MATLAB Filter Design Project

Part III: Digital Filtering

Through Analog to Digital Butterworth Filter.

Written by:

Eduardo Ramos

1001834020

CSE 3313

Professor Jon Mitchell

Introduction:

In the world of engineering, the manipulation of signals to transmit encrypted data across long distances has been a foundation for modern technology. Through these methods, we are able to reduce the burden of transmitting messages across long distances, which facilitates our daily operations. However, with every new solution stems a new challenge to overcome. From the perspective of signals, any object that can produce a signal is bound to overlap with other signals, creating the phenomenon which we refer to as “noise”. This phenomenon by itself is no reason for concern since everything in the natural world is bound to have an element of chaos. Yet, the problem becomes clear once we try to find a signal with desired information in this maze of signal interruptions. Thus, as a solution for this issue, we use filters as an instrument to isolate the signal we need. Filters come in many different variations but for our purpose, we will be focusing on Butterworth filters. The design of this type of filter is perfect for our experimentation since it offers a smooth transition between the passband and stopband, a feature that stands out from the rest. Through a low-pass Butterworth filter, we will be able to remove unwanted frequencies from our sample, which will make our signal much clearer to hear. In this report, we will be using such filter to remove a high-pitch noise from a conversation, where MATLAB will be our primary computational platform.

Approach:

To create an effective filter, we must understand how our signal behaves over time. While there are many effective methods to dissect a signal, a spectrogram would give us the best picture in terms of signal behavior. Thus, after creating a spectrogram of our signal, we observe the following picture:

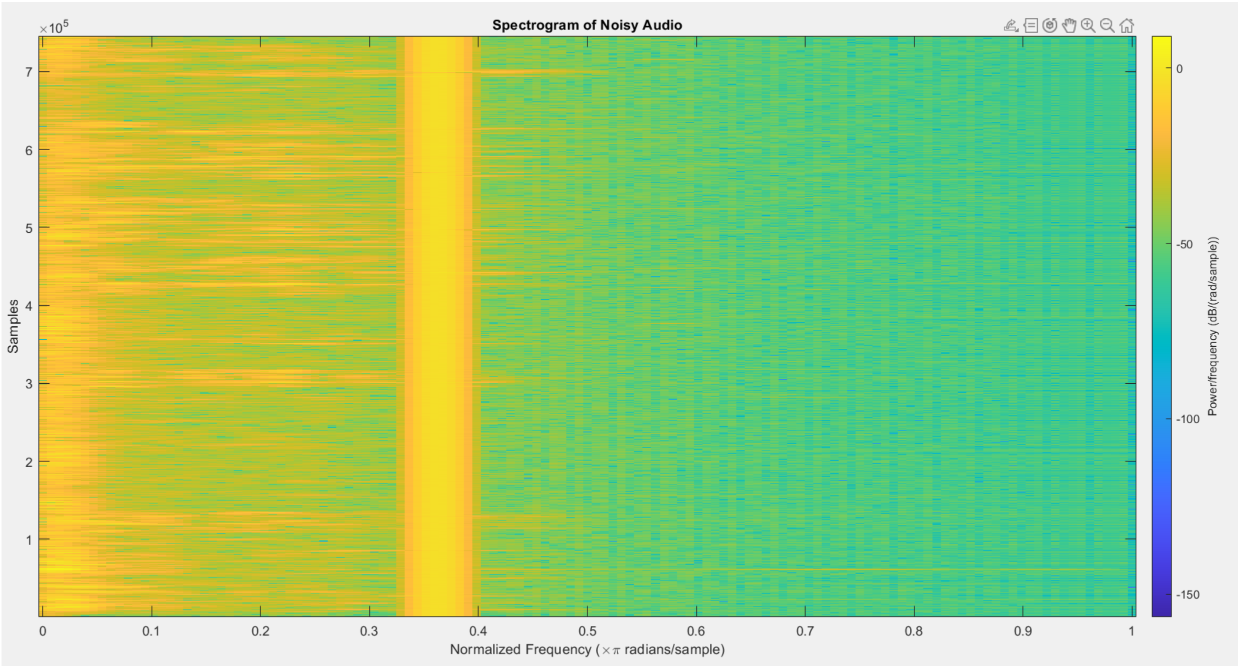

Figure 1: Spectrogram of signal

As we can observe, there is a high distribution of noise in the ranges of .33**π** and 0.4**π**. Thus, our ω_p would be approximately .30**π** and our ω_s would be around .34**π**. Once we have identified our frequencies, we need to identify at which gain our pass and stop bands should be located. To accomplish this goal, we need to look at the DFT of our signal, specifically the logarithmic function of the DFT. From this graph, we can differentiate the gain between our signal and the noise present within the sample. Thus, we observe the following graph of our signal:

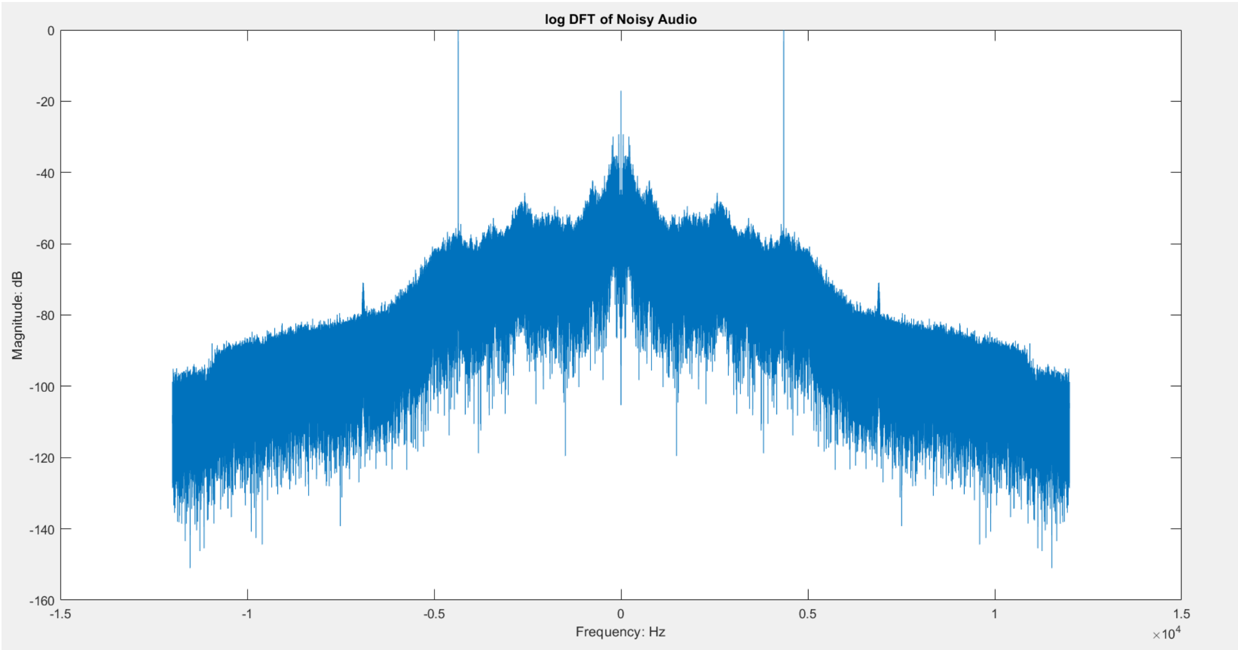

Figure 2: Logarithmic DFT

Upon analyzing the data , we observe that our signal gain is approximately around 29 dB, which would count as our delta p. For our delta s, we want to remove as much noise as possible. Therefore, a gain of 64 dB will allow us to remove unwanted noise from our sample, which drives us closer to our goal. Given that we found our parameters for our Butterworth filter, we are able to map its behavior using the command freqz(), in which we observe the translation from pass to stop band. Using such command yields the following result:

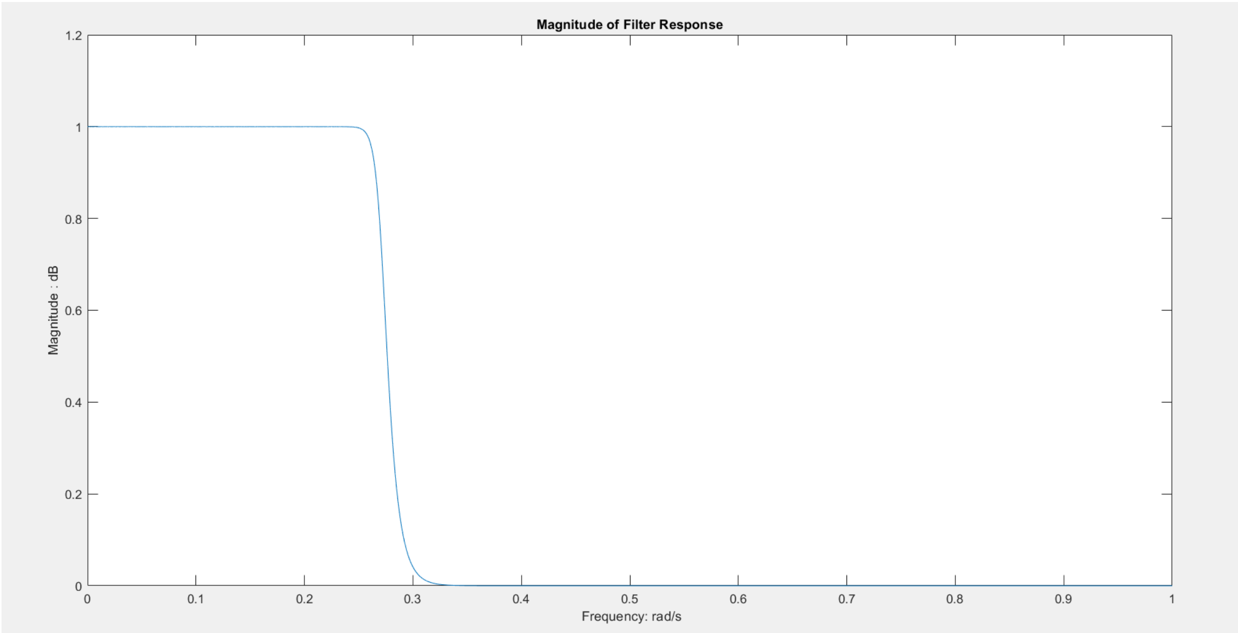

Figure 3: Magnitude of Filter

After completing the necessary calculations, we find that our cutoff frequency is around .2716 rad/s with a filter order of 28. The filter order determines the steepness of our transition curve, where a higher filter order would result in a steeper curve. We can also observe that our transition phase in approximately close to our pass and stop frequencies.

Henceforth, with this filter, we will be able to remove the unwanted signal concentration around our transition frequency.

Results:

After filtering our signal through our Butterworth low pass filter, we observe a substantial difference within our sample. Upon observation, most if not all of the high pitch noise has been eradicated from our sample, making the voice clearer to hear. Furthermore, we can see the impact of our filter by analyzing our signal image before and after the filtering. In so, we observe the following result:

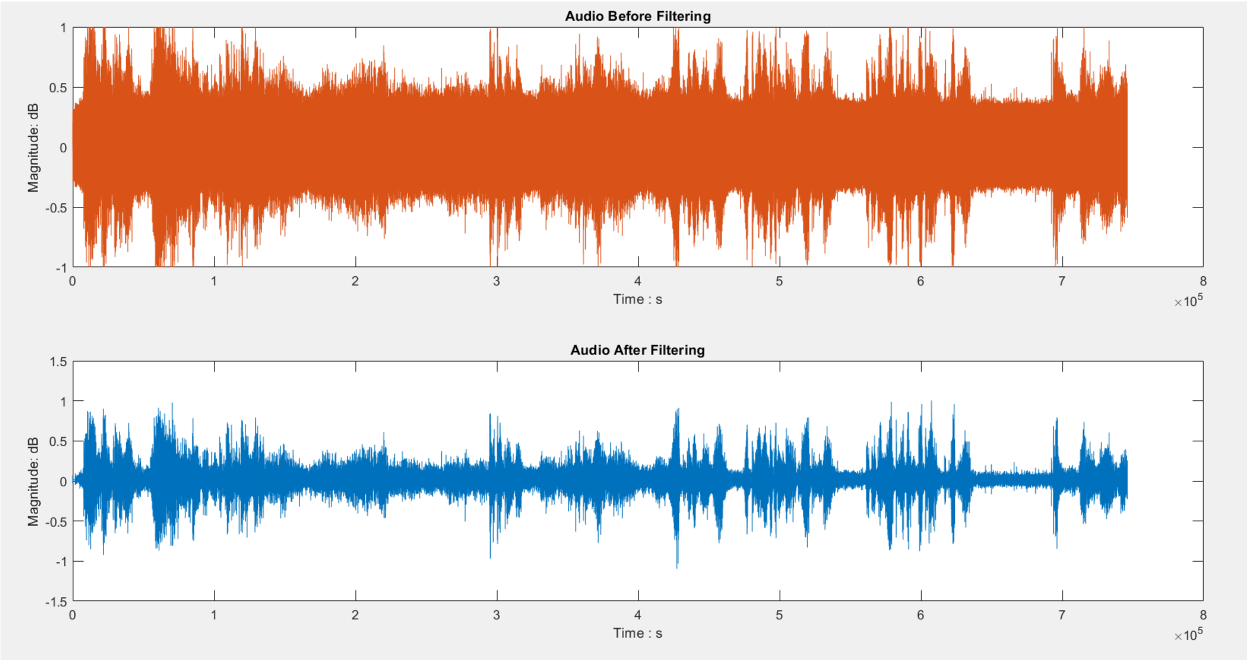

Figure 4: Before and after the filtering

Analyzing the previous graph, we can see how the high points of the signal were conserved while most intermediate sounds were removed from the sample. What this infers is that signals that fulfilled the requirements of our passband were passed along. On the other hand, signals with frequencies that fulfilled the requirements of the stopband were attenuated to the point of being unrecognizable. The effect of these two regions can be observed closely when taking the DFT of our new filtered signal. Performing this operation gives us the following results:

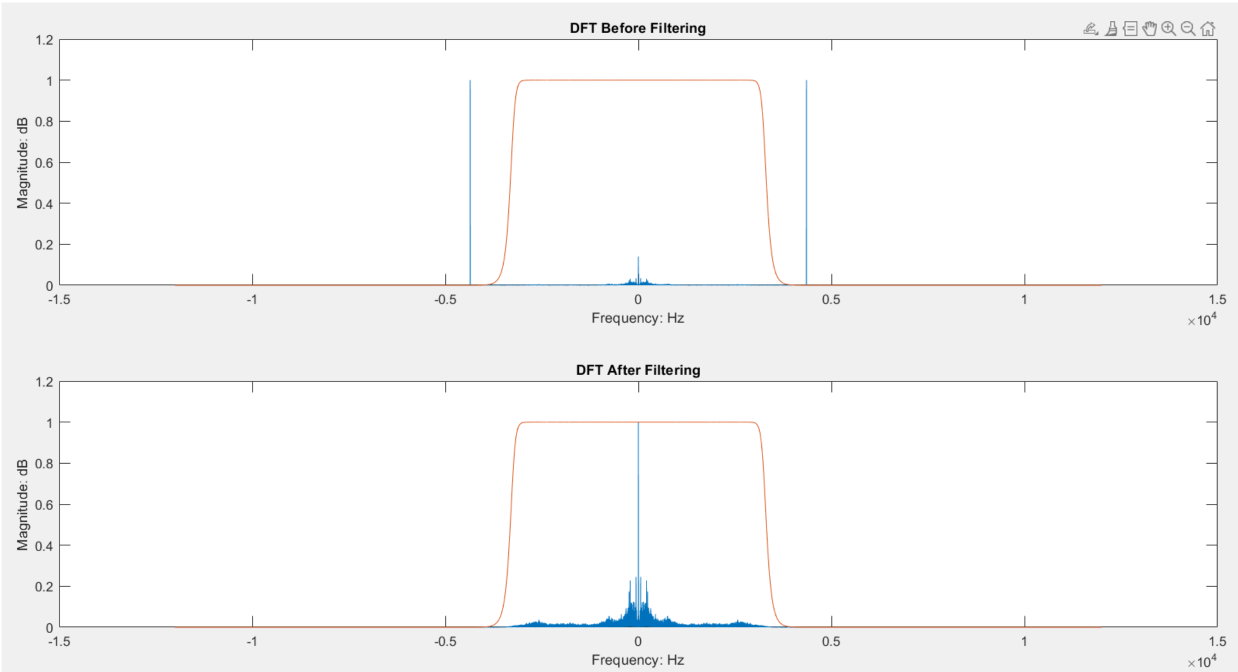

Figure 5: DFT signal before and after filtering

In this graph, we can observe how the noise frequencies outside our range have completely diminished due to the effects of our filter. Furthermore, we can see how the magnitude of our desired signal increased after applying our filter. Overall, the experiment conducted was an excellent demonstration of how filtering can make signal analysis a bit simpler. Of course, with all forms of computational technology, our filter will not remove all disturbances from our sample. However, it does complete the main goal of removing the high pitch noise from our sample. In a bigger sense, it demonstrates the efficiency of Butterworth filter and how it stands out due to its versatility in signal processing. Furthermore, its ability to balance between attenuation and signal processing makes it an important tool in diverse fields such as audio, communication, and control systems. However, like all filters, the usage of this filter must be weighed on the specifications needed for the task ahead. Overall, this filter is a well-rounded choice for developing fast solutions to our filtering needs.