Computer Identification of Instrument and Pitch Based on Harmonic Spectra

TATE THOMAS[1] AND ROSIE DURLAND[2,*]

Department of Physics, Utah Valley University, 800 W University Pkwy, Orem, UT 84058, USA
[*]*11026165@uvu.edu*

Abstract:

1. **Introduction**: Timbre is the perceived sound quality of a note, sometimes referred to as tone color. The great differences in timbre of musical instruments are easy to observe, but difficult to quantify. One way to approximate this quality is by looking at the harmonic signature of a note: a number summary of the amplitudes of the first several harmonics of that note played by a specific instrument. Through the use of harmonic signatures, we hoped to develop an algorithm that could identify specific instruments, and the pitches they played.

2. **Method**: We used two instruments, a guitar and a ukulele, and collected recordings of one string being played at a time. We collected 10 recordings for each single string. Then, by using a Fast Fourier Transform, we analyzed the harmonic amplitudes of the first 15 harmonics. Afterwards, we used a Linear Discrimination Analysis Classifier (LDAC) to determine what instrument and note was played in new recordings we took. These recordings included single and multiple notes played simultaneously from the different instruments.

3. **Results**: Our LDAC was extremely effective at identifying both the pitch and the instrument of new single-string plucks, achieving 100% accuracy. For two-string plucks, the LDAC accurately determined both pitches for 78.4% of combinations. For two-string plucks that included an open string frequency common to both instruments (E-330 Hz), the LDAC correctly identified the instruments of both notes played in 76.9% of combinations.

4. **Discussion**: Nearly all error in pitch identification can be attributed to a certain feature in our Find All Harmonics algorithm: the algorithm is designed to throw out harmonics that are multiples of a significant peak. Therefore, if two notes are sounded and one is a harmonic of the other, our current LDAC will necessarily struggle to identify both fundamental frequencies. This may be overcome through future research. Error in instrument identification only occurred when E-330 Hz on the Ukulele was played with other Ukulele strings simultaneously. In these cases, the LDAC predicted that E-330 Hz was coming from the guitar. The reasons for this are still unknown to the authors and require future research.

1. Introduction

The varying timbres of different instruments is a phenomenon that is readily observable - most people can identify that a trumpet sounds brassy, and a flute sounds light. The brain is able to group together relevant frequencies, and determine the overall sound of an instrument incredibly quickly [1]. However, this phenomenon is difficult to quantify, both visually and numerically. In 1999, McAdams et. al made progress on this by determining that the spectral envelope shape and the spectral flux of a sound file are the qualities of a sound file most closely related to human discrimination between timbres [2]. If it were possible to train an algorithm to discriminate instruments as quickly as the human mind is able to, we may gain the capability to split a sound file of multiple instruments into a sound file for each instrument, which would be incredibly useful. Our work aims to lay the groundwork in this objective.

The timbre of an instrument is difficult to quantify, but is closely related to the distribution of a given instrument's harmonics. Several researchers have previously used harmonic distributions to split a single sound file into multiple files [3] [4]. Primarily, these studies showed that it is possible to separate a sound file of multiple guitar strings sounding at once into a file for each string. We aim to broaden this research. Our goal is to have a computer algorithm identify what is consistently unique in the harmonic distributions for two separate instruments, and then use its own classifications to identify what instruments and notes were played in a recording. We hope that our research is eventually used to transpose music, or to separate a single sound file of multiple instruments into a file for each instrument.

Advancements in the field of sound file separation are critical to the progression of acoustical science. It is currently very time-intensive and difficult to edit the pitch of an instrument if it is included in the same sound file as another instrument. It is similarly difficult to isolate one voice in a recording of many voices. Analysis of the unique harmonic content of instruments is a critical stepping stone to identifying and separating one instrument from others in a sound file. Unique timbres of voices may be identifiable in a similar fashion.

Research in the realm of sound file separation has vast potential benefits to music transposition and editing, acoustic synthesis, and voice recognition. Countless companies and occupations rely on these technologies, such as recording studios, hearing aid manufacturers, instrument manufacturers, law enforcement, virtual reality companies, and even the Department of Defense. Advancements in this area could result in greater national security through advanced voice recognition, improved sound synthesis for persons wearing hearing aids, and increased efficiency of music recording and producing.

1.1. Theory

1.1.1. Harmonics

Harmonics arise from applying boundary conditions to waves to create standing waves. In our context, we will be using stringed instruments, which have two fixed endpoints at $x = 0$ and $x = L$, with L being the length of our string. In other situations, harmonics can appear in closed or opened tubes, differing from a string by requiring that any wave must peak at an open end. In both situations, standing waves occur due to the reflection of waves when they reach an endpoint. Because of our fixed endpoints, we must have 2 stationary nodes at each end, and n antinodes in between, where $n = 1, 2, \ldots,$ with $n + 1$ nodes in total.

The method for finding an equation that represents our situation involves finding a sine function to describe the n^{th} harmonic wave, then moving each wave through time in opposing directions, i.e. $\frac{1}{2}(\sin(k_n x - \omega_n t) + \sin(k_n x + \omega_n t))$, for some wave number k_n and angular frequency ω_n. We can then simplify this using trigonometric identities and find the values of k and ω. This result was first found centuries ago by Daniel Bernoulli in 1753, which is

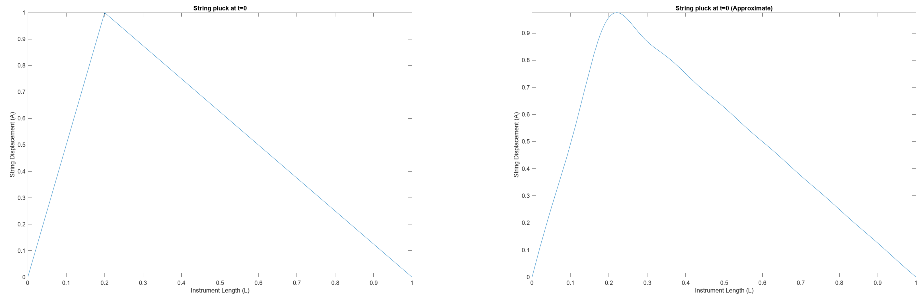
$$y_n(x,t) = a_n \sin\frac{n\pi x}{L} \cos\frac{n\pi ct}{L}, \qquad \text{when } \ 0 \le x \le L \tag{1}$$

where a_n is the n^{th} harmonics amplitude and c is the wave speed. This is given by

$$c = \sqrt{\frac{T}{\mu}}$$

where T is the tension in the string and μ is the mass per unit length [5]. If we say that $\omega_n = \frac{n\pi c}{L}$, then we can find the frequency of the n^{th} harmonic using $\omega_n = 2\pi f_n$. Thus, our equation is

$$f_n = \frac{\omega_n}{2\pi} = n\frac{c}{2L} = nf_0 \tag{2}$$



(a) $P(x,t)$ at $t = 0$.

(b) $Y_{15}(x,t)$ at $t = 0$.

Fig. 1. Two similar solutions of a plucked string at $x_0 = 0.2$, with length of string in units of L and amplitude in units of A: (a) exact solution, and (b) approximation

Not only does Equation (2) show that the fundamental frequency f_0 depends only on the length, density, and tension of a string, it also shows that the overtones are an integer multiple of it. This is very helpful for our description of timbre.

Yet, we still do not know the wave amplitudes a_n of each harmonic, which is largely what gives an instrument it's timbre. To start, we need to provide motivation on why we need these amplitudes to be the specific values that they are.

1.1.2. Plucked Strings

Let x_0 be the location where the string is plucked, and A be the amplitude of the displacement. Assuming the string is displaced by pulling on a single point, we can model our plucked string using this equation [5]:

$$P(x) = \begin{cases} \dfrac{Ax}{x_0} & 0 \le x \le x_0 \\ \dfrac{A(L-x)}{L-x_0} & x_0 < x \le L \end{cases} \tag{3}$$

A visualization of this is shown in Fig. 1a.

When we let go of the string when plucking it, waves will travel down the string in both directions and subsequently reflect off of the fixed endpoints, which we know will cause standing waves. If we extend our equation so that it is periodic, we would then be able to show a complete picture of our situation using $P(x,t) = \frac{1}{2}(P(x-ct) + P(x+ct))$, as done earlier. Our equation with the adjustment is

$$P(x) = \begin{cases} \dfrac{Ax}{x_0} & 0 \le x \le x_0 \\ \dfrac{A(L-x)}{L-x_0} & x_0 < x \le 2L - x_0 \\ \dfrac{A(x-2L)}{x_0} & 2L - x_0 < x < 2L \end{cases} \tag{4}$$

which will now have the ability to oscillate through time. (Note that the string is still only defined on $0 \le x \le L$).

If our string was perfect and could stretch how we would like without any unusual effects, this would model our plucked string exactly. But, not only is Equation (4) difficult to work with, we don't get any useful information about the resulting harmonics (if any at all!). Thus, we would like to find a new equation that describes our old equation using trigonometric functions.

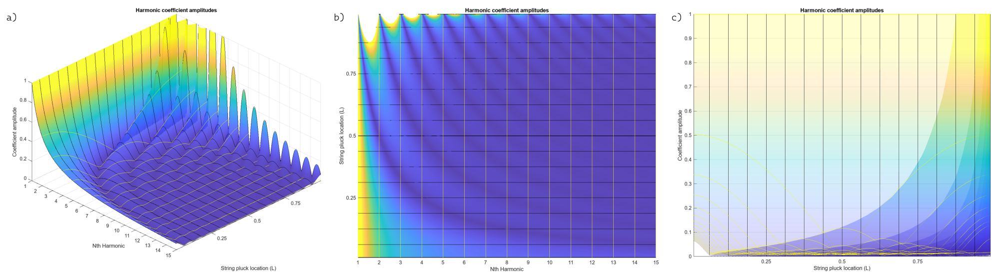


Fig. 2. Equation for a_n plotted as a function of the n^{th} harmonics and the location the string is plucked at x_0 at 3 different angles: (a) (45°, 45°), (b) (0°, 0°), and (c) (90°, 90°), defined as the angle from the top and the azimuthal angle, respectively.

This can be done through Fourier Analysis, although a little tedious. Luckily Perov, Johnson, and Perova-Mello already did this in 2016 [6]. Their resulting equation for the amplitudes of the Fourier components is

$$a_n = \frac{2A \sin \dfrac{n\pi x_0}{L}}{\dfrac{x_0}{L}(1 - \dfrac{x_0}{L})\pi^2 n^2} \tag{5}$$

This a_n is the same a_n as defined from before! If we use this equation for the amplitudes of the harmonic waves, we can use (1), and take the sum of the first m harmonics to approximate $P(x,t)$—that is,

$$Y_m(x,t) = \sum_{n=1}^{m} a_n \sin \frac{n\pi x}{L} \cos \frac{n\pi ct}{L} \approx P(x,t) \tag{6}$$

Fig. 1b shows this approximation at $m = 15$. As m grows larger, the series converges towards $P(x,t)$, which means

$$\lim_{m\to\infty} Y_m(x,t) = P(x,t) \tag{7}$$

Therefore, $Y_m(x,t)$ is the correct solution we're looking for.

Before we continue, there are some important things to mention about Equation (5), the graph of which is plotted on Figure 2. You may notice that the yellow lines of the n harmonics line up their inner nodes with the minima of the graph; this is no coincidence! It turns out that depending on where you pluck the string, you may line up with the nodes of some harmonics, which forbids those harmonics from existing. In general, if the location you pluck the string is around $\frac{L}{n}$, then the n^{th} harmonic will be either dampened or non-existent. The amplitudes also decrease as you go up in harmonics, as expected.

It may not be so obvious, though, that this graph is "symmetric" about $\frac{L}{2}$ (quotations since it clearly doesn't look like it). This is because this function is only technically defined for $n \in \mathbb{N}$—continuity is just shown for better visualization. You can see this for yourself if you follow the amplitudes of the yellow lines along L. A logical argument can also be made that the amplitudes of the harmonics should not be different on opposite sides due to symmetry; if you were to define your coordinate system to go positive in the $-x$ direction (i.e. you look at the string from the opposite side), you shouldn't expect to observe any different amplitudes compared to what they were before.

While we have finally found the harmonic spectra amplitudes that we will be studying, this isn't the complete picture. We still have assumed that the string will not deform as it is stretched and there will be no resulting tension (hence, the Young's modulus of the string equal to 0), along with the assumption that there will be no damping of the harmonic amplitudes as time goes on,

Fig. 3. Experiment set up

due to friction and air resistance. This results in a messy transcendental equation, which is best approximated using numerical methods. Furthermore, it turns out that some frequencies on a string may decay quicker than others—including cases where the fundamental decays faster than the overtones.

We won't go on to prove these and derive equations for them, but these factors will play a significant role in our research. This is due to the nature of our methods: we plan on performing Fast Fourier Transforms on recordings of notes being played. This means that the harmonic amplitudes will decay over the course of the recording, and the resulting FFT will only describe a sort of "average" amplitude of a harmonic over time. We could use a Short-time Fourier Transform, but that would have to be for future research.

2. Method

Our experimental set up features 3 distinct sections: the instruments, the recording, and the processing. The general idea is to take a sample recording of an instrument (or instruments), find the locations and amplitudes of the peaks in the harmonic distribution in the frequency domain, and input that data into a Linear Discriminant Analysis Classifier, where it will determine what instruments and notes were played. Fig. 3 demonstrates our set up.

2.1. Instruments

We used a Guitar and a Ukulele during our experiments. Each instrument was tuned up prior to recording samples to ensure consistent results. During the recordings, strings that were not being played were held down to prevent them from resonating due to potentially shared harmonic frequencies (Fig. 4). We also tried to minimize variation in the plucking of a string by performing a lateral plucking motion in-line with the rest of the strings (compared to a vertical pulling motion in-line with the instrument's cavity) directly above the cavity using only our fingertips.

2.2. Recording

We used a Yeti LE microphone set to the cardioid pickup pattern with the gain at a neutral setting. The cardioid pickup pattern—which only captures the sound directly in front of the microphone—was chosen to minimize external noise. The microphone was then placed at the edge of a desk angled down at around 45°, allowing us to consistently place our instruments about 1 foot away from the microphone by resting it on objects below. The microphone was then wired up to a laptop via USB.

2.3. Processing

The processing of each sample recording was an integral part our research. We discuss below the details of our program and algorithms, all done through Matlab.

Fig. 4. Unused strings being held down during sampling

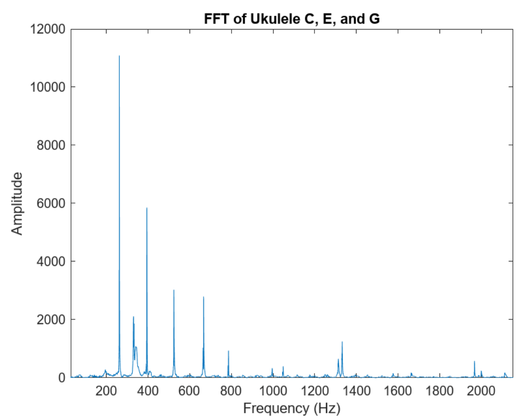


Fig. 5. Example plot of a FFT applied to a recording of a Ukulele playing notes C, E, and G simultaneously

2.3.1. Data Collection and Pre-formatting

Using Matlab's built in audiorecorder object, we collected data by executing the recordblocking() function, which would record for 3 seconds, and extracting the data through getaudiodata(). This returns a vector of amplitude data in the time domain, spaced out according to a specified frequency, chosen to be 44,100 Hz.

From here, we can perform a Fast Fourier Transform (FFT) on our data to analyze from the frequency domain (for real, positive values only), in which we expect to see spikes corresponding to our instrument's harmonics. An example of these spikes is shown in Fig. 5.

For us, we can easily point out the locations and amplitudes of each peak and determine what peaks may be significant by simply looking at it. Computers, on the other hand, don't have this luxury; they can only know what the amplitude of a data point is if they are looking directly at that element in the data vector. This severely complicates our analysis of harmonic distributions, as we must now formulate an algorithm to not only extract these peaks and determine if they are

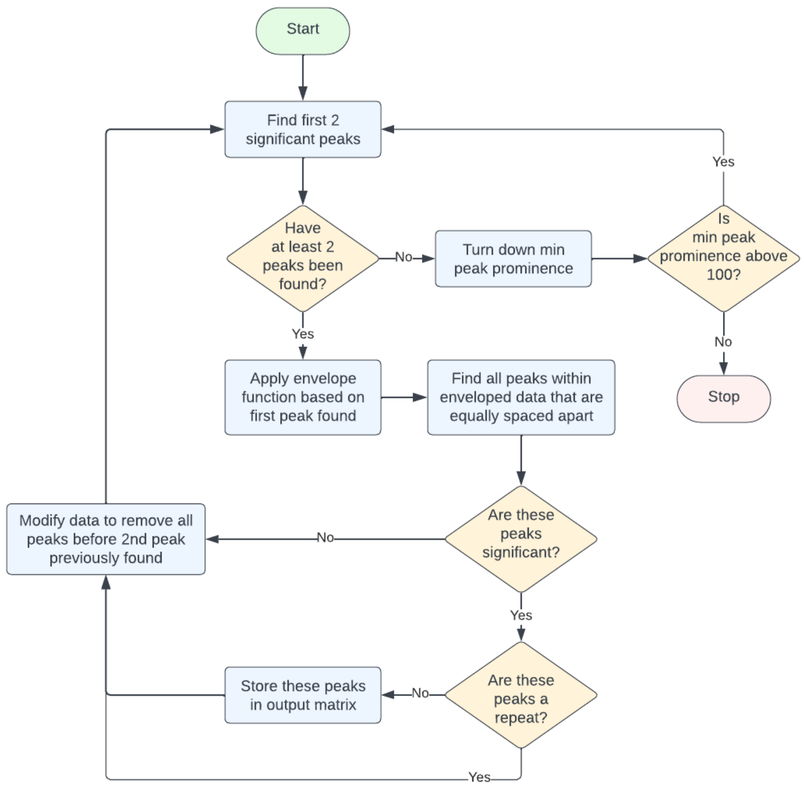


Fig. 6. Flowchart of the logic behind the Find All Harmonics algorithm

truly the harmonic peaks we are seeking after, but it must also work regardless of how many notes are played at once. Our solution to this problem is discussed in the next section.

2.3.2. The Find All Harmonics Algorithm

The idea behind the Find All Harmonics algorithm is as follows: find the first two significant peaks in the data, apply an envelope function to isolate the peaks that occur at multiples of the first peak's frequency, find all peaks within the enveloped data, determine if the resulting peaks are likely to be *new* harmonics, and repeat the process starting at the 2nd peak found in the first step until no more significant peaks are found. A graphical view of the logical flow of the algorithm can be found in Fig. 6.

The implementation of this utilizes Matlab's findpeaks() function, along with the different filters it provides. For step 1—finding the first 2 significant peaks—we used the filters, "*MinPeakProminence*", "*MinPeakHeight*", "*MinPeakDistance*", and "*NPeaks*". The first two are essential to determine if some peak is significant; "*MinPeakProminence*" determines if it is significant relative to the peaks surrounding it, and "*MinPeakHeight*" keeps small peaks from being considered, since the fundamental frequency will most likely be one of the largest, if not the largest, in its harmonic distribution. "*MinPeakProminence*" was chosen to start at 500 and was allowed to decrease by half if no peaks are found (to account for potentially quiet data), but we restricted it from going below a minimum value of 100. "*MinPeakHeight*" was chosen to be 400 (the units were unknown to us, but it corresponds to the volume of a recording). As for

"*MinPeakDistance*" and "*NPeaks*", the former is chosen to keep peaks from appearing too close to each other, while the latter is chosen to keep runtime at a minimum. The values for these were 25 and 2, respectively. The values for each of these filters, along with the filters themselves, were arrived at through experimentation to find values that work best for our purposes.

After finding those first 2 peaks, we can then create our envelope function to isolate the harmonic peaks. First, let f_0 denote the fundamental frequency (or the location of the first peak found). We would like to find a function that satisfies some of these basic requirements: 1) it is periodic, 2) it maxes out at multiples of f_0, and 3) it dampens peaks that fall just outside of those maxima, while eliminating most peaks in between. The simplest function that satisfies each of those requirements is

$$\xi(x) = \cos^n\left(\frac{2\pi}{f_0}x\right), \qquad \text{where } \; n = 2\left\lfloor \frac{x}{200} \right\rfloor + 1 \tag{8}$$

An important feature of this equation is the exponent n. First, notice that n is an odd number; this is to preserve the negative values of the function. Second, as x surpasses factors of 200, n incrementally increases; this allows the cosine function keep a skinny peak as x grows larger (note that the width of the cosine peak still does grow, though, to account for errors in larger frequencies). We can now take our envelope function and apply it to our dataset y to get a new "enveloped" dataset y', that is,

$$y' = y\,\xi(x)\,\Xi(x)\,, \qquad \text{with } \; \Xi(x) = \begin{cases} 1 & \xi(x) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Our binary factor Ξ eliminates peaks in between maxima to avoid any interference they may cause (recall requirement 3).

We can now use y' in a new search for *all* peaks by using findpeaks() again. This time, we will just be using the filter, "*MinPeakDistance*", set to the location of the first peak found earlier subtracted by 25 (to give some wiggle room for error). This will give us a set of peaks that may have spacing like that of a harmonic distribution. Let this set of peaks be denoted by f. Our next step in the algorithm is to determine if this set of peaks is actually a new harmonic distribution or not. We will call this "significance testing".

Our significance testing begins with quantifying how far off each element in f is from what it is expected to be, i.e. our error σ. The expected value for f_i is equal to $f_0(i+1)$, where $i = 0, 1, 2, \ldots, k-1$, with k being the total amount of harmonics used in our analysis (which we chose to be $k = 15$). Thus, the equation for our (percent) error is

$$\sigma_i = \left| \frac{f_i - f_0(i+1)}{f_0} \right| \tag{10}$$

Our criteria for significance, to begin with, is then

$$\text{significant if:} \qquad \forall\, i \in \mathbb{N}\,(\sigma_i < 0.1 \;\;\wedge\;\; i \leq 7)$$

or in other words, each error value σ in the first 7 peaks after the first must be below 0.1 (Note that 7 was chosen based on our findings of harmonic distributions of our two instruments; the 8th harmonic was the cutoff harmonic where some amplitudes became too low to reliably test).

This weeds out a majority of insignificant sets of peaks. It does not, however, account for duplicate harmonics of ones previously found. Let H be the matrix of previously found harmonics, with m rows of different harmonic distributions and k columns of harmonics (this is the same k from earlier). Then f is a duplicate of row μ's harmonics, starting at harmonic a, if

$$f_i = H_{\mu j} \qquad \text{where } \; j = (a+1)(i+1) \tag{11}$$

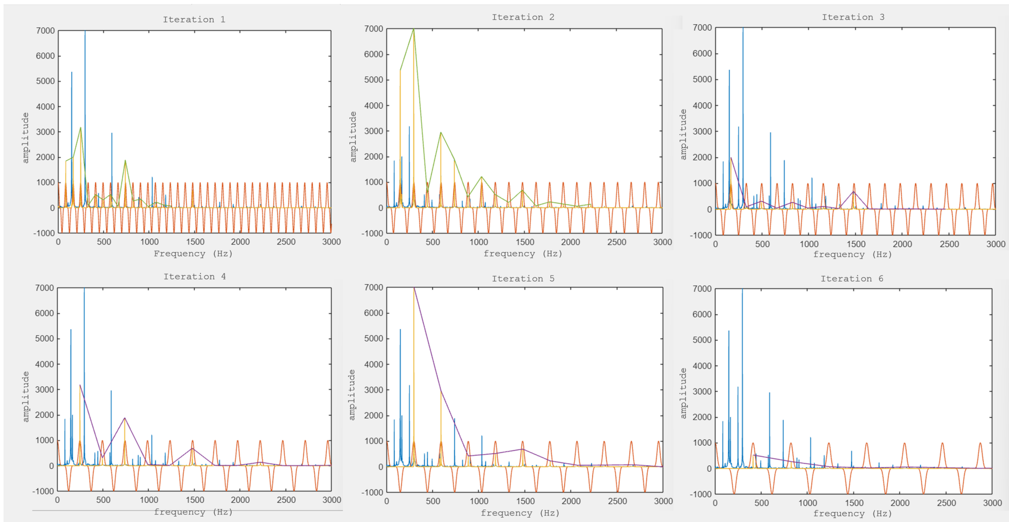


Fig. 7. First 6 iterations of the Find All Harmonics algorithm running on a sample of a Guitar playing strings E-82 and D-147. The blue lines are the peaks from the FFT, the orange line is the envelope function, the yellow lines are the peaks being used after the envelope function was applied, the purple line across the peaks is the current peaks being tested for significance, and the green line shows that those peaks displayed significance and can be determined to be a harmonic distribution.

for $0 \leq \mu \leq m-1$ and all $i \leq \lfloor \frac{k}{a} \rfloor$. This step is skipped if no harmonics have been found yet.

Realistically, as i increases, they may become more out of sync due to a more forgiving envelope function, making it appear as if it isn't actually a duplicate. To get around this, we only tested $i = 0$ and 1, and added a criteria that $y(f_i) \geq 1$ for $0 \leq i \leq 7$. In other words, each of the first 8 harmonics in our new set of peaks must have an amplitude greater than or equal to 1. This results from the fact that larger frequencies will result in smaller amplitudes, so duplicate harmonics will likely have some amplitudes less than 1.

This concludes our significance testing. If our new set of peaks pass all of those requirements, it is more than likely that it is a new set of harmonics. In that case, we will create a new row in H and store our new harmonics in row m. We finally loop back to the top and start the process all over again for the second peak found from the first step, removing all peaks that appear before the second peak (to keep the process moving forward). This matrix H will ultimately be returned from the function once no more significant peaks are found, along with another matrix representing the amplitudes at those harmonics.

The result of this algorithm displays a "sweeping" motion of the envelope function through the data as it tests each significant peak (Fig. 7). Notice how the algorithm correctly detects E-82 and D-147 on the first 2 iterations, and subsequent iterations are just repeats of the harmonics of those notes, so they are being ignored.

While this algorithm has provided much success, it isn't perfect, as we will discuss later. Still, it has been the most reliable method we have found during our research.

2.3.3. Data Collection

For our data collection, we sampled each string 10 times, for both instruments. Since the Find All Harmonics algorithm isn't perfect, we manually picked out what set of harmonics was the correct one after it found some candidates. The amplitudes of the first 15 harmonics from the samples were then saved under their respective instruments and notes.

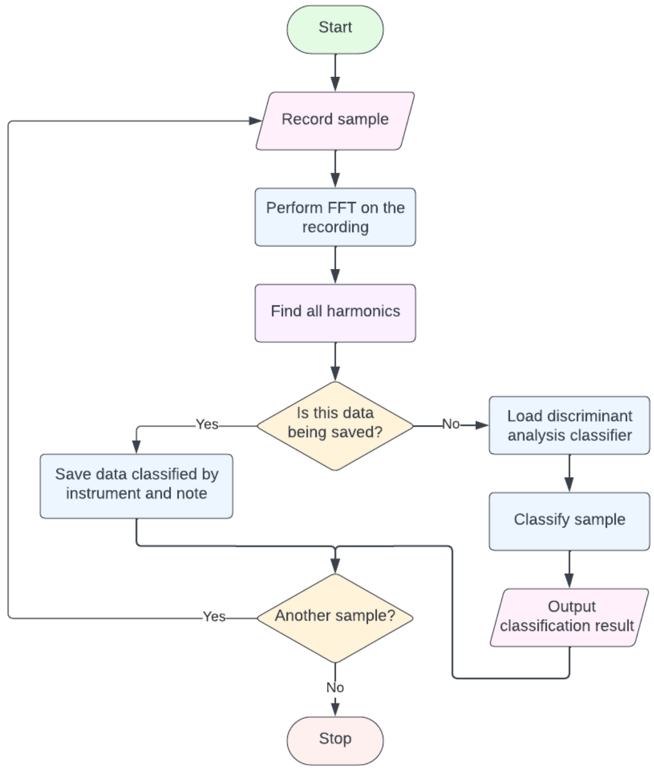


Fig. 8. Flowchart of the logic behind our experiment.

2.3.4. Linear Discriminant Analysis Classification

After collecting all 100 samples, we fed it into a Linear Discriminant Analysis Classifier. We formatted the data with the fundamental frequency first, and amplitudes of harmonics 2-15 divided by the harmonic 1's amplitude. This is to normalize the data input to remove dependence on the volume of an input, with harmonic 1 excluded since it will always be equal to 1, and will provide no use in classification. We also provided another set of numbers representing the index of a note for an instrument. This will be what the classification model will be outputting, which will allow us to correlate the number with an actual prediction.

This concludes our discussion on the processing side of our experiment. Our overall program is visualized in Fig. 8. Our results are discussed below.

3. Results

The primary objective of our research was to explore the answers to two questions:

1. Can our Linear Discriminant Analysis Classifier (LDAC) correctly identify the **pitch** of open guitar and ukulele strings?
2. Can our LDAC correctly identify the **instrument** that a given harmonic signature came from?

We will look at the answers to these questions one at a time, after a disclosure and discussion of our original data set used for performing these tests.

3.1. Data Set Used for Testing

We took ten recordings of each of the open strings of the ukulele and guitar, and labeled each recording with an instrument and the fundamental pitch of the string in Hz, e.g. "Guitar B-247". We then combined the FFT harmonic distribution data from all ten takes of each string into a single chart. For simplified viewing, a single line connects the average amplitude for each harmonic in a given string's distribution, with dots above and below each vertex representing the relative amplitude values for individual takes. The harmonic signatures shown in Figures 9 and 10 feature a dip in the amplitude around the 5th harmonic for both instruments, likely because there is a node for the 5th harmonic near the typical area of pluck, and as such, that harmonic could not be significantly excited. These charts, shown in Figures 9 and 10, were fed to the algorithm as options for identification. It is worth noting that the harmonic with the loudest amplitude, according to our recording software, was not always the fundamental frequency. This was a surprise to the authors. Significant time and tests were spent investigating this phenomenon, but the results were consistent, with no error in experimental setup that could be identified by the authors.

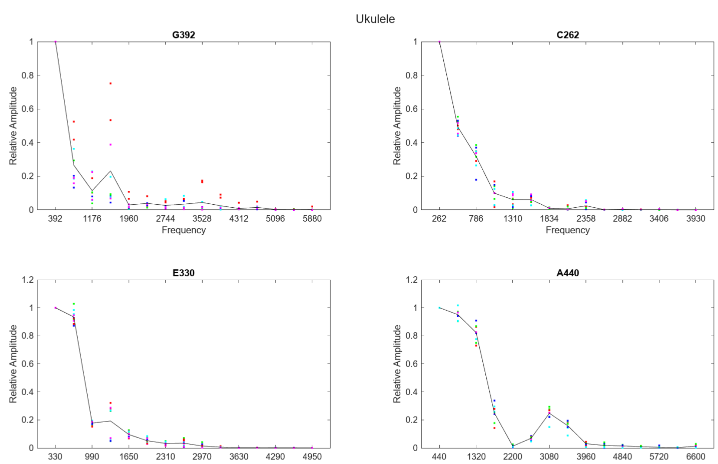


Fig. 9. Harmonic Signatures for Ukulele Open Strings

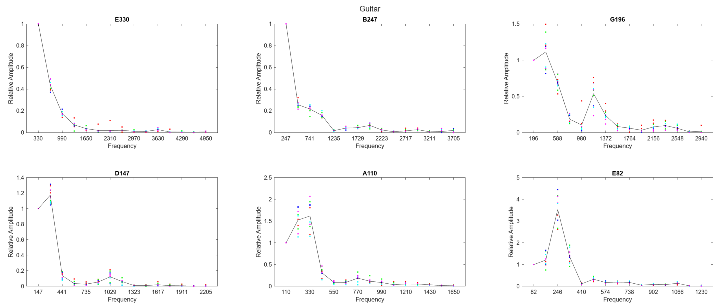


Fig. 10. Harmonic Signatures for Guitar Open Strings

3.2. Pitch Identification

3.2.1. Single-String Pluck

Guitar	
E-82	1
A-110	1
D-147	1
G-196	1
B-247	1
E-330	1

Ukulele	
C-262	1
E-330	1
G-392	1
A-440	1

Fig. 11. Portion Correct for Single-String Pitch Identification

The total amount of single-string tests performed was 30. Three tests were performed for each open string of the guitar and ukulele.

The LDAC was able to classify the pitches from single-string plucks with 100% accuracy for all strings on both Ukulele and Guitar.

3.2.2. Two-String Pluck

		GUITAR						UKULELE		
		E-82								
GUITAR	A-110	.2	A-110							
	D-147	1	.4	D-147						
	G-196	1	1	1	G-196					
	B-247	.33	1	1	1	B-247				
	E-330	0	0	1	1	1	E-330			
UKULELE	C-262	1	1	1	1	0	1	C-262		
	E-330	0	.50	1	1	1	1	1	E-330	
	G-392		1				1	1	1	G-392
	A-440		1				1	1	1	1

Fig. 12. Portion Correct for Two-String Pitch Identification

A total of 125 tests were performed across 37 string combinations. The mode amount of tests performed for each combination was three. For many of these combinations, the algorithm predicted the same thing for all three tests. If its predictions varied for the first three tests, we sometimes increased the number of tests for that combination, in order to improve our gauge of the accuracy of the LDAC. Figure 10 shows the portion of correct guesses from the LDAC to incorrect guesses from the LDAC; an incorrect guess could occur from getting one or both fundamental frequencies wrong. Gray areas represent combinations that weren't tested.

The LDAC correctly identified both pitches for 78.4% of the combinations tested.

3.3. Instrument Identification

We gave the LDAC ten specific categories of data. Each of these categories was labeled with an open-string fundamental frequency, as well as an instrument. These were, subsequently, the only labels that the LDAC was capable of using when classifying new data. This meant that for most string classifications, if the LDAC correctly identified the pitch, it automatically correctly identified the instrument as well. However, the guitar and ukulele do share one open-string pitch: E-330 Hz. It follows that the only cases where the LDAC could make an instrument identification error were classifications on tests that included the fundamental frequency 330 Hz. We turn to such cases below.

3.3.1. Single-String Pluck

Guitar	
E-330	1

Ukulele	
E-330	1

Fig. 13. Portion Correct for Single-String Instrument Identification

Six total single-pluck instrument identification tests were performed, three for each string. The LDAC correctly identified the instrument being played with 100% accuracy for single-string plucks of E-330 Hz on both the guitar and the ukulele.

3.3.2. Two-String Pluck

	Guitar					Ukulele		
	E-82	A-110	D-147	G-196	B-247	C-262	G-392	A-440
Guitar E-330			1	1	1	1	1	1
Ukulele E-330		1	1	1	1	0	0	0

Fig. 14. Portion Correct for Two-String Instrument Identification, Given That E-330 Was Correctly Identified

Because the Find All Harmonics algorithm (FAHA) is imperfect, there were cases of E-330 Hz being played on the guitar or ukulele in which FAHA did not correctly identify 330 Hz as a fundamental harmonic being played. Therefore, we have only considered the LDAC to be performing instrument identification on cases where 330 Hz has been correctly identified as a fundamental frequency, and the LDAC must choose which instrument this harmonic signature most resembles. A total of 46 two-string tests were performed that fit this description. Empty boxes represent string combinations in which FAHA never identified 330 Hz as a fundamental string frequency, so the LDAC did not perform instrument identification. This is a flaw in the FAHA coding, rather than in the LDAC.

In 76.9% of combinations involving the instrument identification of E-330 Hz, the LDAC accurately identified which instrument was playing that frequency.

4. Discussion

4.1. Pitch Identification

The success of our model in single-string pitch classification is promising. With a larger database, it may be possible for a similar algorithm to be able to classify many more frequencies, such as the frequencies of the piano strings.

Classifying two pitches at once proves more challenging. One key feature of the FAHA is that it will throw out peaks of significant amplitude if it finds that these peaks are mathematical multiples (or harmonics) of some lower frequency peak. This means that if two notes are played at once, and one is the harmonic of another, that the higher of the two will be thrown out as a possible fundamental frequency, because it is an exact or near multiple of the other detected fundamental frequency. Below is a figure highlighting key frequencies where this could possibly occur (Fig. 11).

Instrument	Fundamental	2nd Harmonic	3rd Harmonic	4th Harmonic
Guitar	E 82.4	164.8	247.2	329.6
Guitar	A 110	220	330	440
Guitar	D 146.8	293.6	440.4	587.2
Guitar	G 196	392	588	784
Guitar	B 246.9	493.8	740.7	987.6
Ukelele	C 261.6	523.2	784.8	1046.4
U/G	E 329.6	659.2	988.8	1318.4
Ukulele	G 392	784	1176	1568
Ukulele	A 440	880	1320	1760

Fig. 15. First Four Harmonic Frequencies of the Open Strings of Guitar and Ukulele (Hz)

If we map these possible "error zones" onto our data, we find that nearly all of the errors in classification can be described by this exact type of error.

		GUITAR						UKULELE		
		E-82								
GUITAR	A-110	.2	A-110							
GUITAR	D-147	1	.4	D-147						
GUITAR	G-196	1	1	1	G-196					
GUITAR	B-247	.33	1	1	1	B-247				
GUITAR	E-330	0	0	1	1	1	E-330			
UKULELE	C-262	1	1	1	1	0*	1	C-262		
UKULELE	E-330	0	.50	1	1	1	1	1	E-330	
UKULELE	G-392		1				1	1	1	G-392
UKULELE	A-440		1				1	1	1	1

Fig. 16. Portion Correct for Two-String Pitch Identification

Pink: Notes that share 247 Hz as a harmonic frequency
Blue: Notes that share 330 Hz as a harmonic frequency
Yellow: Notes that share 392 Hz as a harmonic frequency

Orange: Notes that share 440 Hz as a harmonic frequency
Empty: No data taken
*In this case, the FAHA skipped over B-247 Hz as a possible fundamental string frequency, for reasons unknown to the authors. This is the only error that cannot be explained by harmonic overlap.

It is promising that nearly all error in pitch identification can be attributed to a known flaw in the written algorithm. Future algorithms may be able to eliminate this flaw by requiring the computer to identify a new fundamental harmonic every time the relative amplitude of a harmonic is abnormally large, based on the original data set.

4.2. Instrument Identification

Our model's success at identifying the instrument correlated with a single-string pluck is promising. It follows that a similar algorithm to the LDAC could be trained on a larger data set to identify more instruments by the harmonic signatures of their notes. Future research will need to be conducted on fretted rather than open strings.

As for instrument identification when two strings are plucked, our model was able to accurately identify both instruments in 76.9% of combinations that included E-330 Hz. All instrument identification errors occurred when the Ukulele E-330 string was played with other Ukulele strings. In these cases, E-330 Hz was classified by the LDAC as coming from the guitar. The reasons for over-prediction of guitar in this particular circumstance is unknown to the authors, and requires further research.

5. Conclusion

5.1. Summary of Findings

1. The LDAC was able to classify both the pitches and the instruments of single-plucked strings with 100% accuracy.
2. For two-string pitch identification, the LDAC correctly identified both pitches for 78.4% of combinations tested.
3. Almost all error in pitch identification can be attributed to the aspect of our algorithm that requires multiples of a fundamental harmonic to be thrown out as possible fundamental harmonics themselves. This resulted in pitch identification error for cases in which one of the two notes played was a harmonic (or frequency multiple) of the other note. Future research may be able to combat this issue.
4. In 76.9% of string combinations that involved the shared open string frequency of E-330 Hz, the LDAC accurately identified which instrument was playing E-330 Hz.
5. All instrument identification error occurred when the Ukulele E-330 string was played with one other Ukulele string. In these cases, the LDAC always classified E-330 Hz as being played by the guitar. The reasons for this error are unknown.

Acknowledgments.

The authors thank Dr. Brian Patchett for his assistance in procuring the microphone used in this research. We also acknowledge the help of Dr. Dustin Shipp on the building of our Find All Harmonics algorithm. This research was greatly furthered by their contributions.

Disclosures.

The authors declare no conflicts of interest.

References

[1] J. W. Gordon and J. M. Grey, "Perception of spectral modifications on orchestral instrument tones," Comput. Music. J. pp. 24–31 (1978).

[2] S. McAdams, J. W. Beauchamp, and S. Meneguzzi, "Discrimination of musical instrument sounds resynthesized with simplified spectrotemporal parameters," The J. Acoust. Soc. Am. **105**, 882–897 (1999).

[3] M. R. Every and J. E. Szymanski, "Separation of synchronous pitched notes by spectral filtering of harmonics," IEEE Trans. on Audio, Speech, Lang. Process. **14**, 1845–1856 (2006).

[4] D. Ramsay, T. Burke, D. Barry, and E. Coyle, "A novel fourier approach to guitar string separation," in *IET Irish Signals and Systems Conference,* (2011).

[5] R. L. Herman, *A Course in Mathematical Methods for Physicists* (CRC Press, 2013).

[6] P. Perov, W. Johnson, and N. Perova-Mello, "The physics of guitar string vibrations," Am. J. Phys. **84**, 38–43 (2016).