Simple Audio Equalizer Using MATLAB GUI

Annisa Maharani(14117046)[1], Felia Azahra(14117072)[2],
Muhammad Rizky Pratama(14117165)[3], Vanesa Adhelia(14117136)[4]
Prodi Teknik Informatika, Jurusan Teknik Elektro, Informatika Dan Sistem Fisika
Institut Teknologi Sumatera
Email: annisa.14117046@student.itera.ac.id[1], felia.14117072@student.itera.ac.id[2], muhammad.14117165@student.itera.ac.id[3], vanesa.14117136@student.itera.ac.id[4]

***Abstract*— Audio equalizers can be used to manipulate signal frequencies to improve sound quality for different listening environments. During analog to digital conversion, audio signals are subject to phase shift, quantization noise, and electromagnetic noise that can be reduced with the help of audio equalizers. Applying filters across multiple bandwidths can alter inaccurate signals in order to provide a higher quality output. This paper presents the analysis and implementation of filtering algorithms in digital signal processing to affect overall sound quality. These algorithms were tested in MATLAB and successful implementation of these algorithms can be observed by finding the impulse response and frequency response of the system. The results show an audio equalization algorithm that was verified by analyzing test signals to compare frequency response before and after equalization**

Keywords- ***Audio Equalizer, Filtering, MATLAB***

I. INTRODUCTION

1.1. Background

One of the most used methods in digital signal processing is audio equalization. Equalizers are utilized in telecommunication systems, recording studios, acoustics, and many devices such as microphones and speakers. Audio equalization provides a way to obtain a cleaner sound since each listening environment changes how the sound is perceived. Phase shift, quantization noise, and electromagnetic noise are also common issues that arise during analog to digital conversion. The purpose of developing an equalizer is to limit these factors that have a negative impact on the quality of an audio signal, which can be accomplished by boosting or cutting parts of the signal.

In digital signal processing (DSP), applications such as MATLAB provide a range of tools that are applicable to audio equalization. These tools offer a useful, simpler way for users to create functions, scripts and models that can be implemented in a range of topics relating to different areas of study. MATLAB, like many other coding applications, contain multiple libraries of convenient toolboxes with algorithms and functions for filtering.

1.2. Objective

The Main objective in this project are:

1. To implement a graphic equalizer in MATLAB
2. To create a band graphic equalizer to control over the amplitude of a sound signal at different resonant frequencies ranging from low frequencies to high frequencies.

II. THEORY

A. Analog versus digital signral

Physical audio signals that we can perceive can be modeled as analog signals, which are continuous both in time and amplitude. However, a computer cannot store an in_nite number of values which disallows the storing of a signal continuous in time. In order for a discrete signal to be classified as digital, the set of values that its magnitude is allowed to adopt must be finite. Thus, an analog signal can be transformed into a digital one by first discretizing it and then approximating the discretized values. Essentially, the discretization and approximation are both inevitable when storing a signal into a computer, and a prerequisite to digital signal processing[1].

B. Sampling and Frame

The process by which an analog signal is converted into a digital is called sampling. Sampling a continuous signal means to periodically measure it and to round and store the measured values. When processing an audio signal, an adequate sampling frequency must be chosen in order to avoid aliasing. The Nyquist-Shannon sampling theorem states that the minimal sampling rate required for perfect reconstruction of the original signal is twice the maximum component frequency of the sampled signal [1].

In order to facilitate the handling of the sampled signal, the computer groups the samples into arrays with predetermined lengths called data frames, depicted in figure 1. The processing of one frame is finished before the next is acquired. This frame-based data format is more efficient regarding speed and is therefore commonly used in real time systems. When choosing a suitable length of the frame, two properties must be balanced: Playback delay and computational speed. A long frame implies more delay but requires less computational power, and the reverse goes for a short frame [2].

C. FIR Filters

The term FIR (finite impulse response) refers to digital filters whose transfer functions depend on a linear combination of previous input signals [2]. The transfer function H(z) for a general FIR filter is given by:

$$H(z) = \sum_{n=0}^{N-1} b_n z^{-n} \qquad (1)$$

where bn is a coefficient for each n and N < 1. The output equation of a FIR fillter in the time domain is a corollary of equation (1), written as:

$$y[k] = \sum_{n=0}^{N-1} b_n x[k-n]. \qquad (2)$$

III. METHODS

This section outlines the sheer mathematical design of the equalizer on the one hand, and its implementation in MATLAB on the other. The mathematical design is further differentiated into filter design and equalizer design.

A. Designing the equalizer

The figure below shows the main design of the MATLAB-based graphic equalizer, which to be separated to 3 different stages each stage responsible of a part of the audio graphical equalizer system

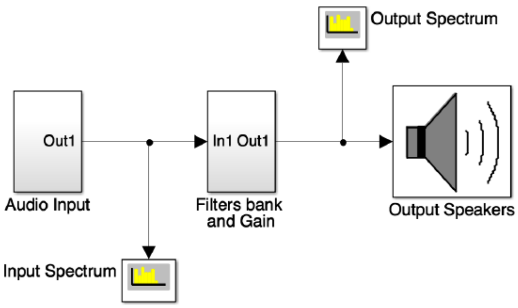

Figure 1. Design of Equalizer

1. First stage: Audio input, which responsible of dragging the audio file and give access to precede the audio file to the next stage.
2. Second stage: Filter bank and Gain, which considered of being the heart of the system where all the filters and gain function beside filtering the audio file and prepare it to the next stage.
3. Third stage: Output speakers, the last stage, Responsible of the output from the second stage after being filtered and to present the equalized audio and results.

The GUI include the following features:

1. Browse button: allow the user to choose different audio file easily from the computer
2. Ten sliders: allow the user to control the gain easily and adjust the bands of the filters
3. Six audio effects: the effects automatically adjust the Band to a specific value for the desired effect.

IV. RESULT AND DISCUSSION

4.1 Functionality

The application, which is builded using MATLAB, has a simple and straight forward interface. In addition to the slider plate i.e the actual equalizer, the application is equipped with some other features making it easier and attractive. The wav file address can be browsed through the “Browse” button. The volume of the audio can be adjusted using the volume slider at the left side. The audio can be paused, resumed and stopped at any time. While at play, the graphics of the original and filtered signal will be displayed. Furthermore, 6 different presets are implemented which are named Pop, Rock, Reggae, Techno, Classical and Party by the genre they are costumized for.

Figure 2. Interface of the Application (1)

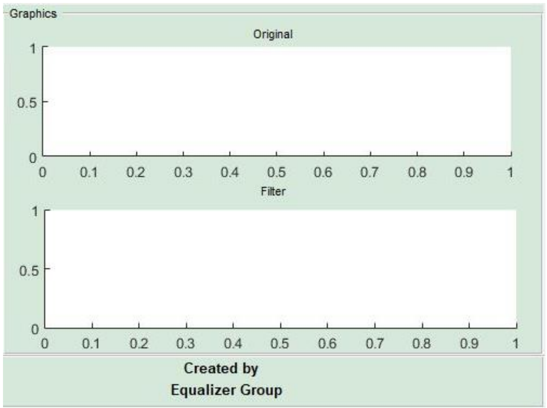

Figure 3. Interface of the Application (2)

4.2 Input Signal

Before filtering the signal, insert the audio file with wav extension as the signal input.

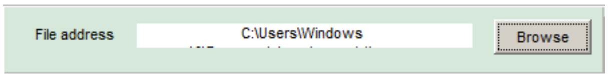

Figure 4. Audio File Address

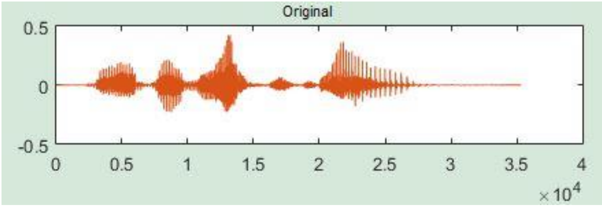

Figure 5. The Graphic of Original Signal

4.3 *Pop Preset*

The adjustment of the equalizer for Pop preset can be seen as shown below in Figure 6.

Figure 6. Equalizer GUI for Pop Preset

Input signal is received, manipulated, and sent to the right destination according to the Pop preset.

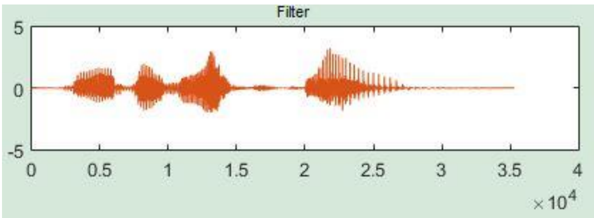

Figure 7. The Graphic of Filtered Signal (Pop)

4.4 *Rock Preset*

The adjustment of the equalizer for Rock preset can be seen as shown below in Figure 8.

Figure 8. Equalizer GUI for Rock Preset

Input signal is received, manipulated, and sent to the right destination accroding to the Rock preset.

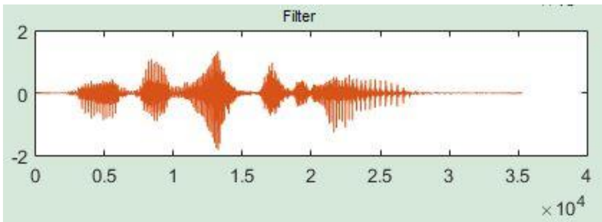

Figure 9. The Graphic of Filtered Signal (Rock)

4.5 *Reggae Preset*

The adjustment of the equalizer for Reggae preset can be seen as shown below in Figure 10.

Figure 10. Equalizer GUI for Reggae Preset

Input signal is received, manipulated, and sent to the right destination accroding to the Reggae preset.

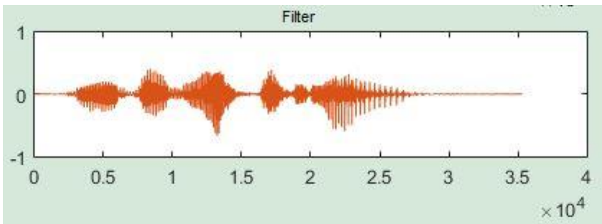

Figure 11. The Graphic of Filtered Signal (Reggae)

4.6 *Techno Preset*

The adjustment of the equalizer for Techno preset can be seen as shown below in Figure 12.

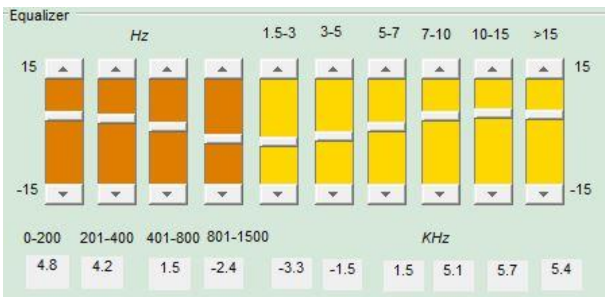

Figure 12. Equalizer GUI for Techno Preset

Input signal is received, manipulated, and sent to the right destination according to the Techno preset.

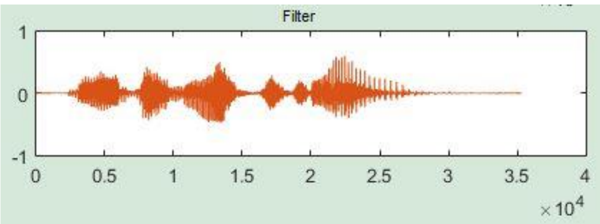

Figure 13. The Graphic of Filtered Signal (Techno)

4.7 Classical Preset

The adjustment of the equalizer for Classical preset can be seen as shown below in Figure 14.

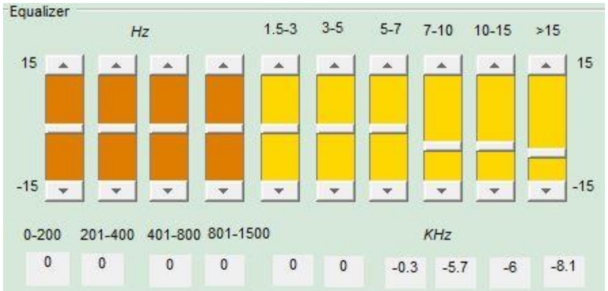

Figure 14. Equalizer GUI for Classical Preset

Input signal is received, manipulated, and sent to the right destination according to the Classical preset.

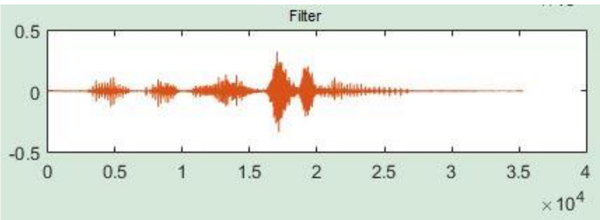

Figure 15. The Graphic of Filtered Signal (Classical)

4.8 Party Preset

The adjustment of the equalizer for Party preset can be seen as shown below in Figure 16.

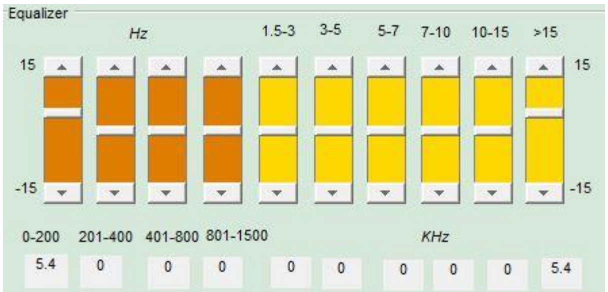

Figure 16. Equalizer GUI for Party Preset

Input signal is received, manipulated, and sent to the right destination according to the Party preset.

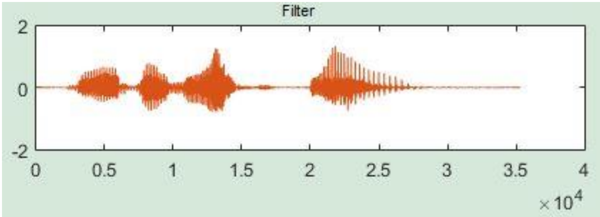

Figure 17. The Graphic of Filtered Signal (Party)

V. CONCLUSIONS

In conclusion according to the assignment papers, digital signal processing covers many areas of our daily routines of many complex systems, however of the technology development happening now days and the wide use of the audio graphic equalizers in the music industry, the systems can be improved by adding more control to the equalizers for example to increase the number of frequency bands used, as the number of bands increases users can have more control and accuracy of the filtering of audio equalizers, while the frequency ranges to be more wider and under control. However after facing many challenges the aim was fully archived with a minimum possibility of errors.

REFERENCES

[1] Steffi Knorn, Signals and Systems, Signals and Systems at department of engineering sciences at Uppsala University, 2018.

[2] Sample and Frame-Based Concepts, https://www.mathworks.com/help/dsp/ug/sample-and-frame-based-concepts.html Accessed: 2020-05-12

[3] Youtube. Digital audio equalizer using matlab, Shubham Jogi, https://youtu.be/H6UiyfmYa54 Accessed : 2020-05-12