IV SIINTEC INTERNATIONAL SYMPOSIUM ON INNOVATION AND TECHNOLOGY
VIII PTI RESEARCH AND INNOVATION WORKSHOP
Modelling and Computational Simulation Applied to Industry

TUNING DEVICE FOR STRING MUSICAL INSTRUMENTS

Mateus dos Santos de Menezes[1] - mateusmenezes95@gmail.com
Élisson Ríller Souza de Oliveira[1] - ellison_09@hotmail.com
Gabriel da Silva Santos[1] - aec.gabrielsantos@gmail.com
Iure de Araújo Pinheiro[1] - iurepinheiro13@gmail.com
Daniel Oliveira de Almeida[1] - daniel.oa95@gmail.com
Lucas Cruz da Silva[2] - lucas.cs@fieb.org.br
[1]Centro Universitário Senai CIMATEC - Salvador, BA, Brazil
[2]Senai CIMATEC, Brazilian Institute of Robotics - Salvador, BA, Brazil

Abstract: The present work shows a guitar tuner prototype. With the use of a microphone and a microprocessor system was identified and showed in a display the fundamental vibration frequency of a determined string when played. Thus, was possible to know which was the musical note played and how far it was from the key tuning. In order to achieve this, digital signal processing techniques have been used, such as: Fast Fourier Transform (FFT), windowing and interpolation.

Keywords: Tuning; Fast Fourier Transform; Digital Signal Processing

1. INTRODUCTION

A reference system for musical notes, such as a scale, is crucial for composition, execution and reproduction of chords and notes in a musical piece. To tune a musical instrument, it is necessary to match the notes emitted by the instrument with this reference system.

Methods used for tuning such as determining the reference off one's memory or using a tuning fork depend on perception of the person carrying out the task and his or her ability to establish a true reference for the former and to track correctly the reference with the auditory system for the latter. Both methods are prone to errors, which could propagate to different instruments on a same tuning session. When employing electronic devices, the risk of failure in the task of tuning reduces, because the reference is clear to the user on a display.

2. THEORETICAL BACKGROUND

2.1 Music Theory

Human perception of different musical notes in string instruments is due to the fundamental frequency generated in their strings' vibration, as in the case of playing different notes in one musical scale or even having one musical note emitted in an octave interval. The western musical scale consists of twelve different notes:

- The natural notes, identified as C, D, E, F, G, A and B. This sequence repeats in a circular way, so that B is adjacent to C.
- The accidents, which happen between two consecutive notes, except for the interval between E and F and B and C. They are C# (D♭), D# (E♭), F#(G♭), G# (A♭) and A# (B♭).

In terms of frequency, equation 1 provides a way to obtain the ones for each musical note:

$$F_{n+a} = F_n \sqrt[12]{2^a} \tag{1}$$

where F_{n+a} is the frequency of the desired note, F_n the frequency of a reference note and $a \in \mathbb{N}, a = 1, 2, \ldots, 12$. This way, a set of 12 consecutive notes forms an octave, and it can be seen that the same note in two consecutive octaves have frequencies separated by a factor of 2.

The usual reference value is the 440 Hz A, referred to as A4. Musical notes in the range C-B around A4 also gain this index, and moving up or down on octaves increases or decreases this index, being a way to identify the musical notes frequency-wise.

2.2 Frequency Analysis

The Fourier Transform (FT, equation 2) is a mathematical method suitable for frequency analysis of vibrating strings signals.

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi\omega t}dt \quad (2)$$

where $X(\omega)$ is the signal representation in frequency domain and $x(t)$ is the signal representation in time domain. Although well defined theoretically in continuous time domain, since the operations in the current problem are executed by microprocessor systems, an alternative for discrete time domain is needed. The discrete counterpart of the FT is the Discrete Fourier Transform (DFT, equation 3).

$$X[m] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{mn}{N}} \quad (3)$$

where N is the total amount of points in the discrete-time signal $x[n]$, $X[m]$ is the frequency content and $m \in \mathbb{N}, m = 0, 1, 2, \ldots, N-1$.

The DFT bears a great computational cost to algorithms running it, due to the complex exponential multiplications. An algorithm focused on eliminating redundancies present in the DFT, called Fast Fourier Transform (FFT), is usually employed in digital signal processing units dedicated to calculation of frequency content. The results are the same as the DFT, with no approximations [1].

In order to avoid misinterpretation of the results, the frequency analysis of a discrete signal mus be executed according to Nyquist's theorem, which states that the maximum frequency on the signal must be at most half the sample frequency F_s [1]. The frequency indexes of $X[m]$ can be obtained with

$$X[f] = X[m\Delta f] \quad (4)$$

and

$$\Delta f = \frac{F_s}{N} \quad (5)$$

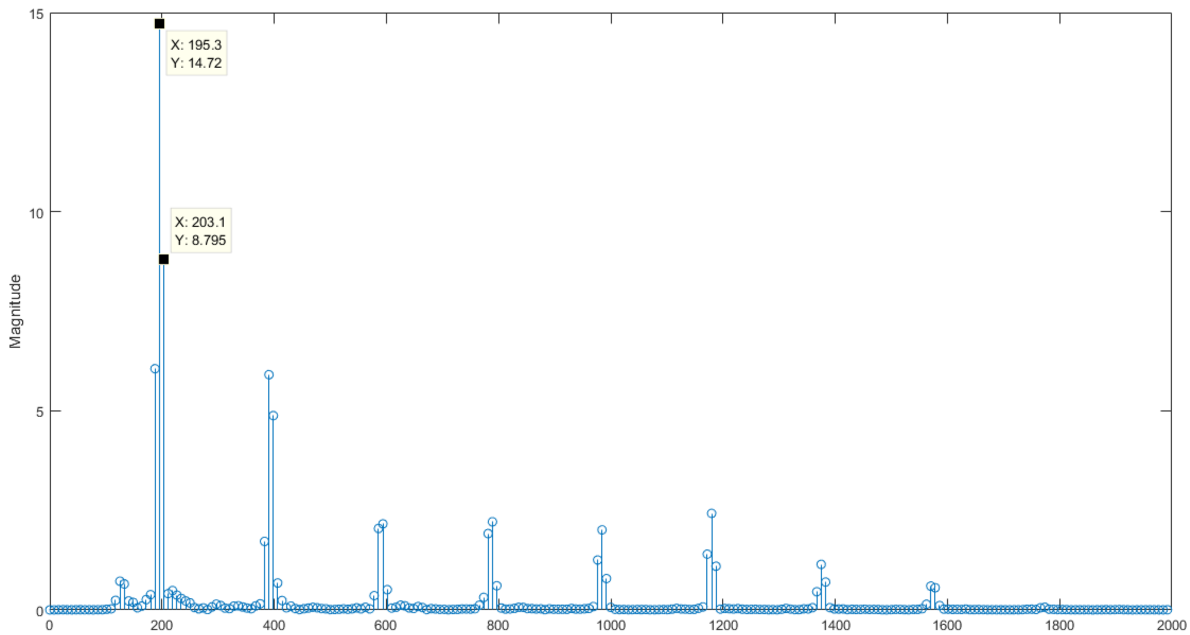
where Δf is referred to as spectral resolution.

If $x[n]$ is a periodic signal with an integer number of complete periods, then the frequency related to this period is multiple of Δf. Since most systems work with fixed F_s, is virtually not possible to have the spectral resolution matching the fundamental frequency of every note emitted by a musical instrument. This constraint result in an undesired effect in FFT result called *spectral leakage*, which leads to presence of spectral content on components that don't actually exist in the signal. Another consequence of the mismatch between fundamental frequency and spectral resolution is the fact that the former is not seen in the frequency indexes of $X[f]$.

In order to overcome the difficulty of not having the frequency of interest represented in $X[f]$, interpolation methods can be used. These methods estimate content not present in the DFT output by evaluating neighbour elements in the output vector. One example of such algorithm suitable for the present work is the one proposed by Grandke [2], which estimates the position of the highest magnitude content of a DFT output.

Figure 1 shows the frequency spectrum of a string vibrating in G4, obtained with FFT. There is a high magnitude component at 195.3 Hz and a neighbour component with slightly less magnitude at 203.1 Hz. For the spectrum shown in Figure 1, Grandke's method estimates the frequency of higher amplitude as being 199.4485 Hz, which is a better guess then the other values (G4 frequency is 197.7 Hz)

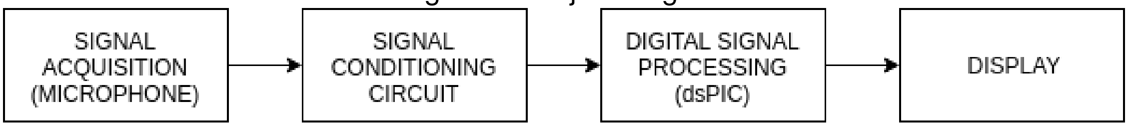
Figure 1- Frequency spectrum of G string



3. METHODOLOGY

Figure 2 shows the methodology employed in the tuner realization.

Figure 2- Project diagram



3.1 Components

All circuits were built in printed circuit boards (PCBs). For the signal conditioning stage, 4 operational amplifiers on single package of integrated circuit TLC274. The first amplifier acts as a voltage buffer, since audio transducers have medium to high output impedance. The second amplifier adds voltage level and positive gain to the signal to meet the microcontroller's requirements. The remaining 2 are used to implement the anti-aliasing filter shown in section 3.2.

The chosen microcontroller is Microchip's dsPIC33EP128MC204 [3]. Its hardware includes relevant peripherals to the problem, such as ADC, DMA (Direct Memory Access) and native support to FFT operations, with instructions such as MAC (Multiply and Accumulate).

3.2 Signal Acquisition and Conditioning

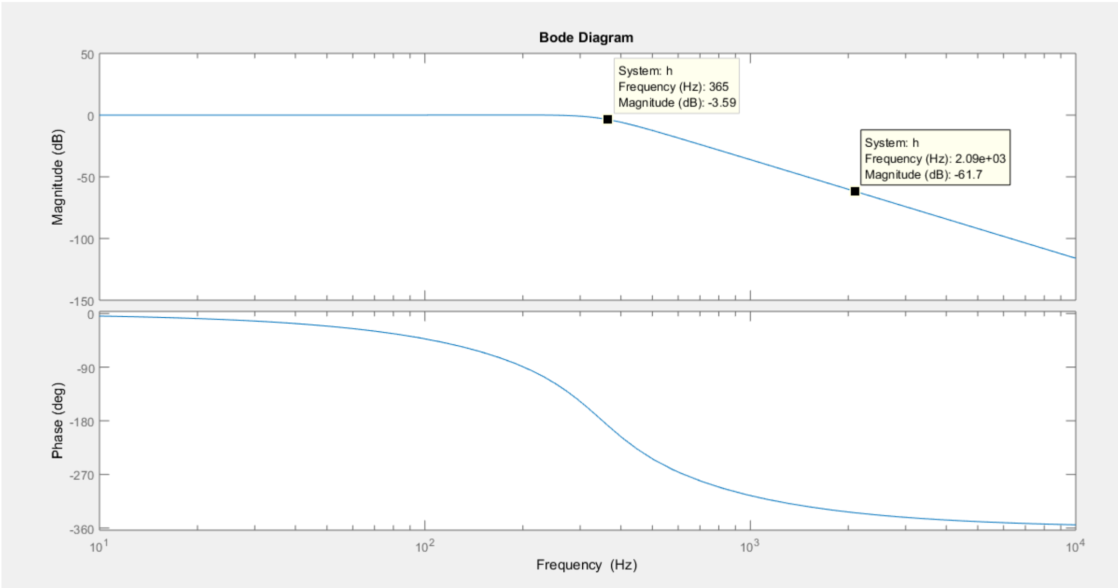
The electrical signal output obtained by a microphone or pickup is in continuous time domain in its nature. In order to have the microcontroller to process it, it must be converted to digital domain using the analog-to-digital converter (ADC) unit, and also shifted to a proper voltage level. The constraint imposed by Nyquist's theorem means that prior to sample and conversion, a low-pass filter must be used. This filter is often called anti-aliasing filter due to the effects (aliasing) caused in calculated spectrum by disregarding Nyquist's theorem [1].

Several approximation functions can be employed in filter realization, such as Butterworth, Bessel and Chebyshev. In this work, Butterworth realization was chosen since it doesn't produce ripple in pass band, resulting in flat response for the desired frequencies [4].

The resulting filter is shown in equation 6. It is of 4th order, with a -3 dB cutoff frequency in 360 Hz and stopband attenuation around -58 dB starting at 2 kHz. Its frequency response is shown in figure 3.

$$H(s) = \frac{1}{4.057\times10^{-14}s^4 + 2.383\times10^{-10}s^3 + 6.876\times10^{-7}s^2 + 1.161\times10^{-3}s + 1} \quad (6)$$

Figure 3- Frequency response of the anti-aliasing filter



The filter was implemented using operational amplifiers, as shown in figure 4.

3.3 Digital Signal Processing

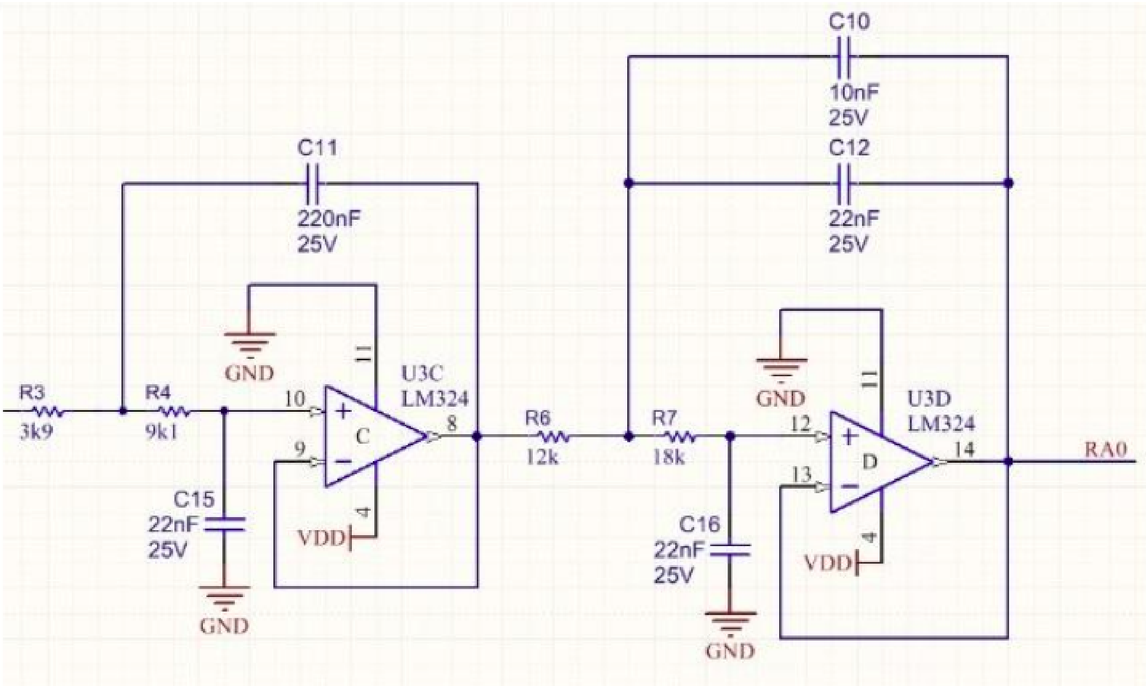
The algorithm for signal processing is depicted in figure 5. After startup configurations at the microcontroller, the sampling stage takes place. A total amount of 512 12-bit samples are acquired during the sampling and conversion step, at a sampling rate of 2 kHz. The DMA plays an important role in storing the samples into memory without requesting the CPU, granting efficiency to the process. The ping-pong buffer method in [5] was used in DMA operations.

Prior to FFT execution,the sample vector is downscaled by a factor of 2 to avoid saturation (following the manufacturer's recommended procedure), then multiplied in time domain by an window function (Hanning window) to reduce the effects of spectral leakage. Finally, zeros are added to the sample vector (zero-padding) to increase the spectral resolution, making a sample vector with $N = 1024$ points, resulting in spectral resolution of 1.95 Hz.

The search for the fundamental frequency is conducted by looking for the frequency content with higher magnitude. Grandke's method is applied to the magnitude spectrum of the sample vector. The frequency pointed by the interpolation method is elected as the actual note being played at the instrument. Once the frequency of the note is known, the system provides information at a display on how distant the current frequency is to a known musical note within the musical scale between C2 and B4.

ISSN: 2447-4215
SESSÃO TÉCNICA: ENGENHARIAS

Figure 4- Anti-aliasing filter



4. RESULTS AND DISCUSSION

The results for the tests are shown in table 1. The table shows musical notes often used as reference tuning in guitars. For each reference note, five samples were taken with frequency estimated for each of them and some statistical measures were generated compared with the reference frequency.

By inspecting table 1, it can be seen that the maximum relative error between the average estimated frequency and the reference note is of 1.13%. Considering the difference between 2 consecutive notes in the musical scale being the factor $\sqrt[12]{2} \simeq 1.059$, it means that the maximum error is less than a quarter the difference between 2 consecutive notes (around 6%).

5. CONCLUSION

This work presented a system that identifies a note being played by a string-based musical instrument. The system was built using analog electronics together with a microcontroller unit programmed with digital signal processing logic. The results show a good precision in the detection method, where the detection error was considerably less than the interval between consecutive notes on the musical scale.

Figure 5- Signal processing flowchart.

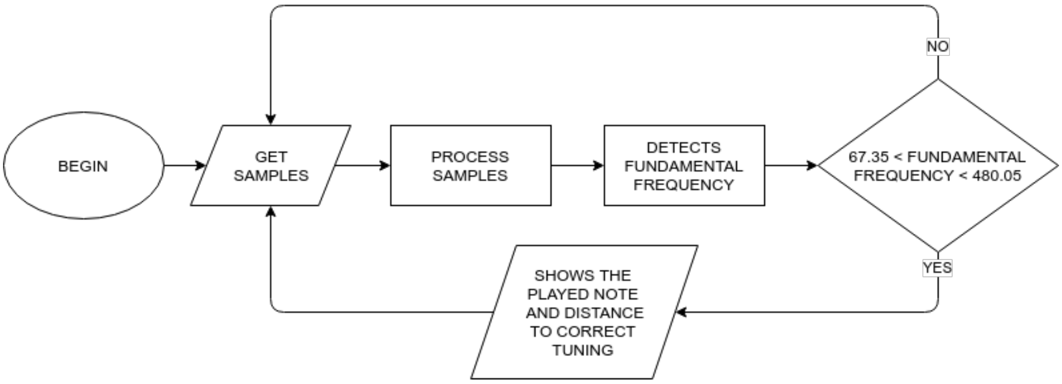


Table 1- Musical notes sampled on the guitar

	E2 (Hz)	A2 (Hz)	D3 (Hz)	G3 (Hz)	B3 (Hz)	E4 (Hz)
1st sample	81.7096	109.8698	147.4800	196.3036	249.0349	329.1102
2nd sample	81.7096	108.4248	149.4300	196.3036	247.0819	329.1102
3rd sample	81.7096	109.7034	147.4800	196.3036	249.0349	329.1102
4th sample	81.7096	110.3775	149.6328	196.3036	252.9410	329.1102
5th sample	81.7096	110.3775	148.4900	196.3036	247.0819	331.0633
Reference	82.41	110.01	146.84	196.01	246.96	329.65
Average	81.7096	109.7506	148.5026	196.3036	249.0349	329.5008
Difference between average and reference	0.7004	0.2594	-1.6626	-0.2936	-2.0749	0.1492
Relative error	0.85%	0.24%	-1.13%	-0.15%	-0.84%	0.05%
Standard Deviation	0	0.8000	1.0282	0	2.3920	0.8735

References

[1] Richard G. LYONS. *Understanding digital signal processing*. Prentice Hall, 3 edition, 2011.
[2] Thomas Grandke. Interpolation algorithms for discrete fourier transforms of weighted signals. 1983.
[3] Microchip Technology Inc. *16-Bit Microcontrollers and Digital Signal Controllers with High-Speed PWM, Op Amps and Advanced Analog*, 2011-2013.
[4] Antonio Pertence Júnior. *Amplificadores operacionais e filtros ativos: teoria, projetos, aplicações e laboratório*. McGraw-Hill, 2 edition, 1989.
[5] Felipe Neves. Ping-pong buffer para sistemas embarcados, 9 2016.