Lab 4

1. Simple IF Data Analysis

For this lab, IF data was used to generate a GPS position solution. The IF data used came from a IFEN SX3 with a 20000000 hertz sampling frequency and an intermediate frequency of 5000445.89 hertz. To analyze this data, 1 second of the IF data was pulled from the data file. The raw data values for the IF data between the times of 0.1 to 1 second are plotted in *Figure 1*.

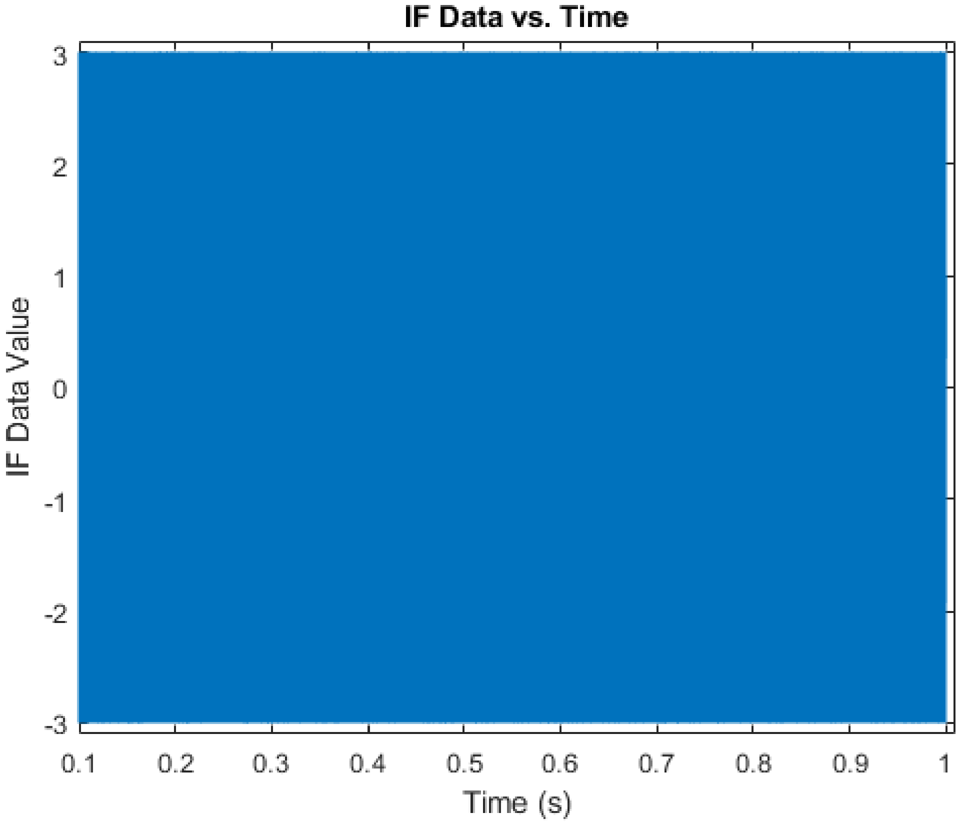


Figure 1: IF Data Over Time.

From *Figure 1*, it can be seen that the IF data values vary between -3 and 3 across the entire time interval. Because the IF data is sampled at 20000000 hertz, however, it is hard to see any trends in the data at such a large time period. Therefore, *Figure 2* shows a zoomed in plot of the IF data at a smaller 0.1 millimeter time period.

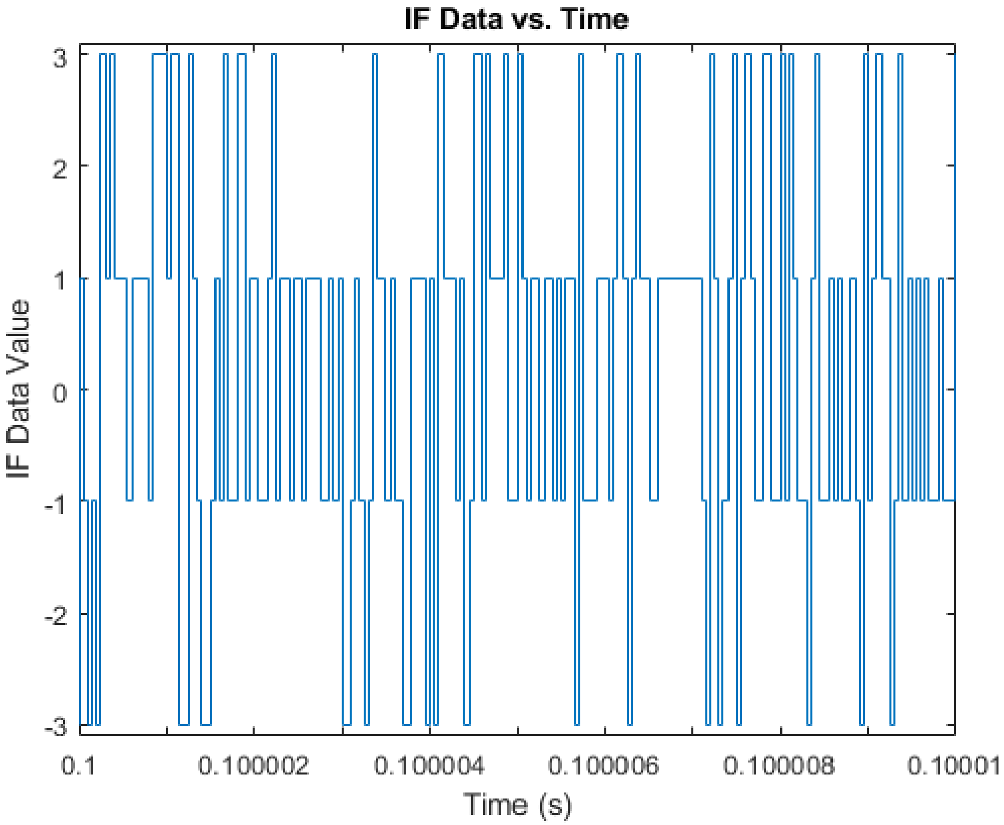


Figure 2: IF Data Over Time Zoomed In.

Looking at the IF data at a smaller time interval shows a more legible depiction of the IF data. For most of the time period the IF data will vary between the values of -1 and 1. However, occasionally the IF data values will spike to either -3 or 3. Every single value of the IF data is either -3,-1,1 or 3 for the entire 1 second time period. This is better shown in the histogram of the IF data shown in *Figure 3*.

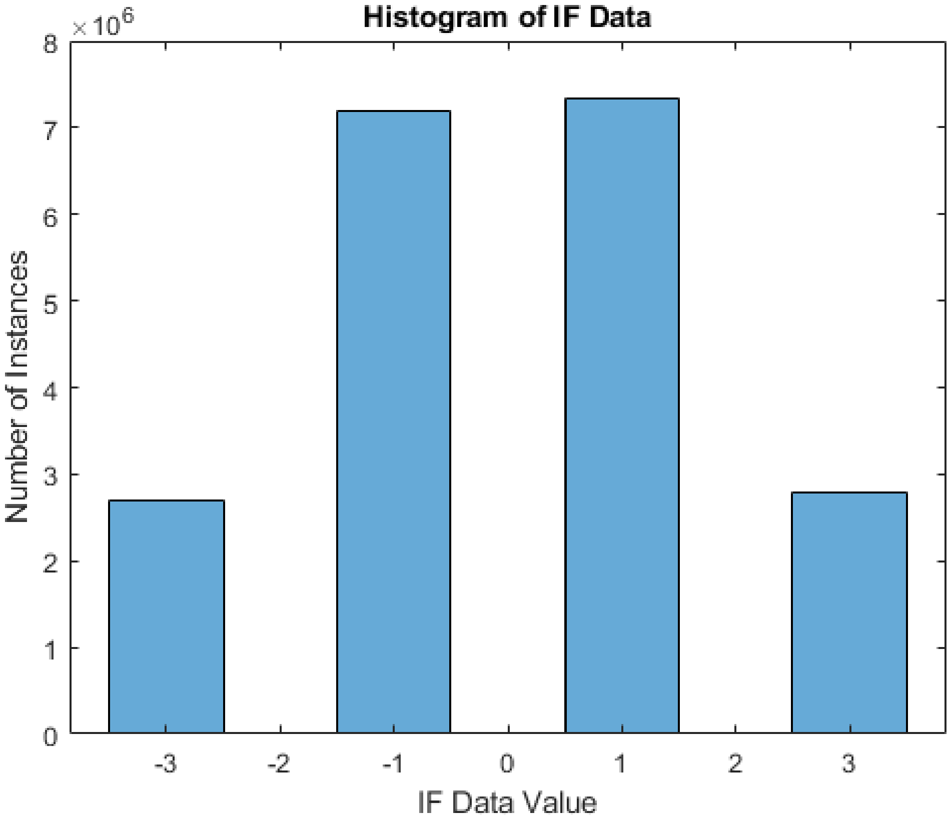


Figure 3: IF Data Histogram.

From the histogram it can be seen that the most common values in the 1 second IF data set are -1 and 1. There are slightly more positive values than negative. This relationship is also true for the 3 values. Generally, for most instances of there being a positive value in the IF data a negative value also exists. However, there is no discernable pattern to these positive and negative values. Additionally, a power spectral analysis was performed on the IF data. *Figure 4* shows the power spectral density of the 1 second of IF data.

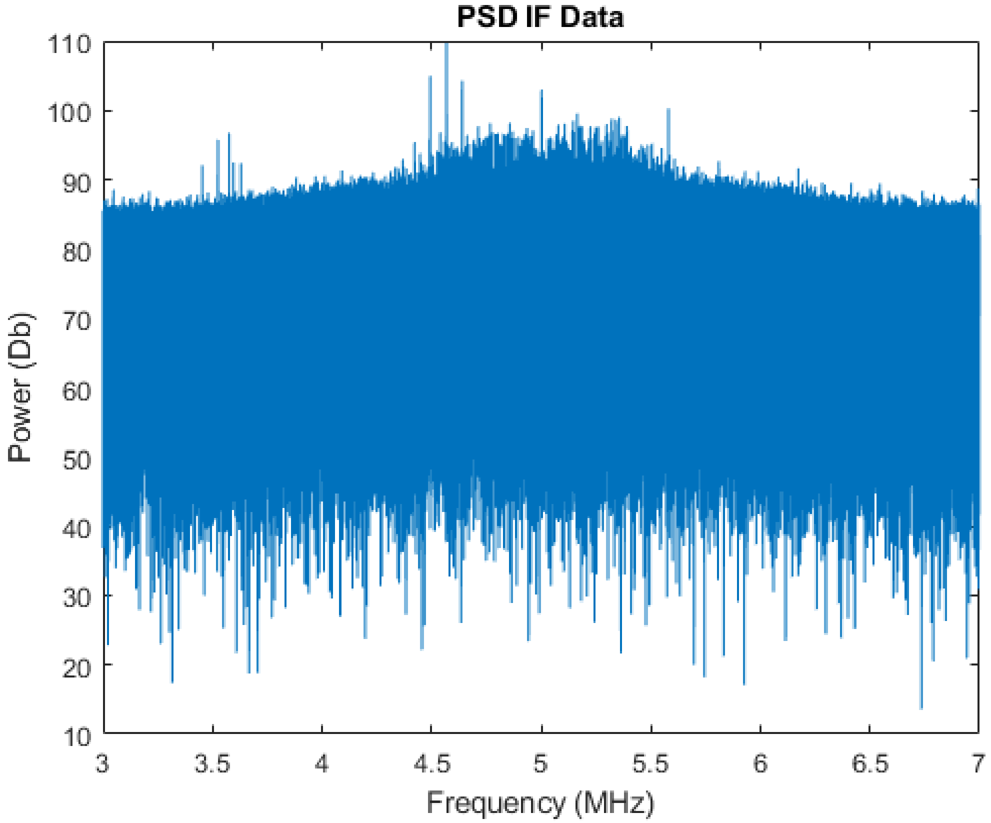


Figure 4: IF Data Power Spectral Density.

The power spectral density shows a power spike around about 5 megahertz with the power at most other frequencies being the same. This spike in power corresponds to the intermediate frequency of the data. Another spike will occur at 15 megahertz after the data repeats. This is what we expect to happen for this set of IF data.

2. **Satellite Acquisition**

The IF data is now used to acquire satellites. To do this, a Costas Phase lock loop (PLL) is used to determine the auto-correlation of the IF data at a range of different Doppler and code phases. The range of Doppler used was 10000 hertz above and below the intermediate frequency in 500 hertz intervals. The range of code phases used was the number of chips in a single PRN sequence. The correlation at each Doppler and code phase was calculated as a combination of the squares of the in-phase and quadrature. The equation to calculate the in-phase and quadrature for any given Doppler and code phase is given as:

$$I = S * CA * cos(2\pi\omega * t) \tag{1}$$

$$Q = S * CA * sin(2\pi\omega * t) \tag{2}$$

where I is the the in-phase, Q is the quadrature, S is the IF signal, CA is the shifted and up-sampled PRN sequence, ω is the Doppler frequency, and t is the time vector. This methodology was used with 1 ms of data to try and acquire satellite 7. *Figure 5* shows the resulting auto-correlation graph for satellite 7.

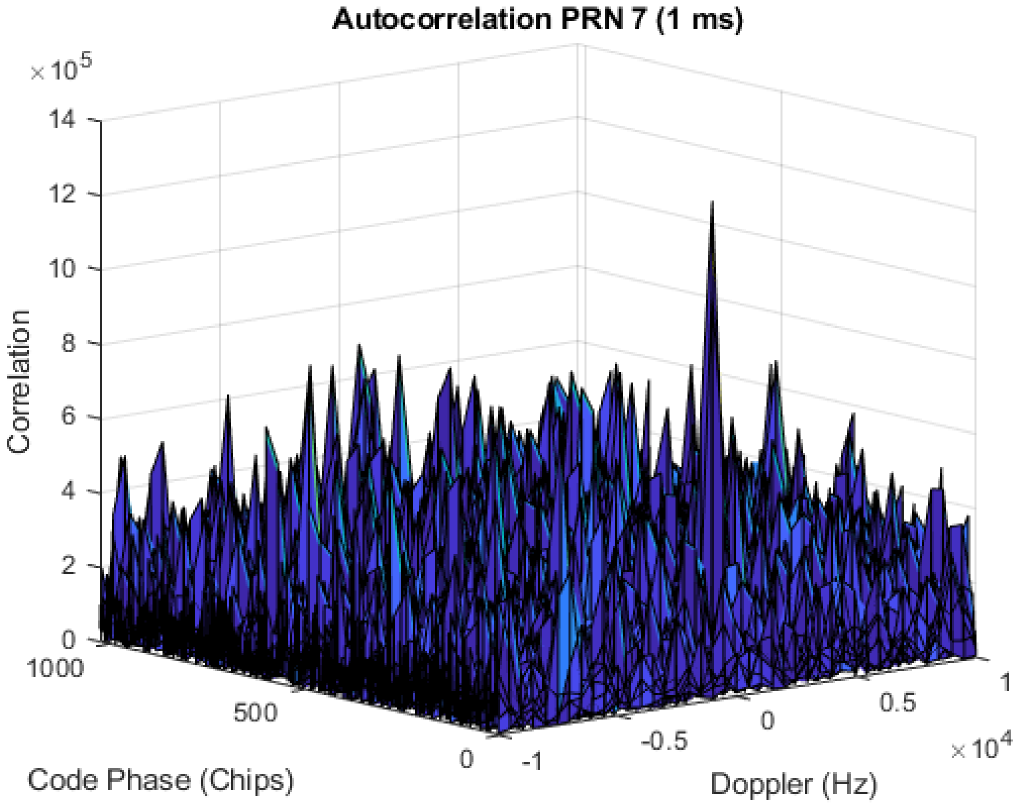


Figure 5: PRN 7 Acquisition 1 ms.

It can be seen from this figure, that there is a noticeable spike in the correlation within this Doppler and code phase windows. Thus, satellite 7 was acquired within the 1 ms of data. The data would be much more uniform if satellite 7 was not available in this time period. This same process was repeated for 10 ms of data in *Figure 6*.

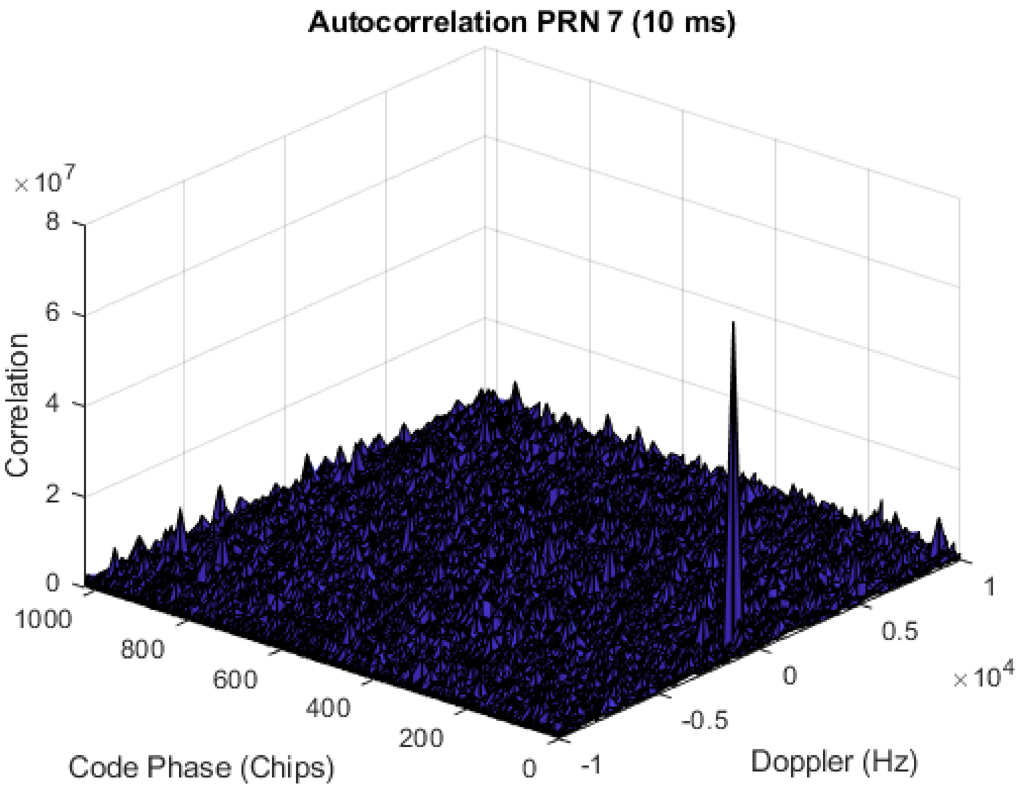


Figure 6: PRN 7 Acquisition 10 ms.

Once again it is shown that satellite 7 has been acquired for this time interval. However, the correlation spike is much more prominent this time around. Increasing the amount of data in the correlation algorithm increases the performance of the acquisition. There is now much more confidence that satellite 7 has been acquired. *Figure 7* shows auto-correlation for satellite 7 over the next 10 ms of data.

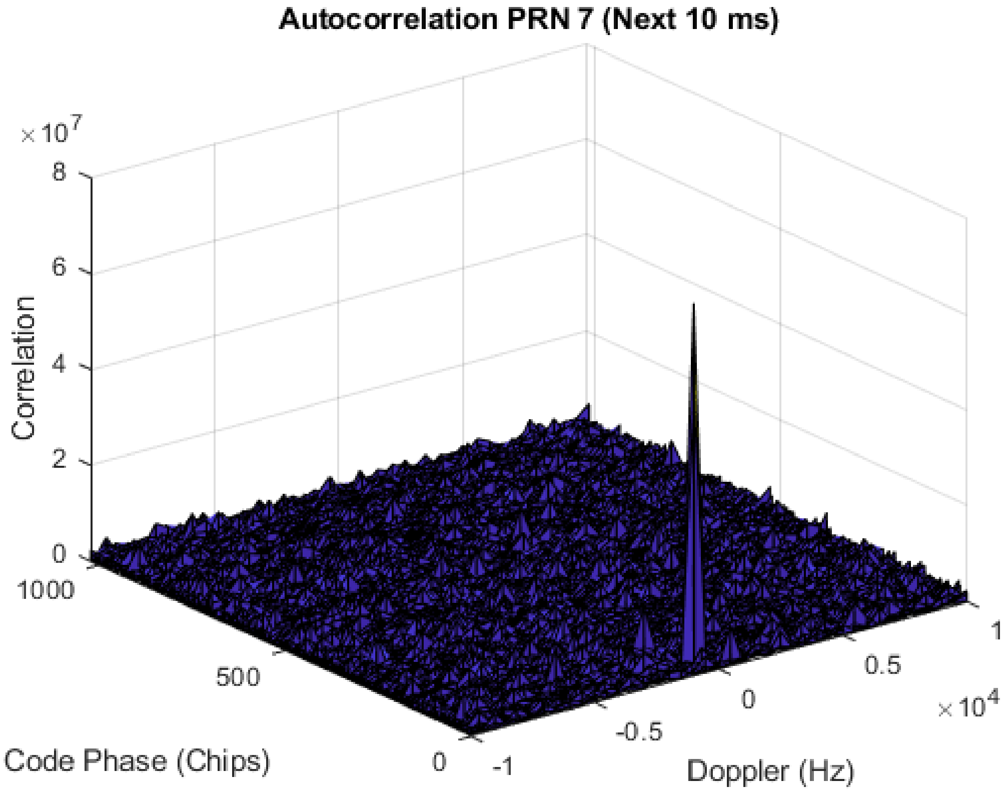


Figure 7: PRN 7 Acquisition Next 10 ms.

Noise was then added to the data sequence with a standard deviation of 6. *Figure 8* shows the resulting auto-correlation graph of the noisy measurement for satellite 7 over 1 ms of data.

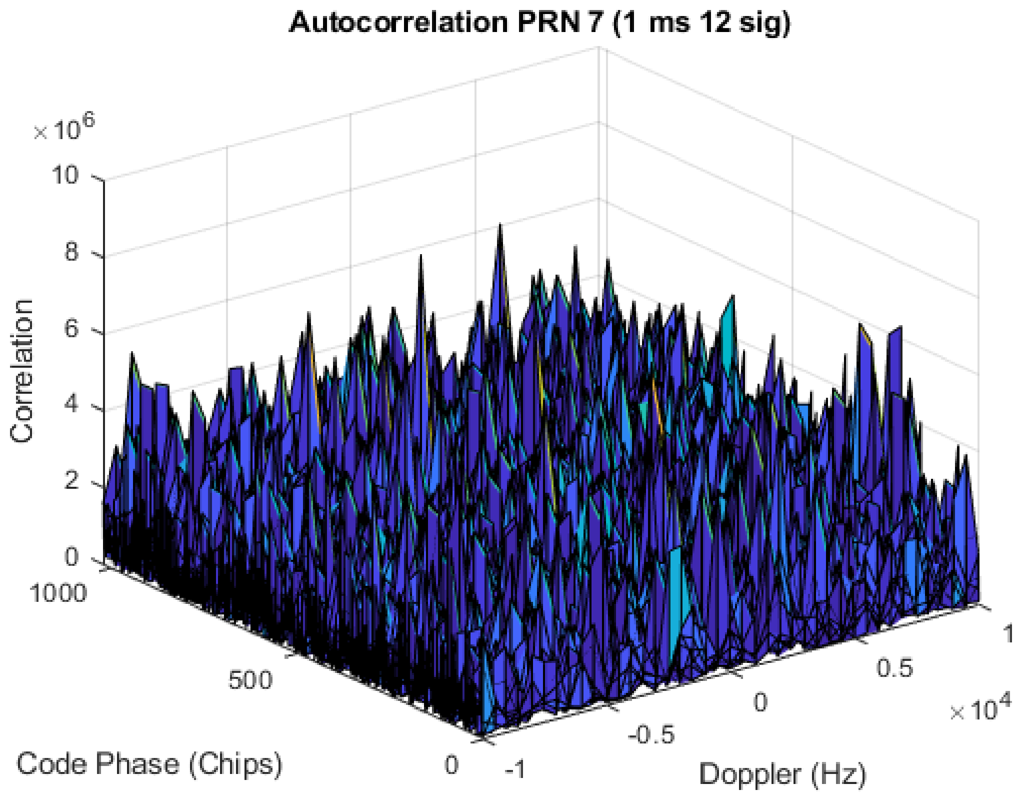


Figure 8: Noisy Satellite 7 (sigma=6) 1 ms.

From the figure, it can be seen that the noise added to data sequence clouds the correlation solution. The peak in correlation is hidden amongst the correlation spikes due to the added noise. There is no longer an ability to acquire satellite 7 for this small time step. To improve the solution, a larger data size will need to be used. *Figure 9* shows the auto-correlation graph for the same noise parameters but with 10 ms of data.

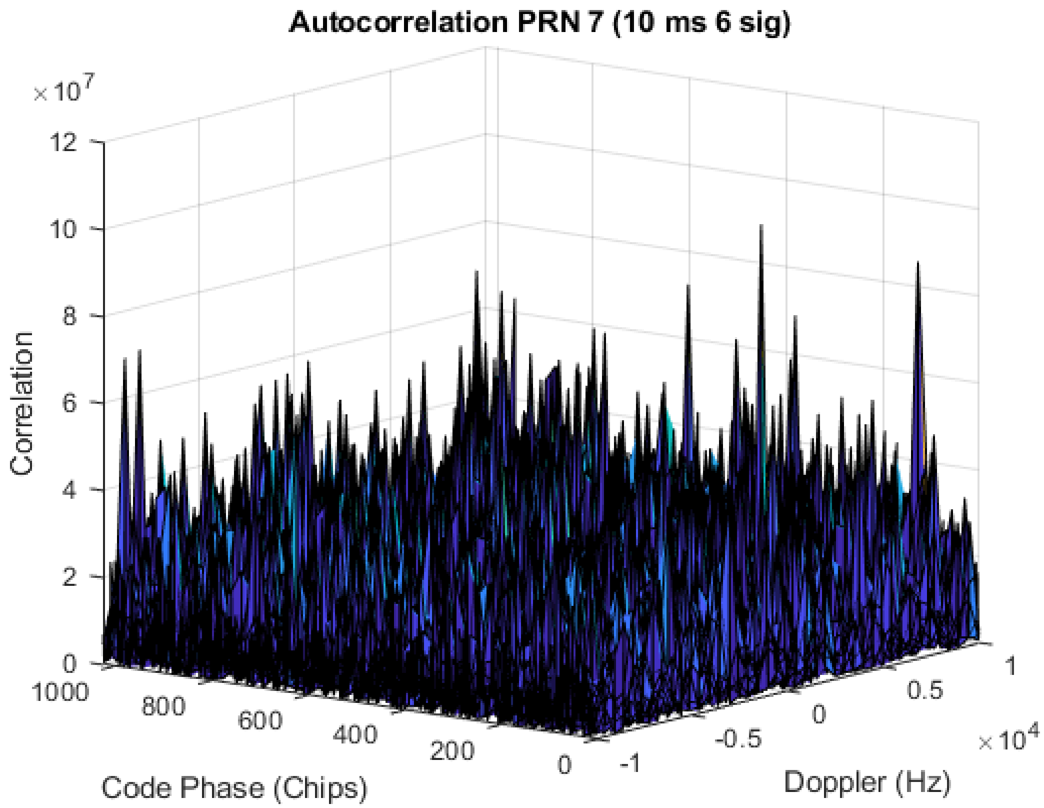


Figure 9: Noisy Satellite 7 (sigma=6) 10 ms.

With the larger data size, the peak in correlation is much more pronounced. It is now possible to determine that satellite 7 has been acquired. The correlation spike is now larger than the noise floor with the larger data size. However, the performance of the acquisition code was much worse with the noisy signal. This same process was repeated but for a standard deviation value of 12. *Figure 10* shows the auto-correlation of satellite 7 with the higher noise value for 1 ms of data.

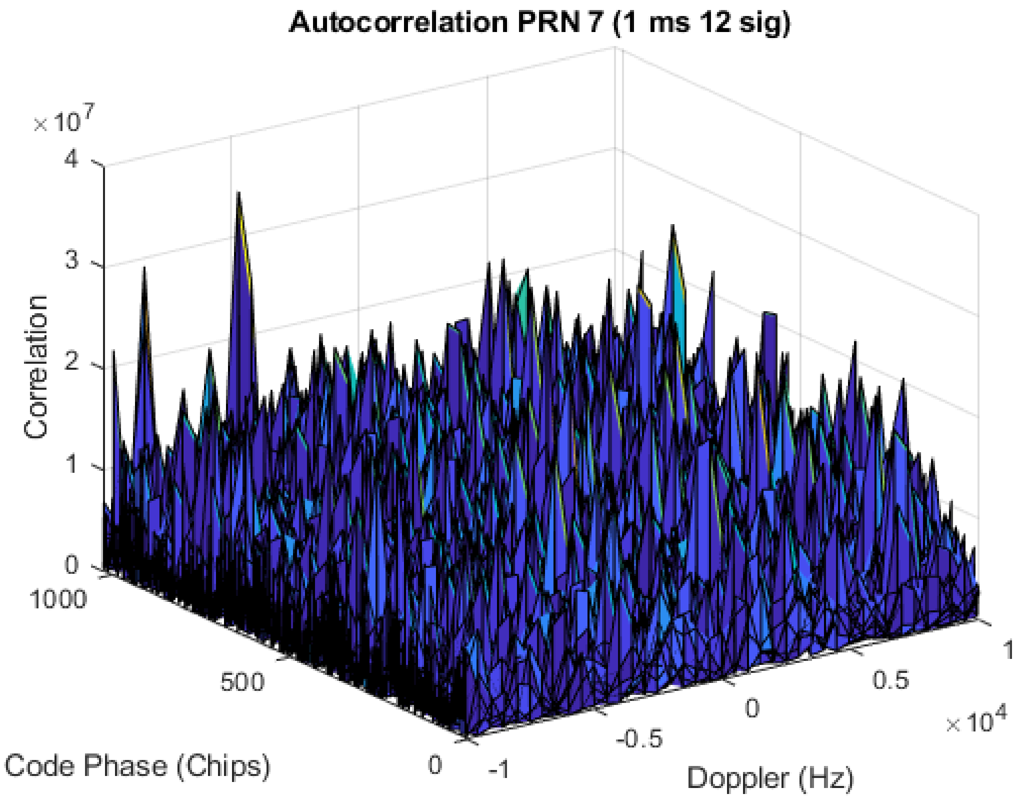


Figure 10: Noisy Satellite 7 (sigma=12) 1 ms.

Once again, it is hard to identify a correlation spike amongst the noise for this small time period. Since the noise standard deviation value is higher, the correlation value is even more unidentifiable amongst the noise. Thus more data needs to be added for the satellite to be acquired. *Figure 11* and *Figure 12* show the auto-correlation graphs for 10 and 20 ms of noisy data respectively.

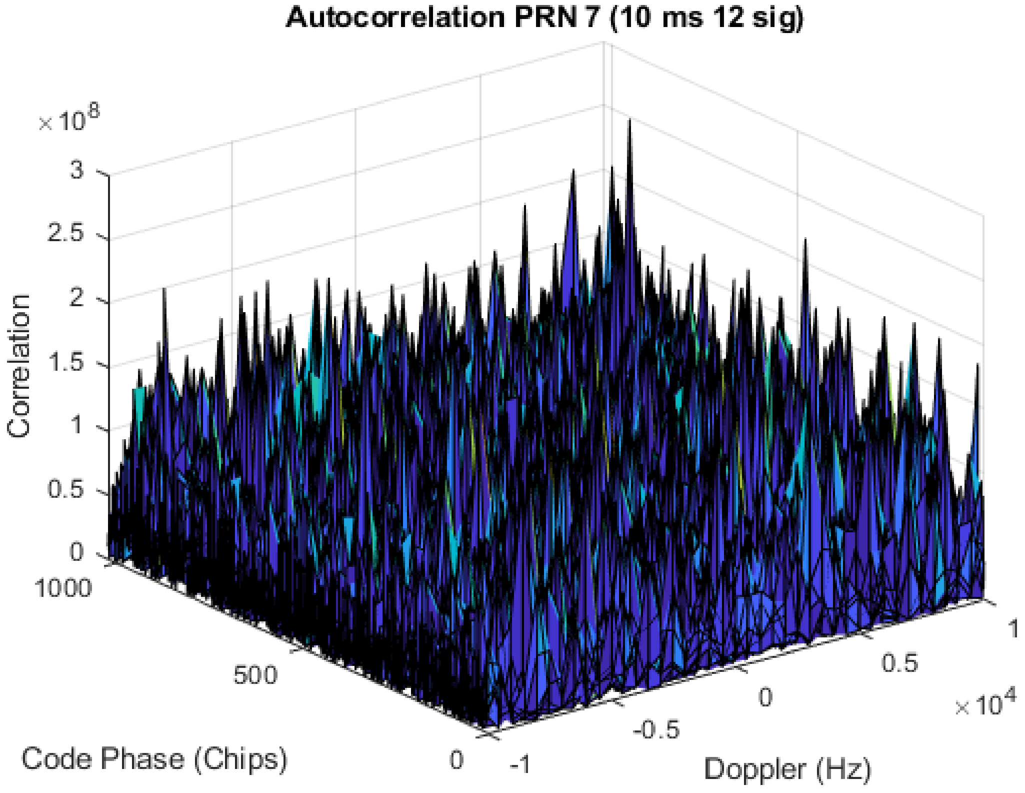


Figure 11: Noisy Satellite 7 (sigma=12) 10 ms.

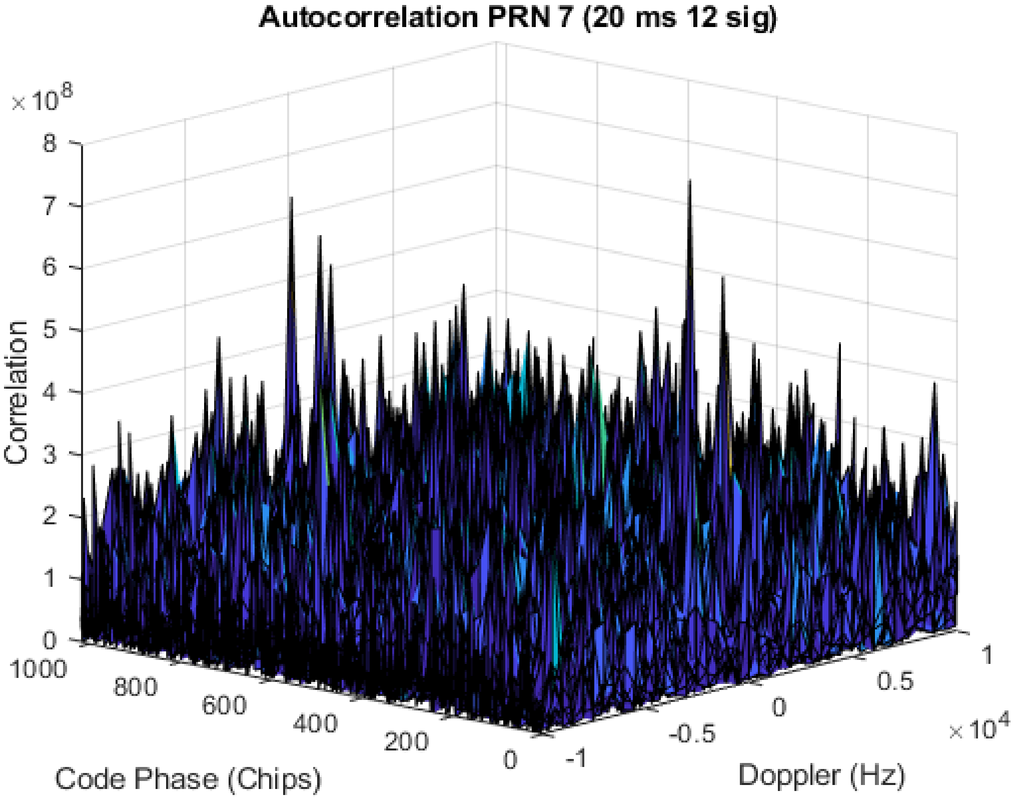


Figure 12: Noisy Satellite 7 (sigma=12) 20 ms.

The increased noise makes it even harder to acquire satellite 7 with even 10 ms of data. It is still possible to see the correlation peak, but the noise clouds the result. When the amount of data is increased to 20 ms, however, the correlation peak is more visible. An increase in noise on the data, requires a subsequent increase in data size to acquire the satellite. This same satellite acquisition methodology was used to acquire more satellites over the 10 ms time period. *Figures 13-17* show the auto-correlation graphs for satellites 1, 14, 17, 21, and 30 respectively.

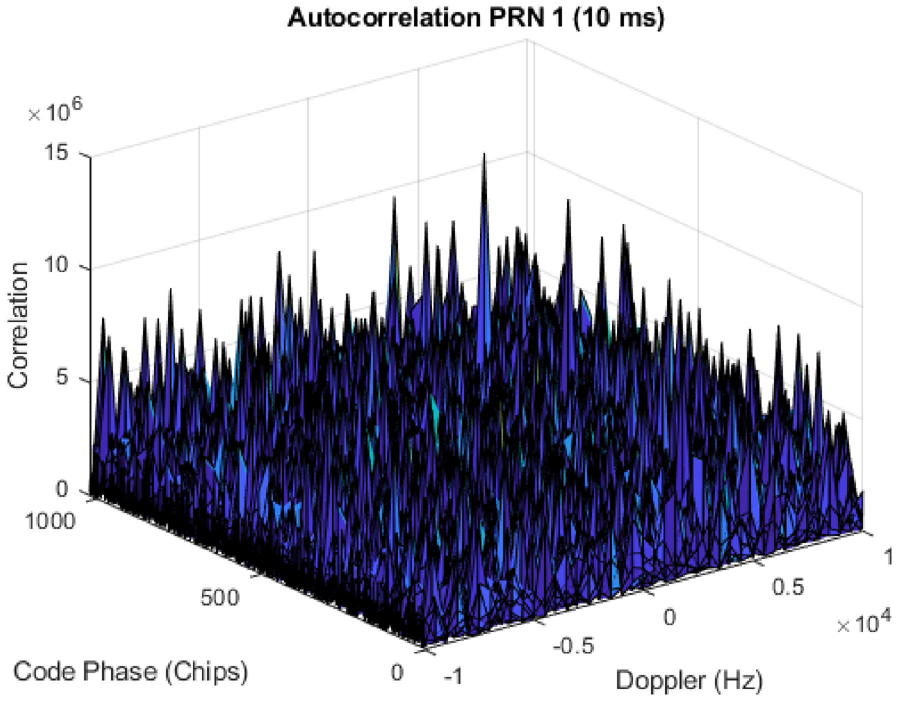


Figure 13: Satellite 1 Acquisition 10 ms.

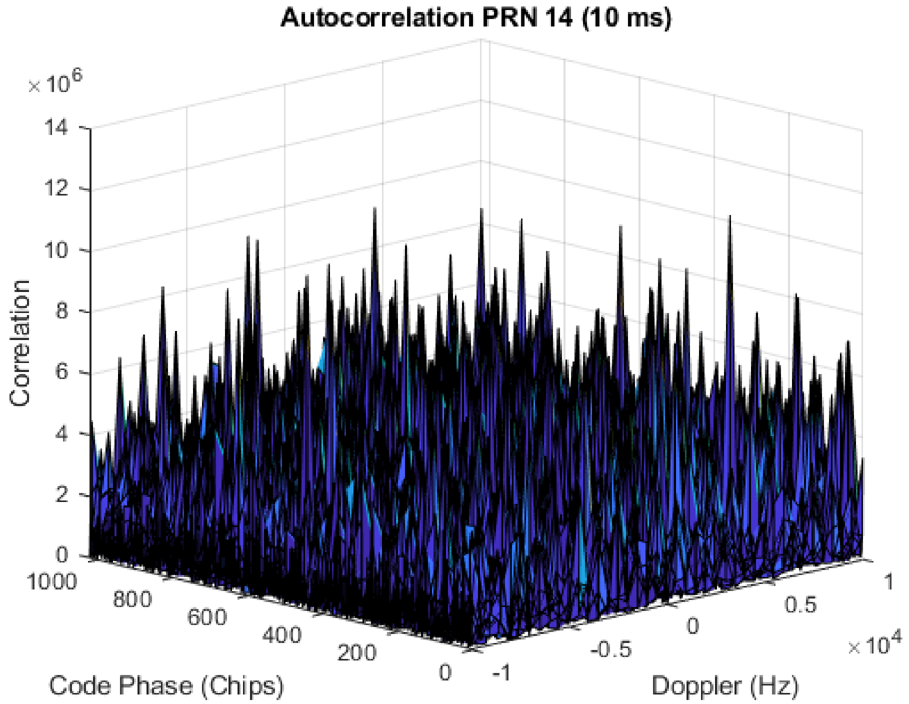


Figure 14: Satellite 14 Acquisition 10 ms.

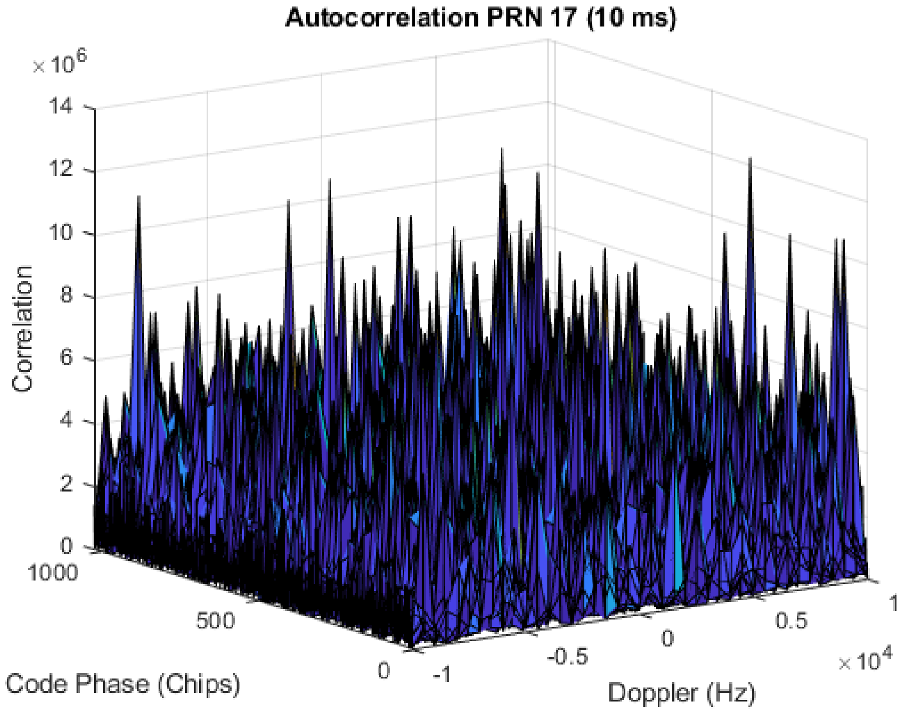


Figure 15: Satellite 17 Acquisition 10 ms.

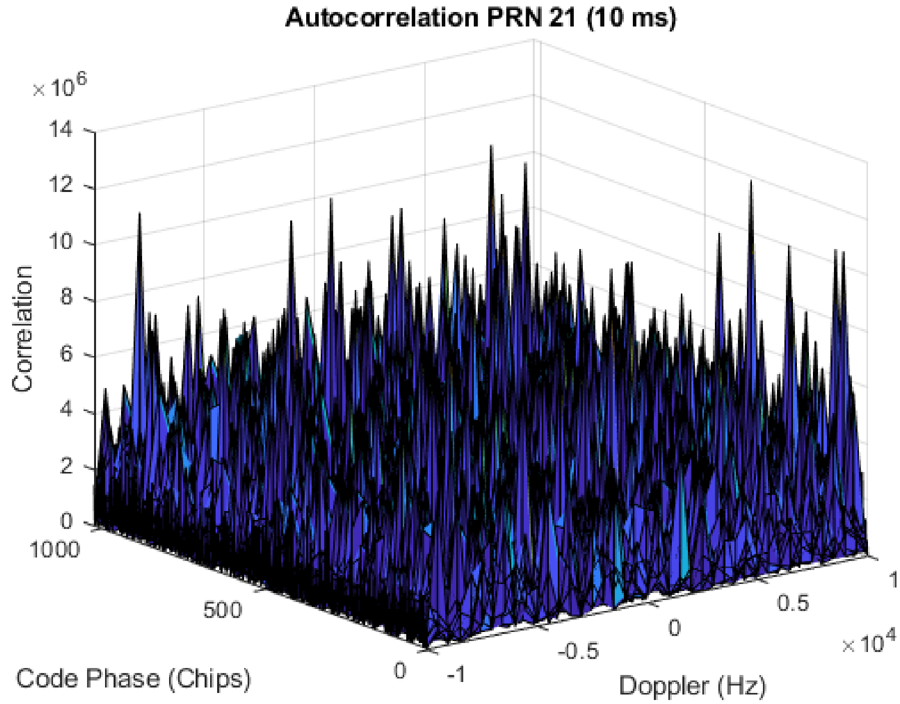


Figure 16: Satellite 21 Acquisition 10 ms.

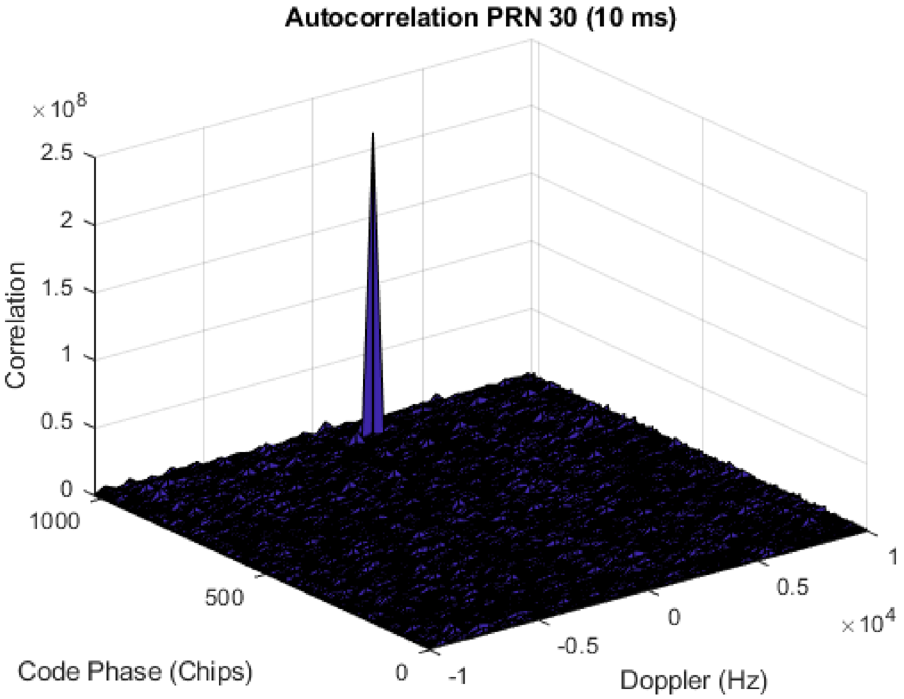


Figure 17: Satellite 30 Acquisition 10 ms.

From the first 10 ms of data clear correlation spike can be seen for satellites 1 and 30. For these two satellites have been clearly acquired. Satellites 14 and 17 also have correlation peaks, but they are a lot less apparent than the other 3 satellites. No correlation spike is seen for satellite 21 for this time period. This same process was repeated for the next 10 ms of data. The results are seen in *Figures 18-22*.

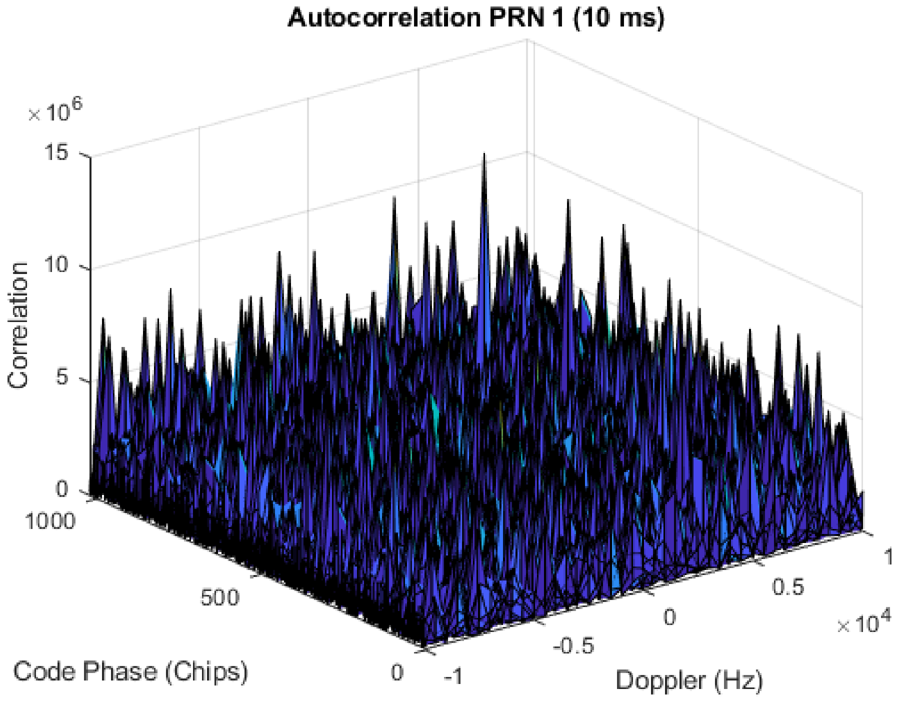


Figure 18: Satellite 1 Acquisition Next 10 ms.

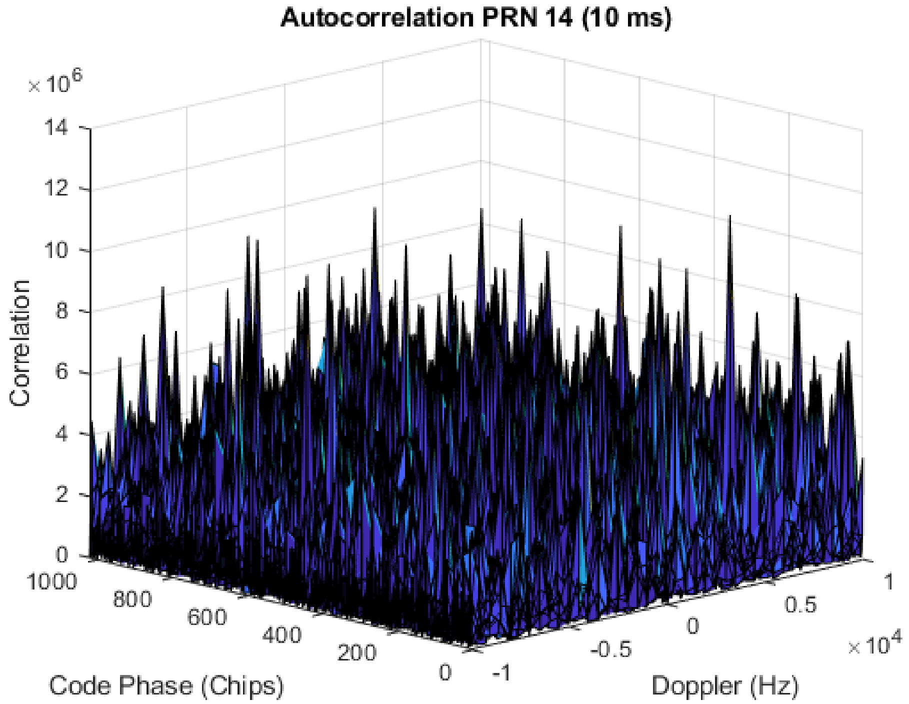


Figure 19: Satellite 14 Acquisition Next 10 ms.

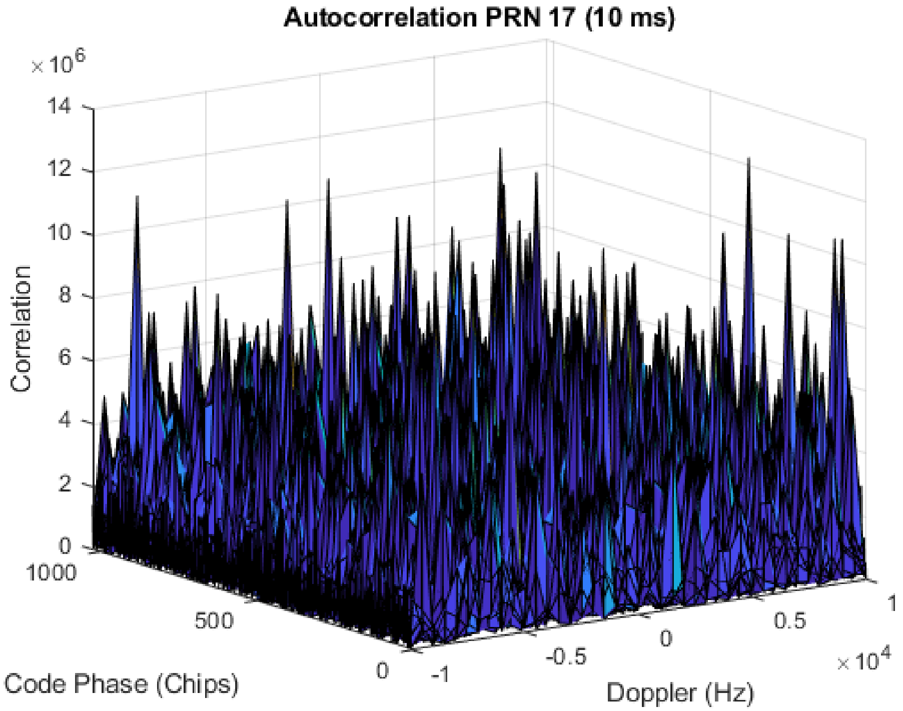


Figure 20: Satellite 17 Acquisition Next 10 ms.

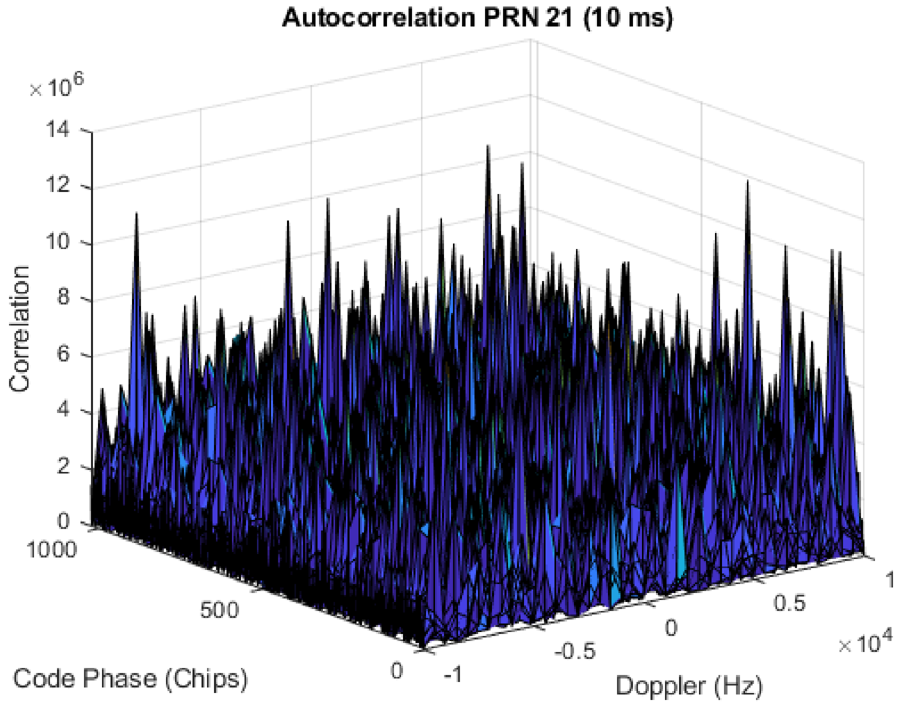


Figure 21: Satellite 21 Acquisition Next 10 ms.

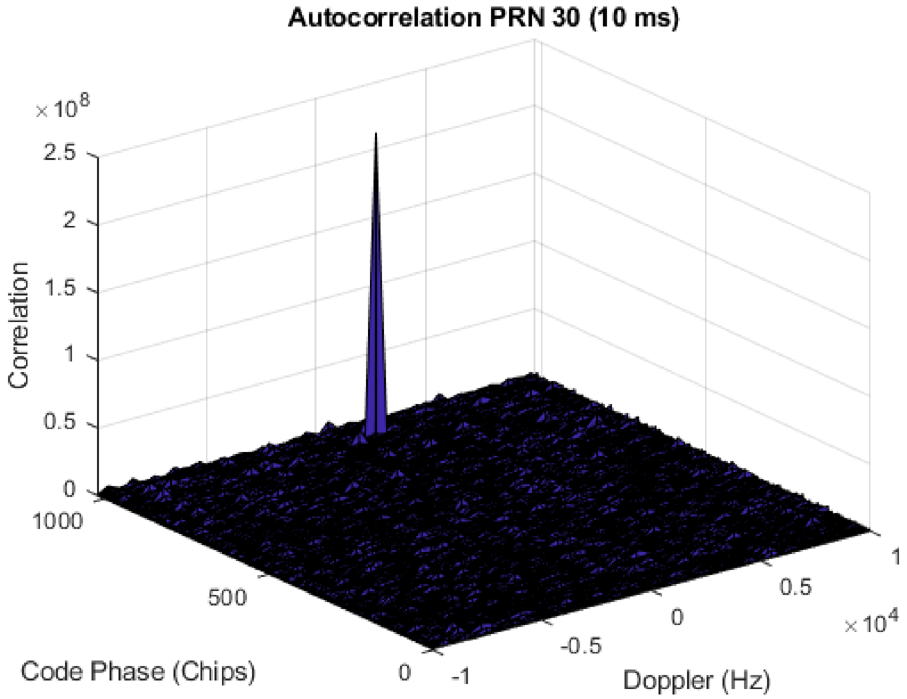


Figure 22: Satellite 30 Acquisition Next 10 ms.

For this next 10 ms of data correlation peaks can now be seen for each of the satellites. The peaks for satellites 14 and 17 become more pronounced, and satellite 21 gains a peak. Satellites 1, 7, 14, 17, 21, and 30 have all been acquired for the first 20 ms of data. This is now more than enough satellites to get a position solution.

Tracking

After acquiring the satellites, tracking was then performed to read the data bits. To perform the tracking a PLL with a 10 Hz bandwidth alongside a 2 Hz DLL. The PLL and the DLL were designed using the following equation:

$$\omega_n = \frac{4BW\zeta}{1+\zeta^2} \tag{3}$$

where $BW_{pll} = 10$, $\zeta_{pll} = 0.707$, $BW_{dll} = 2$, and $\zeta_{dll} = 1.414$. From here the controller gains were calculated using the following equations:

$$\begin{aligned} K_p &= \zeta\omega_n \\ K_i &= \omega_n^2 \end{aligned} \tag{4}$$

These gains result in an Inphase and Quadrature plot similar to the one below for each satellite previously specified. *Figure 23* shows the Inphase and Quadrature for satellite 7.

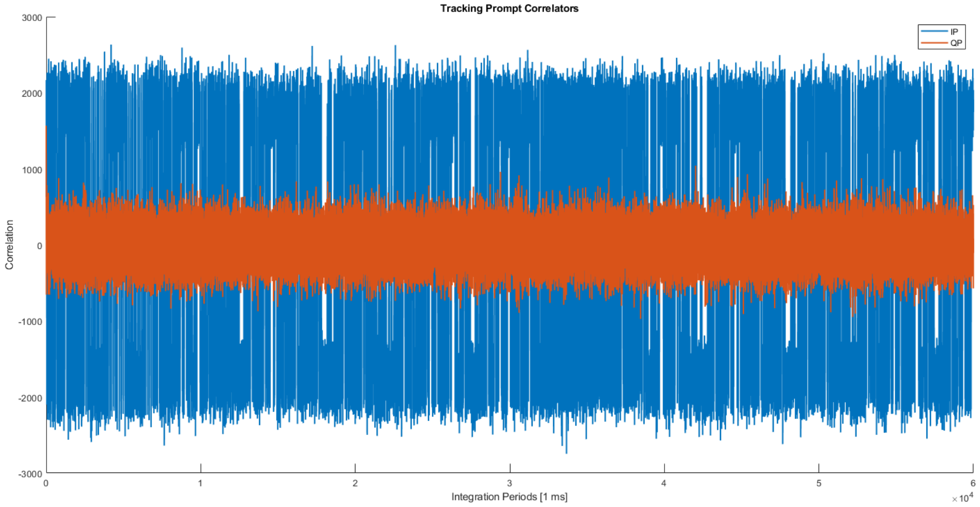
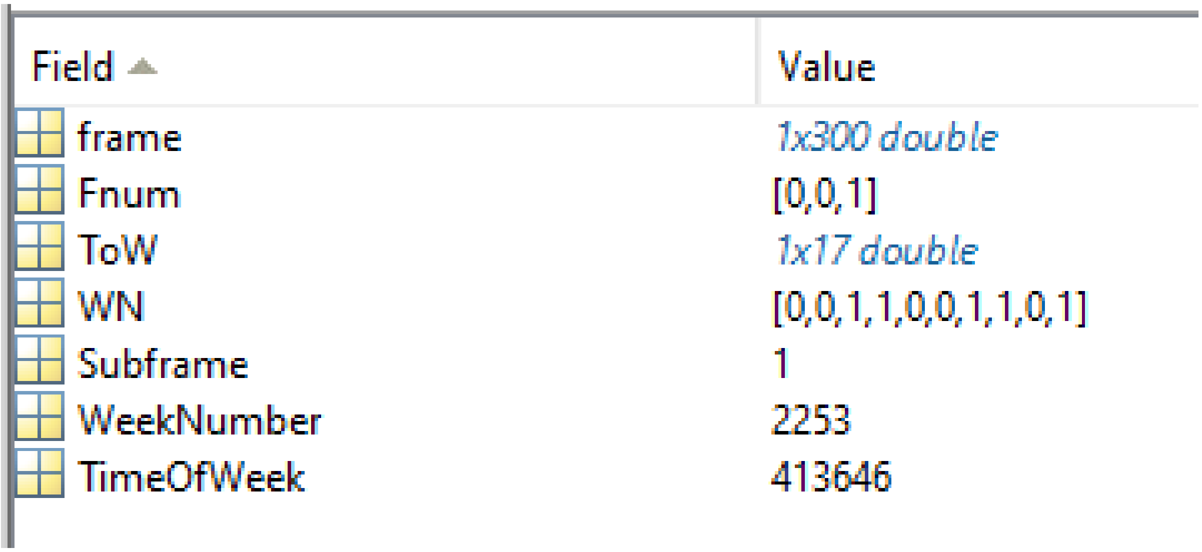


Figure 23: Inphase and Quadrature of SV#14.

It can be seen from the figure that the satellite has been successfully tracked. After tracking has been completed, the data bits are then read after searching for the preamble. For this satellite, the preamble for subframe 4 was the first preamble detected, and the other 4 subframes immediately followed. To read the time of week (TOW) and the week number (WN) for the data, subframe 1 needed to be decoded. The WN number was calculated by decoding the 10 MSB of the 3rd word. After adding the year rollover, a WN of 2253 was found. The TOW was calculated by decoding the first 17 bits of the HOW, or 2nd word. The TOW was found to be 413646. *Figure 24* shows the resulting MATLAB proof of the TOW and WN determinations.



Field ▲	Value
frame	*1x300 double*
Fnum	[0,0,1]
ToW	*1x17 double*
WN	[0,0,1,1,0,0,1,1,0,1]
Subframe	1
WeekNumber	2253
TimeOfWeek	413646

Figure 24: Z-COunt Results for Satellite 7.

After decoding the z-count, the carrier phase and doppler can be determined for the tracked satellite.

Both of these come from the output of the DLL and PLL. *Figure 25* shows the resulting carrier phase and doppler for satellite 7.

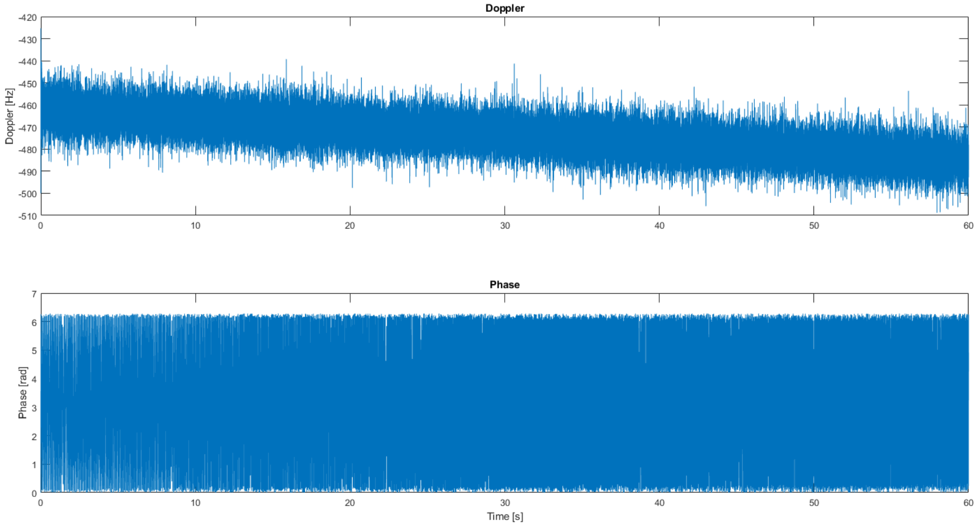


Figure 25: Carrier Phase and Doppler Satellite 7.

The tracking algorithm was then tested for its resilience to noise. Initially, noise with a standard deviation of 6 was applied to the IF data for satellite 7. *Figure 26* shows the resulting Inphase and Quadrature for this noise scenario.

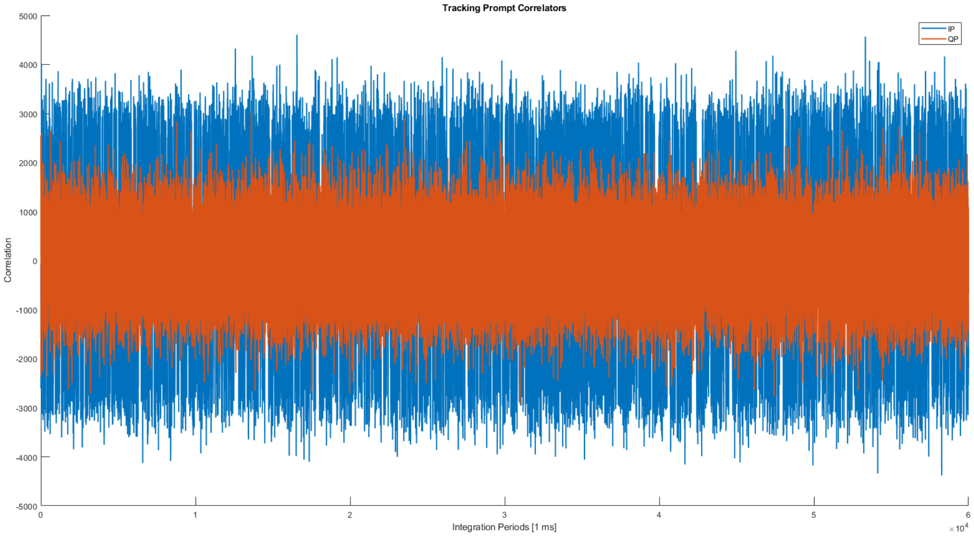


Figure 26: Inphase and Quadrature Satellite 7 (6 sig).

From the figure, it can be seen that the tracking loop can still track the satellite despite this added noise. However this is only works for satellite 7 which is firmly in view. Additionally, when more noise is added to the IF Data, the tracking solution breaks down. *Figure 26* shows the results of increasing the noise to a stanard deviation of 12.

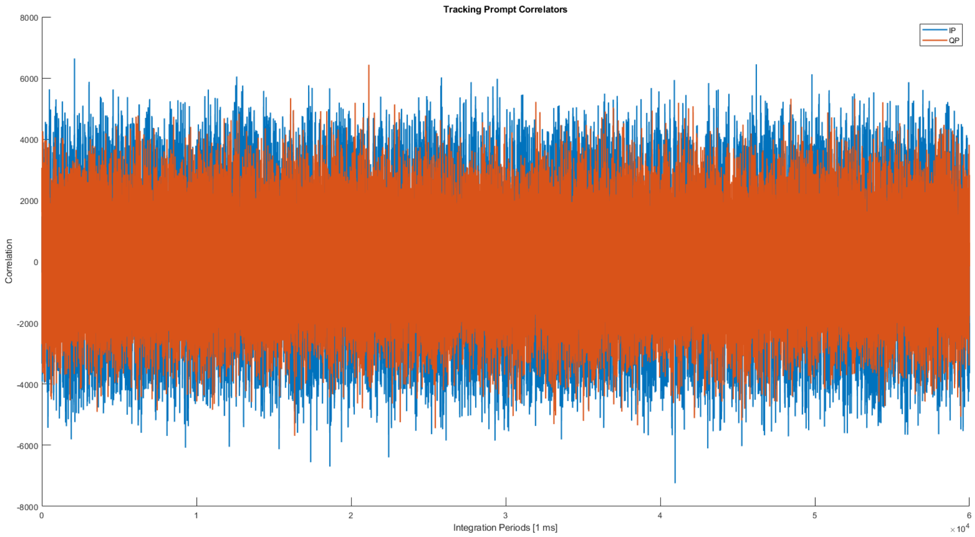


Figure 27: Inphase and Quadrature Satellite 7 (12 sig).

The Inphase and Quadrature are no longer indistinguishable after the addition of more noise. Thus, the satellite is no longer able to be tracked. However, as proven in previous parts, the satellite can still be acquired. It takes less noise to hinder the tracking of a satellite than the acquisition. After reading the data bits, the next step in determining a position solution from the receiver is to determine the psuedoranges. However, we were not able to successfully complete this task before the due date.