Doppler Radar Simulation

Christina Paolicelli

December 8, 2016

1 Overview

1.1 Goals

The goal of this project was to create a utility for those learning about radar and doppler shifts to begin to understand how the input parameters affect the waveforms. The focus of this project is on generating the raw doppler shifted, pulsed, radar signals received by the radar, however, demonstration of techniques to process received signals has also been developed.

1.2 Relation to Course Concepts

The primary digital signals processing concept implemented in this project is sampling. The generation, modification, and interpretation of the radar signals are all determined by using sampling of the overall signal. In addition, in order to maintain proper scaling without reaching MatLab's maximum array size downsampling was implemented.

Matlab's fast fourier transform (fft) and inverse fast fourier transform were implemented in the analysis of the received signal. A matched filter was developed in order to minimize the signal to noise ratio. The matched filter also uses the fast fourier transform and in addition the inverse fast fourier transform.

1.3 Radar Basics

Radars use radio waves to detect the range and velocity of objects or targets. The radars this project looks at are comprised of two components, a transmitter, and a receiver. The transmitter emits radio waves or radar signals and the receiver captures the radar signals reflected from the target. The reflections enable the radar to determine the range of the target due to the received time delay of the signal and the change in frequency of the signal due to the doppler effect of the reflection. The time delay can be determined by $t_D = \frac{2R}{c}$ where c is the speed of light in a free space. The factor of two comes from the signal having to travel from the radar to the target and back. The signal reflected off of the target experiences a frequency shift known as the doppler frequency shift. The doppler frequency can be determined by $f_d = -\frac{2V}{\lambda}$ where V is the target's velocity and λ is the wavelength of the radar signal which is $\lambda = \frac{c}{f}$ where f is the frequency of the carrier signal and c is once again the speed of light in free space. Using these basic principles the following simulation was developed.

2 Related Work

Relationships between this project and the subjects covered in the Digital Signals Processing course are discussed above in the Relation to Coursework section. In general, radar work is highly correlated with digital signals processing, particularly as more complex radars are developed which require more complex pre and post processing to gain higher resolution and better signal to noise ratios. Since the goal of this project was to develop a simple simulation for those beginning to study radars (such as the author) the more complex techniques are not discussed in this report, however, the foundation of digital signals processing from the course allowed for greater understanding and interpretation of the material for radar signals in general.

2.1 Literature

This project applies digital signals processing techniques to a concept outside the scope of this course, therefore it was necessary to gain an understanding of this concept. There were several resources which were of particular use in this process. In particular, Radar Signals by Nadav Lebanon and Eli Mozeson [1] provided an introductory understanding of how radar signals are handled and were invaluable to the development of this project.

Another introductory resource to radars in general which was particularly useful was the coffee can radar project from MIT open courseware [2] as well as Prof. Gregory Charvat's article on Hackaday.io [3].

3 Data Collection

Attempts to find simple radar data during the introductory phase of the project proved mostly unsuccessful. It was found that there are publicly available datasets for SAR data, but this is more complex and two-dimensional. The only available data that was found which was within the parameters that were desired to be studied for this project was the sample dataset provided by the MIT open courseware course, however, it was felt that the single dataset would not be sufficient. It was this lack of data which drove this project to develop a simulation for simplistic radar analysis for those of us beginning to study this field.

The simulation has three variables which can be varied by the user. The range of the target, the velocity of the target and the signal to noise ratio (SNR). A data set was generated where each of these three parameters was independently varied. While one variable was being varied the other two were set to a constant value. The range and the velocity were set to approximately the middle of the range, 23 and 95 respectively. The constant SNR was set to 30, although not the middle of the range this value mirrors the common values seen in literature.

The data collection for each set of values of includes the output from the matched filter in both the time and frequency domains as well as the autocorrelation, correlation, and the raw received signal. The folder containing the data set has been submitted, but here are a few notes on the notation. Each folder is labeled with three numbers in the order Range Velocity SNR, which is also the order they are requested from the user. The code can generate additional plots but the code has been commented out in order to decrease run time.

4 Technical Approach

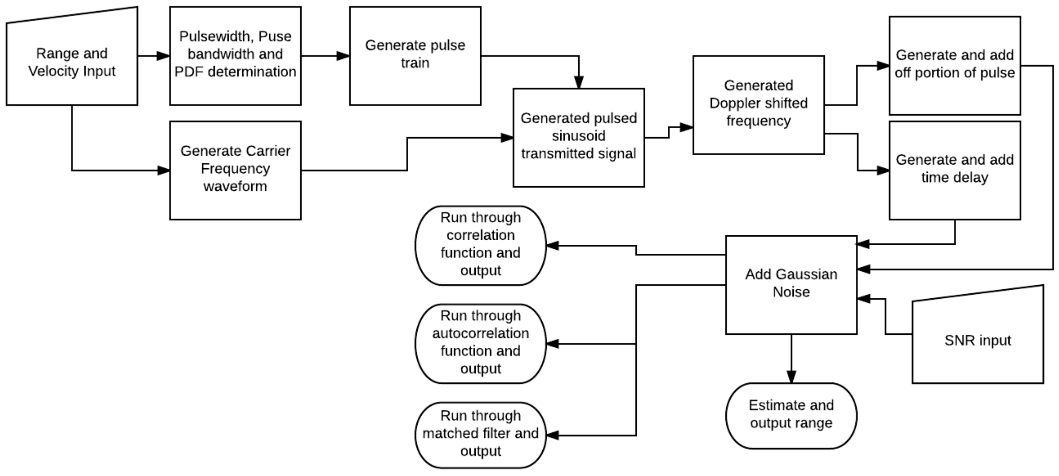
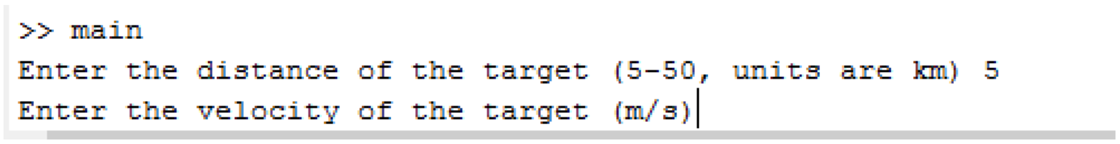

Range and Velocity Input
Pulsewidth, Puse bandwidth and PDF determination
Generate pulse train
Generate Carrier Frequency waveform
Generated pulsed sinusoid transmitted signal
Generated Doppler shifted frequency
Generate and add off portion of pulse
Generate and add time delay
Add Gaussian Noise
SNR input
Run through correlation function and output
Run through autocorrelation function and output
Run through matched filter and output
Estimate and output range

Figure 1: Overview Block Diagram

4.1 Initilizations

The idea behind this simulation is that the user would be able to input the desired range and velocity of the radar target, however, the allowable range is a design parameter of a radar and thus these values must be in specific ranges. The radar simulation was implemented as a pulsed doppler radar, this means that the transmitted signal is the carrier sinusoid pulsed on and off at a specified frequency known as the pulse repetition frequency (PRF). The majority of pulsed doppler radars operate at medium pulse repetition frequency which is from 3kHz to 30kHz, this correlates with a range of 5km to 50km and velocities 17.3m/s to 173m/s (for the implementation's carrier frequency). The user input is checked against these boundaries and an error is generated if the value is outside of them.



```
>> main
Enter the distance of the target (5-50, units are km) 5
Enter the velocity of the target (m/s)
```

Figure 2: Command line user input

Once the range and velocity are specified optimized radar parameter can be determined. These parameters include the pulse bandwidth, the pulse width, and the PRF. The pulse bandwidth is determined using the equation, $pulsebandwidth = \frac{c}{2*R}$. From the pulse bandwidth, the pulse width can be determined by taking the bandwidth's inverse. In order to determine an optimized PDF the carrier frequency must be decided. In this simulation the carrier frequency is hard coded at 13GHz, it would be easy to allow the simulation user to change this variable, however, if it drops too low there are two potential issues. First if $pulsebandwidth < \frac{1}{10}carrierfrequency$ the signal may contain modulation due to the target moving, in addition the point of this simulation is to help people learn about radars if the carrier frequency becomes too low there may not be as many oscillations within a pulse, as I am a beginner with radar myself, it is hard to see what is happening if there are very few oscillations within a pulse, therfor the final value was decided to ensure visual clarity. In order for the desired range and velocity to be accurately detected the PRF must be set such that the range and velocity are less than the respective maximum unambiguous measurement. This is accomplished by calculating minimum bounds for each of these inputs using the following equations: $bound1 = \frac{c}{2R}$ and $bound2 = \frac{|V|4f_c}{c}$. To set the final PRF the maximum of these was chosen and the difference of the bounds was added; for the given inputs this should be in range but a bound check was implemented as well.

4.2 Carrier Signal

The first step in the simulation is the generation of the carrier signal. Although the plot gives no useful information it was plotted for the purpose of the simulation to ensure the sampling resolution was high enough for decent visual reconstruction. The signal of the carrier frequency was calculated as $f_c = sin(\omega_c * t)$ where t is the time vector sampled every $\frac{1}{20f}$ from 0 to 2E-9. To satisfy the Nyquist rate it would have been sufficient to sample every $\frac{1}{4f\pi}$ but by increasing the intervals it made it much easier to visualize on the plot, which although not important for this step as generating the carrier signal is pretty straight forward, is important in the more specific steps of this simulation and therefore it was important to ensure there was sufficient resolution.

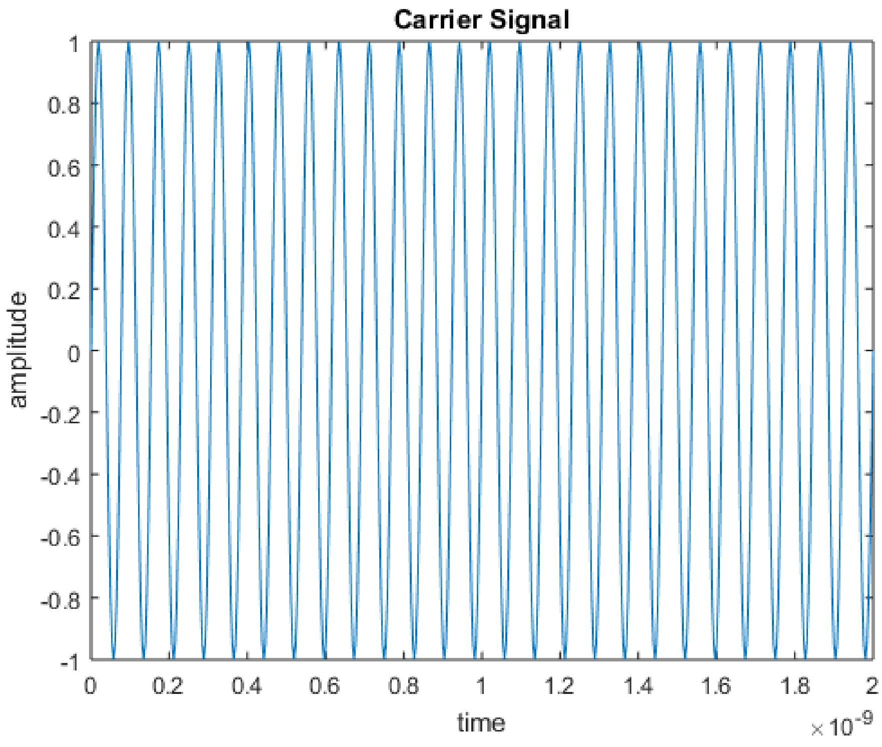


Figure 3: Carrier Signal Plot

4.3 Pulses

The next signal that needed to be generated is the pulse train corresponding to the pulse repetition frequency calculated earlier. The PRF is much lower than the carrier frequency (at least by a factor of 10 by necessity), therefore a separate time vector was calculated. This time vector is sampled at 1/f for 5E6 samples. On and off durations of the pulse were also determined. The duration the pulse is ”on” was set equal to the pulse width multiplied by the period of the pulse (1/PRF). The ”off” duration was set equal to 1 minus the ”on” duration. The pulse train was then calculated using the MatLab code; pulse = double(mod(ts,sum(Tr))<Tr(1));. This was plotted for visual understanding.

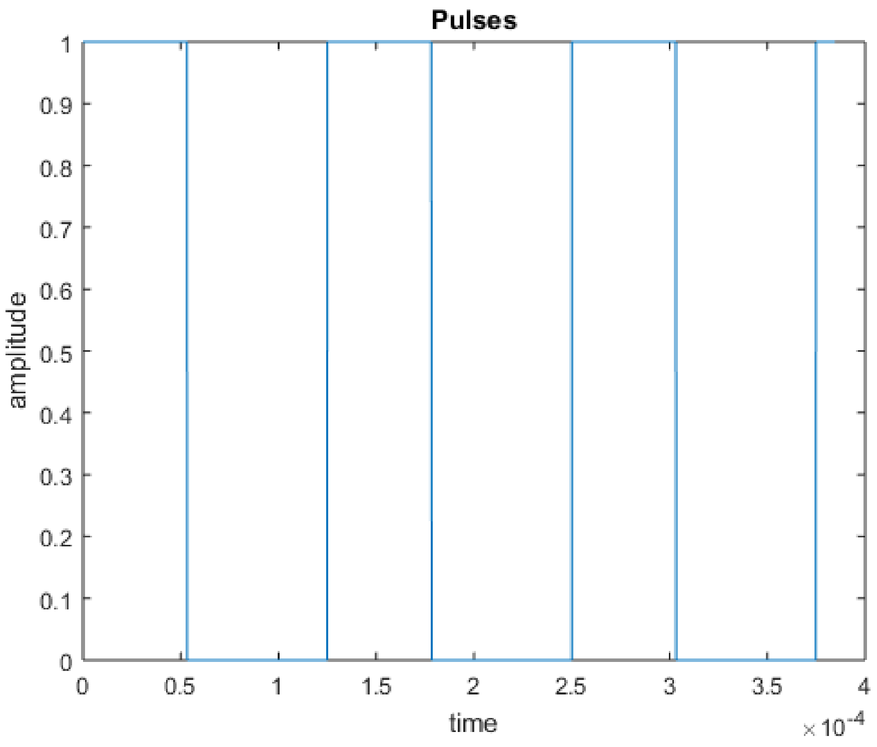


Figure 4: Pulses with PRF of 8kHz

4.4 Transmitted Signal

In order to generate the pulsed version of the carrier frequency signal the two previous signals are combined, therefore the signals need to be re-scaled in order for this to work. The carrier frequency signal is interpolated (upshifted) by a factor of 64 and then 7 copies of the signal are appended to each other. The pulse train was down-sampled by a factor of 5 A new time vector is created with the samples occurring every $\frac{1}{10f}$ from 0 to 7E-5. These signals were plotted on the same graph to show the generation of the pulse, note the carrier frequency signal is still much shorter than the pulse train.

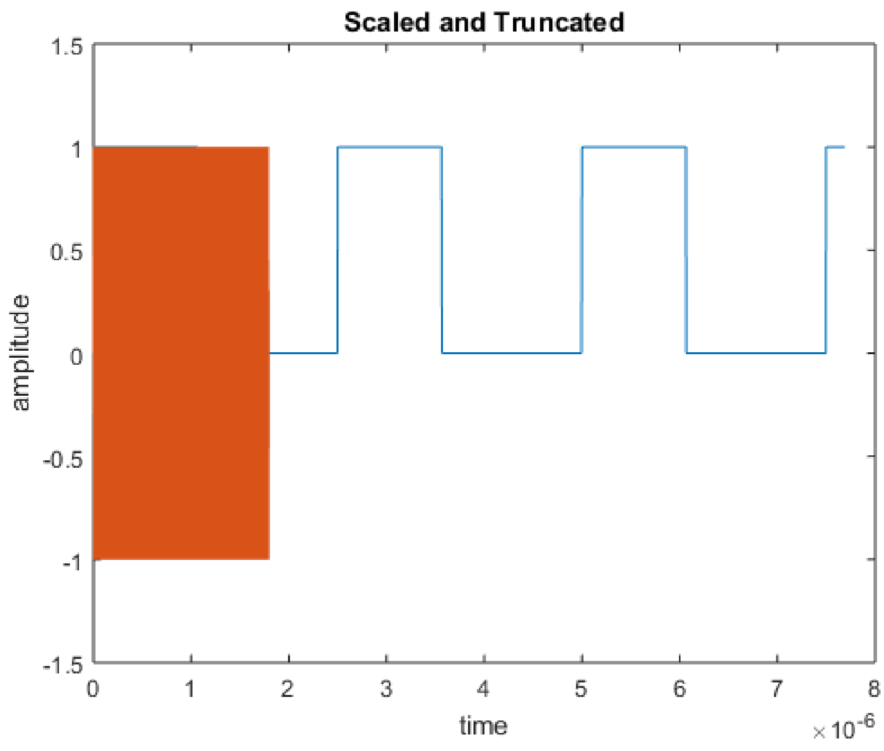


Figure 5: Overlay of pulsetrain and carrier signal

Once the time scales are shifted and the signals are truncated to be the same length the transmitted signal is generated by multiplying the pulse train with the carrier frequency signal.

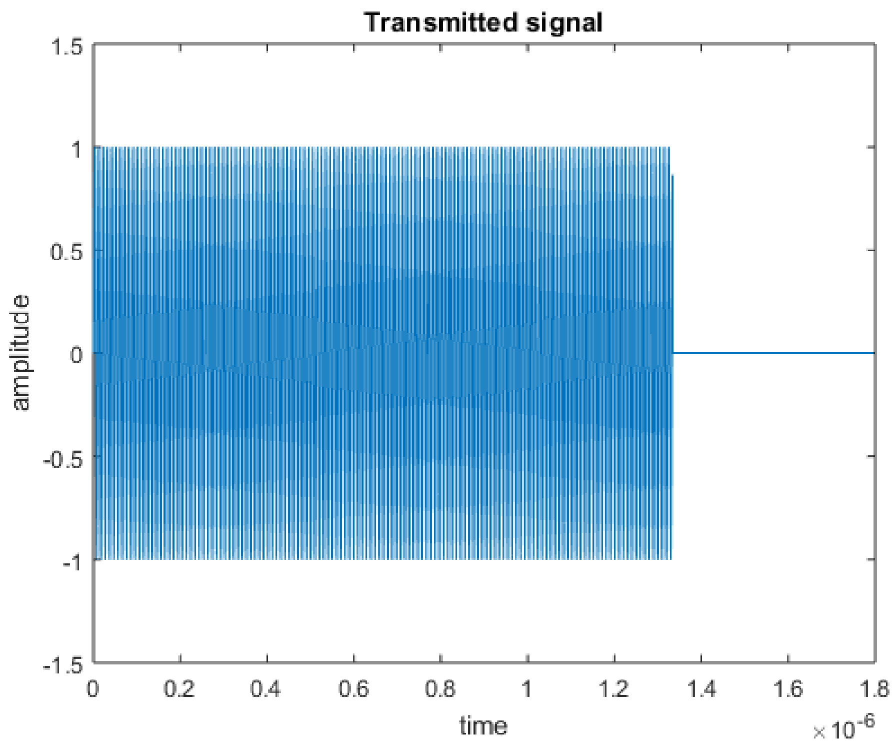


Figure 6: Single Pulse of transmitted signal

4.5 Received Signal

The received signal is comprised of two parts, the pulsed sinusoid which undergoes a doppler shift and the time delay caused by the time necessary for the signal to reach the target and be reflected back to the receiver.

4.5.1 Dopppler Shift

The doppler shift frequency, known as the doppler frequency was calculated using the formula; $f_d = \frac{-2V}{\lambda}$, where V is the target's velocity and lambda is the wavelength. A new time vector is created which is sampled at $\frac{1}{3(f+f_d)}$ from zero to the "on" part of the pulse width. To meet the Nyquist criterion the sampling would only have to be at $\frac{1}{2(f+f_d)}$ but 3 produces a better visual image, although the zoomed out image is indistinguishable when you zoom in 3 has better resolution. The doppler shifted pulse itself is generated by creating a signal equal to $\sin((\omega_c + \omega_d) * timevector)$

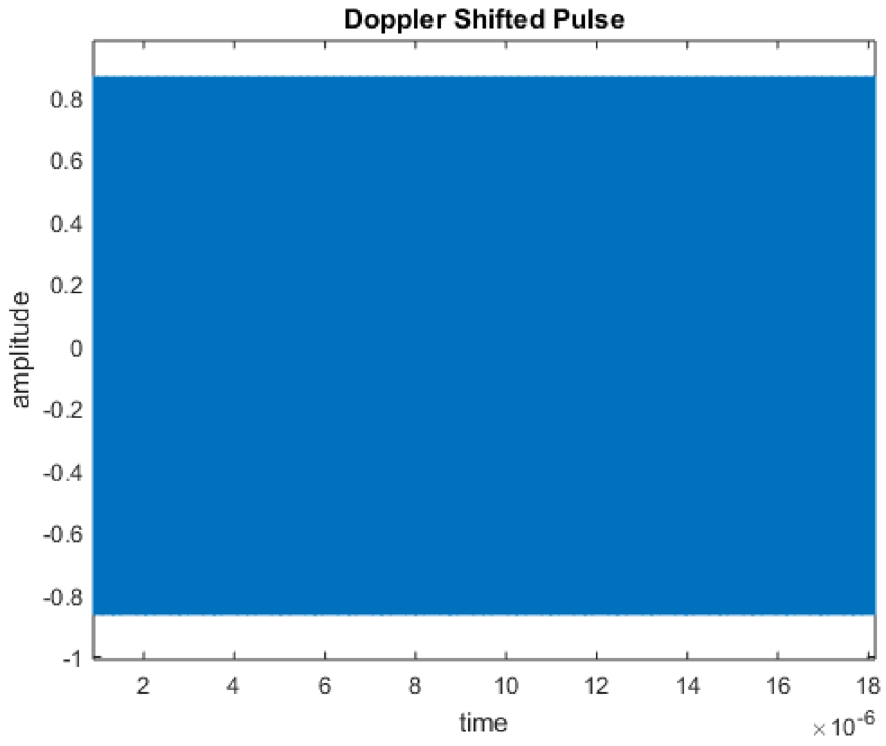


Figure 7: Full Doppler Shifted Signal

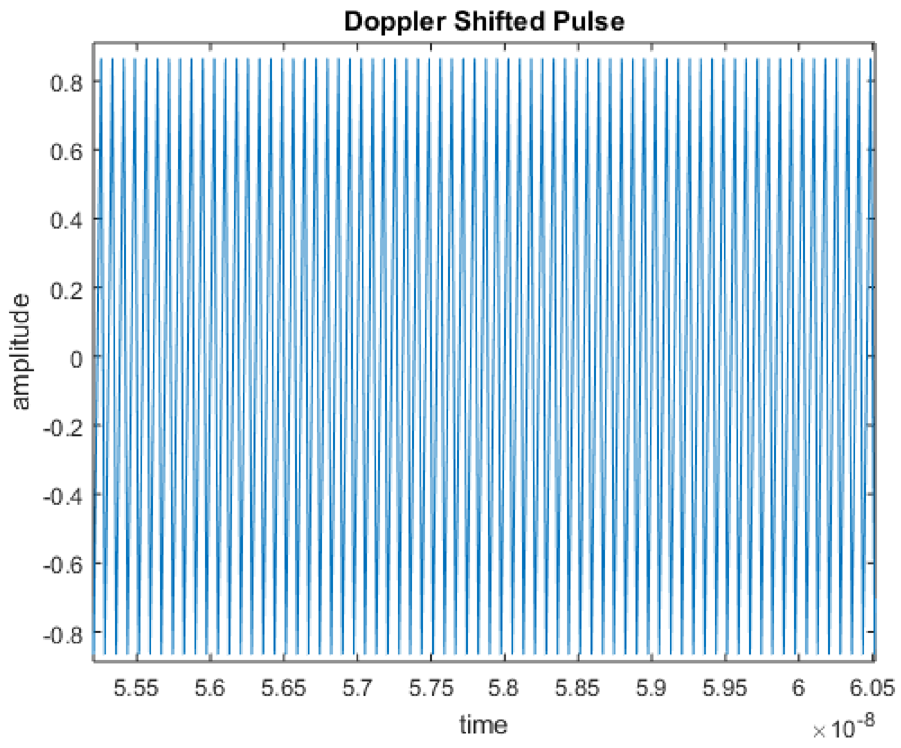


Figure 8: Zoomed in doppler shifted signal

4.5.2 "off" portion

The transmitted signal had both an "on" and an "off" portion. The "on" portion which had frequency fc has been generated with it's doppler shift above, the "off" portion which has a constant zero amplitude needs to be added back to the now doppler shifted signal. Based on the number of samples which correlate to the "on" portion of the signal, the number of additional samples needed for the "off" section of the signal is calculated. A row vector of zeros of this length is then added to the original doppler shifted signal and the time vector is extended in order for MatLab to plot.

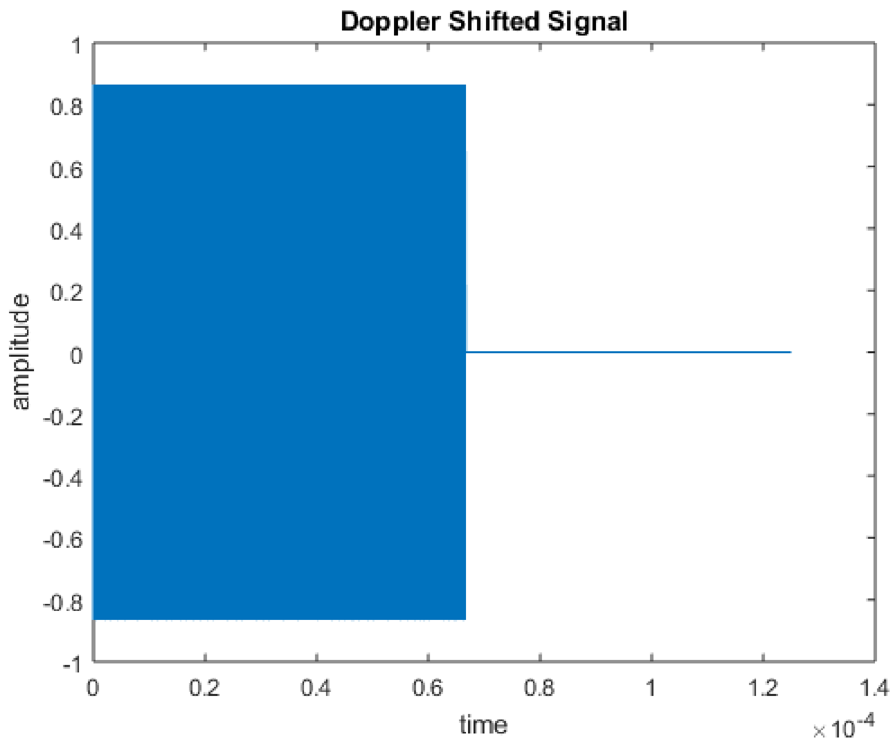


Figure 9: Doppler shifted signal with "off" portion

4.5.3 Time Delay

Despite traveling very fast electromagnetic radiation does not travel instantaneously and therefore the time delay from when the transmitter sends the radar signal to when the receiver receives it is not-negligible. The time delay can be found by the formula; $t_d = \frac{2R}{c}$, where R is the range, c is the speed of light in free space and the factor of 2 comes from the need to travel to the target and back. Once the time delay is known, the number of steps which need to be added to the time vector is determined as are the number of zeros needed to precede the received doppler shifted signal.

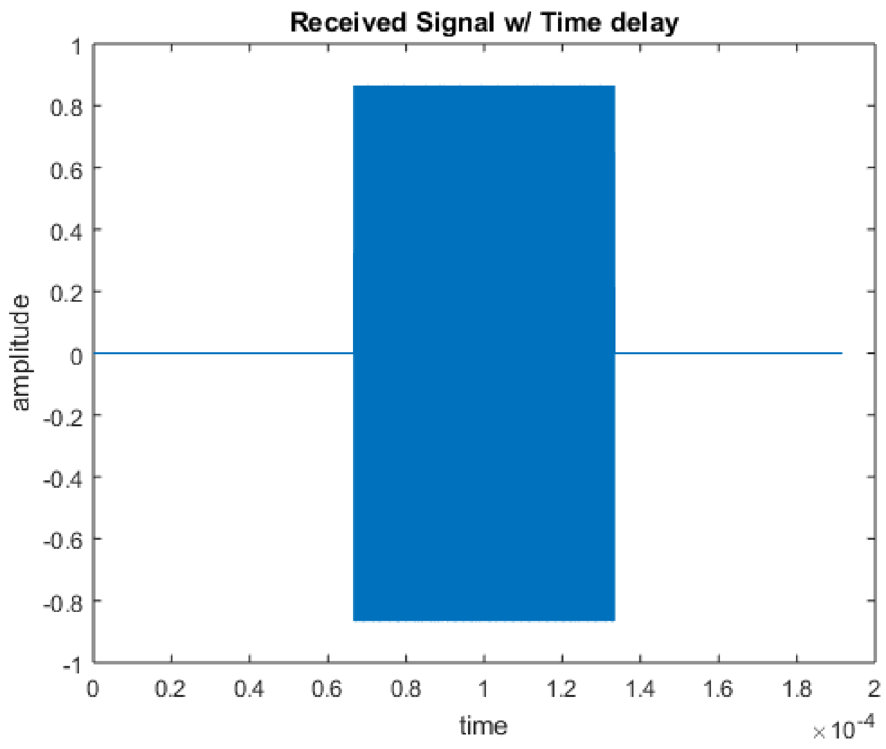


Figure 10: Noiseless Received Signal

4.5.4 Gaussian Noise

The final component of the received signal is the noise or clutter. There are two main potential sources of error; thermal background noise and reflections off other objects, such as the ground. There are models which describe they types of clutter which can occur, to maintain simplicity in this model it was decided to add gaussian white noise as the noise, in the literature this is commonly used to descibe noise and therefore was not considered unrealistc. The user is asked to enter the desired signal to noise ratio(SNR), it was decided to set this as a user input because of the deliberate variation of this factor emphasize the importance of noise, and noise reduction, in signals processing which is important to the study of radar systems. The Matlab function a awgn() was used to generate the noise at the specified SNR, awgn() was also set such that it interprets the SNR as a ratio rather than in dB quantities. This was done because in the literature on radar it was noted that SNR was usually discussed in terms of ratios, not dBs. The received signal with gaussian noise was then plotted.

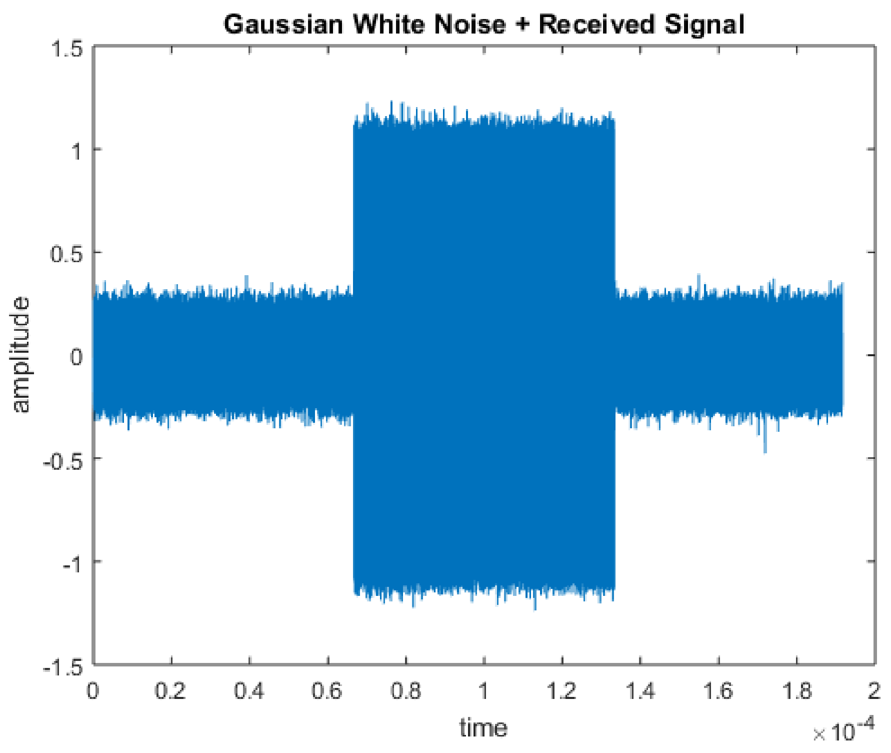


Figure 11: Received Signal with Gaussian Noise

4.6 Range Estimation

Now that the receiver has received the radar signal data extraction and analysis can begin. The first task was to find the range of the target, which was done by keeping track of the time delay - that is the radar knows when it sent out the pulse and it knows when it received the reflected pulse and therefore knows the time delay. The implementation of this in the simulation found the first data point of the signal which wasn't 0 + noise and recorded it's time as the time delay. The range is then estimated using the equation; $Range = \frac{c}{2} * timedelay$. The range estimation was then displayed in the command window. For further analysis, the time delay was then discarded from the signal.

4.7 Correlation

The correlation between two functions is a measure of their similarity. By plotting the correlation between the transmitted signal and the received signal it is possible to see where the transmitted signal occurs (even thought it is now at a slightly diffrent frequency). Note there is noise on the boundaries of the signal, this is possible to eliminate through additional filtering.

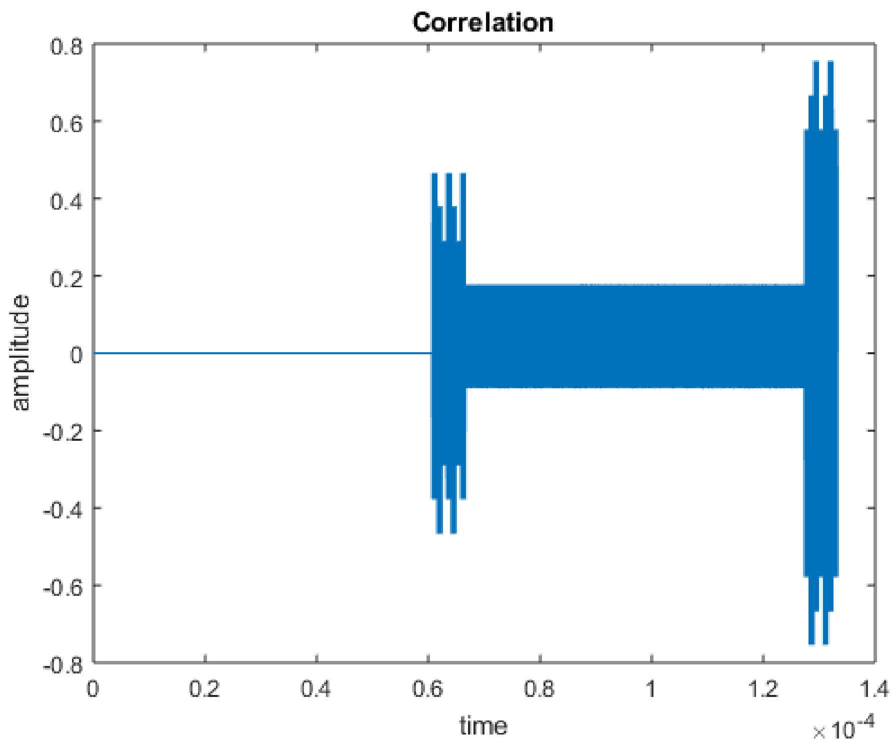


Figure 12: Correlation

4.8 Autocorrelation

Autocorrelation is the correlation of a function with itself. Autocorrelation is useful for finding repeated patterns in a signal, for example, the presence of a periodic signal. For the received signal the autocorrelation results in a triangular shaped function with the peak concentrated where the signal is located and tailing off towards the part of the signal which is simply noise. For this specific analysis, I didn't find the autocorrelation function to be of much use, but as it is common for this type of signal analysis I wanted to present it to the user.

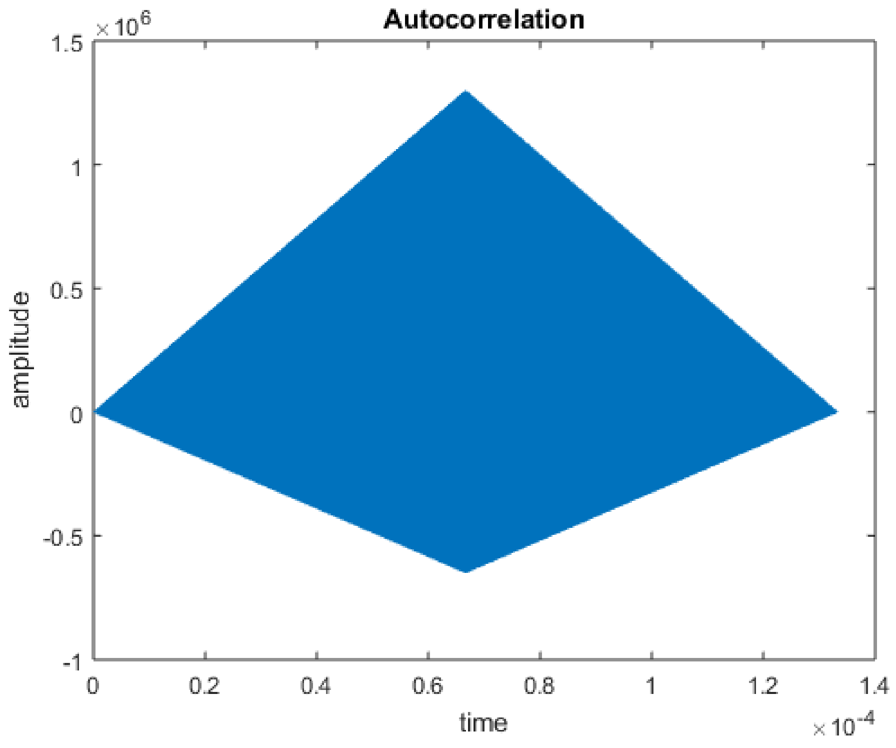


Figure 13: Autocorrelation

4.9 Matched Filtering

The matched filter is the optimal filter for maximizing the signal to noise ratio in the presence of additive random noise like we have in this simulation or is present in radar signals in the real world.

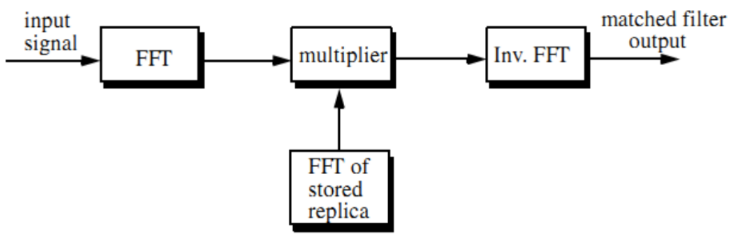


Figure 14: Block Diagram of Matched Filter [6]

The effective implementation of the matched filter is the convolution of the received signal with the transmitted signal. This was done by calculating the fft of both of these signals and multiplying them then calculating the ifft to return the signal to the time domain. The matched filter output can also be analyzed in the frequency domain, this shows that the greatest weight of the filter is applied to components where noise is relatively low and there is a high signal presence.

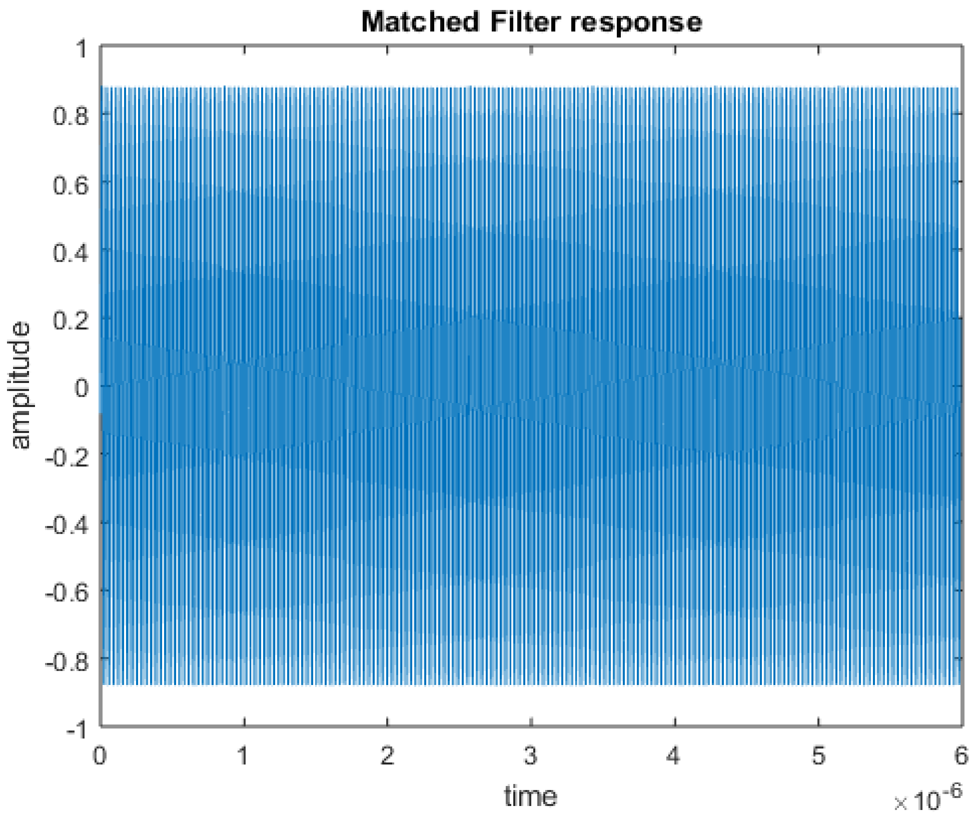


Figure 15: Time domain Matched Filter response

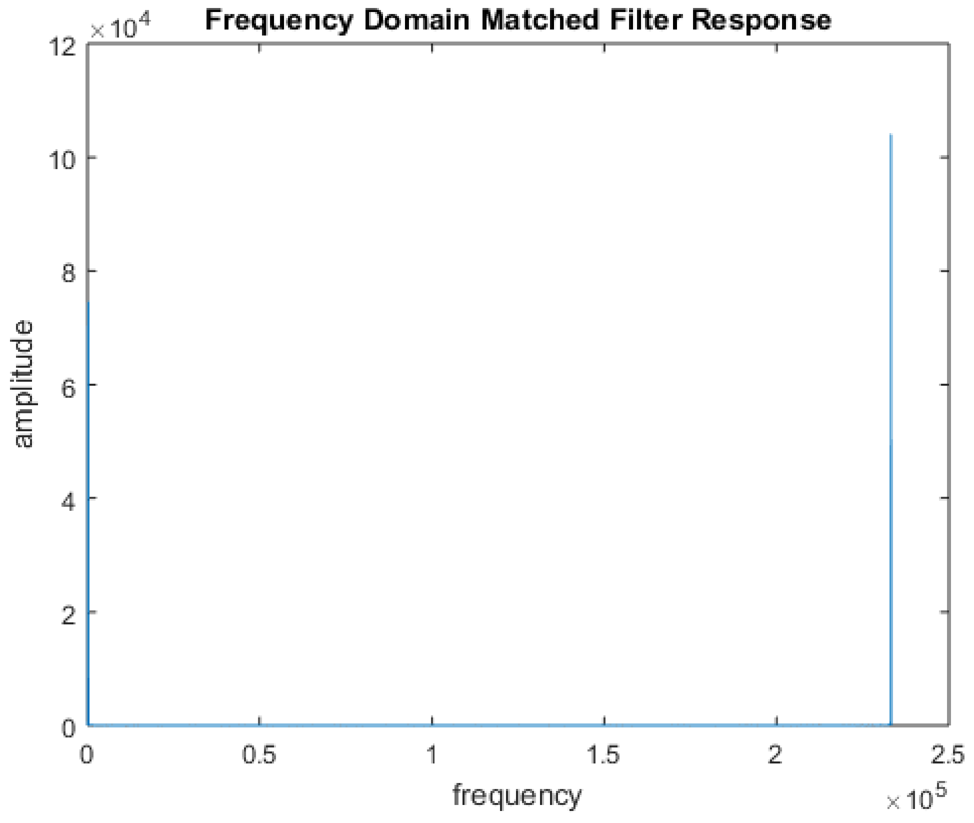


Figure 16: Frequency domain Matched Filter response

5 Intermediate Results

Many of the results shown in the Technical Discussion section are intermediate steps to the final analysis so that the user can have an idea of what is happening behind the scenes. The other intermediate steps and decisions are discussed below.

5.1 FFT for Validation

The mathematical determination for the desired frequency of the received signal is easy to calculate via; $f_r = f_c + f_d$. To check that the signal constructed from the carrier and doppler shift represented this the fft of the signal was plotted and the frequencies of the maximum calculated. These were checked to make sure they aligned with the desired receiver frequency and its harmonics.

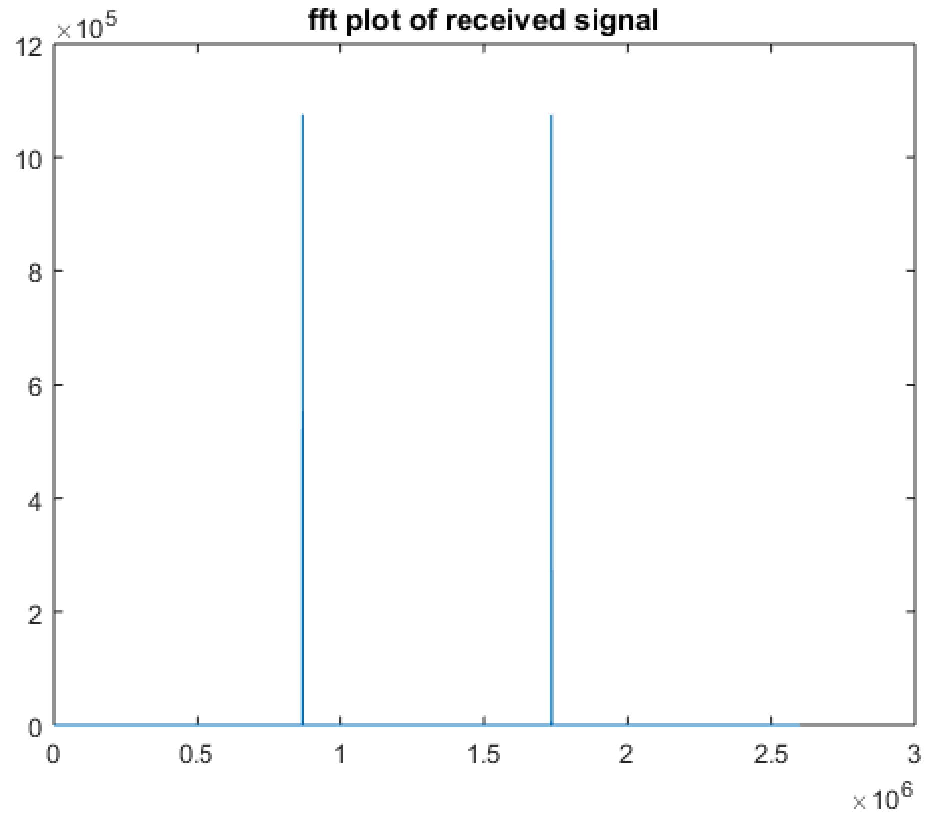


Figure 17: Received Signal FFT

5.2 Threshold Graph

An important concept in radar signal detection is the determination of a threshold value, above which the response is to be considered valid data. In order to determine the levels, the added noise reached the received radar signal was set equal to a constant of 1 (0 was tried but this resulted in no noise and therefore wasn't useful). The maximum deviation from the constant was recorded. The plot below shows one such plot of noise on a constant signal of one.

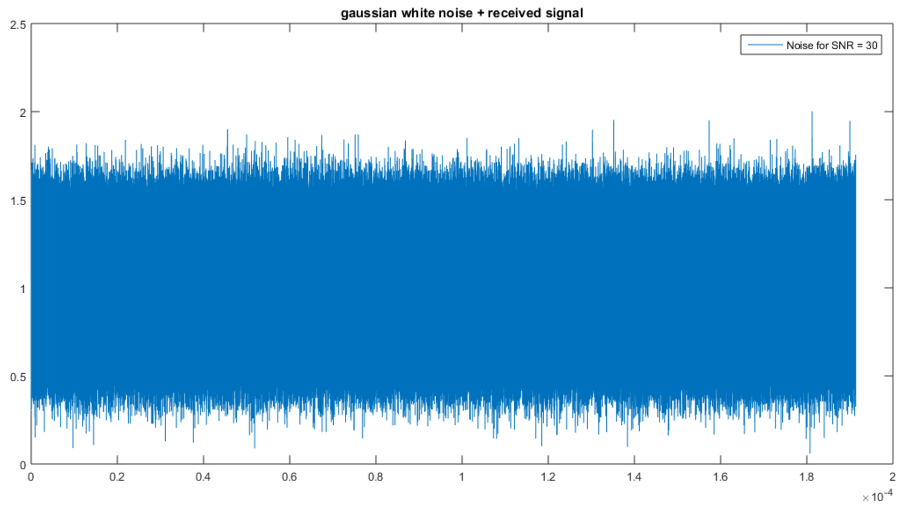


Figure 18: Constant signal with Gaussian Noise; SNR = 30

5.3 Strengths

The strengths of this simulation are its simplicity and its stepwise approach. Because this is approaching radar at its most fundamental level this simulation is able to show each step which I think for beginners learning about radar is very important. The results are particularly clean for tagets in the middle of the range and veloctiyy bands, as these begin to move to the extrems issues were noticed with sufficently sized time windowing.

5.4 Weaknesses

The weakness of this simulation is it's limited capacity to a very small range of what radars are capable of. For example, this mimics the actions of early radars such as those used during world war one. Additional weaknesses occurred at the edge cases of the range and signal to noise ratio. These failure cases are discussed below in the final results section.

6 Final Results

This simulation works particularly well in the middle of its range, as the numbers start to move towards the fringes the data is less clear. A good sample of a good performing run of this simulation can be seen in the Technical overview section of this report. In this example, the data appears mostly centered on the plotted graphs, as the range and/or velocity increase the time delay increases in several cases this pushed the second half of the returned signal off the visible plot.

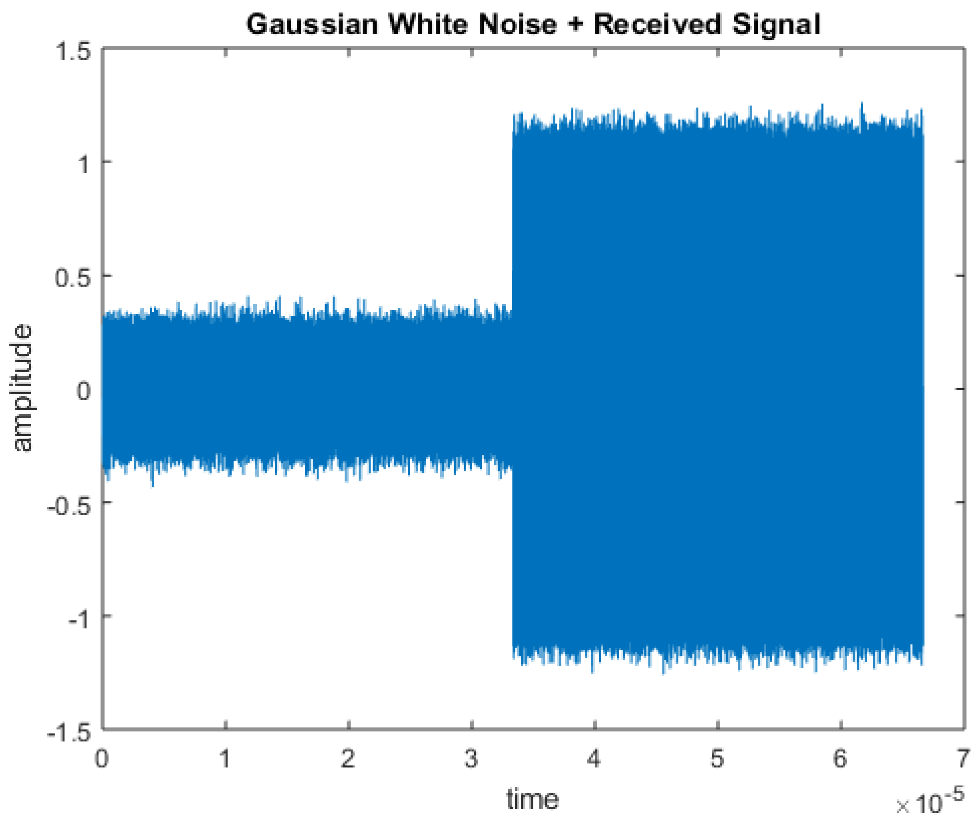


Figure 19: Received Signal, 2nd half lost

There were also several cases where the correlation seems to be more compressed than it should be, or that perhaps the time scale was not properly adjusted to the data set. The figure below shows one of these cases.

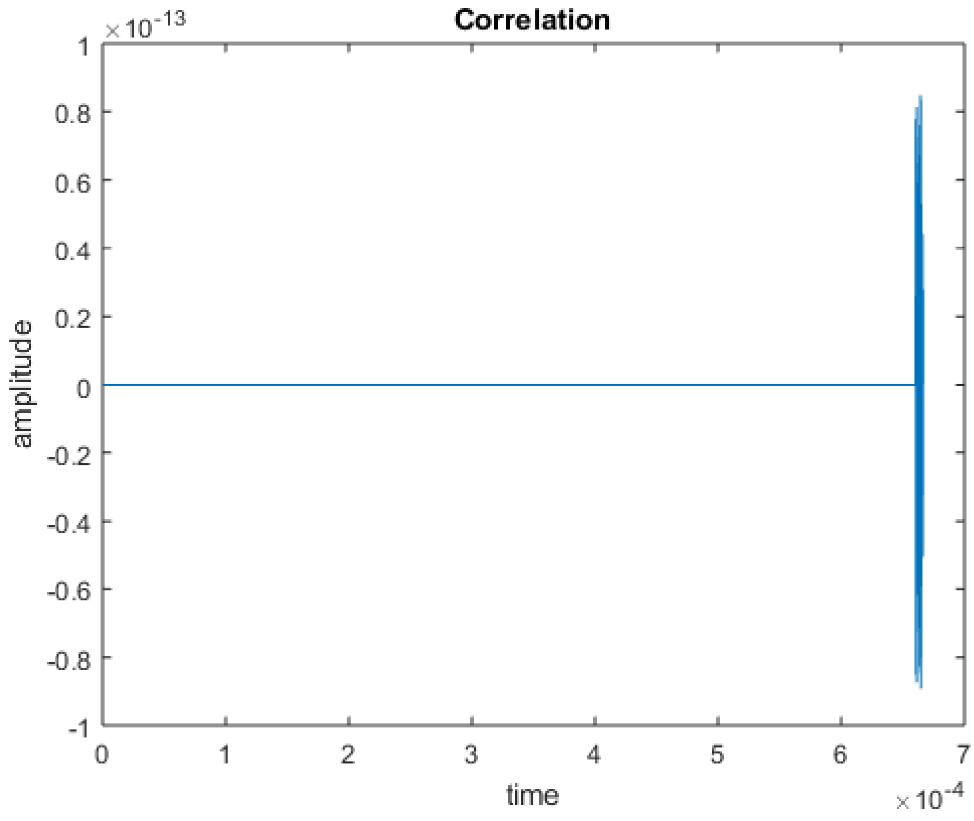


Figure 20: highly compressed correlated response

The last unexpected result that was observed was a shifted and cut off autocorrelation function, again this can be attributed to a non-dynamic time scale window which for exterior range and velocity cases caused the autocorrelation function to shift outside the time window.

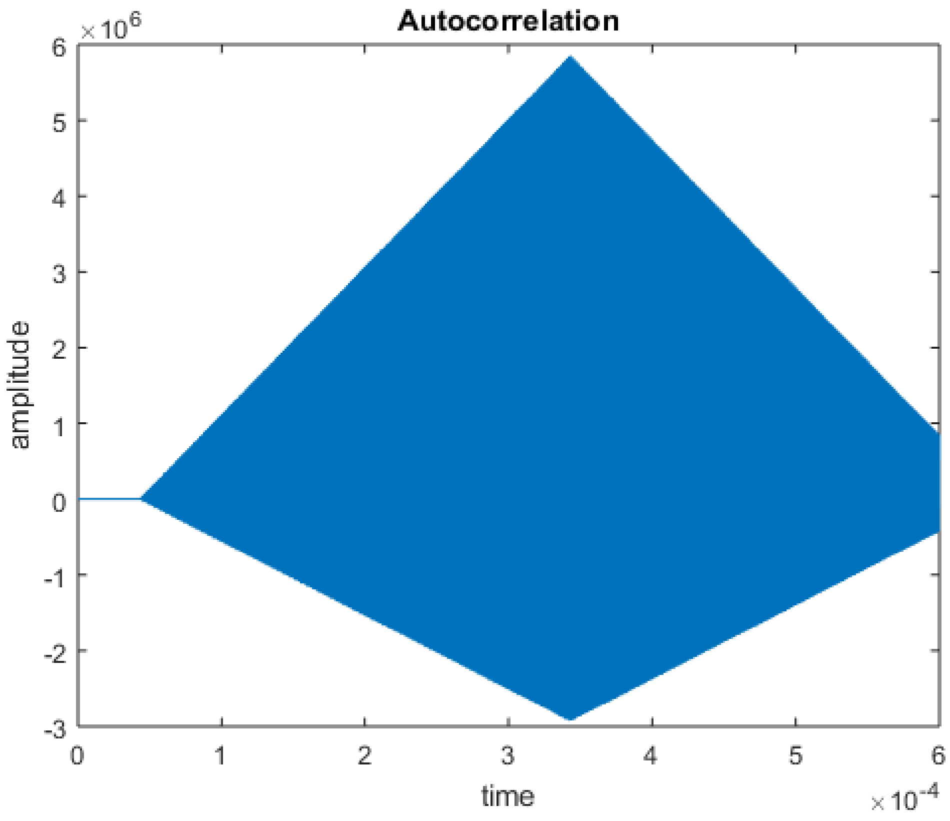


Figure 21: Shifted and cutoff autocorrelation function

A low SNR also caused several issues. The trials with an SNR of 1 and 10 had incorrect ranges, for both the range should have been 23km however with the SNR of 10 the estimated range is 7.4423m and with the SNR of 1 the range returned 1.

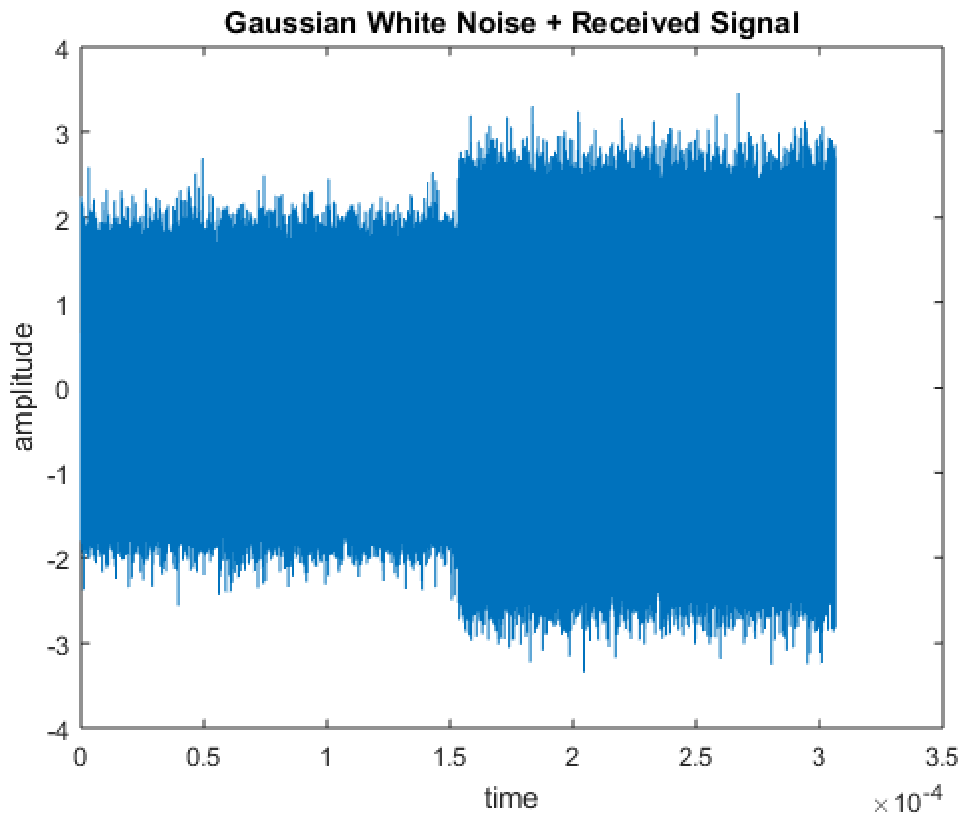


Figure 22: Received signal with SNR of 1

7 Disscussion and Future Work

As with any simulation there are many places to go. This simulation was intentionally simple as I am just learning about radars, as this is a field I am interested in as I learn more this simulation can be improved and expanded, my lack of knowledge in the field I felt throughout the process was one of the limited factors but throughout this process I've learned alot and made the best simulation I possibly could. Given additional time the time scaling issues discovered during testing could have been fixed. In addition the simulation could have been more modular to allow for more user inputs and interaction, specifically through the design of a graphical user interface, I know MatLab has the capability for this but I found it confusing and did not feel that was the best way to spend the limited time.

Now that I've finished the project I would have considered using different software. As great as MatLab is I don't think it was the right platform. If I had to redo this project I would have considered using Simulink or Labview. I've gained a lot of experience in MatLab over this semester and I don't have that level of experience in the other software packages and therefore I would have needed additional time to learn the software as well as the work itself.

Some extensions of this project would be to include other types of radar processing. There are many different routes this could take but one of the most closely correlated with the course concepts would be additional pre and post processing to reduce noise or implement additional coding methods.

8 Refrences

[1]N. Levanon and E. Mozeson, Radar signals, 1st ed. Hoboken: Wiley-Interscience.

[2]Gregory Charvat, Jonathan Williams, Alan Fenn, Steve Kogon, and Jeffrey Herd. RES.LL-003 Build a Small Radar System Capable of Sensing Range, Doppler, and Synthetic Aperture Radar Imaging. January IAP 2011. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

[3]G. Charvat, "Guest Post: Try Radar for Your Next Project", Hackaday, 2016. [Online]. Available: http://hackaday.com/2014/02/24/guest-post-try-radar-for-your-next-project/. [Accessed: 06- Dec- 2016].

[4]"Signal Correlation and Detection II", ECSE 206. [Online]. Available: http://www.eecs.umich.edu/cours
[Accessed: 07- Dec- 2016].

[5]"Correlation and Auto-Correlation". [Online]. Available: *http : //www.appstate.edu/ grayro/comphys/*
[Accessed: 07- Dec- 2016].

[6]T. Pardhu, A. Sree and K. Tanuja, "Design of Matched Filters for Radar Applications", Electrical and Electronics Engineering: An International Journal (ELELIJ), vol. 3, no. 4, 2014.

[7]R. Nitzberg, Radar signal processing and adaptive systems, 1st ed. Boston: Artech House, 1999.