ENGR 071 Final Project
Spectrum Analysis of Sound Signals

Sominder Gautam Mohan       Ruisen Eric Liu       Peng Zhao

May 3, 2014
Introduction

In this project, we utilize Matlab to collect audio data and subsequently analyze to determine the frequencies produced by musical instruments. In particular, this project examines the accuracy in which we can perform spectral analysis to accurately exhibit the right musical notes produced by pianos and stringed instruments. It was found that for singular, isolated notes, direct spectral analysis can accurately reveal the right notes played by a piano, but had limited success with a plucked stringed instrument, like a guitar. Chords were found to be impractical for decomposition without additional mathematical manipulation.

Theory

For this project, we used the technique of spectral analysis to determine the dominant frequencies in a live audio sample. We limited the audio sample to single, tonal notes played on various kinds of instruments. Based on the relative amplitudes of the discrete sinusoidal frequencies in the spectrum, we can determine the pitch of the note. However, because our data is coming from acoustic instruments, we face the problem of overtones throwing off our analysis.

An acoustic instrument can be thought of (given a sufficiently precise model) as a linear system. As we know, linear systems resonate, and we can derive normal modes that describe each component of their movement. We used a cello, guitar, and piano for this project, so we will examine the motion of a vibrating string. A string’s normal modes can be described as a harmonic series, where we split the string into even portions.

![Figure 1: Normal Modes of a String](image)

Figure 1: Normal Modes of a String
The frequency each portion resonates at is \( n \) times the fundamental frequency, where \( n \) is the number of divisions of the string. Therefore, when we bow a string, strike a piano key, or pluck a guitar string, the sound we perceive is the fundamental frequency with all the superimposed components of the normal modes. The sound produced by the normal modes are collectively called *overtones*.

The magnitude of successively higher overtones decreases exponentially. However acoustic instruments are designed to emphasize certain overtones. This makes them sound unique, lending each instrument a certain *timbre* recognizable by the human ear. This provides a problem for our goal. If a certain overtone is over-emphasized, its magnitude may be close to the fundamental frequency, causing difficulty to properly analyze the spectrum.

In addition, the overtones of tonal notes played on acoustic instruments are themselves tonal. The 1\textsuperscript{st}, 2\textsuperscript{nd}, 3\textsuperscript{rd}, and 4\textsuperscript{th} overtones respectively produce: 1 *octave*, 1 *octave + fifth*, 2 *octaves*, 2 *octaves + third*. All of these notes are valid pitches, so detecting false positives is difficult.

**Results**

Here is a table summarizing our program’s performance:

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Note</th>
<th>Theoretical Freq</th>
<th>Detected Freq</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piano</td>
<td>C4</td>
<td>261.626</td>
<td>261.8092</td>
<td>0.07%</td>
</tr>
<tr>
<td></td>
<td>G4</td>
<td>391.995</td>
<td>390.0147</td>
<td>0.5%</td>
</tr>
<tr>
<td>Guitar</td>
<td>A2</td>
<td>110.0</td>
<td>112.011</td>
<td>1.8%</td>
</tr>
<tr>
<td></td>
<td>G3</td>
<td>195.998</td>
<td>395.4128</td>
<td>101.7%</td>
</tr>
<tr>
<td></td>
<td>C3 &amp; E3*</td>
<td>130.8 &amp; 164.8</td>
<td>500.6763</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* This is a Triad, which our program doesn’t handle. We tried only to see what happens.
The next figure shows the original Piano C4 sound signal, and its spectrum analysis:

**Figure 2:** Piano Note: C4

The next figure shows the original Piano G4 sound signal, and its spectrum analysis:

**Figure 3:** Piano Note: G4
The next figure shows the original Guitar A2 sound signal, and its spectrum analysis:

**Figure 4:** Guitar Note: A2

The next figure shows the original Guitar G3 sound signal, and its spectrum analysis:

**Figure 5:** Guitar Note: G3
The next figure shows the original Guitar C3 & E3 chord sound signal, and its spectrum analysis:

![Original Guitar C3 E3 Chord Sound Signal](image)

![Spectrum Analysis of Signal](image)

**Figure 6:** Guitar Cord: C3, E3

**Discussion**

Our spectral analysis of individual piano tones accurately reveals the expected note as well the presence of the additional overtones. In Figure 2, we see high magnitudes matching with C4 (261 Hz), C5 (523 Hz), and G5 (784 Hz). This corresponds with our theoretical prediction of the 1st and 2nd overtones (1 octave, 1 octave + fifth). In Figure 3, the primary note G4 (392 Hz) is accompanied by G5 (784 Hz), the 1st overtone.

We note that the spectral analysis of the piano notes is extremely clear, we yield minimal noise detection, as indicated by the narrow clusters around the expected frequencies. This reflects how the sound is produced by the instrument: pressing a piano key forces a hammer to hit the corresponding note, which reverberates through the air directly towards the microphone. The effect on other strings not corresponding to overtones within the piano is minimal as their respective frequencies are asynchronous.

The same cannot be said in regards to notes produced through the guitar, as sounds waves are reflected and amplified primarily through the irregular shape of the hollow sound box that serves as the body of the guitar. As a result, the resulting combination of waves is quite chaotic, producing the rich timbre we associate by ear with the pluck of a guitar. Mechanically however, the spectral analysis yields results that are far less clear to the computer.
In Figure 4, our spectral analysis reveals the correct note: A2 (110 Hz). However, we see a greater amount of ambient noise at higher frequencies, with no clear indication of the next immediate overtone at 220 Hz. In Figure 5, we note that the magnitude of the 1st overtone G4 (392 Hz) actually exceeds that of the note of the primary note G3 (196 Hz). This is clearly a case in which the guitar produced sound waves that were far more synchronous, amplifying the magnitude of all the overtones. Indeed, this is the sole example in which the 2nd overtone D5 (587 Hz), 3rd overtone G5 (784 Hz), and 4th overtone B5 (988 Hz) are present. This matches our theoretical predictions that the overtones correspond respectively up 1 octave, 1 octave + fifth, 2 octaves, and 2 octaves + third.

Figure 5 shows the results when we intentionally play a chord of E3 and G3; We expect frequencies at 131 Hz and 165 Hz, but result in a primary magnitude at 500 Hz. 500 Hz does not even correspond to a given music note: it lies between B4 and C5, the 51st and 52nd key on the piano respectively. Clearly, we would need additional mathematical analysis to reconstruct the original signal prior to spectral analysis if we wish to accurately recognize individual notes within chords.

In addition, we would like to note that some other external factors may contribute to noise within our spectrums. The placement of the microphone, structure and natural frequency of the recording room could all impact the way the sound waves travels from instrument to receiver. If were to pursue this project further, we would also need to find an alternative method of data collection. Matlab features a non-instantaneous recording program that does not allow the user to pause, collect data, and resume recording in a precise manner. As a result, we were unable to perform concurrent spectral analysis while recording, severely limiting the potential to analyze further possibilities, such as sequential notes and rhythm.
Appendix

Matlab Main File

```matlab
clear all; clf;

%% Access Audio Device — Blue Snowball Microphone

% assuming that the snowball has devID 3. this might change. to find devid, % use "audiodevinfo" and search the struct for 'Blue Snowball (Core Audio)'
snowball = audiorecorder(44100,16,1,0);
Fs = 44100; % from snowball

%% Get 1 second of recording data (make sure sound is already playing)
recordblocking(snowball, 1);
data = getaudiodata(snowball);

figure(1);
plot(data);
title('Data');

%% Take expensive FFT of data to highlight underlying spectrum

t0 = clock;
N = 32678; %number of points in FFT
spectrum = fft(data, N);
f = 0:1/N:.5-(1/N);
absSpectrum = abs(spectrum(1:N/2));

[y, i] = max(abs(spectrum));
maxfreq = Fs*((i-1)/N) %real-world frequency
note = FrequencyToNote(maxfreq)
ms = round(etime(clock, t0) * 1000)

figure(2);
plot(f(1:743)*Fs, absSpectrum(1:743));
title('Spectrum');
```

Frequency to Note Conversion

```matlab
function [ out ] = FrequencyToNote( freq )
%FREQUENCYTONOTE Summary of this function goes here
```
Detailed explanation goes here

\[ n_{88} = 12 \times \log_2 \left( \frac{\text{freq}}{440} \right) + 49; \]

\[ \text{octave} = \text{ceil} \left( \frac{n_{88} - 4}{12} \right) ; \]
\[ \text{noteNum} = \text{rem} \left( \text{round} ( n_{88} ) , 12 \right) ; \]
\[ \text{noteStr} = ' ' ; \]
\[ \text{switch} \ \text{noteNum} \]
\[ \quad \text{case} \ 1 \]
\[ \quad \text{noteStr} = 'A' ; \]
\[ \quad \text{case} \ 2 \]
\[ \quad \text{noteStr} = 'A\#' ; \]
\[ \quad \text{case} \ 3 \]
\[ \quad \text{noteStr} = 'B' ; \]
\[ \quad \text{case} \ 4 \]
\[ \quad \text{noteStr} = 'C' ; \]
\[ \quad \text{case} \ 5 \]
\[ \quad \text{noteStr} = 'C\#' ; \]
\[ \quad \text{case} \ 6 \]
\[ \quad \text{noteStr} = 'D' ; \]
\[ \quad \text{case} \ 7 \]
\[ \quad \text{noteStr} = 'D\#' ; \]
\[ \quad \text{case} \ 8 \]
\[ \quad \text{noteStr} = 'E' ; \]
\[ \quad \text{case} \ 9 \]
\[ \quad \text{noteStr} = 'F' ; \]
\[ \quad \text{case} \ 10 \]
\[ \quad \text{noteStr} = 'F\#' ; \]
\[ \quad \text{case} \ 11 \]
\[ \quad \text{noteStr} = 'G' ; \]
\[ \quad \text{case} \ 0 \]
\[ \quad \text{noteStr} = 'G\#' ; \]
\[ \text{end} \]
\[ \text{out} = \text{strcat} ( \text{noteStr} , ' ', \text{num2str} ( \text{octave} ) ) ; \]
\[ \text{end} \]