Simulink Implementation of a Single Frequency GPS Receiver

Ahmed Sami | 260568821
Alam Parihar | 260570908

Project Supervisor: Prof. Harry Leib

1. Abstract

Traditionally the algorithms that make up a complete software defined GPS are done in assembly, C/C++ or Matlab. This requires a large team to write and test algorithms, which is both costly and time consuming. This project is intended to provide a basis for clear, easy to follow and modify implementation of a L1 carrier GPS receiver. Our simulation employs the use of the graphical programming environment Simulink to design and test models.This project aims to make the fundamental functionality of a GPS receiver more visible and the inner workings simpler to examine and analyse. We hope that our implementation will help pave the way for other simpler methods of implementing and understanding intricate systems.

2. List of Abbreviation

CDMA - Code Division Multiple Access
BPSK - Binary Phase Shift Keying
LO - Local Oscillator
C/A - Coarse/Acquisition
LFSR - Linear Feedback Shift Register
FT - Fourier Transform
PFSS - Parallel Frequency Space Search
PCPS - Parallel Code Phase Search

3. Introduction

Writing the algorithms for a software defined GPS receiver is a large task and requires mastery of the programming language in use. The goal of this project is to implement a complete single channel GPS receiver in Simulink - a graphical programming language which would facilitate the simulation, modelling and implementation of the system by simplifying the complexity and allowing for more straightforward system debugging. Working in a GUI environment will help to reduce the time and resources spent writing out complete algorithms in order to process the different types of signals present in a GPS system. This project aims to investigate the methodologies and fundamental workings of the GPS receiver and allow us to be able to implement the acquisition, tracking and finally the position calculation stages of a GPS receiver. Successful use of Simulink in order to model a complex system like GPS could be used as a basis to model other complicated systems in a similar manner.

4. GPS and SatNav Overview

4.1. SatNav System Architecture

A satellite navigation (satnav) system is one that provides autonomous geo-spatial positioning, allowing users to determine their location at a given point in time to a high precision. Satnav systems can generally be described in terms of three segments (subsystems) shown in the figure below.

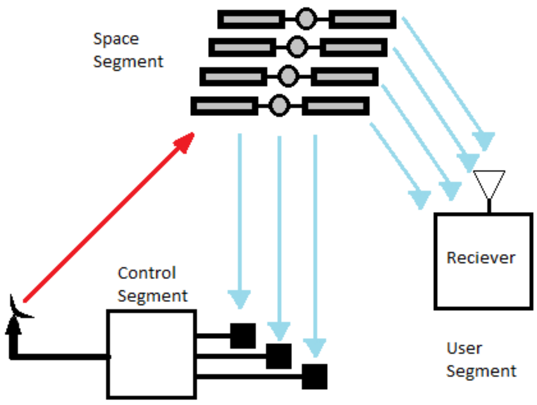

Figure 1: SatNav System Architecture

4.1.1. Control Segment

The control segment of a Satnav system such as GPS is comprised of a global network of ground stations that facilitate the operation of the entire system. The tasks assigned to the control segment include monitoring, tracking and adjusting satellite orbital parameters as well as updating navigation data being broadcast by each satellite.

4.1.2. Space Segment

The purpose of the space segment is to transmit radio navigation signals containing data messages sent from the control segment. The space segment of the GPS system is comprised of a constellation of 24 satellites distributed in 6 nearly circular orbital planes, allowing for 4 satellites to always be visible to a user. The satellites exhibit a Medium Earth Orbit (MEO), with an orbital period of approximately 12 hours. Each satellite transmits its data modulated onto the same carrier frequency.

4.1.3. User Segment

In order to make use of the transmitted data information, it is required to use a radio receiver/processor (capable of operating at high GPS frequencies) in conjunction with an antenna to capture the signal. These software and hardware components are classified under the user segment, which is responsible for performing signal processing and data calculations to determine global spatial positioning.

4.2. Position Calculation Requirements

Ideally, pythagoras theorem could be employed to use 3 points of reference in order to calculate one's position through either triangulation or trilateration; however this is only an option when all receiver and satellite clocks are synchronized.

For a satnav system to produce results with accuracies on the order of a meter would require synchronization between satellite and receiver clocks on the order of nanoseconds. A clock operating at this accuracy for long periods of time would be quite large and expensive. Instead, each satellite signal contains coded information which can be used by the receiver to determine the time and position of transmission. This allows for the incorporation of a fourth satellite to solve a system of 4 unknown equations as shows in the figure below.

$$
\begin{aligned}
cD_1 &= [(x_1 - x_r)^2 + (y_1 - y_r)^2(z_1 - z_r)^2]^{1/2} + c\Delta_r \\
cD_2 &= [(x_2 - x_r)^2 + (y_2 - y_r)^2(z_2 - z_r)^2]^{1/2} + c\Delta_r \\
cD_3 &= [(x_3 - x_r)^2 + (y_3 - y_r)^2(z_3 - z_r)^2]^{1/2} + c\Delta_r \\
cD_4 &= [(x_4 - x_r)^2 + (y_4 - y_r)^2(z_4 - z_r)^2]^{1/2} + c\Delta_r
\end{aligned}
$$

Figure 2: SatNav Position System of Equations

Here, D_x represents the difference in time between the transmission and reception of signals relative to the receiver clock, also known as the *pseudo delay*. Multiplication of this value with the known propagation speed (speed of light; C) results in a pseudorange, representing the straight line distance to the satellite at time of transmission. By using the known coordinate values, distance, and a biasing clock offset(Δ_r), we can solve for the unknown spatial parameters (denoted by the subscript r).

4.3. GPS Signal Properties

4.3.1. Carrier Frequency

The GPS system occupies a very high frequency band (> 1GHz), including some carrier frequencies that are intended for military use only. The scope of this project will only be concerned with the L1 C/A band (also known as the legacy signal) which is broadcast by all satellites at a frequency of 1575.42MHz. Having the carrier at such a high frequency allows for less ionospheric interference, and reduced interference from other high power radio signals transmitters. It is worth noting that since each satellite transmits on the same carrier frequency, a captured signal will have data from every visible satellite. A method for ensuring each of this signals can be received without cross interference will be discussed in the following sections.

4.3.2. Modulation and Multiple Access

The GPS system signals are based on the principle of code division multiple access, which allow receivers to decode chosen satellite signals out of the aggregate of signals being broadcast on a carrier frequency. This method of multiplexing is realized by the modulation of data onto the carrier wave through binary phase shift keying (BPSK) . BPSK uses a direct sequence spread spectrum (DSSS) of binary bits which are used in an XOR operation to modulate the data. This modulated data is then used to phase shift the carrier signal by 180° at every change in binary value (denoted by a rising or falling edge). The DSSS used by the L1 carrier is the known as the coarse acquisition (C/A) code. This code runs at a rate of 1,024,000 bits/second, and modulates data which runs at a rate of 50 bits/second (denoted by C and D respectively in the figure below).

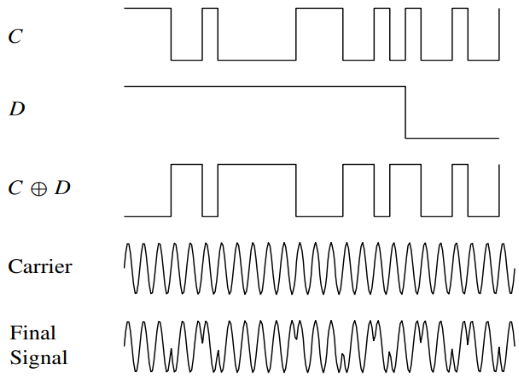

Figure 3: BPSK Modulation Example
(from Reference [2])

Note that another DSSS called the P code exists in the GPS system, however the sequence is not known to the public, as it is intended for military/government use only.

4.3.3. C/A Code

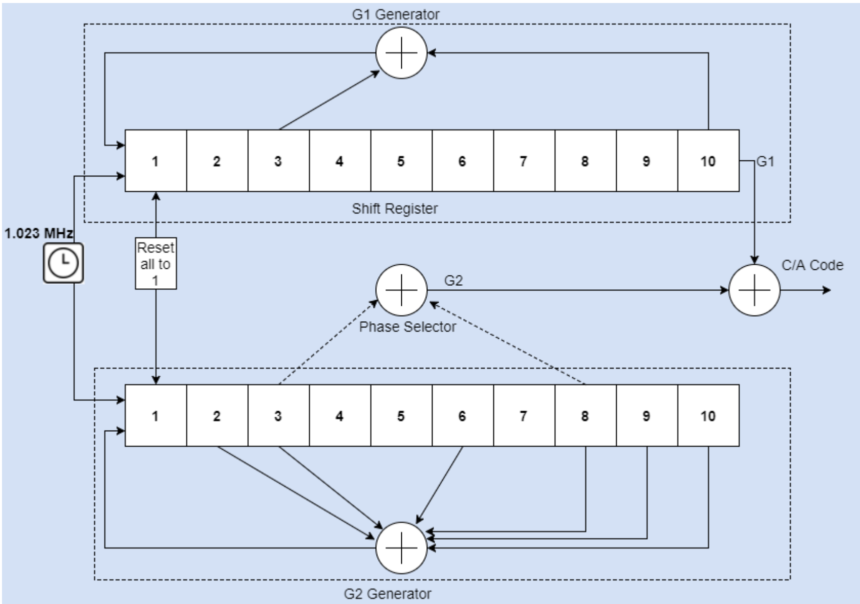

Figure 4: C/A Code Generator

The 1023 bit C/A Code is used to spread the signal and distinguish between each of the 32 different GPS satellites. It is generated using 2 10-bit LFSRs driven by a 1.023 MHz clock. The registers are initially loaded with all 1's , the feedback of the first register is the binary addition of the 3rd and 10th cells and the output sequence is the G1 signal from the 10th cell. The feedback of the second register can be seen in the diagram above, while its output, the G2 signal, is determined by tapping and adding pairs of cells. These pair additions make up the phase selector and correspond to the ID's of the 32 satellites. The G1 and G2 signal are then added to form the 1023 bit C/A Code. For more information on the exact cell pairs and satellite ID configurations see [2].

The C/A codes are PRN sequences based on Gold Sequences, they are essential to understand and use in this application due to their correlation properties. Different C/A codes have little to no cross-correlation with one another, making it easy to distinguish between them. Furthermore, the same C/A code has almost no autocorrelation with a phase shifted version of itself [3].

5. GPS Receiver

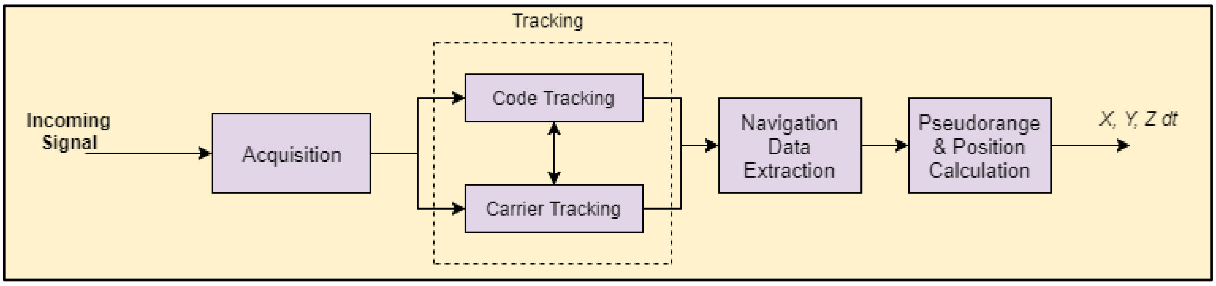

Figure 5: GPS Receiver Block Model

5.1. GPS Receiver Architecture

The receiver's purpose is to take in the incoming signal, accurately demodulate the carrier wave and C/A code and use the data bits to work out its position.

The first stage in a GPS receiver is the Acquisition stage, this receives the aggregate of the satellite signals and processes it to determine which of the 32 satellites are visible and provides rough estimates of the incoming signals carrier frequency and code phase.

The parameters estimated in Acquisition are then passed onto the the Tracking stage which tracks changes due to doppler shifting and movements in the code phase and carrier frequency over time in the current block of data. This stage is used to refine the parameter estimation from Acquisition and outputs accurate values of these 2 parameters.

The code phase and carrier frequency values are then used to demodulate the incoming signal and the 50 Hz Navigation data bits are extracted, these data bits contain all the information the receiver requires from each satellite. The navigation data is then decoded according to US Interface standards and the information is used to work out relative time , psuedoranges and finally, receiver position.

5.2. GPS Receiver Front-End

A typical GPS Receiver front end consists of an antenna receiver the signal and converting it to DT, filtering and amplification and downconversion from the L1 frequency to the IF frequency is also done at this point. Our project contains almost none of this as it is a software implementation and we are ignoring downconversion from L1 and are starting off assuming all signals are modulated on the the IF frequency directly.

The front-end parameters used in [1] are:
IF = 9.548 MHz
Sampling Frequency = 38.192 MHz

Our parameters are :
IF = 9.207 MHz
Sampling Frequency = 32.768 MHz

These values that we have chosen are explained later in the section on the Acquisition stage below.

6. Acquisition

This semester we have almost completely implemented the first stage of a GPS receiver, namely acquisition.The purpose of acquisition is to determine all visible satellites to the user. It calculates rough estimates of the Incoming signal’s code phase and carrier frequency by correlating the signal with locally generated PRN codes and locally generated carrier waves. After identifying a visible satellite, the parameters are passed on to the tracking stage.

6.1. Design Parameters

Before explaining the actual acquisition implementation methods, we will first discuss some of the specific design parameters and constraints relating to the length of the data to be examined, the number of frequency searches to determine the carrier frequency and the size of the FFT used in acquisition.

6.1.1. Data Size

The data size for acquisition processing affects the performance and must be taken into the design considerations.

A longer data length yields more accurate results, however too long a data record results in complex calculation due to repeating C/A codes and increased calculation time. Another factor limiting the data length is the Navigation data bit transitions. If a bit transition occurs it will affect the BPSK modulation by spreading the spectrum when it shouldn’t be , therefore bit transitions must be avoided. Since the Navigation data bits are transmitted at a rate of 50 Hz, this results in a maximum possible 1 bit transition every 20ms (20 C/A code lengths). If a bit transition occurs in the first 10 ms then it is guaranteed to not occur in the second 10 ms therefore the data length should not exceed 10ms.

A balance is needed in order to limit data bit transitions, but still be long enough to obtain accurate information. A data length of 1ms would ensure 1 complete C/A code therefore no lost information, and is small enough to reduce the possibility of navigation data bit transitions as well as the doppler effects and calculation time on the C/A code.

6.1.2. IF & Frequency Increments

Standard IF used is 9.548 MHz. We found that we are only able to use integer multiples of our clock rate of 1.023 MHz. In order to stay as close as possible to 9.548 MHz and preserve that high frequency, we used a multiple of 9 to have a baseband IF of 9.207 MHz.

Doppler shifting causes deviates from the nominal IF value of the carrier frequency of the incoming signal. In the worst case the expected frequency value can be offset by ±10 kHz from the nominal value with rapidly moving receiver and ±5 kHz in static receivers[3].

In this project we have chosen to consider our receiver implementation a static one, meaning we only have a 10 kHz range of frequency deviations as opposed to 20kHz.

In order to remove the carrier from the incoming signal, accurate estimates of the signal's frequency need to be generated and compared against the incoming carrier. Checking frequencies in the range with too large a step between one other results in low accuracy, steps to small results in redundancy.

It is found that with 1ms of data it is sufficient to generate local carrier signals with frequency steps of 500 Hz. Over the 10 kHz deviation range this gives us 21 possible frequencies (including the baseband IF).

6.1.3. FFT Size

FFTs and DFTs can be used in acquisition to determine the carrier frequency by checking frequency components that cross a determined threshold; see Acquisition Methods below.

The Sampling frequency of 38.192 MHz, meaning 38,192 samples over the 1 ms, and DFT of size 38,192 were the parameters used in [1]. Given that FFTs and IFFTs are significantly faster than simple DFT implementations and Simulink is optimized for FFTs, we would choose FFT implementations over DFT. The problem then arises that FFTs require an input of radix-2 and 38,192 does not satisfy this. We have chosen to do our sampling of the incoming signal at a rate of 32.768 MHz, therefore implementing a 32,768-point FFT which corresponds to 2^{15}

6.2. Acquisition Methodology

There are 3 main methods applied in the GPS acquisition phase, software implementations generally are incentivised to use the Parallel Code Phase Search Acquisition method due to the lack of care for hardware requirements. We will examine and compare the functionality of the 3 methods below and explain why we have chosen that method.

6.2.1. Serial Search Acquisition

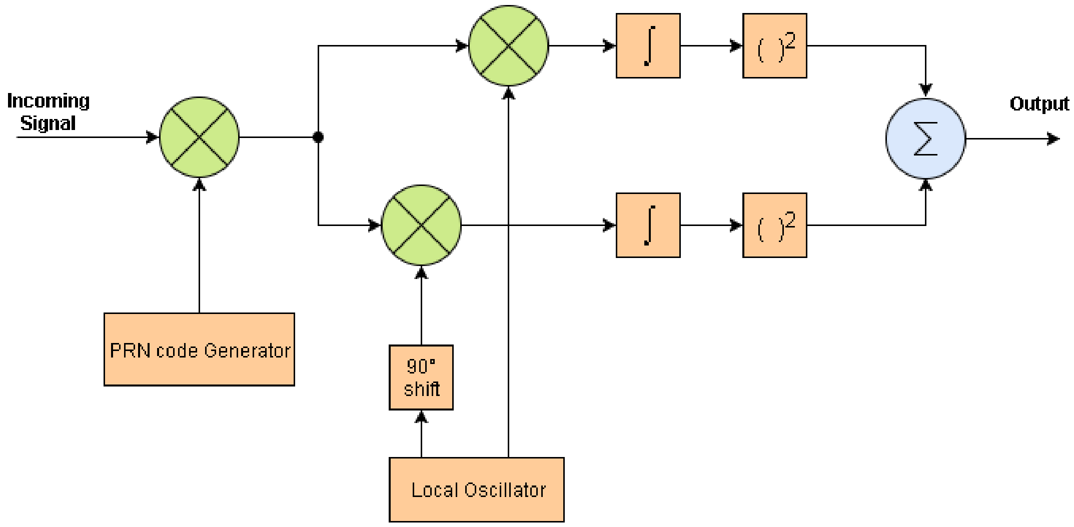

Figure 6: Serial Search Acquisition Block Model

First a PRN code is generated and multiplied with the incoming signal, this needs to be done for all 1023 possible phase shifted versions of all 32 PRN sequences. Resulting in 32,736 PRN codes to check. The next step is to multiply this with a locally generated carrier (and a 90 degree shifted version of the carrier - Sin and Cos). The carrier, which runs for 1ms, is generated at a frequency = IF ± Freq. Step, to cover all 21 possible frequencies spanning over the ± 5kHz range.

The incoming signal, local PRN code and locally generated carrier are all sampled at the same preset sampling frequency (32.768 MHz in our case). The 2 signals after the LO multiplication are integrated to sum up all 32,768 data points and then squared to obtain the signal power. These 2 signals are then summed together to provide the correlation data between the incoming signal and the generated PRN code and carrier. If the generated PRN's code is time-aligned with the incoming signal's code and the generated carrier frequency is within 500 Hz of the incoming carrier then the correlation output should cross a threshold (determined during testing), indicating the visibility of the satellite and marking coarse estimates of the signal's carrier frequency and C/A code to be passed onto tracking.

This sequential search method yields high accuracy however is it a process involving tens of thousands of iterations and measure, therefore it results in large acquisition time. The following methods significantly reduce this processing time by eliminating the need to sequentially search all possible values of the one of the 2 parameters.

6.2.2. Parallel Frequency Space Search Acquisition

This method parallelizes the frequency parameter through the use of a FT, thus eliminating the need to sweep through all 21 different possible frequencies for each PRN sequence.

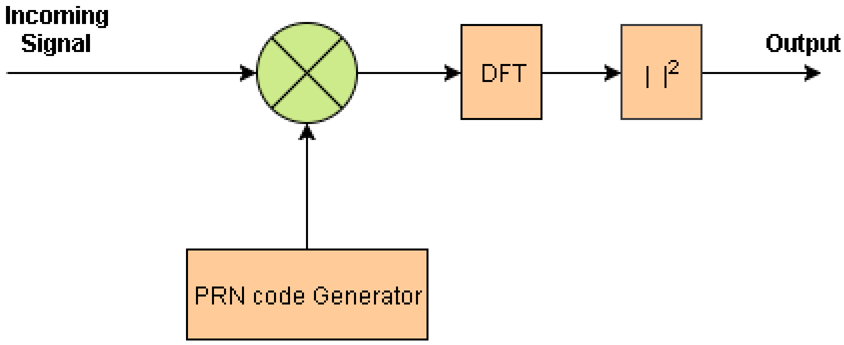

Figure 7: Parallel Frequency Space Search Block Model

The incoming signal is multiplied with a locally generated PRN code, for all 0 to 1022 different code phase alignments of all 32 PRN codes. The FT of that signal is then taken, converting it into the frequency domain. The FT will result in a complex signal, there the absolute value is taken and squared to receiver the signal power. If the generated PRN code is time-aligned with the incoming code, the output will have a distinct peak magnitude above a preset threshold at the incoming signals carrier frequency and the PRN code phase.

This method reduces the number of searches per generated PRN from searching all code phases and all 21 frequencies to just all 1023 code phases. This is a less time-consuming algorithm and is generally preferable, however it reduces the accuracy compared to the Serial Search[3].

6.2.3. Parallel Code Phase Search Acquisition

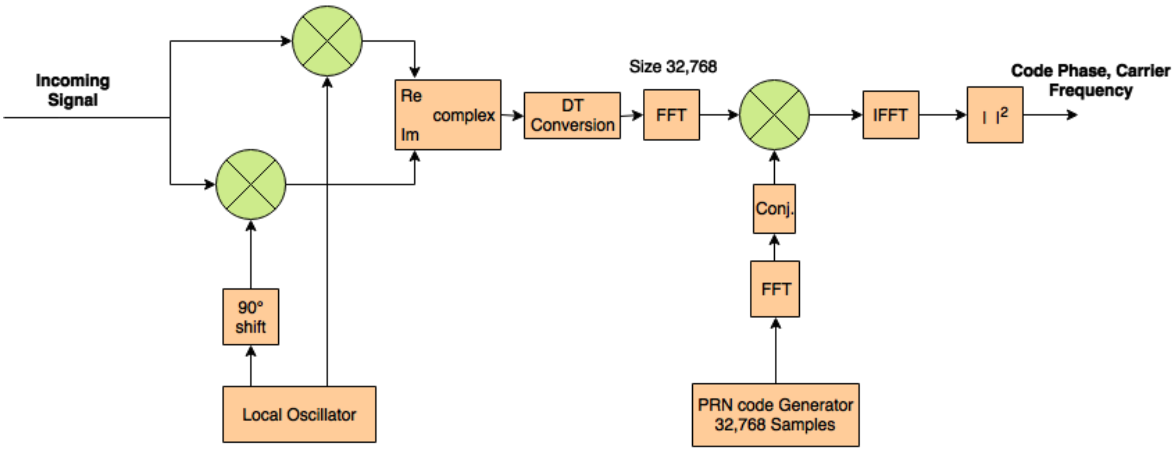

Figure 8: Parallel Code Phase Search Acquisition

Our chosen method of implementation is shown in the block diagram below.

This method parallelizes the code phase parameter, eliminating the need to sweep through all 1023 code phases for every PRN and leaving only the 21 possible frequencies to sweep through.This is done through circular cross-correlation between the incoming signal and the generated carrier and generated PRN code with no phase shifts.

The incoming signal is multiplied by a locally generated carrier wave (and a 90 degree shifted version just as in the Serial Search method), sweeping through all possible 21 frequencies. The 2 LO multiplications are then combined into a complex signal and are sampled at a frequency of 32.768 MHz, over 1ms. This signal is then fed into a 32,768-point FFT.

At the same time the PRN code is generated and transformed into the frequency domain using the same method. The PRN code FFT output is complex conjugated and multiplied with the carrier FFT as part of the correlation method. The result is then taken back into the time domain using and IFFT and the absolute value is taken and squared to give the time domain correlation value between the input and the generated PRN code. Once again, if at the tested frequency and PRN code there is a peak magnitude above a certain threshold, it would indicate the visibility of a satellite and mark the code phase and confirm the carrier frequency.

This method is clearly the most efficient, it provides the same accuracy as in the serial search [3] however it significantly reduces the number of iterations per PRN tested (21 iterations). This is clearly the most efficient and superior method and is why it was chosen.

Below is a table showing the relative execution times and number of iterations needed per PRN code:

Acquisition Method	**Relative Execution Time**	**Number of Iterations per PRN**
Serial Search	87	21,483
PFSS	10	1023
PCPS	1	21

Table 1: Acquisition Method Comparisons

7. Simulink Design Implementation

7.1. Acquisition Model

Using the design parameters described in the previous section, we were able to implement our simulink model from the block model design.

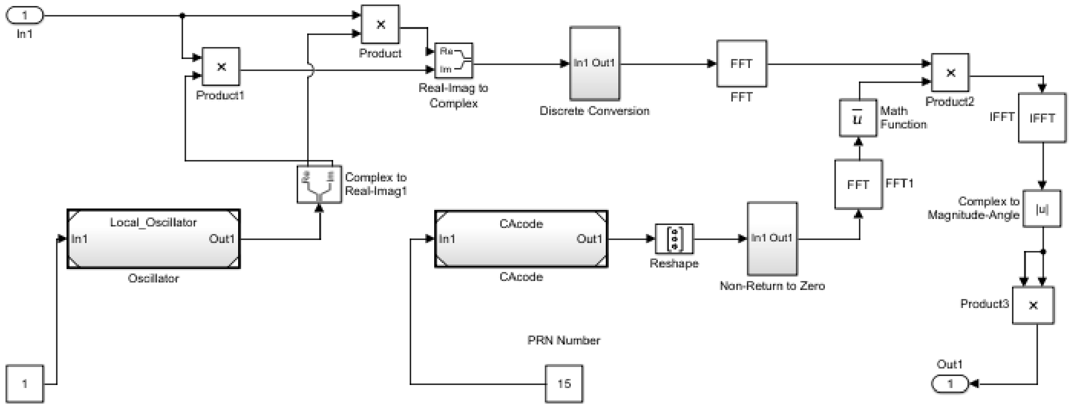

Figure 9: Simulink Acquisition Model

Our model is almost a direct implementation of the block model. We have written matlab code for the local oscillator and C/A code generator, which have been implemented as function blocks in simulink. A description of these models and simulation results are included in the following section. The non-return to zero block was created to change the C/A code form the set {0,+1} to (-1, +1,}. It performs this action by doubling the chip value and then subtracting one.

The fast fourier transform utilized in our model requires for the input to be in the form of 2^n (where n is a positive integer). We have chose to perform our fourier transform with 32,768 samples (n=15), corresponding to a sampling frequency of 32.768 MHz. This value was chosen as it is close to example front end sampling frequencies. In this implementation, each C/A code chip is sampled 32 times, giving us the total required 32,768 samples to feed into the fourier transform.

7.2. Submodel Simulations

7.2.1. C/A Code Generator

The C/A code generator implemented in our acquisition model is based on matlab code found on the mathworks file exchange. It takes as an input the PRN (satellite) number from 1-32, which is used to select the taps of the shift register explained earlier. There is also an option to resample chips at an integer rate, by changing a value within the model; as mentioned above, we have set this resample value to 32.

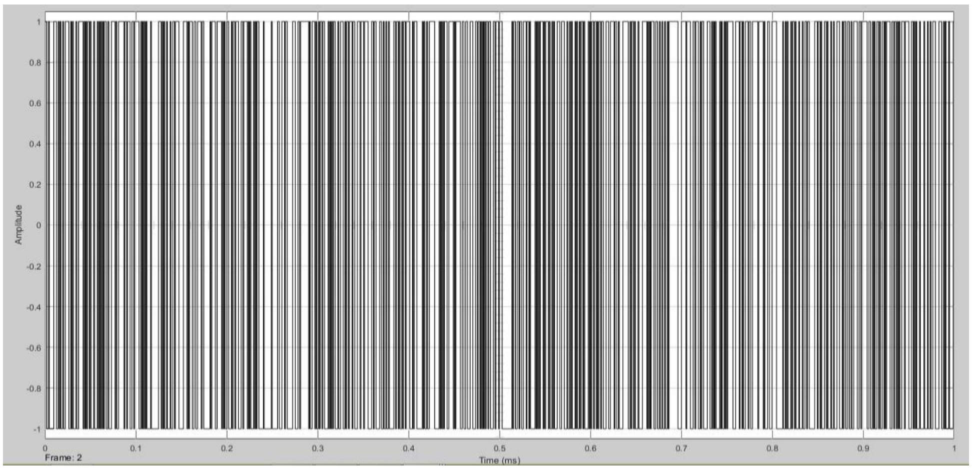

Figure 10: C/A Code for PRN 1

This figure serves to visually show how many chips occur (1024) in 1ms (1 full C/A code). Each horizontal line (signal edge) in this simulation represents a binary change from the previous chip (-1 → +1 or +1 → -1). Each one of these edges will cause the phase shift in our BPSK modulation.

The C/A code generator was tested by choosing random PRN numbers and comparing them to known values provided in reference text [2].

7.2.2. Local Oscillator

The local oscillator was created using a Matlab function block. The matlab code was written to take as an input the desired frequency step to be checked and generates a complex signal using a complex exponential function. This supplies us with both the required oscillator wave as well as the 90^{O} phase shifted version.The simulation below shows both waves operating at a frequency of approximately 9.2MHz.

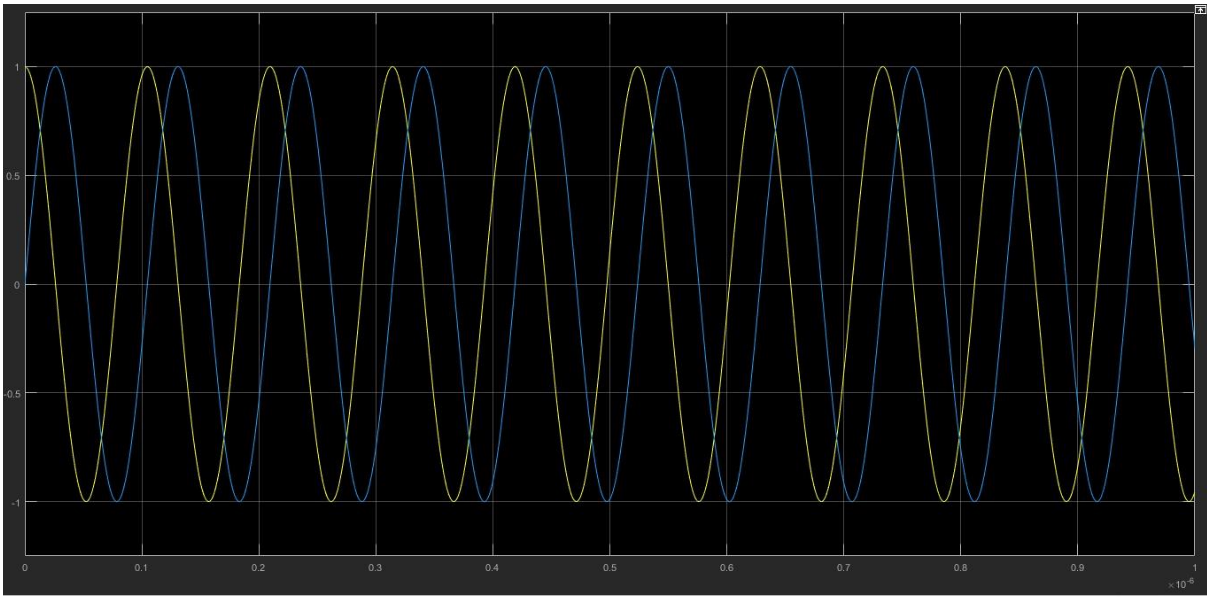

Figure 11: Local Oscillator Simulation (~9.2MHz)

8. Acquisition Stage Testing

8.1. Test Signal Generator

Testing of the acquisition stage requires an input signal with properly modulated data on an appropriate carrier frequency. It was determined that the most appropriate way of creating such a signal would be to simply XOR a 50Hz square wave (simulating data) with a real generated C/A code representing a satellite. This data is put into non return to zero format and modulated through multiplication onto a carrier frequency left to our choosing. With the L1 band, the carrier frequency is a multiple of the C/A code chip rate, allowing for optimal BPSK. As mentioned previously a hardware front end will down convert this carrier to an intermediate frequency, with our chosen example having a value of ~9.5MHz. Using this knowledge we have chosen to modulate our test signal onto a carrier frequency of

9*(C/A chip rate), or 9.216MHz, to maintain an optimal modulation and IF range. For the purpose of describing a base frequency, we will refer to this value of 9.216MHz as our IF.

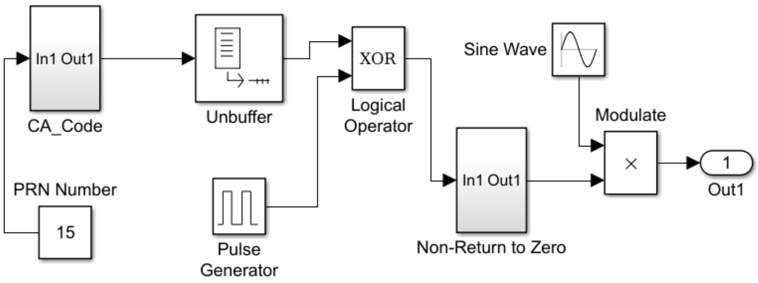

Figure 12: Signal Generator

Note that at the moment any doppler frequency shift is manually entered as an addition to our chosen carrier. For future testing with the tracking stage it may be required to employ the use of a voltage controlled oscillator to simulate a moving satellite signal with a changing doppler frequency shift.

To confirm that this signal generator functions correctly, a scope was attached to the output when the PRN number 15 is fed into the system. The waveform below shows the results of the BPSK, which does indeed look correct. The frequency of the sine wave carrier in this figure is approximately 9.2MHz.

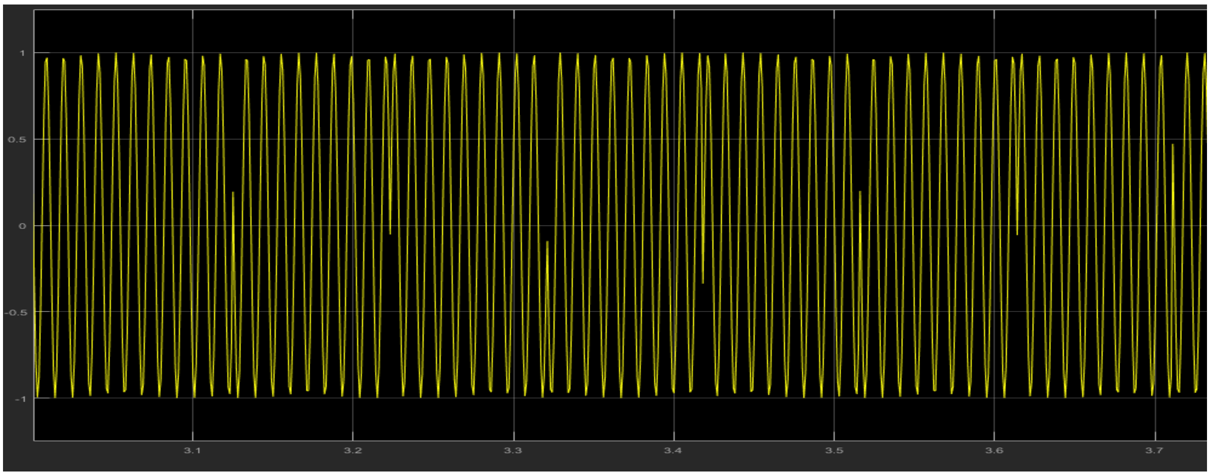

Figure 13: Test Signal Simulation (9.2MHz)

8.2. Single PRN Single Frequency Test

The first step in testing our Acquisition stage was to confirm the ability to detect a single PRN code. The test signal generator was set to use a PRN code of 15 at our IF of 9.216MHz, and 1ms of output data was written to a file; this file was then used as an input to our Acquisition stage. Simulation of the acquisition stage was first performed with a PRN of 15 at our base IF, and the output array was saved to a file. A second simulation was performed with the same input file for a PRN of 20 at base IF. The results of these tests are

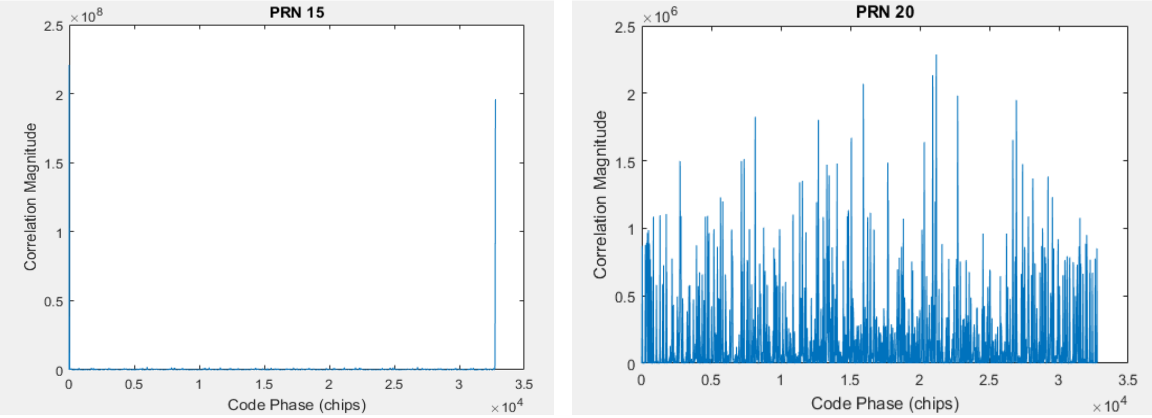

Figure 14: Single PRN Test Results for PRN 15 & PRN 20

shown in the figures below, as plots of the output array.

For a PRN of 15 which is visible to the system, the plot show a strong correlation, with the magnitude being on the order of 1x10^8. The second plot shows that for a PRN of 20, which is not visible to the system, there is no distinct correlation peak, and any peak that does occur has a magnitude on the order of 1x10^6.

The next test was performed with a similar test signal, still of PRN 15 but with a carrier frequency shift of +5000Hz. The plot below used a PRN of 15, with the local oscillator at base IF.

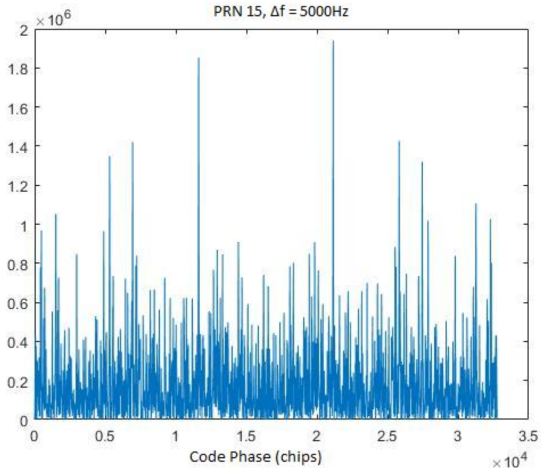

Figure 15: Single PRN test for PRN 15, frequency shifted 5000Hz

As can be seen, there is no distinct correlation peak, as the incoming signal is at IF+5000Hz. Further test results with varying frequency confirmed that even with correct PRN, the correlation only has a distinct peak with sufficient magnitude at the correct frequency +/- ~500Hz.

8.3. Multiple PRN Frequency Sweep Test

After verifying that our Acquisition stage was able to correctly identify the correlation of a single PRN on a signal, it was essential that we broaden the scope of our testing. Four test signal generators were summed together to create one signal as shown below.

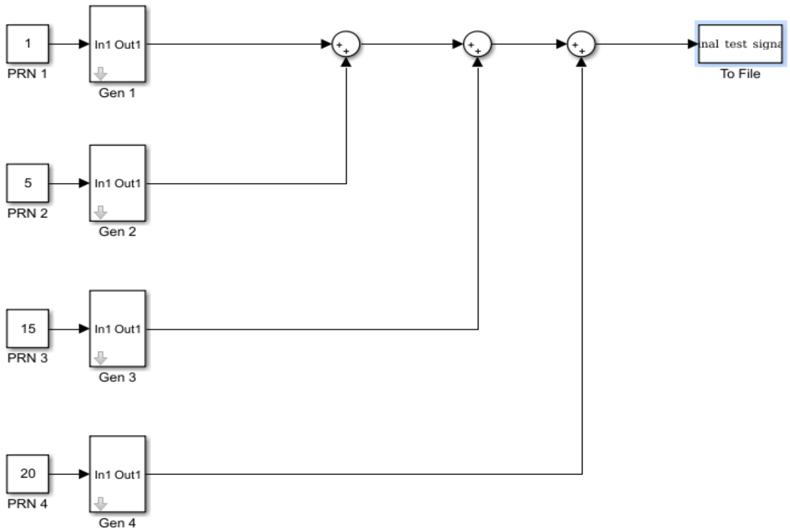

Figure 16: Multiple PRN Test Signal Generator

PRN codes of 1, 5, 20 were generated on the signal, as well as the PRN of 15 shifted half a code length. Since our final version of this testing stage will require a frequency sweep of 21 steps for each PRN, we adjust the acquisition stage oscillator to step up 500Hz every 1ms. The test signal was generated for 21ms in order to perform the 21 correlation checks on the signal. The 3 dimensional frequency sweep plots (centred around IF) for a few test PRNs are shown in the following figures.

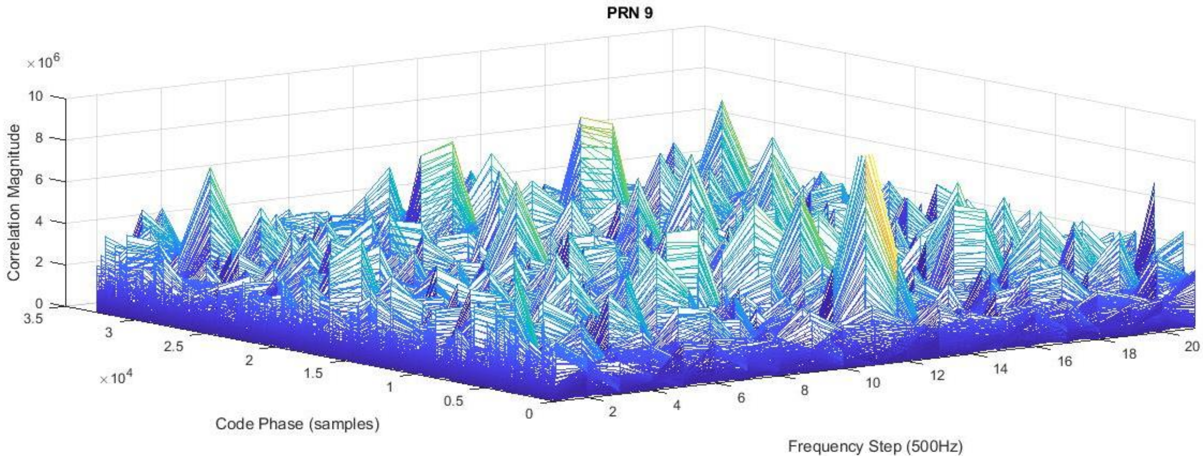

Figure 17: Frequency Search Test Result PRN 9

For PRN 9, which is not visible on our signal, we can see that there is no distinct peak at any frequency step. Any peak that does occur is on the order of magnitude of 1x10^6; this confirms that the acquisition stage only recognizes visible PRN codes.

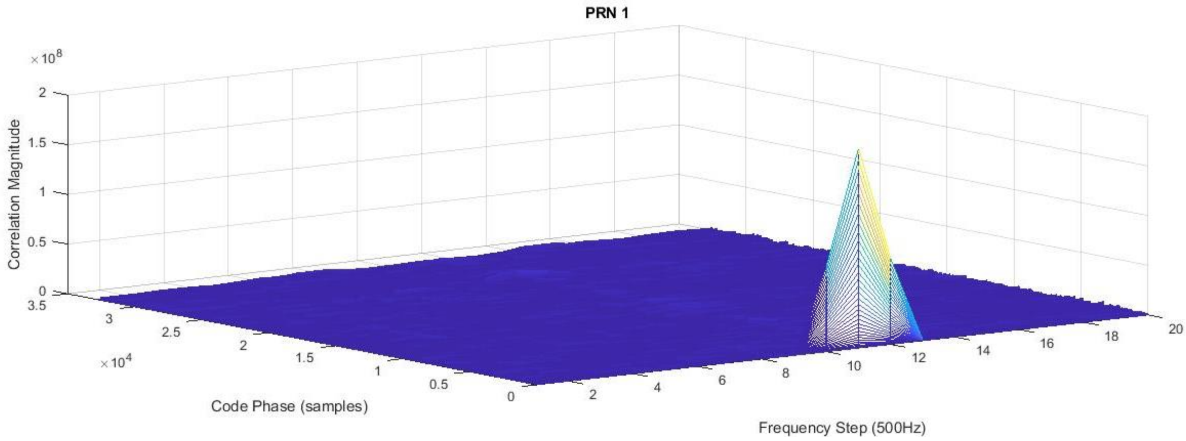

Figure 18: Frequency Search Test Result PRN 1

For PRN 1, it can be seen that a distinct correlation peak occurs in the middle of the frequency sweep steps (at IF), with a magnitude on the order of 1x10^8.

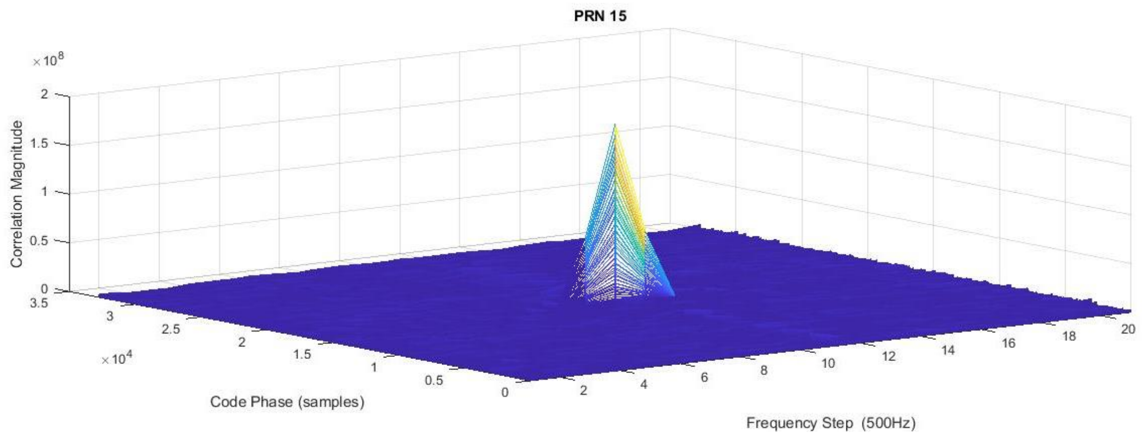

Figure 19: Frequency Space Search Test Result PRN 15

For PRN 15, it can be seen that a distinct correlation peak occurs in the middle of the frequency sweep steps (at IF), at a code phase equal to half of the chips. This paired with the correlation magnitude confirms the correct correlation detection.

Using the results of this testing, we must now determine an appropriate threshold to set for the correlation. A magnitude that surpasses this threshold will tell us that a particular satellite is visible to the system. A final function must then be added which employs the use of this threshold to store the PRN number, code phase, and frequency of a visible satellite.

9. Plan For Next Semester

The next semester tasks primarily consist of:

- ❖ Finalizing, Testing and Comparing Acquisition Stage

- ➢ Implementing LUT for the local oscillator instead of generating signals each time
- ➢ Determining necessary visible satellite correlation threshold
- ➢ Implementing automated system that identifies 4 closest satellites and their parameters from aggregate signal
- ➢ Testing our acquisition and comparing with known results from resources
- ➢ Comparing our acquisition implementation to other methods

- ❖ Implementation and testing of Tracking System
    - ➢ Implementing the code tracking loop
    - ➢ Implementing carrier tracking loop
    - ➢ Putting together tracking top level system and combining with acquisition
    - ➢ Testing tracking and comparing implementation with given resources

If time-permitting we would begin extracting the navigation data from the signal using the parameters calculated in tracking and decode it as best as possible to retrieve actual psuedoranges and position information. However, the tracking phase will be the main focus of the project and the semester. We will ensure that the tracking and acquisition phase are working and are implemented in such a way that future students can build on the project to finalize the actual receiver position calculations. We will divide the work such that different necessary sections are done and tested by each member and we will work together to combine the sections and test the full system.

Below is our expected completion date timeline for the next semester:

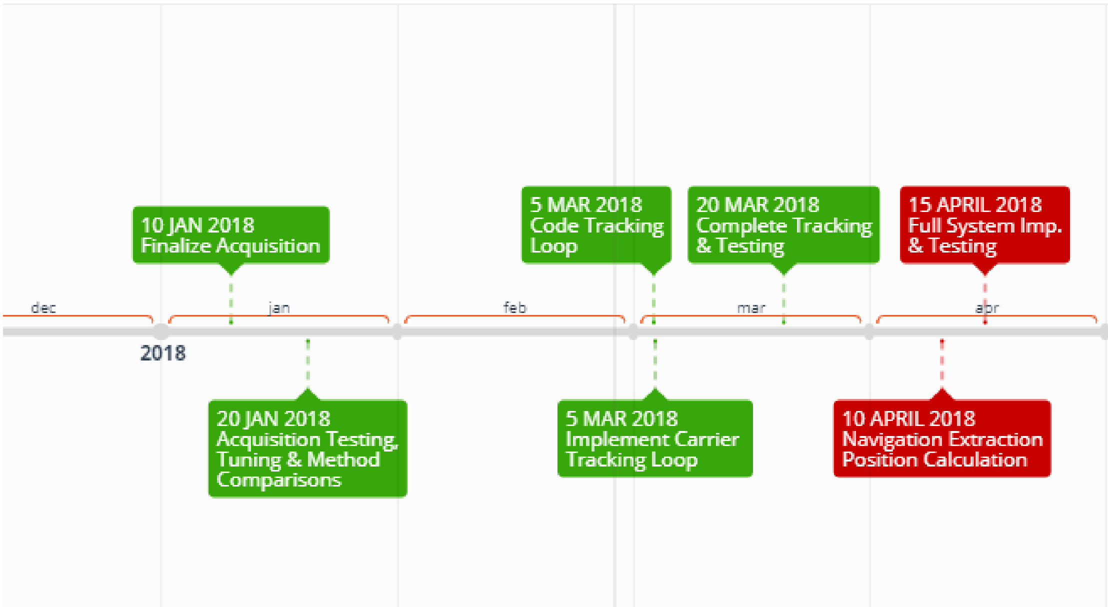

Figure 20: Timeline for Next Semester

10. Impact on Society and the Environment

Due to its nature, there are no real impacts or costs to consider in our project. However we can say that our project benefits other students and GPS enthusiasts by allowing them to analyse and build on our implementation easily in Simulink, displaying that complex electrical systems can be more readily understood and implemented in graphical programming environments such as Simulink.

11. Teamwork

We started off working on the project together as we were both new to the world of GPS and Simulink. After we have a solid understanding of the system and how to proceed we split the necessary sections of the acquisition phase, for example one of us had the task of designing the C/A code generator and implementing that while the other worked on the Local Oscillator. We worked on it individually at times and together at others, collaboration occured by testing together, combining systems, sharing resources, explaining reasoning and helping out with problem solving when 1 member needed it.

Problems arose infrequently but were mainly due to temporary unavailabilities, which is expected and is usually solved by one person taking on a little extra bit of the tasks that period and the other person making up for it the following week. Another problem that was differing or unclear understanding on some of the general concepts, this was always resolved (and will continue to be) by explaining our understanding to one another and coming to a mutual conclusion.

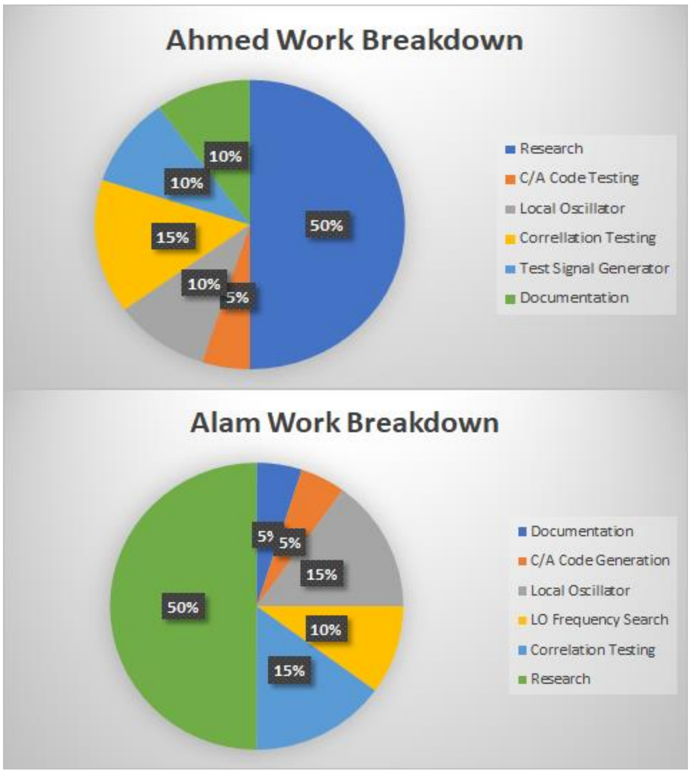

Figure 21: Work Allocation Chart for Current Semester

12. Conclusion

Alot was accomplished this semester, alot of research and studying was done to understand the nature of SatNav signals and GPS and GPS receiver functionalities. Experience in analysing and implementing such systems was gained, as well as the invaluable Simulink experience. We learned how GPS works and the processes a receiver must do to actually work out its position. In particular we learned a lot of signals and mathematical related concepts such as the value of correlation, spreading techniques and FT considerations. In particular we have a fundamental understanding of the Acquisition phase and its purpose in relation to the other stages and actual receiver positioning.

The next steps will be to finalize acquisition and move on to the tracking phase. Insight was gained in signal and communication systems as well as teamwork, project management, how to do effective research and implement that research into the project and accurately debug and identify issues.

13. References

[1]G. Hamza, A. Zekry and I. Motawie, "Implementation of a complete GPS receiver using simulink", *IEEE Circuits and Systems Magazine*, vol. 9, no. 4, pp. 43-51, 2009.

[2]J. Betz, *Engineering satellite-based navigation and timing*. .

[3]K. Borre, *Software-defined GPS and Galileo receiver*. Boston: Birkhhaüser, 2007.

[4]J. Tsui, Fundamentals of global positioning system receivers. Hoboken: Wiley, 2005.