Speech Scrambler

Ivan Arevalo

Description:

The purpose of this report is to understand and implement a speech scrambling method with matlab. We will be deriving the speech scrambling method through analysis in continuous time and discrete time. We will then implement our method in matlab and write a script that can receive and send scrambled signals. The idea of scrambling a signal is to have a secure way to transmit information without concern for the information to get intercepted.

Method:

Starting out with a continuous time signal and taking the fourier transform we get figure 1. In this figure, we show a triangular frequency spectrum to illustrate this method.

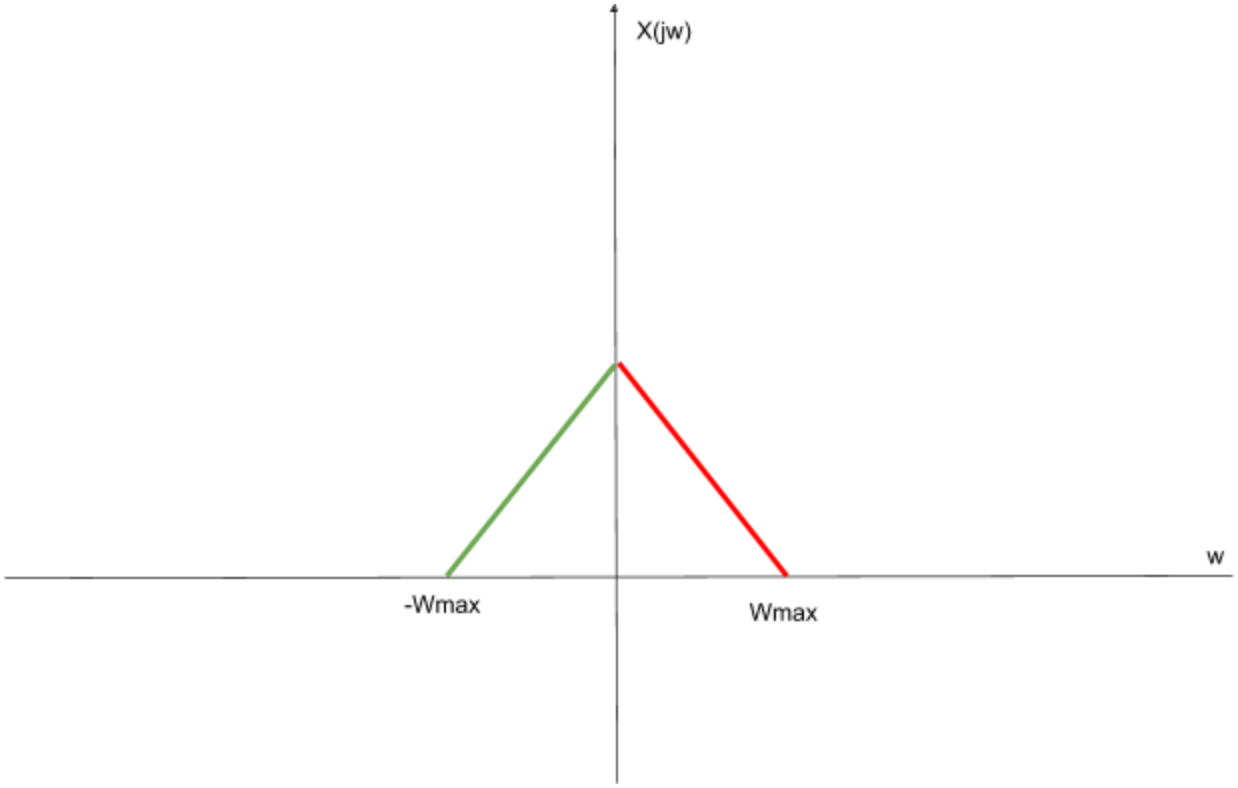

Figure1

Next, we must modulate the signal by $cos(w_c t)$ to make shift the frequency of the signal to $\pm w_c$ as shown in figure 2. This will also decrease the magnitude of the frequency spectrum by half, but since we are able to amplify the resulting signal, it does not affect our result.

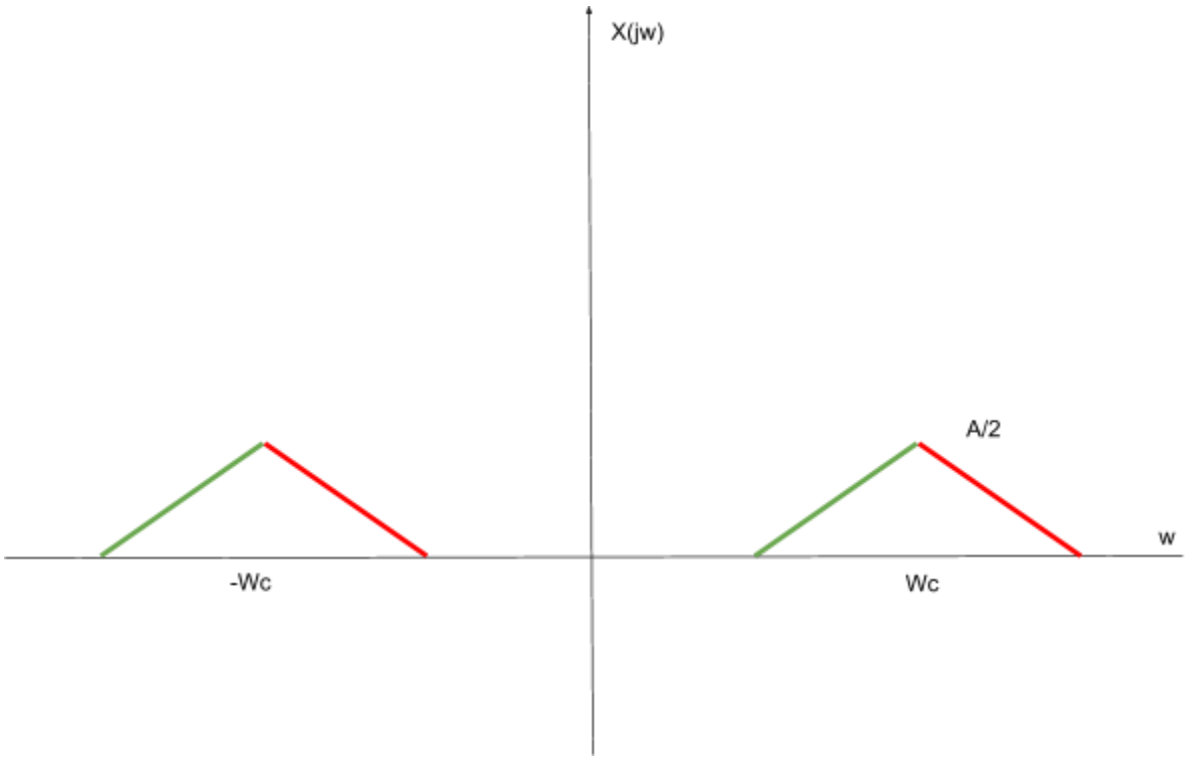

Figure 2

Next, we need to highpass our signal with a cutoff frequency of Wc as shown in figure 3.

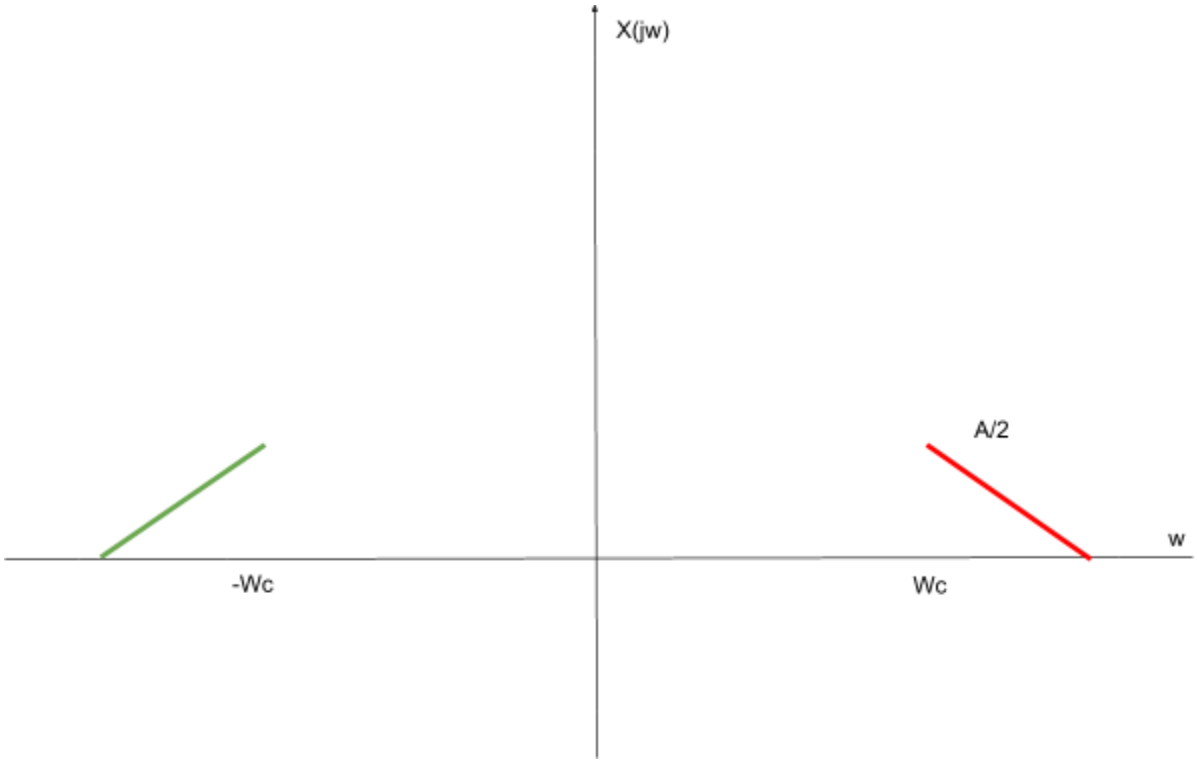

Figure 3

We must modulate the signal once more by $cos(w_c + w_m)$ and get the result shown in figure 4 and low pass the signal as in figure 5.

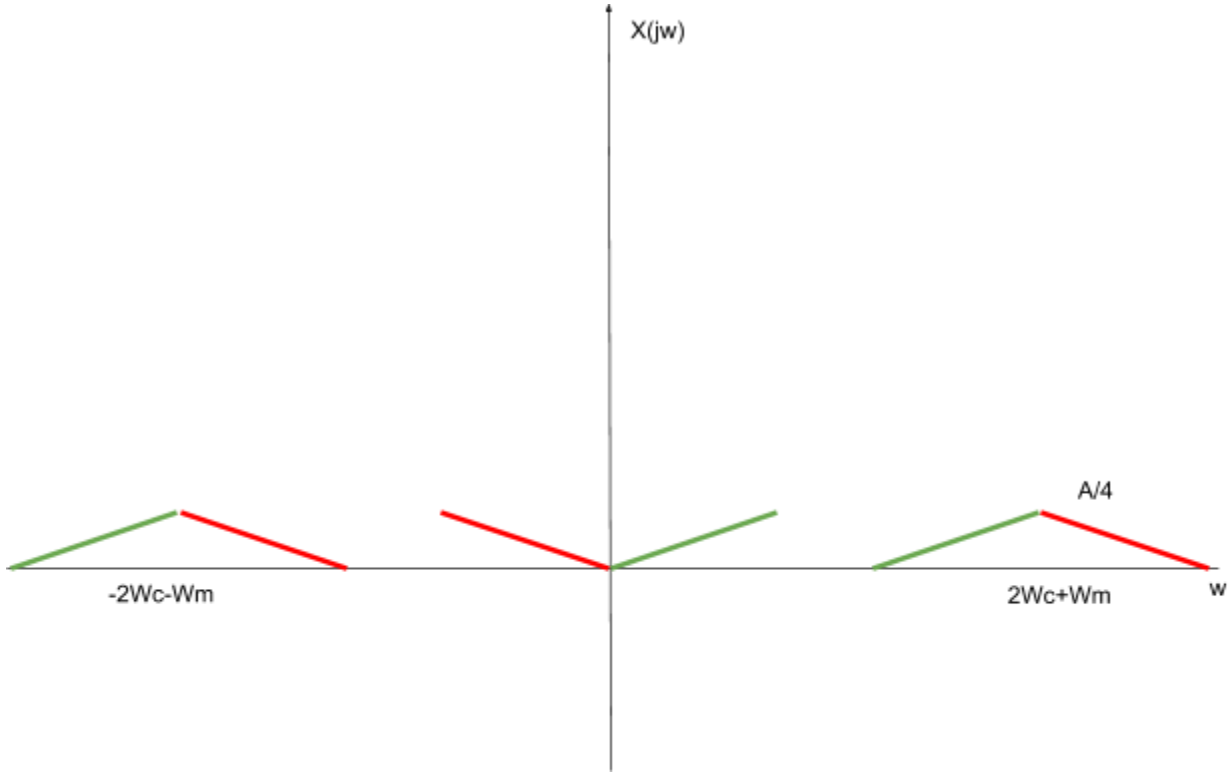

Figure 4

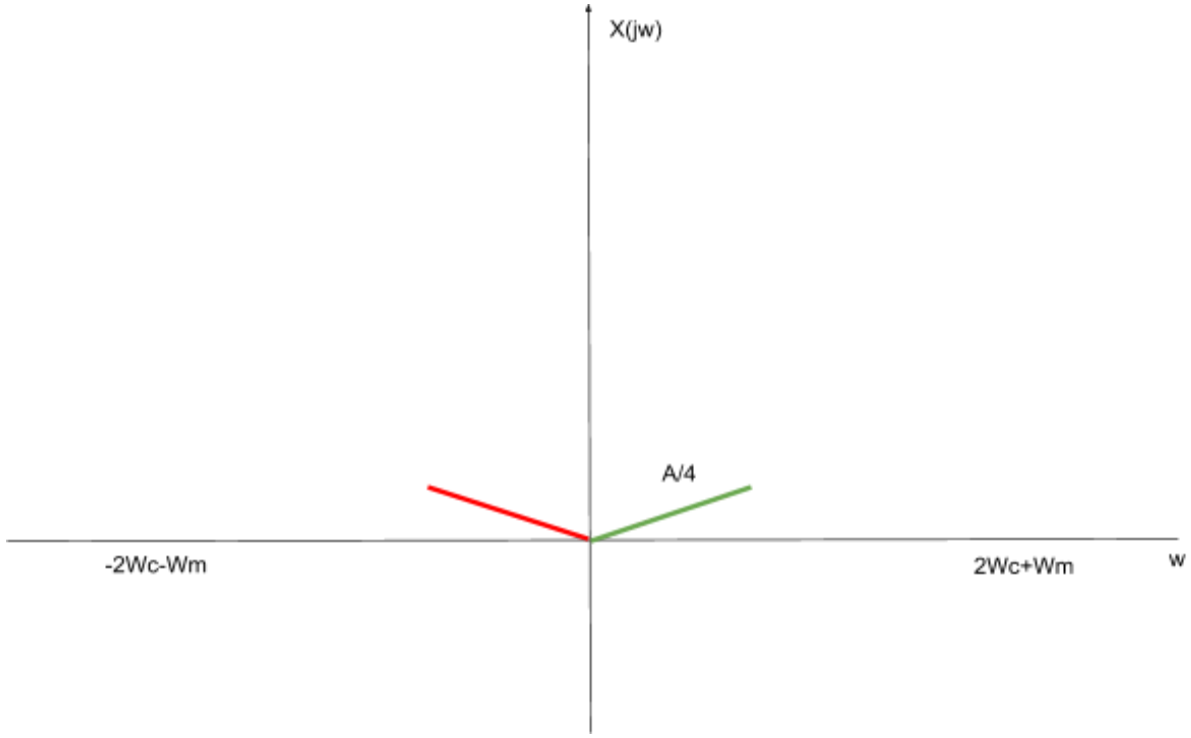

Figure 5

At this point, our signal has been scrambled as we see the higher frequencies have greater magnitude than the low frequencies. Furthermore, we can amplify the signal to get the same magnitude signal back. We may also analyze this method once sampled and turned into a discrete signal. We will see this proves to be easier to implement than the continuous time case because of the periodic nature of discrete time signals. Given a sampled signal and taking the Fourier Transform of it, we arrive at figure 6. It illustrates the periodicity of the signal in the frequency domain and gives us some insight into easily scrambling the signal.

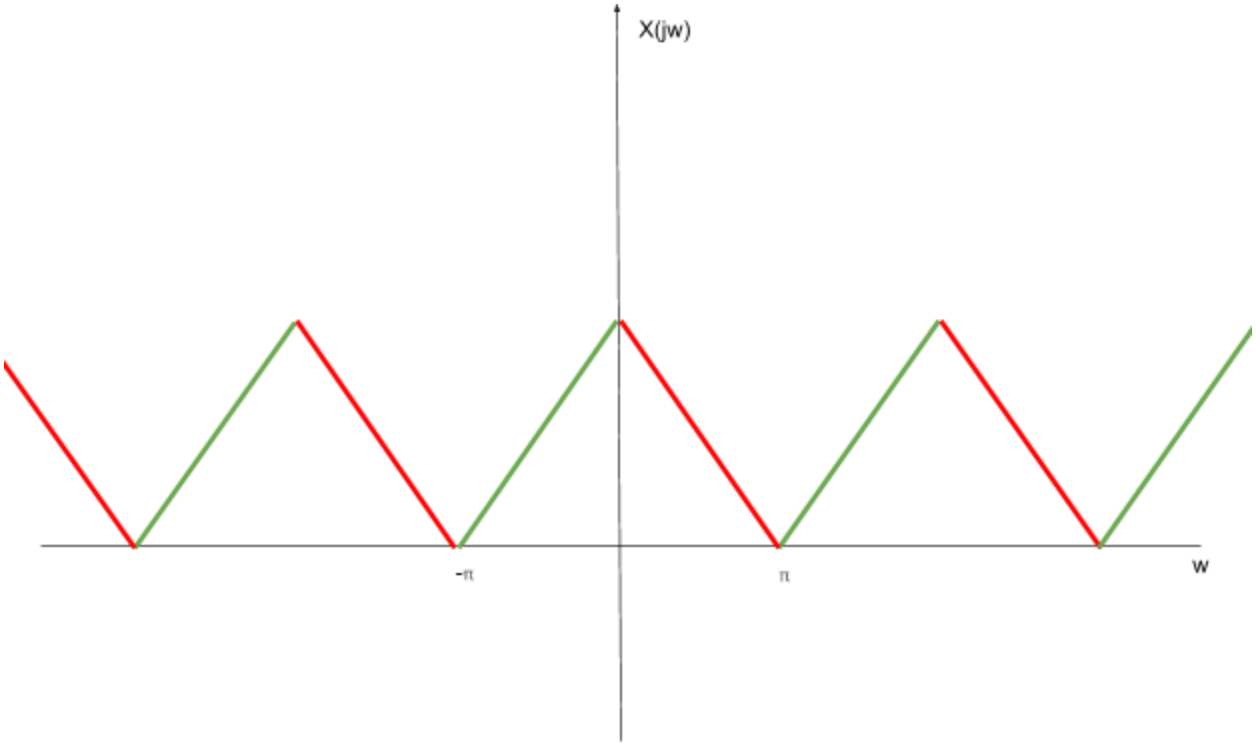

Figure 6

Similar to our continuous time analysis, we will modulate the discrete time signal to shift the frequency spectrum, and since our signal is periodic, we only need to modulate the signal once by $e^{j\pi n}$ as shown in figure 7 and then filtered with a low pass filter to grab the scrambled frequency (figure 8). For our purposes, since we are using matlab to implement this method, we will need to use FFT. The method in this case, is quite similar but keeping in mind that FFT goes from 0 to N-1 instead of $-\pi$ to π. and we need to multiply the signal by the correct complex exponential $e^{jw_{masx}n}$.

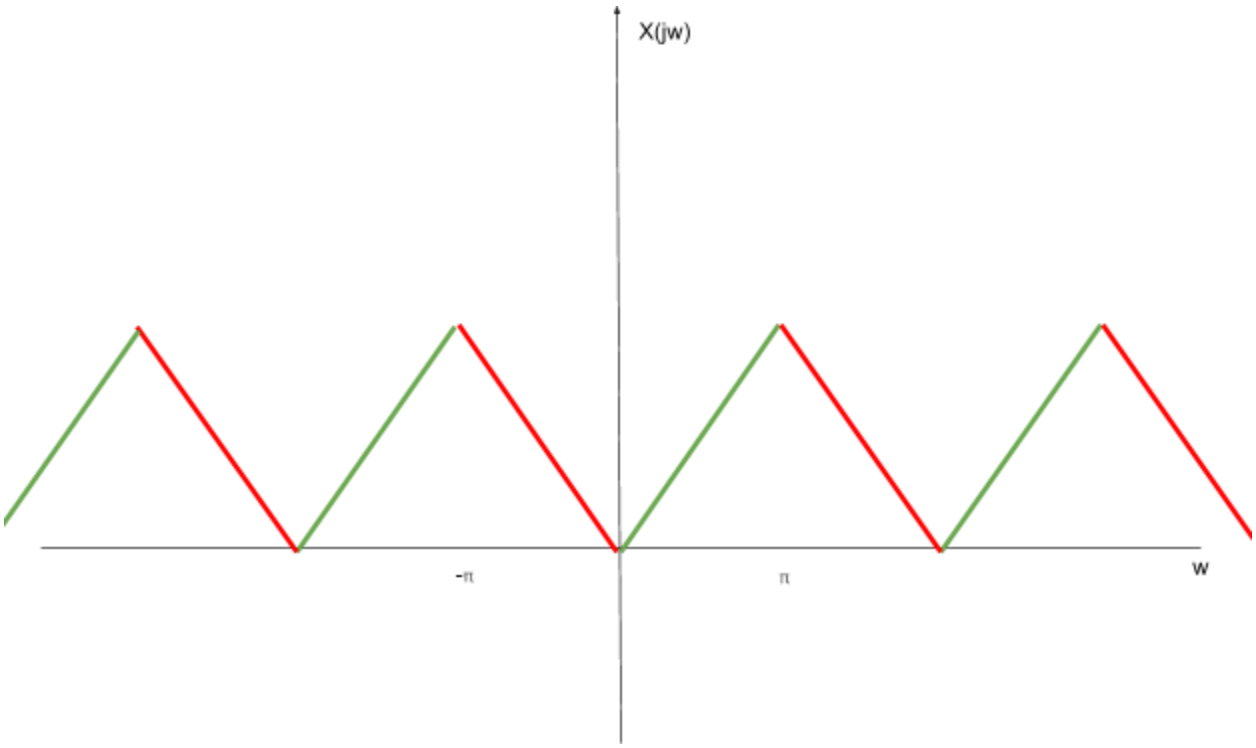

Figure 7

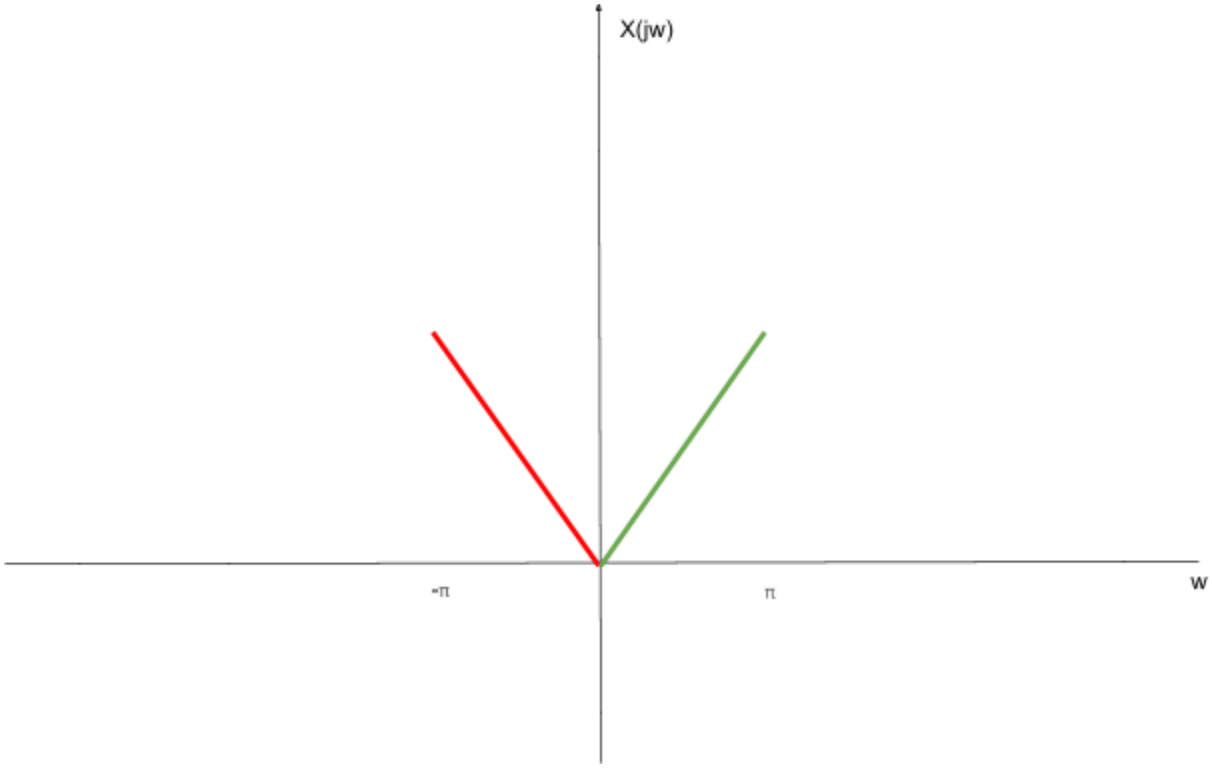

Figure 8

Outcome:

Figure 9 and 10 show the frequency spectrum of the original audio signal and the scrambled signal respectively. We can see how, just as as depicted in our derivation, the frequency spectrum has been scrambled enough to make the signal incomprehensible. Furthermore, notice that if we were to repeat the process once again, we would get our original signal back.

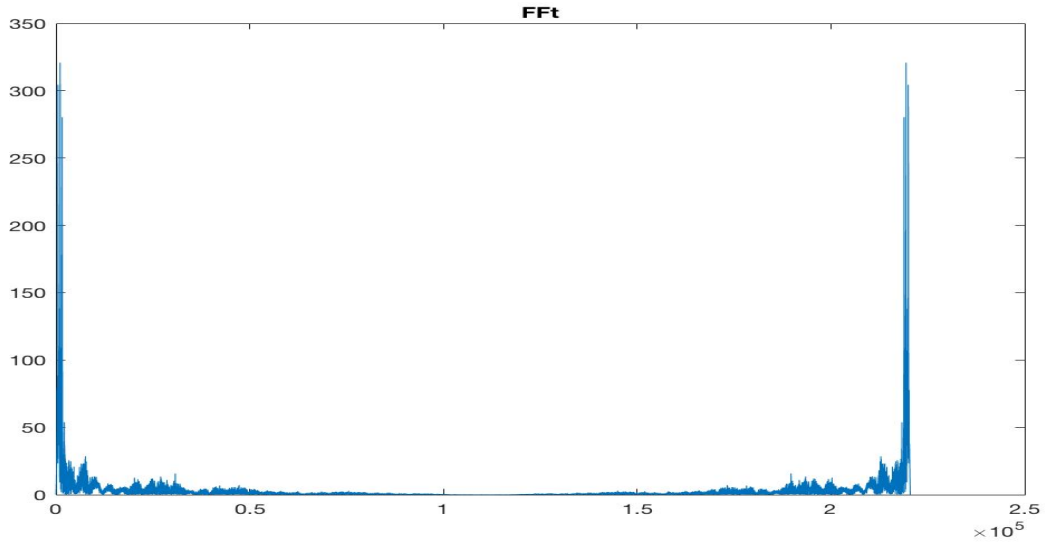

Figure 9

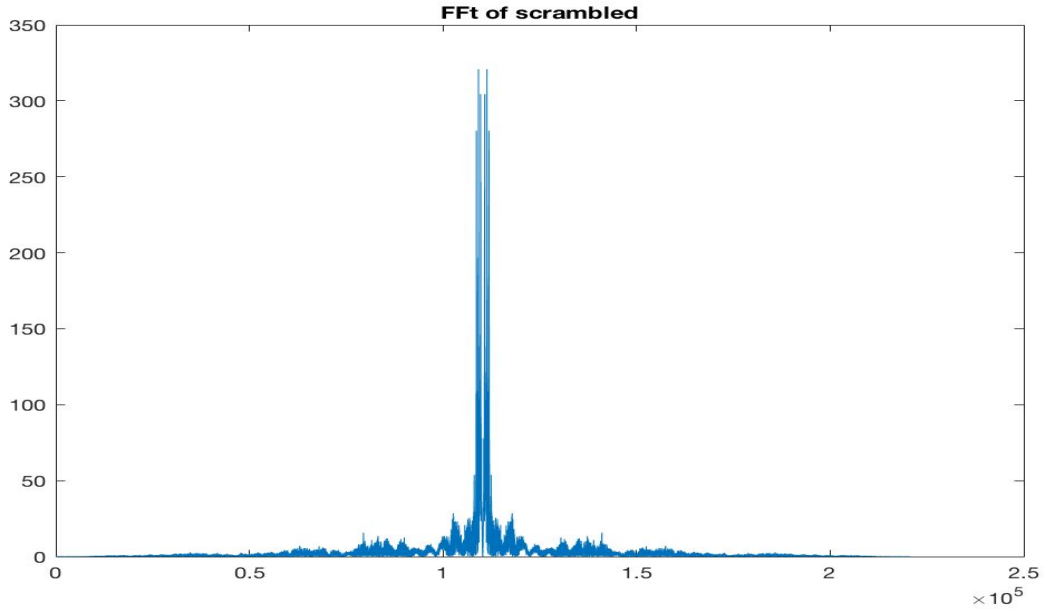

Figure 10

Discussion:

From completing this assignment, I have acquired a better understanding of how modulation works and how it can be implemented. I learned a simple method to scramble and descramble signal, but understand there are more robust speech scrambling methods. In our Matlab simulation, we implemented two different methods of applying the speech scrambling. In our first simulation, we used the FFTSHIFT function to shift the frequency of the signal and used it once again to shift back and get original signal. This method worked successfully but didn't require any understanding of the original method. For this reason, we decided to implement our derived method explicitly by multiplying the original discrete signal by the complex exponential. Our explicit implementation of the method worked successfully, as it scrambled the signal and made it very difficult to understand. Our first method though, seemed to have worked better as the scrambled signal sounds like a bunch of random noise while our implicit implementation scrambled signal somewhat resembles the original.