Development of a triple input musical instrument tuner using Yin algorithm

Arnie Gedanken Branta, Emanuel Marques Martins, Verônica Isabela Quandt, Solivan Arantes Valente[2]

Department of Electrical Engineering, Positivo University, Curitiba – Paraná, BRAZIL

***Abstract*— In music, it is important that the instruments sound harmonic and reproduce sound in a repeatable and universal manner. Humans have difficulty in distinguishing very similar sounds, therefore, a device capable of doing so is very useful. The tool used for this is called an instrument tuner. There is a huge variety of such devices in the market, usually capable of detecting sound and frequency from specific instruments, due to the way sound is produced. Musicians that play different instruments must own multiple tuners to suffice their needs. This article details the development of an optimized pitch detection device, running on an Android smartphone, capable of detecting and processing sound signals emitted by different kinds of instruments, with frequencies ranging from 27.5 Hz to 4186 Hz in a non-expensive manner.**

***Keywords*— Android, Instrument Tuner, Java, Musical Instruments, Smartphone.**

I. INTRODUCTION

One of the most important characteristics for the development of musical balance is pitch, which is a representation of the sound frequency. This is how musical notes were created, representing sounds in a universal and repeatable way.

Just as measuring instruments need calibration, musical instruments also need adjustments. The device responsible for assisting in this procedure is the tuner. According to [1], the tuning of musical instruments is as old as the musical scale. Nowadays, the tuning process is performed mostly through electronic tuners, which are small, light and have good accuracy.

Many of the electronic tuners available in the local market are limited to specific types of instruments. Some models are compatible with a wider range of instruments, but are very expensive.

According to [2], the human ear cannot detect frequency variations less than 0.5 Hz, and these variations influence how harmonic the final song will be. Due to the difficulty of the human ear to perceive slight frequency variations, there is a need to use specific tools for this.

The many ways in which instruments produce sound make it necessary to use different mechanisms to capture the played notes.

This article describes the development of a device that aims to unify in a single device the various ways of capturing acoustic signals emitted by instruments of different categories, helping musical groups to tune their instruments with a single tuner, instead of using specific tuners to each type of instrument, resulting in lower investment and greater portability.

II. THEORETICAL FRAMEWORK

The sound produced by most musical instruments is the result of a sum of integer multiple frequencies of a fundamental frequency. It is the fundamental that defines the note played and, therefore, it has the most relevance to a tuning device.

2.1 Related Work

There are several ways to identify the fundamental frequency of a given signal. One is to apply the Fast Fourier Transform (FFT) and look for the frequency with the greatest magnitude. However, to get the resolution needed for a tuner, this technique may require a large FFT size, that is, a large number of bins. The relationship between the sample rate and the number of bins requires that to increase the frequency resolution, it is necessary to reduce the sampling rate or increase the number of bins. The sample rate cannot be reduced below the Nyquist theorem limit, so the solution is to increase the number of bins.

Santos [3] performed comparison tests of FFT calculations on several Arduino models: Arduino Mega with ATmega 1280, Arduino Duemilanove with ATmega 168 and Arduino Duemilanove with ATmega 328. He concluded that ATmega168 can store 128 points for the FFT calculation, compared to 512 points in ATmega1280. This condition is only obtained if no other variables need to be stored or processed by the microcontroller, which is unrealistic, since a platform like Arduino is rarely used to perform this type of calculation without any other input or variable. Thus, he concluded that depending on the desired application, it is more feasible to use Arduino as a data logger and process information on a computer.

Becchi [4] developed an automatic acoustic-electric guitar tuning system. It used the Arduino Due platform, equipped with a 32-bit ARM processor to perform input data processing and then control motors to adjust the instrument tuning. Due to limitations of the board used in FFT calculations, the author was unable to perform accurate readings and adjustments. According to him, the maximum desired inaccuracy value for the device was 0.5 Hz. The value obtained, nevertheless, was approximately 4 Hz, far above the ideal. As a solution, the author used Arduino as a data logger and a computer for processing. The results obtained using MATLAB were quite satisfactory, since the processing power of a computer is much higher than that of Arduino.

Using a computer to tune a musical instrument is not a portable solution. Modern smartphones have sufficient processing power, provided the right algorithm is used, to perform pitch detection.

Cheveigné and Kawahara [5] developed an algorithm to perform fundamental frequency estimation in speech and musical sounds, called YIN. The algorithm is based on the autocorrelation method and adds further steps to the process to reduce errors, as shown in Fig. 1. These 6 steps, performed in sequence, cause the gross error percentage to decrease from 10%, if only the autocorrelation method is used, to approximately 0.50%.

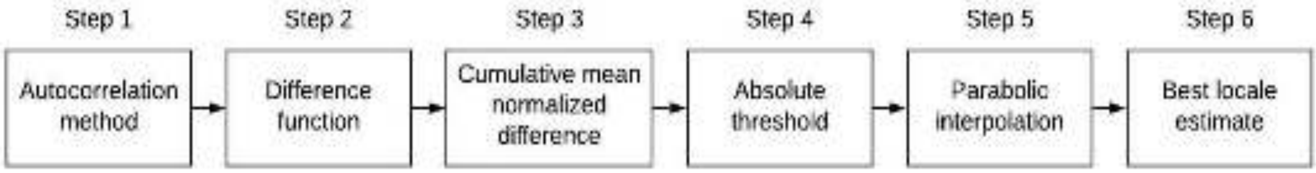

Fig. 1: YIN algorithm steps

Araujo and Trevisan [6] developed a system for acquiring and processing a signal from an electric guitar. Two different frequency detection algorithms were implemented and their results compared. The algorithms tested were YIN and McLeod. Four different tests were performed to evaluate which algorithm has the best accuracy and precision rate. The authors concluded that McLeod's method has a lower error rate in cases where only one note is played, slowly. On the other hand, the YIN algorithm performs better when several notes are played simultaneously or when the same note is played repeatedly over a short period of time. YIN was also more accurate, presenting a smaller deviation between readings than McLeod.

II.2 System Development

The developed device is a chromatic tuner that operates freely in the 12 semitones of each octave of the musical scale, ranging from A0 (27.5 Hz) to C8 (4186 Hz). It is intended to assist musicians in tuning various types of instruments through a single device, making the tuning process more compact and faster, as there will be no need to carry and handle different devices.

The system consists of two essential parts: the first is the hardware, that contains the electronic components responsible for capturing and converting the sound wave from the musical instrument to an electrical signal that can be read and processed by the smartphone. The second part is the application installed on a smartphone, which processes and compares the fundamental frequencies captured with the frequencies defined for each note, to identify which note is being played and, using the mobile phone screen, returns this information to the user. With this information, the user can make adjustments to the instrument until tuning is achieved.

There are three components responsible for capturing the sound, an electret microphone, a piezoelectric transducer and a line input. The user selects, in hardware, which of the three inputs to use, depending on the instrument to be tuned. This way, only one instrument can be tuned at a time, from one of the three inputs. The line input is basically a ¼ inch (6.35 mm) jack, which through a cable interconnects an instrument that has an integrated pickup to the tuner.

The piezoelectric transducer captures the sound wave through the mechanical vibration of the instrument. Since instruments tend to dissipate these string vibrations quickly, the signal duration are quite short. However, for instruments that do not have integrated pickups, tuning through mechanical contact is the best alternative, as the capture is not affected by background noise.

The electret microphone captures the sound without mechanical contact with the instrument. This form of sound capture is very prone to background noise. Therefore, it is necessary that the capture is performed in an environment with the least amount of noise possible.

This alternative, despite external factors, returns satisfactory results if used in the recommended scenario.

The analog signal is inserted into the phone via the ⅛ inch (3.5 mm) headphone connector, on the microphone pin. The power supply comes from the cell phone USB, through USB On-The-Go (OTG), which provides a 5 V DC power [7].

In addition to the three capture devices, there is also a signal treatment step, as the three inputs have very different amplitude levels, with output voltages incompatible with what the smartphone can receive in its headphone connection. After the signal injection into this input, the phone is responsible for converting it from analog to digital. This way, the signal becomes data, ready for use in the software. The system overview can be seen in Fig. 2.

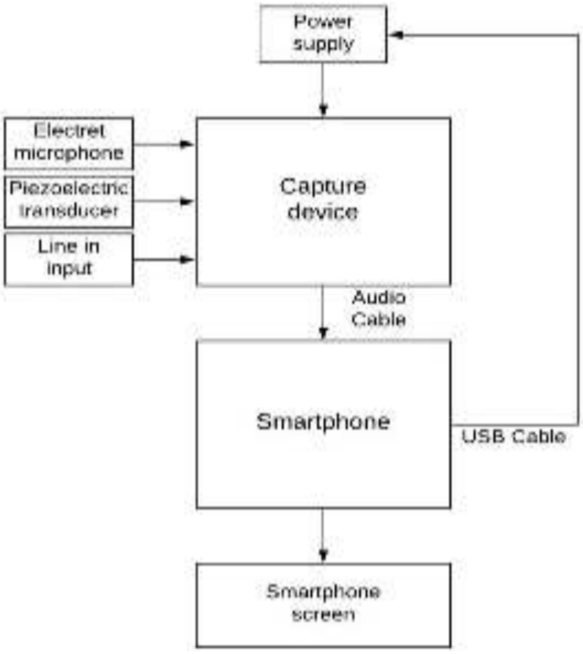

Fig. 2: System overview

The processing of the signal obtained through the hardware is done in an Android application. The first step of the software is responsible for identifying the fundamental frequency. This is accomplished through an implementation of the YIN algorithm by the TarsosDSP library [8]. In this step the sampling and buffer parameters are defined. For this system, a sample rate of 44.1 kHz and a buffer size of 22050 were quite sufficient.

After identifying the fundamental frequency, the algorithm compares this obtained data with the musical notes frequencies, which are calculated through (1) [2].

$$f = 440 . 2^{n/12} \quad (1)$$

Where n is the number of semitones above or below A4 (La in the 4th octave), tuned to 440 Hz. This way, it is possible to calculate the fundamental frequency of any musical note. The purpose of this step is to recognize which note of the musical scale the identified fundamental frequency is. That is why it is necessary to calculate the frequencies of the musical notes.

In addition to the comparison, the deviation between the note being played and the correct pitch is also calculated. This deviation is given in *cents*, a logarithmic measure that divides one octave into 1200 cents. Since 12 semitones form one octave, there are 100 cents between adjacent semitones and 200 cents between adjacent tones. The deviation between two frequencies, one played and the other in perfect pitch, is given by (2) [9].

$$N = 1200 . \log_2 \left(\frac{f_2}{f_1} \right) \quad (2)$$

Where f_1 is the reference frequency (calculated through (1) and f_2 is the input frequency (identified by the YIN algorithm). The application flowchart can be seen in Fig. 3.

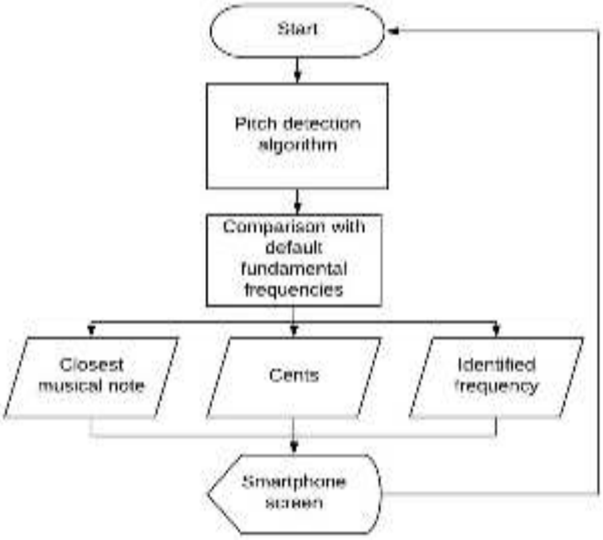
Start
Pitch detection algorithm
Comparison with default fundamental frequencies
Closest musical note
Cents
Identified frequency
Smartphone screen

Fig. 3: Application flowchart

The software was developed in Java, through Android Studio. The minimum Android version required for this app is 6.0 Marshmallow. The graphical interface was developed in XML and includes the following information: identified musical note and octave, fundamental frequency of the captured signal, input signal level, adjacent musical notes and a chromatic wheel, that shows the deviation, in cents, graphically. The interface can be seen in Fig. 4.

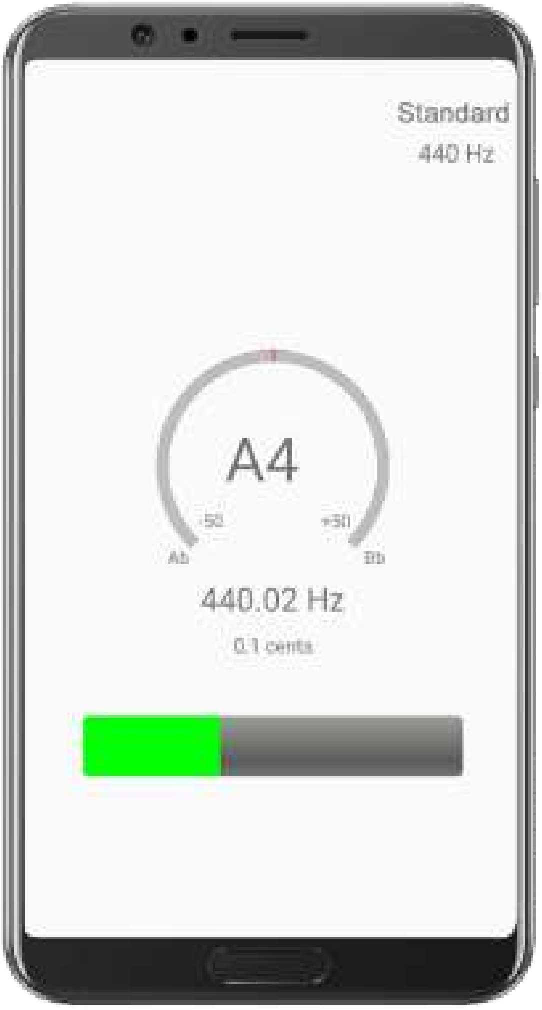

Fig. 4: Graphical interface of the application

III. RESULTS

To validate the effectiveness of the device developed, it was compared to other similar commercial tuners. As it has three different inputs, three different instruments were used in the tests, one for each input.

The device developed can be seen in Fig. 5. The parts are identified by numbers {1} to {9}. {1} is the input signal peak meter LED. It is an indication that the signal level is saturating. LEDs {2} and {3} indicate which of the inputs is selected. If LED {2} is on, line or piezoelectric input is selected. If LED {3} is on, the electret microphone input is selected. {4} is the switch that selects the input. The electret microphone is identified by {5}. There is a potentiometer for system input gain adjustment, identified by {6}. {7} is the 6.35 mm connector, which serves the line and piezoelectric inputs. {8} is the USB connector that transfers power from the phone to the tuner device. Lastly, {9} is the 3.5 mm connector that transfers the analog signal into the smartphone.

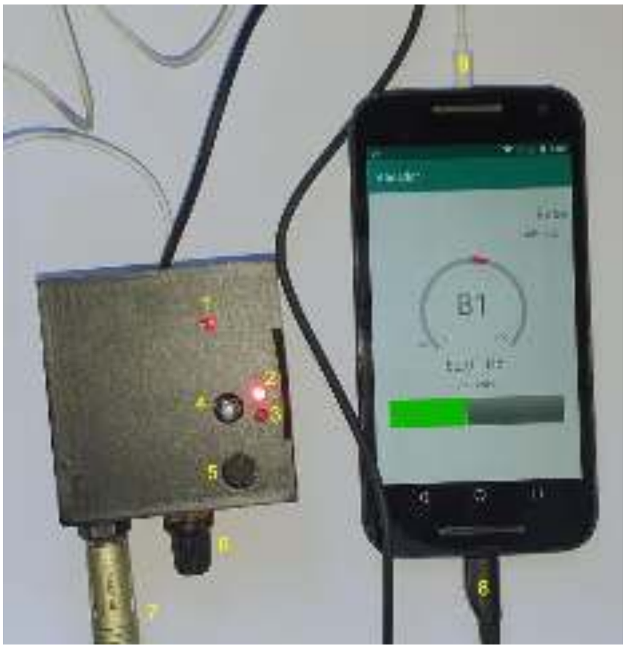

Fig. 5: Device developed in operation

The prototype has low cost and easily obtainable components. The estimated cost in electronic components was $8. The case was built in a 3D printer and cost $10.

The test for validating the line input was performed using an electric bass. The results were compared with three other commercial devices: Zoom B3 [10], Korg CA-1 [11] and Mooer Baby Tuner [12]. The comparison was made by playing an instrument string while connected to all tuners simultaneously. As it is a direct connection to the instrument, there is no interference between the signal emission and capture. This tuning method is best suited for instruments with built-in pickups.

The piezoelectric input was tested using a cello. The device developed was compared to the Aroma AT-200D [13] tuner. For this test, the Korg CM-200 [14] transducer was used with the device. The test was performed by coupling both tuners to the instrument bridge simultaneously. Then, the instrument was played and both readings were recorded.

The electret microphone input was tested using a soprano recorder (fipple flute). The comparison was made with the Korg CA-1 [11] tuner. The test was performed by placing both tuners next to each other and playing a note on the recorder. The Table 1 summarizes the tests results.

Table 1: Tests results

Line input				
Device	Tuner developed	Zoom B3	Korg CA-1	Mooer Baby Tuner
Fundamental frequency	98.03 Hz	N/A	N/A	N/A
Musical note	G	G	G	G
Octave	2	N/A	N/A	N/A
Deviation	0.6 cents	0	0	0
Piezoelectric transducer input				
Device	Tuner developed		Aroma AT-200D	
Fundamental frequency	65.34 Hz		N/A	
Musical note	C		C	
Octave	2		N/A	
Deviation	-1.8 cents		Between 0 and -5 cents	
Electret microphone input				
Device	Tuner developed		Korg CA-1	
Fundamental frequency	782.97 Hz		N/A	
Musical note	G		G	
Octave	5		N/A	
Deviation	-2.3 cents		Between 0 and -5 cents	

It is possible to notice that of all the tuners tested, the one described in this paper is the only one that informs the octave, fundamental frequency captured and the deviation in numerical format. Other devices report deviation graphically, with different scales. For this reason, when the deviation is close to zero, the result indicated is rounded. This is not a result of inaccuracy; it is just a feature of these devices.

IV. CONCLUSION

This paper presented the development of a universal tuner. The results presented prove the functionality, comparable to commercial solutions. In some respects, it presents more information than other devices tested. Some of this information is very relevant to less experienced musicians, such as the octave. Other information is very relevant for experienced musicians who want perfect tuning, such as precise deviation in decimal scale.

For those who want as few cables as possible, the USB connection can be a downside. For future improvement, the implementation of an internal battery system is suggested.

ACKNOWLEDGEMENTS

We are grateful for the opportunity and support given by the Positivo University teachers, in particular Solivan Arantes Valente and Verônica Isabela Quandt.

REFERENCES

[1] Barbour, J. M. (1951). Tuning and temperament: a historical survey. East Lansing: Michigan State College Press.

[2] Regeena, M. L., Saji, A. K. (2009, December). A digital guitar tuner. International journal of computer science and information security. Dubai, UAE.

[3] Santos, N. P. (2012, June). Arduino e o cálculo da FFT. Revista PROGRAMAR, 35th ed, pp.14-17. Retrieved from https://www.revista-programar.info/edicoes/edicao-35

[4] Becchi, N. G. (2017). Sistema automatizado para afinação de violão elétrico [Bachelors dissertation on the Internet]. Lajeado: Centro Universitário UNIVATES. Retrieved from https://www.univates.br/bdu/bitstream/10737/1664/1/2017NatanGabrielBecchi.PDF

[5] Cheveigné, A., Kawahara, H. (2002, April). YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, pp.1917-1930. Retrieved from http://audition.ens.fr/adc/pdf/2002_JASA_YIN.pdf

[6] Trevisan, C. D., Araujo, V. V. (2018). Electric guitar digitizer [Bachelors dissertation on the Internet]. Curitiba: Universidade Tecnológica Federal do Paraná. Retrieved from http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/10124/1/CT_DAELN_2018_1_03.pdf

[7] Maxim Integrated (2002, December). USB On-The-Go Basics: AN1822. Retrieved from https://www.maximintegrated.com/en/app-notes/index.mvp/id/1822

[8] Six, J., Cornelis, O., Leman, M. (2014). TarsosDSP, a Real-Time Audio Processing Framework in Java. Retrieved from https://github.com/JorenSix/TarsosDSP

[9] Roberts, G. E. (2016). From Music to Mathematics: Exploring the Connections. Baltimore: Johns Hopkins University Press.

[10] Zoom B3 (2019). Zoom B3 Bass Effects & Amp Simulator Pedal. Retrieved from: https://www.zoom-na.com/products/guitar-bass-effects/bass/zoom-b3-bass-effects-amp-simulator-pedal

[11] Korg CA-1 (2019). CA-1 – Chromatic Tuner. Retrieved from: https://www.korg.com/us/products/tuners/ca_1/

[12] Mooer Baby Tuner (2019). Baby Tuner – Shenzhen Mooer Audio Co., Ltd. Retrieved from: http://www.mooeraudio.com/product/Baby-Tuner--96.html

[13] Aroma AT-200D (2019). AT-200D Tuner. Retrieved from: http://www.aromamusic.cn/en/products-info30.html

[14] Korg CM-200 (2019). CM-200 – Contact Microphone. Retrieved from: https://www.korg.com/us/products/tuners/cm_200/