Applied Research and Smart Technology Vol. 3 No. 2 (2022) 49-55

ARS Tech

Applied Research and Smart Technology

Authors retain copyright

Journal homepage: www.journals2.ums.ac.id/index.php/arstech

ISSN 2722-9645

Research Article

Development of an acoustic guitar tuner and graphical user interface (GUI) using MATLAB

Mohammad Mainul Hasan[1*], Saiful Islam[2]

[1] *Department of Mechatronics Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia.*
[2] *Department of Systems Engineering, Kochi University of Technology, Kami, Japan.*

[*]Corresponding author: hasan.mainul@live.iium.edu.my

Permalink (DOI): https://doi.org/10.23917/arstech.v3i2.1185

ARTICLE INFO

Article history:

Received 06 October 2022
Revised 30 October 2022
Accepted 08 November 2022
Available online 27 December 2022
Published regularly 31 December 2022

Keywords:

Acoustic signal
Butterworth filter
Fast Fourier Transform
MATLAB GUI
Tuning

ABSTRACT

Beautiful music requires precisely tuned instruments. Tuning a guitar is necessary for any guitarist on their first day. This work demonstrates an understanding of the critical characteristics that must be considered while developing an acoustic guitar tuner and the logical process of designing such a tuner. The study aimed to create an algorithm using the Butterworth Filter and Fast Fourier Transform (FFT) capable of adequately analysing the frequency spectrum of a plucked guitar string to determine its fundamental frequency. The developed system compared the detected frequency and the standard frequency of the picked guitar string, which then requested the user on the tuning state of that corresponding string. The error in frequency detection was found to be in the order of 0.02%. The MATLAB App Designer tool created a Graphical User Interface (GUI) that users could use to easily tune guitars using the generated application. The implemented tuner overcame the steep learning curve and high sensitivity of traditional ones.

1. INTRODUCTION

Properly tuned instruments are essential for a musician to generate beautiful music. Tuning a guitar is changing the instrument's frequency to produce the proper arrangement of notes, a fundamental skill every guitarist must learn when they pick up their device. The talent is not so much in tune as in developing the ear for tuning. As any musician is aware, adjusting the pitch of an instrument may be challenging for an inexperienced individual and requires years, if not decades, of practice to master. Due to this, beginning guitarists' progress in their learning may be slowed down or even stopped. In order to alleviate the tuning process and make the most out of each training session, a digital guitar tuner can help automatically tune the instrument in the proper frequencies.

One of the most ubiquitous stringed instruments, the practical concept of digitally tuning a guitar opens up a vast technological and procedural landscape. Many researchers have worked on integrating hardware components and software algorithms to develop a guitar tuner application to assist new guitar students in tuning their instruments. Salcedo et al. [1] employed a piezoelectric transducer at the headstock to capture sound through the vibration of the guitar body and adjust it with a microcontroller device and frequency estimation method. Šarga et al. [2] implemented a digital guitar tuner using a myRIO microcontroller to detect the signal frequency and program it with LabView. Melo et al. [3] introduced the NAO robot to help new learners to tune the guitar. NAO robot showed the result of the guitar's tuning condition using an LED light blinking. Kumar et al. [4] designed an automated guitar tuner using cepstral analysis and a fuzzy controller on an Arduino board. Even though cepstral analysis for frequency identification is more prevalent in speech processing, an effort was made to adapt the approach for musical sounds, and promising results were observed.

Software-based approaches with Graphical User Interface (GUI) have gained equal popularity over the years. Busch et al. [5] used Parseval's theorem and Fast Fourier transform (FFT) to analyse the post-processing technique of sound waves. Harčarik et al. [6] explained the basic principle and importance of the Fast Fourier Transform (FFT) for acoustic sound waves over time-dependent series. Mary et al. [7] developed a MATLAB algorithm and added a hamming window to eliminate the noise from the original frequency. GUI is also made using MATLAB GUI builder to help the user tune the guitar. Patrașco [8] used Java programming to build a guitar tuner application for the Android operating system. A guitar tuner system has been designed by Deo Kumar et al. [9] in MATLAB Simulink, utilising a Blackman-Harris window to filter the signal. Rahnamai et al. [10] introduced the digital guitar tuner. FFT is used to determine the difference between the actual frequency and reference frequency. A fuzzy logic controller is designed using MATLAB Simulink, and a motor is attached to tune the guitar automatically. Using the JAVA program, Stanojevic et al. [11] developed a guitar tuner application. A comb filter has been used to suppress the unwanted signal. Sourav et al. [12] designed and developed a GUI using NI LABVIEW 2010 to tune and detect the chord of an acoustic guitar. A comparative investigation of the signal processing methods for guitar tuning has been made by Kulkarni et al. [13]. The authors found FFT much faster than the power spectral density algorithm.

Similar investigations on the tuning process have also been made for other musical instruments. Narongsak et al. [14] have devised an actuating mechanism to tune a Thai dulcimer instrument. A pickup sensor below the string picks up the sound, and the frequency is decoded using Goertzel Algorithm. Fraser [15] employed lasers to capture the vibrational frequency from light reflected off a drumhead. Wang et al. [16] implemented a violin tuner by capturing the string frequency using a Goertzel filter embedded in a microcontroller. The implemented system allowed the strings to be tuned to a $1Hz$ precision. Based on previous literature, it is evident that the effectiveness of a tuner is highly influenced by the signal-processing techniques employed. However, a guitar tuner using a combination of FFT and Butterworth filters has not been attempted. In addition, a GUI that is built with the MATLAB App Designer tool has not been established.

This paper introduces a new versatile approach to signal filtering and frequency detection for tuning an acoustic instrument. The algorithm is developed using MATLAB software. A dynamic low-pass Butterworth filter is used to eliminate unwanted noise signals. The filter's cut-off frequency is varied based on the string to be tuned. FFT is then used to detect the natural frequency of the plucked guitar string, and the result is compared to a list of standard frequencies. For a better user interface, the newly introduced MATLAB app designer tool has been used to make a GUI that allows a user to tune the guitar quickly. Applications developed with the MATLAB app designer have the advantage of faster runtime compared to the legacy GUI editor provided with MATLAB. Moreover, the software can be packaged to be used portably on various platforms. The system can be easily modified to tune guitars with any number of strings.

2. MATERIALS AND METHOD

An overview of the signal processing steps for the developed guitar tuner is shown in Figure 1. The operation starts with recording the sound of the guitar string, which is to be tuned. The subsequent steps apply signal conditioning techniques to the input signal to determine the fundamental frequency, and the application of the tuning state of that guitar string makes a decision. The following sections will explore each stage in more detail.

2.1 Recording and filtering

The B3 (fundamental frequency = $246.9Hz$ from Table 1) tuned guitar string is struck, and the sound is used as input to the developed tuner. The sound is recorded using a BM-100FX Condenser Microphone from an Antares® DX34M acoustic guitar. Figure 2 depicts the Fast Fourier Transform of the input signal. The most prominent peak in the magnitude spectrum corresponds to a frequency of $740.2Hz$, a noise signal. In fact, the actual frequency of the plucked B3 string

(247*Hz*, as labelled in Figure 2) is the signal with the second-largest amplitude. A fair amount of noise in the signal is indicated by higher frequency components in the spectrum, which may adversely influence the computations necessary for frequency detection. Hence, a Butterworth filter is applied to the signal to retrieve the string's original frequency. This filter was used due to its extremely flat frequency response in the passband and high roll-off in the stopband. As a result, the filtered response is fast and precise [17]. It is suggested to utilise a low pass filter with a cut-off frequency of around 500 Hz to the signal generated by a 1st-order Butterworth filter. This cut-off frequency was chosen since the highest fundamental frequency among the six strings does not exceed more than 330*Hz*, as per Table 1.

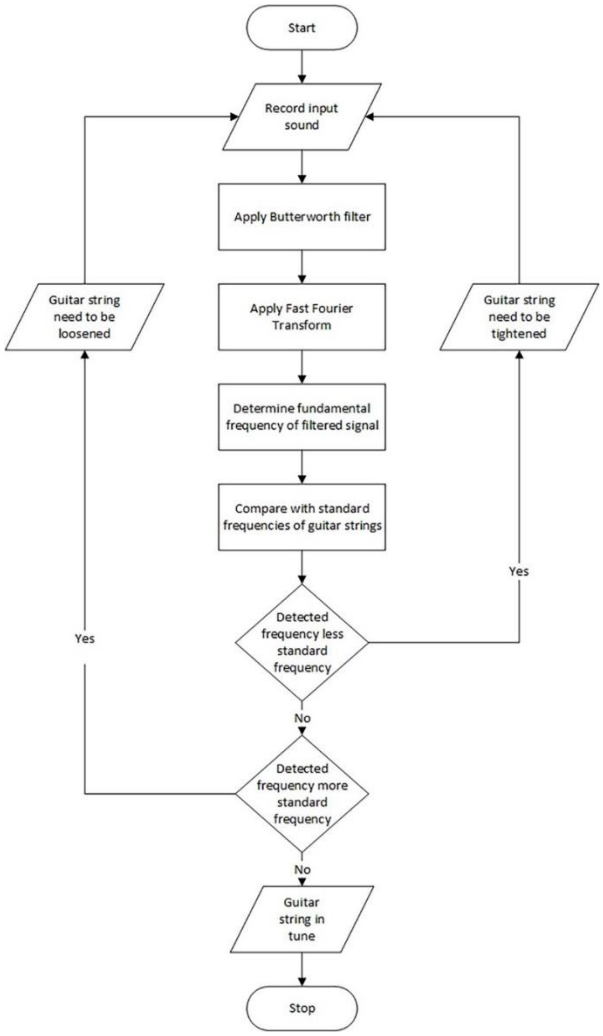

Figure 1. Flow chart for the signal processing steps.

The standard frequencies in Table 1 have been used in the MATLAB script as standard frequencies to compare with the recorded signal frequency. It can be noticed that the highest standard frequency is about 330*Hz* (1st string). For this reason, the cut-off frequency was chosen at no more than 500*Hz*. However, for the 1st string (E2), the fundamental frequency of the string was too low than the selected cut-off frequency of the filter. This allowed unwanted higher-frequency components to leak through the filtered response, making the peak frequency detection inaccurate. Hence, a separate low-pass Butterworth filter of 4th order was designed with a cut-off frequency of 100*Hz*.

Table 1. Standard frequencies for guitar strings [18].

Open Guitar Strings	String Number	Key Number	Standard Frequency (*Hz*)
E	6	E2	82.41
A	5	A2	110.00
D	4	D3	146.83
G	3	G3	196.00
B	2	B3	246.94
E	1	E4	329.63

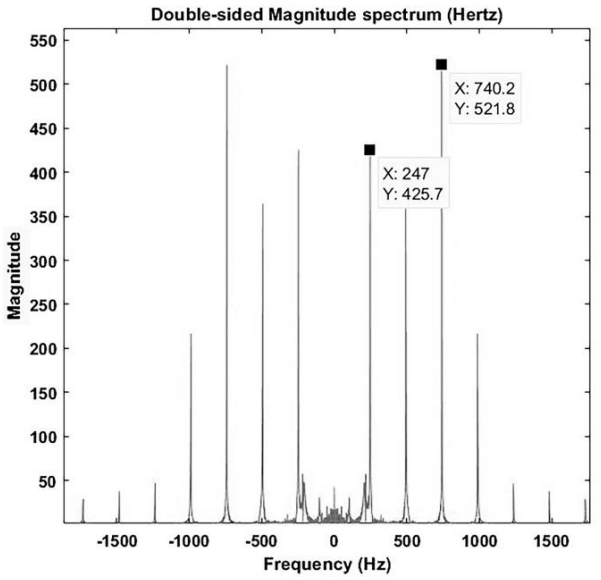

Figure 2. Frequency representation of B3 (in tune) signal.

2.2 Fundamental frequency detection

After filtering the signal, FFT is applied to the filtered output to remove high-frequency noises above the expected range. An FFT is a highly efficient version of the Discrete Fourier Transform (DFT), which translates discrete signals from the time domain to the frequency domain. The FFT function in MATLAB is based on the Cooley-Tukey algorithm [19], which breaks a larger DFT into smaller DFTs to improve processing performance and decrease complexity. The syntax of the built-in FFT function found in MATLAB is given by the syntax *fft(x)*, where *x* is the column matrix consisting of the amplitude

values of the sampled audio signal. The output of the FFT function gives us the Fourier transform values of each row in the input column matrix.

Furthermore, the spectral representation of the filtered signal can be plotted, as shown in Figure 3. The fundamental frequency of a string is the frequency that corresponds to the highest magnitude in the FFT. This frequency is determined by the *max()* function in MATLAB. From the magnitude spectrum of the filtered signal in Figure 3, it can be noticed that higher-order frequency components have been drastically attenuated. This made accurate frequency detection of the played note possible. The error in frequency detection can be calculated as follows:

Standard frequency of a tuned B3 string = 246.94Hz

Detected frequency of the tuned B3 string = 247Hz

$$Percent\ error = \frac{247-246.94}{246.94} = 0.024297\% \quad (1)$$

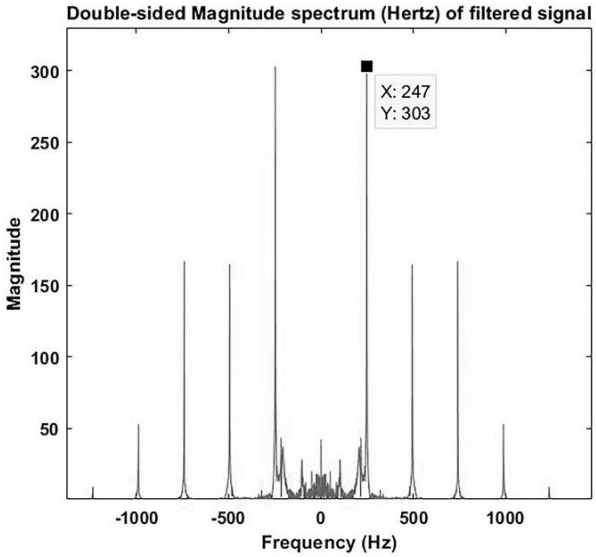

Figure 3. Frequency representation of low pass filtered B3 (in tune) signal.

2.3 Designing graphical user interface (GUI)

As shown in Figure 4, a graphical user interface (GUI) has been developed to provide a straightforward interface for carrying out the entire procedure. This was done using MATLAB App Designer, which has been integrated into the main application from version R2016a onwards. A particular string can be tuned by turning the knob to the correct position. The "Record" button is then pressed to start recording. A display box instructs when the user should begin playing the string. At the same time, the green indicator turns red to indicate recording has started. After 5 seconds, the program prompts the player to stop playing. The software then performs the necessary evaluation. The outcome of the analysis and the recommended action is then displayed on the respective display boxes. The first output window indicates the standard frequency for the selected string.

A display box is also given to show the frequency identified as a result of signal processing. The difference between the measured frequencies of the string to its standard frequency is then provided in the last output box. This gives the user context for adjusting an out-of-tune string to the correct frequency.

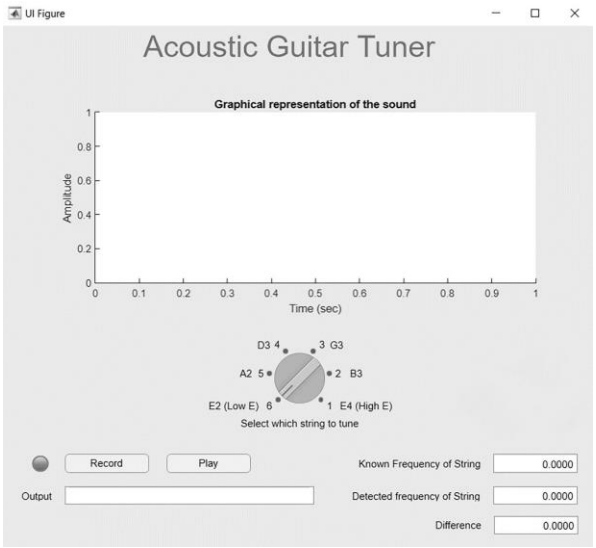

Figure 4. GUI layout.

3. RESULTS AND DISCUSSION

The frequency of the recorded note is compared to the required standard value (Table 1) for that note. The user can choose which string they want to tune at the start of the process, and the relevant standard frequencies are used for analysis here in the output phase. Fender® website provides in-tune notes that are about greater than a 1Hz difference in their fundamental frequency [20]. Therefore, it is assumed that the detected frequency does not need to be exactly equal to the standard frequencies of the played note for precise tuning. For this reason, a tolerance of ±1.5Hz of the standard frequency is used in this guitar tuner for comparison. The result is calculated as follows: If the signal frequency is less than the standard frequency minus 1.5Hz (tolerance), the string must be tuned up or tightened (Figure 5). If the signal frequency exceeds the standard frequency plus 1.5Hz (tolerance), the string needs to be tuned down or loosened (Figure 6). Lastly, if all the above conditions fail, the string is definitely in tune (Figure 7).

In order to evaluate the efficacy of the developed guitar application, an investigation consisting of 18 experimental runs was conducted. Three tuning states were considered for each guitar string: high, in-tune and low, after which the string is struck and the sound recorded with a microphone. Frequency measurement is

also conducted before and after the Butterworth filtering to determine its effectiveness. Finally, the experimental and standard frequency difference, percentage prediction error, and tuning state are calculated.

Based on the higher frequencies detected before filtering for the low-frequency strings: D3, A2 and E2, it can be inferred that the original signal was more susceptible to interference from high-frequency noise signal. The signal filtration process thus allowed for accurate determination of the string frequency by extracting the original signal from noise. The maximum percentage detection error for in-tune strings used in the study was equal to 0.930, which indicates that the tuner application is efficient, as illustrated in Figure 8.

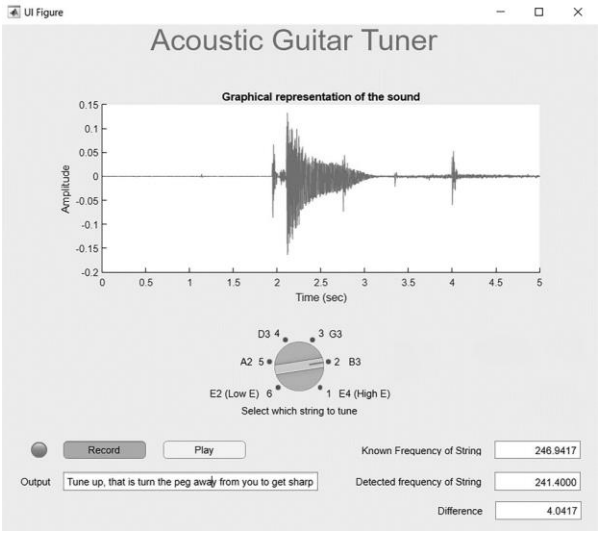

Figure 5. The output of the guitar tuner for an under-tuned B3 string.

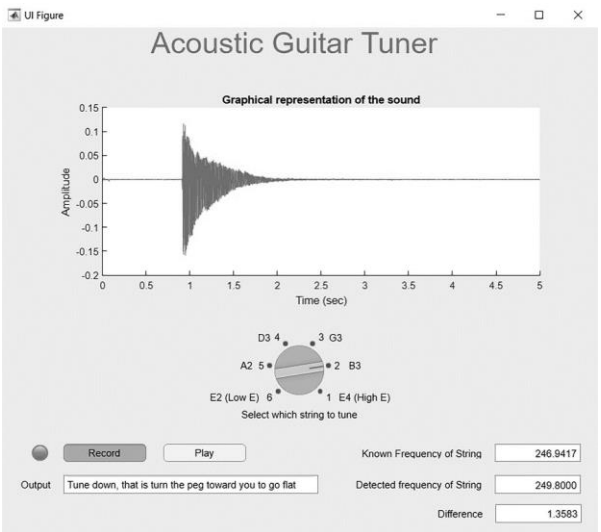

Figure 6. The output of the guitar tuner for an over-tuned B3 string.

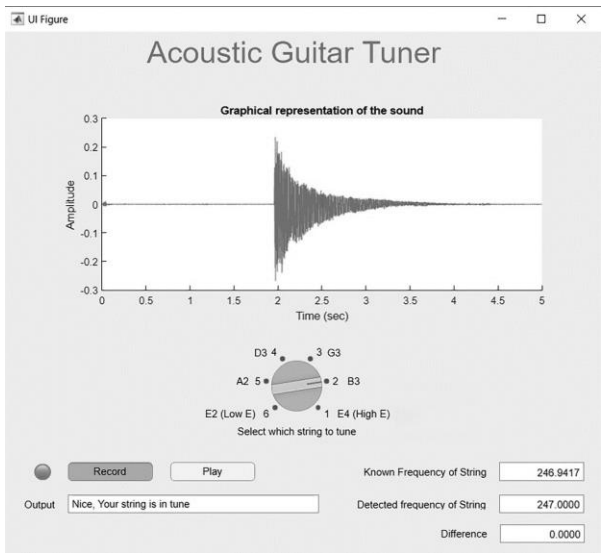

Figure 7. Frequency representation of low pass filtered B3 (in tune) signal.

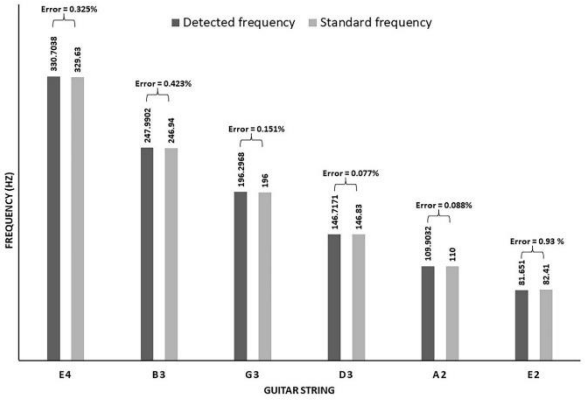

Figure 8. Performance analysis of guitar tuner.

4. CONCLUSION

The paper describes the design of a digital guitar tuner as well as the factors involved in tuning a guitar. The fundamental frequency of the notes is a crucial component to consider, and methods for determining it are also explored. The complete signal-processing steps which make this guitar tuner work has been presented in detail. The filter design used to remove noise and harmonics has also been outlined. An algorithm analysis found the detection error percentage to be less than 1%, demonstrating that it is an accurate system.

Developing the GUI made the application much more practical in that even those who do not know how to use MATLAB can still easily use the application to tune a guitar. The user interface was designed to be as intuitive as possible. All efforts were made to incorporate as much information as possible into the allotted space. Writing code for the call-back functions for each control was necessary to implement these functionalities. The

parameters of each control were adjusted, and scripts were built to alter parameters in response to user input and signal processing output.

CONFLICTS OF INTEREST

The author declares that no competing financial interests could have appeared to impact the work.

ACKNOWLEDGEMENT

The authors would like to thank the International Islamic University Malaysia and Kochi University of Technology Japan for supporting the research.

REFERENCES

[1] J.S. Salcedo, I.M. Gila, I.R. Ruano, A.S. Garcia, E.E. Esteves, J.G. Ortega, and J.G. Garcia, "Design and development of a low cost automatic stringed instrument tuner", *Jornadas de Automática*, pp. 604–610, 2019. https://doi.org/10.17979/spudc.9788497497169.604

[2] P. Šarga, and D. Demečko, "Design and realisation of the guitar tuner using MyRIO", *Journal of Automation and Control*, Vol. 5, No. 2, pp. 41–45, 2017. https://doi.org/10.12691/automation-5-2-2

[3] R. Melo, R. de Paulo Monteiro, J.P.G. de Oliviera, B. Jeronimo, C.J. Bastos-Filho, A.P. de Albuquerque, and J. Kelner, "Guitar tuner and song performance evaluation using a NAO robot", *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, pp. 1-6, IEEE, 2020. https://doi.org/10.1109/LARS/SBR/WRE51543.2020.9307156

[4] A. Kumar, S. Srivastava, M. Chandra, and Sahoo G, "Guitar tuner using cepstral analysis and fuzzy controller on Arduino board", *Microsystem Technologies*, 2018. https://doi.org/0.1007/S00542-017-3623-2

[5] T. Busch and T. Busch Consulting, "Practical applications of digital signal processing theory practical applications of digital signal processing theory", *Proceedings of 26th International Congress on Acoustics and Vibration,* 2019. https://doi.org/10.1007/s00542-017-3623-2

[6] T. Harčarik, J. Bocko, and K. Masláková, "Frequency analysis of acoustic signal using the fast Fourier transformation in MATLAB", *Procedia Engineering*, Vol. 48, pp. 199–204. 2012. https://doi.org/10.1016/j.proeng.2012.09.505.

[7] M. Lourde, and A. K. Saji, "A digital guitar tuner," 2009, *arXiv preprint arXiv:0912.0745*.

[8] A. Patrașco, "Itune guitar tuner application for android driven mobile devices", *Technical-scientific Conference of Collaborators, PhD Students and Students, Technical University of Moldova*, Nov. 2014, pp. 134–135. Accessed: May 08, 2022.

[9] B. Deo Kumar, A. Kushwaha, A. Kumar, and A. Agarwal, "Design and implementation of digital guitar tuner using MATLAB", *The International Conference on Advance Computing and Innovative Technologies in Engineering*, ICACITE 2021, pp. 547–549. 2021. https://doi.org/10.1109/ICACITE51222.2021.9404728

[10] K. Rahnamai, B. Cox, and K. Gorman, "Fuzzy automatic guitar tuner", *Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS*, pp. 195–199, 2007. https://doi.org/10.1109/NAFIPS.2007.383836

[11] M.S. Stanojević, and M.R. Bjelić, "Digitalni štimer za gitaru", *Proceedings of Papers, 2011, pp. 1574–1577*, https://doi.org/10.1109/TELFOR.2011.6143860

[12] S. Sourav, S. Balamurugan, R. Marimuthu, R. Sudha, and A. Bagubali, "Acoustic guitar tuner and identification of chords using LabVIEW", *Global Journal of Pure and Applied Mathematics*, Vol. 11, No. 3, pp. 1171–1178, 2015.

[13] U. Kulkarni, S. Kaushik, L. Lobo, and R. Sonkusare, "Comparative study of digital signal processing techniques for tuning an acoustic guitar", *The 7th International Conference on Smart Structures and Systems (ICSSS 2020)*, pp. 1-6, 2020. https://doi.org/10.1109/ICSSS49621.2020.9202368

[14] N. Tirasuntarakul, and A. Dheeravongkit, "An automatic multi-string musical instrument tuner using one-to-many micro actuating mechanism", *Proceedings of the International Conference on machine learning and Machine Intelligence*, pp. 64–67. 2018. https://doi.org/10.1145/3278312.3278323

[15] Z. Fraser, "Laser tuner: a novel approach to pitch detection on a drumhead", *The Canadian Science Fair Journal*, Vol. 2, No. 5, 2019. https://doi.org/10.18192/csfj.v2i32020120122

[16] Z.J. Wang, and C.Ortega-Sanchez, "Electronic assisting violin tuner", *TENCON 2012 IEEE Region 10 Conference*, 2012. https://doi.org/10.1109/TENCON.2012.6412214

[17] M. Shouran, and E. Elgamli, "Design and implementation of Butterworth filter", *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 9, no. 9, 2020, Accessed: May 09, 2022.

[18] D. Ramsay, T. Burke, D. Barry, and E. Coyle, "A novel Fourier approach to guitar string separation", *IET Irish Signals and Systems Conference*, June 2011.

[19] M. Frigo and S.G. Johnson, "FFTW: An adaptive software architecture for the FFT", *International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 3, pp. 1381–1384, 1998, https://doi.org/10.1109/ICASSP.1998.681704

[20] B. Alfredo, "Automatic acoustic guitar tuner", *Thesis, Department Massachusetts Institute of Technology. Department of Mechanical Engineering,* 2005. http://hdl.handle.net/1721.1/32878