Brandon Wright

ECE 6950

Fall 2023

**WAV-to-MIDI Audio Processing MAT LAB Program**

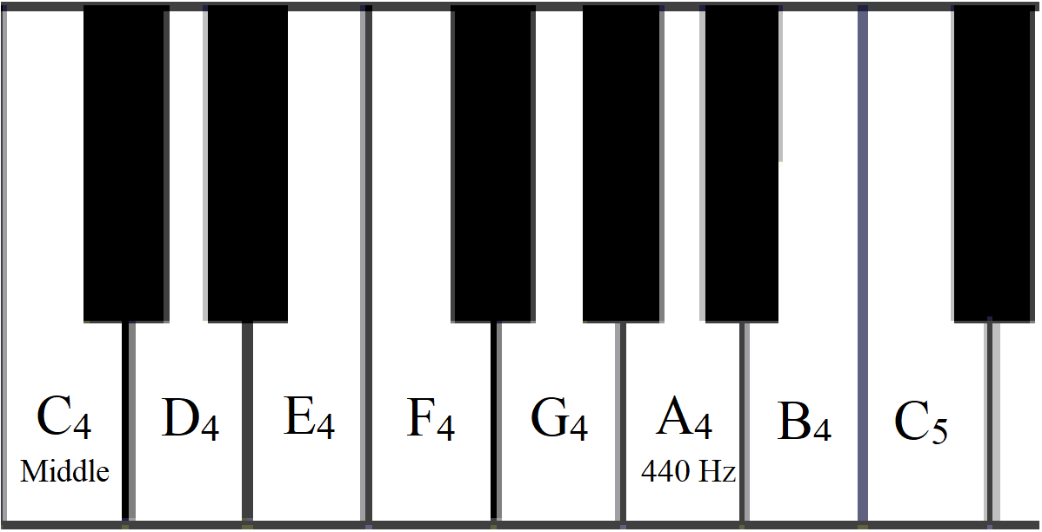
**Abstract –**

**I. INTRODUCTION**

*A. Music Notes and 12-TET Tuning*

Source DSP First chapter 4

* Insert picture of keyboard with keys

****

**Figure 1.** Music notes for one octave on keyboard

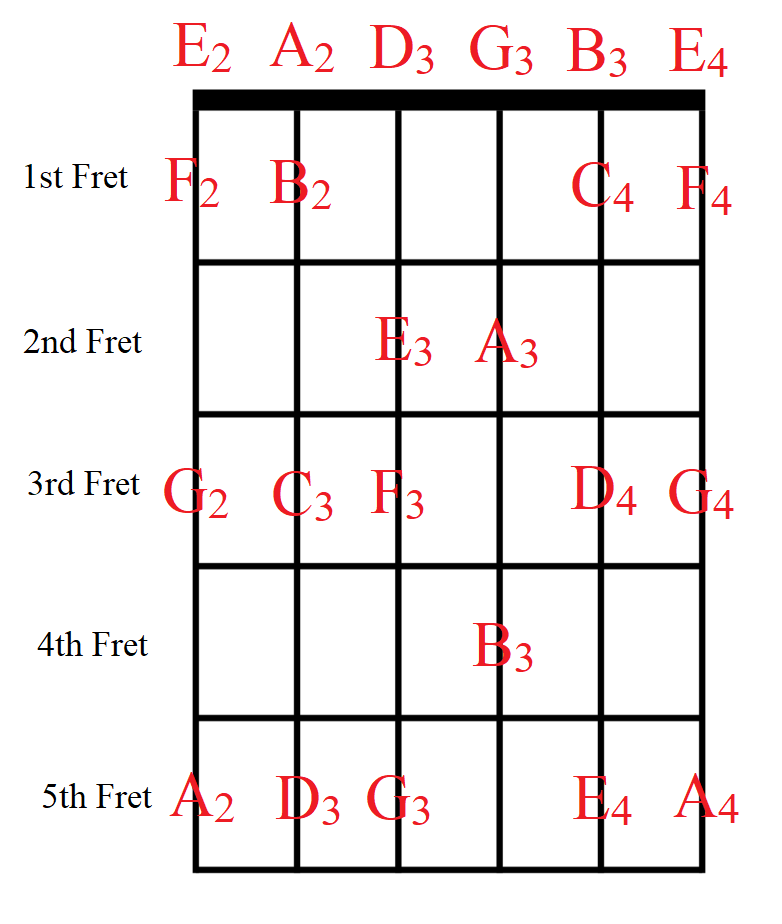
* Insert equation for keyboard notes

**Equation 1.** Converting frequency to music note

(DSP first? Or is there a better source?

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Note** | **C4** | **C♯4/D♭4** | **D4** | **D♯4/E♭4** | **E4** | **F4** | **F♯4/G♭4** | **G4** | **G♯4/A♭4** | **A4** | **A♯4/B♭4** | **B4** |
| **f0 (Hz)** | ~262 | ~277 | ~294 | ~311 | ~330 | ~349 | ~370 | ~392 | ~415 | 440 | ~466 | ~494 |

**Figure 2.** Frequencies of octave beginning with Middle C (C4)

* 

**Figure 3**. Notes on guitar fret board

* Explain how electric guitar works

*B. Musical Instrument Digital Interface (MIDI)*

Source <https://www.instructables.com/What-is-MIDI/>

* Explain what it is
* History
* Hardware and software compatibility
* Use in the music industry
* Sequence of bytes/file format
* How MATLAB handles midi
* MIDI inputs and outputs on computer
* DAWs and audio recording
* VST Instruments

*C. Digital Audio Sampling*

- ADC and sampling frequency

- Info on how windows handles sound

- Wave file format

- Technical specifications on M-Track Solo

*D. Pitch Detection with CREPE Neural Network*

- What is a neural network

- How does it learn/how is it trained

- More about CREPE in matlab

*E. Physics of sound*

- Sound pressure and volume

- How brain detects pitch

- Fundamental frequency and harmonics

**II. METHODS**

The audio from the guitar was recorded and processed using a custom script in MATLAB.

*A. Algorithm*

The following algorithm was used to convert the WAV file into a MIDI signal

**Algorithm:** WAV-to-MIDI Conversion

**Input:** N samples of soundwave amplitude for n = 1, 2, 3, … N

Find f0[n]

Find notes corresponding to f0[n]: note = 12\*log2(f0/440) + 49

Find sound pressure levels Lp[n]

Find values of n where notes change (change\_indeces)

Find note\_peaks, note\_lengths, and midi\_notes

for (i = 1; i++; i < length(change\_indeces))

if i == 1

midi\_notes(i) = notes(change\_indeces(i));

note\_peaks(i) = max(Lp(i:change\_indeces(i)));

note\_lengths(i) = change\_indeces(i);

else if i == length(change\_indeces)

note\_peaks(i) = Lp(i);

midi\_notes(i) = notes(change\_indeces(i));

note\_lengths(i) = change\_indeces(i) - change\_indeces(i-1);

else

midi\_notes(i) = notes(change\_indeces(i));

note\_lengths(i) = change\_indeces(i) - change\_indeces(i-1);

peak = max(findpeaks(pressure\_levels(change\_indeces(i-1):change\_indeces(i))));

if(isempty(peak)); peak = 0; end

note\_peaks(i) = peak;

end

end

Find time t = 0:1/Fs:length(notes)/Fs

Generate MIDI signal

for(i = 1; i++; i < length(change\_indeces)

if(midi\_notes(i) == 0)

volume(i) = 0

else

volume(i) = floor(127\*note\_peak(i)/100);

end

midi\_message =

{

command = “Note”;

channel = 1;

note = midi\_notes(i);

velocity = volume(i);

duration = note\_lengths(i)/Fs

timestamp = t(change\_indeces(i - 1) + 1);

}

end

*B. Determining Pitch and Note*

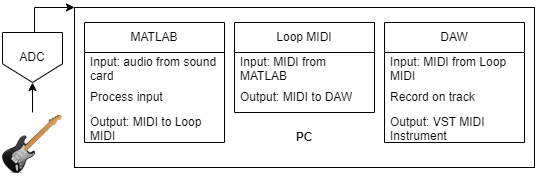
*C. Generating MIDI with Correct Note Length and Timing*

*D. Setting up Windows to record MIDI to DAW*

*E. Hardware and software setup*

Audio was recorded by plugging a Fender Telecaster electric guitar into the ¼ inch line input for an M-Audio M-Track Solo USB recording interface. The M-Track solo was connected through a USB to a personal computer. Audio processing was performed in MATLAB using plugins from MathWork’s Audio Toolbox library. A new MIDI track was created and set to record in the DAW, and a new audio recording was started.

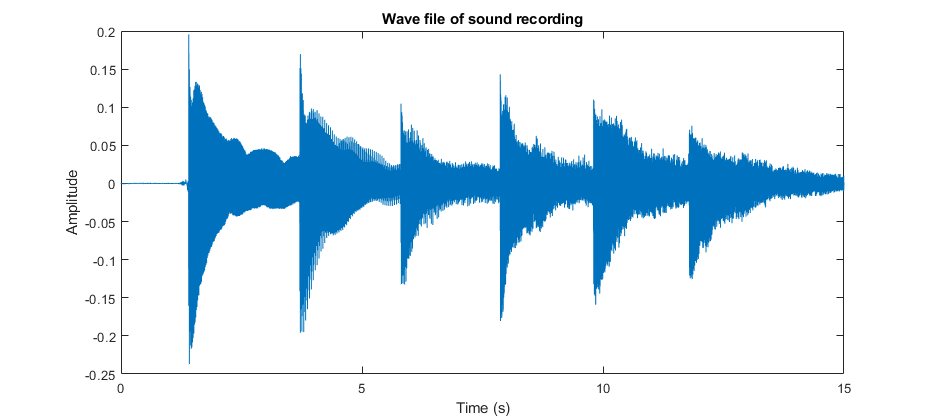
A Guitar short guitar riff was recorded in MATLAB and run through a custom script to determine the melody’s change in pitch, their corresponding musical note values along with the timing and volume information of all the notes. The script then generated a matrix of bytes from a MIDI signal, which encoded the note, its volume, its duration, and its timestamp. The MIDI signal was then streamed into Loop MIDI, which took the signal from the selected MIDI output port and looped the MIDI back into a Windows MIDI input port. Next, the DAW took the input from Loop MIDI and recorded the signal into the track as it was played in MATLAB. The recording session on the DAW was ended, and a track was successfully recorded. Next, by changing the VST MIDI instrument plugin, and desired digital instrument could be used to play back the notes of the guitar riff. Additionally, when opening up the MIDI track in the DAW, individual MIDI notes could be dragged and dropped to change pitch or timing.



**Figure \*\*\*.** Audio hardware and software setup

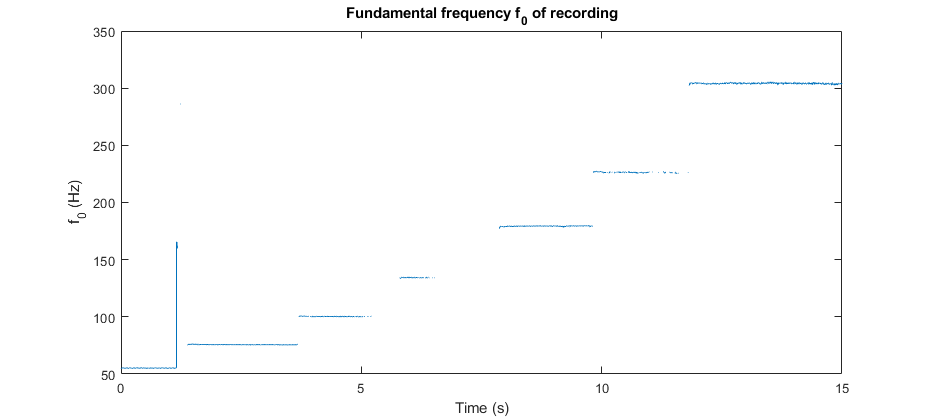
**III. RESULTS**

*A. Recording Audio*

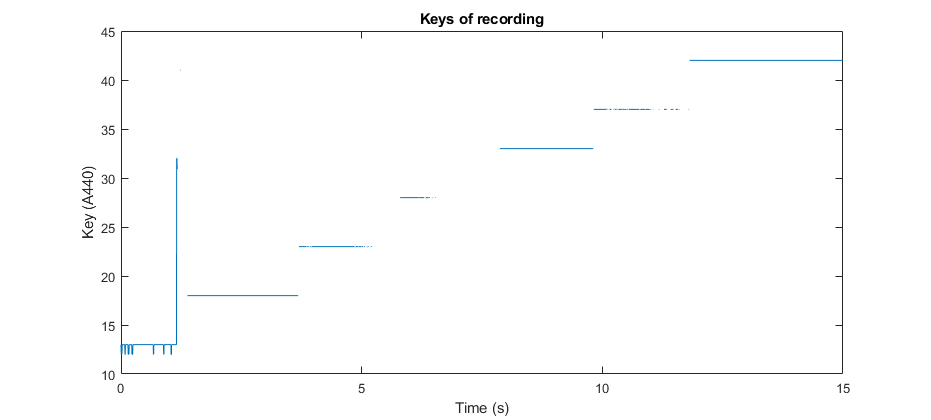


**Figure \*\*\*.** Sound wave of recording

*B. Detecting Pitch and Note*

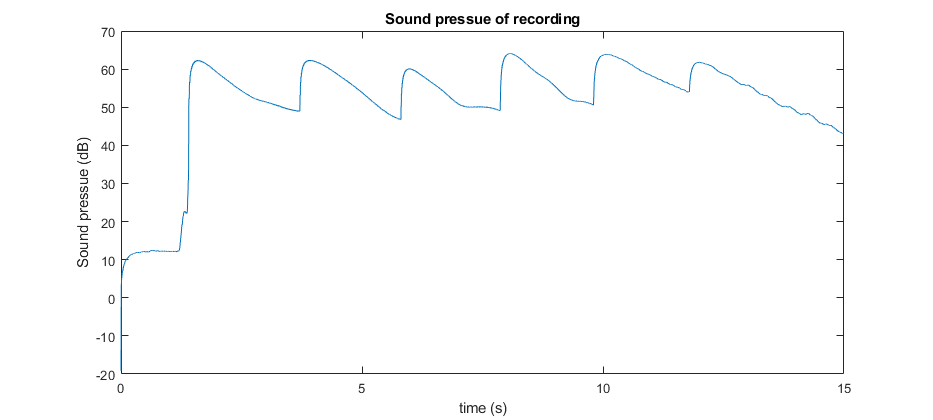


**Figure \*\*\***. Fundamental frequency of open strings on guitar



**Figure \*\*\*.** Keys of open strings on guitar

*C. Detecting sound pressure*



**Figure \*\*\*.** Sound pressure of recording

*D. Description of how well it worked*

**IV. CONCLUSION**

* Summary
* Possible improvements
* Lessons learned

**References**