Musical Tuner

Michael McNamara and Davey Proctor

Introduction to Intonation

Music, like all sound, is physically an oscillating pressure wave that propagates through a medium and is audible by our ears. A musical note, however, has many harmonic properties that allows people to identify and classify it. Some of these qualities of a note's waveform include timbre, attack, and voice, which all give different musical instruments distinct sounds. One of the most basic properties of a note is pitch, which is what determines the actual note on a chromatic scale people will hear (see Figure 1).

Musical tuners measure a note's pitch and compare it to a standard of intonation, returning whether the note was sharp or flat and allowing musicians to calibrate their instruments in different environments. This tuning especially matters in ensemble settings, because if two musicians that are out of tune play the same note, the difference between their frequencies will create interleavings and sound terrible. Tuning remains important even when alone, however, because significant differences in tone are palpable to our ears, and poor intonation will sound bad to any audience. Electronic tuners have been around for many decades: many physical and digital tuners use a microphone to measure pitch, but some plug into an audio jack or clip on to the instrument directly.

From Pressure to Pitch

We can measure sound with a microphone, which is essentially a low-latency pressure sensor for air waves. To sense pitch, we sample from an analogue microphone pin to produce a discrete time series of pressures which represent a sound wave.

But how do we go from a discrete time-series representation of a wave to statistical summary values such as its frequency? The answer lies in reconstructing the wave from its samples and deconstructing this data into a Fourier series (see Figure 2). An accurate representation of a continuous wave can be reconstructed from discrete samples as long as it those samples occur at a minimum of twice the waveform's highest frequency harmonic. A Fourier series is a representation of a periodic function as a sum of sines and cosines: any periodic function can be represented this way as long as it has a finite number of oscillations and discontinuities per cycle.

Using a Fast Fourier Transform (FFT) library, we compute the pitch as the frequency between the major peaks of the waveform fitted to the samples. The Fast Fourier Transform is a divide-and-conquer algorithm, reducing the complexity in finding the Fourier series transformation (and thus the burden on the Arduino) from O(n^2) in the naive application of the definition to O(n*log(n)), where n is the number of wave samples.

Music Represented Mathematically

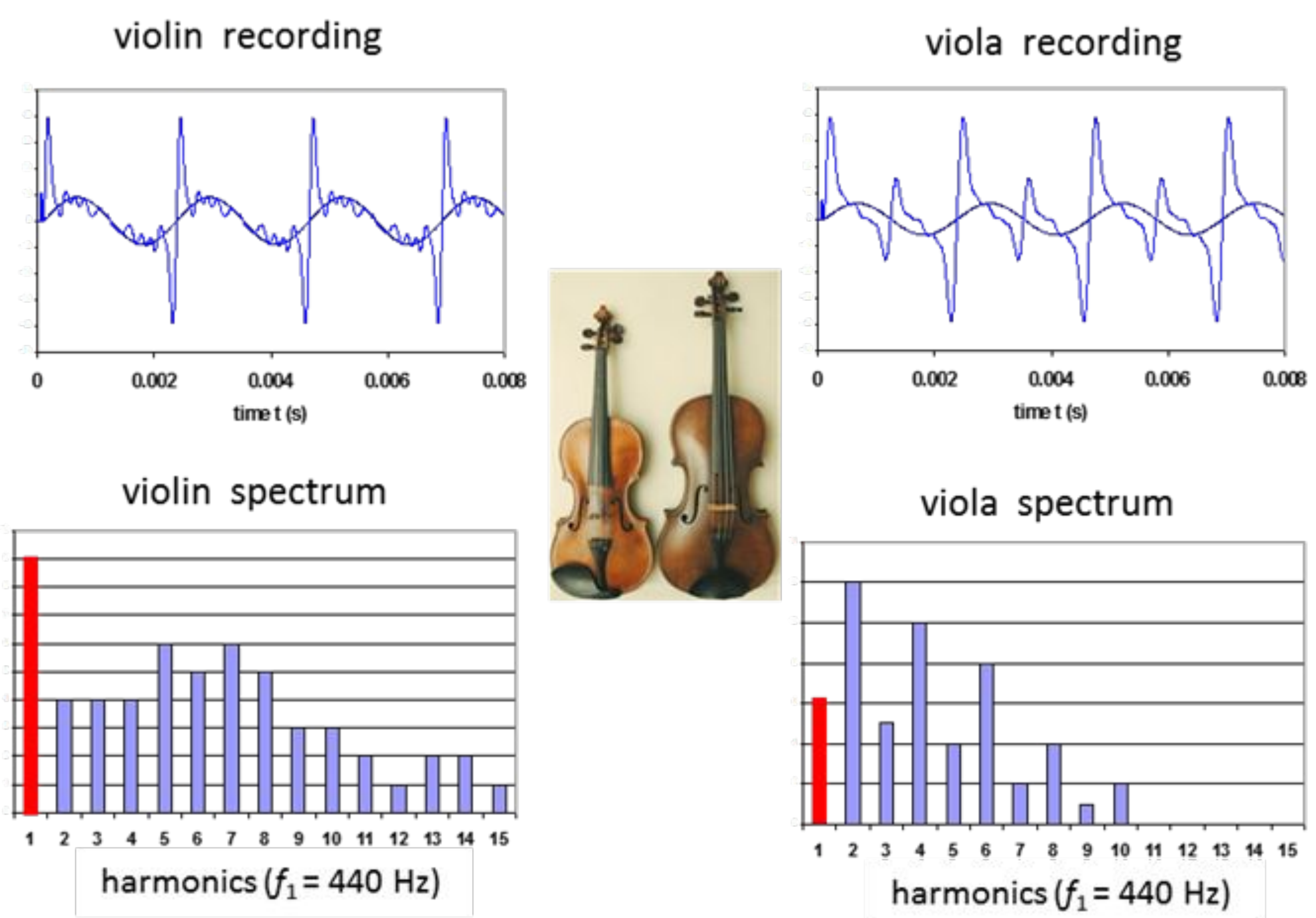

Figure 1: Note A (440 Hz) played on violin and viola. The violin has a "richer" sound due to higher harmonics. Source: http://www.physics.usyd.edu.au/teach_res/hsp/sp/mod31/m31_strings.htm

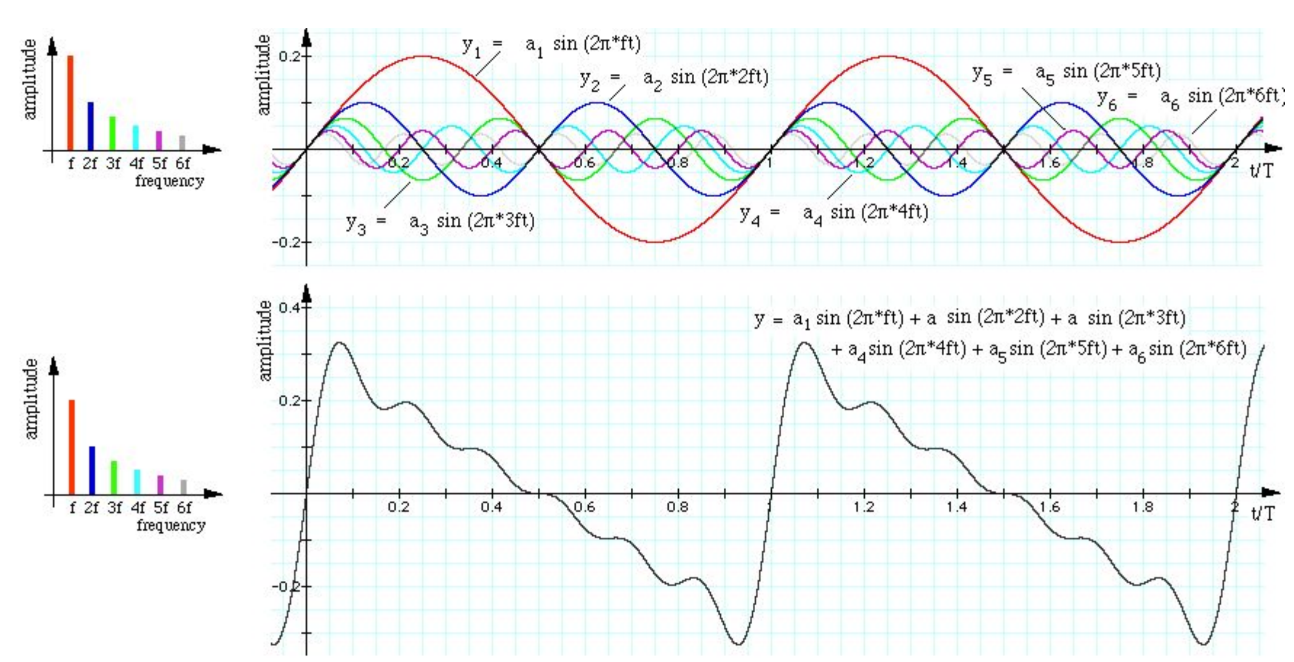

Figure 2: Fourier transformation: example periodic function represented as a sum of sinusoids. Source: https://newt.phys.unsw.edu.au/jw/musical-sounds-musical-instruments.html

System Design

Our system supports two types of user input: sound perceived by the microphone and button presses that trigger reference pitches to be played. The speaker plays the reference pitch as long as the button is held down. The musician can gauge their intonation by ear with the reference pitch, supplementing the feedback provided by the LEDs -- relative to a note, blue is low, green is in tune, and red is high -- and the serial monitor -- this shows more specific info on pitch and tuning.

To implement this, a polling model continuously cycles through reading buttons and sampling pitch. As long as a button is held low, in an inner loop we'll play the corresponding reference pitch, which will be either A (440Hz), A# (466Hz), C (523Hz), and F (698Hz). During this inner loop, a square wave with the desired frequency is sent to the speaker via the Arduino's built-in tone() function.

If no buttons are held down, we cycle through one sample of microphone reads and compute via FFT the pitch, turning on the a single LED and outputting pitch & tone information to the serial monitor.

Hardware Integration

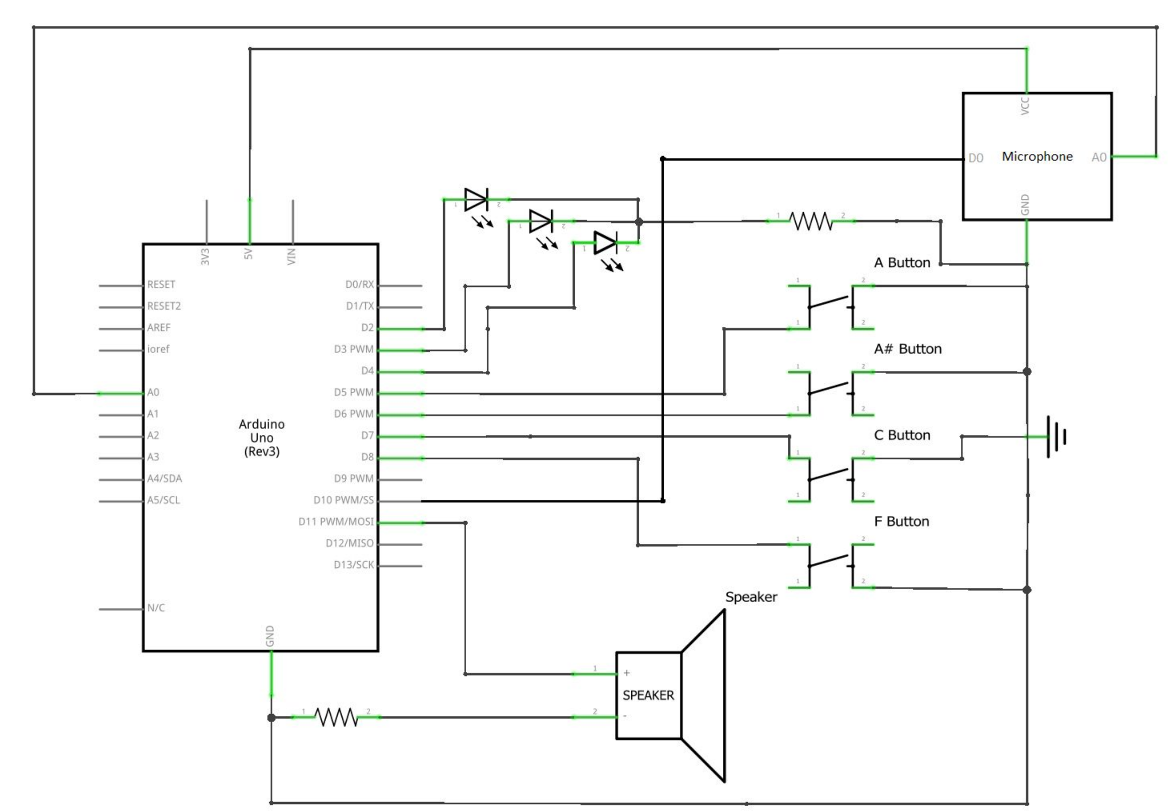

Figure 3: Circuit diagram for the tuner we created. The microphone and four buttons provide user input, and the speaker plays reference pitches when queued. Only one reference pitch can be played at a time.