NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

ASSIGNMENT 1 - AUDIO EQUALIZER REPORT

Hu Xiaoxiang
U1521319A
EEE

21 Oct, 2017

Contents

1 Introduction **1**
1.1 Background 1
1.2 Motivation 1
1.3 Objective 1
1.4 Report Outline 1

2 Design And System Construction **1**

3 Algorithm **2**

4 Results And Analysis **2**
4.1 Results 2
4.2 Analysis 9

5 Conclusion And Recommendations **9**

References **9**

1 Introduction

1.1 Background

Equalization technique has been applied in signal processing area for a long period. By definition, equalization is the process of adjusting the balance between frequency components within an electronic signal. An audio equalizer is typically used for multiple frequency controls to adjust sound tone. To be specific, an equalizer can compensate for deficiencies in a sound pickup or reduce extraneous sounds, such as noise. Besides, instrument clarity can be improved through equalizer, too. For example, boosting a sounds harmonics gives the impression of more presence and brightness and decreasing a sounds harmonics gives the impression of a dull, less dazzling sound.

1.2 Motivation

The development of equalizer was mainly motivated by the requirement of high quality audio output. The audio output quality of loudspeakers may not be uniform due to size, mechanical and cost constraints, even though they are usually designed to have a fairly uniform response across the frequency spectrum. Therefore, the involvement of audio equalizer is necessary to flatten this response, or to shape it according to listener preferences.

In the early years, the audio equalizer was first implemented by analog filters. However, the proliferation of digital audio sources in recent decades, such as the Internet and the USB, gives an opportunity to apply digital signal processing to audio signals. Especially, there are many advantages, which include simplification of design and verification, greater flexibility and reliability, and significantly superior sound quality, allowing the creation of digital equalizer that can outperform its analog counterparts.

1.3 Objective

The objective of this project is to design and simulate an digital audio equalizer using Matlab. It includes both implementation and evaluation of the audio equalizer.

1.4 Report Outline

This report mainly includes the method used to implement the audio equalizer and performance analysis based on real audio signal input. The implementation procedure briefly explains how to use 'App Designer' in Matlab to create a Graphical User Interface (GUI) as well as the algorithm of the FIR filter. The focus of performance analysis is on displaying the equalizer's simulation result and discussing the differences among various filter setting and outcomes.

2 Design And System Construction

The entire audio equalizer consists of 3 major blocks: audio input, digital audio equalizer and audio output. The audio input is implemented through Matlab's internal audioread() function. The audio output, which includes the DAC and reconstruction filter, is built on Matlab's sound() function.

The GUI is designed by using Matlab's App Designer. As shown in Figure 1, each frequency band in the equalizer is controlled by a slider. The 'Apply' button collects the values of sliders and then pass them to fir2() function to generate filters' coefficients b. The 5 'plot' buttons plot the figure of different responses of the FIR filter, which including ideal frequency response, actual frequency response and impulse response. The 'load' button can let user select any audio files from their directories as filter input and meanwhile pre-processes the input file data. The 'play' button is the implementation of a sound() function to reconstruct the filtered data and output analog signal. The 'stop' button is used to stop the music player.

Figure 1: Equalizer GUI Layout

3 Algorithm

Function b=fir2(n,f,m,npt,lap,wind) is used to implement the multi-band filter of this audio equalizer. n, f, m, npt, lab and wind is the representation of filter order, frequency interval, magnitude corresponding to each frequency interval, number of grid points, length of region around duplicate frequency and window function respectively. Since the value of filter tap (512) is specified, the filter order can be determined to be 511, given the formula: $filterorder = filtertap - 1$. f the frequency range is fixed and m is calculated by the inputs from sliders. The default value of npt and lab is used and window function chebwin() is chosen as the input vaule of wind.

fir2 uses frequency sampling method to design FIR filter. The desired frequency response function is interpolated linearly onto a dense, evenly spaced grid of length npt, which is default set to 512. fir2 also sets up regions of lap points around repeated values of f to provide steep but smooth transitions. To obtain the filter coefficients, the function first applies an inverse fast Fourier transform to the grid. After that, the result multiplies by the specified window function and output the coefficients of the numerators.

For example, to obtain the coefficients of a low pass filter like Figure 2, the algorithm samples the desired frequency response first and then maps the sampling points to a grid. The total number of sampling points relies on the value of npt. Another attribute lap, as shown in Figure 3, decides the width between two adjacent band. In other word, less the value of lab, steeper the transitions from edge to edge. The sampling points are stored in a vector and sent to Inverse Fast Fourier Transform (IFFT) to calculate the filter coefficients b.

4 Results And Analysis

4.1 Results

Simulation setting:

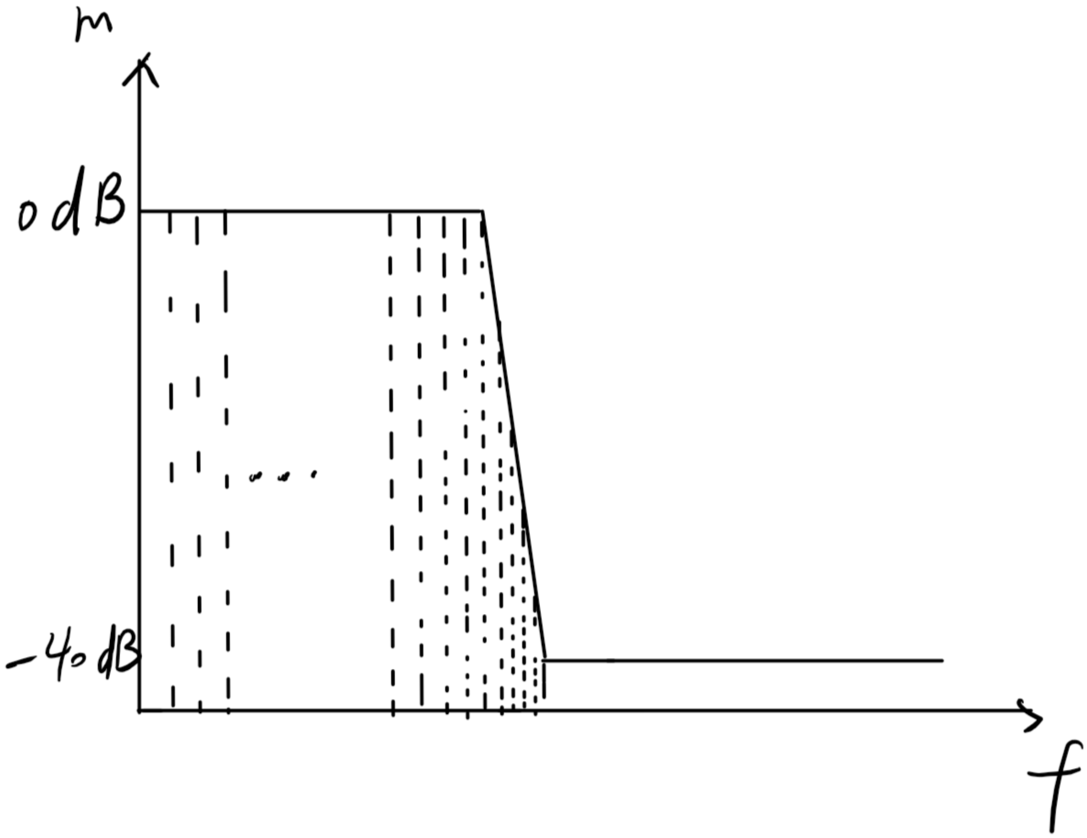

Figure 2: Frequency Sampling

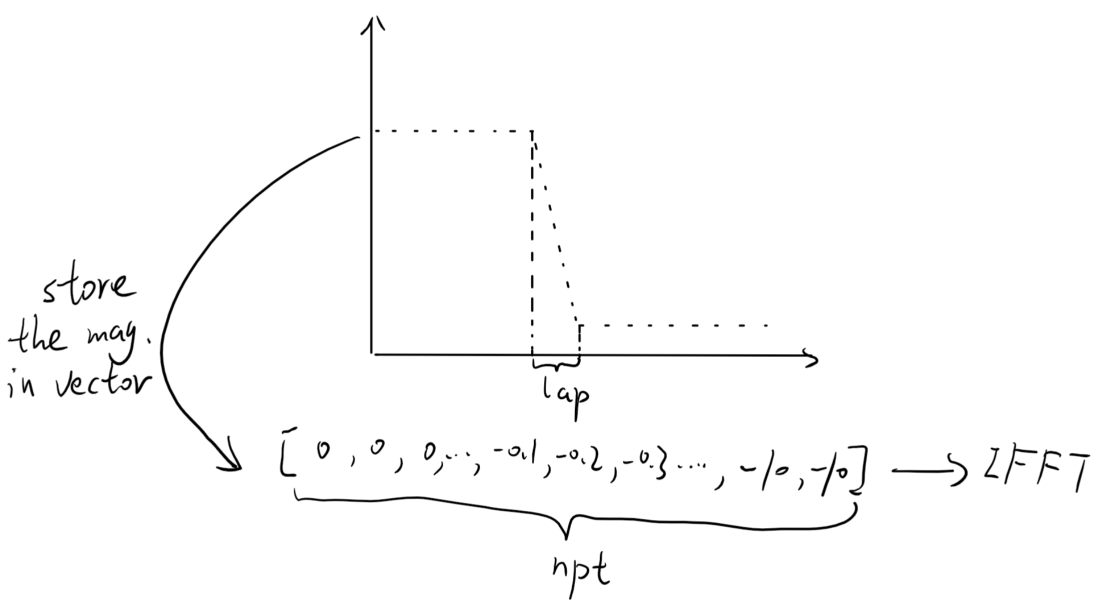

Figure 3: Inverse Fast Fourier Transform

a) Signal components below 500 Hz and above 4000 Hz being enhanced.

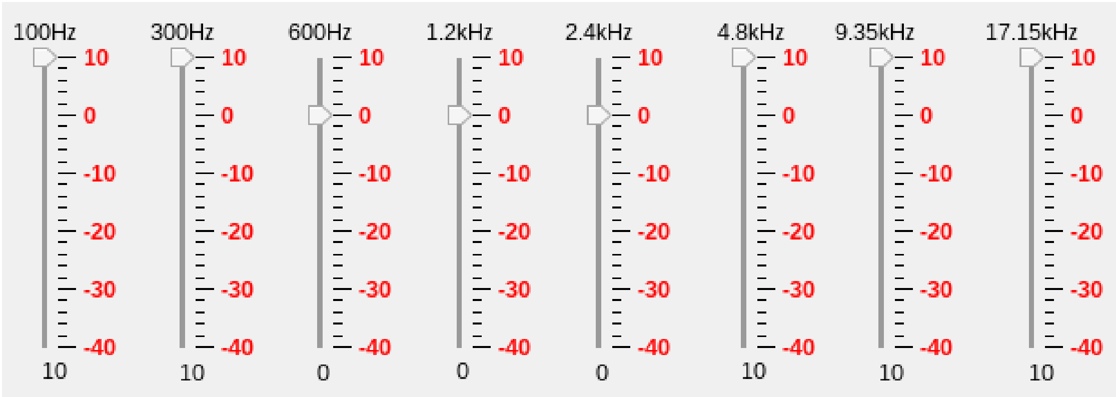

Figure 4: Setting A

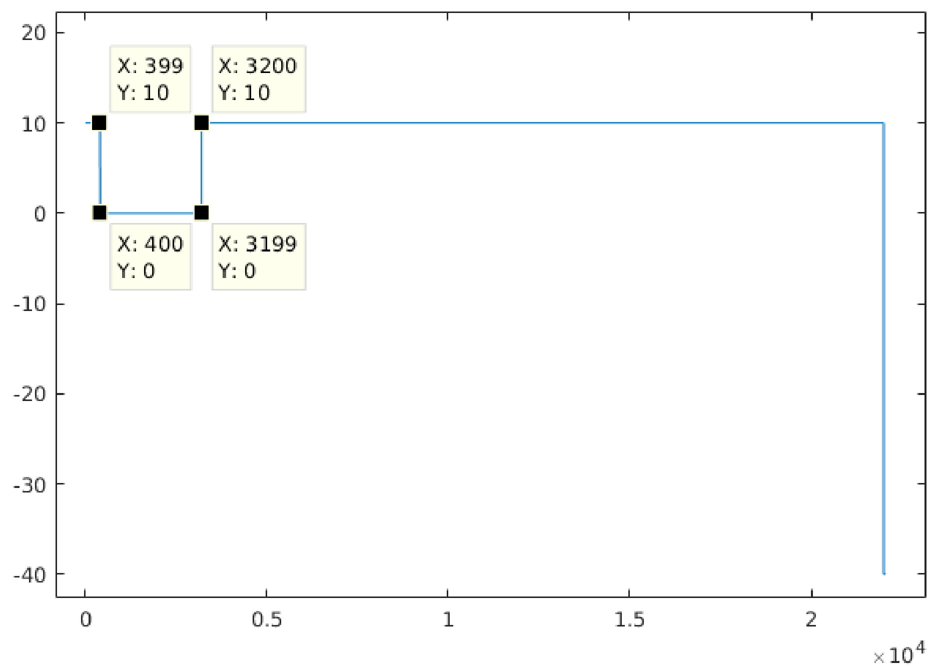

Figure 5: Ideal Frequency Response

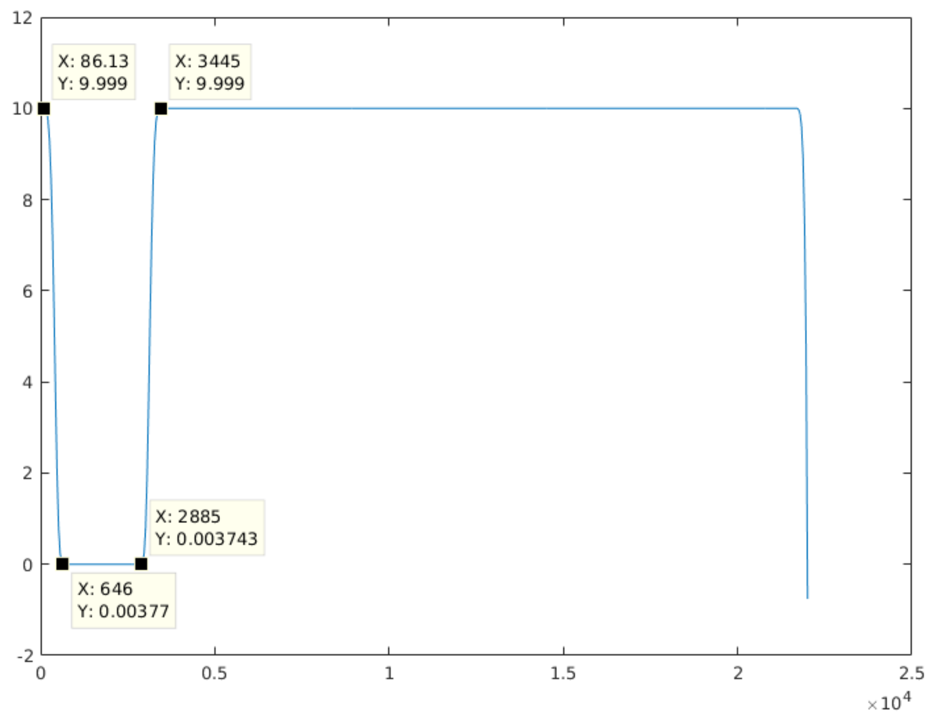

Figure 6: Actual Frequency Response

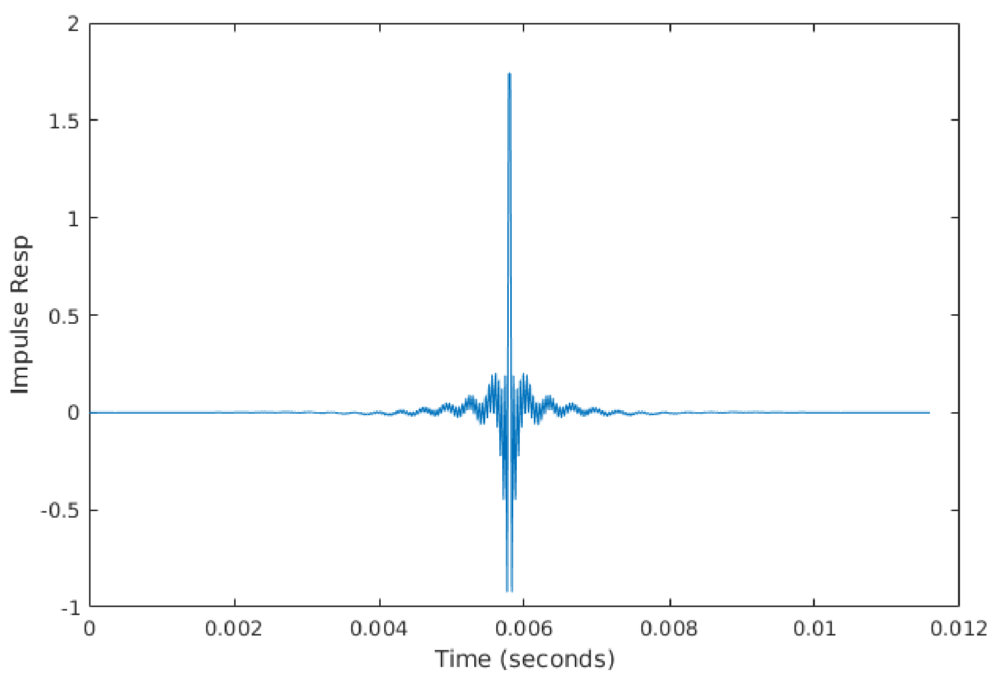

Figure 7: Impulse Response

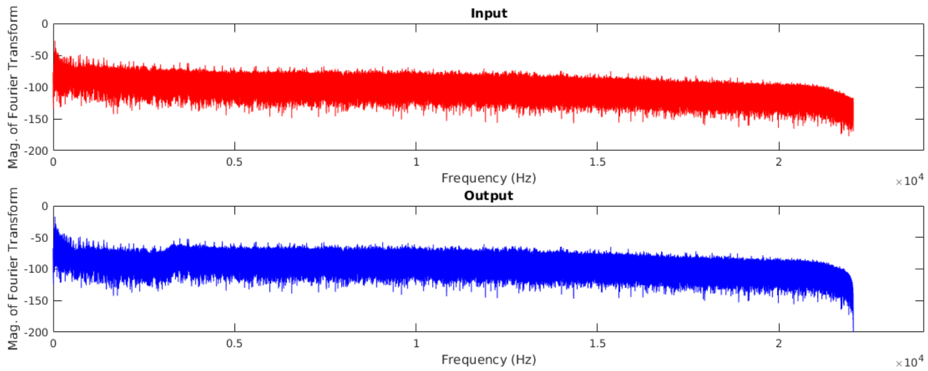

Figure 8: Frequency Spectra

b) Signal components between 500 Hz and 4000 Hz being enhanced.

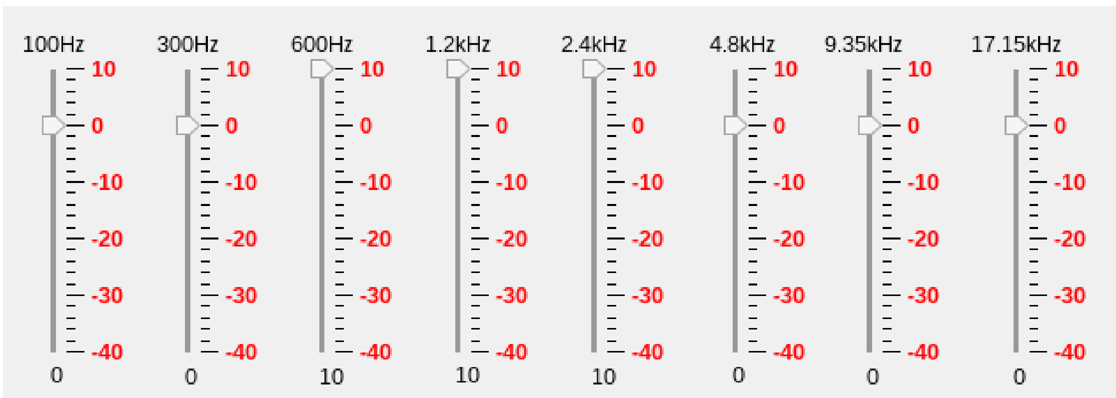

Figure 9: Setting B

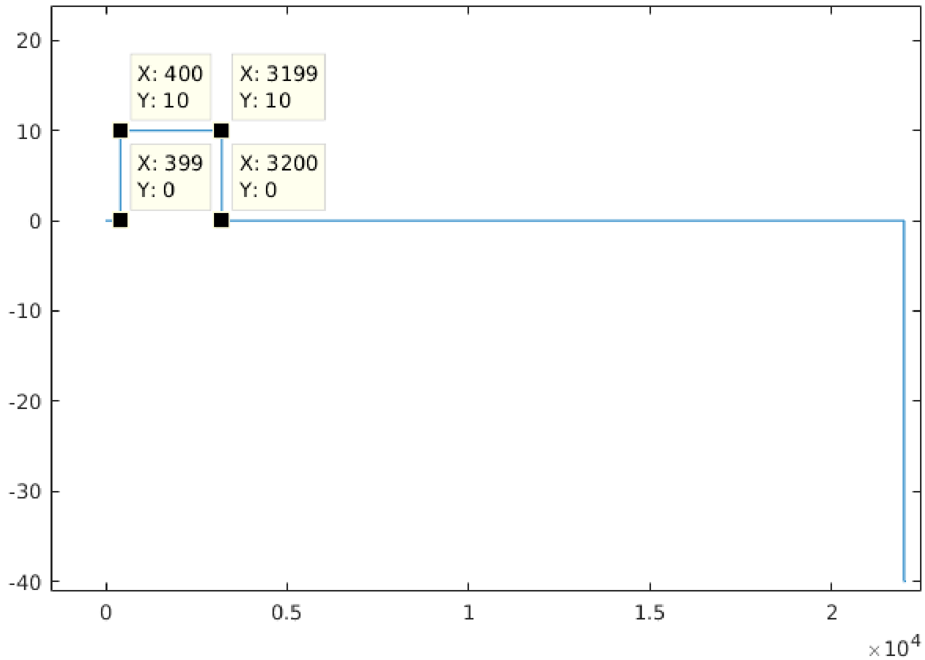

Figure 10: Ideal Frequency Response

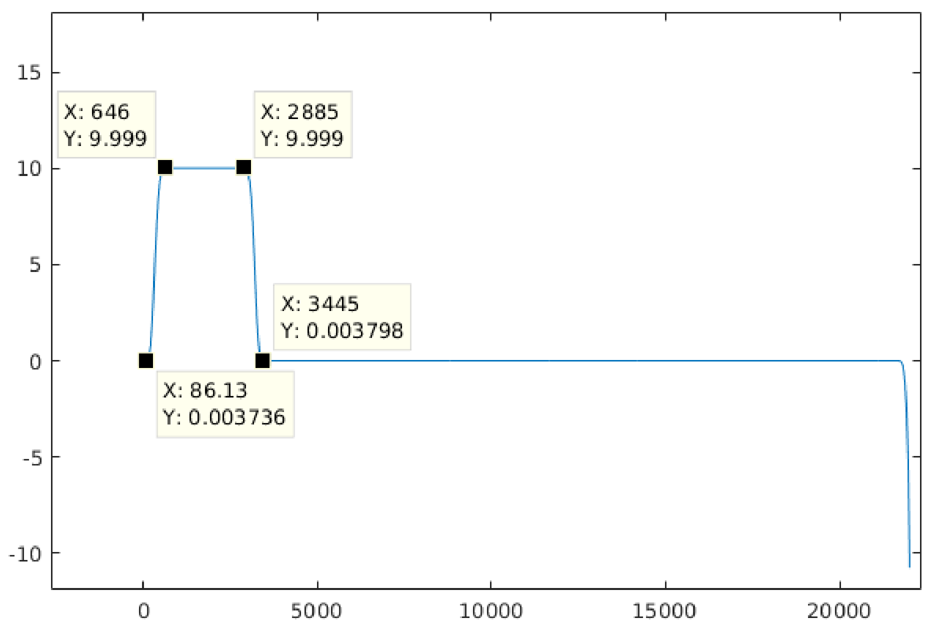

Figure 11: Actual Frequency Response

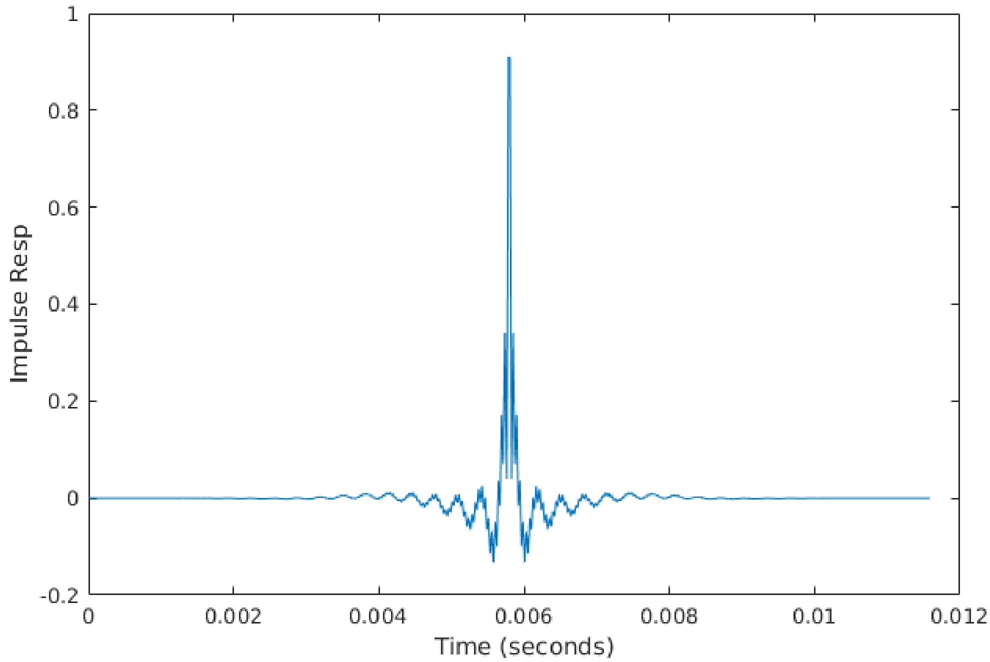

Figure 12: Impulse Response

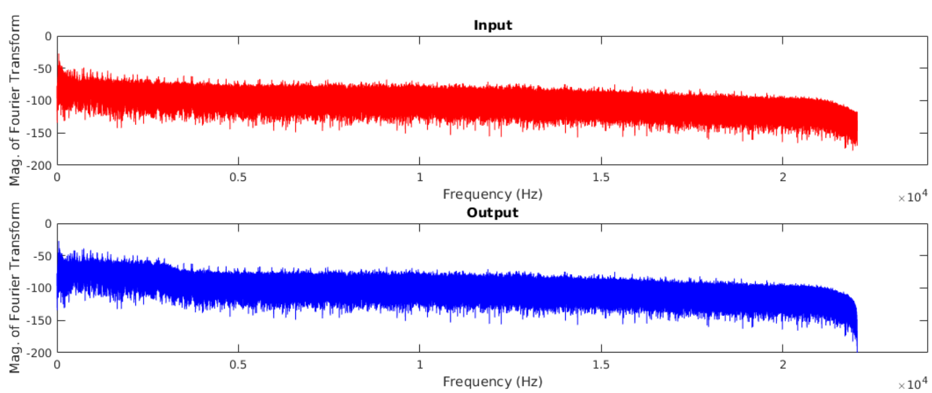

Figure 13: Frequency Spectra

4.2 Analysis

Generally, the shape of the output frequency spectrum corresponds to the shape of the FIR filter's frequency response. Figure 4 and 9 show two different configurations of the equalizer. By setting the gain of band 1,2,6,7,8 to +10dB, signal components below 400 Hz and above 3200 are enhanced. On the second configuration, where band 3,4,5 are set to +10dB, signal components between 400 Hz and 3200 Hz are enhanced as a result.

Compared to the plot of ideal filter frequency response (Figure 5 and 10), it is not difficult to notice that the transitions between different band edges of actual frequency response is more smooth. The cursor on Figure 6 and 11 shows that the transition bandwidth of setting (a) is about 560 Hz, and this is the main reason why the performance on low frequency band downgrades significantly. Because the bandwidth of low frequency band is very narrow, which is 200 Hz typically, a wide transition bandwidth like 560 Hz can cause unwanted result while filtering.

To solve this problem, one way is to increase the filter order. For example, by increasing the filter order from 511 to 800, the frequency response plot on Figure 14 shows that the transition bandwidth reduces about 200 Hz. However, even with the increment of filter order by almost 300, the result is still not desirable. Actually, to design an optimized FIR filter by using frequency sampling method, a fairly big filter order is required so as to compensate this defect.

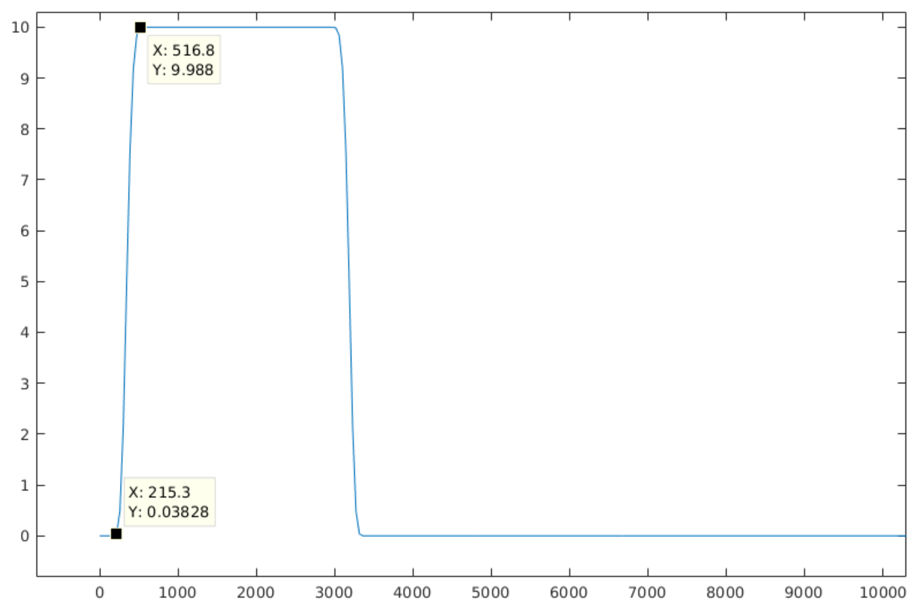

Figure 14: Frequency Response of FIR Filter with 800 Filter Order

5 Conclusion And Recommendations

In summary, the FIR filter is able to fulfill the requirement of an audio equalizer. The result shows that the equalizer performs better on higher frequency band compared to lower frequency band. Generally, the frequency sampling method is a simple way to implement FIR filter. However, this method requires longer filter length to achieve a more precise solution. In comparison to other design method, such as window design method, least squared method and the Parks-McClellan method, the frequency sampling method can have overall more error. Thus, it is recommended to implement the digital equalizer by using other methods for a better solution on low frequency band if a higher filter order is not allowed.

References

[1] Rusty Allred and Ryan Hsiao, "The Advantages of Digital Equalization," in International IC-China, Conf., 2000

[2] Mathworks (2017) fir2
[online] Available at: https://www.mathworks.com/help/signal/ref/fir2.html

[3] StackExchange (2017) Difference between frequency sampling and windowing method
[online] Available at: https://dsp.stackexchange.com/questions/31905/difference-between-frequency-sampling-and-windowing-method