Presented to the **Gokongwei College of Engineering Department**

De La Salle University - Manila

A.Y. **2022 - 2023**

In partial fulfillment

of the course

Signals, Spectra and Signal Processing Laboratory (LBYEC4A - EK2)

Design and Implementation of a Guitar Tuner using MATLAB

Submitted by:

DAYRIT, Bettina Gaille H.

GUEVARRA, Gia Kyla S.

RODRIGUEZ, Mariah Venice A.

Submitted to:

Mr. Stephen Ruiz

April 14, 2023

I. ABSTRACT

The use of a guitar tuner is essential for guitar players to ensure the accuracy of their notes when playing their instruments. This project aims to develop a digital guitar tuner using digital signal processing (DSP) principles, such as filtering, sampling interval, and Fast Fourier Transform (FFT). The guitar tuner is designed and implemented in MATLAB and offers five tuning modes: standard tuning, drop D tuning, double drop D, DADGAD tuning, and open D tuning. The tuner compares the input audio signal to the standard frequencies of the guitar notes, and then converts the signal to the frequency domain using FFT. The signal is then filtered, and the tuning is indicated by the results of the frequency comparison found in the input audio signal.

The students were successful in developing a digital guitar tuner based on DSP principles, which proved to be accurate in detecting correct notes and indicating whether they were in tune or not. The tuner has five different tuning modes with a total of 10 notes available for checking, all of which passed the testing phase and met the objectives of the project. However, one of the limitations of the project is that it cannot accommodate all harmonics because it is limited to one harmonic only. Nonetheless, the project demonstrates the practical application of DSP in audio processing and achieved its goal of developing a digital guitar tuner.

II. INTRODUCTION

Guitar tuning is a fundamental aspect of playing the guitar, and it is an essential skill that every musician must be proficient with. Tuning a guitar guarantees that the strings are adjusted at the correct pitches for chords, scales, and riffs to sound right [1]. A guitar that is not properly tuned can sound off-key and incorrect. Digital guitar tuners are available in a variety of designs, including pedal tuners, clip-on tuners, and digital applications [2]. Each variety has advantages and limitations, but they are all highly accurate and can tune to many tuning patterns. This allows guitar players to experiment across numerous musical styles and genres.

In the past, guitar players relied on traditional methods, such as tuning forks or pitch pipes, to tune their instruments [3]. However, these methods were often inaccurate and time-consuming.

As a result of technological advancements, digital guitar tuners have become increasingly popular among guitar players. They offer a reliable and effective method for tuning guitars using advanced algorithms and signal processing techniques. These tuners can correctly identify the standard frequencies of guitar notes and provide visual feedback, such as LED lights or LCD screens, to show whether a string is in tune [4].

III. THEORETICAL CONSIDERATION

Digital tuners are electrical devices that are used to precisely tune musical instruments. These devices work by analyzing the frequency of a sound generated by an instrument and comparing it to a list of fixed frequency values that correspond to specific notes [5]. These types of tuners are commonly utilized in a variety of musical instruments such as guitars, violins, and others. They are known for their great precision and ease of use, in comparison to traditional tuners. The focus for this paper is exclusively on the digital tuner designed specifically for guitars.

A guitar can be tuned using a variety of tuning modes, and each mode has a distinctive sound and set of properties. The most common tuning mode is standard tuning, in which the six strings are tuned to E, A, D, G, B, and E notes from lowest to highest string. On the other hand, drop D tuning lowers the tuning of the lowest string from E to D, creating a deeper and heavier sound. Double drop D tuning is similar to drop D tuning but with the addition of lowering the tuning of the highest string from E to D as well. Next, DADGAD tuning is a popular alternative tuning mode in which the guitar is tuned to D, A, D, G, A, and D notes from lowest to highest string, producing a distinct and open sound. Lastly, open D tuning, in which the guitar is tuned to D, A, D, F, A, and D notes from the lowest to the highest string, providing a rich and resonant tone.

The Fast Fourier Transform (FFT) is a commonly utilized mathematical technique which converts a signal from the time domain to the frequency domain [7]. The Fourier transform can be particularly beneficial in understanding typical signals and correcting errors in the signals. FFT decomposes a signal into multiple sinusoidal waves of different frequencies and amplitude [7-a]. By analyzing the signal in the frequency domain, it is possible to better understand its properties and characteristics, such as the dominant frequencies, harmonic components, and noise.

The Fourier transform is a mathematical tool for converting a signal into its frequency domain representation. In practice, however, it is not practical to examine a full signal at once because it is frequently infinite or hard to comprehend. As a result, windowing is required, which entails separating the signal into smaller parts or windows and then analyzing each window independently using the Fourier transform [7-b]. Windowing functions are mathematical functions that are applied to a signal to lessen the distortion produced by finite length processing [8]. Blackman-Harris and Hanning are two windowing functions used in digital signal processing. The Hanning window is a commonly used windowing function that tapers the edges of a signal to zero [7-c]. On the other hand, the Blackman Harris window has a more complex windowing function than Hanning. This type of windowing has a sharper transition than Hanning. It is intended to prevent spectral leakage, or the transfer of signal energy from one frequency bin to another in the frequency domain [7-d].

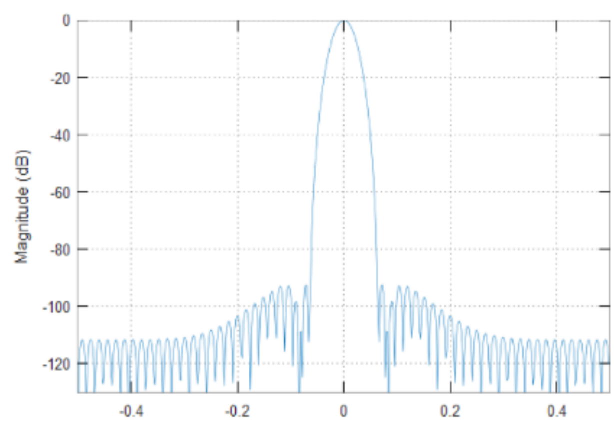


Figure 3.1. Blackman-Harris Window Waveform [9]

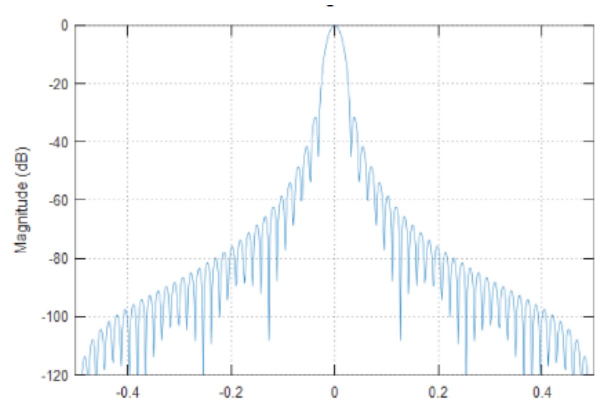


Figure 3.2. Hanning Window Waveform [9]

The figures above show a sample waveform for Blackman Harris and Hanning window waveforms. These two windows are used in audio processing applications, such as speech recognition, audio filtering, and musical signal processing. In the context of the guitar tuner project, Blackman-Harris and Hanning windows were utilized to enhance the accuracy of frequency analysis by minimizing spectral leakage and increasing the differentiation between frequency peaks.

IV. METHODOLOGY

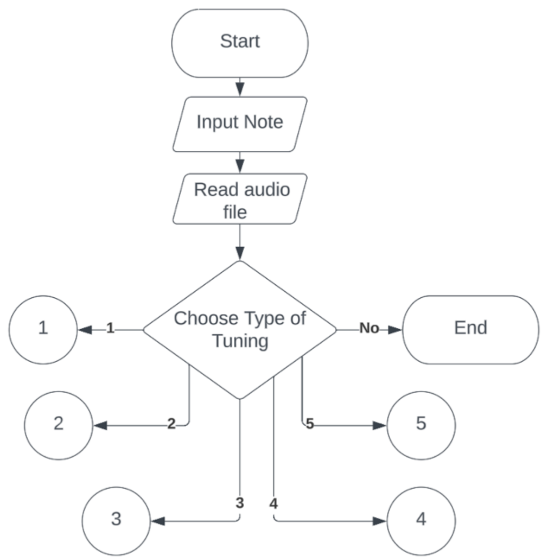


Figure 4.1. General Flowchart

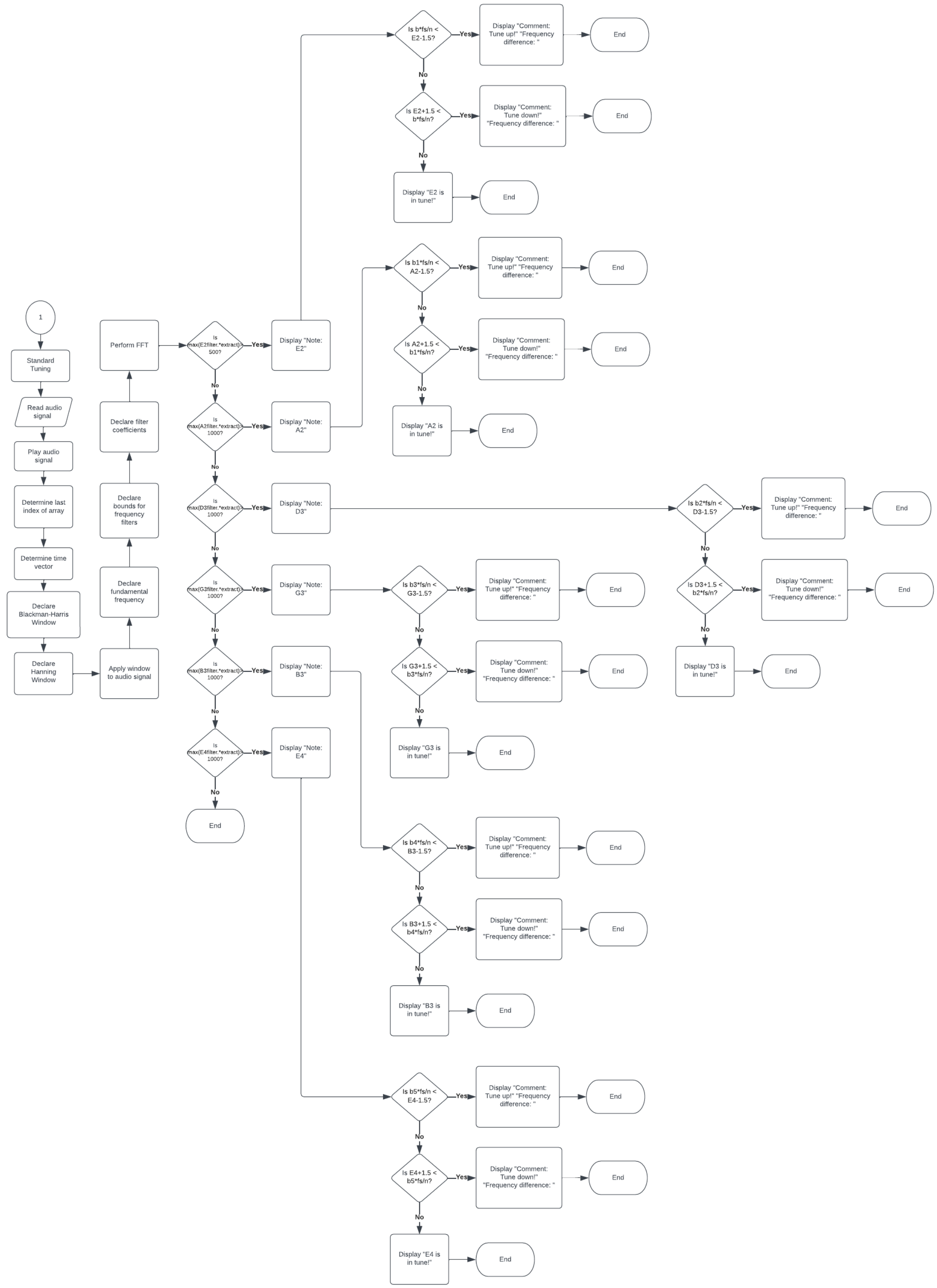


Figure 4.2. Flowchart for Standard Tuning

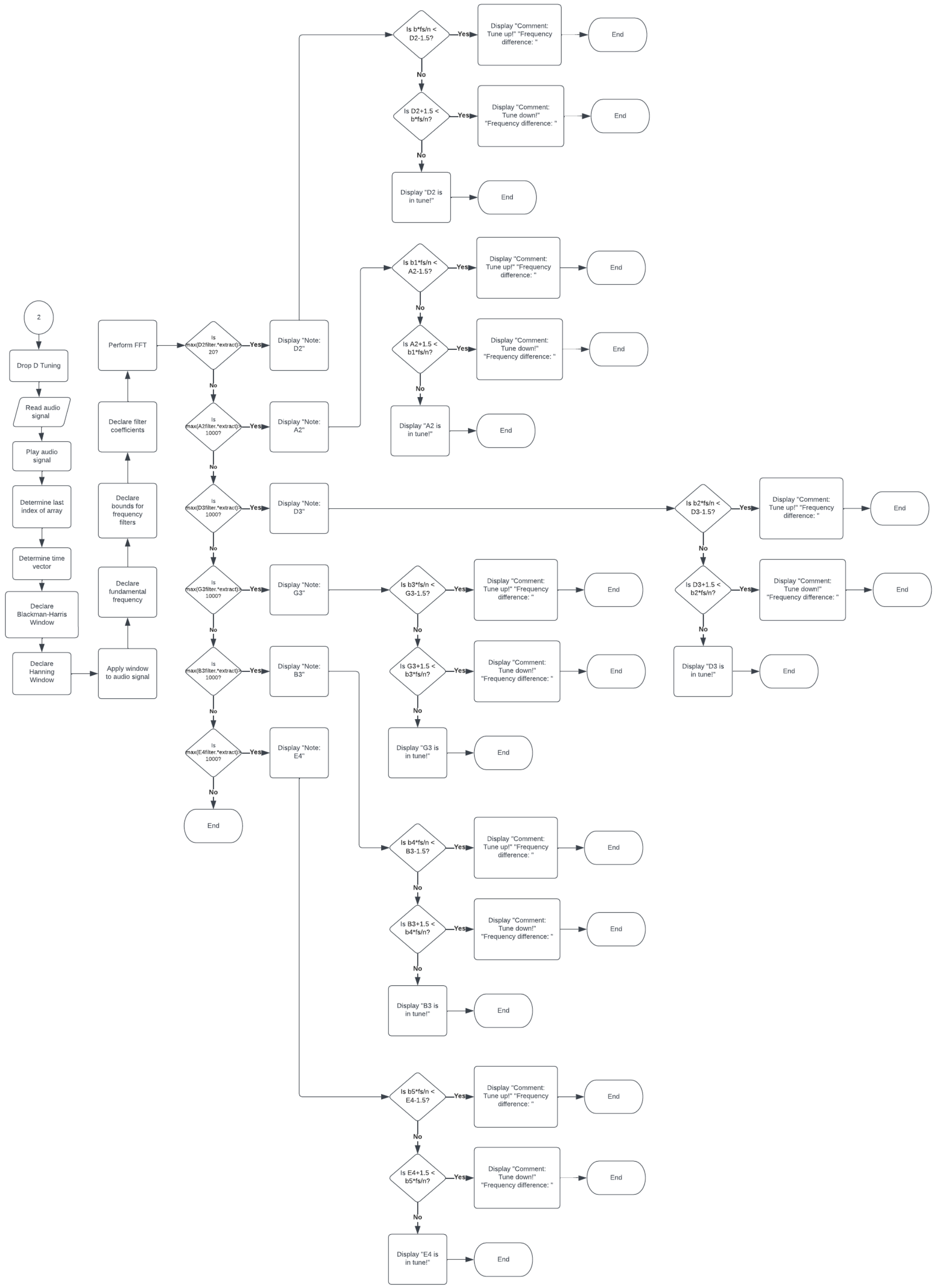


Figure 4.3. Flowchart for Drop D Tuning

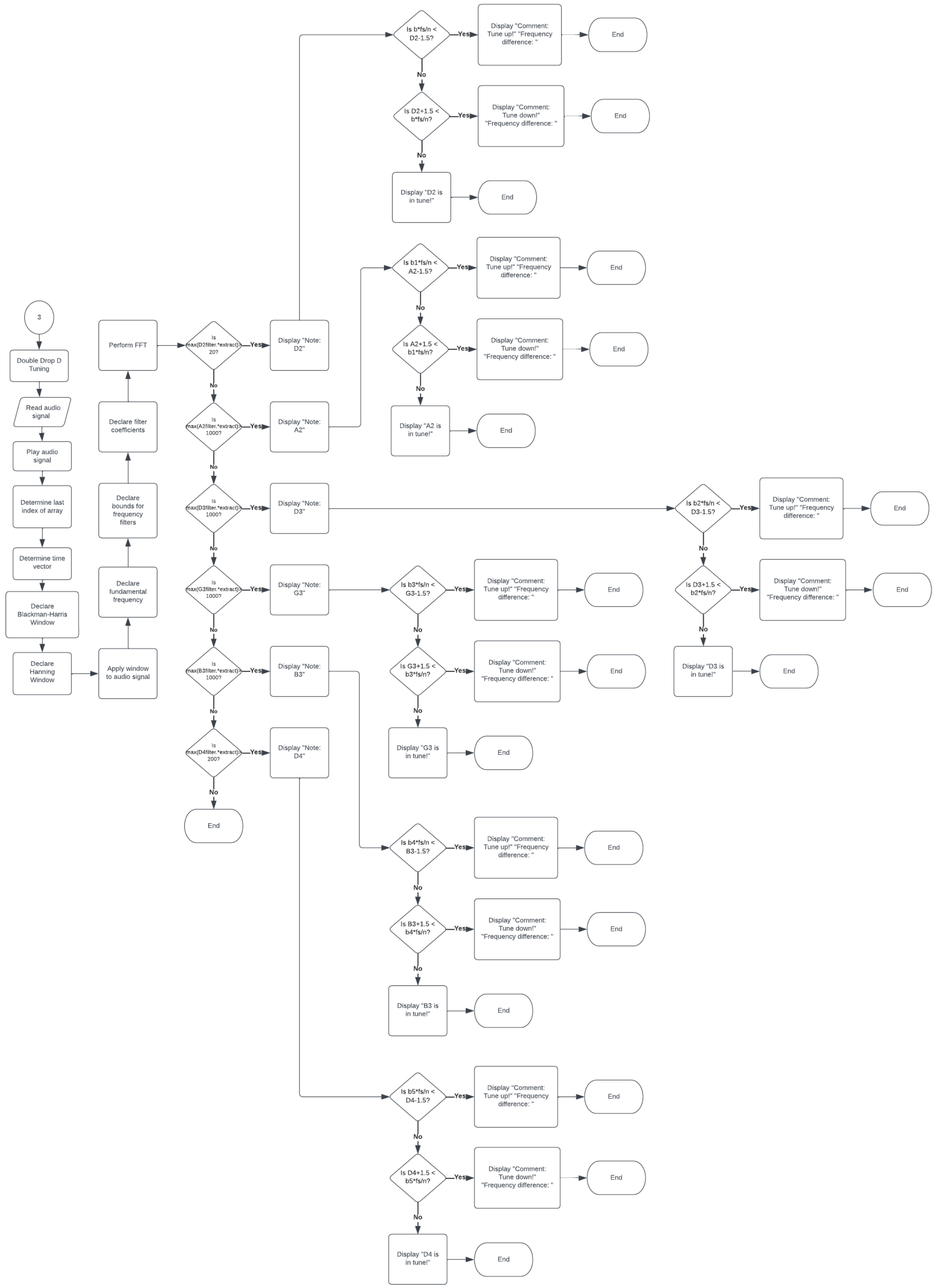


Figure 4.4. Flowchart for Double Drop D Tuning

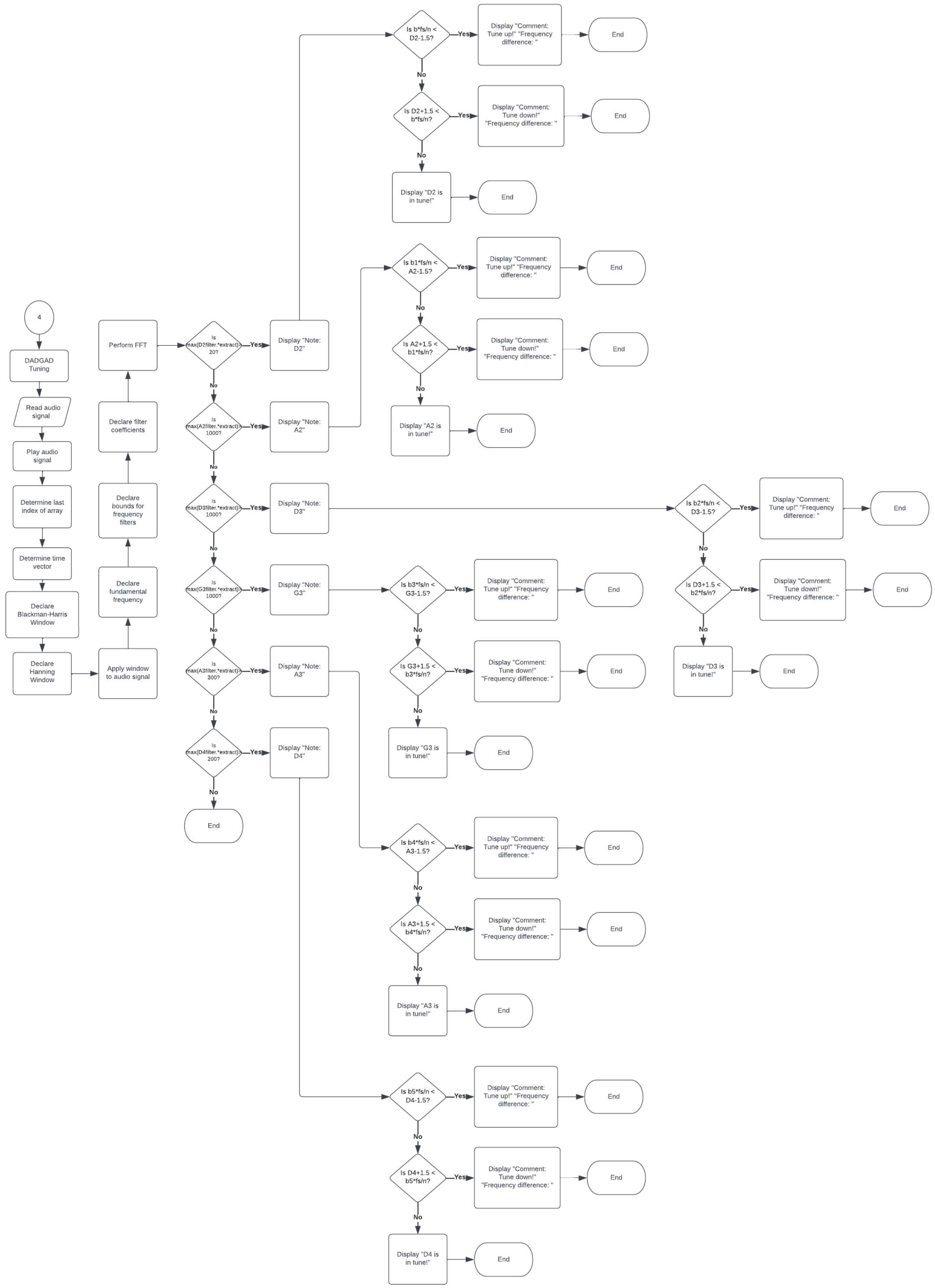


Figure 4.5. Flowchart for DADGAD Tuning

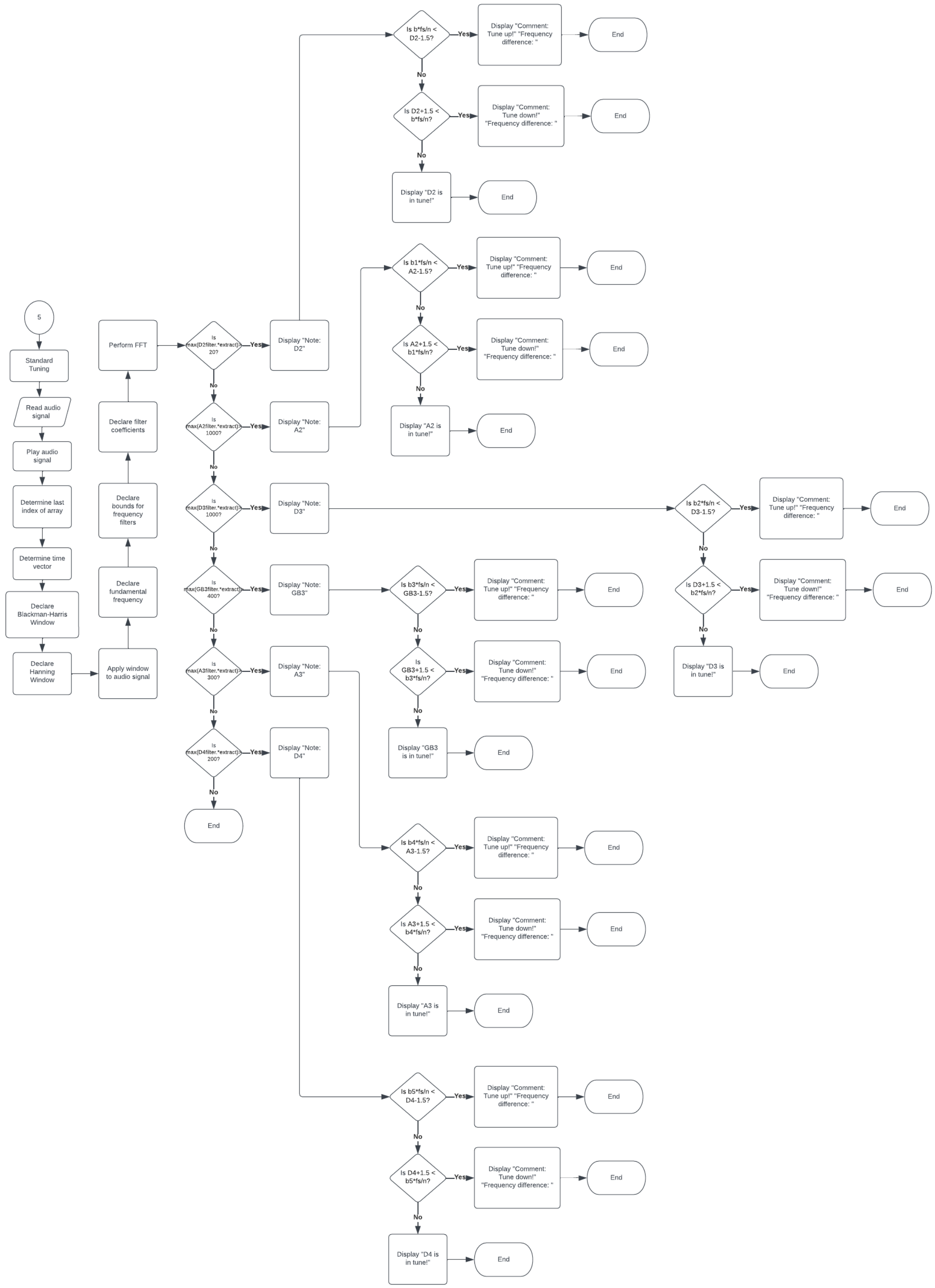


Figure 4.6. Flowchart for Open D

In making the guitar tuner, the code was made for the user to input the file name and file type of the audio signal to be used. After entering the audio signal, the code is meant to read the inserted signal using the function *audioread()* and plot the waveform of the audio signal. After the plot of the audio signal, the code is meant to ask which type of tuning would the user want to use. This is where the function of each tuning modes are considered. A nested if-else statement is utilized to make the user choose which type of tuning will be executed. If the user chooses 1, the code will execute Standard tuning, 2 for Drop D tuning, 3 for Double Drop D tuning, 4 for DADGAD, and 5 for Open D tuning. If the user enters the wrong input, the code will display the message “Wrong input!”.

The function for Standard tuning starts with reading the audio signal using *audioread()* command. The inserted audio will be played and its waveform will be plotted after windowing. Blackman-Harris and Hanning window are performed to make the audio signal intelligible when the filtering process comes. Generally, windowing is used to force the signal to be periodic in time record. This is because FFT assumes that the signal it processes is periodic for every time record. If the signal is not periodic, FFT will not assume the frequency components accurately. Hanning window is used to attenuate the audio signal at both ends of its time duration to zero. Blackman-Harris is utilized as well as it has a relatively high dynamic range for the main lobe and this can do more attenuation at side lobes. After plotting the audio signal, the fundamental frequencies and bounds of the frequency filters are declared into MATLAB. The input parameters are referenced from the notes used for standard tuning which are E2, A2, D3, G3, B3, E4. After declaring the parameters to be referenced, the filter coefficients are also coded and FFT is to be applied to the audio signal.

The next part of the code is for checking which note the audio signal is fit in. The bounds declared earlier serves as a tolerance for each note that will categorize the input signal. If it’s less than the frequency of the note, it will display “Tune up!” as an indication that the audio signal inserted must be tuned up to fit with the frequency of the note that it is reading. The frequency difference is also displayed for the user to see. If the audio signal inserted is greater than the note it’s reading, the program will display “Tune down!” as well as its frequency difference. If the audio signal inserted is in tune, the program will then display “(Note) is in tune!”. The “note” part in the display shows which note the audio signal is tuned. This flow of tuning is also the same for the other tuning modes, the only difference are the notes referenced for tuning. For Drop D tuning, the notes used are D2, A2, D3, G3, and E4. Double Drop D tuning uses D2, A2, D3, G3, and D4. DADGAD tuning uses D2, A2, D3, G3, A3, D4. Lastly, Open D uses D2, A2, D3, GB3, A3, D4.

V. RESULTS

The following section details the results of the project codes,

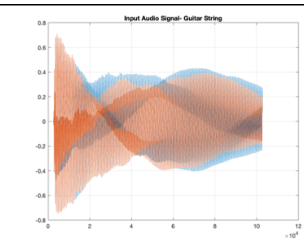
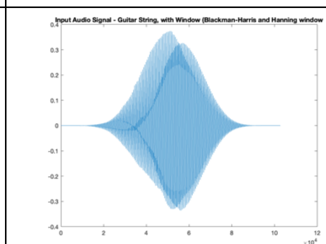
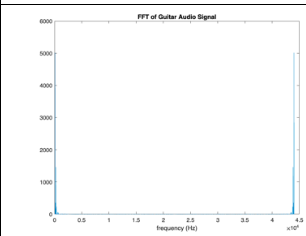
Note	Plot of Input Audio Signal	Plot of Input Audio Signal with Window	FFT of Input Audio Signal
E2			

Table 5.1. Plots from Note E2 Testing using Standard Tuning

```
Note: E2
E2 is in tune!
```

Figure 5.1.Tuning Result from E2 Testing Using Standard Tuning

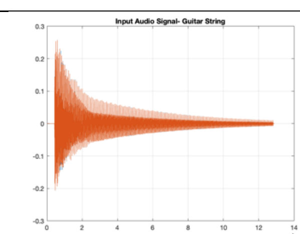
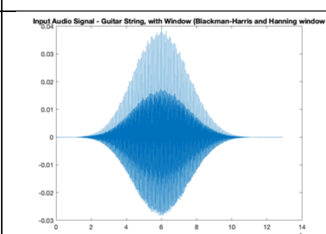
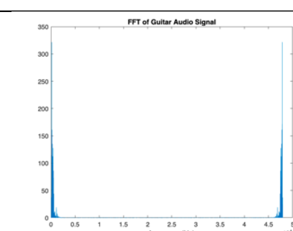
Note	Plot of Input Audio Signal	Plot of Input Audio Signal with Window	FFT of Input Audio Signal
D2			

Table 5.2. Plots from Note D2 Testing using Drop D Tuning

```
Note: D2
D2 is in tune!
```

Figure 5.2. Tuning Result from D2 Testing Using Drop D Tuning

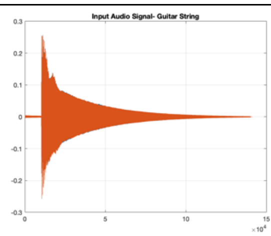
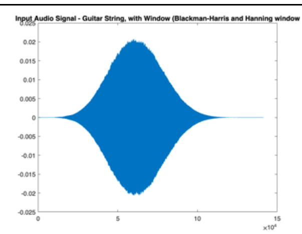
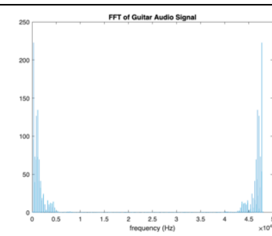
Note	Plot of Input Audio Signal	Plot of Input Audio Signal with Window	FFT of Input Audio Signal
D4			

Table 5.3. Plots from Note D4 Testing using Double Drop D Tuning

```
Note: D4
D4 is in tune!
```

Figure 5.3. Tuning Result from D4 Testing Using Double Drop D Tuning

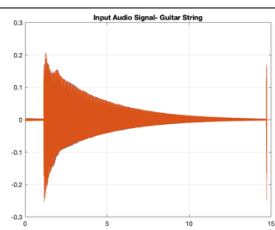
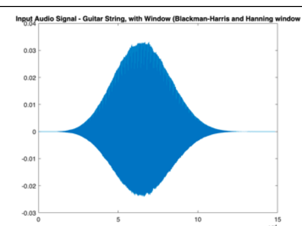
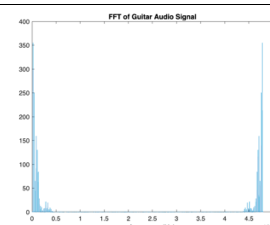
Note	Plot of Input Audio Signal	Plot of Input Audio Signal with Window	FFT of Input Audio Signal
A3			

Table 5.4. Plots from Note A3 Testing using DADGAD Tuning

```
Note: A3
A3 is in tune!
```

Figure 5.4. Tuning Result from A3 Testing Using DADGAD Tuning

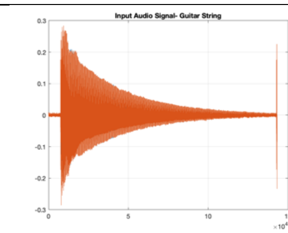
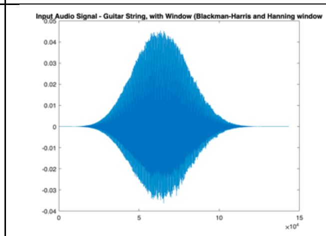
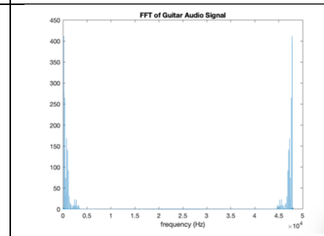
Not e	Plot of Input Audio Signal	Plot of Input Audio Signal with Window	FFT of Input Audio Signal
GB3			

Table 5.5. Plots from Note GB3 Testing using Open D Tuning

```
Note: GB3
GB3 is in tune!
```

Figure 5.5. Tuning Result from GB3 Testing Using Open D Tuning

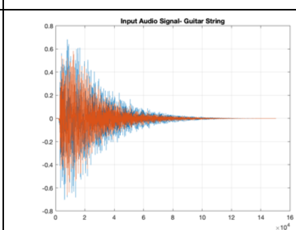
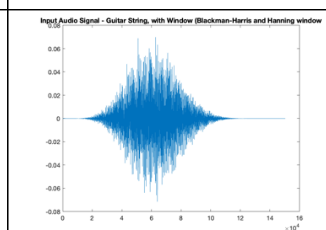
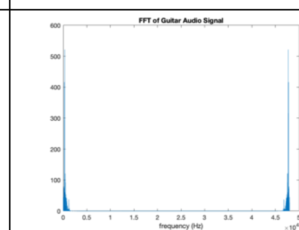
Note	Plot of Input Audio Signal	Plot of Input Audio Signal with Window	FFT of Input Audio Signal
Out of Tune			

Table 5.5. Plots from Out of Tune Testing using DADGAD Tuning

```
Note: A3
Comment: Tune up!
Frequency difference:
    3.4865
```

Figure 5.5. Tuning Result from Out of Tune Testing Using DADGAD Tuning

VI. DISCUSSION

Upon finishing the codes simulation for the guitar tuner, based on the results, the group achieved the main goal of the project: to design and implement a guitar tuner using MALTAB. With the 5 different modes, if-else statements were utilized to differentiate which function of tuning to use, wherein the interface is user dependent as it prompts the user to input the guitar string audio signal to be tuned, and it gives the freedom to choose which tuning mode they will utilize.

The structure of each tuning mode is almost the same, first, the audio signal is read, and the audio signal is played. Then, because the project is frequency-dependent, it needs a clear distinction for the peaks in the frequency, therefore, windows are design for the input signal to pass through, namely Blackman-Harris and Hanning windows. These window filters are used because it provided distinctions within the frequency peaks as compared to other filters and windows.

As for the declaration of the standard frequencies, the group has analyzed that there needs to be a range upon checking, hence the declaration of bounds to be used in the frequency filters so that the frequency has a tolerance to follow. For example, an input audio with a frequency of 85 will still be applicable in E2 for tune down and A2 for tune up. With this, the detected frequency is allowed to have a slight deviation from the standard frequencies of the note.

One of the most important process in the tuning proper is the use of FFT in the input audio signal. From theory, FFT is used when converting from time domain to frequency domain; with this, FFT is needed in the frequency conversion of the input audio signal. This is because as declared in previous discussions, the project is heavily frequency-based since it compares frequency to declare the notes and the tuning process. Moreover, with FFT, the fundamental frequency is determined—one of the key components in the tuning proper. Lastly, if-elseif-else statements with a tolerance of 1.5 are utilized in the checking of the frequency per note. Essentially, it uses the note filters multiplied by the FFT of the signal to compare. This part of the coding process is the most tricky as it uses a threshold amplitude as a condition. With this, various factors have to be considered because of the nature of an acoustic signal: harmonics.

Results show that the notes played has displayed great accuracy, detecting the correct notes and declaring if it is in tune or not. Since the codes are of 5 different tuning modes, there are 10 distinct notes available for checking—all of which also passed the testing phase, and ultimately achieving the objectives set for the project. Observed limitations to this project however, include, the concept of harmonics. The code is only limited to one harmonic: Since there are different harmonics present in a guitar string, the codes cannot accommodate all frequencies of different harmonics, resulting in an unsuccessful tuning process wherein it is not able to detect the note, therefore, there is no display of a note and its tuning process.

VII. CONCLUSION

In doing this design project, the group was able to utilize their knowledge in using the MATLAB software to create the codes for the guitar tuner as well as their general knowledge of audio signal processing. Specifically applying filters and windowing to the inserted audio signal. The group was able to meet the objectives set for this design project – which is to design a digital guitar tuner with 5 different modes in MATLAB and show if the inserted string is too sharp, too flat, or in tune. Since the guitar has 5 different tuning modes, this makes the tuner flexible for the preference of the user. However, certain limitations apply for every project designed. As for this tuner, the limitation is that it only deals with one harmonic. Given this limitation, this digital guitar tuner leaves a room for improvement and innovation if there are other researchers that will reference this tuner.

VIII. AUTHOR'S CONTRIBUTION

Member	Contribution
DAYRIT, Bettina Gaille H.	Result and Discussion, Codes
GUEVARRA, Gia Kyla S.	Abstract, Introduction, Theoretical Considerations, Codes
RODRIGUEZ, Mariah Venice A.	Methodology, Conclusion, Codes

IX. REFERENCES

[1]M. & Arts, “Guitar Tuning for Beginners,” *The Vault at Music & Arts*, Sep. 25, 2020. https://thevault.musicarts.com/guitar-tuning-for-beginners/ (accessed Apr. 14, 2023).

[2]J. F. last updated, “Best Guitar Tuners 2023: 8 Great Clip-On, Pedal and App-based Tuning Options For All Guitarists,” *Guitar Player*, Sep. 02, 2022. https://www.guitarplayer.com/gear/best-guitar-tuners (accessed Apr. 14, 2023).

[3]“Guitar Tuner Types? IMPORTANT Options For Your Setup! - Guitar Skills Planet,” *guitarskillsplanet.com*, Nov. 06, 2020. https://guitarskillsplanet.com/guitar-tuner-types/ (accessed Apr. 14, 2023).

[4]“Tune Your Guitar,” *strettonpayne.com*. https://strettonpayne.com/how-to-tune-your-guitar/#:~:text=What%20is%20a%20digital%20tuner

[5]MasterClass, “What Is a Guitar Tuner? Learn How to Tune Your Guitar With Instrument Tuners,” 2021. https://www.masterclass.com/articles/what-is-a-guitar-tuner-learn-how-to-tune-your-guitar-with-instrument-tuners

[6]“Tuning Tip: Standard Tuning,” *www.fender.com*. https://www.fender.com/articles/setup/tuning-tip-standard-tuning (accessed Apr. 14, 2023).

[7]“Understanding FFTs and Windowing Overview.” Available: https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf

[8]“Product Documentation - NI,” *www.ni.com*. https://www.ni.com/docs/en-US/bundle/labview/page/lvanlsconcepts/windowing_signals.html#:~:text=Use%20smoothing%20windows%20to%20improve (accessed Apr. 14, 2023).

[9]K. Gharaibeh, “Assessment of Various Window Functions in Spectral Identification of Passive Intermodulation,” *Electronics*, vol. 10, no. 9, p. 1034, Jan. 2021, doi: https://doi.org/10.3390/electronics10091034.

[10]“An introduction to alternate tunings: drop D, open D major and DADGAD,” *Guitar.com | All Things Guitar*. https://guitar.com/lessons/introduction-alternate-tunings-drop-d-open-d-major-dadgad/ (accessed Apr. 14, 2023).

[11]“How to tune the guitar to standard tuning,” *www.guitar-chord.org*. https://www.guitar-chord.org/articles/standard-tuning.html

[12]Marty, “Double Drop D Tuning - DADGBD,” *The Acoustic Guitarist*, Jun. 10, 2020. https://theacousticguitarist.com/double-drop-d-tuning-dadgbd/ (accessed Apr. 14, 2023).