IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 5, October, 2024, pp. 1881-1890
Journal homepage: http://iieta.org/journals/isi

Fast Fourier Transform for Guitar Tuner Synchronization

Eddy Triswanto Setyoadi

Department of Information System, Institut Informatika Indonesia Surabaya, Surabaya 60189, Indonesia

Corresponding Author Email: eddy@ikado.ac.id

https://doi.org/10.18280/isi.290520

Received: 19 March 2024
Revised: 5 September 2024
Accepted: 21 September 2024
Available online: 24 October 2024

Keywords:
guitar, tuning, android, Fast Fourier Transform algorithm, synchronization

ABSTRACT

To produce appropriate chord harmonies on the guitar, tuning or tuning the strings is required. However, most guitar learners perform tuning manually based on hearing. This will certainly take a long time because, in the tuning process, the user must turn the string knob repeatedly to get a harmonious and precise tone. Although there are currently many guitars tuning applications on Android, in the tuning process, users must turn the string knob manually. This research aims to create a tool called Learn Guitar Chords to perform the tuning process automatically, and the results are fast and accurate according to the frequency of standard guitar string tones using the Fast Fourier Transform (FFT) algorithm. FFT can convert signals from the time domain into the frequency domain, a series of numbers in the time domain $f(x)$ is converted into the frequency domain $f(u)$. Considering the test results using the black box testing method that has been carried out, it can be said that the Fast Fourier Transform-based guitar tuning synchronization design application on Android can obtain the frequency of user input properly. In addition, accuracy testing was also carried out by testing the tuning process by comparing it with 2 applications, namely Absolute Guitar and Guitar Tuner. The results obtained from the application comparison prove that the accuracy of the tuning process in the Learn Guitar Chords application is very good because it can produce the same results as other applications. Although the equal temperament scale is one of the most popular tuning techniques for stringed instruments, other techniques should also be considered because it is used in various musical instruments.

1. INTRODUCTION

The guitar is a musical instrument that has strings where the strings have a tone with a certain frequency. Tuning is the process of changing the frequency of a musical instrument to create the desired song arrangement. The tuning process can be confusing for new guitarists, especially since there are many different ways to tune a guitar. Every time a string on the guitar is plucked, this tuner can output tone parameters. However, the cost of tuners on the market is generally rather high. Using mobile applications can make it simpler for beginners to use and receive it for free. The Fast Fourier Transform (FFT) method is used in the working concept to convert the input time domain signal to a frequency domain.

Hasan and Islam [1] developed an acoustic guitar tuner and graphical user interface (GUI) using MATLAB as part of prior research on the guitar tuning procedure in this fundamental frequency research obtained through the Fast Fourier Transform (FFT) method. Venkataramanan et al. [2] designed a Sasando tuner application that has a high level of accuracy in detecting the fundamental frequency of the 32-string Sasando type using the Fast Fourier Algorithm Transform (FFT) and Harmonic Product Spectrum (HPS). Zhuo [3] presents the state-of-the-art for stringed instrument tuning tools, examines the issues, and suggests an automatic tuning system based on wireless sensor networks using those tools for widely used stringed instruments. Tampubolon and Nasution [4] designed an automatic guitar tuner using an 8-bit microcontroller. In this research, they developed an automatic tuner that uses an Arduino Uno to process audio signals. Processing the data from the ADC of the Arduino Uno to measure the time it takes to complete one full wave is how the frequency reading is calculated. To ensure the frequency is accurate, the frequency of the nearest string is calculated after the frequency value is obtained. The stepper motor will rotate in several steps if it is found that the frequency is off. According to the research, this tool works better when the frequency read is greater than the reference frequency, but it can tune successfully when the frequency is lower than the reference frequency.

Šarga and Demečko [5] made a guitar tuner based on MyRIO, which works in the LabVIEW environment. Thakuria et al. [6] developed a musical instrument tuner using audio signal processing techniques to determine if a musical note is flat or sharp. It emphasizes the importance of identifying the fundamental frequencies of musical audio signals for tuning. Melo et al. [7] explore the use of a NAO robot to aid children in learning acoustic guitar by implementing a guitar tuner and song performance evaluation system. Key aspects include optimizing note detection through analysis of window length and employing Goertzel's algorithm for real-time pitch detection. The system also evaluates song performance with a metronome and was tested successfully, even in noisy environments, although challenges were noted with detecting notes around 680 Hz. Pradana et al. [8] created an automatic guitar tuner based on the open-tuning approach.

Branta et al. [9] made musical instrument tuning with the Yin Algorithm using a microphone, piezoelectric transducer, and line input. Although the cost is low, it turns out that the precision of the tools produced is quite high at 1%. Stange et al. [10] successfully constructed a tuning procedure by implementing a dynamic adaptive scheme. This procedure is realized through a program called Just Intonation. Kumar et al. [11] made guitar tuning using MATLAB where to improve accuracy; before entering the FFT process, the signal is first filtered using the Finite Impulse Response (FIR) technique. introduces an automatic tuning system designed for multi-string musical instruments, focusing on the Thai hammered dulcimer. The system features a one-to-many micro actuating mechanism that uses a single motor to independently and simultaneously adjust the tension of multiple strings to achieve a target frequency. It employs pickup sensors to monitor string vibrations and the Goertzel algorithm for frequency detection, alongside a tuning time predictor to optimize the process. The goal is to simplify the tuning process for both beginners and professionals, which traditionally requires significant expertise and time [12].

2. RELATED WORK

2.1 Guitar tuner

A guitar tuner is a tool that helps to bring the guitar strings' pitch according to the desired or standard tones. To achieve the required pitch, the guitar must first be picked to establish a frequency, then the strings must be chosen to be tuned [13]. The objective of a guitar tuner is to aid musicians in accurately tuning the strings of a guitar to the correct pitch. It does this by detecting the pitch of a note played on a string and comparing it to the desired pitch for that note. If the pitches do not match, the tuner provides the necessary information to the musician to adjust the string's tension and bring the note into tune. The goal is to ensure that the guitar produces the correct sound, which is crucial for accurately and harmoniously playing music [14].

The string tension is then adjusted. While manual completion of this process is possible, many people find it difficult and time-consuming. Thus, the emergence of automatic guitar tuners can be attributed to technological improvements. These devices automate the tuning process with various components, such as acrylic, push button, servo motor, Op-Amp, LED, and LCD. The Arduino microcontroller initializes the frequency and instructs the servo motor to alter the string tension. The Op-Amp amplifies the frequency produced when the guitar is picked. After that, the results are shown on the LCD, which also serves as a tuning indicator [8, 15].

2.2 Design of automatic guitar tuner

Creating an automated guitar tuner comprises multiple stages and elements, such as frequency analysis, motor control, and signal processing [16]. Below is a general process outline:

(1) Signal Acquisition: A microphone records the guitar's sound, which is then transformed into an electrical signal and stored in a .wav file.

(2) Frequency Analysis: Using cepstral analysis, the .wav files are processed to determine the fundamental frequency of each frame. The fundamental frequency of the played note is also assessed using this methodology.

(3) Comparison and Error Calculation: The intended set frequency and the estimated note frequency are compared. A fuzzy logic controller receives the difference between the specified and calculated frequency as input.

(4) Fuzzy Logic Controller: The fuzzy logic controller produces an output based on the difference in the input. This output produces a pulse width modulation (PWM) signal with a variable duty cycle.

(5) Motor Control: A motor driver circuit amplifies the PWM signal's output and rotates a motor at a variable speed in the desired direction. This modifies the string's tension, which causes the string's frequency to alter to reach the required pitch.

(6) Testing and Verification: A string is tuned to a certain note to test the system. When the fuzzy output is almost zero, the process ends. An automatic guitar tuner application can be used to validate the system.

In Sarga's research, Šarga and Demečko [5] discussed the design and realization of the Algorithm of guitar tuners. The algorithm appears to involve the following steps:

(1) A microphone attached to the myRIO device captures the sound of the guitar string.

(2) The myRIO device acts as an A/D (analog-to-digital) converter, converting the microphone's analog signal into a digital signal.

(3) The measured frequency and its difference from the required frequency of the tuned string are then calculated from the digital signal.

(4) A user interface displays this data, including a meter that displays deviation from the desired frequency, the frequency's current numerical value, the desired frequency of the tuned string, and its associated musical notation.

(5) The program's infinite loop repeats the measurement.

2.3 Synchronization of musical instrument

The term "synchronization" in music refers to the phenomenon whereby the auditory system's neural activity matches the perceived sound's rhythms. It is thought that successful auditory perception depends on brain-environment synchronization. The most prominent isochronous pulse in music, which causes listeners to move their bodies or tap their feet, is the beat, which frequently causes synchronization to take place. In contrast to the amplitude envelope, which measures intensity fluctuations over time, the researchers of the study by Burger et al. [17] discovered that the spectral flux of music, which refers to variations in pitch and sound quality, evoked the strongest neural synchronization. Additionally, it was discovered that slower beat rates, high levels of familiarity, and beats that were simple to detect made this synchronization stronger.

Research about synchronization was conducted by Weineck et al. [18], who investigated the phenomenon of neural synchronization to music, focusing on how different musical features, tempo, familiarity, enjoyment, and ease of beat perception affect this synchronization. The study used techniques such as stimulus-response correlation (SRCorr), stimulus-response coherence (SRCoh), and temporal response functions (TRFs) to assess neural synchronization. The results showed that the strongest neural synchronization was to the spectral flux of music, which captures information about rhythmic structure. This synchronization was stronger with slower beat rates, familiar music, and music with easy-to-perceive beats. The strength of neural synchronization did not differ between musicians and non-musicians. The study also found that the SRCorr and SRCoh measures were strongly correlated with the TRF correlations, suggesting that both methods provide similar results. The findings highlight the importance of tempo and musical features in neural synchronization to music.

In research conducted by Mao et al. [19], the synchronization technique involved having participants move in time to musical stimuli while capturing their movements with an optical motion capture system. The difference between the movement and the musical beat was calculated over time to evaluate the relationship between the movement and the musical phase. From this difference distribution, the negentropy statistic-the additive inverse of Shannon entropy-was extracted as a gauge of synchronization precision at various metrical levels. This measurement, which has been applied to research on neural synchrony, is reliable in identifying both in-phase and anti-phase synchronization.

2.4 Fast Fourier Transform algorithm

A frequently used tool in digital signal and image processing is the Fast Fourier Transform (FFT). Cooley and Tukey proposed 1965 that the discrete Fourier transform (DFT) be computed using much fewer mathematical operations. The foundation of the FFT algorithms is the factorization of the Fourier matrix into a set of sparse matrices, or matrices with many zero entries. The original matrix is represented as a product of a log 2 N sparsely structured matrix in the Cooley-Tukey algorithm. This algorithm has roughly (N/2) log complexity [20].

Numerous applications have required further development and adaptation of the FFT algorithm. For instance, there are DFT algorithms for real-valued sequences with lengths ranging from 3 to 9 and FFT algorithms for real-valued series. Fewer operations are required because these algorithms don't work with complex numbers. The FFT algorithm has been widely applied to several disciplines, including digital image processing, acoustics, and signal processing [20, 21].

Jannereth and Esch [22] also uses the FFT method, but FFT is used to analyze the tones of various musical instruments. This study examines the timbres of various musical instruments using the Fast Fourier Transform (FFT). Converting data in a signal from the time domain to the frequency domain using the FFT algorithm is significantly simpler and faster to perform than by hand. Audacity®, a program that implements the FFT algorithm to compute frequency data from a finite sample of sound and generates a frequency spectrum, is the program used in this investigation to obtain the frequency spectrums of various instruments. This frequency spectrum plot clearly shows the constituent timbre frequencies present in each sound sample, enabling timbre analysis. The timbre of a musical instrument is quantitatively examined using FFT software.

3. PROPOSED MODEL

Even though there are currently many guitars tuning applications on Android, in the tuning process, the user has to turn the string knobs manually. The main objective of the research conducted by Kumar was to identify the key characteristics that must be considered when designing guitar tuners. This research used a note played, and the program determines if the pitch needs to go up, down, or is correct. The FFT expertise, windowing, filters, identification of pitches and their relationship to basic frequency, MATLAB, and Simulink were all used in the algorithm. Their design could only be adjusted to a restricted number of notes, only six (E2, A2, D3, G3, B3, E4) [11]. Besides that, Becchi [23] created an automated technique for tuning acoustic-electric guitars in 2017. It processed input data using a 32-bit ARM processor on the Arduino Due platform, which it utilized to operate motors to fine-tune the instrument. However, from the result, it could not obtain precise readings and modifications because of the board's restrictions when doing FFT computations. Becchi stated that 0.5 Hz was the maximum acceptable error value for the apparatus. Despite this, the obtained value was above the ideal, at about 4Hz.

This research explores previous studies on frequency analysis methods. As in Shin et al. [24] which focuses on the comparison of Multiple-Measurement Sparse Bayesian Learning (SBL) with Fast Fourier Transform (FFT) for frequency analysis, especially in passive solar systems. This research uses SBL to exploit multiple measurements in the space and time domains, improving frequency detection by utilizing sparse estimation techniques. This approach is compared to the FFT, which is traditionally used for frequency analysis but often suffers from noise and low resolution due to its reliance on a limited signal length. Findings show that SBL significantly improves frequency detection by providing higher resolution and reduced noise compared to FFT. However from the findings, the computation of FFT is more efficient than SBL, this efficiency is due to its ability to compute the Discrete Fourier Transform (DFT) in \(O(N \log N)\) time, which is significantly faster than the computational complexity involved in SBL. Research Hamdan et al also about frequency spectrum and time-frequency analysis for the need or objective tools to evaluate the acoustic properties of violins, which is crucial for parents, novices, and instructors in selecting quality instruments. The research employs time-frequency analysis (TFA) using two main tools: PicoScope oscilloscopes for Fourier spectra and Adobe Audition for spectrogram analysis. These methods were used to analyze the fundamental frequencies and harmonic overtones of the G, D, A, and E strings from four violins manufactured by Eurostring, Stentor, and Suzuki. The research results indicate significant variations in harmonic patterns and overtone intensities among the violins analyzed. In this research, the Fast Fourier Transform (FFT) is used to extract spectral features from the musical signals of violins. FFT is a popular frequency analysis technique, particularly for non-stationary signals where statistical properties vary with time. It helps in identifying the fundamental and overtone frequencies, which are crucial for distinguishing the unique sound characteristics of each instrument. While Time-frequency analysis (TFA) refers to a method that describes the sound in the time-frequency plane. It allows for the identification of dominant frequencies at specific times, offering a detailed view of how frequency content evolves. TFA is particularly useful for analyzing complex tones where individual harmonics are difficult to distinguish by ear [25].

So, from the previous research on frequency analysis, the FFT algorithm was chosen because FFT provides a comprehensive frequency spectrum of the entire signal, which is useful for identifying the fundamental frequencies and harmonics present in the sound. This is particularly beneficial for stationary signals where the frequency content does not change significantly over time. Besides that, FFT can process signal transformations from the time domain to the frequency domain, this algorithm can be used to perform fast and efficient sound transformations. FFT was chosen because the process of changing the signal into frequency involves a decimation process, namely decimation in time (DIT) and decimation in frequency (DIF).

Based on the existing problems, a tool is made that can carry out the tuning process automatically (the string knob rotates according to the desired tone), and the results are fast and accurate according to standard guitar string tone frequencies. Various problems encountered in this tuning process are presented visually through a fishbone diagram in Figure 1.

This research was carried out in the following stages:

(1) Observation: by exploring similar guitar tuner applications.

(2) Interviews with target users, namely beginners who will learn guitar instruments. Interviews are important because they are a source of information by the author in making interface design and guitar learning material suitable for beginner levels.

(3) System analysis and design, namely designing the interface of the application and designing the database structure in the application. Analysis of the sound inputted by the user will be converted into frequency waves (Hz) using the Fast Fourier Transform algorithm.

(4) Implementation of the system, namely testing the installation of applications on Android-based mobile devices.

(5) Test and evaluation, namely testing the application evaluating the errors in the application, and making improvements so that the application can be more effective.

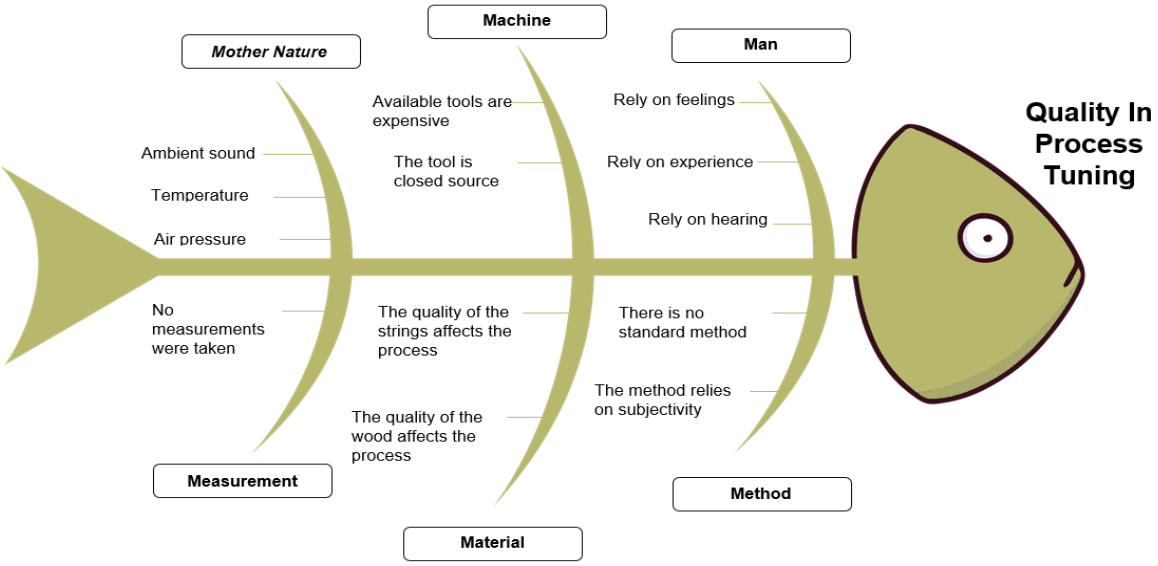

Figure 1. Fish bone diagram of problems in the guitar tuning process

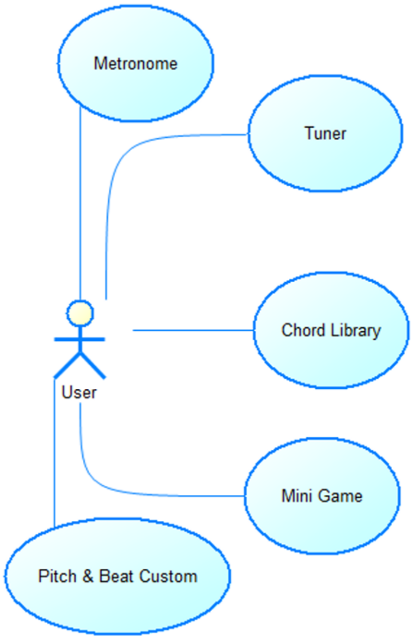

Figure 2. Use case diagram of the learn guitar chords application for tuning

After the interview, to better understand the design and workflow of the system, a use case diagram was created as shown in Figure 2.

In Figure 2, it is explained that one user actor can choose the metronome, tuner, chord library, mini-game, or custom pitch & beat menu. Users can choose the tuner menu to adjust the pitch of each string on the guitar; in this menu, the user will pluck the guitar strings alternately and adjust the pitch. In the metronome menu, users can practice guitar playing on beat by setting the beat per minute (bpm) and beat as desired. The system will display the existing chord dictionary and the chord's sound in the chord library menu. In the mini-game menu, users can choose to practice chord diagrams and chord sounds or practice with a guitar by sounding certain chords. In the pitch & beat custom menu, the user will select a song from the device, and then the user can also set the song beat and the pitch used in the song.

In Figure 3, the user will provide input in the form of sound by strumming a guitar string, after which the system will perform windowing to reduce spectral leakage that can arise due to low sample rate (Fs). After that, FFT is carried out, which aims to convert the signal from the time domain into the frequency domain; the number series in the time domain $f(x)$ is converted into the frequency domain $f(u)$, after that if it is detected, the system will display the results of matching the standard guitar tuning frequency if the system will not reprocess. Here is the formula for window hamming:

$$w(n) = 0{,}54 + 0{,}46\,cos + \frac{n\pi}{M}, -M \leq n \leq M$$

The FFT algorithm will be used to convert the signal from the time domain to the frequency domain following windowing. The length of the symmetrical spectrum produced by the FFT technique is equal to the value of the sampling frequency. After passing the calculation process with equation FFT has been successfully calculated, the frequency value that corresponds to the target is obtained. For instance, to identify tone A, it will determine whether or not the detected frequency corresponds to 110Hz. The presence of the detected frequency in comparison to the target tone's range yields the match. The following equation yields the range for any tone:

$$x = f_{n-1} + (\tfrac{fn - fn{-}1}{2})$$

$$y = f_n + (\tfrac{fn{+}1{-}fn}{2})$$

with:

f_n=target tone frequency (A)
f_{n-1}=frequency of the note before the target (e.g., 82)
f_{n+1}=the frequency of the tone after the target tone (e.g., 146)
x=upper limit
y=lower limit

With a target tone frequency of 110 and using the matching formula, we obtained:

$$x = 82 + (\tfrac{110 - 82}{2}) = 96$$

$$y = 110 + (\tfrac{146 - 110}{2}) = 128$$

So, if we are going to measure tone A, then the range is depicted in Figure 4. The following is the Fast Fourier Transform equation:

$$f\,(u) = \sum_{x=0}^{x=N-1} f(x)\left(cos\frac{2\pi ux}{N} + j\sin\frac{2\pi ux}{N}\right)$$

When open the application, the first page that appears is the tuner menu, where the user can see the tone matches on the guitar strings that are plucked, and the user can choose the sequence of string numbers to match the tone. At the top is a picture showing the tone match indicator in the tuning process; at the bottom is a picture of a guitar with the tone of each string with standard rules. In this menu, the user can provide sound input by hacking guitar strings; then, the system will process the sound signal and provide the user with a frequency-matching result.

The design interface must also be considered so that users can understand better and not get bored easily. This sub-chapter will provide an overview of the interface design in the application. When opening the application, the first page that appears is the tuner menu, as in Figure 5, where the user can see the tone match on a plucked guitar string, and the user can choose the sequence of string numbers to match the tone. At the top is a picture showing the tone match indicator in the tuning process; at the bottom is a picture of a guitar with the tone of each string with standard rules. In this menu, the user can provide sound. input by hacking guitar strings, the system will process the sound signal and give it to the user a frequency-matching result.

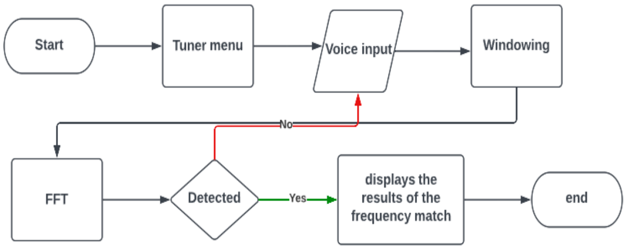

Figure 3. Tuner flowchart

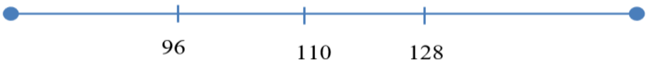

Figure 4. The frequency range that represents tone A

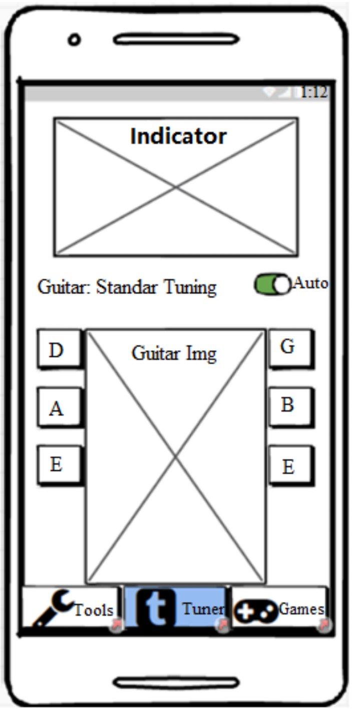

Figure 5. Main display (tuner menu)

4. EXPERIMENT RESULT AND EVALUATION

The minimum specifications of the “Learn Guitar Chords” application to run properly include:

a. Smartphone specifications:
- 64-bit CPU 1.5GHz quad-core Snapdragon 410, GPU Adreno 306.
- 1.5GB RAM.
- Screen size 5 inches, 540x960 pixels.
- Free memory internal storage 500MB.
- 8GB ROM.

b. Software specifications:
- Android operating system version 5.1 Lollipop with API 22.

Table 1. Smartphone specification

No.	Type	Redmi 2	Samsung A50
1	Android	Oreo 8.1	Pie 9.0
2	CPU	Qualcomm Snapdragon 625	Exynos 9610
3	GPU	Adreno 506	Mali-G72 MP3
4	RAM	3GB	6 GB
5	ROM	32GB	128 GB
6	Display	720x1440 pixel	1080x2340 pixel

Table 2. Guitar specification

No.	Type	Redmi 2	Samsung A50
1	Nylon	Osmond C500	D'Addario EJ49
2	Steel	Yamaha CPX 500 II	Elixir Nanoweb Phosphor Bronze

The smartphone and guitar specifications used during the trial can be seen in Table 1 and Table 2.

Figure 6 is the tuner menu's main page. On this page, there are three navigation buttons at the bottom of the page, which are useful for moving to the tools and mini-game pages. On this page, the user can start and stop the tuning process by pressing the start tuning button. After pressing the start button, the user can provide input with guitar strings, which then the system will display the results of the matched sound frequency. The order of the strings on the guitar is marked with the notation E, B, G, D, A, E, as shown in the picture of the lowest guitar string. The symbol "b" indicates the result of matching the sound frequency, which is low/less than the frequency it should be, while the symbol "#" indicates the matched sound frequency is high/more than the frequency it should be, in the middle the symbol "–" indicates the notation of the tone being matched.

After the user strums the guitar strings, the system will perform a windowing process to reduce spectral leakage caused by the low sample rate (Fs), after which the Fast Fourier Transform method is applied to get the frequency from the sound input through the smartphone's built-in mic. After that, the FFT is completed, and the results are displayed if the frequency is lower than that of each guitar string. The "#" symbol's indicator light will illuminate if the frequency is greater than it should be, signalling that the user must loosen the strings by rotating the string knob on the guitar headstock.

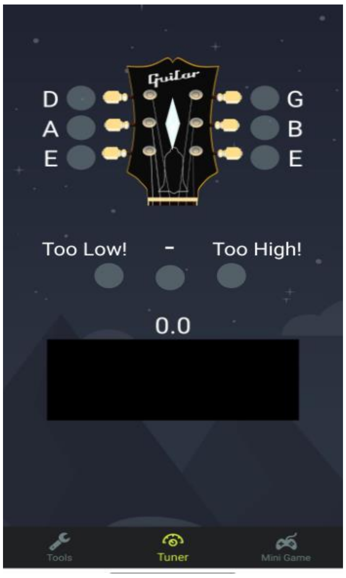

Figure 6. Tuner menu main page

Figure 7. Initial test conditions for tuner pages

Table 3. Scenarios for testing page tuner

No.	Test Scenario	Expected Results
1	Using the tools button	The system displays the tools page
2	Use the tuner button	The system displays the tuner page
3	Use of mini-game buttons	The system displays a mini-game page
4	Scan voice input	The system can detect the input sound and display the sound frequency results

The following will explain the black box testing process. The process includes testing the main page and testing the tuner page. Three musicians with varying levels of experience-advanced, intermediate, and beginner-performed this test process. This test was run to see how well the system could detect the user's input sound and execute button actions from the tuner page. The functions to be tested were scanning the user's voice input, using the tools button, using the tuner button, and using the mini-game button. The initial condition of the main page test can be seen in Figure 7.

Figure 7 shows the test's initial conditions: The system displays the tuner page. This page has buttons for tools, tuners, and minigames. Users can provide voice input on this page using the smartphone's default mic. The following is a scenario Table 3 from the tuner page test. In addition, testing was also carried out by testing the system's ability to carry out the functions of the buttons on the metronome page. The functions to be tested are the use of the +, -, number picker bpm, play, and beat buttons. The initial condition of the metronome page test can be seen in Figure 8. This test scenario can be seen in Table 4.

In addition to using black box testing in this research, a questionnaire is also used to measure user experience in using Learn Guitar Chords. The questionnaire was given to 30 respondents who were randomly selected by distributing the application (apk) and questionnaires in the form of Google Forms to several groups in WhatsApp and Telegram social media. Respondents can answer questions by selecting the appropriate answer.

The results of the black box testing trials will include pictures of the final conditions and scenario tables of the trial results. The results obtained include testing the main page and testing the song page. The application gave the same results after being tested by two testers in the beginning and intermediate musician categories. The outcomes of all the tests that have been run are listed below.

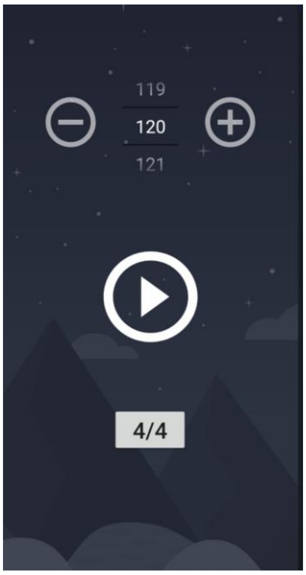

Figure 8. Initial condition of metronome page testing

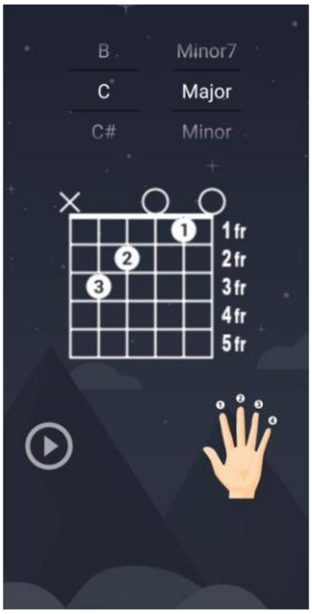

Figure 9. Initial condition testing chord library page

Table 4. Metronome page testing scenario

No.	Test Scenario	Expected Results
1	Use of the key	The system will increase the bpm by 1 point
2	Use of the-key	The system will decrease bpm by 1 point
3	Using the number picker bpm	The system will change the bpm
4	Play button usage	Play button usage
5	Use of the beat button	Use of the beat button

Table 5. Chord library page testing scenario

No.	Test Scenario	Expected Results
1	Use of the number picker chord button	The system will replace the image and sound of the selected chord
2	Use of the number picker chord type button	The system replaces the image and sound of the selected chord.
3	Use of the play button	The system will play the sound of the selected chord

Table 6. Transpose and beat page testing scenario

No.	Test Scenario	Expected Results	The Result
1	Use of the open file button	The system will open a dialog file	Succeed
2	Use of the pitch reset button	The system sets the pitch value to+0	Succeed
3	Use of the tempo reset button	The system sets the tempo value to 100%	Succeed
4	Use of the play button	The system plays the selected audio file	Succeed
5	Use of the skip forward button	The system moves forward in 3 seconds.	Succeed
6	Use of skip backward button	The system moves back 3 seconds	Succeed
7	Use of the pitch seek bar button	The system changes the pitch of the audio file	Succeed
8	Use of the tempo seek bar button	The system changes the tempo of the audio file	Succeed
9	Use of the audio seek bar button	The system changes the playback time position of the audio file	Succeed

Figure 9 shows that the initial condition of the test is that the system displays the chord library page. This page has number picker chord buttons, number picker chord types, and play buttons. Table 5 is a test scenario for the chord library page.

Figure 10 shows the test's initial condition: the system displays the transpose and beat page. This page has buttons: open file, reset pitch, reset tempo, play, skip forward, skip backward, seek bar pitch, seek bar tempo, and seek bar audio. Table 6 is a transpose and beat page testing scenario.

To prove the accuracy of the tuning process, a test was conducted on the tuning process by comparing the results of the tuning process of the three applications, namely Learn Guitar Chords (LGC), Absolute Guitar, and Guitar Tuna as shown in Table 7. The test process carried out is by performing the tuning process by plucking the guitar strings in sequence, this process is carried out using the LGC application first then continued with the tuning process in the Absolute Guitar application, and then the Guitar Tuna application. The results obtained from the application comparison prove that the accuracy of the tuning process in the Learn Guitar Chords application is very good because it can produce the same results as other applications.

Users of the application or respondents also will be given a series of questions or statements as part of a data collection procedure called a questionnaire. The findings of the questionnaire technique experiment conducted with thirty respondents may attest to the application's high level of user-friendliness. Respondents claim that the application is also incredibly user-friendly, which can aid in the user's improvement of musical abilities.

Table 7. Tuning process on Redmi 2

Redmi S2

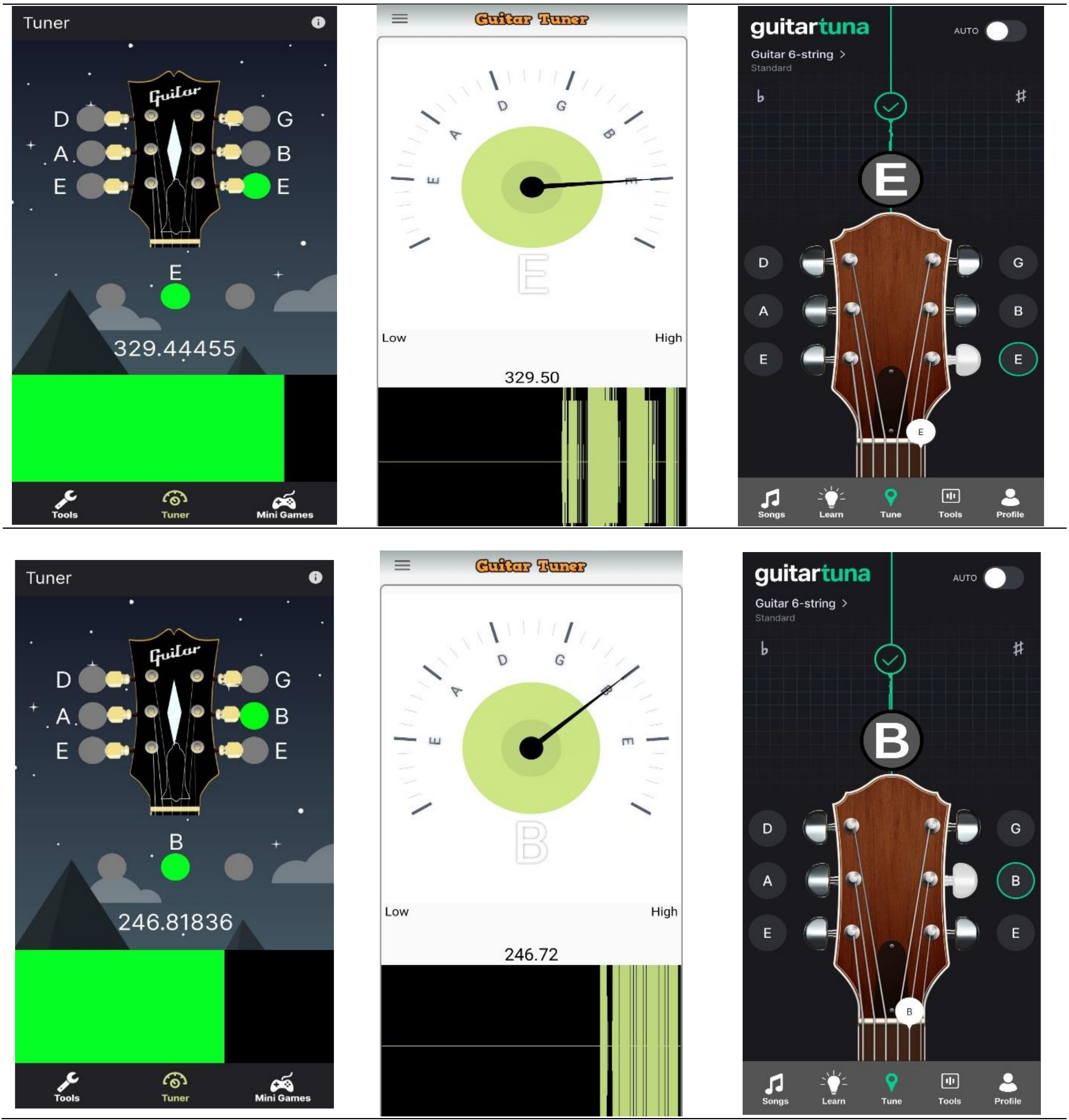

Table 8. Question topic in questionnaire

No.	Question Topic
1.	Users are interested in learning the guitar instrument.
2.	User experience in learning guitar instruments.
3.	User experience has tried similar applications.
4.	The user's opinion of the application's interface design.
5.	Accuracy of the tuner in the application.
6.	User opinion on the features provided.
7.	User experience about the application that has been used, whether all information provided or displayed on the application is clear and understandable.
8.	Ease of use of the application.
9.	Improved user guitar skills after using the application.
10.	Whether users will recommend the application to relatives.

Table 8 presents the 10 questions unrelated to the gender inquiry. The responses were analyzed using the Descriptive Statistics Frequency approach in the SPSS application. To test the accuracy and validity of a questionnaire, it is necessary to test its validity. The validity test is a test used to show the extent to which the measuring instrument used in a measure what is measured. The validity test of the questionnaire using an error rate of 5% and a free degree of 28 obtained from the number of respondents minus two so that $r_{tabel}=0.3610$, which explains that all questions are valid where $r_{count} \geq r_{tabel}$ for 10 question items. Furthermore, to ascertain whether the research questionnaire used to collect data on research variables is reliable or not, it is necessary to conduct a reliability test. From the results of the reliability test, it can be seen that the

Cronbach-alpha is 0.869 or 86.9% so it can be concluded that the reliability of the questionnaire has strong reliability if the Cronbach-alpha value is more than 0.8 [26].

Based on the results of testing conducted using a questionnaire to 30 respondents, the Learn Guitar Chords application is very useful and useful for users. application is very useful and useful for users:

- More than 70% of respondents support the idea that the application's interface design is attractive.
- More than 70% of respondents support that the application is informative so that it can provide useful information for users.
- More than 70% of respondents support the application's ability to improve musical skills, especially guitar.
- 100% of respondents support that respondents will recommend the Learn Guitar Chords application to fellow users.
- More than 80% of beginner musicians' respondents think that the Learn Guitar Chords application is very helpful in improving their musical skills.

Figure 10. Initial condition testing transpose and beat

5. CONCLUSION

Several steps are involved in tuning a guitar, including signal capture, frequency analysis, and error calculation. A microphone records the signal, which is then converted to an electrical signal. The fundamental frequency of the played note is then ascertained by analyzing the signal's frequency. Next, the system compares the measured frequency with the tuned string's needed frequency.

Several test scenarios have been put forth to verify the system, such as displaying different pages using different buttons, detecting input sounds using voice input, and displaying the sound frequency findings. The fundamental frequency of a particular signal has been found using the Fast Fourier Transform (FFT) method. This method could call for a large FFT size to provide the resolution required for a tuner. Even though there are a lot of guitar tuning apps available, most of them need the user to turn the string knobs manually. Research has been done to create an automatic guitar tuning system that can perform the tuning procedure automatically and deliver quick and accurate results.

The accuracy of the tuning process is done by comparing the results with 2 applications and proving that the tuning results with 2 applications have almost the same results. Suggestions for further development for tuner applications on musical instruments, especially guitars, can consider algorithms that are able to overcome the symptoms of the appearance of harmonic tones on each string and make it possible to use feature extraction methods to get more accurate results.

REFERENCES

[1] Hasan, M.M., Islam, S. (2022). Development of an acoustic guitar tuner and graphical user interface (GUI) using MATLAB. Applied Research and Smart Technology (ARSTech), 3(2): 49-55, https://doi.org/10.23917/arstech.v3i2.1185

[2] Venkataramanan, V., Koul, P., Kurani, P., Parmar, N. (2024). String Instrument Tuner using signal processing algorithms. In Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET-2024), pp. 1482-1492. https://doi.org/10.2991/978-94-6463-471-6_144

[3] Zhuo, P. (2021). Optimization of intelligent tuning system for stringed instruments based on wireless sensor network. Journal of Sensors, 2021(1): 6538185. https://doi.org/10.1155/2021/6538185

[4] Tampubolon, L., Nasution, T.H. (2023). Design of automatic guitar tuner using 8-bit microcontroller. In 2023 7th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM), Medan, Indonesia, pp. 311-314. https://doi.org/10.1109/ELTICOM61905.2023.10443179

[5] Šarga, P., Demečko, D. (2017). Design and realization of the guitar tuner using MyRIO. Journal of Automation and Control, 5(2): 41-45. https://doi.org/10.12691/automation-5-2-2

[6] Thakuria, U., Narayanan, S., Kumbhare, R. (2019). Musical instrument tuner. Asian Journal for Convergence in Technology (AJCT). http://www.asianssr.org/index.php/ajct/article/view/875.

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

[8] Pradana, A.B., Ramadhan, A.D., Aldila Fajar, Putra, J.T. (2021). Design of automatic guitar tuner with the open

tuning method based on a single board microcontroller. Science Tech: Jurnal Ilmu Pengetahuan Dan Teknologi, 7(2): 67-77. https://doi.org/10.30738/st.vol7.no2.a10296

[9] Branta, A.G., Martins, E.M., Quandt, V.I., Valente, S.A. (2019). Development of a triple input musical instrument tuner using Yin algorithm. International Journal of Advanced Engineering Research and Science, 6495(12): 380-384. https:// doi.org/10.22161/ijaers.612.40

[10] Stange, K., Wick, C., Hinrichsen, H. (2018). Playing music in just intonation: A dynamically adaptive tuning scheme. Computer Music Journal, MIT Press, 42(3): 47-62. https://doi.org/10.1162/comj_a_00478

[11] Kumar, B.D., Kushwaha, A., Kumar, A., Agarwal, A. (2021). Design & implementation of digital guitar tuner using MATLAB. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 547-549. https://doi.org/10.1109/ICACITE51222.2021.9404728

[12] Tirasuntarakul, N., Dheeravongkit, A. (2018). An automatic multi-string musical instrument tuner using one-to-many micro actuating mechanism. In Proceedings of the 2018 International Conference on Machine Learning and Machine Intelligence, Ha Noi, VietNam, pp. 64-67. https://doi.org/10.1145/3278312.3278323

[13] Saji, A.K., Regeena, M. (2009). Digital guitar tuner. International Journal of Computer Science and Information Security, 6(2): pp. 82-88, https://doi.org/10.48550/arXiv.0912.0745

[14] Villaran-Valdivia L (Inventor). (2014). Automatic guitar tuner. United States patent US. https://patents.google.com/patent/US20140033893A1/en.

[15] Gomathy, S., Maheswari, P.M., Aparna, P., Prabhakaran, S, Kumar, N.Y. (2022). A review on guitar tuners. International Journal of Research and Analytical Review (IJRAR), 9(2): 4-10.

[16] Kumar, A., Srivastava, S., Chandra, M., Sahoo, G. (2018). Guitar tuner using cepstral analysis and fuzzy controller on Arduino board. Microsystem Technologies, 24: 2429-2436. https://doi.org/10.1007/s00542-017-3623-2

[17] Burger, B., London, J., Thompson, M.R., Toiviainen, P. (2018). Synchronization to metrical levels in music depends on low-frequency spectral components and tempo. Psychological Research, 82: 1195-1211. https://doi.org/10.1007/s00426-017-0894-2

[18] Weineck, K., Wen, O.X., Henry, M.J. (2022). Neural synchronization is strongest to the spectral flux of slow music and depends on familiarity and beat salience. Elife, 11: e75515. https://doi.org/10.7554/eLife.75515

[19] Mao, X.B., Ma, Z.G., Yu, F., Xing, Q.J. (2017). A continuous-flow memory-based architecture for real-valued FFT. IEEE Transactions on Circuits and Systems II: Express Briefs, 64(11): 1352-1356. https://doi.org/10.1109/TCSII.2017.2683642

[20] Majorkowska-Mech, D., Cariow, A. (2022). Some FFT algorithms for small-length real-valued sequences. Applied Sciences, 12(9): 4700. https://doi.org/10.3390/app12094700

[21] Yuan, M., Ma, Z., Yu, F., Xing, Q. (2019). A novel address scheme for continuous-flow parallel memory-based real-valued FFT processor. Electronics, 8(9): 1042. https://doi.org/10.3390/electronics8091042

[22] Jannereth, E., Esch, L. (2021). Analyzing timbres of various musical instruments using FFT and spectral analysis. Journal of Student Research, 10(1). https://doi.org/10.47611/jsrhs.v10i1.1292

[23] Becchi, N.G. (2017). Sistema automatizado para afinação de violão elétrico. UNIVATES. https://www.univates.br/bdu/bitstream/10737/1664/1/2017NatanGabrielBecchi.PDF.

[24] Shin, M., Hong, W., Lee, K., Choo, Y. (2021). Frequency analysis of acoustic data using multiple-measurement sparse bayesian learning. Sensors, 21(17): 5827. https://doi.org/10.3390/s21175827

[25] Hamdan, S., Musib, A.F., Sawawi, M., Othman, S.H. (2021). The frequency spectrum and time frequency analysis of different violins classification as tools for selecting a good-sounding violin. Wacana Seni Journal of Arts Discourse, 20: 27-40. https://doi.org/10.21315/ws2021.20.3

[26] Wijaya, M.C., Kloping, Y.P. (2021). Validity and reliability testing of the Indonesian version of the eHealth Literacy Scale during the COVID-19 pandemic. Health Informatics Journal, 27(1): 1460458220975466. https://doi.org/10.1177/1460458220975466