# LBYEC4A – EK3

***Signals, Spectra and Signal Processing Laboratory***

**Final Project Proposal**

## Design and Implementation of a Guitar Tuner of using MATLAB

Bettina Gaille H. Dayrit
Gia Kyla S. Guevarra
Mariah Venice A. Rodriguez

## PROJECT DESCRIPTION

The development of this project involves the application of digital signal processing principles that were covered both in the laboratory and lecture course. Various concepts, such as filtering, sampling interval, and the Fast Fourier Transform (FFT), were employed to design and implement an effective guitar tuner. In addition, the project explores the audio processing features available in MATLAB.

The standard frequencies of the guitar notes are included in the MATLAB code for the guitar tuner and are compared to the input audio signal for tuning purposes. Furthermore, there are two modes available for this project: standard tuning and drop d tuning. Regular tuning involves tuning the guitar strings to the notes E, A, D, G, B, and E, beginning with the lowest (thickest) string and progressing to the highest (thinnest) string. The lowest string is tuned down one whole step (two frets) from E to D in drop d tuning, while the other strings remain in standard tuning. The output of the code indicates whether the use should tune up or down depending on the sharpness and the flatness of the sound, enabling them to make necessary adjustments to achieve the desired tune for the specific string being tuned.

Frequency is denoted as the number of cycles per second. The unit of frequency is Hertz (Hz), named after the French physicist Heinrich Hertz. In signal processing, frequency is a key factor as it is present most of the time when processing signals. Frequency has the capability to describe the characteristic of a signal and can be used as an input parameter to other signal processing techniques [1]. Audio signals contains unwanted noises that are normally filtered out. However, filtering an audio signal can also remove the information needed while filtering the unwanted noises. With this, there are different ways to filter an audio signal without removing or filtering out the necessary information inside the signal. The common digital filters used are Butterworth, Chebyshev Type 1 and 2, and Elliptic filters. It is important to analyze the signal first to know which type of filter to be used as the parameters for each filter is dependent on the information of the signal to be processed. Butterworth filter has a nominally flat frequency response in the passband – which makes it ideal to be used in processing audio signals. Chebyshev filters allows ripples in the passband, but this makes the roll-off to be faster compared to Butterworth. Lastly, elliptic filters contain ripples both in its passband and stopband. Elliptic filters are mostly used if very quick transitions are needed between the passband and stopband [2]. In audio signals, sampling rate is the parameter obtained from a continuous signal that is used to create discrete or digital signal. The Nyquist-Shannon theorem or Nyquist principle is used to compute for the sampling rate in a signal. This theory describes that the sampling frequency must be greater than twice the maximum frequency of the signal sampled. Using the Nyquist theorem to determine the sampling rate lessens the chances of aliasing and improves the reconstruction of the signal [3]. In digital signal processing, spectral analysis is widely used to see the frequencies present in a signal. During the early times, analog filters were used to break down the signal into different frequency components. In the modern age, spectral analysis is performed through various digital solutions and one of these is FFT or Fast Fourier Transform. The way FFT works is by breaking down the signals and performing repeated multiplication. These redundant steps of computation are organized by the algorithm and later on factored out in terms of matrices. Through FFT, the repeated computations are eliminated which makes the overall calculations faster [4].

With the theoretical concepts discussed, this will then be utilized in the project to achieve the intended application, a digital guitar tuner. Through the use of MATLAB, the tuner is constructed using its filtering, measurement, and analysis capabilities. Initializations that is required for the input audio signal (pluck of the guitar string to be tuned) are its resolution with respect to the sampling rate. This is because a low resolution affects the accuracy of the frequency estimation. With this, a study by Lourde and Saji declares that a suitable resolution of 0.5Hz is suitable for the purposes if designing a guitar tuner. [5] With the declaration of the standard frequencies of the open string of a guitar and the drop d tuning standard frequencies, this will be the main comparator to the recorded input audio signal by the user. However, an input audio signal is in the time domain, to properly compare this signal to the frequencies of the guitar, it is to be converted to the frequency domain using Fast-Fourier Transform (FFT). [6]

After the conversion to the frequency domain, unwanted noise will be expected in the signal, so this will be passed through a filter; the maximum and minimum frequencies of the guitar strings will be taken into account as to make sure that these frequencies will not be filtered. [5-a] Lastly, the results of the compared frequency found in the input audio signal of the guitar vs. the standard frequencies of the open guitar string will be the indicator of the tuning. All of which are applied during the project proper to ensure that the intended application is achieved—as with the references cited in this section, the basic implementation of a guitar tuner has shown success across other studies; with the project's main differentiator as having two modes: regular tuning and Drop D tuning, this will be possible through the concepts discussed and applied in MATLAB.

## *METHODOLOGY*

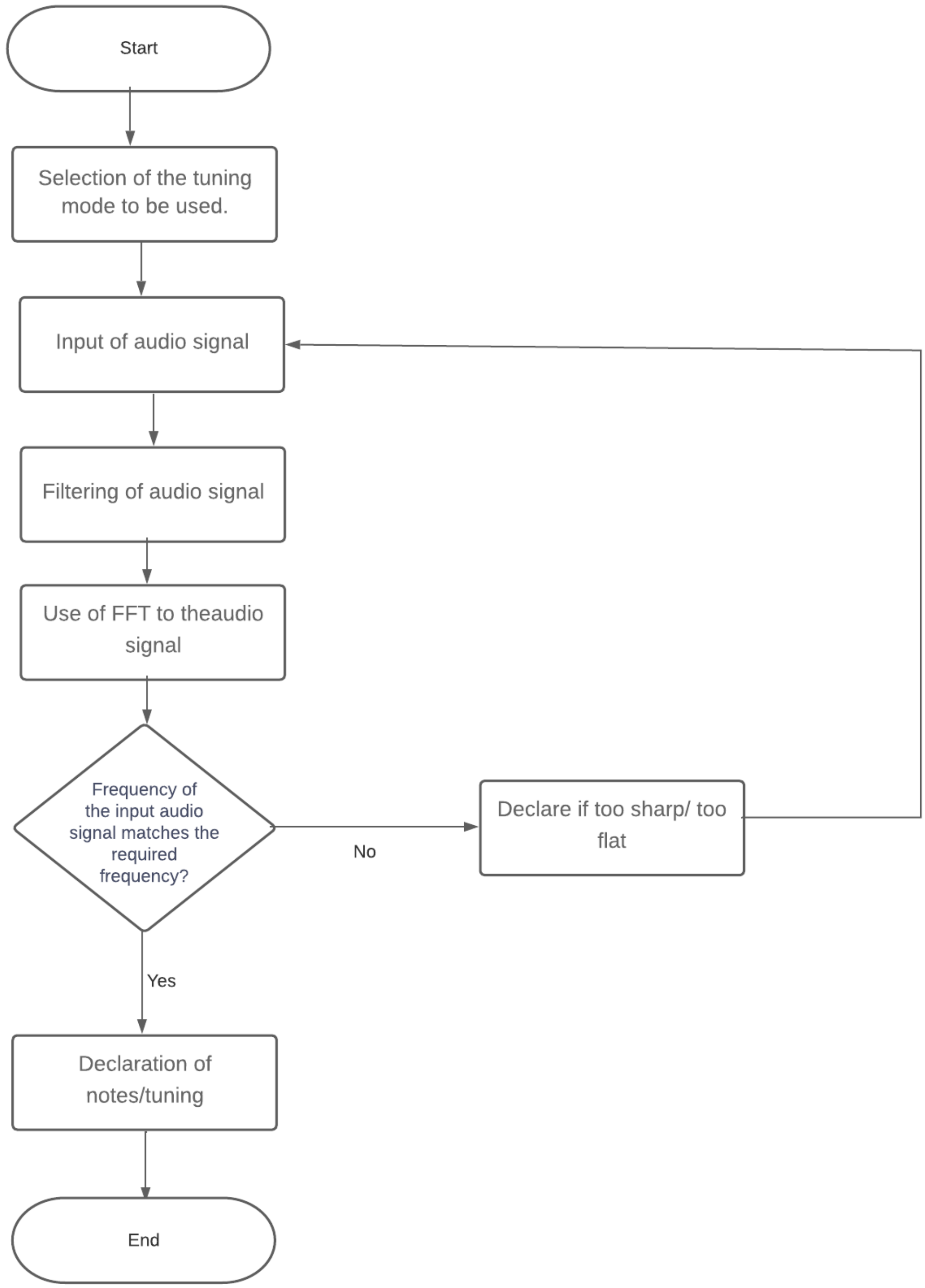


Figure 2.1. Flowchart of the Project Methodology

Overall, the group designs the project methodology to be the following:

1. Selection of the tuning mode to be used.
2. Recording and the input of the individual guitar string to the code.
3. The input audio signal is to be passed through a filter to eliminate unwanted noise and focus solely on frequencies within the range of the standard frequency of the guitar strings.
4. Use of FFT in the audio signal to transform the domains from time to frequency; to also determine the fundamental frequency of the signal.
5. Comparison of the frequency in the filtered signal and the standard frequency of the guitar strings.
6. If frequency is not in tune, output the declaration if the frequency is too sharp or too flat, else, when the frequency is in tine, output the declaration of the note in tune.

Description of Digital Signal Processing concepts that will be used to develop the project:

A. Filters
    - Before processing the input signal using FFT and other techniques, a bandpass filter is used to filter the unwanted noise present in the signal. In this design project, a Butterworth filter will be used using the *butter* function. A Butterworth filter has a flat frequency response in its passband. The specific parameters needed to generate a Butterworth filter is the sampling frequency, cutoff frequency (both in Hz), and the filter order.

B. Sampling Interval
    - The sampling interval refers to the time duration between consecutive samples taken during the process of digitizing an analog signal. It is determined by the sampling rate, which refers to the number of samples taken per second. The higher the sampling rate, the smaller the sampling interval between consecutive samples. Higher resolutions enhance guitar tuning accuracy but necessitate more power and storage. To optimize accuracy and efficiency, the guitar tuner's resolution and sampling interval must be carefully balanced.

C. Fast-Fourier Transform (FFT)
    - Essentially, an FFT is a method used in DSP to transform signals in the time domain to the frequency domain. Commonly used in the audio processing analysis, MATLAB has a built-in command that performs FFT on the given input signals. Using the function *fft,* its parameter is the input signal itself while it outputs a complex vector whose elements are the various frequency components of the signal.

## *SCHEDULE OF ACTIVITIES (Provide a timetable or Gantt chart of your deliverables. Indicate also whom and when the specific deliverables will be accomplished.)*

| | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 | Week 13 | Week 14 |
|---|---|---|---|---|---|---|---|---|
| | 28-Feb | 7-Mar | 14-Mar | 21-Mar | 28-Mar | 4-Apr | 11-Apr | 18-Apr |
| Initial Project Proposal | | | | | | | | |
| Final Project Proposal | | | | | | | | |
| Coding Proper | | | | | | | | |
| Testing | | | | | | | | |
| Finalization of Code/Project Review | | | | | | | | |
| Making of Poster | | | | | | | | |
| Recording of Demonstration | | | | | | | | |
| Demonstration | | | | | | | | |
| | All members | All members | Mariah Rodriguez | Gia Guevarra | Bettina Dayrit | All members | All members | |

Figure 3.1. Gantt Chart of Deliverables

## REFERENCES


[1] “What is frequency?,” Northwestern.edu, 2023. https://www.qrg.northwestern.edu/projects/vss/docs/communications/1-what-is-frequency.html%20 (accessed Mar. 14, 2023).

[2] R. Thakur, M. K. Pandey, and N. Gupta, “Filtering of Noise in Audio/Voice Signal,” 2018 3rd International Conference on Contemporary Computing and Informatics (IC3I), Oct. 2018, doi: https://doi.org/10.1109/ic3i44769.2018.9007299.

[3] “Sampling rate (audio) - Glossary - Federal Agencies Digitization Guidelines Initiative,” Digitizationguidelines.gov, 2023. https://www.digitizationguidelines.gov/term.php?term=samplingrateaudio#:~:text=Sampling%20rate%20or%20sampling%20frequency,a%20discrete%20or%20digital%20signal. (accessed Mar. 14, 2023).

[4] R. Oshana, “Overview of Digital Signal Processing Algorithms,” DSP Software Development Techniques for Embedded and Real-Time Systems, pp. 59–121, 2006, doi: https://doi.org/10.1016/b978-075067759-2/50006-5.

[5] M. Lourde and Anjali Kuppayil Saji, “A Digital Guitar Tuner,” 2009. Available: https://arxiv.org/ftp/arxiv/papers/0912/0912.0745.pdf

[6] M. M. Hasan and S. Islam, “Development of an Acoustic Guitar Tuner and Graphical User Interface (GUI) using MATLAB,” Applied Research and Smart Technology (ARSTech), vol. 3, no. 2, pp. 49–55, Dec. 2022, doi: https://doi.org/10.23917/arstech.v3i2.1185