Suppressing a Sinusoidal Interference Using a Windowed Integrator

Abraham Paroya & Jalen Guan
paroyaabraham@temple.edu, jialiang.gaun@temple.edu

I. Introduction

The purpose of this project was to create a windowed integrator system that suppresses a sinusoidal interference in a .wav audio file. When the original audio file is played the voice of a man mixed in with a loud tone can be heard. The goal of this project is to remove this tone using a windowed integrator system. To create this system, a basic understanding of signal processing in MATLAB is necessary. The ability to use loops and displaying data in MATLAB is also critical in analyzing and modifying the original and denoised sample once passed through the system. Also, an understanding of Fourier transforms, and the data produced from using a transform is beneficial in finding a correct window size for the integrator.

II. Methods

The window integrator utilizes the input-output relation in the equation below to traverse through a vector and sum together all the values within a vector.

$$y(n) = \sum_{k=n-N+1}^{n} x(k) \quad (1)$$

In this summation formula, N is denoted by a number to specify the window size for the operation. When the summation is calculated all values in a window when k < 1 will be removed from the original sample vector. To complete this operation, first the sampled data from the wave file should be read into an array. Then a for loop is created to loop through the data 18,000 times for each sample and then sum together the pieces within that window size. For example, if N = 7, starting from the beginning of the data vector, the equation below to represent what values should be included in the summation and what values will be negated in the original data vector.

$$k = n - N + 1 \ (2)$$

Subbing in N = 7 into equation 2:

$$k = n - 7 + 1$$

This will make the values 1 to 6 equal to 0 in the original data. For a window of 8 this will be 1 to 7 and, for window of 9 it will be 1 to 8. Essentially this windowed integrator will run by looping through the sampled data and summing together either 7, 8 or 9 pieces of data depending on the N value. Then stores it into the index which will be from 1 to 18,000. Once the new vector of data is produced, a Fourier transform must be done on the data to get the frequency response of the filtered data to see if the tone has been removed or not.

III. Results

First the graph of the frequency response of the original sample prior to removing the tone is produced. Observing this graph will give a better understanding of the frequencies that compose the signal and what needs to be removed.

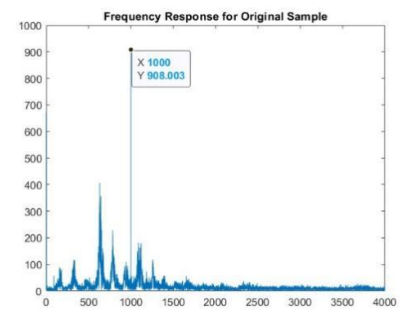

Figure 1.1 – Frequency Response (Original Sample)

As shown, the signal with a frequency of 1000 Hz has the most power in the overall audio file. This can be perceived as the obstructing tone when listening to the audio. When the integrator is run on the sample with the varying window sizes and then a fast Fourier transform is performed on the data, the following result is produced.

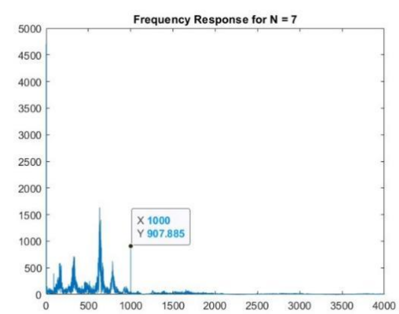

Figure 1.2 – Frequency Response (N=7)

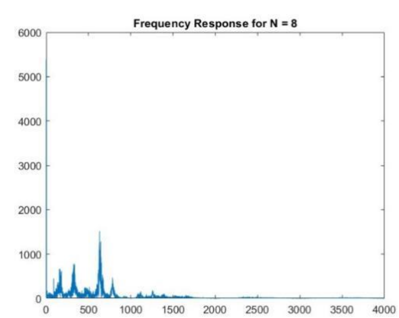

Figure 1.3 – Frequency Response (N=8)

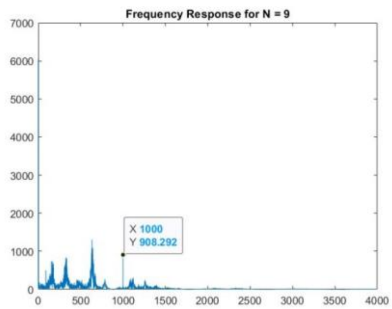

Figure 1.4 – Frequency Response (N=9)

As shown in the above figures, the tone that was discovered with a frequency of 1000 Hz was present when the windowed integrator used a window size of 7 and 9 but completely disappeared with a window size of 8. This is evident when the audios from all three of the different window sizes are played back. For 7 and 9 the tone is still largely present and audible but for a window size of 8 is not. This can also be proven mathematically when using an equation to calculate window size:

$$Window\ Size = \frac{Sample\ Rate}{Frequency} \quad (3)$$

Where T is the period of the sinusoid that will be removed from the windowing and the sample rate is the sample rate from the audio sample. The sample rate of the original sample is given within the .wav file as 8000 Hz and the frequency of sinusoid designated for removal is 1000 Hz as shown in the figure 1.1. Substituting these values into equation 4, a window size of 8 is the result which matches our results.

IV. Discussion

In this project, we set out to create a windowed integrator that will remove a tone from an audio file. We found an experimental approach to finding the correct window size to use to remove a tone using this system. By using different values for the window size, we were able to view the frequency response using a fast Fourier transform of the output and determined what changes had been made to the data. We found that the size 8 window integrator was the best in interference suppression which matched our hypothesis because it matched with the sampling frequency of the sample and can grab the peaks of the sample better. These results are very useful and insightful into the process of signal processing. These results show the importance of frequency response plots and how they can be used to manipulate data.