Digital Signal Processing: Filtering Quail Calls

Fred Bolouri
Brennae Stinson
Anindita Das

Texas Tech University

October 2021

Table of Contents

1. Introduction

A data acquisition system consisting of four microphones, placed near several quail habitats, documented calls from these species over time. Each of these microphones recorded calls which are audible through .wav files. The signals retrieved from the data acquisition system almost always contain background noise from the quail environments such as trees rustling, other animals, and wind. The objective of our designed filtering system was to reduce the background noise from the signal so the call would be more apparent to detect both audibly and visually. The following report details intermediate results of this filtering and analysis of the results obtained from the filter. This analysis includes retrieving and inspecting the time and frequency domain graphs, and the resulting .wav files of the signals.

Let us first define a noisy signal and a clean signal. As mentioned above, the background noise is present in almost all audio files. The reason behind this is that some quails are closer to the microphones while making the calls, hence are more apparent in a recording (the value of their frequency-domain component is higher), whereas, other quail calls made from a greater distance, making the background noise is much more prominent when the quail calls are acquired from a greater distance. Therefore, for the purposes of this project, we will define a noisy signals as ones where the quail calls were far from the microphones and a clean signals as ones where the quail calls were close to the microphone.

Both noisy signals and clean signals were analyzed while coming up with an implementation for this filter. The difference between these two types of signals can clearly be seen in their time domain graphs as shown in Figure 1. The location of quail calls can be seen in the clean signal while they are more difficult to see in the noisy signal's time domain graph. The clean signals obtained from the acquisition system were used to estimate the bandwidth of the quail calls. Similar bandwidths were determined from neighboring microphones. The Spectrogram from this signal clearly shows that the locations of the calls are between 1000 and 1500 Hz to around 3200 Hz, however the part of the call we wish to preserve is between 1850 and 2150 Hz as shown in Figure 2 below (The yellow color in the spectrogram - This frequency range is what we hear and recognize as a "Quail Call" when listening to the audio files). Another way of analyzing the clean signal is by taking its Fourier Transform as was done in Figure 3. The spikes of the FFT graph illustrate the indices of the quail calls.

To confirm the bandwidth of the quail calls the FFT of a noisy signal was taken also depicted in Figure 3. From this FFT the physical (f_0) and normalized frequency(ω_0) of the noisy signal were calculated giving an approximate location of the frequencies at index k = 1932. Using equation 1, the physical frequency was calculated to be 1855 Hz, confirming the bandwidth shown by the spectrogram.

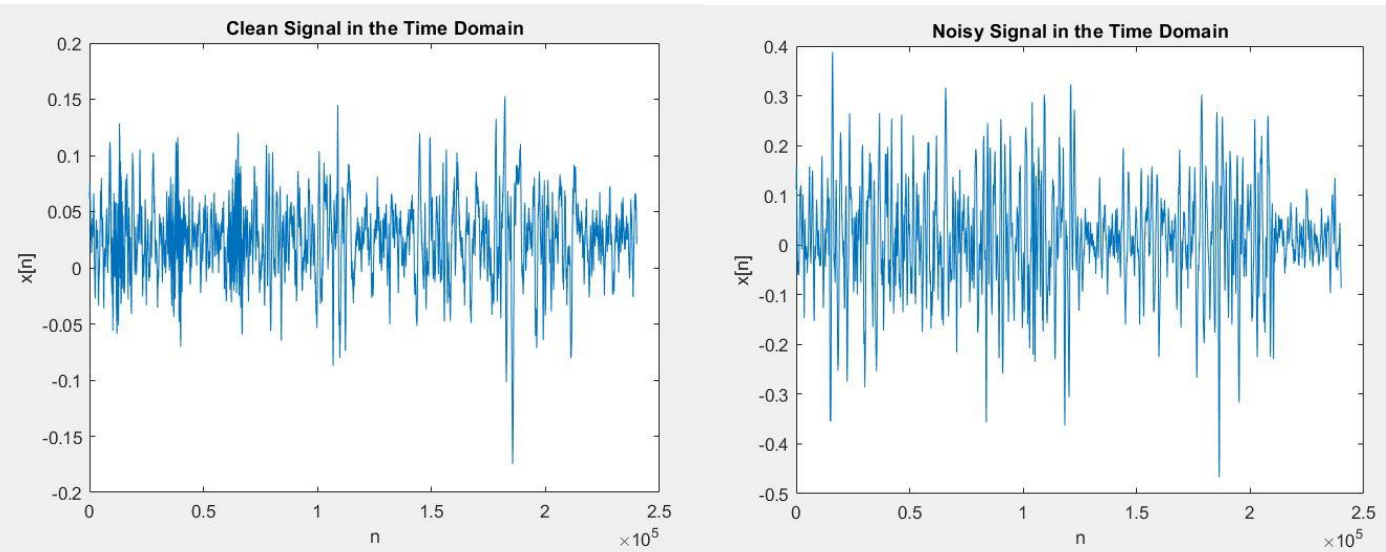

Figure 1: Clean and Noisy Signals in the Time Domain

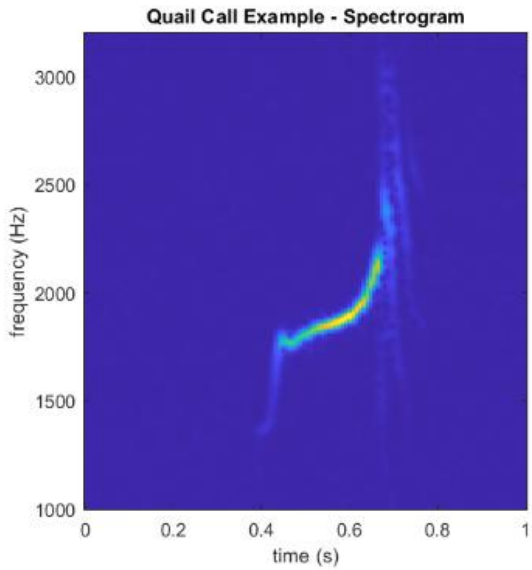

Figure 2: Spectrogram of Quail Call [6].

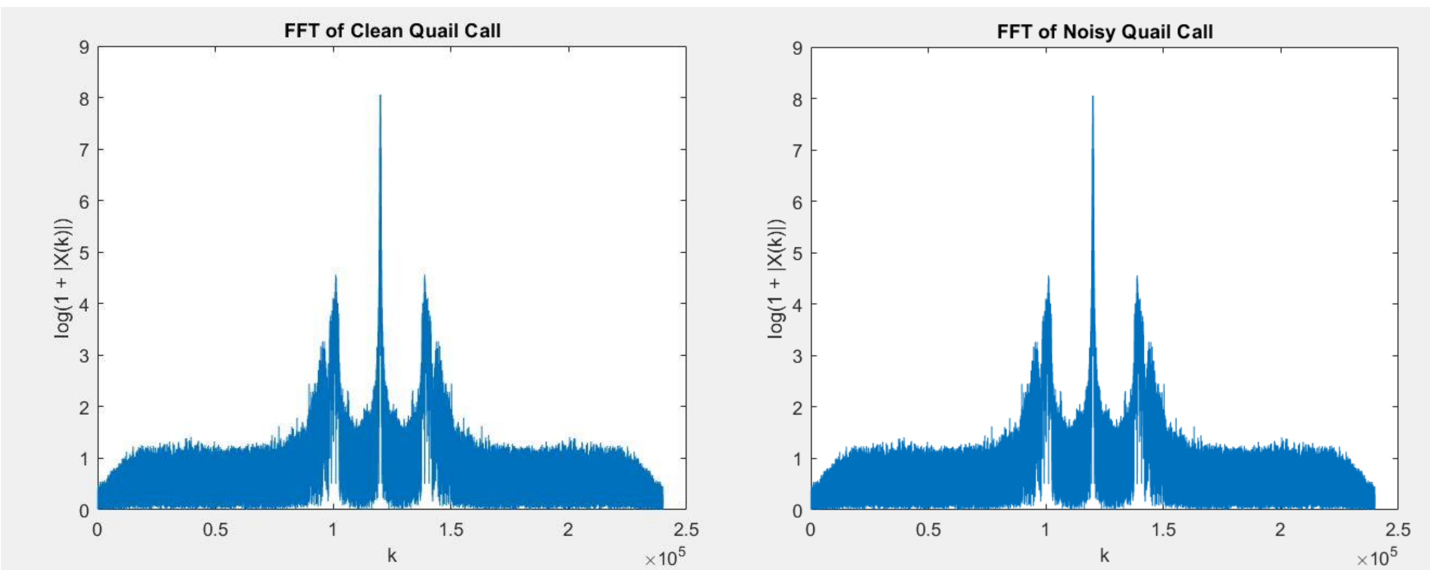

Figure 3: FFT of Clean and Noisy Signals [2].

$$\frac{f_0}{f_s} = \frac{k}{N}$$

Equation 1: Physical Frequency [1].

2. Filter Design

To begin the filter design process, we tested out several Band-pass filters with the passband located within our bandwidth. These filters provided signals with relatively less noise but still contained enough background noise to overpower the quail calls. Due to these unsatisfactory results, we moved to try an alternative approach to this task. We started by assuming that the noise model of our signal was an additive in nature, meaning the signal is a summation of our clean signals (containing quail calls) and background noise. The additive model used for the design is shown in Equation 2. Then, we decided to construct a Band-stop filter at our bandwidth to attenuate the quail calls to lower levels resulting in a signal containing just the noise using the MATLAB filterDesigner tool [4]. Due to limitation of this project, a filter of order 6 in direct form II structure designed. The signals were sampled at 24 kHz and the cutoff frequencies were set at the bandwidth of the quail calls. Using the filter() function, the filter was applied to the noisy signal and then subtracted from it, resulting in a signal with reduced background noise [3].

$$Noisy\ Signal = Clean\ Signal + Background\ Noise$$

$$Filtered\ Signal = Background\ Noise$$

$$Clean\ Signal = Noise\ Signal - Filtered\ Signal$$

Equation 2: Additive Noise Model [7].

The frequency response, phase response and z- plot of the constructed filter is depicted below in figures 4 and 5. The frequency response shows a notch-like response at our calculated normalized frequency of 1850 Hz [1]. The phase response of the filter depicts an exponential decay as the normalized frequency is approached, and a spike at the frequency location. The z-plot shows three poles and three zeros located along the unit circle. By analyzing the characteristics of these plots, the accuracy of the filter can be determined. Since each of these plots behaved accordingly, we decided that the designed filter would be adequate for our goal.

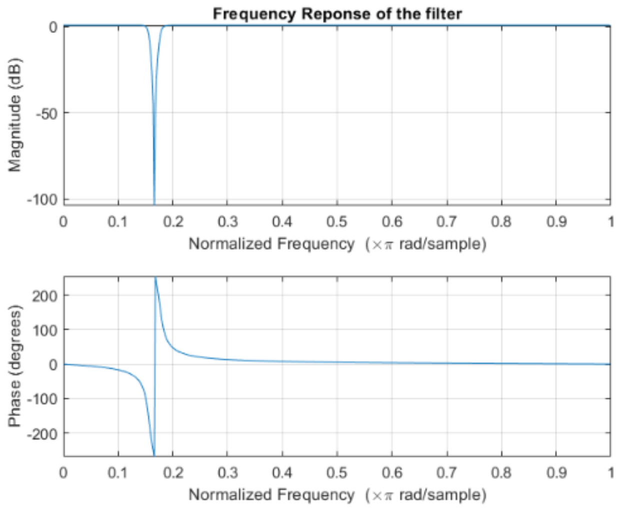

Figure 4: Frequency and Phase Response of Filter

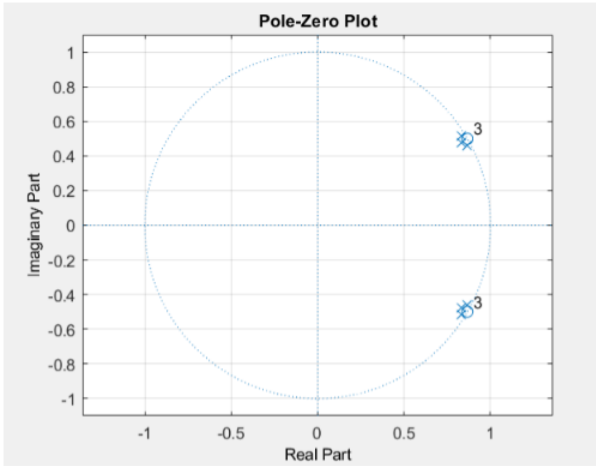

Figure 5: Pole-Zero Plot of Filter

3. Resulting Signal

After applying the filter to the signal, the filtered signal was subtracted from our original signal dampening the background noise within it. Evidence of this effect can be seen through both the Time Domain and FFT plots of the signal as illustrated in Figures 6 and 7. From the Time Domain graph, the location of the calls can easily be detected. The Fourier Transform of the signal shows more prominent spikes at the frequency location of the quail call. Based on these figures and the resulting .wav file of the clean signal, we concluded that our filter design technique worked properly

For a complete demonstration of our work, please refer to the live script "Project1_Part2.mlx"or the pdf "Project1_Part2_MATLABoutput.pdf".

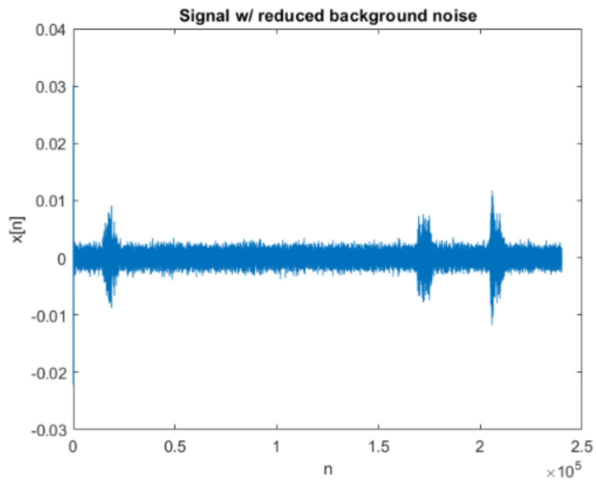

Figure 6: Signal with Reduced Background Noise

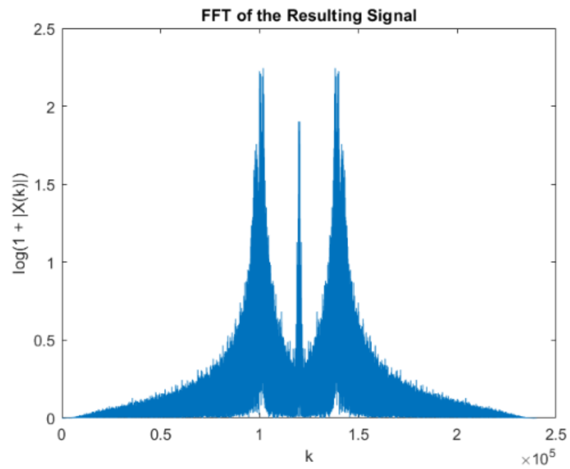

Figure 7: FFT of Resulting Signal [2].

References

1. Karp, Tanja. Physical Frequency Equation and Notch Filter.
2. MATLAB Documentation. Fast Fourier Transform. https://www.mathworks.com/help/matlab/ref/fft.html?s_tid=doc_ta.
3. MATLAB Documentation. Filter. https://www.mathworks.com/help/matlab/ref/filter.html?s_tid=doc_ta.
4. MATLAB Documentation. filterDesigner. https://www.mathworks.com/help/dsp/ref/filterdesigner.html?s_tid=doc_ta.
5. MATLAB Documentation. Findpeaks. https://www.mathworks.com/help/signal/ref/findpeaks.html.
6. MATLAB Documentation. Spectrogram. https://www.mathworks.com/help/signal/ref/spectrogram.html?s_tid=doc_ta.
7. Sari-Sarraf, Hamed. Additive Noise Model and Call Detection.

Appendix A

In the following Appendix, we continued testing our filter design technique by applying it to a clean signal. The resulting signal shows prominent quail calls in the Time Domain as shown in Figure 8. The .wav file created from this application, had very little background noise.

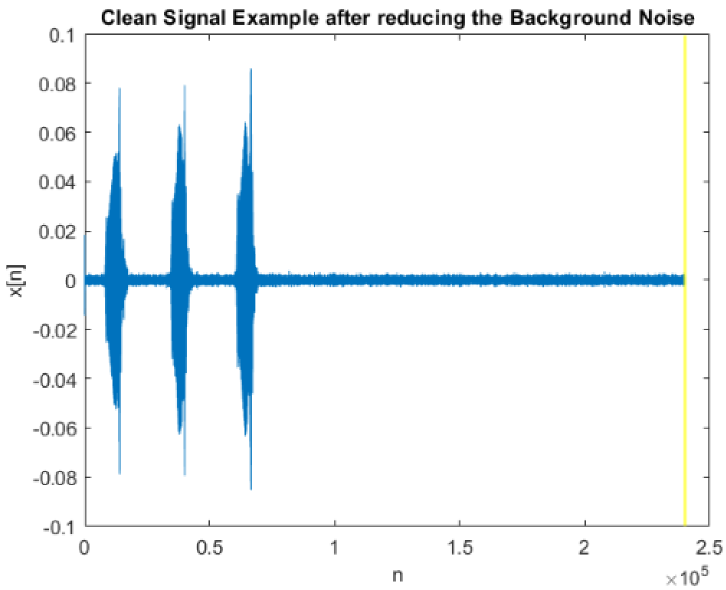

Figure 8: Clean Signal Time Domain Graph After Reducing Background Noise.

Appendix B

The following Appendix details a Call Detection Algorithm constructed using the Local Maxima of the signal. After applying the filter to a clean signal and a signal containing more background noise, The calls can be detected using the findpeaks() function in MATLAB. An illustration of this algorithm applied to each of the signals is depicted in Figure 9.

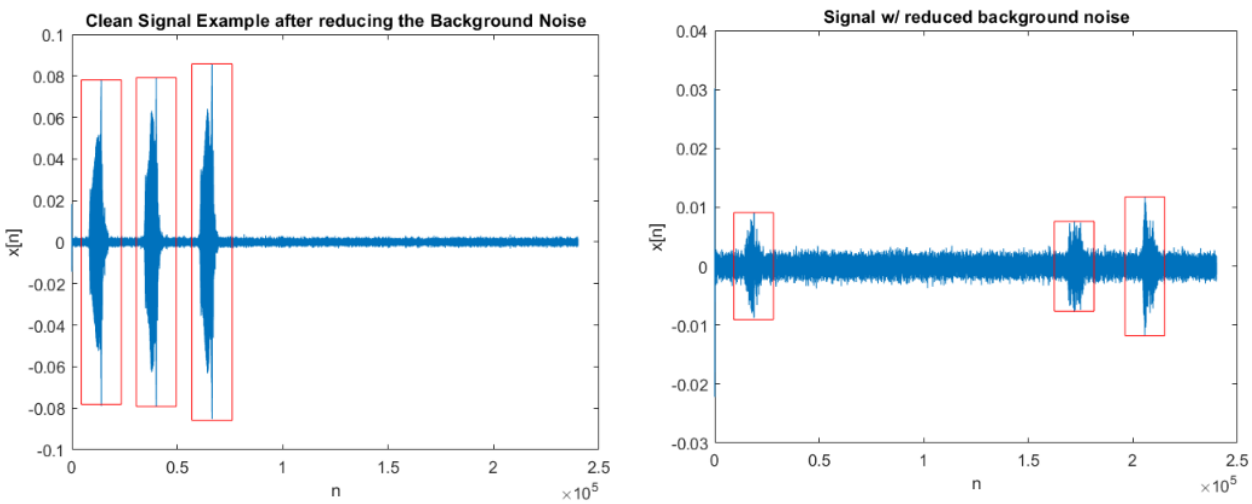

Figure 9: Local Maxima Algorithm Visualization [7].