**Project Report**

On

**DIGITAL EQUALIZER USING FIR FILTER**

**By**

Rishikesh Devarmani (172065)

Shubhamkumar Jogi (172068)

Faisal Belwadi (172091)

Under the Guidance of

Dr. M. S. Deshpande

**Course: Signals and Systems**

**2017-18 (Semester-I)**

**S. Y. B. Tech**



Bansilal Ramnath Agarwal Charitable Trust’s

Vishwakarma Institute of Technology, Pune- 411037.

Department of Electronics and Telecommunication Engineering

**Abstract:**

The design and implementation of a 5-band linear phase digital audio equalizer system is presented. Beginning from analysis/synthesis filterbank, an innovative uniform and non uniform bands audio equalizer is derived using multirate properties. In literature fixed frequency response equalization has well-known problems due to algorithms implementation. The idea of this work derives from different techniques employed in filter banks to avoid aliasing in the case of adaptive filtering in each band. The effectiveness of the algorithm is shown comparing it with a simple filterbank and with an octave band equalizer based on frequency domain technique. The solution presented here has several advantages in terms of linear phase and uniform frequency response avoiding ripple between adjacent bands.

**Project mapping to COs and POs:**

|  |  |  |
| --- | --- | --- |
| **CO** | **CO Statements** | **Correlation** |
| **CO1** | Classify signals and systems as discrete/continuous, linear/non-linear, causal/non-causal, time-variant/invariant, etc. |  |
| **CO2** | Determine the response of the LTI system to any input signal | **✔** |
| **CO3** | Analyze continuous-time periodic signals using Fourier series |  |
| **CO4** | Apply Fourier transform to obtain spectrum of a continuous time energy signals | **✔** |
| **CO5** | Analyze discrete-time signals and systems using discrete Fourier transform | **✔** |
| **CO6** | Solve second order differential equations using Laplace transform and to test the stability of the given system |  |

|  |  |  |
| --- | --- | --- |
| **PO** | **PO Statements** | **Correlation** |
| **PO1** | Engineering knowledge | **3** |
| **PO2** | Problem analysis: | **2** |
| **PO3** | Design/development of solutions | **2** |
| **PO4** | Conduct investigations of complex problems | **1** |
| **PO5** | Modern tool usage | **3** |
| **PO6** | The engineer and society | **2** |
| **PO7** | Environment and sustainability | **1** |
| **PO8** | Ethics | **2** |
| **PO9** | Individual and team work | **3** |
| **PO10** | Communication | **3** |
| **PO11** | Project management and finance | **2** |
| **PO12** | Life-long learning | **2** |

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

1. **Introduction:**

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 and using algorithm modification 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 and reduce the effects of miscellaneous noise. 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 analysing test signals to compare frequency response before and after equalization.

In digital signal processing (DSP), MATLAB provides 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. MATLAB’s Butterworth function was used in order to design filters for this equalizer.

Fast Fourier Transform (FFT) is useful to study the signal in frequency domain which is being used in the project.

1. **Objectives:**

The main objective of this project is to produce successful 5 band equalizer for a sound system. Equalizer is a kind or a audio equalizer used necessary to quality output from a sound system. Audio equalizer separates a range of selective audio frequencies and this enables listener t or operator to select which frequencies to be amplified or attenuated and therefore produce quality sound without significance loss in treble, bass, mid-range audio frequency.

This 5 band audio equalizer, also known as first octave equalizer, will enable a range of audio frequencies to be equalized at a low cost.

There are 5-channels controls for the different frequencies of the first octave equalizer. Each channel control has a definite relationship with each other this could be done by observing the numerical values of the frequencies for which the control provided: 250Hz, 500Hz, 2000Hz, and 4000Hz.

The project also aims to analyze the frequency components of the filtered (or equalized) signal.

1. **Methodology**

**Input**

**Read audio signal & determine parameters**

**Implement equalization filters**

**Run MATLAB GUI to equalize**

**Reconstruct signal components**

**Plot Data**

**Output**

**Newly Modified audio signal**

*Figure 1: Flowchart of audio equalization algorithm*

The steps that are being followed in the designing of the digital audio equalizer is as shown in figure 1.

In the design process, the FIR filters were implemented in a way to create the proposed multi-band filtering algorithm. These discrete Finite Impulse Response filters (FIR) obtain the coefficients from the Butterworth transfer function to equalize a test signal.

Implementing Filters

h2=fir1(100,2\*250/Fs,'lowpass');

h3=fir1(100,[2\*250/Fs 2\*500/Fs],'bandpass');

h4=fir1(100,[2\*500/Fs 2\*2000/Fs],'bandpass');

h5=fir1(100,[2\*2000/Fs 2\*4000/Fs],'bandpass');

h6=fir1(100,[2\*4000/Fs 2\*20000/Fs],‘highpass');

The filtered signal is then multiplied with the value of the slider assigned by the user which represents the gain of the filter. All the five signals are then added together as per the following equation,

y=y2\*C(1)+y3\*C(2)+y4\*C(3)+y5\*C(4)+y6\*C(5)

where,

y = filtered output signal

ya [a=2,3,4,5,6] = individual filter signal

c(b) [b=1,2,3,4,5] = value of the slider i.e. the gain of the filter

The newly modified (or equalized) signal can be played and plotted using Fast Fourier Transform (FFT) in order to analyse different frequency components of the signal.

1. **System Implementation:**

The design of a multi-band equalizer begins by obtaining the input signal and gain values for each filter. It is important to note that MATLAB uses the normalized magnitude instead of the decibel value; therefore, this input parameter must be converted.

To build the filters for signal equalization, the Butterworth function in MATLAB was used alongside digital filters and amplifiers.

The Butterworth filter design allows users to easily customize the number of bands and the filter parameters required for specific design purposes.

Once the signal had been equalized in Simulink, the signal was reconstructed to obtain the output sampling data.

This 5-band equalizer design utilizes a ‘low pass’, ‘high pass’, and three ‘band pass’ filter types in order to equalize an input signal. The cut-off frequencies used during implementation are shown in Table 1.

*Table 1: Frequency ranges for 5-band equalizer design*

|  |  |
| --- | --- |
| **Filter Type** | **Cut-off Frequency in Hz** |
| Low pass | 250 |
| Band pass | [250,500] |
| Band pass | [500,2000] |
| Band pass | [2000,4000] |
| High pass | 4000 |

Using MATLAB, a vector containing the sampled data from the input was created at the specified sampling frequency of 44100 Hz. The Fourier Transform of the input signal was performed in order to plot the normalized magnitude response of the signal with its corresponding frequencies.

A system order of five was chosen for each filter since it provided an accurate filter design by incorporating enough previous sampled inputs.

Including higher design orders would have made the computation time much larger and would have not made a significant impact on increasing the accuracy of the filter.

Using the MATLAB function, ‘audioread’, a vector containing the sampled data from the input was created at the specified sampling frequency of 44100 Hz. The Fourier Transform of the input signal was performed in order to plot the normalized magnitude response of the signal with its corresponding frequencies.

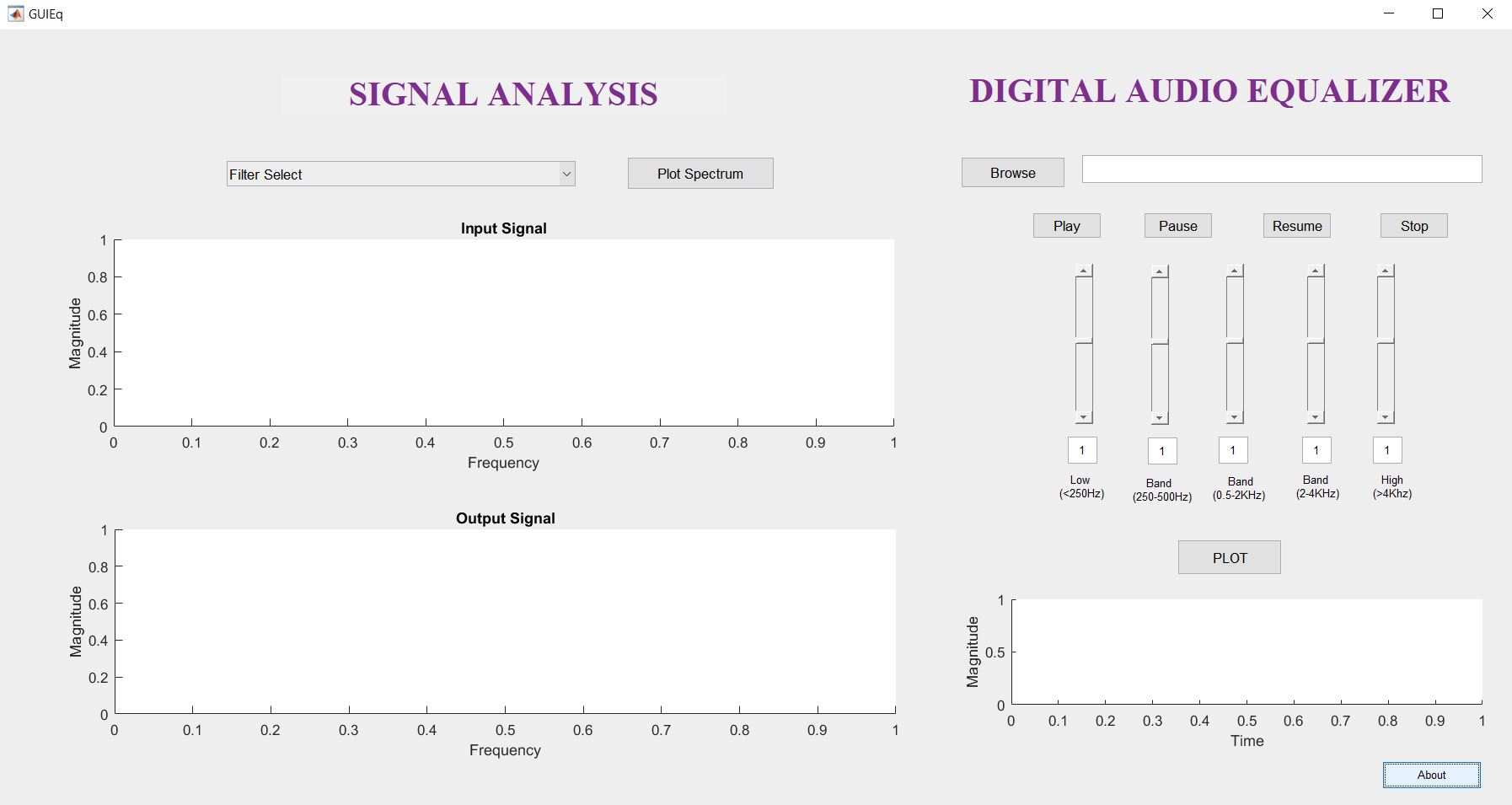
A system order of five was chosen for each filter since it provided an accurate filter design by incorporating enough previous sampled inputs. Higher design orders would have increased the computation time and would have not made a significant impact on increasing the marginal accuracy of the filter.

These discrete finite impulse response filters (FIR) obtain the coefficients from the Butterworth transfer function to equalize a test signal.

Each path was combined in series with a gain amplifier in order to boost or cut the signal by the desired gain amount. The gain values were referenced as a variable in the MATLAB code, and can be changed to any desired value.

**** This sixth path in the model serves as a forward unity path to reconstruct the parts of the signal that were not altered by the filters. Once reconstructed, the output signal was written to the workspace and a vector was created with the filtered data at the sampling frequency. This data could then be written out and played as an audio signal for verification testing.

*Figure 2: Block Diagram of Equalizer System*

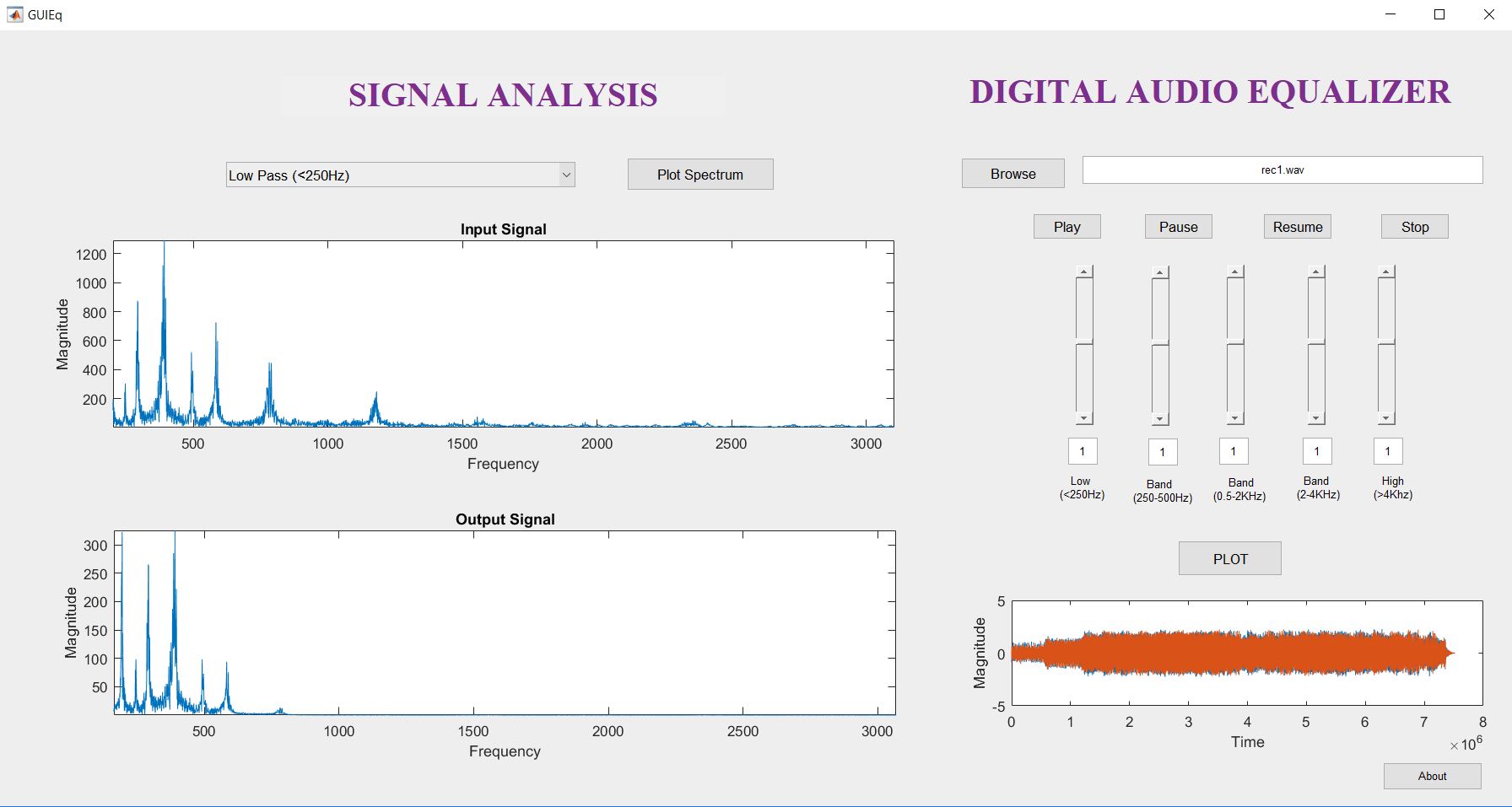
**

*Figure 3: Final GUI Design*

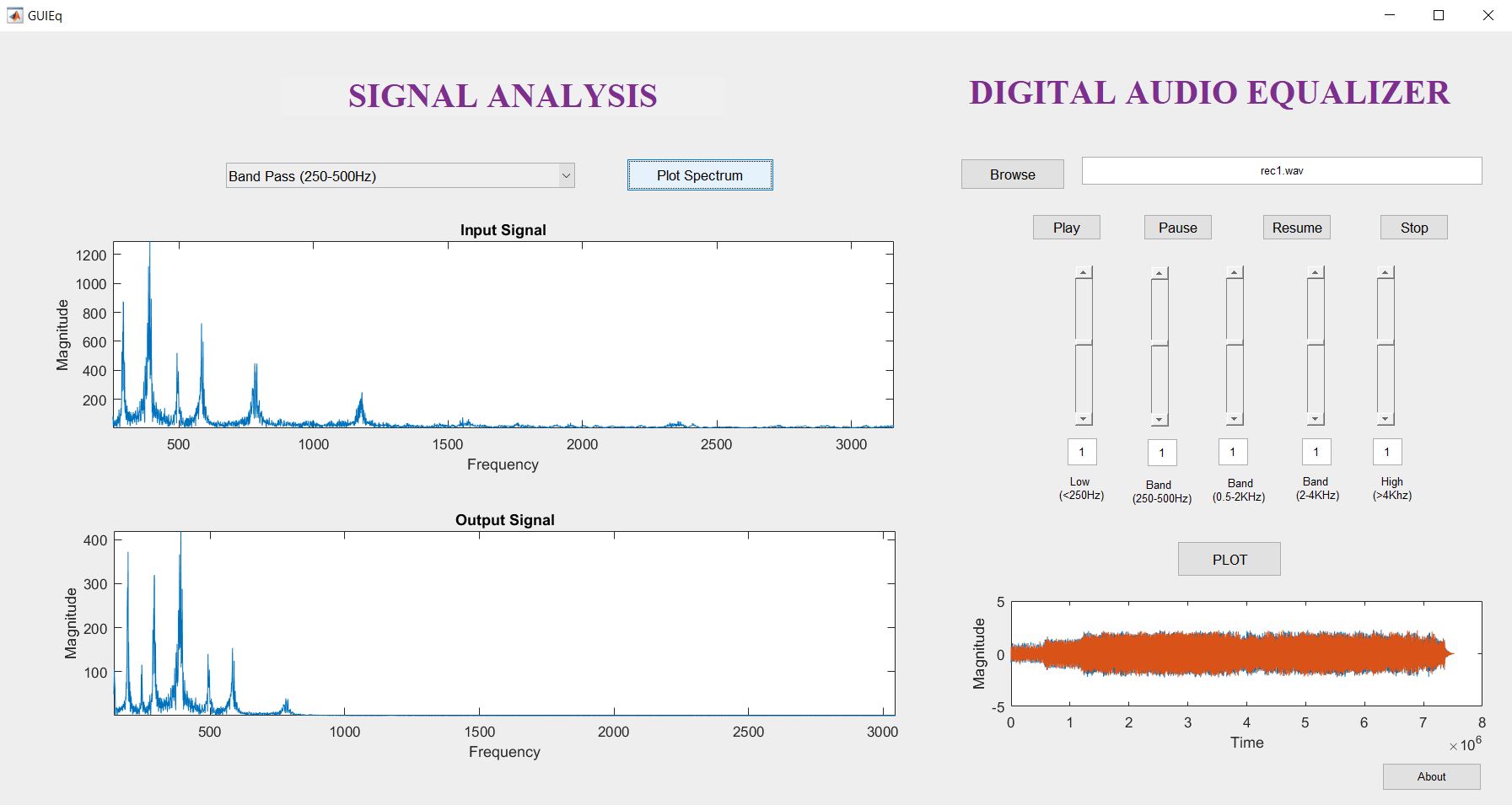
The above figure shows the final GUI view of the project which includes both the parts viz. Digital Equalization and Signal analysis.

1. **Applications:**
2. Equalizers are built to control the loss and gain of frequencies within a sound system. This allows a sound system to sound natural and full. It also gives it the ability to maximize volume while eliminating feedback. Many stereos today are built with a graphic equalizer right in the system. But for a high-end stereo system this unit is generally separate and allows fine tuning. Since there are not usually any microphone inputs in a home stereo application, the adjustments usually do not have to take into account the ambient sound in the room, but do help compensate for the "acoustics" in the room.
3. Equalizers are frequently used in public address systems to sharpen the sound and reduce echoes. Stadiums, sports arenas and other venues will want a good sound system with a good equalizer.
4. Churches, with their unusually angled rooms and ceilings will especially benefit from having an equalizer in the sound system. As churches often have multiple microphones and speakers, a stereo equalizer is a must.
5. Schools will want an equalizer to maximize sound output in various venues from auditoriums to gyms. Basketball and volleyball events are enhanced by quality audio equipment allowing for crisp, clear announcements.
6. Bands, and other live traveling shows will perhaps find the equalizer most useful, as it is nearly impossible to construct a good sound system for every venue without adjusting for frequencies that will create feedback.
7. Most studios have an equalizer, as it is very useful for coordinating the various microphones and sound inputs into the system. It can also reduce and eliminate ambient noises like air-conditioners that may hum in the background.
8. **Results:**

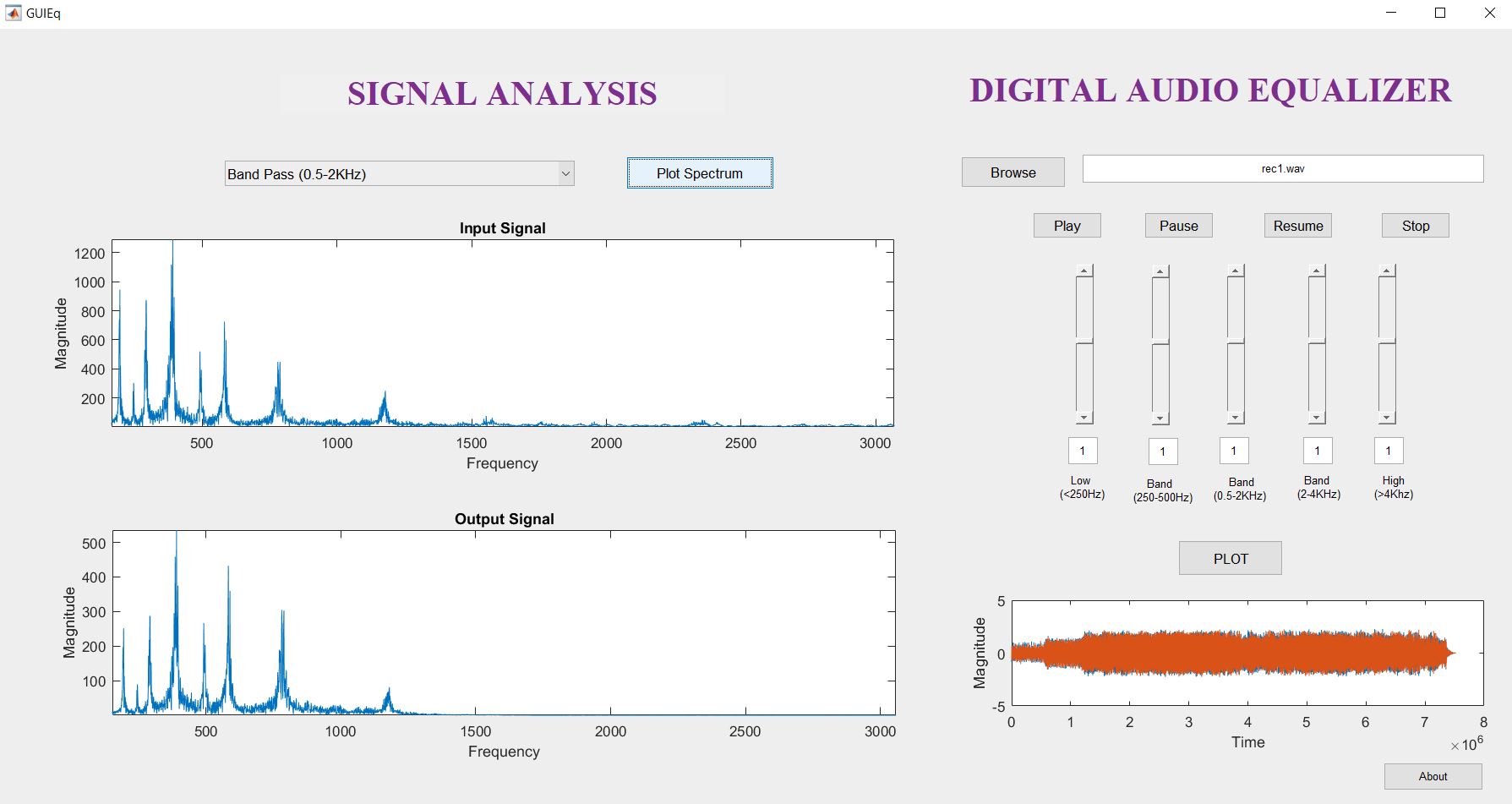
The analysis of the filtered audio signal is as shown in the following figures. The figure 4 displays the low frequency components (less than 250 Hz). The outputs of the three band pass filters are as shown wherein figure 5 shows the fundamental frequency components in the range of 250 Hz to 500 Hz, figure 6 shows the fundamental frequency components between 500 Hz to 2000 Hz and figure 7 displays the fundamental frequency components in the range of 2000 Hz to 4000 Hz. The high frequency components are depicted in figure 8 which are above 4000 Hz i.e. the output of the high pass filter.

****

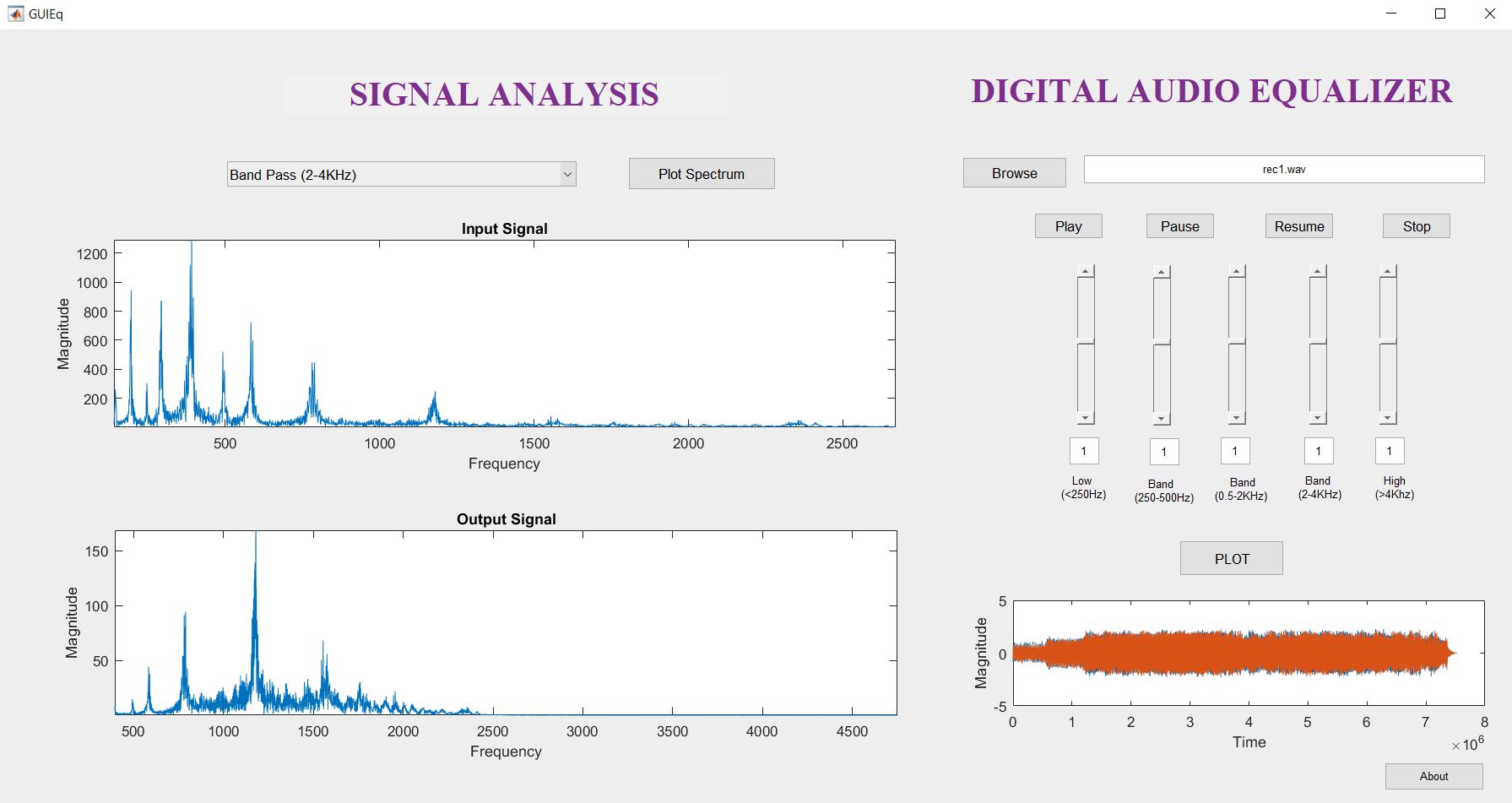
*Figure 4: Low pass filter output (<250 Hz)*

**

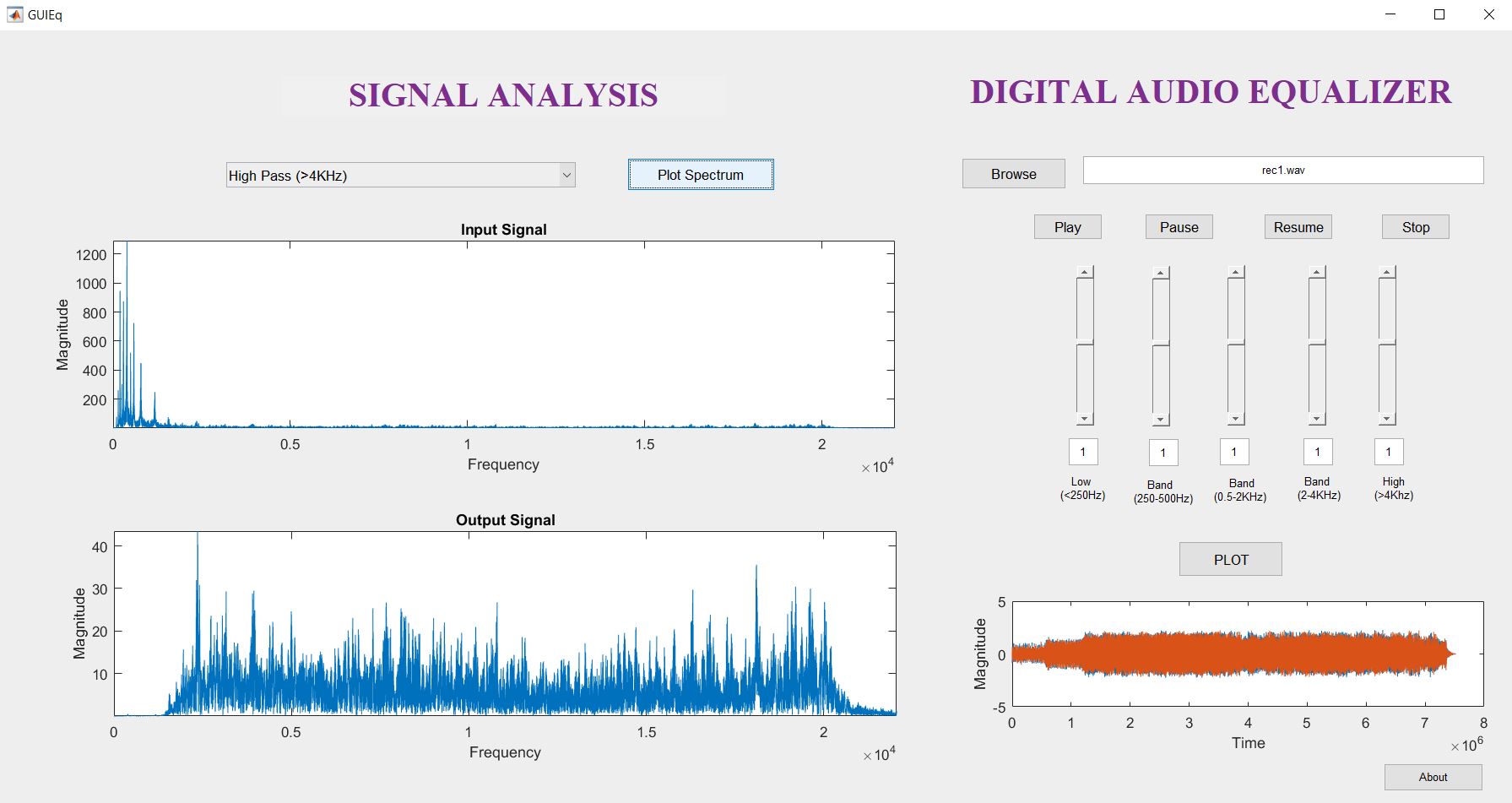
*Figure 5: Band pass filter (250-500Hz)*

**

*Figure 6: Band pass filter (0.5-2 KHz)*



*Figure 7: Band pass filter output (2-4 KHz)*



*Figure 8 :High pass filter output (>4 KHz)*

1. **Conclusion**

An audio equalizer is an effective way to manipulate and improve the quality of an input signal. There are many methods that can be used in order to improve the quality of a signal in MATLAB. For this research, the Butterworth filter was utilized in order to reduce any interference created in the processing of signals across multiple bandwidths. The result of filtering an input signal is an uninterrupted signal that improves upon the sound quality of the input. Signals that are subject to noise, phase shifts, and frequency interference will often manipulate the signal in such a way that it is difficult to process or hear. Graphical methods are useful in providing a visual way to observe the characteristics of a signal.

Equalizers are an essential component of communications and sound systems because of the ability to improve sound quality and drain out miscellaneous noise. MATLAB have proved to be extremely useful tools for digital signal processing. The methods applied in this paper can be implemented to build customized equalizers using the various properties and filter types with the Butterworth syntax.

1. **References:**

[1] Building an Equalizer, EE 262, Lab Section A2, Department of Electrical Engineering, University of Missouri-Rolla.

[2] Implementation of a Multi-band Equalizer in MATLAB and Simulink Using Algorithm Manipulation, Wesley C. Caruso, Benjamin C. Fisk, Jacob D. Whaley, Suhayb Alsaedi, Benjamin D. McPheron School of Engineering, Computing & Construction Management, *Roger Williams University.*

[3] Digital Filters with MATLAB, Ricardo A. Losada, The MathWorks, Inc., May 18, 2008.

[4] MathWorks Documentation