Designed filter to get filtered Output

Filtered Output is obtained , noisy signal perform convolution with designed filter output .

Code for Filtered Output:

```
y_1=conv(y1,h);

 sound(y_1,fs_1)
 plot(y_1);xlabel('frequency');
 ylabel('y_1'); title('Filtered Signal')
```

Comparison Noisy and Filtered Signal

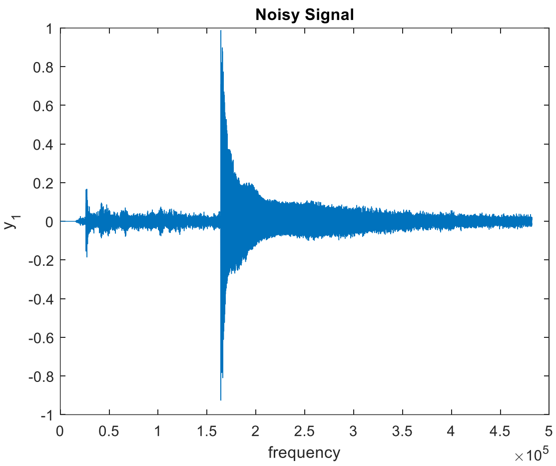


Figure 03: Noisy input of E_thin String

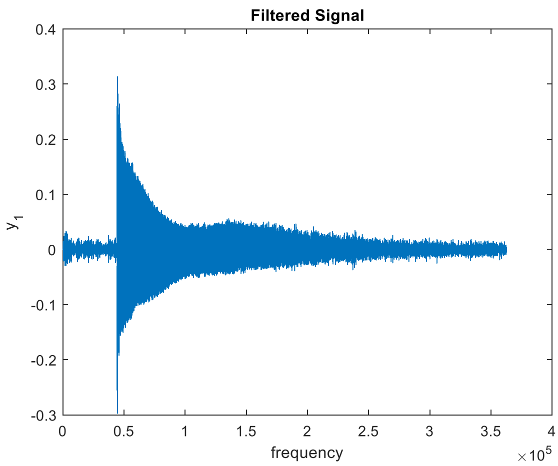


Figure 04: Filtered onput of E_thin String [4]

Same is the procedure , we have to done for rest of strings Design Filter first, give input of noisy signal of certain string and at the end compare the input and output of signals and write the conclusion that bandpass FIR filter with hamming window will provide the useful data with little noise to us.

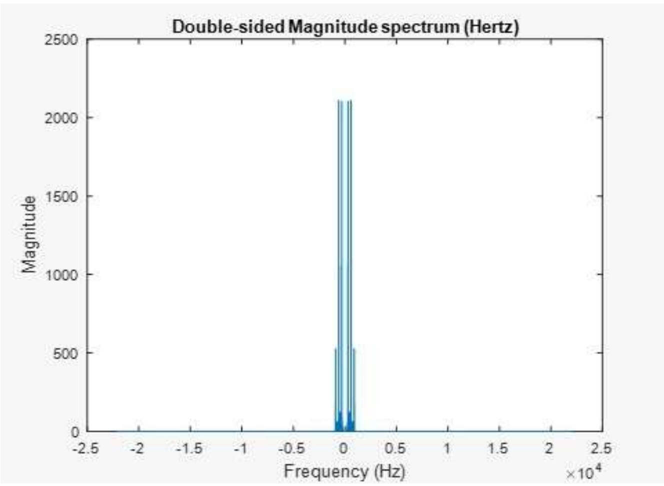
IV. FINDING THE FUNDAMENTAL FREQUENCYOF FILTERED SIGNALS:

The 4rd step is similar 2^{nd} step, find frequency response of each signal after passing through filter to get noiseless signal, using FFT and calculate the peaks of maximum frequency of frequency responses and compare them with the standard frequencies of strings. For Finding FFT, every string must have same matrix size so that's why trim and zero-padding is performed to get same size of six audio signal.

Code for FFT of Filtered Signal of E_Thin String:

```
N=23165;
 figure;
 X_mags = abs(fftshift(fft(y_2)));
bin_vals = [0 : N-1];
N_2 = ceil(N/2);
fax_Hz = (bin_vals-N_2)*Fs1/N;
plot(fax_Hz, X_mags)
xlabel('Frequency (Hz)')
ylabel('Magnitude');
title('Double-side Magnitude spectrum (Hertz)');
 sound(y_2,44100)
 %plot(fs_1,abs(y_1))
 sound(y_2,fs_2)
```

FFT of Filtered Output of E_thin String



Filtered audios having Fundamental frequencies peak, such close to standard frequencies of strings so we used designed FIR filter to filter the signal from noisy signal.

Name: HAMID SHAHID
REG NO: 2019EE304

V. Performing IFFT

This will done to get matrix of filtered signal to convert it back oi audio signal having less noise as compare to original noisy Input signal

VI. CONCLUSION

Design of a guitar tuner is done. This describes the details on the factors involved in tuning of a guitar. The important factor being the fundamental frequency of the notes, the methods to find the filtered output. The factors that affect fundamental frequency detection such as sampling frequency, sampling duration, background noise and the logarithmic nature of signal intervals in musical notes etc discussed. The concept that the harmonic nature of guitar notes could be used to accurately determine the fundamental frequency.

VII. REFERENCS

[1] Mary Lourde R., Anjali Kuppayil Saji, ”A Digital Guitar Tuner”, International Journal of Computer Science and Information Security,Vol. 6, No. 2, 2009
[2] Jeremy F. Alm, James S. Walker, ”Time-Frequency Analysis of Musical Instruments”, 2002 Society for Industrial and Applied Mathematics, SIAM REVIEW, Vol. 44, No. 3, pp. 457–47
[3] Peiqing Cheng. Digital filtering and fast Fourier transform [M]. Beijing: Tsinghua University Press, 1988: 67-75
[4] Gorrell, L. & R. S, “Automatic Guitar Tuner”, Senior Design Project, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, December 2002.
[5] Chapagain, Ashutosh. (2019). Sound Editing using Fourier Transform (Term Paper). 10.13140/RG.2.2.13640.16645