UNIVERSITY OF TEXAS ARLINGTON

MATLAB Filter Design Project

Prepared by
Athiya Manoj
UTA ID: 1001843726
CSE 3313 : Signal Processing
University of Texas at Arlington
December 12th, 2022

Table of Contents

I Abstract

This technical report presents an analysis of the use of a Butterworth filter for audio filtering unwanted noise from an audio clip. The Butterworth filter is a type of low-pass filter that is commonly used to remove unwanted high-frequency noise from an audio signal. This filter is characterized by its flat frequency response, which makes it effective at removing noise without altering the overall frequency content of the signal. Discrete Fourier transform and the Butterworth filter are two important filtering techniques used in signal processing. Using Discrete Fourier Transform (DFT) to filter out undesirable frequencies and a Butterworth filter which is a frequency-selective signal processing device to suppress sounds that occur at higher frequencies, we can filter out all the unwanted sounds from an audio clip. This report describes the utilization of Discrete Fourier transform and the Butterworth Filter with the proper parameters and MATLAB built-in functions and filter design instructions to create a filter as needed in order to filter out undesirable noises from an audio clip of a conversation. The results show that the Butterworth filter is an effective and efficient tool for removing noise from audio signals.

II Project Overview

1 Objective

The field of audio engineering often requires the use of specialized tools and techniques to enhance the quality of recorded sound. One such technique is audio filtering, which involves removing unwanted noise or frequency components from an audio signal. In this technical report, we will focus on the use of a Butterworth filter for audio filtering. The Butterworth lowpass filter is a commonly used circuit to reduce the frequency response of a signal. It has been used in many applications, such as feedback control and signal processing.

The first use of a Butterworth filter was back in 1930 by Edwin E. Hanley, who discovered that it helped to enhance filtering qualities and lessen noise distortion. In the past, engineers would often use one of two different filters: Chebyshev or Bessel. These two filters have been found to have similar properties, but with some key differences. The Butterworth filter has been shown to be better than both Chebyshev and Bessel at achieving the same results, but it does take more processing power and memory space than either of them.

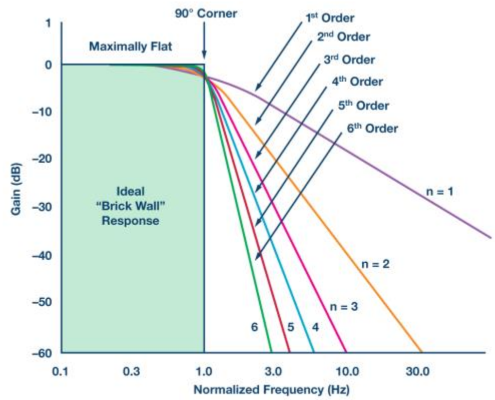

Figure 1. Ideal Butterworth filter transition band with different orders.

In order to accurately filter an audio signal, a Butterworth filter is the most popular choice. In order to achieve the ideal response curve. The higher the Butterworth order, the higher the number of cascaded stages there are within the filter design, and the closer it becomes to the ideal "brick wall" response. However, in practice this "ideal" frequency response is unattainable as it produces excessive passband ripple. The purpose of this project was to develop a Butterworth lowpass filter that eliminates the unwanted noise of thunder and a high-frequency tone from that of a conversation.

III Implementation and Working

1 Problem

We are to create a digital filter using MATLAB filter design commands that will eliminate noise from the audio sample and identify the desired and undesirable frequency ranges for the audio sample using DFT and spectrogram analysis.

2 Approach

To implement a Butterworth filter, we first need to determine the cutoff frequency, which is the frequency at which the filter begins to attenuate the signal. The cutoff frequency is typically chosen based on the desired application and the characteristics of the noise to be removed. Once the cutoff frequency has been determined, we can design the filter using the Butterworth polynomial. This polynomial is used to calculate the filter coefficients, which determine the filter's frequency response. The filter is then implemented using these coefficients in a digital filter design algorithm, such as the bilinear transform method. In terms of its working, the Butterworth filter operates by attenuating the amplitude of frequency components above the cutoff frequency. This attenuation is gradual, with the filter providing a constant gain for frequencies below the cutoff frequency and a decreasing gain for frequencies above the cutoff frequency. The result is a smooth transition from the passband to the stopband, with no sharp cutoff or resonances. Once the filter has been implemented, it can be applied to an audio signal in real-time. The filtered signal is then output, with the unwanted high-frequency noise removed.

The project was broken down into the following parts in order to implement the Butterworth low pass filter:

1. Audio Frequency Analysis
2. Filter Design
3. Filter Implementation

To begin, I prompted the user for input for the .wav audio file and stored the input file into "filename". I then proceeded to call audioread to read data from the file named filename, and return sampled data, y, and a sample rate for that data, Fs. After that, using MATLAB's spectrogram() function I constructed a spectrogram depicting the Fourier Transform of the noisy conversation sound signal over time, spectrogram of output signal and spectrogram of output signal using difference equations. The zero-frequency component of the Fourier transform was moved to the array's center using the fftshift() function. The speech frequencies start at roughly 0.016 Hz, with a harmonic at 318 Hz, and the high frequency noise is at 4353 Hz, according to the DFT plot.

There are several challenges that I encountered when implementing a Butterworth filter to filter unwanted noise from an audio signal. These challenges include:

1. Choosing the appropriate cutoff frequency: The cutoff frequency is an important parameter of the Butterworth filter, as it determines the range of frequencies that will be attenuated by the filter. Choosing the appropriate cutoff frequency can be difficult, as it depends on the characteristics of the noise to be removed and the desired application. If the cutoff frequency is too low, the filter may not effectively remove the noise, while if it is too high, the filter may alter the overall frequency content of the signal.
2. Dealing with phase distortion: The Butterworth filter is a linear phase filter, which means that it does not introduce phase distortion to the signal. However, other digital filter design algorithms, such as the bilinear transform method, can introduce phase distortion. This can result in a shift in the phase of the filtered signal, which can affect the perceived quality of the audio.
3. Accounting for the limitations of digital filters: Digital filters, including the Butterworth filter, are subject to limitations such as finite precision and quantization errors. These limitations can affect the performance of the filter and result in unwanted artifacts in the filtered signal. Careful design and implementation of the filter are necessary to minimize these effects.
4. In the audio clip, the thunder is of very low frequency and up into the voice range. So, there is overlap between the thunder and the conversation. However, selecting a stopband frequency of 3500 Hz, did lessen the high frequency noise. As a result, I utilized the following numbers to form the filter: wp = 300, ws = 3500, Rp = 3, and Rs = 30.

Additionally, I also used the buttord() function [n, Wn] = buttord(Wp,Ws,Rp,Rs) to return the lowest order, n, of the digital Butterworth filter with no more than Rp dB of passband ripple and at least Rs dB of attenuation in the stopband. Wp and Ws are respectively the passband and stopband edge frequencies of the filter, normalized from 0 to 1, where 1 corresponds to π rad/sample. The scalar (or vector) of corresponding cutoff frequencies, Wn, is also returned. Then, used the output arguments n and Wn as inputs to butter() function to design a Butterworth filter. Then, to view the filter response from 0 to pi for 1000 points, I used the freqz() function. Then using the filter() function, I was successfully able to smooth out high-frequency fluctuations in the audio and remove periodic trends of a specific frequency from the data.

IV Plots and Results

Using MATLAB filtering commands to apply the filter designed above to the audio sample, we get the following plots that demonstrate the effectiveness of the filter at removing noise:

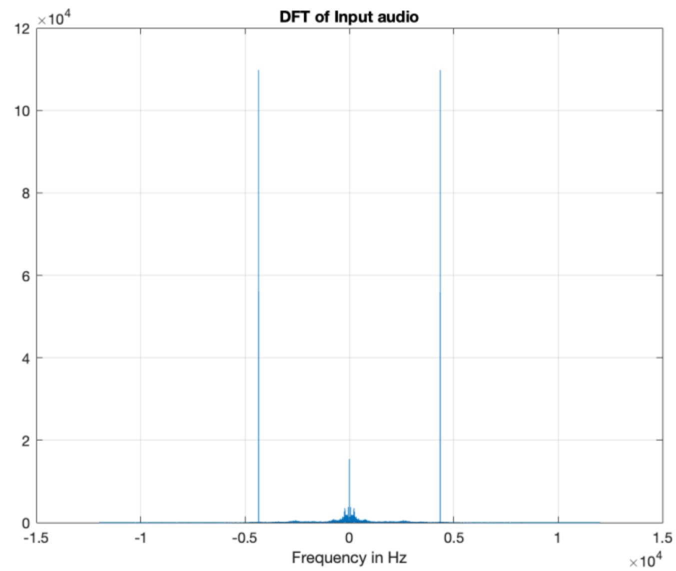

Figure 2. DFT of Input Audio File.

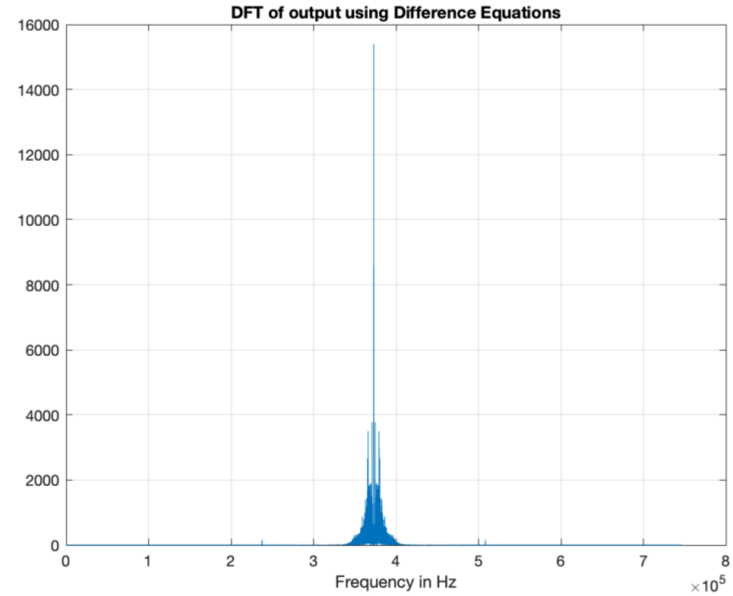

Figure 3. DFT of Output Audio File.

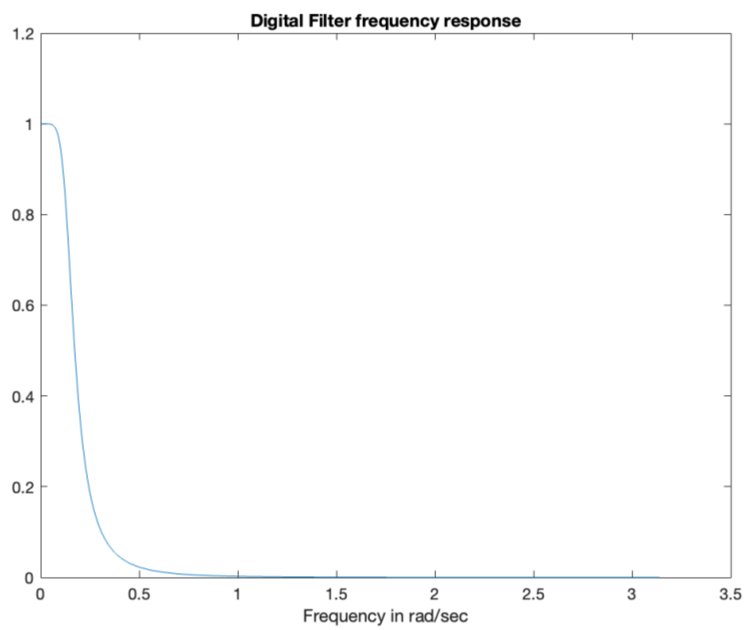

Figure 4. Digital Filter Frequency Response Plot.

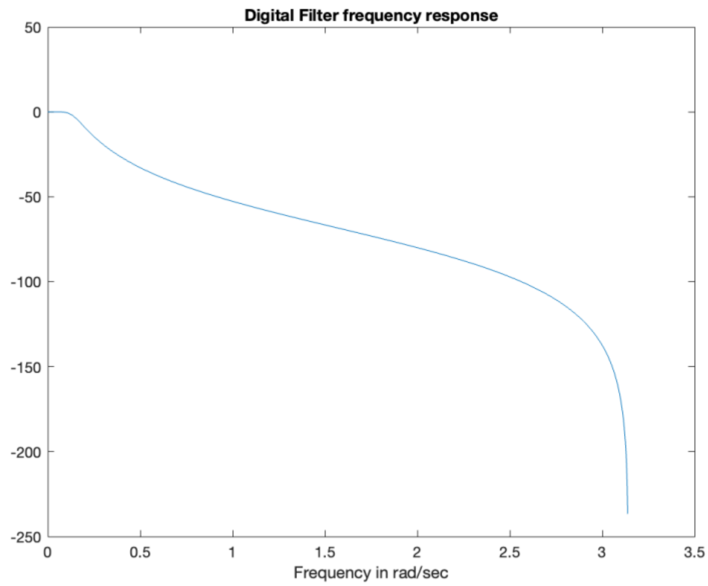

Figure 5. Digital Filter Frequency Response Plot.

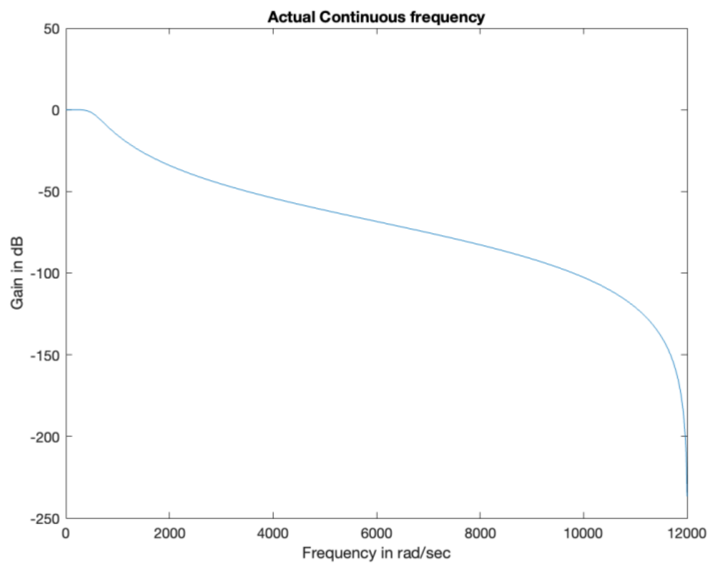

Figure 6. Actual Continuous Frequency Response Plot.

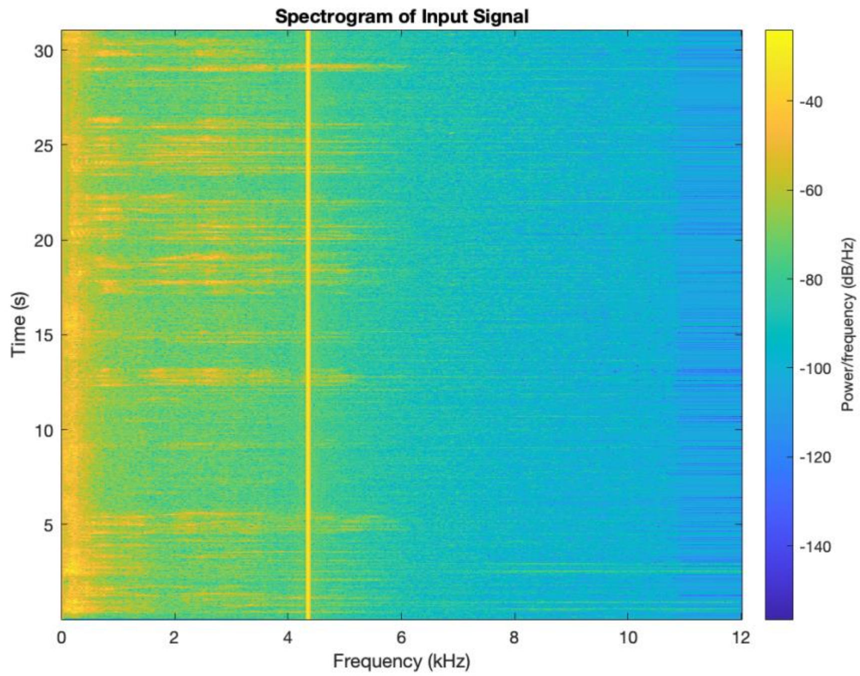

Figure 7. Spectrogram of Input Signal.

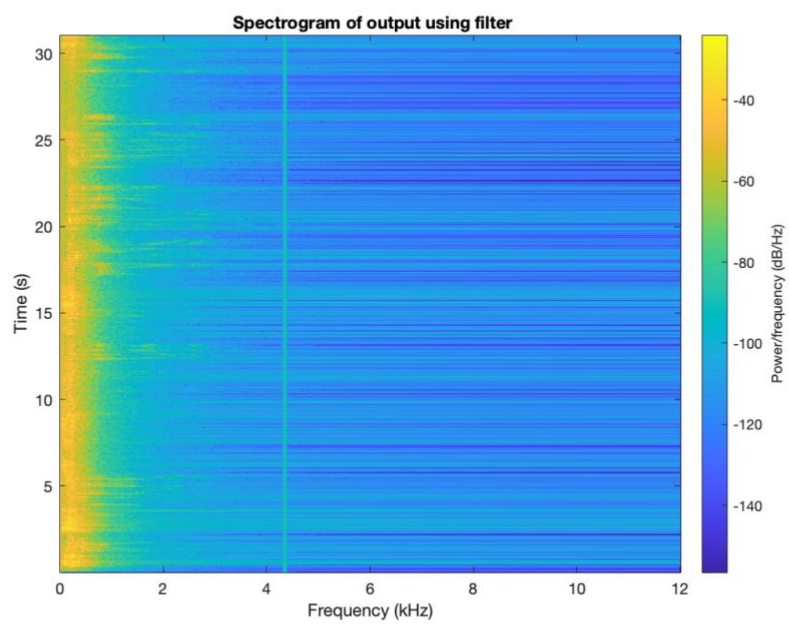

Figure 8. Spectrogram of Output Signal using Filter.

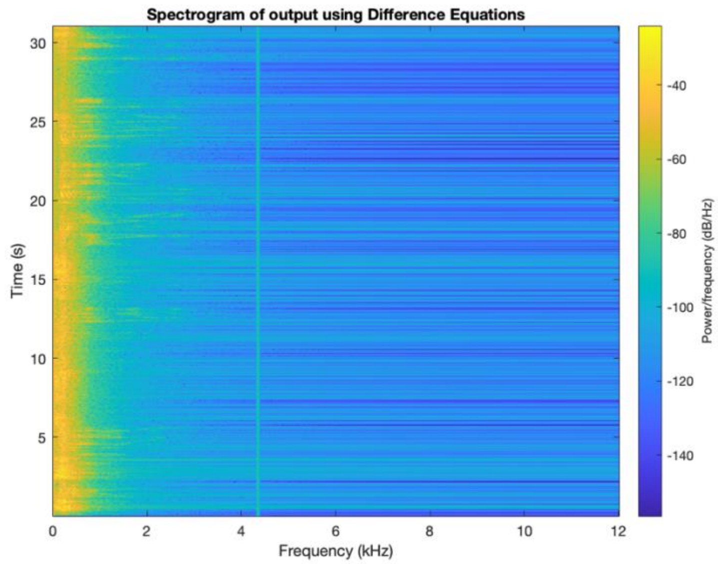

Figure 9. Spectrogram of Output Signal using Difference Equations.

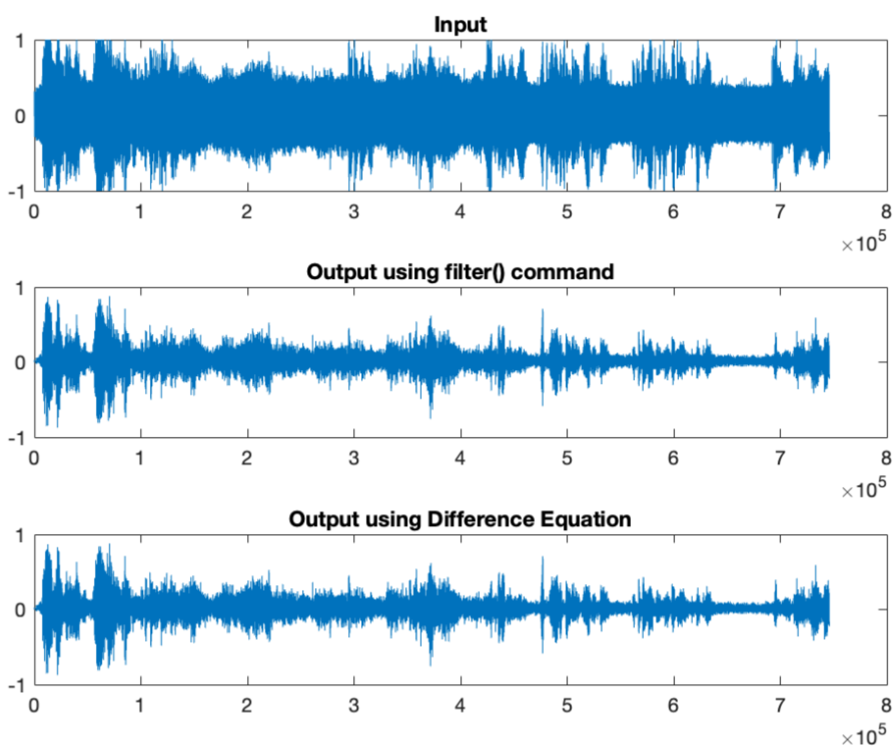

Figure 10. Comparison of magnitudes of the Input, Output using filter() and Output using difference equations.

V Discussion of Results

According to Figure 2, the conversation frequencies start at roughly 0.016 Hz, with harmonics at 318 Hz, while the high frequency noise is at 4353 Hz. The filtered audio file's DFT is displayed in Figure 3 after the Butterworth filter has removed some of the thunder and high frequency sound noise. We can infer from a comparison of Figure 2 and 3 that the filter preserved the desired speech on the passband frequencies while reducing the noise in the stop band frequencies. The order (N) and Cut-off frequency (Wc) are determined by the filtering procedure to be 3 and 0.0460, respectively.

The high frequency noise is depicted by the straight yellow line in Figure 7, between 4 and 6 kHz. But after applying the Butterworth filter to remove the noise, we can observe from Figure 8 and 9 that the line has somewhat diminished, indicating that the noise from the original audio sample has been successfully eliminated. The thunder is very low frequency and can even reach the vocal range, which is a drawback. Therefore, the dialogue and the thunder are concurrent. As a result, we are unable to totally block the thunder. However, setting up a stopband frequency of 3500Hz to create the filter, we were able to reduce most of the noise from the audio clips. According to Figure 10, which compares the original and filtered audios, most of the noise frequencies that needed to be eliminated have been successfully filtered as compared to the input figure, which retains the conversation frequencies.

We can infer from the filtered audio by listening to the output audio clip that it is actually a conversation between two characters Bob Woodward and Deep Throat from the 1976 film "All the President's Men." The conversation can be recorded as:

- Supposedly he's got a lawyer with $25,000 in a brown paper bag.
- Follow the money.
- What do you mean? Where?
- Oh, I can't tell you that.
- But you could tell me that.
- No, I have to do this my way. You tell me what you know, and I'll confirm. I'll keep you in the right direction if I can, but that's all. Just... follow the money.

VI Conclusion

In conclusion, a Butterworth filter is a type of low pass audio filter that is commonly used to remove unwanted noise from an audio signal. This filter is characterized by its flat frequency response, which means that it does not boost or cut any particular frequency range. The Butterworth filter is known for its simplicity and effectiveness in removing noise, and it is widely used in a variety of audio applications. One potential advantage of using a Butterworth filter for audio filtering is that it is a linear phase filter, which means that it does not introduce phase distortion to the signal. This can result in a clearer, more natural-sounding audio signal. However, there are also some limitations to using a Butterworth filter for audio filtering. One potential limitation is that the filter is not as effective at removing low-frequency noise, such as hums and buzzes, as it is at removing high-frequency noise. Additionally, the filter may not be well-suited for applications that require a sharp cutoff or a specific frequency response curve. In conclusion, the results of implementing a Butterworth filter for audio filtering are generally positive, with the filter being effective at removing high-frequency noise without introducing phase distortion. However, the filter may not be suitable for all applications, and other types of filters may be needed for certain tasks. By implementing my own Butterworth Filter, I was able to gain practical reinforcement of the theoretical concepts previously discussed in class, making it a great resource that allowed me to apply my knowledge in a tangible real-life situation. Through the experimentation and work done in this project, I was able to recognize the parts of the system used to remove noise from audio signals in signal processing.

VII Works Cited

Stability and control of a suspended Pendulum Fabry-Perot cavity. (n.d.). Retrieved December 12, 2022, from http://www.phys.ufl.edu/ireu/IREU2020/pdf_reports/Perry_Adelaide_FinalReport_2020.pdf

WP. Butterworth filter order and cutoff frequency - MATLAB. (n.d.). Retrieved December 11, 2022, from https://www.mathworks.com/help/signal/ref/buttord.html

X. Shift zero-frequency component to center of spectrum - MATLAB. (n.d.). Retrieved December 11, 2022, from https://www.mathworks.com/help/matlab/ref/fftshift.html

Xie, S. (n.d.). *Practical filter design challenges and considerations for precision adcs*. Practical Filter Design for Precision ADCs | Analog Devices. Retrieved December 11, 2022, from https://www.analog.com/en/analog-dialogue/articles/practical-filter-design-precision-adcs.html