A Multiple-Antenna Software GPS Signal Simulator for Rapid Testing of Interference Mitigation Techniques

by

Russell Powell

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 10, 2016

Keywords: GPS, CRPA, USRP, Software-Defined Radio, Intermediate Frequency, Pseudorange

Approved by

David M. Bevly, Chair - Professor of Mechanical Engineering
Stanley J. Reeves - Professor of Electrical and Computer Engineering
David A. Cicci - Professor of Aerospace Engineering

Abstract

This thesis details the design of a software GPS signal simulator and illustrates how it can be used to support rapid testing of GPS interference mitigation research. The MATLAB-based signal simulator is capable of generating digital GPS signals at intermediate frequencies (IF) for multiple-element, controlled-radiation-pattern-antenna (CRPA) configurations in jamming environments. The simulator was developed to support novel GPS interference mitigation research conducted in the GPS and Vehicle Dynamics Laboratory at Auburn University. The digital signals can be analyzed with software acquisition, tracking, and positioning techniques or can be converted to analog signals at the GPS radio frequencies and played to multiple receivers via cabling using the Universal Software Radio Peripheral (USRP) software-defined-radio (SDR) platform.

An IF GPS signal for a single antenna generated by the software simulator was played-back and recorded simultaneously with USRPs and compared to an IF signal for the same scenario generated by a hardware GPS simulator that was also recorded with a USRP. Position, velocity, pseudorange, Doppler frequency, and carrier-to-noise ratio measurements calculated by a software receiver from the simulated IF signal, the played-back and recorded IF signal, and the signal generated by the hardware simulator are compared to evaluate the baseline performance of the software simulator. Results of this experiment illustrate that a software simulator paired with the USRP is comparable in performance to the hardware simulator. Additionally, a dynamic scenario generated by the software simulator was played-back and recorded with USRPs and also played to a hardware GPS receiver. Position results calculated by a software receiver from the simulated and played-back and recorded signals were compared to results calculated by the hardware receiver to further evaluate the

performance of the software simulator. These results illustrate that the software simulator is capable of generating GPS signals for dynamic trajectories.

To illustrate the capabilities provided by a software-based GPS simulator, signals along with three types of simulated jammers were generated and are illustrated in the frequency, time, and histogram domains . Each interference signal was played-back and recorded with USRPs, and results are provided to illustrate the steps that must be taken to capture the effects of simulated jammers in a USRP playback or record. Additionally, a 4-element CRPA interference simulation was generated by the software simulator and analyzed with a software receiver that was modified to include several interference mitigation techniques. Results provided by the modified software receiver illustrate that a software GPS simulator is a powerful tool for developing GPS interference mitigation techniques. Lastly, the simulated CRPA signals were played-back and recorded simultaneously with multiple USRPs to examine the feasibility of using relatively inexpensive SDRs to simulate multiple-antenna GPS scenarios. Results from this test illustrate that a multiple-USRP setup is a useful tool for testing robust interference mitigation techniques.

Acknowledgments

First, I would like to thank my parents and family for being wonderful examples for me to follow in life and for teaching me the value of hard work! I would like to thank my beautiful fiancé Emily for supporting me in every way throughout my graduate studies! You have taught me the true value of strength! I am so proud of what you have accomplished and I'm so excited for our future together! I would like to thank Dr. David Bevly for giving me a chance in graduate school! I gave it my best shot and I cannot thank you enough for the opportunities you have provided me moving forward as I start my career. I would like to thank Scott Martin and Josh Starling for always being open to provide ideas and suggestions related to my research. I would like to thank members of the GAVLAB for many fun times and laughs at group lunches and in lab conversations. Last, and absolutely not least, I would like to thank Andrew Eick, Dan Pierce, Brently Nelson, Josh Starling, Stephen Geiger, Tyler Sherer, Cody Salmon, Sarah Preston, and many others who were always open to play a game (or 3) of ping pong (especially PLPP!). I don't think I could have kept my sanity in grad-school without it!

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As dependence on Global Positioning System (GPS) signals for civilian and military applications increases, the more they become at risk to possible interference attacks. The GPS signals are received at Earth at a very low power which makes them susceptible to both intentional and non-intentional interference. Luckily, there has been extensive research conducted on interference detection and mitigation techniques to combat this issue. In order to utilize the full potential of current GPS signals and future signal designs in an interference environment that is constantly changing, research on interference mitigation techniques must continue to evolve. This calls for intuitive but flexible simulation and testing methods to support novel research on interference mitigation techniques. Testing can be conducted with high-fidelity, multiple-element hardware GPS signal simulation platforms but these are often too expensive for research in some environments. Therefore, in order for interference mitigation research to be maximized at all levels, there is need for the continuing development of reliable, inexpensive, and re-configurable GPS simulation and testing methods.

1.2 Prior Work

Since GPS signals powerful navigation tools, a significant amount of research has been conducted on novel interference detection and mitigation techniques. In order to test new technologies, simulation methods are needed. Becuase state-of-the-art, high-fidelity hardware simulators capable of generating multiple-antenna GPS inteference scenarios are very

expensive, importance has been placed on inexpensive but relaiable GPS simulation techniques. In [10] a software simulator developed in MATLAB was used to test a Simulink-based software receiver developed to test next generation receiver technologies. Similarly, in [13] and [33], software GPS simulators were developed to test emerging software receiver technologies. In [5] and [6], the NAVSYS Corporation describes a MATLAB-based software simulator and an open-source Universal Software Radio Peripheral (USRP)-based record and playback simulation architecture. In [12], a simple IF GPS signal simulator was described and the errors induced by the USRP to the playback and record of live-sky and simulated signals were examined.

In [4], NAVSYS Corportation describes a hybrid USRP-based simulation setup that is used to simulate GPS signals or to record and playback live-sky, multiple-antenna interference scenarios for simulation purposes. In [26], a design for multi-antenna space-born GPS receiver capable of operating in orbits above the GPS constellation was proposed. In this environment, above the GPS satellites, the GPS signals availible for navigation are sparse, at different power levels, and susceptible to un-intentional interference. IF GPS signals along with near-far interference (mixed GPS signals at different power levels) and multi-path were generated for a multi-antenna configuration to test the beamforming techniques in a software version of the proposed revceiver. In [22], the pseudoranges, Doppler frequencies, ranging codes, and navigation data for a user scenario were extracted from a hardware simulator and used to simulate GPS signals for multi-antenna configurations in a software simulation.

The GPS and Vehicle Dynamics Laboratory (GAVLAB) at Auburn University and many other research groups are currently interested in GPS interference detection and mitigation-related research. In [7] a process called successive interference cancellation (SIC) was used to mitigate a spoofing attack. In SIC, the receiver tracking loops are used to create a replica of the attacking signal which is subtracted from the raw signal allowing for recovery of the authentic signal. Currently, importance is being placed on research that examines how interference mitigation techniques that utilize CRPA configurations affect the carrier phase

measurement and carrier phase positioning in dynamic jamming environments [31]. In order to evaluate these effects, GPS data with interference is needed. Live-sky testing events are a valuable tool that cannot be totally replicated but they are expensive to attend and the user is allowed little or no control over the test events. A multiple-element GPS simulator capable of generating interference waveforms provides a researcher control the simulation but these are also very expensive. The software GPS simulator described in this thesis was designed as a cost-effective option that gives the researcher complete control of the simulation environment for rapid design and development of interference mitigation techniques. The simulator can generate digital IF GPS signals with interference to test mitigation techniques in the software domain. The digital IF signals can be converted to analog signals at the GPS radio frequencies (RF) using the relatively inexpensive USRP SDR platform to test hardware technologies. Importance has also been placed on research that quantifies the negative effects induced into simulation environments by the relatively inexpensive USRPs. In [19] errors added by the USRP in the record and playback of live-sky signals are analyzed.

1.3 Contributions

This thesis further quantifies development and performance of a software-based GPS signal simulator. Results obtained from simulated signals are compared to ones generated by a hardware simulator. Additionally, results obtained from a playback and record of the simulated signals using the USRP SDR are compared to results obtained from the original simulated signal and the signals generated by the hardware simulator. This analysis further quantifies the performance of the USRP SDR and expends on their capability to simulate GPS scenarios. This thesis also details the simulation of GPS signals along with interference for multi-antenna configurations. The contributions this thesis makes to the research field are as follows:

- Illustrates direct generation of GPS signals with several types of jamming for a CRPA configuration

- Provides analysis of USRP playback and record of simulated GPS signals and jammers with emphasis on capturing the signal environment
- Examines testing of multiple types of GPS interference mitigation techniques using a completely software-based, multiple-antenna GPS signal simulator
- Examines testing of multiple types of interference mitigation techniques using simulated, multiple-antenna GPS signals that were played-back and recorded using USRPs

1.4 Thesis Outline

In Chapter 2, the basics of GPS navigation are introduced with emphasis on how they pertain to GPS signal simulation. In Chpater 3, the design of the software simulator is detailed and the USRP playback and record methodology and setup are illustrated. In Chapter 4, baseline results provided by the software simulator and USRP platforms are analyzed. In Chapter 5, jammer models and interference mitigation techniques are introduced, interference simulations are described, and results are analyzed. Conclusions and future-work are discussed in Chapter 6 and an appendix is provided in Chapter A and are followed by a list of references and an Appendix containing relevant information.

Chapter 2

Basics of GPS Navigation

2.1 Introduction

The Global Positioning System (GPS) has revolutionized modern navigation techniques. Originally, it was designed to provide accurate position and velocity measurements to military operations, but it has exploded into the civilian realm in recent years. This chapter will introduce the basic concepts behind this powerful navigation system with focus on how they pertain to simulation of GPS signals.

2.2 Coordinate Systems

Three coordinate systems are referenced in this thesis: World Geodetic System 1984 (WGS-84), East-North-Up (ENU), and a local body frame. The WGS-84 coordinate system is an Earth-centered-Earth-fixed (ECEF) global system that rotates with the Earth. The origin of the WGS-84 is the Earth's center of mass, the z-axis passes through the Earth's conventional terrestrial pole (CTP), the x-axis lies in the equatorial plane and passes through the intersection of the CTP and the Mean Greenwich meridian, and the y-axis lies in the equatorial plane and completes the right handed coordinate system [15]. A point in the WGS-84 system can be represented with three position coordinates x, y, and z, or 2 angles latitude ϕ and longitude λ and a measure of height h. GPS receivers calculate position and velocity in the WGS-84 and the GPS satellites broadcast ephemerides that allow receivers to calculate the positions of the satellites in the WGS-84. The ENU coordinate system is a local frame with an origin fixed to a reference WGS-84 position. The positive x-axis E points towards geodetic east, the positive y-axis N points towards geodetic north, and the positive

z-axis U points up with respect to the reference position. The ENU frame is commonly referred to as a navigation frame and will be denoted with subscript n [17]. The body frame is a local system used to describe the motion of a moving object and is denoted with subscript b. The origin is fixed to some point on the body (usually the center of gravity), the positive x-axis x_b points forward through the origin in the direction of motion, the positive y-axis y_b points starboard out the right side of the body, and the positive z-axis z_b points upward [17]. The navigation and body frames are used to describe and simulate the receiver trajectories detailed in Section 3.2.1.

2.3 GPS Position Solution

The GPS position calculation is based on a process called trilateration. In trilateration, if the position of an object and the distance between the object and the user are known at the same instant of time, the user's position can be calculated in the same coordinate system [2]. In [15] a helpful 2-dimensional example is presented. Imagine transmitting stations on Earth broadcasting arbitrary signals simultaneously. The user knows the precise 3-D position of each station, the instant of time each signal was transmitted, and the speed of the propagating signals a-priori. The user can calculate when each signal was received providing the signal transit time, which is the difference between the received and transmit times. The transit time can be converted to a distance between the receiver and station using the signal propagation speed. The user has to be located somewhere on a circle with radius r_k with the transmitter at the center. A range equation can be developed using the Pythagorean theorem and is given below in Equation (2.1)[15].

$$\sqrt{(x_{s_k} - x)^2 + (y_{s_k} - y)^2 + (z_{s_k} - z)^2} = r_k \tag{2.1}$$

If this equation can be formed for three transmitters, the user's 3-D position, x, y, and z, can be solved. This is the basic concept used in the GPS position calculation, except the

transmitters are the GPS satellites. Each satellite transmits a keenly designed signal that allows the receiver to calculate the position of each satellite, transmit time of each signal, and distance to each satellite at the same instant.

The preceding discussion assumed the user and transmitter were synchronized to the same clock, which is not the case in the GPS scenario. Each GPS satellite uses it's own clock for transmitting signals and each receiver uses a unique clock for sampling and processing. This can introduce massive errors to the user-satellite ranges formed in Equation (2.1) because errors caused by the time offsets are compounded due to the signals traveling at the speed of light ($\approx 3 \times 10^8 \frac{m}{s}$). To account for this, the GPS control station maintains a true system time called GPS time. The control station monitors the deviations of each satellite clock from GPS time and provides corrections which are modulated on the transmitted GPS signals. These are used by the receiver to correct error caused by the satellite clock on the ranges in Equation (2.1). Because receiver clocks tend to be less stable than the satellite and control station clocks due to size, weight, power, and cost (SWAP-C) the range error caused by the receiver clock is left as an unknown bias to be solved in the postion calculation [2]. Equation (2.1) is modified to include receiver clock bias dt_r and the satellite clock bias dt_{s_k}. This is shown below in Equation (2.2).

$$\sqrt{(x_{s_k} - x)^2 + (y_{s_k} - y)^2 + (z_{s_k} - z)^2} + cdt_r = r_k + cdt_{s_k} \tag{2.2}$$

It should be noted that range in meters (m) is equal to the time difference in seconds (s) multiplied by the speed of light c in meters per second (m/s). This now leaves four unknowns to be solved which requires four distinct equations. This means that four GPS satellites must be available to the user in order to solve for 3-D position and the receiver clock bias. Usually, and hopefully, more than four satellites are available to the user. This presents an over-determined system which can be solved using a variety of estimation techniques. Generally, a recursive least squares approach is used which is outlined in [15] and [2] and many other sources.

2.4 Pseudoranges

In Section 2.3 the ranges to each GPS satellite were considered to be uncorrupted by external error sources. In reality, the signal propagation paths are obstructed by many different sources that affect the signal propagation time. Atmospheric and multipath errors are the largest external culprits that affect the GPS signals. Fortunately, many of these errors can be accounted for in mathematical models and accounted for in the range equations [15]. The corrupted ranges are called pseudoranges and a high level model is provided in Equation (2.3).

$$\rho_k = r_k + cdt_{s_k} + cdt_r + r_{EXTERNAL} \tag{2.3}$$

The pseudorange ρ for satellite k is equal to the true user-satellite range r_k plus errors caused by the satellite and receiver clocks dt_{s_k} and cdt_r, and any external range errors that can be modeled $r_{EXTERNAL}$. The pseudorange is the basic measurement required to simulate GPS signals and is discussed further in Section 3.2.4. Range errors that can be modeled in the software simulator are discussed in Section 3.2.5.

2.5 GPS Signals

The GPS signals consist of a RF carrier modulated by a ranging code and a navigation data message. The software simulator detailed in this thesis is capable of generating L1 GPS signals with a carrier frequency of $f_{L1} = 1575.42$ MHz. The L1 signal consists of a RF carrier modulated by a ranging code, called a CA code, that can be decoded by all receivers and another RF carrier modulated by an encrypted ranging code, called a P(Y) code, that can only be used for military purposes. The L1 GPS signal model in the time domain at the reciever with K satellites in view is provided by Equation (2.4) [15].

$$S_{L1}(t) = \sum_{k=1}^{K} \sqrt{2P_{C_k}(t)} C_k(t) D_k(t) cos(2\pi f_{L1} t + \theta_{L1}) + \sum_{k=1}^{K} \sqrt{2P_{P_k}(t)} P_k(Y)(t) D_k(t) sin(2\pi f_{L1} t + \theta_{L1}) \tag{2.4}$$

In the above equation, $P_{C_k}(t)$ and $P_{P_k}(t)$ are the L1 civilian and military signal powers for satellite k; $C_k(t)$ and $P_k(Y)(t)$ are the CA and P(Y) ranging codes; and $D_k(t)$ is the navigation data message. The ranging codes are a set of pseudo-random noise (PRN) binary codes with special properties that allow the receiver to acquire and track GPS satellite signals and calculate the vital pseuduorange measurements. Each satellite transmits a unique CA and P(Y) code with chipping frequencies of $f_{CA} = 1.023$ MHz and $f_P = 10.23$ MHz respectively. The properties that allow receiver to acquire and track the CA code are discussed further in Section 2.7 and generation of the CA code in the signal simulation is discussed in Section 3.2.6. The navigation data is a binary message, consisting of ones and zeros, transmitted at 50 Hz that contains satellite orbit information, clock corrections, health data, and other various information needed by the user. The navigation data allows the receiver to determine when each availible GPS signal was transmitted and to calculate the satellite positions at the same instant. The navigation data structure is discussed further in Section 3.2.6. The binary-low values in the ranging codes and navigation message are converted from 0 to -1 before modulating onto the signal. This process is called binary phase shift keying (BPSK) and essentially produces 180° phase shifts by amplitude modulations of 1 and -1 [15]. Also, the CA code repeats twenty times between sucessive navigation data bits and each satellite generates the start of a CA code sequence at the same time as a new navigation data bit [32]. Current and future GPS signal structures are discussed extensively in the GPS Interface Control Document (GPS ICD) [25] and in many other academic references similar to [15] and [2].

2.6 GPS Satellites

Originally, the GPS satellite constellation was designed to place 24 active satellites in six orbital planes with four satellites in each plane spaced to achieve optimum satellite geometry over the entire Earth at any instant. There are now 32-plus satellites with 32 actively transmitting valid GPS signals. Users can expect to see 6-8 satellites at most locations on Earth without external obstructions [15]. The satellite accelerations $\ddot{\bar{r}}$ can be modeled using Newton's law of universal gravitation enhanced to account for accelerations caused by external perturbations. This is illustrated by Equation (2.5).

$$\ddot{\bar{r}}_s = -\frac{GM}{r_s^3}\bar{r}_s + F(\bar{r}_s, \dot{\bar{r}}_s, t) \tag{2.5}$$

The first term of Equation (2.5) $\left[-\frac{GM}{r_s^3}\bar{r}_s \right]$ is known as the two-body equation of motion and descirbes the acceleration of an orbiting body with no external perturbations. The second term $F(\bar{r}_s, \dot{\bar{r}}_s, t)$ accounts for the accelerations of the body caused by various perturbations. The main external perturbations on GPS satellites are caused by forces from the Non-central gravitational field of Earth, other heavenly bodies (especially the Sun and the Moon), and solar radiation pressure [15]. More information about these perturbing forces and their interactions with orbiting bodies can be found in [24]. The two-body orbit can be described by six parameters, 3-D position and velocity $\bar{r}_s$ and $\dot{\bar{r}}_s$, or in terms of geometrical Keplerian orbital elements a, e, i, Ω, ω, and ν. The semi-major axis a and eccentricity e define the size and shape of the orbit. The inclination i, right ascension of the ascending node Ω, and argument of perigee ω are angles that define the orbit's orientation in space. The true anomaly ν is an angle that classifies the position of the satellite on the orbit described by the preceding orbital elements [15]. The GPS control station maintains and predicts the orbit of each satellite and calculates a quasi-Keplerian set of orbital elements called an ephemeris. The ephemeris consists of the six nominal orbital paramters and nine terms designed to account for the changes of the orbit over a specifed time due to perturbations [15]. These are

uploaded to the GPS satellites by the control station. The satellite transmits the ephemeris in the navigation message and once the user receives and decodes the complete set, position of the satellites is calculated at the desired instant. The satellite position calculation is detailed in Section 3.2.3.

2.7 GPS Receivers

The typical GPS receive chain consists of two stages: a hardware front-end and software signal processing. A high-level illustration of a generic GPS receive chain is shown below in Figure 2.1. The antenna converts electromagnetic fields into electrical voltages and currents which consist of the desired RF GPS signals along with interference and thermal noise [2]. The RF signal is then amplified and down-converted to an intermediate frequency (IF) signal. After down-conversion, the IF signal is band-limited and converted to digital samples with an analog-to-digital converter (ADC). The digital samples are then passed to the signal processing stage of the receiver.

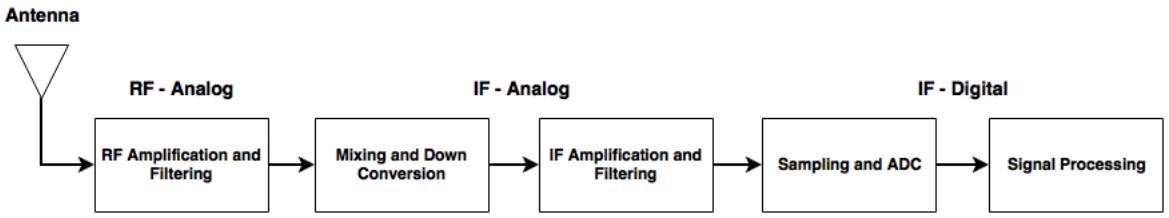

Figure 2.1: Generic GPS Receive Chain

The signal processing stage consists of two steps: acquisition, and tracking. In acquisition, the receiver uses the cross-correlation and auto-correlation properties of the ranging codes to determine which satellites are in view and the code phase and carrier frequency of each. The PRN codes are designed to be mostly uncorrelated with other PRN codes and mostly uncorrelated with themselves in time except for zero delay. The incoming GPS signals are first multiplied by locally generated PRN code with varied code phase. The signals are then mixed with two locally generated carrier signals, one that is in phase and one that

is phase shifted 90^o, which are typically varied in frquency steps of 500 Hz [3]. This multiplication and mixing provides an in-phase I and quadrature signal Q that are integrated, sqaured, added, and correlated. If the correlation value exceeds a predefined metric, the signal from the corresponding satellite is aquired and the code phase and carrier frequency estimates are passed to the tracking stage [3]. The tracking stage refines the initial estimates of the code phase and carrier frequency, keeps track of them over time, and demodulates the navigation data message [3]. A delay lock loop (DLL) is used to track the code phase and a phase lock loop (PLL) is used to track the carrier frequency and phase. The DLL and PLL are esentially control systems that drive the error in the code phase, carrier frequency, and carrier phase between measured and replica signals to a minimum value [15]. If the signals are successfully tracked, the navigation data can be extracted from the incoming signal providing the transmit time and ephemeris allowing the receiver to calculate the corresponding pseudoranges needed for the receiver position calculation. Extensive dicussions on GPS receiver front-end design, acqusitition, tracking, and data demodulation can be found in [2], [3], and [15].

2.8 GPS Interference

When GPS signals reach a receiver they are at a power of about -160 dB which is below the local thermal noise floor. The signal properties and keenly designed acquisition and tracking algorithms allow users to navigate using these weak signals. GPS interference can be defined as any electromagnetic signal that is interacting with the GPS signals and deteriorating the pseudorange estimation [14]. This can include error sources that impede pseudorange estimation by inducing a delay on the signals like multipath and atmospheric effects, bleed-over from other communications signals into the GPS band, accidental transmission of signals in or around the GPS band, or intentional generation of signals designed to interfere with the receiver acquisition and tracking process by raising the noise floor to mask the GPS signals [14]. This intentional generation of interference is called jamming. Jammer

signals are designed to increase the noise power around the receiver to deteriorate or even cease acquisition and tracking in target receivers. The GPS signal at a receiver antenna in the presence of interference is modeled by Equation (2.6).

$$S(t) = S_{L1}(t) + I(t) + N(t) \tag{2.6}$$

where $S_{L1_k}(t)$ is the L1 GPS signal described by Equation (2.4), $I(t)$ is the RF interference, and $N(t)$ is the thermal noise. Receivers naturally account for some interference because of the clever design of the signal and processing techniques, but if the power of the interference is significantly higher than that of the received GPS signal and thermal noise, the receiver acquisition, tracking, and positioning can be severely deteriorated. The GPS signal power, signal-to-noise ratio, and jammer-to-noise ratio are discussed further in Sections 3.3.2 and 5.3 respectively. Specially designed antennas and signal processing techniques can be used to detect and mitigate jammer attacks. Examples of these are discussed further in section 5.4.

Chapter 3

Software Simulator Development

3.1 Introduction

The software simulator detailed in thesis is designed to generate GPS signals that can be acquired and tracked by the methods described in Section 2.7. The signals are generated to simulate a desired user scenario including position, velocity, time, interference, antenna configuration, and other parameters. This is executed by generating the pseudoranges and corresponding ranging codes that convey the desired user position and velocity to the GPS acquisition and tracking loops, modulating the GPS signal with a navigation data message that allows the receiver to calculate transmit time and the satellite constellation, and generating noise, interferences, and other parameters desired for the simulation. An overview of the software simulator is provided in Figure 3.1. User settings including time, trajectory, and antenna configuration are used to generate the user-satellite ranges in the scenario simulation. These ranges are used to generate the ranging code and carrier signal which is modulated with a simulated navigation message in the signal simulation. The output of the signal simulation is a sampled GPS signal at an intermediate frequency. This IF signal can be analyzed directly by a software receiver or can be up-converted to an analog signal at the GPS RF with a software defined radio and analyzed by hardware GPS receivers via cabling. The user scenario simulation is detailed in Section 3.2 and the signal simulation is detailed in Section 3.3.

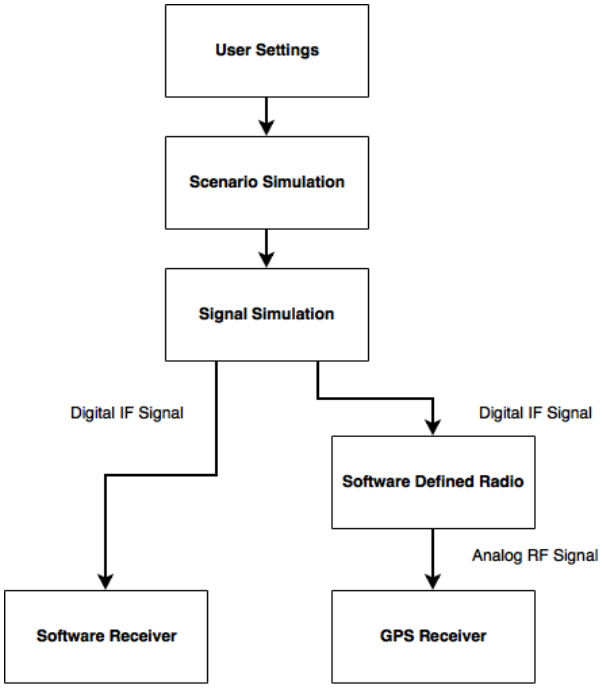

Figure 3.1: Software GPS Simulator Overview

3.2 Scenario Simulation

The scenario simulation utilizes the user-desired settings to generate the parameters required to simulate GPS signals for a specified time and trajectory. The user settings include time, trajectory, antenna configuration, and toggle commands. The toggle commands are used to turn on or off different aspects of the simulator like atmospheric errors, noise, filters, ADC, interference, etc. As seen on the left side of Figure 3.2, the GPS satellite constellation is generated over the simulation time and since the user trajectory is given a-priori, the corresponding user-satellite ranges can be calculated at each epoch. The right side of Figure 3.2 illustrates the functional steps of the scenario simulation. In the initilization step, the user trajectory is modeled and converted to the WGS-84 frame and the ephemerides corresponding to the user time are downloaded from the NASA CDDIS website [8]. Using the ephemerides, the position of all 32 GPS satellites are calculated and the satellites visible to the user in the simulation are determined. In the last step of initialization, the initial transmit time of each visible satellite is calculated using an iterative process that is detailed

in Section 3.2.2. After the initial transmit times are calculated, the true position and clock correction term for each visible satellite are calculated at each epoch of the simulation. The equations used to calculate the satellite positions and clock corrections are detailed in Section 3.2.3. Next, the true user-satellite ranges are calculated and range errors are modeled and incorporated to the true ranges via the pseudorange model that is detailed in Section 3.2.4. The clock-based and atmospheric range error models are described in Section 3.2.5. Other parameters of interest to the user, like signal power or Doppler frequency, can also be generated in the scenario simulation. The calculations in the scenario simulation are executed at a frequency of $f_{scenario}$ which was set to 100 Hz for the simulations detailed in this thesis. The pseudoranges are sent to the signal simulation and used to generate the code and carrier of the simulated GPS signals. After the scenario simulation, in an intermediate step, the CA code and navigation message are generated. The methods used to generate the CA code and navigation data are detailed in Section 3.2.6.

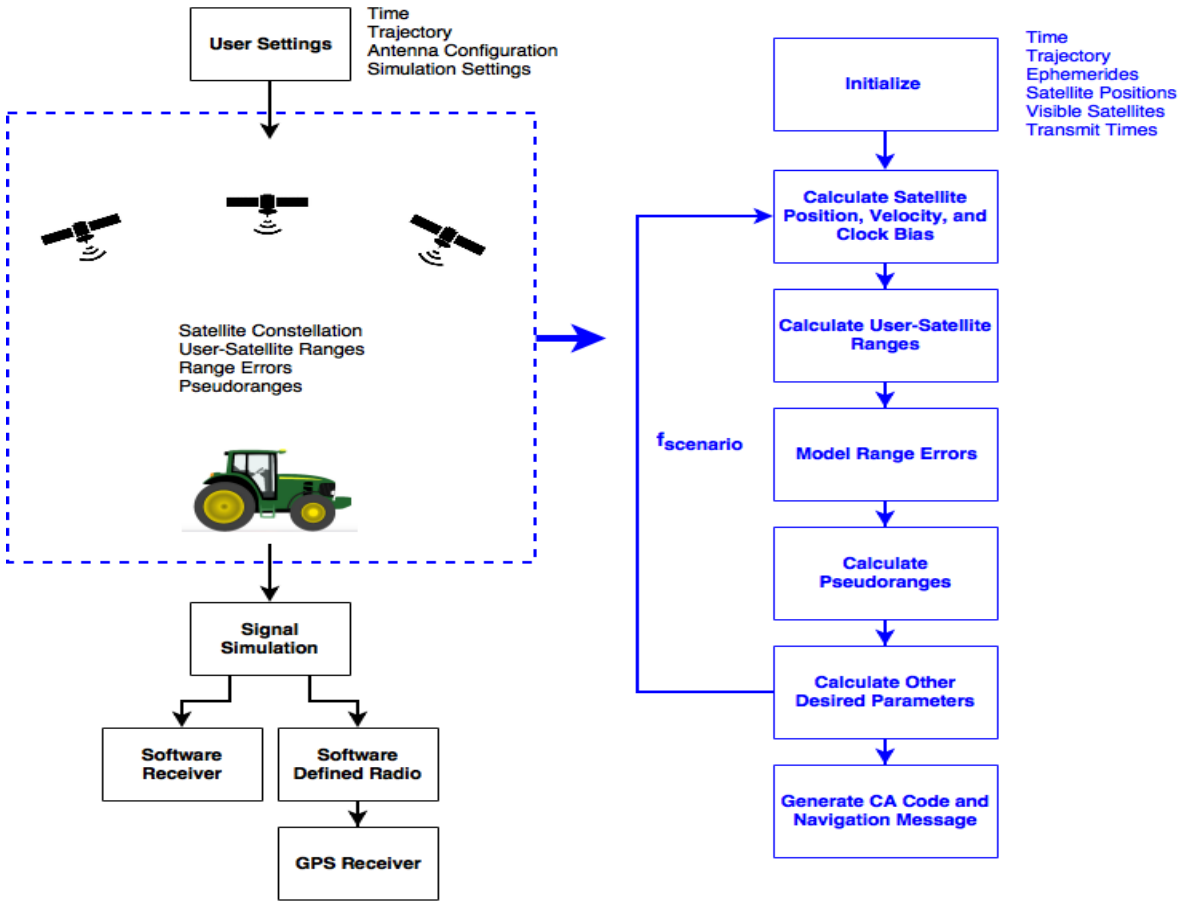

Figure 3.2: Scenario Simulation Overview

3.2.1 Trajectory Simulation

In the software simulator, it is assumed that the user tajectory to be simulated is known a-prori. The user positions x, y, and z and velocities v_x, v_y, and v_z in the WGS-84 coordinate system are provided to the simulation by the user. Since the simulator detailed in this thesis is designed to test and develop algorithms that utilize multiple antenna configurations to mitigate GPS interference, the positions of multiple antennas must be simulated. To do this, the user can define antenna configuration unit vectors with respect to the reference antenna in the local body frame. In this thesis, the reference antenna unit vector $\bar{e}_1^b$ is set to $\big[0, 0, 0\big]$. This essentially means that the a-prori positions input to the simulation are of reference antenna and the other antennas are defined with respect to the reference. This can

be modified if necessary. The unit vectors of the remaining antennas $\bar{e}_{1a}^{b}$ are set to represent the desired antenna configuration in the local body frame. First, the WGS-84 positions and velocities of the reference antenna are converted to the ENU navigation coordinate system. An overhead view of a 4-element antenna configuration in the ENU navigation frame is illustrated in Figure 3.3.

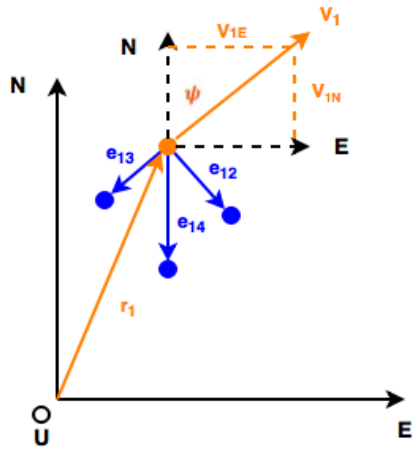

Figure 3.3: 4-antenna configuration in ENU Navigation Frame

The position and velocity of the reference element in the navigation frame $\bar{r}_1^n$ and $\bar{v}_1^n$ are the converted WGS-84 positions and velocities. The position of the other antenna elements in the navigation frame are found using Equation (3.1).

$$\bar{r}_a^n = \bar{r}_1^n + \bar{r}_{1a}^n \tag{3.1}$$

where $\bar{r}_{1a}^n$ is the position vector from the reference element to element a in the ENU navigation frame. This is found by rotating the local body frame to the navigation frame. In Figure 3.3, the body frame is offset from the navigation frame by the angle ψ which is called the yaw angle or heading [17]. The yaw angle represents a rotation about the fixed z-axis. In this case, the navigation and body frame share the same z-axis. The position vector from the reference antenna element to element a is converted to the navigation frame using Equation (3.2).

$$\bar{r}_{1a}^{n} = \bar{R}_{\psi}\bar{e}_{1a}^{b} \tag{3.2}$$

where $\bar{R}_{\psi}$ is the transformation matrix from the local body frame to the navigation frame for the rotation about the z-axis and is formulated using Equation (3.3) [16].

$$R_{\psi} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

The yaw angle ψ is found using Equation (3.4).

$$\psi = \tan^{-1}\frac{v_{1E}}{v_{1N}} + \delta \tag{3.4}$$

where a 4-quadrant inverse tangent function is used [17]. The angle δ in Equation (3.4) is an offset angle that allows the user to set the initial orientation of the antennas and to ensure the antennas are oriented in the desired direction in a dynamic trajectory. If the trajectory is static, the heading angle will be zero and the body frame unit vectors will be aligned with the ENU navigation frame axes. The user can set the offset angle to orient the antenna configuration to a desired direction in the navigation frame. It should be noted that Equation (3.4) assumes no sideslip or heading error for vehicle trajectories, but this can be modeled if necessary. After the positions of all antenna elements are calculated in the navigation frame, they are converted back to the WGS-84 and used in the scenario simulation. The trajectory simulation described in this section is a simple method that allows the user to orient an antenna configuration based only on the heading of the vehicle trajectory that is simulated. This is sufficient for simple simulation of antenna orientation with respect to a jammer location that ensures the simulated interference signals impede on the antennas in a desired manner. Higher fidelity trajectory models that include roll and pitch angles of the antenna element and vehicle can be implemented into the simulation if desired by the user.

3.2.2 Transmit Time

At the initial receiver time t_{r_i}, there are K satellites in view to the user, each with a different distance from the user. This means that each received signal was transmitted at a different time. This is illustrated on the left side of Figure 3.4. In the scenario simulation, the initial transmit time t_{t_k} of each satellite must be generated in order to calculate the satellite positions and the corresponding user-satellite ranges throughout the simulation. Because the Earth rotates between the signal transmit and receive time, an iterative process is used to calculate the initial transmit times [13].

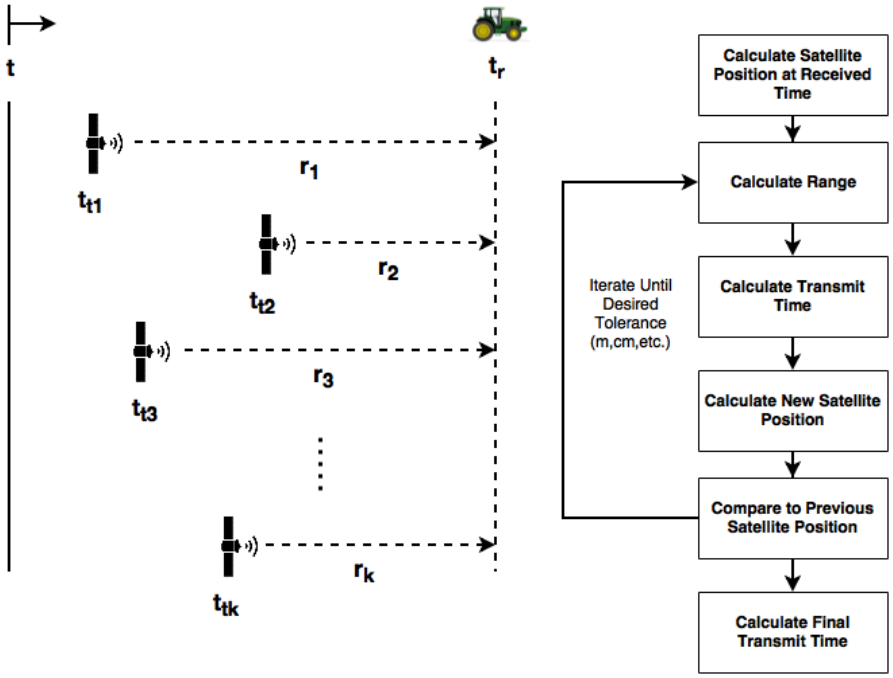

Figure 3.4: Transmit Time Calculation

Illustrated on the right side of Figure 3.4, given the position of a satellite $x_{s_{k1}}$, $y_{s_{k1}}$, and $z_{s_{k1}}$ and the user position x, y, and z at the initial receiver time t_r, the range between the user and the satellite at this instant r_{k1} is calculated using Equation (3.5).

$$r_{k1} = \sqrt{(x_{s_{k1}} - x)^2 + (y_{s_{k1}} - y)^2 + (z_{s_{k1}} - z)^2} \tag{3.5}$$

The corresponding transmit time $t_{t_{k1}}$ is calculated using Equation (3.6).

$$t_{t_{k1}} = t_{r_k} - \frac{r_{k1}}{c} \tag{3.6}$$

Next, new satellite positions $x_{s_{k2}}$, $y_{s_{k2}}$, and $z_{s_{k2}}$ are calculated using the transmit time $t_{t_{k1}}$ and a new range r_{k2} is calculated using Equation (3.7).

$$r_{k2} = \sqrt{(x_{s_{k2}} - x)^2 + (y_{s_{k2}} - y)^2 + (z_{s_{k2}} - z)^2} \tag{3.7}$$

The difference between the two ranges r_{k1} and r_{k2} is calculated using Equation (3.8).

$$\delta r_k = |r_{k2} - r_{k1}| \tag{3.8}$$

If a desired tolerance is met, the final transmit time for satellite k is calculated using Equation (3.9).

$$t_{t_k} = t_{r_k} - \frac{r_{k2}}{c} \tag{3.9}$$

If the tolerance is not met, the satellite positions $x_{s_{k1}}$, $y_{s_{k1}}$, and $z_{s_{k1}}$ are re-initialized to $x_{s_{k2}}$, $y_{s_{k2}}$, and $z_{s_{k2}}$ and the process is repeated until the tolerance is met. This is repeated for each visible satellite in the simulation. Successive transmit times in the scenario simulation are calculated by adding the simulation time step $\delta t_{scenario} = \frac{1}{f_{scenario}}$ to the previous transmit times.

3.2.3 Satellite Position and Clock Correction

After the initial transmit times are known, the corresponding satellite positions and clock corrections can be calculated using the downloaded ephemeris. The name and description of each ephemeris parameter is provided in Appendix A.1. The satellite position calculation is fully detailed in the GPS ICD [25] and a good translation is provided in [32]. First, the satellite's mean motion n_k is calculated using Equation (3.10)

$$n_k = \sqrt{\frac{\mu}{a_k^2}} + \triangle n_k \tag{3.10}$$

where μ is the Earth's gravitational constant ($\mu = 3.986005 \times 10^{14}\ m^3/s^2$), a_k is the satellite's semi-major axis ($a_k = \sqrt{a_k}^2$), and $\triangle n_k$ is the mean motion difference. Next, the transmit time t_{t_k} is corrected to account for the difference between the ephemeris epoch time t_{oe_k} and possible GPS week crossovers in Equation (3.11).

$$t_{t_k} = \begin{cases} t_{t_k} - 604,800 & \text{if } t_{t_k} - t_{oe_k} > \frac{604,800}{2} \\ t_{t_k} + 604,800 & \text{if } t_{t_k} - t_{oe_k} < \frac{-604,800}{2} \end{cases} \tag{3.11}$$

The mean anomaly M_k is found using Equation (3.12).

$$M_k = M_{0_k} + n(t_{t_k} - t_{oe_k}) \tag{3.12}$$

Where M_{0k} is the mean anomaly at the ephemeris reference time provided in the ephemeris. Next, the non-linear eccentric anomaly E is calculated using an iterative approach [32]. The initial eccentric anomaly E_{k_i} is set to the mean anomaly M_k. Successive eccentric anomaly values are calculated using Equation (3.13).

$$E_{k_{i+1}} = M_k + e_k \sin E_{k_i} \tag{3.13}$$

The difference between $E_{k_{i+1}}$ and E_{k_i} is calculated and if it is less than a pre-determined value, $E_{k_{i+1}}$ is set as the eccentric anomaly. Next, the satellite clock correction δt_{s_k} is calculated using Equation (3.14)

$$\delta t_{s_k} = a_{f0_k} + a_{f1_k}(t_{t_k} - t_{oc_k}) + a_{f2_k}(t_{t_k} - t_{oc_k}) + Fe_k\sqrt{a_k}\sin E_k - T_{GD_k} \tag{3.14}$$

where a_{f0_k}, a_{f1_k}, and a_{f2_k} are the clock correction terms, t_{oc_k} is the clock correction reference time, and T_{GD} is the estimated group delay differential provided in the ephemeris. The term $Fe_k\sqrt{a_k}\sin E_k$ with $F = -4.442807633 \times 10^{-10}\ s/m^{1/2}$ in Equation (3.14) accounts for the relativistic effects between the satellite and user. This is the same correction described in Equation (2.2) and is used to incorporate the satellite clock error into the pseudorange equation. The transmit time is now corrected again using Equation (3.15) to account for the satellite clock correction.

$$t_{t_k} = t_{t_k} - \delta t_{s_k} \tag{3.15}$$

Next, the true anomaly ν_k and the argument of latitude ϕ_k are calculated using Equations (3.16) and (3.17) respectively.

$$\nu_k = \arctan \frac{\sqrt{1-e_k^2}\sin E_k/(1-e_k\cos E_k)}{(\cos E_k - e_k)/(1-e_k\cos e_k)} \tag{3.16}$$

$$\phi_k = \nu_k + \omega_k \tag{3.17}$$

Where ω_k is the argument of perigee provided by the satellite ephemeris. Next, three correction terms $\delta\phi_k$, δr_k, and δi_k are calculated using Equations (3.18 – 3.20)

$$\delta\phi_k = C_{us_k}\sin 2\phi_k + C_{uc_k}\cos 2\phi_k \tag{3.18}$$

$$\delta r_k = C_{rs_k}\sin 2\phi_k + C_{rc_k}\cos 2\phi_k \tag{3.19}$$

$$\delta i_k = C_{is_k}\sin 2\phi_k + C_{ic_k}\cos 2\phi_k \tag{3.20}$$

where i_k is the inclination angle and $idot_k$ is the inclination angle rate provided by the ephemeris. These terms are used to correct the argument of latitude ϕ_k, radius r_k, and inclination i_k using Equations (3.21) through (3.23).

$$\phi_k = \phi_k + \delta\phi_k \tag{3.21}$$

$$r_k = a_k(1 - e_k \cos E_k) + \delta r_k \tag{3.22}$$

$$i_k = i_k + \delta i_k + idot_k(t_{t_k} - t_{oe_k}) \tag{3.23}$$

Next, the longitude of the ascending node Ω_k is corrected to account for the angle between the ascending node and the Greenwhich meridian using Equation (3.24)

$$\Omega_k = \Omega_k + t_{t_k}(\dot{\Omega}_k - \dot{\Omega}_e) - \dot{\Omega}_e t_{oe_k} \tag{3.24}$$

where $\dot{\Omega}_e = 7.2921151467 \times 10^{-5}\ rad/s$ is the rotation rate of the Earth. Finally, the position $\bar{r}_{s_k}$ for satellite k can be calculated using Equation (3.25).

$$\bar{r}_{s_k} = \begin{bmatrix} x_{s_k} \\ y_{s_k} \\ z_{s_k} \end{bmatrix} = \begin{bmatrix} r_k \cos\Omega_k \cos\phi_k - r \sin\Omega_k \cos i_k \sin\phi_k \\ r_k \sin\Omega_k \cos\phi_k + r \cos\Omega_k \cos i_k \sin\phi_k \\ r_k \sin i_k \sin\phi_k \end{bmatrix} \tag{3.25}$$

The satellite positions $\bar{r}_{s_k}$ and the clock correction term δt_{s_k} are generated at each epoch of the scenario simulation and are used to model the user-satellite pseudoranges detailed in Section 3.2.4.

3.2.4 Pseudorange Generation

After the satellite positions are calculated, the user-satellite pseudoranges are generated. In Equation (2.3), the pseudorange is modeled as the true user-satellite range plus error sources that affect the range measurement in GPS receivers. The error sources are caused by the satellite and receiver clock offsets, atmospheric effects, rotation of the Earth between transmit and receive times, multipath, and other sources that obstruct the signal propagation path. A generic pseudorange model in meters is provided in Equation (3.26) [15]

$$\rho = r + c[\delta t_r - \delta t_s] + I_\rho + T_\rho + \epsilon \tag{3.26}$$

where r is the true user-satellite range, δt_r and δt_s are the user and satellite clock offsets respectively, I_ρ is the error caused by the ionosphere, T_ρ is the error caused by the troposphere, and ϵ is unmodeled effects. The receiver clock offset is different across GPS receivers and is estimated in the least squares position caluclation so it is not modeled in this thesis. Also, the effect caused by the rotation of the Earth between the signal transmit and receive time must taken into account [18]. Equation (3.26) is modified to include this correction in Equation (3.27)

$$\rho_k^a(t) = r_k^a(t) + r_{e_k}^a(t) - cdt_{s_k}(t) + I_k(t) + T_k(t) \tag{3.27}$$

where $\rho_k^a(t)$ is the user-satellite pseudorange for satellite k and antanna a. The atmospheric errors $I_k(t)$ and $T_k(t)$ are assumed constant between antennas because of the small spacing of CRPA elements. The error caused by the rotation of the earth between signal transmit and receive times $r_{e_k}^a(t)$ is calculated using Equation (3.28) [18].

$$r_{e_k}^a(t) = \frac{\omega_e}{c}(x_{s_k} y_a - y_{s_k} x_a) \tag{3.28}$$

The pseudorange model provided in Equation (3.27) is the code-based pseudorange and is the basic measurement calculated by all GPS receivers. The code-based pseudorange is defined as the difference between signal reception time determined by the receiver clock and the transmission time determined by the satellite clock [15]. A GPS receiver measures the code-based pseudoranges in the acquisition and tracking loops and uses them to calculate position. In the context of GPS signal simulation, the code-based pseudoranges generated by Equation (3.27) are used to calculate the phase of the CA code chips and navigation data bits that are modulated onto the simulated carrier signal. The receiver under test aquires and tracks the simulated signal, measures the modeled pseudoranges, and calculates

the desired simulated receiver position. The tracking stage of a receiver also measures the carrier phase of the received signal which is defined as the difference between the receiver generated carrier signal and the received carrier phase measurement [15]. The phase of the received signal $\Phi_k(t)$ is related to the phase of the signal at the satellite in cycles in Equation (3.29)

$$\Phi_k(t) = \Phi_u(t) - \Phi_k(t-\tau) + N \tag{3.29}$$

where $\Phi_u(t)$ is the receiver generated phase, $\Phi_k(t-\tau)$ is the carrier phase of the signal at the transmit time, τ is the signal transit time, and N is the integer ambiguity. The integer ambiguity is an unknown number of whole cycles between the satellite and receiver. Equation (3.29) is simplified to obtain a generic carrier phase model in cycles in Equation (3.30)

$$\Phi_k(t) = \frac{r_k(t, t-\tau)}{\lambda} + N \tag{3.30}$$

where $r_k(t, t-\tau)$ is the range between the user at receive time t and the satellite at the transmit time t_{t_k} and λ is the signal wavelength ($\lambda_{L1} = \frac{c}{f_{L1}} \approx 19cm$). The pseudorange from Equation (3.27) is substituted into Equation (3.30) to obtain to form a carrier phase measurement model in cycles provided by Equation (3.31) [15].

$$\Phi_k^a(t) = \frac{r_k^a(t) + r_{e_k}^a(t) - cdt_{s_k}(t) - I_k(t) + T_k(t)}{\lambda_{L1}} + N \tag{3.31}$$

It should be noted that the only difference between the pseudorange model in Equation (3.27) and the range portion of the carrier phase model in Equation (3.31) is that the ionospheric error is subtracted in the carrier phase model. The reason for this is discussed in Section 3.2.5. The range portion of the carrier phase measurement is known as the carrier-based pseudorange and is described by Equation (3.32).

$$\phi_k^a(t) = r_k^a(t) + r_{e_k}^a(t) - cdt_{s_k}(t) - I_k(t) + T_k(t) \tag{3.32}$$

The carrier-based pseudorange $\phi_k^a(t)$ is used to generate the phase of the simulated carrier signal. This process is detailed in Section 3.3.1.

3.2.5 Range Errors

The range errors discussed in Section 3.2.4 affect the the receiver range measurement in the following ways: the clock errors induce timing biases in the actual measurement process and the atmospheric errors deteriorate the actual range that is measured. These cause errors in the user position estimates and must be accounted for in the receiver for optimal performance. When simulating GPS signals, these errors must be included to ensure a complete and accurate simulation environment but can be omitted or changed for custom scenarios or initial algorithm development. Also, there are many other errors that affect the range measurements and if they can be modeled intelligently, they can be included in the simulation by adding them to the code and carrier-based pseudorange equations.

The satellite clock error $dt_{s_k}(t)$ causes the largest error in the pseudorange measurement. The satellite clock errors are typically around 100 nanoseconds but since the GPS signals travel at the speed of light, the offsets can cause a 30 kilometer range error at the receiver [15]. The satellite clock correction must be included in the psedudorange models when simulating GPS signals because GPS receivers automatically account for this in their position solution.

When the GPS signals propagate through Earth's atmosphere they are refracted, which means that their velocity is changed [15]. This velocity change affects the signal transit time to the receiver thereby corrupting the measured user-satellite range. First, the signal propagates through the ionosphere which is a layer of ionized gases at about 50-100 kilometers above the Earth's surface. The ionosphere is a dispersive medium which means that the refractive index of the gas depends on the signal frequency. When the GPS signal propagates through the ionosphere, a phenomena called code-carrier divergence happens, which means that the code and carrier signals exhibit different refractive indexes becuase they are modulated with different frequencies ($f_{CA} = 1.023$ MHz and $f_{L1} = 1575.42$ MHz).

The CA code is delayed and the carrier signal is advanced when propagating through the ionosphere, meaning that the receiver measures the code-based pseudorange too long and the carrier-based range or carrier phase too short [15]. This is accounted for in receivers in Equations (3.27) and (3.31) and when simulating GPS signals in Equations (3.27) and (3.32). In the software simulator detailed in this thesis, the Klubuchar method is used to model the ionospheric errors [21]. The parameters required to generate this model are broacasted in the GPS navigation message and the method is designed to reduce error in the range measurement by about 50 %.

After propagating through the ionosphere, GPS signals reach the troposphere which is the lower part of Earth's atmosphere and consists of dry gases and water vapor. The troposphere is not a dispersive medium and delays both the code and carrier on the modulated GPS signal [15]. The amount of delay depends on the elevation angle of the impeding signal and the refractive index of the air along the signal path which depends on the pressure and temperature of the gases and vapor. In the software simulator, the wet-dry delay mapping method described in [15] is used to model the troposphere error. Since the scope of this thesis is not simulating complicated atmospheric models, only simple models were used to add errors to the simulation to provide a somewhat realistic signal. If a more detailed or complex atmospheric model is needed for certain applications, the corresponding range errors can be simply added to Equations (3.27) and (3.32). In the software simulator, the ionosphere and troposphere errors can be turned off for baseline analysis.

3.2.6 CA code and Navigation Data Generation

The GPS signals are designed as code division multiple access (CDMA) signals which means that they are transmitted at the same center frequency but modulated with codes that are nearly uncorrelated with each other [32]. This allows receivers to discern between different satellites in the acquisition stage. The CA code is a Gold code with a chipping rate of 1.023 MHz which repeats every millisecond. The CA codes are generated from the

product of two 1023 bit PRN sequences, called G1 and G2, that are generated by linear shift registers with 10 stages driven at 1.023 MHz [32]. Shift registers have n bits and the total length of the sequence is $2^n - 1$. The CA code generation registers have 10 bits providing a 1023 bit sequence ($2^{10} - 1 = 1023$). The initial 10 bits of both registers are set to ten ones and the output is dependent on the feedback path. Bits 3 and 10 are used in the feedback path of G1 and bits 2, 3, 6, 8, 9, and 10 are used in G2. The values of these bits are modulo-2 added (logical XOR) and the resulting values are set as the first bit of the register and the remaining bits are shifted right. The output of the G1 register is the modulo-2 addition of the feedback path and the output of the G2 register is the modulo-2 addition of two code phase bits. The outputs of G1 and G2 are modulo-2 added again and the result is the corresponding CA code chip. The satellite identification is determined by the G2 code phase bits which can be found in the ICD [25]. The length-1023 sequence is generated for each visible satellite in the simulation and modulated onto the carrier signal. The modulation scheme is detailed in Section 3.3.3. A block diagram of the CA code generation is provided in Figure 3.5.

The navigation data is a binary message that is generated and modulated onto the carrier signal at 50 bits per second. The complete navigation message transmitted by each satellite consists of 25 1500-bit-long frames, each containing 5 subframes. Each subframe contains ten 30-bit words [3]. Subframes 1-3 are repeated in each frame and contain satellite clock corrections, health information, and the ephemerides. There are 25 different versions of subframe 4 and 5 which contain a less-accurate version of the ephemerides called an almanac, the ionospheric model, and other various parameters. The first two words of each subframe are called the telemetry word (TLM) and handover word (HOW). The TLM word contains a preamble which is used for bit synchronization and the HOW contains a truncated version of the transmit time which is used to measure the user-satellite range in the tracking loops. Complete versions of subframes 1-3 and only the TLM and HOW words in subframes 4 and 5 are generated in the software signal simulator described in this thesis. The

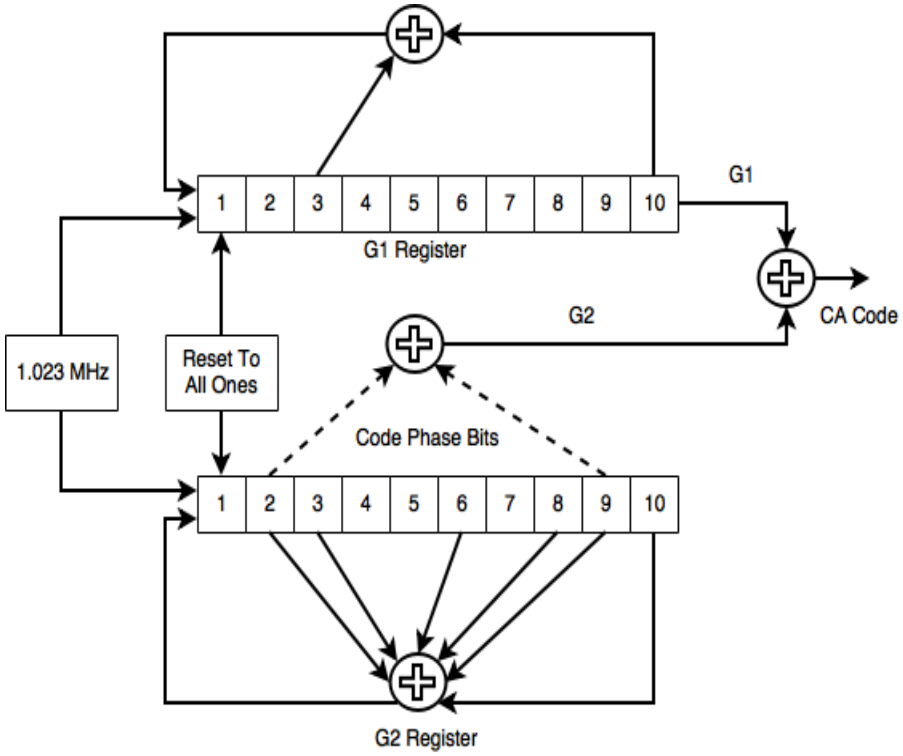

Figure 3.5: CA Code Generation [15]

downloaded ephemerides for each visible satellite are encoded in accordance to the ICD [25] and included in subframes 2 and 3. The initial transmit times calculated in Section 3.2.2 are used to generate the corresponding HOW words for each subframe. A full explanation of the navigation message can be found in the ICD [25] and a diagram illustrating the navigation message stucture is provided in Figure 3.6.

3.3 Signal Simulation

The signal simulation uses the code and carrier-based pseudoranges generated in the scenario simulation to generate the composite GPS signal. The software simualtor generates two types of GPS signals: Real signals at an intermediate frequency (IF signals) and complex baseband I and Q signals (IQ signals) with an intermediate frequency of zero. Seen on the left side of Figure 3.7, the real-IF signals for each antenna are modeled subsequent to a generic GPS receiver front-end and the complex-IQ signals are modeled subsequent to a

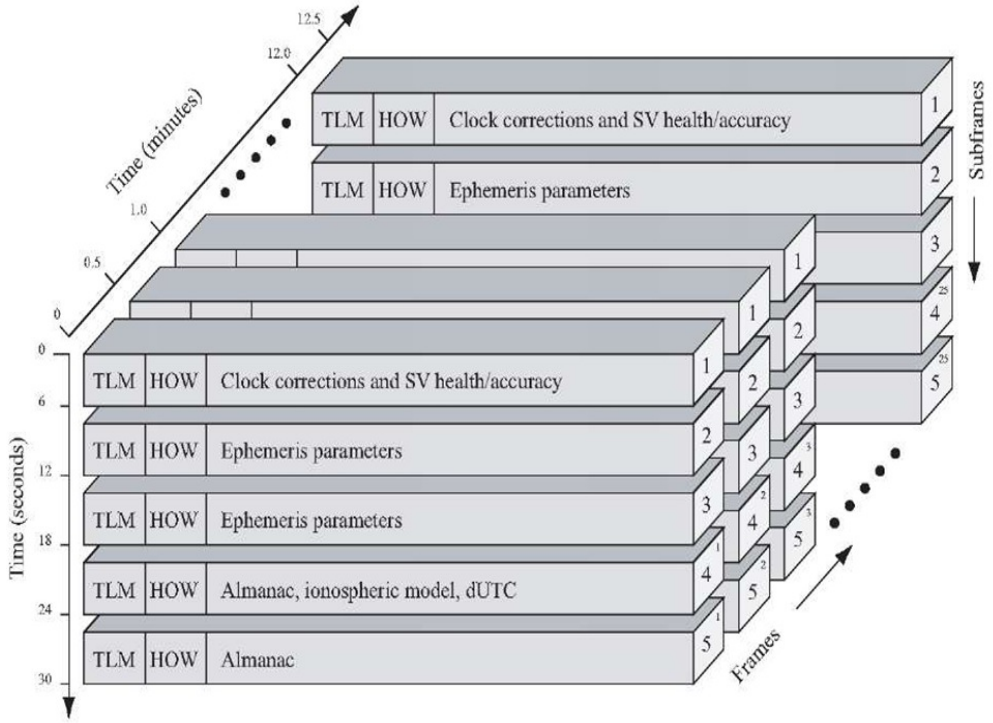

Figure 3.6: Navigation Data Message Structure [3]

software defined radio record process. Both signals are generated as sampled signals which are corrupted with thermal noise, filtered, and quantized to resemble the output of a GPS receiver front-end or an SDR record function. The real-IF signals can be analyzed with a software receiver and the complex-IQ signals can be converted to analog signals at the GPS RF using a software defined radio. A functional block diagram of the signal simulation for one antenna is provided on the right side of Figure 3.7. First, the CA code and navigation data are either loaded or generated. Next, the code and carrier-based pseudoranges are interpolated from the scenario simulation frequency $f_{scenario}$ to the desired signal sampling frequency f_s. Using the interpolated code-based pseudoranges, the CA code and navigation data phases are calculated and the corresponding CA code chips and navigation data are selected from the pre-generated values. The interpolated carrier-based pseudoranges are used to generate the simulated carrier phase and the signal for each visible satellite is generated by modulating the CA code chips and navigation data bits onto a carrier wave generated

using the carrier phase. The signals for each satellite are then summed and combined with white Gaussian noise (WGN) at a user-specified power level and interference signals are added if desired. Finally, the composite signal is band-pass filtered if desired and quantized to a user-specified number of bits and saved to a file for processing.

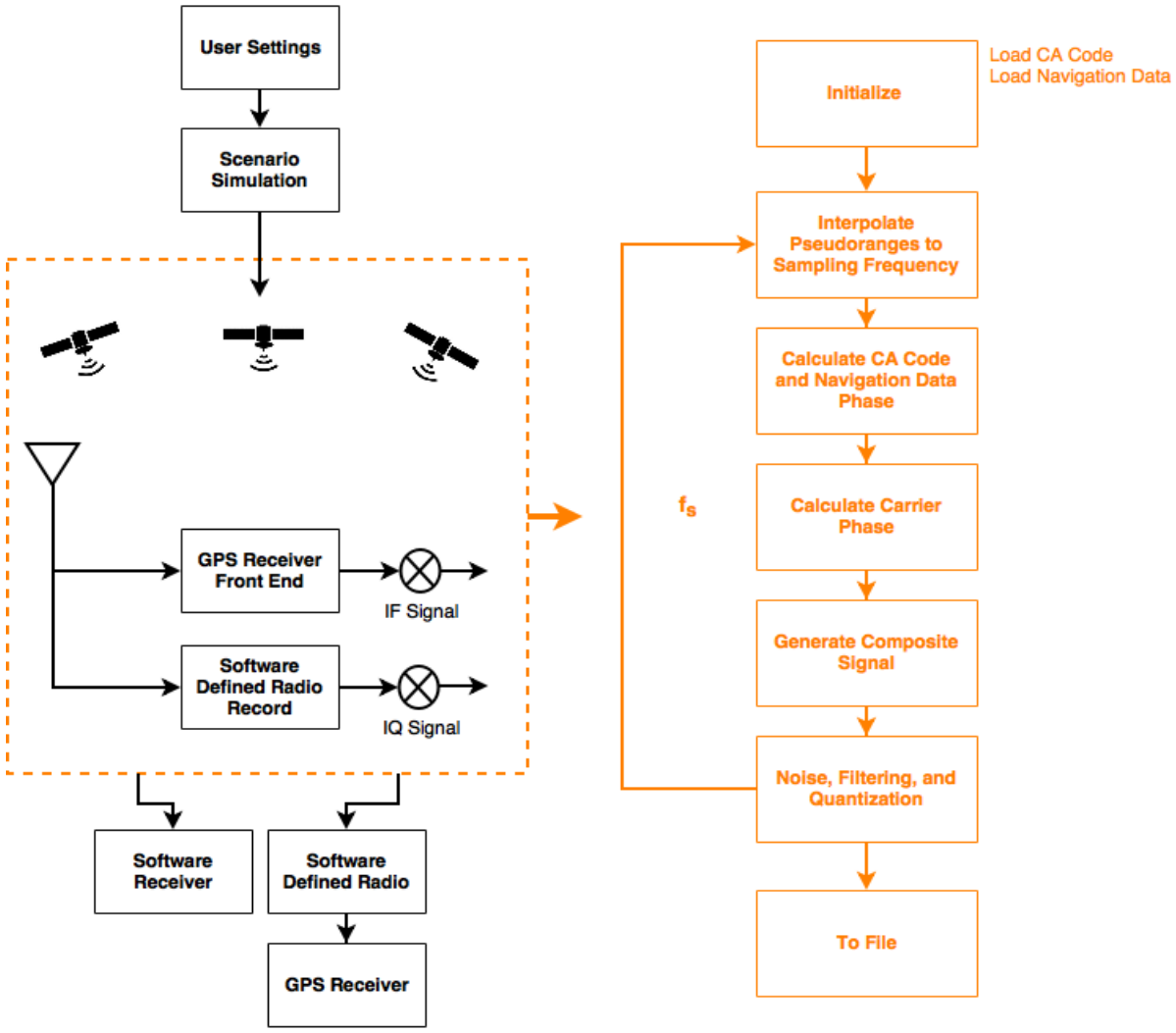

Figure 3.7: Signal Simulation Overview

3.3.1 Signal Models

The simulator detailed in this thesis is capable of generating real-valued IF signals in which the corresponding frequency spectrum is centered at a user desired intermediate frequency. It can also generate complex baseband signals called in-phase and quadrature (IQ signals) in which the frequency spectrum is centered at an intermediate frequency of zero. The real-IF signals are generally analyzed by software receivers and the complex-I and Q signals are converted by a SDR to analog signals for playback into a hardware GPS receiver.

Frequency domain plots of a real-IF and a complex-I GPS L1 CA code signal without thermal noise generated by the software simulator are provided in Figures 3.8 and 3.9

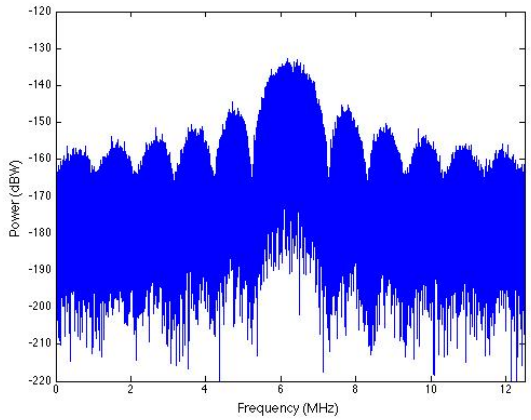

Figure 3.8: Real IF Signal Spectrum

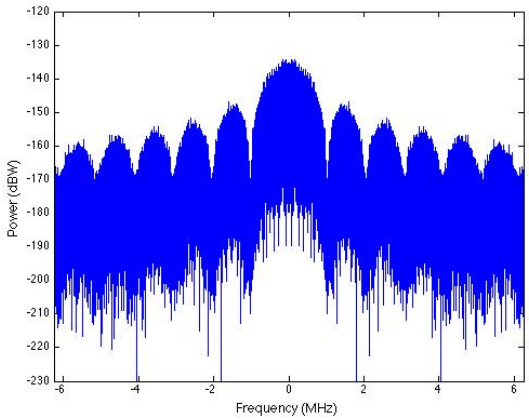

Figure 3.9: Complex I Signal Spectrum

The software simulator generates the real-IF and complex-IQ signals directly for computational purposes but they can be converted to and from each other in post processing. The phasor diagram in Figure 3.10 illustrates the conceptual difference between real-IF signals and complex-IQ signals. A generic-real IF signal is given in Equation (3.33).

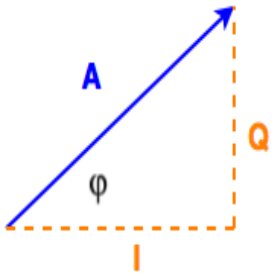

Figure 3.10: Phasor Diagram

$$S_{IF} = Acos(2\pi ft + \phi) = Re\{Ae^{j(2\pi ft+\phi)}\} \tag{3.33}$$

The I and Q signal formulation essentially maps the changes of amplitude and phase of a signal over time [20]. Generic I and Q signals are provided by Equations (3.34) – (3.35).

$$I = Acos(\phi) = Re\{Ae^{j\phi}\} \tag{3.34}$$

$$Q = Asin(\phi) = Im\{Ae^{j\phi}\} \tag{3.35}$$

The software simulator generates real IF signals modeled at the point subsequent to the front-end of a GPS receiver. In the front-end, RF GPS signals are amplified, downconverted to an intermediate frequency, filtered, sampled, and quantized to form a digital signal. This illustrated by Figure 3.11. The simulator generates complex-IQ signals modeled subsequent to the typical USRP record process. The USRP amplifies and downcoverts the RF GPS signals to baseband. The USRP then samples and quantizes the downconverted signal and saves interleaved digital I and Q samples to a file. Another USRP can upconvert the I and Q samples saved in the file back to the analog signal that was recorded. This process is illustrated in Figure 3.12.

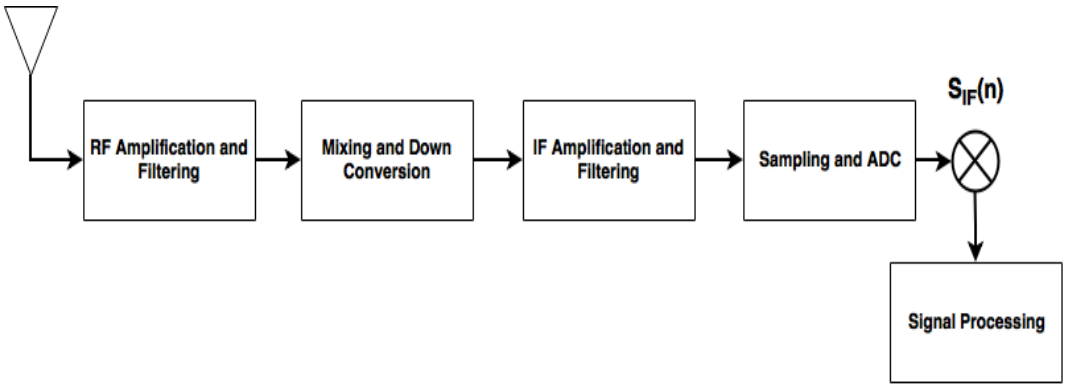

Figure 3.11: Real IF Signal Generation

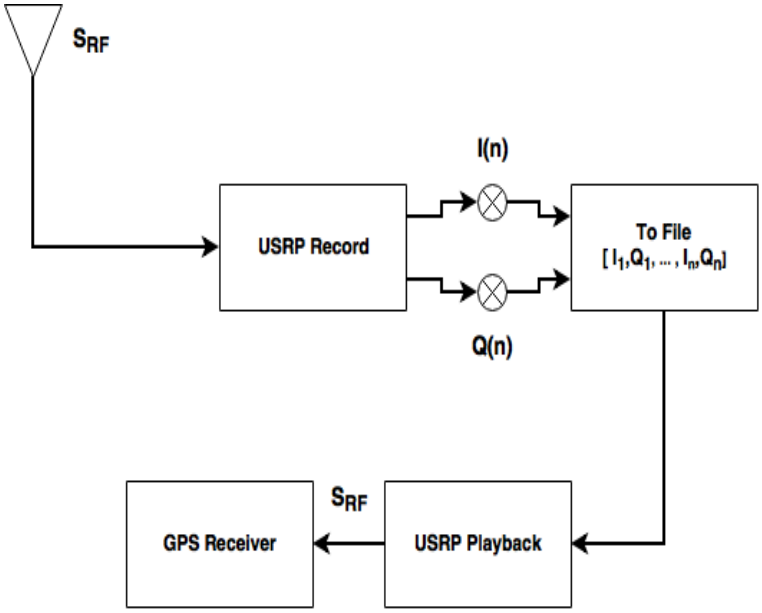

Figure 3.12: Complex I and Q Signal Generation

The real-IF signal for satellite k, antenna a, and sample n is generated using Equation (3.36) [13]

$$S^a_{IF}(n) = \left[\sum_{k=1}^{K} \sqrt{2P^a_k(n)}C^a_k(n)D^a_k(n)cos\Big(2\pi\Big(\frac{f_{IF}n}{f_s} - \frac{\phi^a_k(n)}{\lambda_{L1}}\Big)\Big)\right] + J^a(n) + N(n) \quad (3.36)$$

where $P^a_k(n)$ is the signal power in Watts, $C^a_k(n)$ is the CA code chip, $D^a_k(n)$ is the data message bit, f_{IF} is the intermediate frequency in Hz, f_s is the sampling frequency, $\phi^a_k(n)$ is the carrier-based pseudorange in meters, $N(n)$ is the thermal noise, and $J^a(n)$ is interference. The thermal noise is assumed constant between antennas. The Doppler frequency shift is not explicitly incuded in Equation (3.36) but is embedded in the product of 2π and $\frac{\phi^a_k(n)}{\lambda}$[13]. The complex I and Q samples are generated using Equation (3.37).

$$I^a_k(n) + jQ^a_k(n) = \left[\sum_{k=1}^{K} \sqrt{2P^a_k(n)}C^a_k(n)D^a_k(n)e^{(-j2\pi\frac{\phi^a_k(n)}{\lambda_{L1}})}\right] + J^a(n) + N(n) \quad (3.37)$$

The complex baseband samples are interleaved and saved to a file in the same format that the USRP saves recorded samples. The signals generated by Equations (3.36) and (3.37) are generated using a block processing scheme in the MATLAB version of the simulator for computational purposes. A block of signal samples are generated and saved to a file at each iteration of the signal simulation. A block size 20 milliseconds was used in the software simulator. After the signals are generated and thermal noise is added, they are filtered if desired and quantized to a user-desired number of bits. The CA code and data bit phase calculations are detailed in Section 3.3.3 and noise, filtering, and quantization are described in Section 3.3.4.

3.3.2 Signal Power and Signal to Noise Ratio

The GPS signals are very weak when they reach the Earth's surface (about -160 dBW or 10^{-16} Watts) [15]. They are so weak that they are buried beneath the local thermal noise floor. In this thesis, the thermal noise is modeled as white Gaussian noise (WGN) which is the same power at all frequencies. The thermal noise has a power spectral density (PSD) of N_0 in Watts/Hz which is a measure of the noise power in the receiver front-end bandwidth. Typically, GPS receivers use the carrier-to-noise ratio $\frac{C}{N_0}$ in dB-Hz to quantify each satellites's power in the receiver bandwidth. The carrier-to-noise ratio is dependent on receiver bandwidth and usually ranges between 35 and 50 db-Hz in most receivers in normal noise conditions. Since every GPS receiver has a unique front-end bandwidth, a true signal-to-noise ratio $\frac{S}{N}$ in dB can be used to approximate the signal power of each simulated satellite. The signal power in Watts $P^a_{k_W}(n)$ for satellite k and antenna a from Section 3.3.1 is generated using Equation (3.38).

$$P^a_{k_W}(n) = 10^{\left(\frac{P_{N_{dB}} + \frac{S}{N}}{10}\right)} \tag{3.38}$$

The signal power for each satellite in the simulation is calculated based on the user-specified noise power $P_{N_{dB}}$ in dB and signal-to-noise ratio. The noise power is selected to match a desired receiver front-end amplification factor. This method is used because of it's simplicity but the carrier-to-noise ratio of the signals will vary across different receivers used for analysis because they have different front-end bandwidths. If custom receiver front-end bandwidths and carrier-to-noise ratios are desired, the signal power in Watts for each satellite is calculated using Equations (3.39 – 3.42).

$$P^a_{k_W}(n) = 10^{\left(\frac{N_0 + \frac{C}{N_0}}{10}\right)} \tag{3.39}$$

$$N_0 = 10 \log N_{0_W} \tag{3.40}$$

$$N_{0_W} = \frac{P_{N_W}}{B} \tag{3.41}$$

$$P_{N_W} = 10^{\left(\frac{P_{N_{dB}}}{10}\right)} \tag{3.42}$$

Where N_{0_W} is the noise PSD in Watts/Hz, P_{N_W} is the noise power in Watts, and B is the desired bandwidth in Hz. In the software simulator, the bandwidth in Equation (3.41) is approximately equal to half of the signal sampling frequency. The equations described in this section assumed constant signal-to-noise and carrier-to-noise ratios across simulated satellites but the simulator is capable of generating different values for each satellite in custom power simulations. Also, more complicated models of specific receiver front-ends can be used to calculate the local noise PSD using the effective noise figure and the temperature of the front-end electronics [15].

3.3.3 CA Code and Data Message Phase

The code-based pseudoranges generated in Section 3.2.4 are reflected on the simulated signal via the CA code and data message phase. The phase of each are generated in such a way that a receiver measures the the simulated pseudoranges of each visible satellite and calculates the desired simulated receiver trajectory. Receivers measure the intial pseudorange of each satellite channel by searching for the start of a navigation data subframe. Specifically, the receiver searches for the preambles provided in the TLM word at the beginning of each subframe of the navigation data. Illustrated by Figure 3.4, at the initial receive time t_{r_i} there are K satellites in view, each a different distance from the receiver. This means the first subframe that the receiver decodes corresponds to the closest satellite visible to the user. The first satellite transit time is set between 63 and 85 milliseconds and the remaining satellite subframe starts and transit times are measured relative to the start of the earliest arriving subframe [3]. The receiver then moves the measurement indexes to find the start of the CA code in the same frame providing an initial pseudorange measurement for each channel with a resolution corresponding to the sampling frequency [3]. This process is repeated at the receiver measurement frequency for subsequent pseudoranges.

When simulating GPS signals, the navigation data bits modulated onto the signal are selected such that the receiver decodes the preamble of the closest satellite first and the preambles of the remaining satellites in order of increasing initial user-satellite distance. The data bit phase for satellite k, antenna a and sample n is generated using Equation (3.43)

$$D_k^a(n) = \lfloor \left(t_{t_k}^a(n) - \lfloor t_{r_i} \rceil \right) f_{data} \rfloor \tag{3.43}$$

where $t_{t_k}^a(n)$ is the transmit time and f_{data} is the data message frequency. When the navigation data is generated in Section 3.2.6, the initial transmit times included in the initial HOW must be rounded when they are converted to binary. Since GPS signal transit times

are between 63 and 85 milliseconds, these will all round to the initial receive time t_{r_i} which is rounded in Equation (3.43) to account for numerical errors in the year-month-day to GPS time-of-week initialization. The transmit time is generated using Equation (3.44)

$$t^a_{t_k}(n) = t_r(n) - \frac{\rho^a_k(n)}{c} \tag{3.44}$$

where $t_r(n)$ is the current receiver time and ρ^a_k is the code-based pseudorange in meters. The user time at sample n is calculated using Equation (3.45).

$$t_r(n) = t_{r_i} + \frac{n-1}{f_s} \tag{3.45}$$

The data bit phase for sample n generated in Equation (3.43) is used to select from the pre-generated binary navigation message for each satellite in the simulation. This formulation ensures the first bit of the navigation message corresponding to the shortest initial user-satellite range is generated first and the subsequent first bits are generated in ascending order of initial user-satellite range. For the first 63 - 85 milliseconds of the simulation, Equation (3.43) generates negative values. These values are set to one, meaning that no satellites are are active in the simulation. When the data bit phase for a satellite value becomes positive, one (+ 1) is added to it and it is used to select the data bit that is modulated on the simulated signal at that sample. The data bit phase for each satellite will become positive in ascending order of initial user-satellite range until all satellites enter the simulation. This process is illustrated further Equation (3.46).

$$D^a_k(n) = \begin{cases} 1 & \text{if } D^a_k(n) < 0 \\ D^a_k(n) + 1 & \text{if } D^a_k(n) >= 0 \end{cases} \tag{3.46}$$

The CA code phase for satellite k, antenna a, and sample n is generated in a similar manner using Equation (3.47).

$$C_k^a(n) = \lfloor \left| t_{t_k}^a(n) - \lfloor t_{r_i} \rceil \right|_{10^{-3}} f_{CA} \rfloor + 1 \tag{3.47}$$

Where f_{CA} is the CA code frequency and $\left| t_{t_k}^a(n) - \lfloor t_{r_i} \rceil \right|_{10^{-3}}$ is the modulus of $t_{t_k}^a(n) - \lfloor t_{r_i} \rceil$ by 10^{-3} seconds (1 millisecond). This ensures that the CA code chip selection value repeats every millisecond. Also, $C_k^a(n)$ values that equal zero are set to 1023 which is last chip of the CA code sequence. The CA code phase calculated in Equation (3.47) is used to select from the pre-generated binary CA code sequences.

3.3.4 Noise, Filtering, and Quantization

The noise added to the signals in Equations (3.36 – 3.37) is modeled as WGN which has a constant PSD at all frequencies [15]. A simulated real-IF GPS signal with WGN added is illustrated by Figure 3.13.

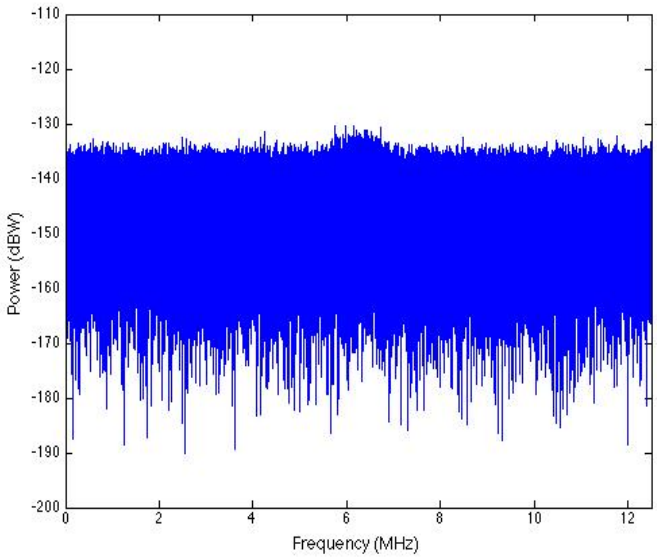

Figure 3.13: Frequency Domain of IF Signal with White Gaussian Noise

After noise is added, if desired, the IF signal can be band pass filtered. This is accomplished via a digital filter that can be customized to the the user's needs to match receiver

front-end specifications. The real IF-GPS signal pictured in Figure 3.13 filtered by a digital Equiripple band pass filter with a 4 MHz pass-band is illustrated by Figure 3.14.

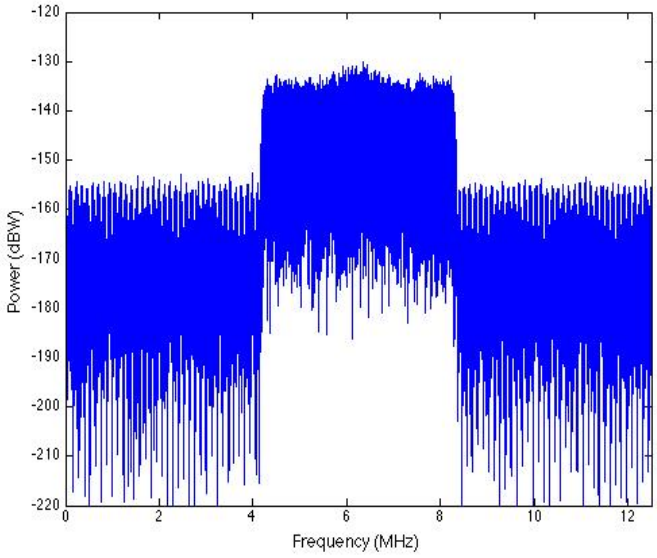

Figure 3.14: Frquency Domain of IF Signal with White Gaussian Noise and Equiripple Band Pass Filter

Before the simulated signal samples are saved to a file, they are quantized to a user-specified number of levels to simulate the ADC in a GPS front-end or in the USRP. Quantization is the process in which a continuous analog signal is converted to a set of discrete values [1]. The number of quantization levels N and quantization size Q are calculated using Equations (3.48 – 3.49) respectively [1]

$$N = 2^n \tag{3.48}$$

$$Q = \frac{V_{max} - V_{min}}{N} \tag{3.49}$$

where n is the desired number of bits used to represent each level of an analog signal and $(V_{max} - V_{min})$ is the dynamic range of the ADC. A simple example of a quantized sine wave is provided in Figure 3.15.

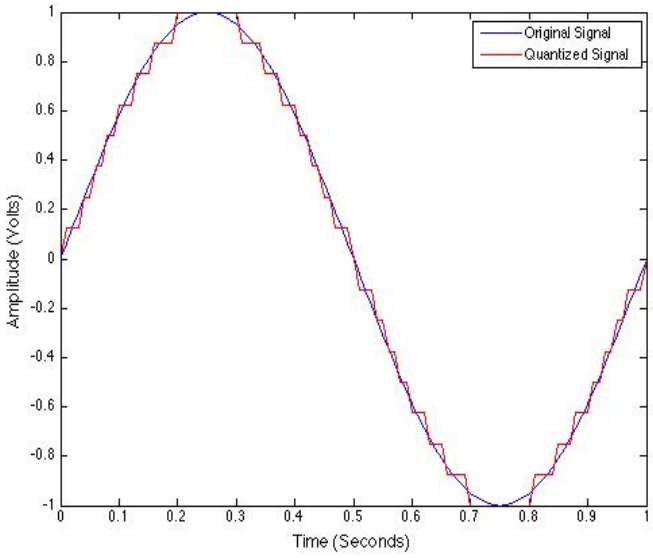

Figure 3.15: Quantized Sine Wave

GPS receivers can use single-bit quantization to keep the dynamic range of the front-end small and the computational requirements low but receivers that must operate in degraded signal environments like jamming must use multiple bit quantization to ensure the there is enough dynamic range to process the GPS signals swamped with interference [32]. This requires implementation of automatic gain control (AGC), in which the receiver dynamically increases or decreases the amplification of the signals in the front-end to ensure the maximum number of quantization levels are utilized while avoiding saturation [3]. Automatic gain control requires a closed loop feedback system which can be difficult to implement in a software simulation. In [13] and [33] a 2-bit quantization scheme is utilized in which the quantization threshold is switched between 2 values based on the input signal power such that the samples are quantized to a desired statistical pattern. In this thesis, a simple AGC approximation is used to generate real IF signals. Since the signal and interference power levels are known a-priori, the quantization threshold of each block of simulated signal samples can be set accordingly to ensure that the maximum number of quantization levels are utilized. Essentially, the quantization threshold is set to a constant value that ensures that the maximum number of quantization levels are used under normal conditions. If

interference is simulated, the quantization threshold is set to a second value that is higher than the maximum simulated interference power level. When recording signals with a USRP, the pre-record gain and attenuation is set manually to ensure the recorded GPS signals are below the quantization threshold. Usually, extra attenuation is added to account for change in signal power due to change in elevation angle or an increase of noise power due to interference. This process is essentially executed in reverse when simulating complex I and Q signal samples modeled subsequent the USRP receive chain. The signals, noise, and iterference power are scaled to meet the quantization threshold of the USRP ADC which depends on how many bits are used to save each sample. For example, 8-bit samples provide a dynamic range of ± 128 Volts. Examples of the final simulated GPS IF signals without quantization and with 4-bit quantization in the time domain are provided in Figures 3.16 and 3.17 respectively. After quantization, the simulated samples are encoded to a specific data type and saved to a file for processing.

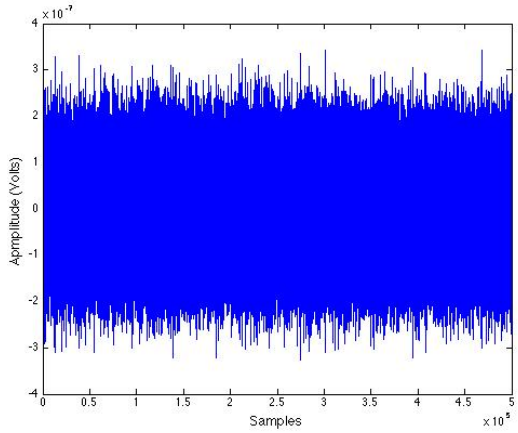

Figure 3.16: IF GPS Signal With No Quantization

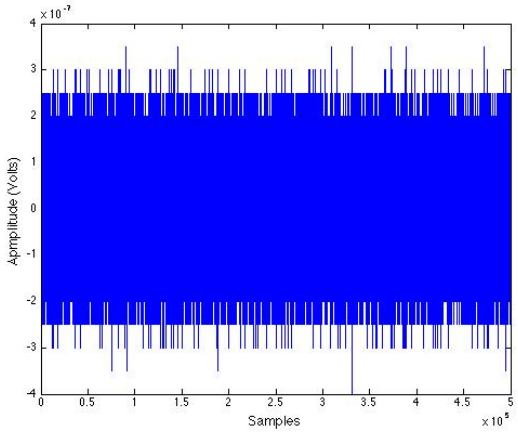

Figure 3.17: IF GPS Signal With 4-bit Quantization

3.4 USRP Playback and Record

The complex-I and Q signals generated by the software simulator can be upconverted from a digital baseband signal to an analog signal at the GPS radio frequencies using the

USRP software defined radio. In this thesis, the USRP N210 paired with the WBX daughterboard from Ettus Research are used for record and playback of GPS signals [27]. The WBX daughterboard provides a frequency range of 50 MHz to 2.2 GHz. The field programmable gate array (FPGA) in the USRP N210 is capable of recording and playing back 8 and 16 bit samples with maximum bandwidths of 50 and 25 MHz respectively [29]. The USRP utilizes a 10 MHz signal for frequency reference and a pulse-per-second (PPS) signal for sampling which are provided from on-board references or can be provided externally. On-board options include a temperature compensated crsytal oscillator (TCXO) or an oven compensated crystal oscillator (OCXO). The OCXO can operate stand-alone or it can be driven by an external GPS signal, which is referred to as a GPS disciplined oscillator (GPSDO). It has been shown in [12] and [19] that the internal TCXO reference produces excessive phase noise when recording and playing back with the USRP causing errors in the GPS tracking loops. Therefore, the OCXO and OCXO/GPSDO combination were used when recording or playing back the simulated signals detailed in this thesis. The timing sources can be shared across multiple devices via a multiple-input-multiple-output (MIMO) cable. The MIMO cable allows for multiple-USRP configurations in which the sample clocks and times are aligned across the multiple devices [28]. A generic example of a two-USRP setup is provided in Figure 3.18.

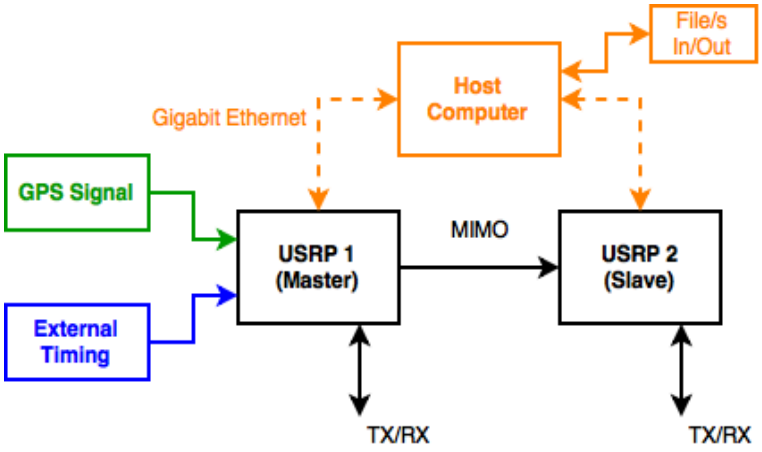

Figure 3.18: 2-USRP Setup

The host computer controls the USRPs via gigabit ethernet. The computer pulls from an external file source when playing back and saves to a file sink when recording. If an external timing source is used, it can be provided to each USRP or it can be shared from the master to the slave via the MIMO cable. A GPS signal can be provided to the internal GPS driven OCXO on the master USRP and shared to the slave via the MIMO cable to allow for synchronous and specified start times. This capability allows for simultaneous start times at different geographic locations. A user can simultaneously record live-sky GPS signals at different locations to capture different signal scenarios. This capability also allows for simultaneous multiple-element record or playback of GPS signals. The setup in Figure 3.18 is extended to achieve simultaneous 4-element record or playback. This configuration is illustrated in Figure 3.19.

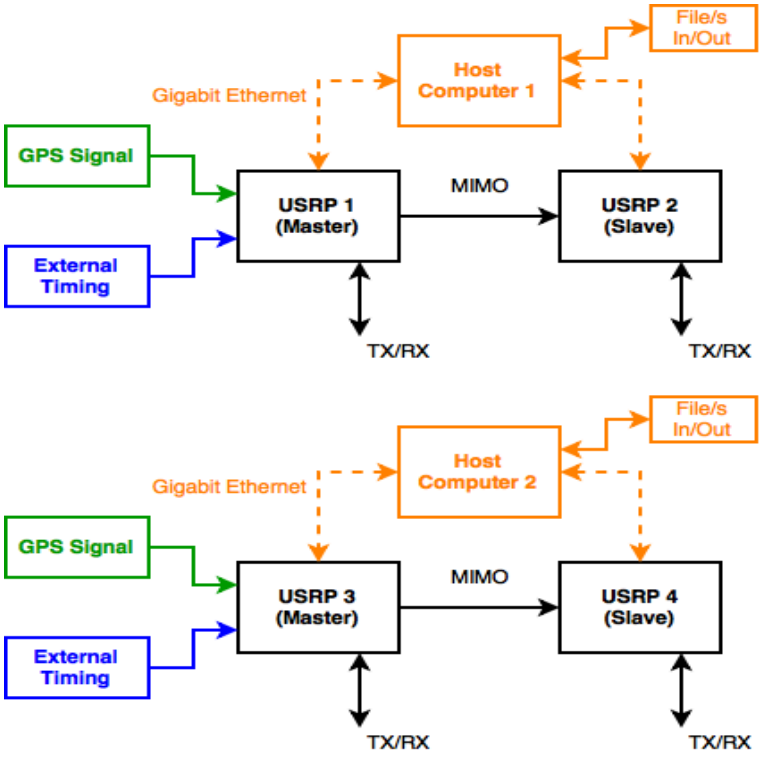

Figure 3.19: 4-USRP Setup

The GAVLAB has partnered with Integrated Solutions for Systems (IS4S) to develop and test their Software-Defined, USRP-based, RF Record, Playback, and Analysis System (SURRPAS) [9]. The SURRPAS utilizes custom software developed using the open source USRP Hardware Driver (UHD) library availible from Ettus Research to interface with the USRPs [27]. The software suite enables synchronous record or playback from multiple devices [9]. The GAVLAB has created a version of the SURRPAS which is illustrated in Figure 3.20.

Figure 3.20: GAVLAB SURRPAS Setup

The GAVLAB SURRPAS rack includes four USRPs driven by two host computers to provide for maximum sampling capabilities. It also includes an Ettus Octoclock for 8 external timing and PPS references if desired. The SURRPAS also includes a signal conditioning rack, which amplifies and attenuates GPS signals to desired levels in record or playback. A block diagram of one element in the signal conditioning box is illustrated in Figure 3.21.

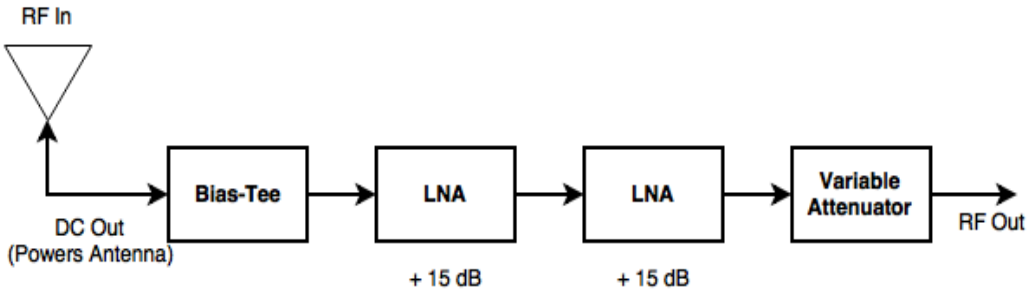

Figure 3.21: SURRPAS Signal Conditioning Element

The bias-tee provides DC power to the antenna, each low-noise-amplifier (LNAs) adds about 15 dB of amplification and the variable attenuator is used to adjust the amplification level to ensure optimum quantization. An optional passive GPS splitter can be added to the signal conditioning racks for dual frequency recordings. The current signal conditioning

rack contains four of the signal conditioning elements illustrated in Figure 3.21. This system allows for flexible and repeatable testing of GPS signal scenarios. The GAVLAB and IS4S have used mutliple SURRPAS to record live-sky GPS interference scenarios at testing events. The signals recorded at these type of events can be brought back to the laboratory and analyzed with software receivers or can be played back to hardware receivers for rapid testing and development. In this thesis, the SURRPAS are used to playback and record signals generated by the software GPS signal simulator for analysis. Single-element playback results are provided in Chapter 4 and multiple-element playback and record of CRPA interference scenarios are analyzed in Chapter 5.

Chapter 4

Software Simulator Results

4.1 Introduction

In this chapter, the software simulator is analyzed in the frequency, time, and histogram domains by comparison to data recorded by a real GPS receiver front-end. Furthermore, the software simulator performance is compared to a commercial-off-the-shelf (COTS) hardware GPS signal simulator. Position, velocity, pseudoranges, Doppler frequencies, and carrier-to-noise ratios of scenarios generated by the software and hardware simulator are compared. Additionally, position results of a dynamic trajectory scenario generated by the software simulator are analyzed in this chapter.

4.2 Comparison to Real GPS Receiver Front-End Data

To ensure that the software simulator can generate signals modeled subsequent a specific GPS receiver front-end, a comparison is made to data taken from a real GPS front-end. The software simulator is compared to data provided with the software receiver detailed in [3] which is described in Table 4.1.

Table 4.1: Real GPS Front-End Data Description [3]

Date	05/07/2005
Time (UTC)	19:10:41
Latitude (°)	40.008047496522
Longitude (°)	-105.262687571629
Altitude (m)	1638.020027550563
Sampling Frequency (MHz)	38.192
Intermediate Frequency (MHz)	9.548
Band Pass Filter Width (MHz)	6
ADC Bits	4

Frequency, time, and histogram plots of the real and simulated signals are provided in Figures 4.1 through 4.6.

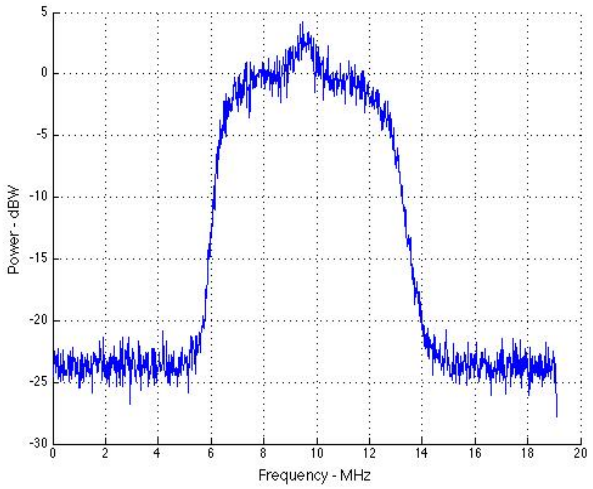

Figure 4.1: Frequency Domain of Real Front-End Signal

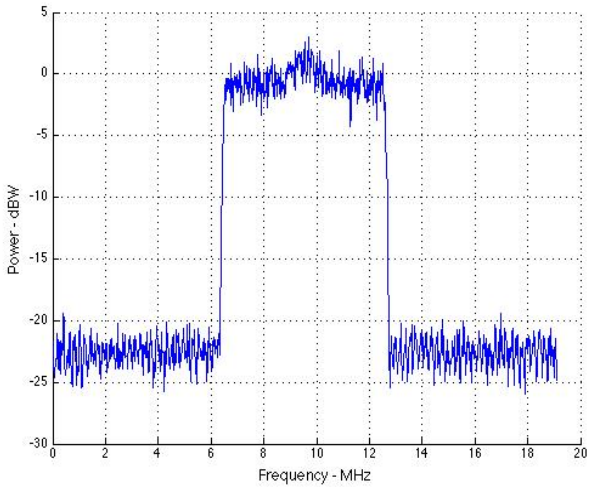

Figure 4.2: Frequency Domain of Simulated Front-End Signal

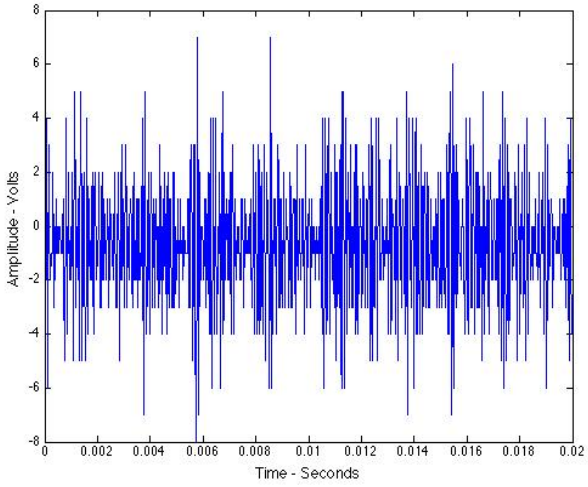

Figure 4.3: Real Front-End Signal vs. Time

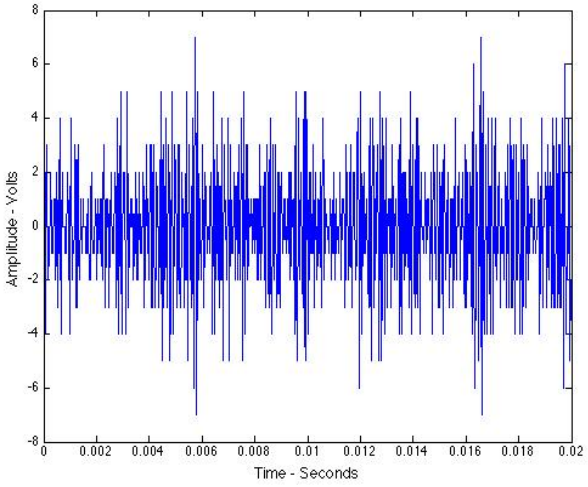

Figure 4.4: Simulated Front-End Signal vs. Time

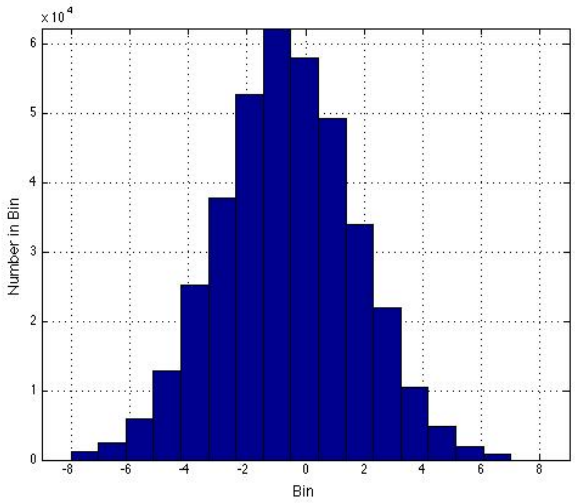

Figure 4.5: Histogram of Real Front-End Signal

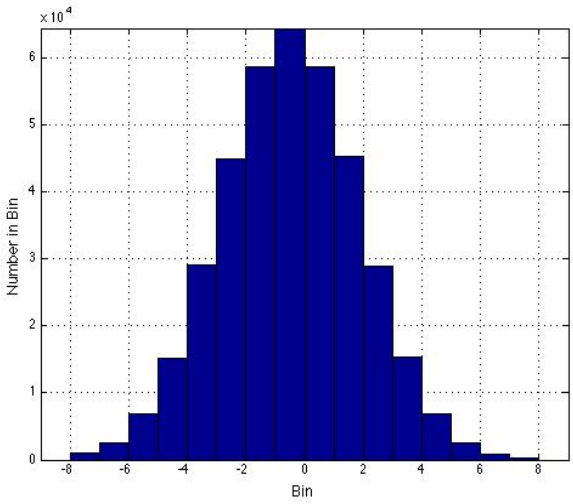

Figure 4.6: Histogram of Simulated Front-End Signal

The frequency plots in Figures 4.1 and 4.2 have similar shapes. The magnitudes of each are similar, the pass bands are about 6 MHz wide, and the stop bands are attenuated about 25 dB. Both plots also display the 2 MHz-wide bulge of the main lobe of the GPS CA code signal. The transition region from the pass-band to the stop-band of the simulated signal is much sharper than that of the real signal because a digital filter was used. In reality, an analog filter will produce a more gradual transition region like that of the real front-end signal. This can be accounted for in the software simulator by choice and design of the BPF. The time domain plots, Figures 4.3 and 4.4 respectively, resemble each other. It should be noted that constant signal-to-noise ratios were simulated, so the actual time domain amplitude values are not identical. The histogram plots show that both the real and simulated signals utilize all 16 levels of the 4-bit ADC. Also, both histograms resemble that of WGN, which is expected since the noise on GPS signals is modeled as such.

4.3 Comparison to Hardware Simulator

In order to quantify the performance of the software simulator, it is compared to a spectracom GSG-6 hardware simulator. The GSG-6 allows for plug and play real-time RF

simulation of multiple GNSS constellations for a single antenna [30]. The comparison scheme is illustrated in Figure 4.7.

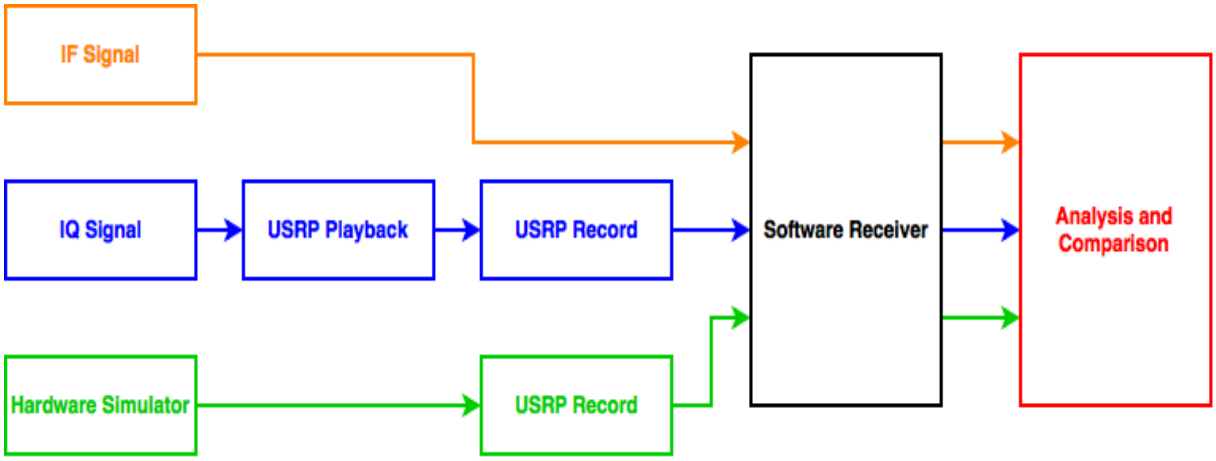

Figure 4.7: Software Simulator Verification Scheme

An IF signal for a static scenario generated by the software simulator was played-back and recorded using the USRPs. An RF signal for the same scenario generated by the hardware simulator was recorded by a USRP. The software receiver capable of computing a Kalman filter position solution detailed in [23] was used to analyze the results provided from these scenarios. A description of the simulation scenario is provided in Table 4.2.

Table 4.2: Static Scenario Description

Date	11/30/2013
Time (UTC)	19:00
Duration (minutes)	20
Latitude (°)	32.602236
Longitude (°)	-85.489192
Altitude (m)	201
Sampling Frequency (MHz)	25
Intermediate Frequency (MHz)	6.25
Carrier-to-Noise Ratio	45
Ionosphere Model	None
Troposphere Model	None
Mask Angle (°)	5

Errors caused by the ionosphere and troposphere were neglected for baseline analysis and a mask angle of 5° was implemented to avoid positioning errors caused by bad satellite geometry in the analysis. Sky plots of the acquired satellites for each of the comparison scenarios are provided below in Figures 4.8 through 4.10.

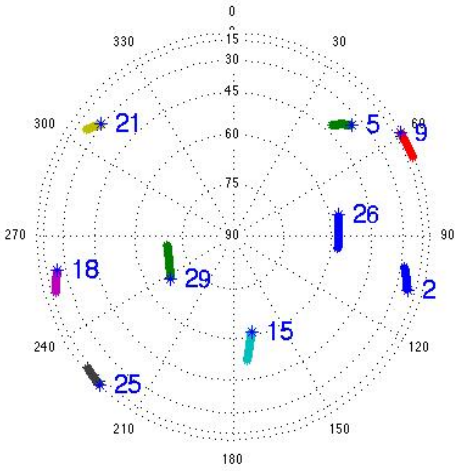

Figure 4.8: Sky-Plot of Record of spectracom GSG-6 Signal

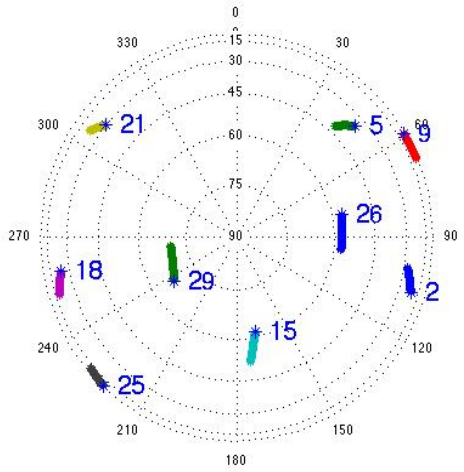

Figure 4.9: Sky-Plot of Simulated IF Signal

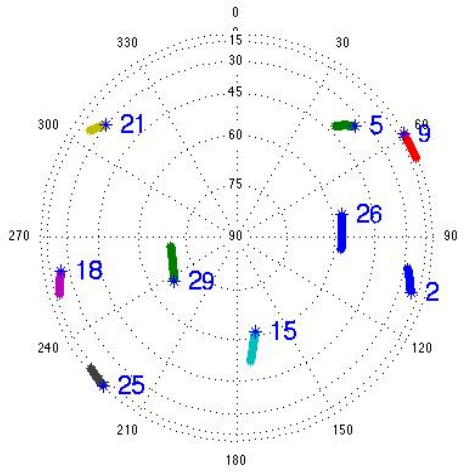

Figure 4.10: Sky-Plot of Playback and Record of Simulated IF Signal

The software receiver acquired the same satellites and calculated the same satellite trajectories for each comparison scenario. Final WGS-84 and ENU position results of each comparison scenario are provided in Figures 4.11 through 4.14.

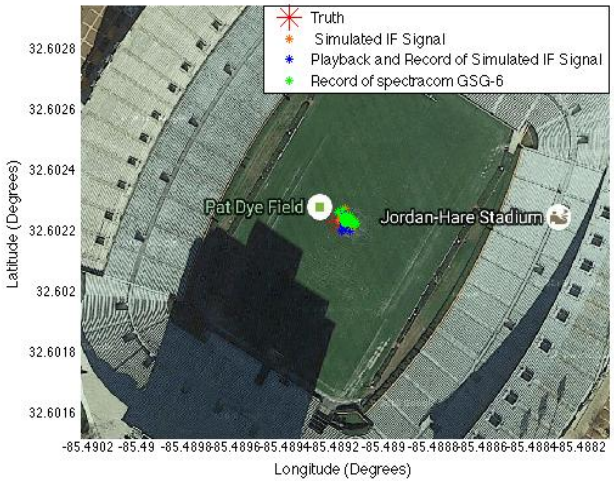

Figure 4.11: WGS-84 Position Results of Static Scenario

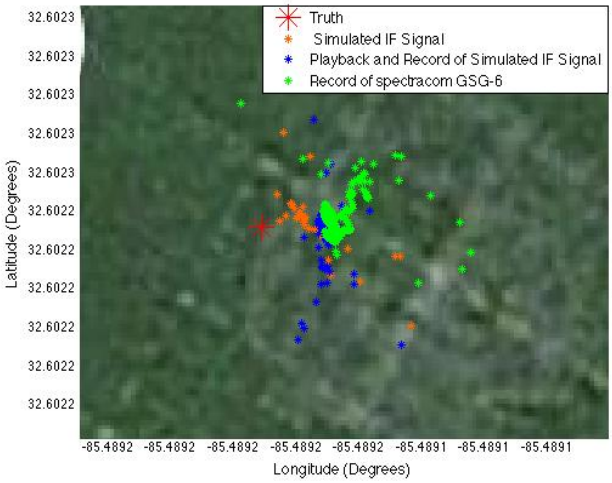

Figure 4.12: WGS-84 Position Results of Static Scenario - Zoomed-In

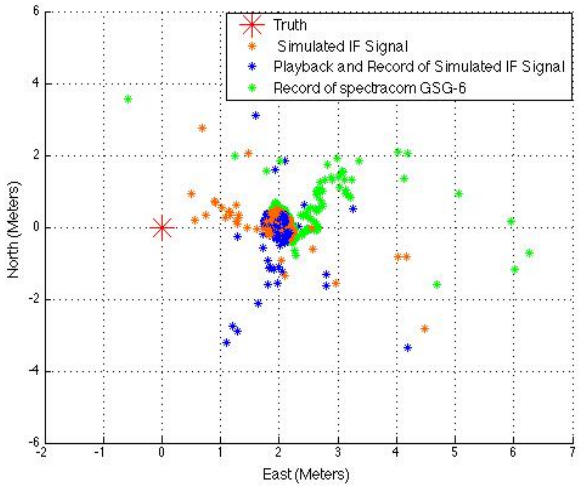

Figure 4.13: 2D ENU Position Results of Static Scenario

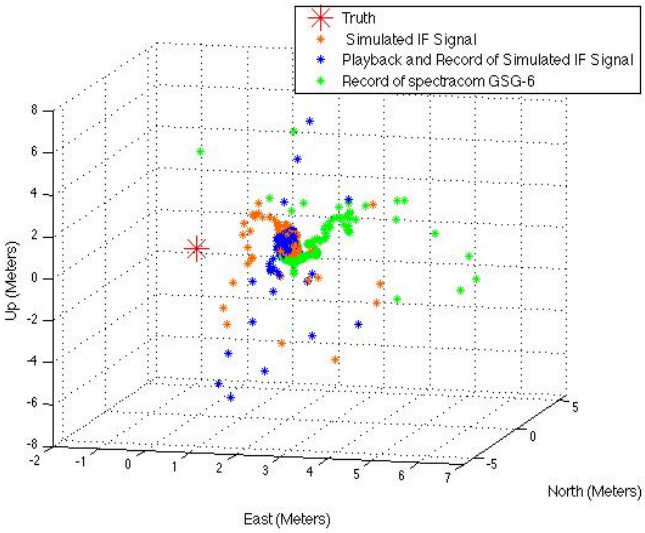

Figure 4.14: 3D ENU Position Results of Static Scenario

As seen in the figures above, each scenario provided similar position results. To further describe the positions plotted above, the position and velocity error from truth of each comparison scenario are illustrated in Figures 4.15 and 4.16.

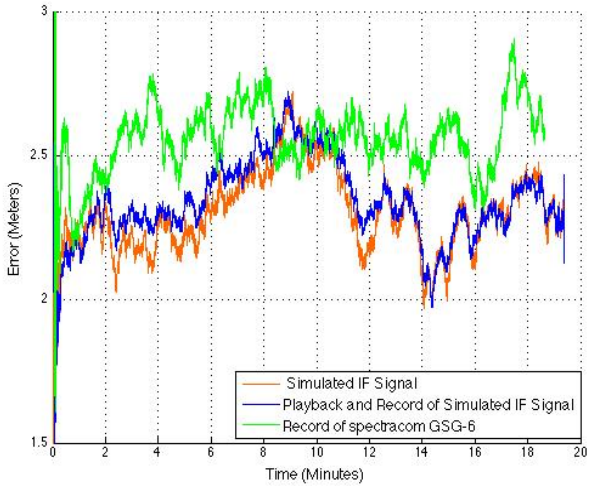

Figure 4.15: Position Error of Static Scenario

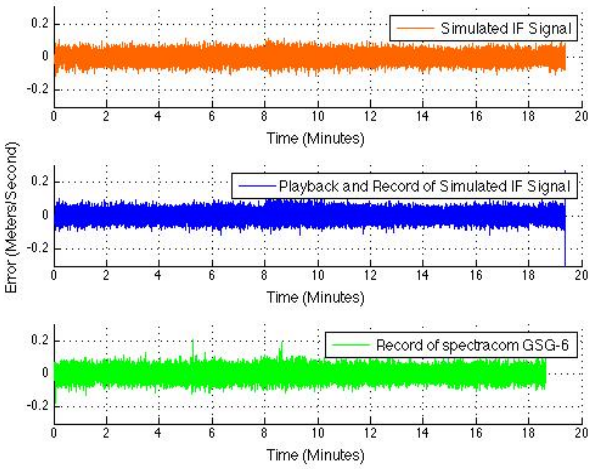

Figure 4.16: Velocity Error of Static Scenario

The simulated signals provide comparable position and velocity error trends as the signal generated by the COTS spectracom simulator. Interestingly, the simulated signal that was played back and recorded with a USRP provides a very similar position error trend as the

one provided by the original simulated IF signal. This means that the USRP playback and record functions do not significantly degrade the position solutions of the target receiver under normal conditions. The mean, standard deviation, and variance of the position and velocity errors are provided in Tables 4.3 and 4.4.

Table 4.3: Position Error Statistics

Value	Simulated	Playback and Record of Simulated	Record of spectracom GSG-6
Mean (m)	2.303	2.342	2.556
Std Dev (m)	0.142	0.141	0.120
Variance (m)	0.020	0.020	0.014

Table 4.4: Velocity Error Statistics

Value	Simulated	Playback and Record of Simulated	Record of spectracom GSG-6
Mean (m/s)	4.886×10^{-5}	3.531×10^{-5}	-3.484×10^{-5}
Std Dev (m/s)	0.026	0.026	0.029
Variance (m/s)	6.908×10^{-4}	6.921×10^{-4}	8.644×10^{-4}

Seen in Table 4.3, the average position error of the simulated signal and the playback and record of the simulated signal are slightly smaller than the signal generated by the hardware simulator. However, the hardware simulator produced a smaller standard deviation and variance of position error. The velocity errors are all near zero-mean with similar standard deviation and variance.

The software simulator is compared to the hardware simulator in the measurement domain by analysis of the pseuduranges and Doppler frequencies for each scenario. The pseudorange is the fundamental positioning measurement in GPS receivers and the Doppler frequency measurement is directly related to the accuracy of the carrier phase measurement in the tracking loops. The pseudorange and Doppler frequency verses time for each scenario of PRN 2 are provided below in Figures 4.17 and 4.18. The same plots for the remaining satellites available in the simulation are provided in Appendix A.2.

The pseudoranges and Doppler frequencies follow similar trends but a bias is noticeable in some of the pseudorange plots in Appendix A.2. This bias is caused because the software

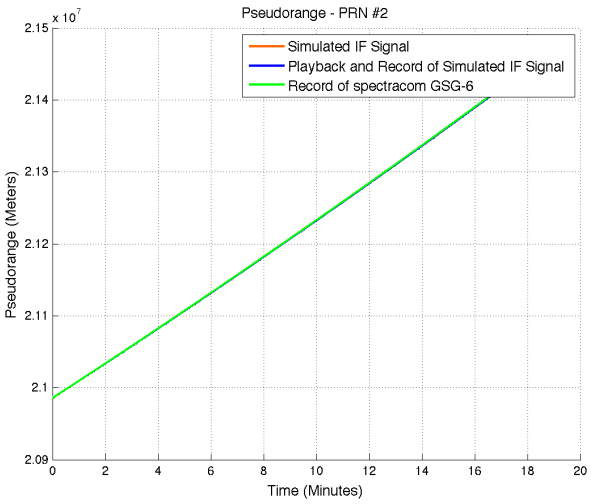

Figure 4.17: Pseudorange PRN 2

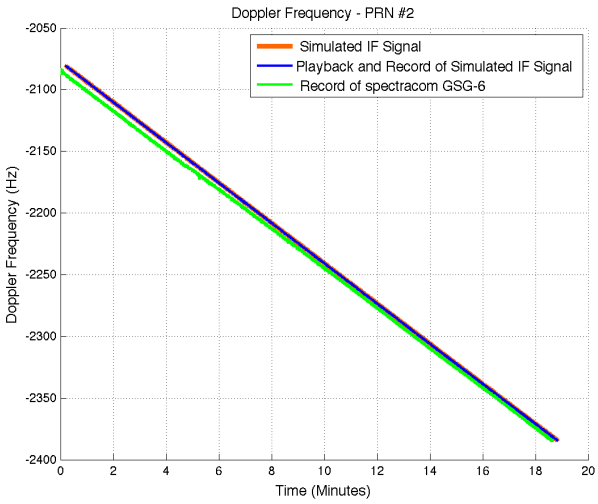

Figure 4.18: Doppler Frequency PRN 2

receiver outputs pseudoranges that are corrected with the estimate of receiver clock bias. The receiver estimated different clock biases for each scenario thereby causing the pseudorange bias. This correction is reversed and the corresponding raw pseudorange estimates are illustrated in Figures A.17 through A.25 in Appendix A.2. Also, it can be seen that the Doppler estimate is noisier for some satellites in the simulation. This is likely due to lower carrier-to-noise trends for the USRP record of the hardware simulator. The carrier-to-noise estimates for each scenario of PRNs 2 and 5 are illustrated in Figures 4.19 and 4.20. Plots of the carrier-to-noise ratios of the remaining satellites are provided in Figures A.26 through A.32 in Appendix A.2.

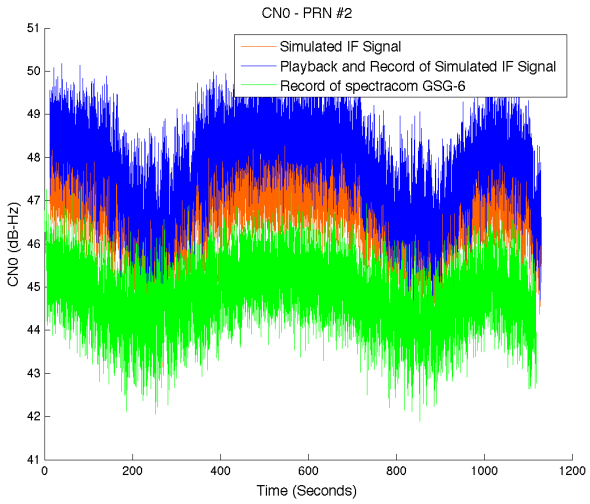

Figure 4.19: Carrier-to-Noise Ratio PRN 2

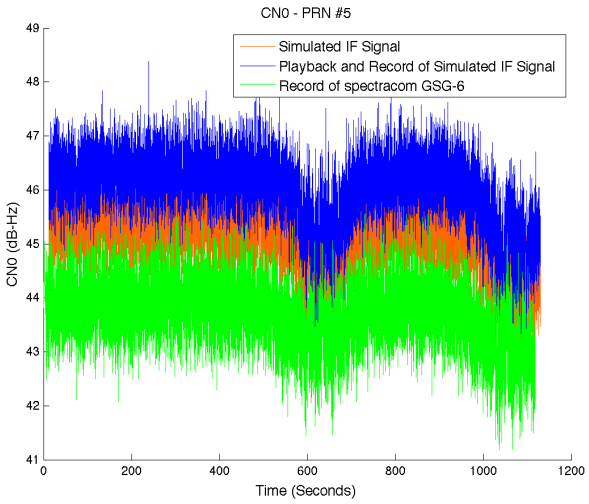

Figure 4.20: Carrier-to-Noise Ratio PRN 5

In general, the USRP record of the hardware simulator produced lower CN0 values than the USRP record of the simulated signal. Also, the playback and record of the simulated signal produced slightly higher CN0 values from the original simulated signal. This is most likely due to noise being clipped by the DAC or ADC in the USRP playback and record processes thereby causing lower noise floors and higher CN0 values. The differences between the software and hardware carrier-to-noise ratios could also be caused by unmodeled effects in the software simulation that are modeled in the hardware simulator. Fluctuations in CN0 values can also be seen for each scenario. These are not caused by the software simulator or the USRPs because they are experienced in each scenario.

The results provided in this section illustrate that the software simulator is comparable in performance to the hardware simulator. Also, the results show that the USRP can successfully playback and record GPS signals while maintaining a reasonable degree of positioning accuracy under normal conditions. This section also illustrates the attention to detail that must be taken when selecting the gain and attenuation when recording and playing back GPS signals with the USRP. The simulations in this section were not perfect in this aspect but a reasonable degree of accuracy was still maintained. The accuracy of the USRP playback or record can degrade very quickly if sufficient gain or attenuation is not applied

to ensure the signal power is in the range of the ADC or DAC. This is discussed further in Section 5.5.2.

4.4 Dynamic Trajectory Scenario

An IF signal for a dynamic receiver scenario was generated by the software simulator and analyzed with the software receiver provided in [3]. Additionally, the IF signal was played-back and recorded with USRPs and played-back to a Ublox hardware receiver. A block diagram describing this experiment is provided in Figure 4.21.

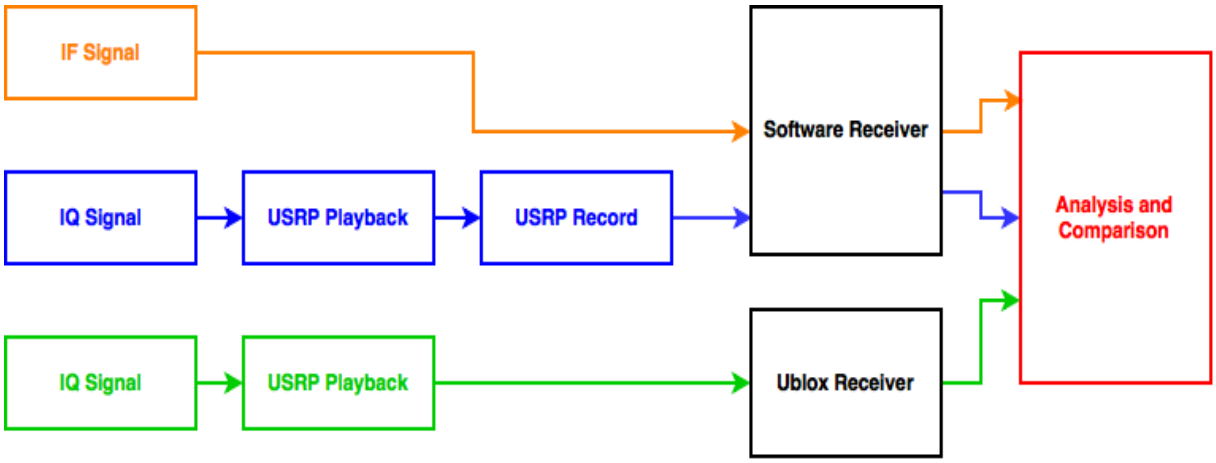

Figure 4.21: Dynamic Trajectory Analysis Scheme

The dynamic trajectory was created to replicate a path around the National Center for Asphalt Testing (NCAT) track in Opelika, Alabama. Measurements of the track dimensions were used to create a simple trajectory that travels around the track at a constant velocity. The dynamic scenario is described in Table 4.5.

Table 4.5: Dynamic Scenario Description

Date	11/30/2013
Time (UTC)	19:00
Duration (seconds)	210
Initial Latitude (°)	32.602236
Initial Longitude (°)	-85.489192
Initial Altitude (m)	193.645
Velocity (m/s)	25
Sampling Frequency (MHz)	12.5
Intermediate Frequency (MHz)	3.125
Carrier-to-noise Ratio	45
Ionosphere Model	None
Troposphere Model	None
Mask Angle (°)	5

The final positions measured by the software receiver and the Ublox receiver are provided in the WGS-84 and the corresponding local ENU frame in Figures 4.22 through 4.27.

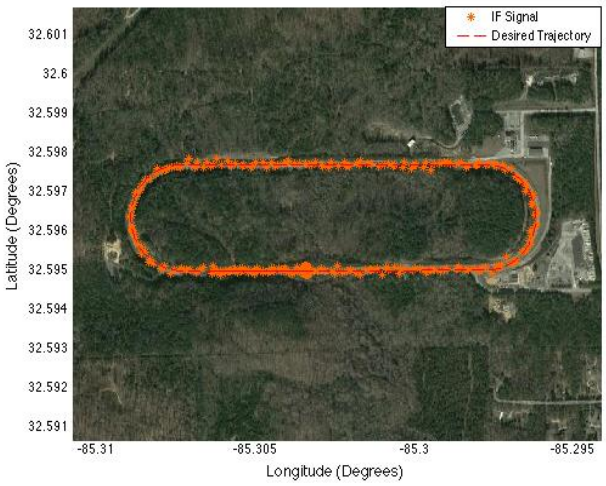

Figure 4.22: WGS-84 Position of Dynamic Scenario - Simulated Signal

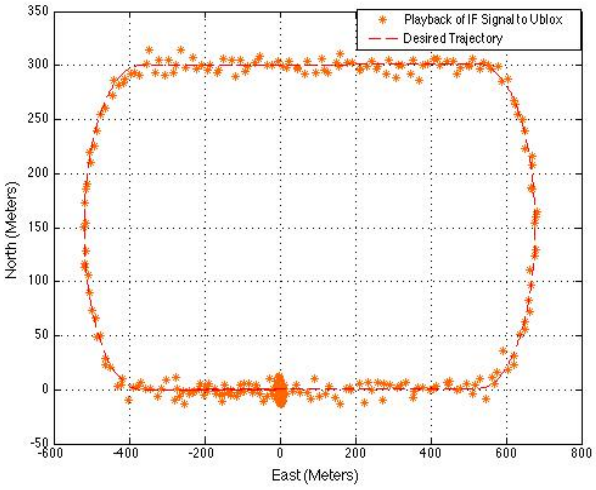

Figure 4.23: ENU Position of Dynamic Scenario - Simulated Signal

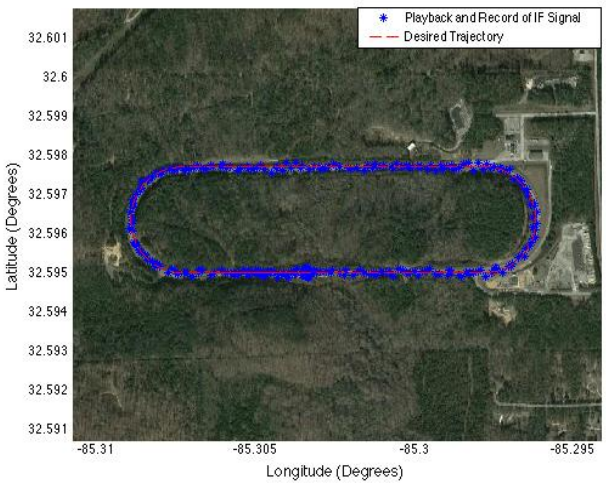

Figure 4.24: WGS-84 Position of Dynamic Scenario - Playback and Record of Simulated Signal

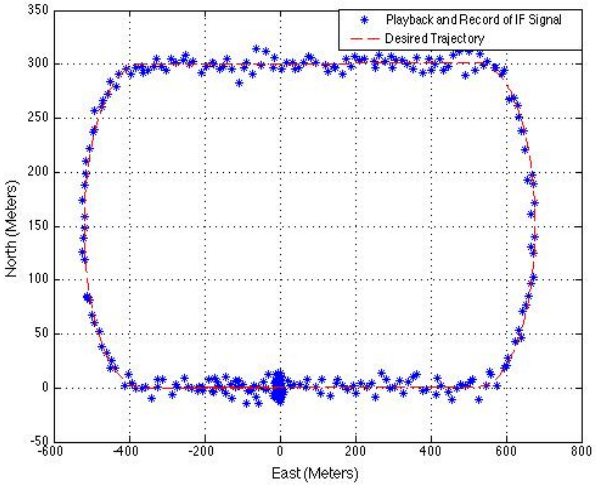

Figure 4.25: ENU Position of Dynamic Scenario - Playback and Record of Simulated Signal

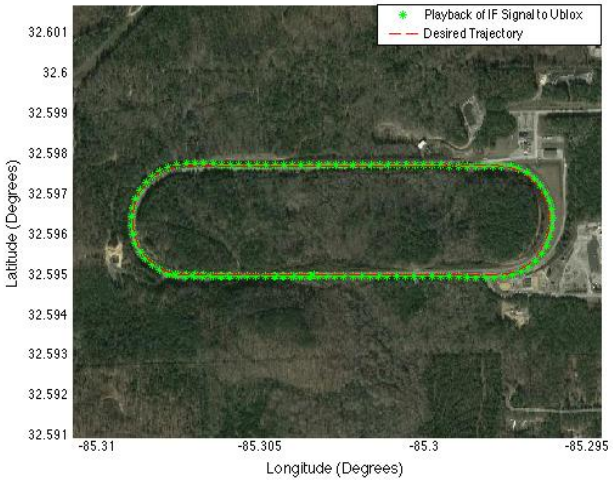

Figure 4.26: WGS-84 Position of Dynamic Scenario - Playback of Simulated Signal to Ublox

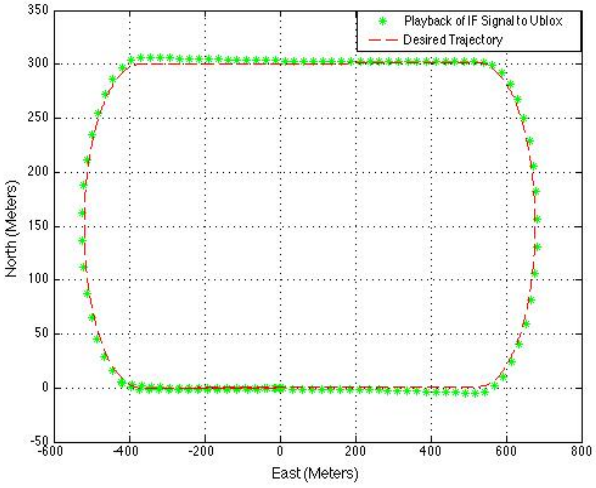

Figure 4.27: ENU Position of Dynamic Scenario - Playback of Simulated Signal to Ublox

The measured trajectory for each comparison scenario followed the desired trajectory within the accuracy limits of the GPS L1 CA code receiver used to analyze the data. Therefore this chapter has shown that the software simulator can provide useful GPS signal data for basic dynamic trajectories and the USRP is capable of capturing the desired dynamics in playback and record. To this point, only simple trajectories have been simulated by the software simulator but more complex trajectory models can be implemented if necessary int the future.

Chapter 5

Interference Mitigation Development and Testing

5.1 Introduction

GPS interference is defined as any electromagnetic source that interacts with the GPS signals and disrupts the process of estimating position and time [14]. This thesis details the design and application of a software GPS signal simulator capable of generating GPS signals along with man-made interference called jamming. Jamming is defined as intentional transmission of RF energy to hinder a navigation service by masking GPS signals with noise [14]. Since signals from GPS satellites reach Earth at a very low power (around -160 dBW) additional interference above the thermal noise floor can affect GPS receiver acquisition, tracking, and positioning. Interference saturates the ADC converter in the front-end of GPS receivers and causes the real GPS signal to be quantized at lower levels or completely lost in the presence of strong interference. The GAVLAB has developed a software receiver capable of running multiple types of digital interference mitigation techniques that utilize multiple antennas to mitigate interference on GPS signals. The software simulator gives the user full control of the simulation environment and access to truth values which are beneficial when developing interference mitigation techniques in the software receiver. The jammer-to-signal ratio is described in Section 5.2, jammer models are detailed in Section 5.3, an introduction to the interference mitigation techniques is provided in Section 5.4, and test results are provided in Section 5.5.

5.2 Jammer-to-Signal Ratio

The signal-to-noise and carrier-to-noise ratios described in Section 3.3.2 are deteriorated in the presence of interference. Interference algorithms are evaluated based on their performance at certain jammer-to-signal ($\frac{J}{S}$) ratios. In the software simulator, the jammer power is calculated based on the signal power calculated in Section 3.3.2 and a user specified $\frac{J}{S}$ ratio. The jammer power in Watts for jammer j, sample n, and antenna a is calculated for a true signal-to-noise simulation without accounting for a specific receiver bandwidth using Equation (5.1)

$$P_{j_W}^a(n) = 10^{\left(\frac{P_{N_{dB}} + \frac{S}{N} + \frac{J}{S}}{10}\right)} \tag{5.1}$$

If a specific front-end bandwidth is to be modeled the jammer power is calculated using Equation (5.2)

$$P_{j_W}^a(n) = 10^{\left(\frac{N_0 + \frac{C}{N_0} + \frac{J}{S}}{10}\right)} \tag{5.2}$$

where the noise PSD N_0 is calculated using Equations (3.40 – 3.42) in Section 3.3.2. The methods of calculating jammer power described in the section provide a user with simple means of simulating jammers with a desired power compared to the simulated noise and GPS signal. More sophisticated jammer power models that vary with the line-of-sight distance from the simulated position to the simulated jammer position or on the gain figure of a specific antenna can be implemented to the simulator if necessary.

5.3 Jammers

GPS interference signals can be classified based on their carrier frequency and bandwidth compared to the GPS carrier frequency and bandwidth. In general, out-of-band interference refers to interference whose carrier frequency is near the GPS band and in-band interference

refers to interference whose carrier frequency is located within the GPS band. Out-of-band and in-band interference are described further by Equations (5.3 – 5.4)respectively [14].

$$f_{int} < f_{GPS} - \frac{BW_{GPS}}{2} \quad or \quad f_{int} > f_{GPS} + \frac{BW_{GPS}}{2} \tag{5.3}$$

$$f_{GPS} - \frac{BW_{GPS}}{2} < f_{int} < f_{GPS} + \frac{BW_{GPS}}{2} \tag{5.4}$$

Furthermore, in-band interference can be classified as narrowband (NB), wideband (WB), or continuous-wave (CW) [14]. NB interference occupies a bandwidth smaller than that of GPS, WB interference occupies a bandwidth comparable to that of GPS, and CW interference foucuses all of it's power at one frequency and is also called tone-interference [15]. Interference signals can also be swept across multiple frequencies and waveforms. An example of this is a chirp interference, which is a strong signal that is swept across a frequency range that interferes with the received frequency of all GPS satellites in the chirp bandwidth [14]. Jammers can also be pulsed and are classified by their pulse width, frequency, and duty cycle [14]. Three types of jammers are simulated for the tests in this thesis: A CW jammer, a pulsed-CW jammer, and chirp-CW jammer. The CW jammer is simulated using the direct path portion of the two-path CW model detailed in [15] and is illustrated in Equation (5.5)

$$J^a(n) = \sqrt{2P_J^a(n)}cos\Big(\frac{2\pi f_J n}{f_s} - \frac{2\pi R_D}{\lambda}\Big) \tag{5.5}$$

where f_J is the carrier frequency of the jammer and R_D is the the length of the direct path from the jammer to the receiver and is described by Equation (5.6).

$$R_D = \sqrt{R^2 + (h_J - h_R)^2} \tag{5.6}$$

where R is the line-of-sight distance between the jammer and receiver and h_J and h_R are the heights of the jammer and receiver respectively [15]. This model is further illustrated by Figure 5.1.

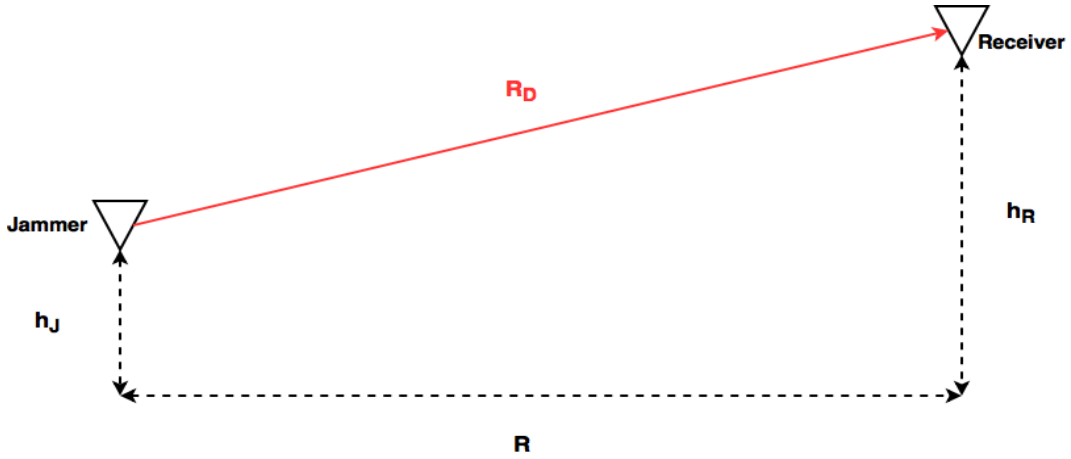

Figure 5.1: Jammer and Receiver Path

A pulsed version of the CW jammer is simulated by multiplying Equation (5.5) by a square wave of ones and zeros with the desired pulse width, frequency, and duty cycle. A simulated GPS L1 CA code signal with a CW jammer generated by Equation (5.5) is illustrated in the frequency domain with and without simulated thermal noise by Figures 5.2 and 5.3 respectively.

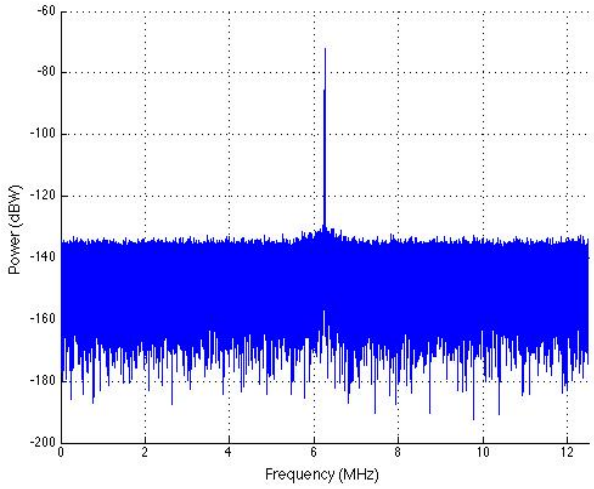

Figure 5.2: Frequency Domain of L1 GPS Signal with CW Jammer and Thermal Noise

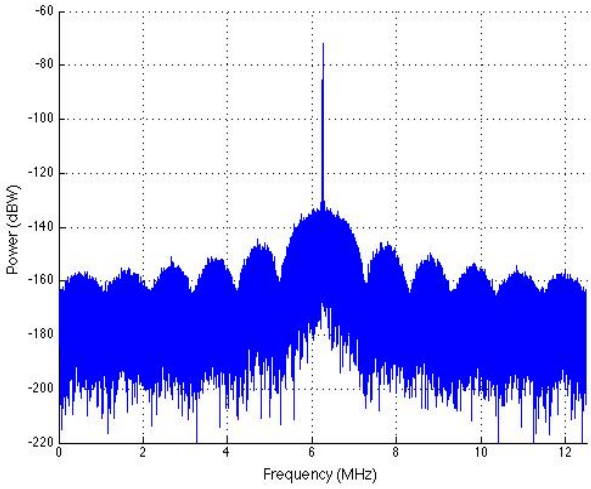

Figure 5.3: Frequency Domain of L1 GPS Signal with CW Jammer

The CW jammer was generated with the same carrier frequency as the simulated GPS signals and it can be seen that the power of the CW jammer in the frequency domain is focused at the IF of the simulated GPS signal. Frequency and time domain plots of a simulated GPS signal along with the pulsed-CW jammer are provided by Figures 5.5 and 5.4.

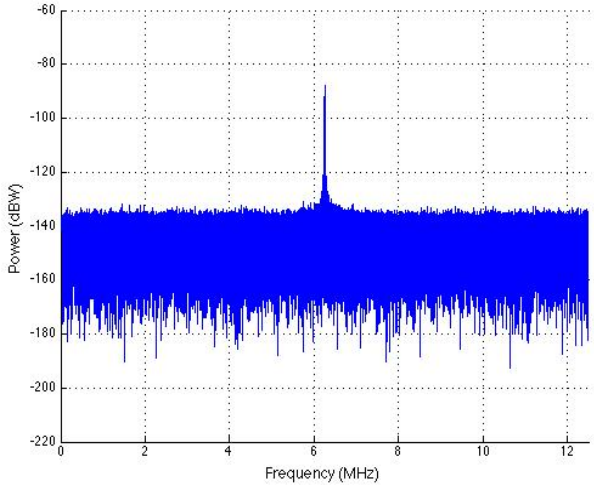

Figure 5.4: Frequency Domain of L1 GPS Signal with Pulsed CW Jammer

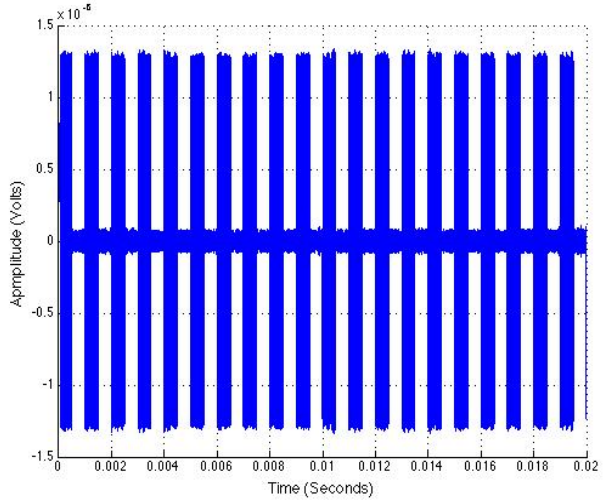

Figure 5.5: Pulsed CWI Jammer vs. Time

The frequency domain of the pulsed CW jammer is similar to that of the regular CW jammer and the pulsing effect can be seen in the time-domain plot. The chirp jammer was generated by varying the jammer frequency over a desired sweep bandwidth in a desired amount of time. A frequency domain plot of a simulated GPS signal with the chirp-CW jammer is provided in Figure 5.6.

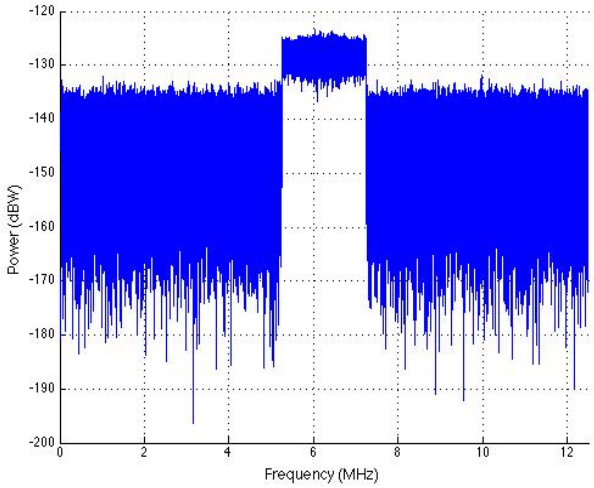

Figure 5.6: Frequency Domain of GPS Signal with Chirp Jammer

It can be seen that the power of the chirp-CW jammer is swept across multiple frequencies close to the GPS IF. The jammers described in this section are simple implementations that allow a user to develop and test interference mitigation algorithms in software receivers. More complex models can be implemented to the software simulator if desired. These jammer models were implemented into the software simulator and used to test mitigation techniques. Results from these tests are provided in Section 5.5.

5.4 Interference Mitigation Techniques

In the presence of interference, multiple-antenna array configurations can be used to combine the received signals to maximize the ratio between the true GPS signals and interference [11]. Essentially, these combinations allow the user to manipulate the gain pattern of the antenna array to optimally null interference or add amplification in the direction of the

desired signals. The antenna arrays, called controlled-radiation-pattern antennas (CRPA), are spaced in such a way that causes a phase difference between elements. With knowledge of the CRPA spacing, the incoming signals can be scaled and shifted to adversely or advantageously interfere with each other to null interference or place additional gain in the direction of the impeding GPS satellite signals. A simple way to combine the signals is to multiply the samples of the incoming signal by a complex weight and to sum across A antenna elements. This is illustrated by Equation (5.7) [11].

$$S(n) = \sum_{a=1}^{A} S_a(n) w_a(n) \tag{5.7}$$

The weighting scheme calculated in Equation (5.7) can be implemented into a software receiver to allow for digital processing of signals from a CRPA array. A high level block diagram of a software receiver is provided in Figure 5.7 [3].

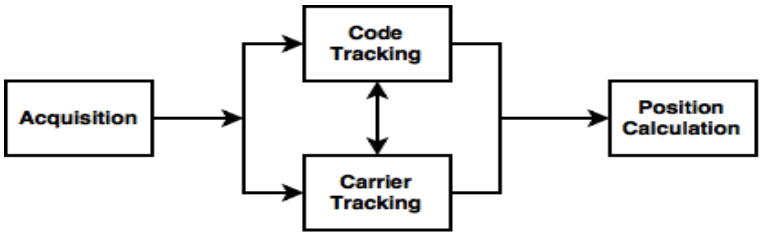

Figure 5.7: High-Level Block Diagram of a Software Receiver

Acquisition provides initial estimates of the code phase and carrier frequency which are tracked and refined in the tracking loops. The output of the tracking loops is used to decode the navigation data bits, calculate the pseudoranges, and calculate receiver position. In [11], the tracking loops are modified to include the complex weighting scheme which is used to implement different types of digital interference mitigation algorithms that utilize signals from CRPA arrays. The complex weighting scheme is added subsequent to the carrier wipe-off and prior to the code wipe-off in the software receiver tracking loops. The modified software receiver with the complex weighting scheme for multiple antennas is illustrated by Figure 5.8.

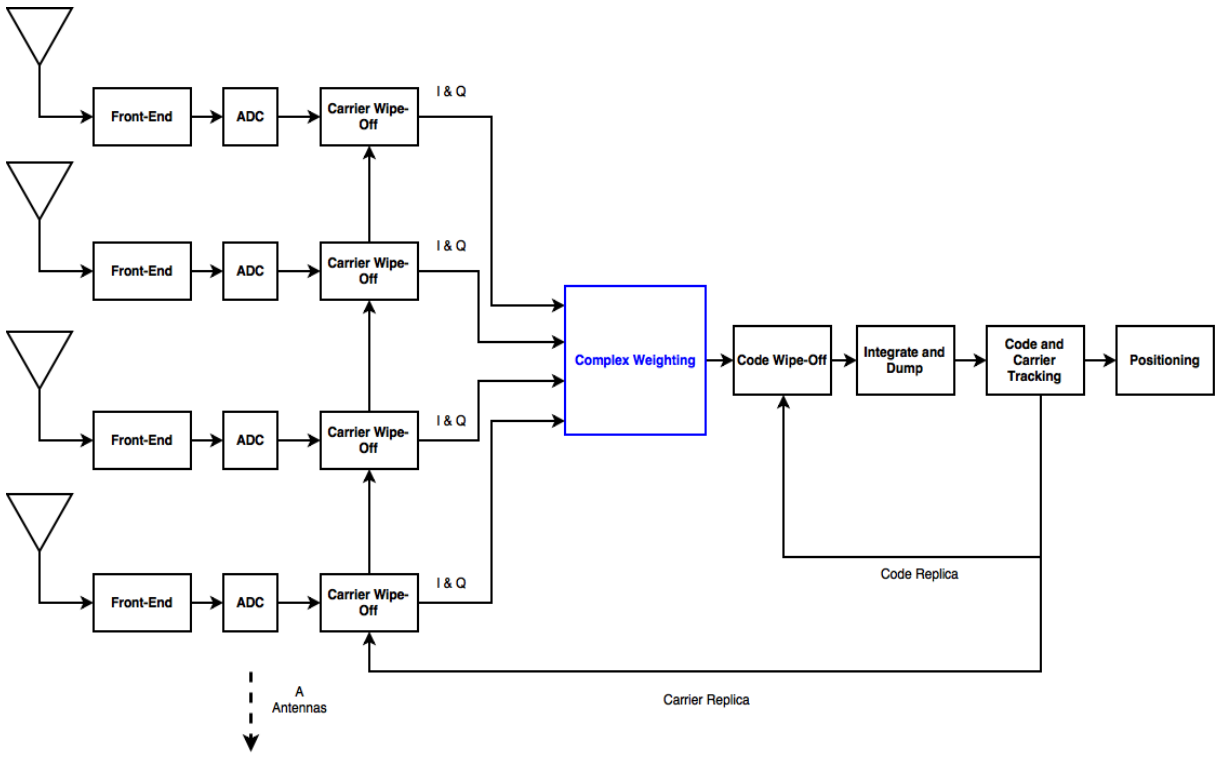

Figure 5.8: Antenna Array Software Receiver

The GAVLAB has modified the software receiver provided with [3] to include a complex weighting similar to the one detailed in [11]. The software receiver is used to analyze how interference mitigation techniques affect carrier phase positioning in dynamic jamming environments [31]. The software signal simulator detailed in this thesis is used to generate GPS signals in jamming environments for multiple-antenna arrays to support this analysis.

The GAVLAB has implemented various types of deterministic and adaptive interference weighting algorithms into the modified software receiver. Deterministic algorithms require prior knowledge of the jammer or satellite positions to work effectively and adaptive algorithms require no prior knowledge of the interference and signal environment. This thesis details two types of deterministic weighting algorithms, beam-forming and null-steering, and one type of adaptive approach, power-minimization. Beam-forming leverages knowledge of the antenna configuration and GPS satellite signal elevation and azimuth to create a digital look direction towards the visible satellites. Essentially, the phase of the reference antenna is

shifted to produce maximum gain in the direction of the incoming signal. The beamforming weight for each antenna is calculated using Equation (5.8) [11].

$$w_a(n) = e^{-j\left(2\pi \frac{\bar{e}_a \bullet \bar{r}_{s_k}}{\lambda_{L1}}\right)} \tag{5.8}$$

where $\bar{e}_a$ is the unit vector to antenna a from the the reference element in the local body frame and $\bar{r}_{s_k}$ is the unit-vector in the direction of satellite k. Null-steering leverages knowledge of the jammer location to produce a null in the direction of the jammer. The signals from the auxillary elements are phase shifted in such a way that when they are combined, they are in-phase with the reference element. This phase of each element is calculated using Equation (5.9)

$$\phi_a(n) = 2\pi \frac{\bar{e}_a \bullet \bar{r}_{aj}}{\lambda_{L1}} \tag{5.9}$$

where $\bar{r}_{aj}$ is the unit vector pointing from the antenna a to the jammer location. The signals from each antenna are phase shifted using Equation (5.10) and scaled using Equation (5.11)

$$a(n) = \left[1, e^{-j\phi_2(n)}, \ldots, e^{-j\phi_a(n)}\right]^T \tag{5.10}$$

$$b(n) = \left[1, -\frac{1}{A-1}, \ldots, -\frac{1}{A-1}\right] \tag{5.11}$$

where $a(n)$ and $b(n)$ are both A x 1 vectors. The null-steering weight is then calculated using Equation (5.12).

$$w(n) = a(n) \bullet b(n) \tag{5.12}$$

A desirable CRPA gain pattern is provided in Figure 5.9. The black lines represent GPS signals and the red line represents a jammer. It can be seen, in this case, that gain is

placed in the general direction of the GPS signals and a null is placed in the direction of the jammer.

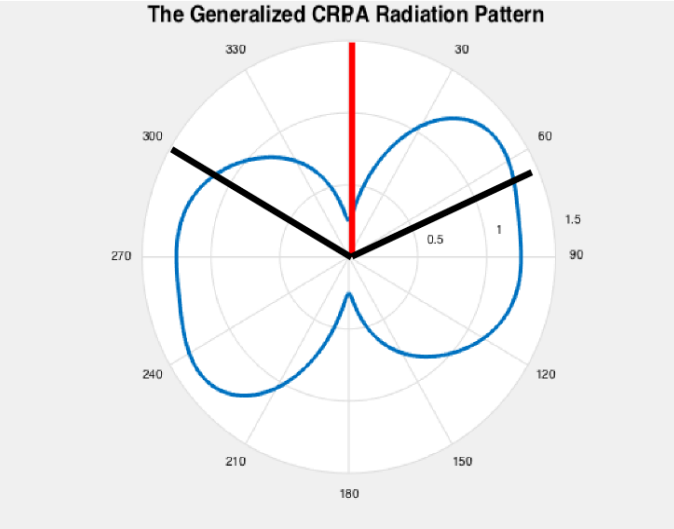

Figure 5.9: CRPA Gain Pattern

Power minimization leverages the fact that harmful interference is at a higher power than that of the GPS signals and thermal noise to optimally point nulls in the direction of interference [34]. Blind power-minimization points nulls in the direction of the interfering sources without any a-priori information of the jamming environment by minimizing the total output power of the combined signals. The optimum weight for blind power-minimization as detailed in [34] is provided by Equation (5.13).

$$w(n) = \frac{1}{\bar{\delta}_1^T \bar{R}_{xx}^{-1} \bar{\delta}} \bar{R}_{xx}^{-1} \bar{\delta} \tag{5.13}$$

where $\bar{\delta} = [1, 0, ..., 0]^T$ is the A x 1 constraint vector and $\bar{R}_{xx}$ is the A x A spatial autocorrelation matrix which is provided in Equation (5.14).

$$\bar{R}_{xx} = E\big[s(n)s^H(n)\big] \tag{5.14}$$

where $s(n)$ is the signal output of all the antenna elements at sample n and the superscript H is the Hermitian transpose. Extensive research has been conducted on different types of

interference mitigation similar to the ones described in this thesis. This section has provided a basic explanation of three different types of mitigation weighting algorithms. The digital beam-forming, null-steering, and power-minimization techniques are tested with signals generated by the software simulator for a simple CRPA configuration. The simulations are detailed and results are provided in Section 5.5.

5.5 Interference Testing

GPS signals along with the three types of jammers described in Section 5.3 for a single antenna generated by the software simulator were played-back and recorded with a USRP. The played-back and recorded interference signals were compared to the original simulated signal in the frequency and time domains to ensure the USRP can successfully capture each in the playback and record process. Additionally, the signals were compared via histogram to ensure the samples of the valid GPS signals and jammers were quantized correctly during the playback and record process. Results and discussion from these tests are provided in Section 5.5.2. To evaluate the ability to test interference mitigation algorithms with the software simulator and the USRPs, a 4-antenna CRPA interference scenario was generated and played-back and recorded simultaneously with two SURRPAS consisting of 8 total USRPs (4 playback and 4 record). The simulated and played-back and recorded interference signals were processed by the beam-forming, null-steering, and power-minimization algorithms in the modified software receiver. Results and discussions from these tests are provided in Section 5.5.3.

5.5.1 Further Motivation for GPS Interference Mitigation Testing

To further motivate the importance of GPS interference mitigation and to illustrate the benefits provided by a software GPS simulator capable of generating interference scenarios, a jammer simulation was generated and analyzed using a software receiver with no mitigation

and the power-minimization technique. The jammer simulation is described below in Table 5.1.

Table 5.1: Jammer Simulation

Number of Antennas	4
Antenna Spacing (meters)	0.09
Reference Antenna Latitude (°)	32.602236
Reference Antenna Longitude (°)	-85.489192
Reference Antenna Altitude (Meters)	201
Jammer Type	CW
Length (Seconds)	120
Time Jammer is Engaged (Seconds)	60
Jammer Duration (Seconds)	60
Intermediate Frequency (MHz)	1.25
Sampling Frequency (MHz)	5
Jammer Frequency (MHz)	1.25
$\frac{C}{N_0}$ (dB-Hz)	45
$\frac{J}{S}$ (dB)	50
Data Type	Signed 8-bit

Final positions and the corresponding position error provided by the software receiver with no mitigation and the power-minimization technique are illustrated below in Figures 5.10 through 5.12.

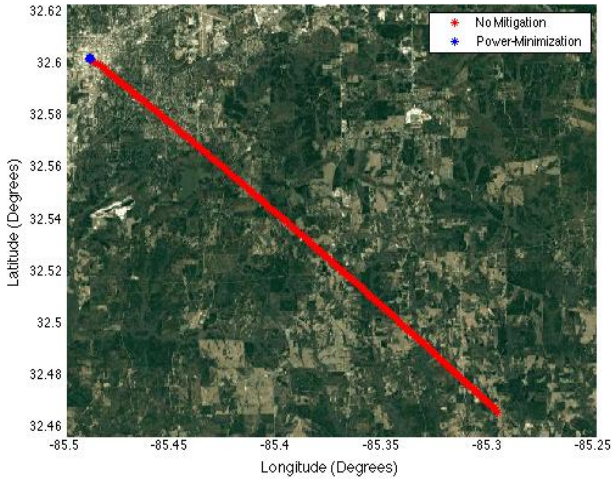

Figure 5.10: WGS-84 Positions of Jammer Simulation

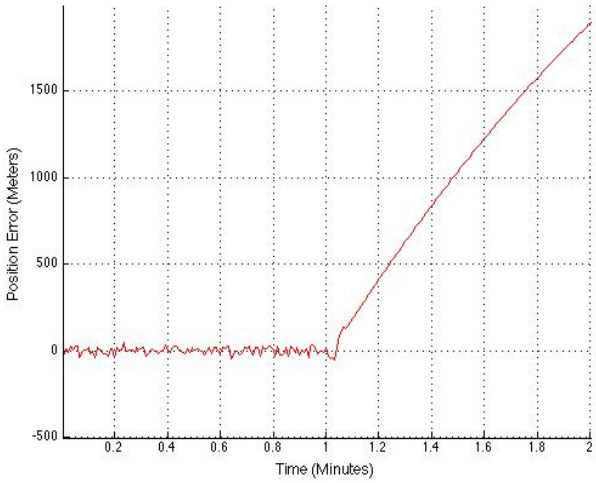

Figure 5.11: Position Error with No Mitigation

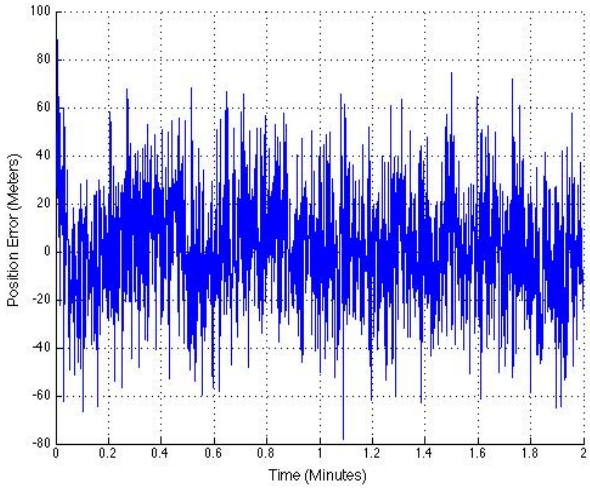

Figure 5.12: Position Error with Power-Minimization

It can be seen that the receiver was unable to calculate a reliable position solution when the jammer was turned on and no mitigation techniques were used. Conversely, power-minimization was able to provide accurate positions with the jammer engaged. This illustrates the importance that must be placed on developing and testing GPS interference mitigation techniques. The software GPS signal simulator detailed in this thesis is capable

of providing useful interference data needed to develop and test the mitigation techniques described in Section 5.4 and future novel techniques.

It should be noted that the the position solution distribution provided by the software receiver used in the preceding discussion depends on the sampling frequency of the signal. This is because the pseduorange measurement resolution depends on the sampling frequency [3]. This is illustrated in Equation (5.15).

$$\delta\rho = \frac{c}{f_s} \tag{5.15}$$

The pseudorange measurement resolution $\delta\rho$ in meters per sample is the speed of light c divided by the sampling frequency f_s. Higher sampling frequencies provided more accurate pseudorange measurements providing tighter position distributions. This is illustrated in Appendix A.3 by Figures A.33 and A.34. This software receiver is also used in the following sections.

5.5.2 Jammer Playback and Record Analysis

GPS signals along with the CW, pulsed-CW, and chirp-CW jammers described in Section 5.3 were generated by the software simulator and played-back and recorded with USRPs. The jammer simulations are described in Tables 5.2 through 5.4.

Table 5.2: CW Jammer Simulation

Length (Seconds)	180
Time Jammer is Engaged (Seconds)	90
Jammer Duration (Seconds)	90
Intermediate Frequency (MHz)	3.125
Sampling Frequency (MHz)	12.5
Jammer Frequency (MHz)	3.125
$\frac{C}{N_0}$ (dB-Hz)	45
$\frac{J}{S}$ (dB)	30
Data Type	Signed 8-bit

Table 5.3: Pulsed-CW Jammer Simulation

Length (Seconds)	180
Time Jammer is Engaged (Seconds)	90
Jammer Duration (Seconds)	90
Intermediate Frequency (MHz)	3.125
Sampling Frequency (MHz)	12.5
Jammer Frequency (MHz)	3.125
Duty Cycle (%)	50
Repetition Frequency (Hz)	1000
$\frac{C}{N_0}$ (dB-Hz)	45
$\frac{J}{S}$ (dB)	30
Data Type	Signed 8-bit

Table 5.4: Chirp-CW Jammer Simulation

Length (Seconds)	180
Time Jammer is Engaged (Seconds)	90
Jammer Duration (Seconds)	90
Intermediate Frequency (MHz)	3.125
Sampling Frequency (MHz)	12.5
Jammer Frequency (MHz)	3.125
Sweep Range (MHz)	2
Sweep Time (Seconds)	0.02
$\frac{C}{N_0}$ (dB-Hz)	45
$\frac{J}{S}$ (dB)	30
Data Type	Signed 8-bit

For baseline analysis, frequency domain plots and a histogram of the portion of the signals with no jammer are provided in Figures 5.13 through 5.16.

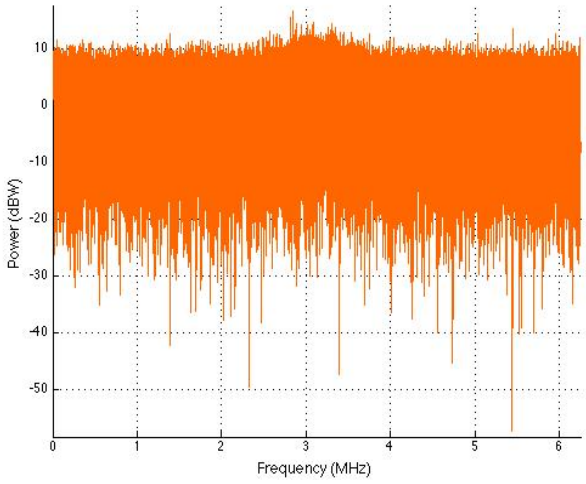

Figure 5.13: Frequency Domain of Clean Signal

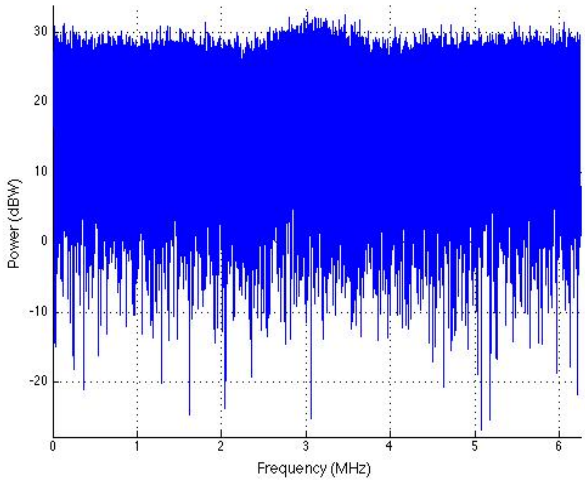

Figure 5.14: Frequency Domain of Playback and Record of Clean Signal

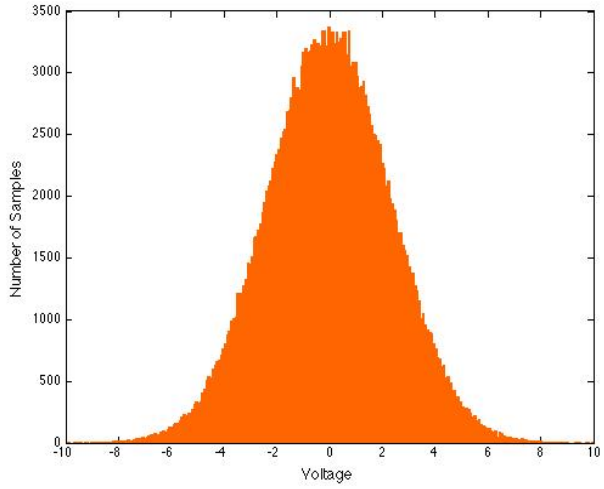

Figure 5.15: Histogram of Clean Signal

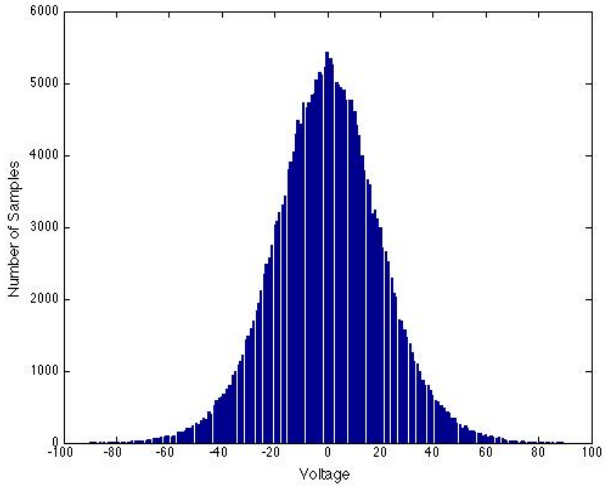

Figure 5.16: Histogram of Playback and Record of Clean Signal

The main lobe of the GPS CA code signal can be seen in both frequency domain plots. It should be noted that the playback signal is amplified using the signal conditioning rack in the SURRPAS to achieve optimum quantization when recording. This is why the played-back and recorded signal is shown at a slightly higher power than the simulated signal in the frequency domain plots. This is also why the voltages are higher in the histogram of the played-back and recorded signal. As, expected, the histograms of the clean signal resemble that of WGN because the noise on the signal is modeled as such.

Frequency domain and histogram plots of the CW jammer simulations are provided by Figures 5.17 through 5.20.

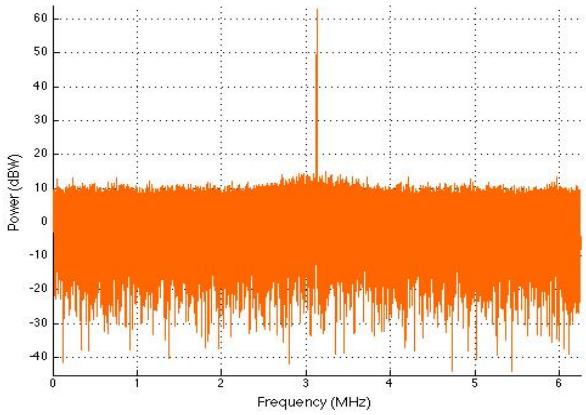

Figure 5.17: Frequency Domain of Simulated Signal with CW Jammer

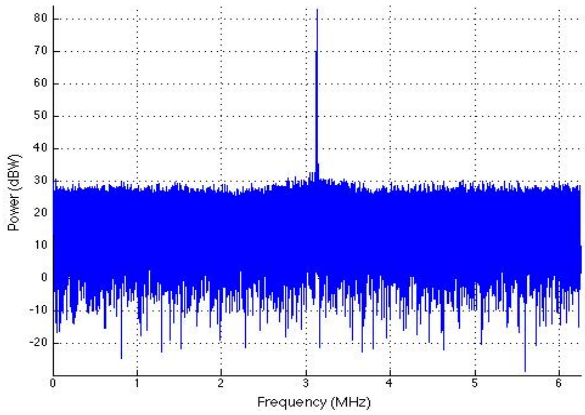

Figure 5.18: Frequency Domain of Playback and Record of Simulated Signal with CW Jammer

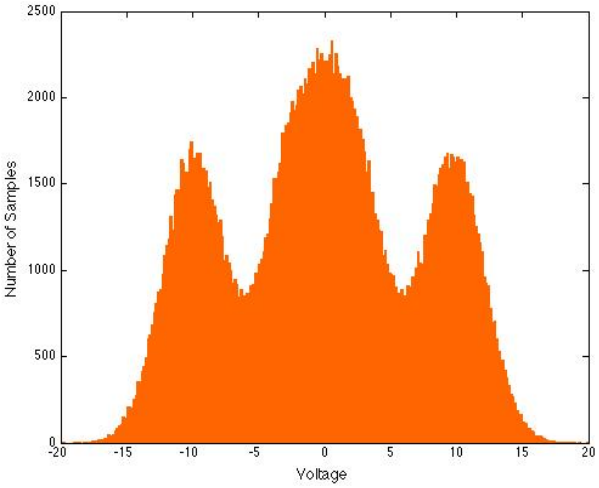

Figure 5.19: Histogram of Simulated Signal with CW Jammer

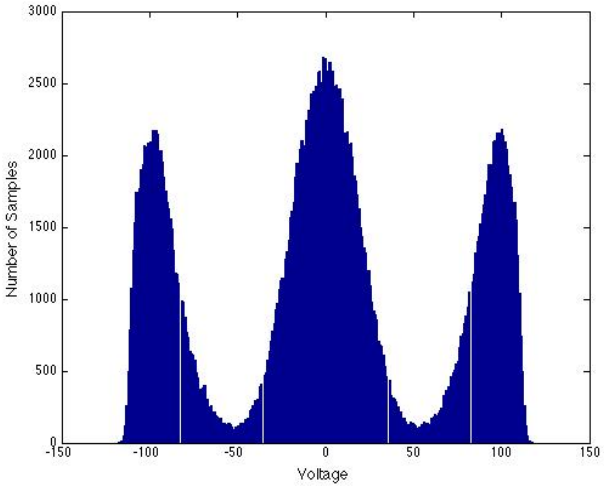

Figure 5.20: Histogram of Playback and Record of Simulated Signal with CW Jammer

The finite impulse-like effect of the CW jammer in the frequency domain was captured by the playback and record process. The histograms of the the CW jammer illustrate the careful process that must be used when playing-back and recording GPS signals with interference using the USRP. Since signed 8-bit integers were used when simulating, playing-back, and recording the signals, the equivalent dynamic range was $\pm$ 128 Volts with a resolution of

about 1 Volt. When recording GPS signals with the USRP, the desired signal must be amplified to an acceptable level to quantize the signal efficiently. When a jammer at a higher power is added to the signal, it will exceed the ±128 Volts if the composite signal is amplified too high. Since the goal is to record the GPS signal and jammer without inducing errors for future processing, this must be avoided. Saving 16-bit integers when recording or simulating provides a higher dynamic range but requires more computer processing power. Since a relatively low $\frac{J}{S}$ was simulated in this test, the signal and the CW-jammer were captured in the playback and record process. Signals with higher $\frac{J}{S}$ values will saturate the recording more easily and measures must be taken to avoid this issue.

Histograms of cases in which the composite interference signal was not amplified efficiently are provided in Figures 5.21 and 5.22.

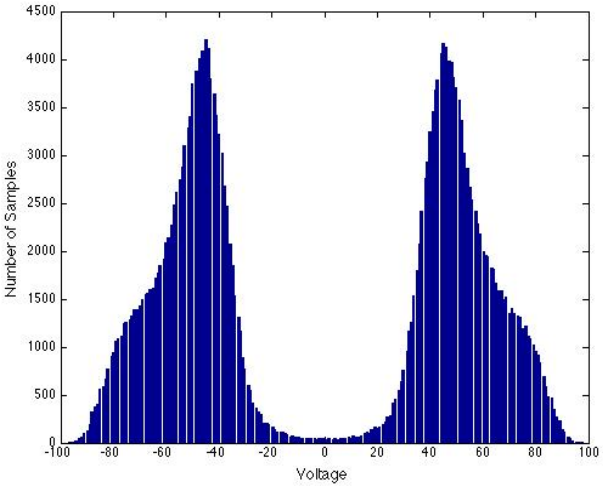

Figure 5.21: Histogram of Playback and Record of Signal with CW Jammer - Low Amplification

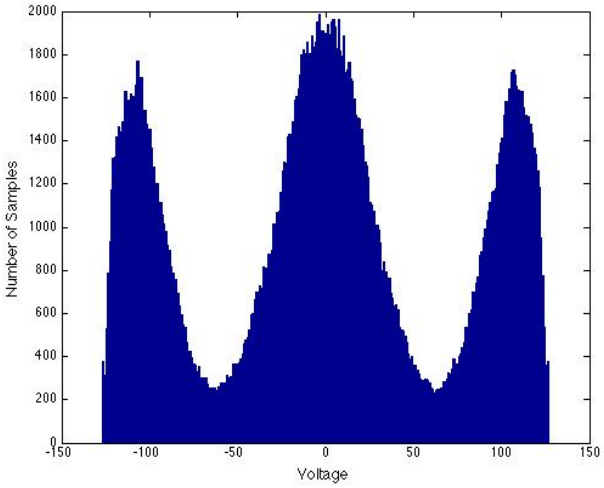

Figure 5.22: Histogram of Playback and Record of Signal with CW Jammer - High Amplification

In Figure 5.21, the signal is not amplified enough to save the samples of the real GPS signals at the lower levels of the 8-bit quantization scheme. The Gaussian-shaped portion of the histogram is clearly missing. This plot resembles the effect on the ADC in the front-end of a GPS receiver caused by the presence of CW interference [14]. Essentially, the real GPS signal is lost because the AGC in the receiver squeezes the incoming signals to match the

range of the ADC thereby "swamping" the ADC with the interference. In Figure 5.22, the signal was amplified too high causing the USRP to slightly clip the jammer signal. This is not desirable because the jammer will not be captured in the recording thereby affecting any future processing on that signal.

Frequency domain, time domain, and histograms of the simulated and played-back and recorded pulsed-CW jammer signals are provided in Figures 5.23 through 5.28.

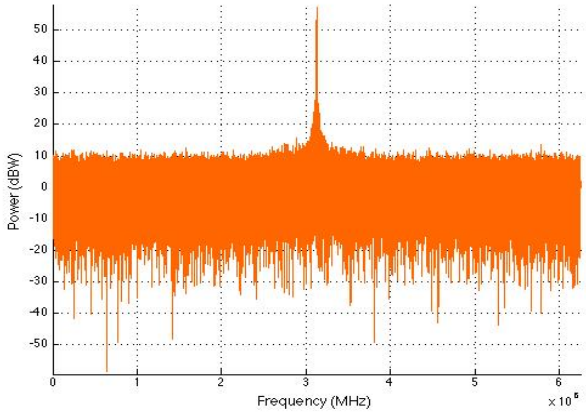

Figure 5.23: Frequency Domain of Simulated Signal with Pulsed-CW Jammer

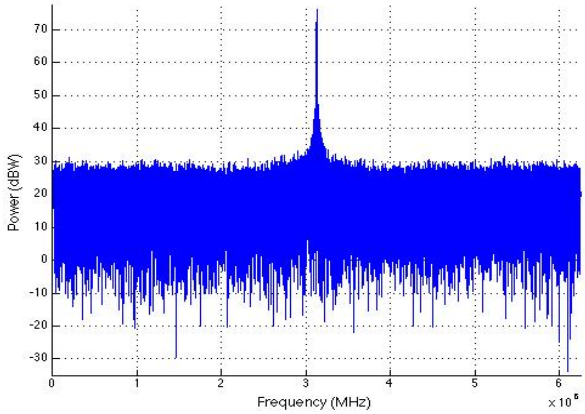

Figure 5.24: Frequency Domain of Playback and Record of Simulated Signal with Pulsed-CW Jammer

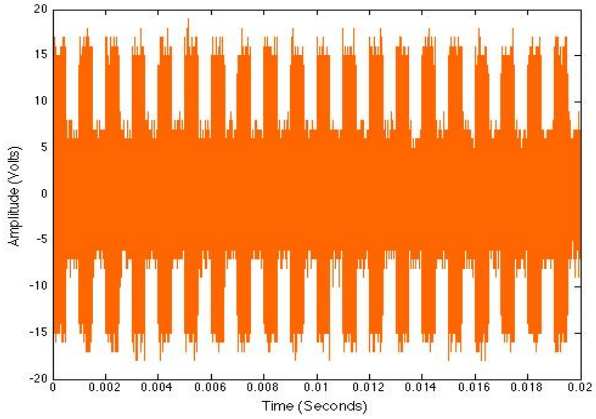

Figure 5.25: Time Domain of Simulated Signal with Pulsed CW-Jammer

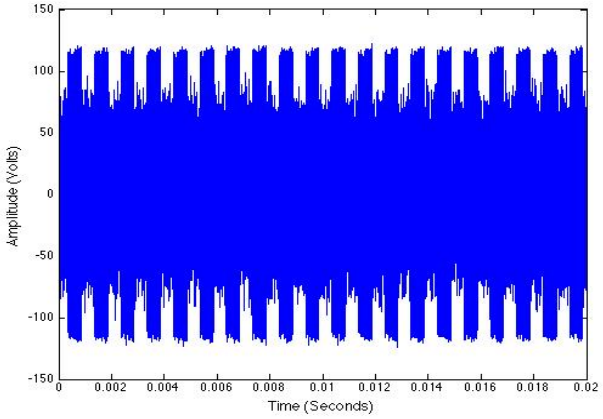

Figure 5.26: Time Domain of Playback and Record of Simulated Signal with Pulsed-CW Jammer

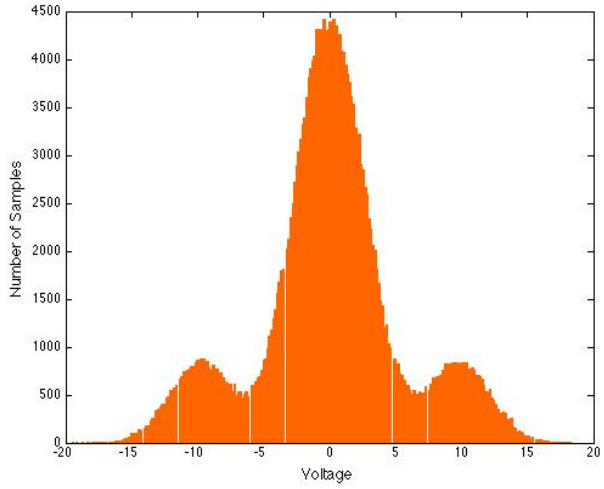

Figure 5.27: Histogram of Simulated Signal with Pulsed-CW Jammer

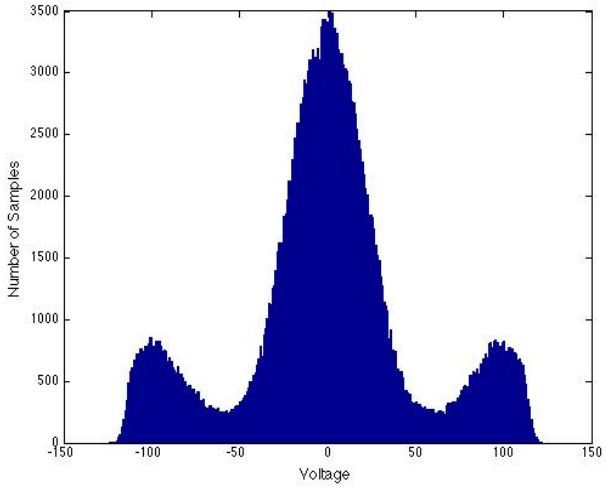

Figure 5.28: Histogram of Playback and Record of Simulated Signal with Pulsed-CW Jammer

The frequency domain characteristics of the pulsed-CW were captured by the USRP playback and record and the pulses in the time domain can be seen in both signals. The shape of the histograms are very similar, meaning optimum amplification was used when playing back and recording the simulated interference signal. The chirp-CW jammer is illustrated in the frequency and histogram domains by Figures 5.29 through 5.32.

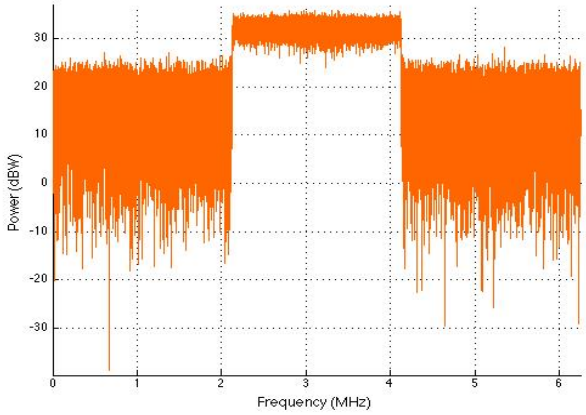

Figure 5.29: Frequency Domain of Simulated Signal with Chirp-CW Jammer

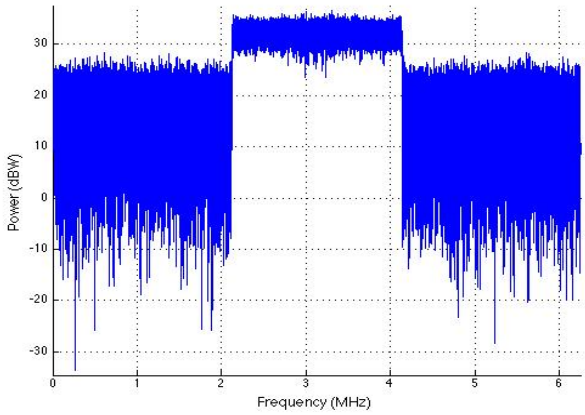

Figure 5.30: Frequency Domain of Playback and Record of Simulated Signal with Chirp-CW Jammer

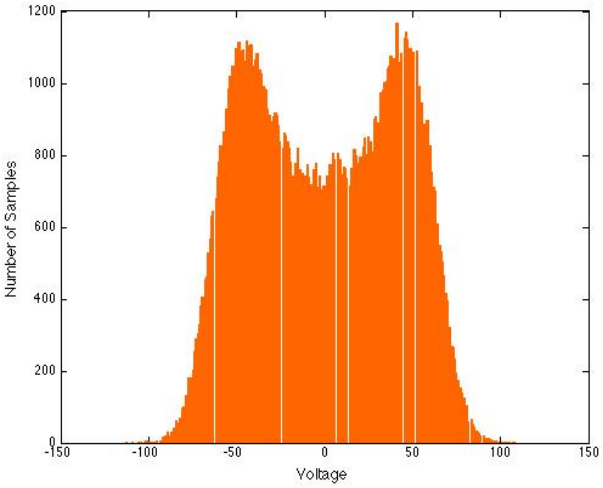

Figure 5.31: Histogram of Simulated Signal with Chirp-CW Jammer

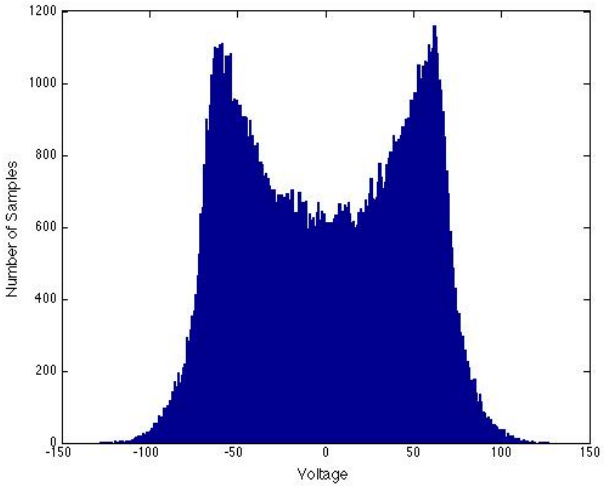

Figure 5.32: Histogram of Playback and Record of Simulated Signal with Chirp-CW Jammer

The USRP playback and record successfully captured the characteristics of the chirp jammer.

The results provided in this section give insight to GPS simulation and testing with a software simulator and the USRP. Careful attention must be taken when playing back and recording interference signals with the USRP. For playback, the signal should be simulated in such a way that the GPS signal and interference are quantized efficiently. When

recording, the signals must be amplified the optimum amount to ensure good quantization for the given ADC and data type. This analysis is also beneficial when preparing for live-sky interference testing events. These tests are very expensive to attend, so it is desirable to record the interference scenarios accurately. The software simulator can be used to test the recording platform with different jammer-to-signal ratios before these events to ensure accurate recordings of the interference signals at testing events.

5.5.3 CRPA Simulation

A 4-element CRPA interference scenario generated by the software simulator was used to test the beam-forming, null-steering, and power-minimization techniques in the software receiver described in Section 5.4. The interference scenario is described in Table 5.5 and the positions of the reference antenna and jammer are provided in Figures 5.33 and 5.34.

Table 5.5: 4-Element CRPA Interference Simulation

Number of Antennas	4
Antenna Spacing (Meters)	0.09
Reference Antenna Latitude (°)	32.602236
Reference Antenna Longitude (°)	-85.489192
Reference Antenna Altitude (Meters)	201
Jammer Type	Continuous-Wave
Jammer Latitude (°)	32.600612
Jammer Longitude (°)	-85.490659
Jammer Altitude (Meters)	210
Length (Seconds)	180
Time Jammer is Engaged (Seconds)	90
Jammer Duration (Seconds)	90
Sampling Frequency (MHz)	5
Intermediate Frequency (MHz)	1.25
Jammer Frequency (MHz)	1.25
$\frac{C}{N_0}$ (dB-Hz)	45
$\frac{J}{S}$ (dB)	30
Data Type	Signed 8-bit

The simulated CRPA signals were played-back and recorded simulataneously with 2 SURRPAS (8 total USRPs). A picture of the setup is provided in Figure 5.35.

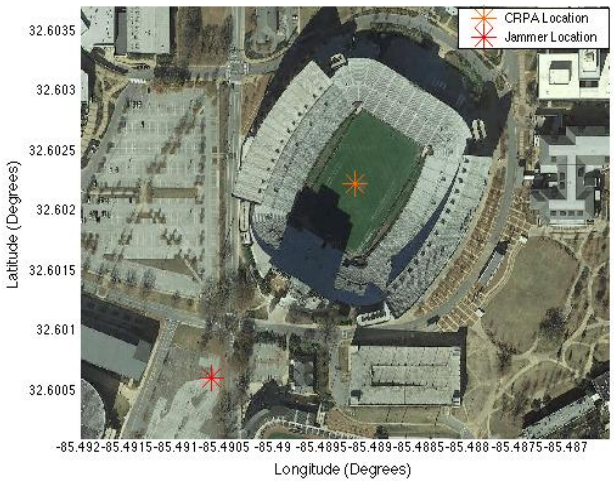

Figure 5.33: CRPA and Jammer Positions - LLA

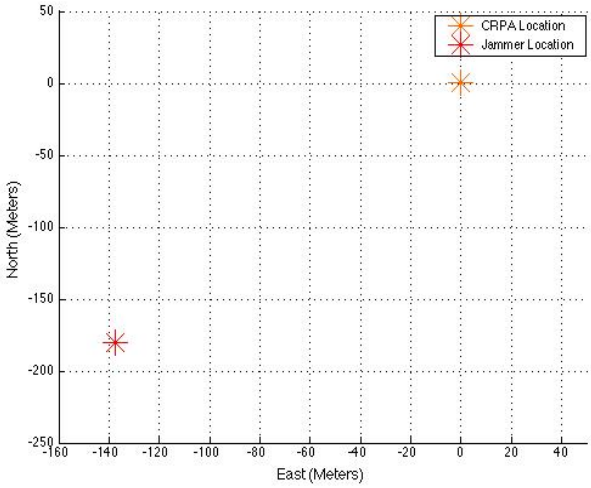

Figure 5.34: CRPA and Jammer Positions - ENU

Figure 5.35: USRP CRPA Playback and Record Setup

In this setup, SURRPAS # 1 was used for playback and SURRPAS # 2 was used for recording. A laptop was used to SSH into and control each host computer. Due to unanticipated malfunctions of both GPSDOs in the driver-USRPs in SURRPAS # 2, only one host computer was used to drive the 4-element USRP record. Because of this, a maximum

sampling frequency of 5 MHz was used to avoid data overflow. The playback SURRPAS was driven by two host computers and each USRP was started synchronously via the current GPS time provided from live-sky GPS signals. All timing was provided externally via the GPS-driven Octoclock and all pertinent cables were the same length. The simulated and played-back and recorded signals for each element of the CRPA simulation are shown in the frequency and histogram domain below in Figures 5.36 through 5.51. These plots correspond to the portion of the signals when the CW-jammer is engaged.

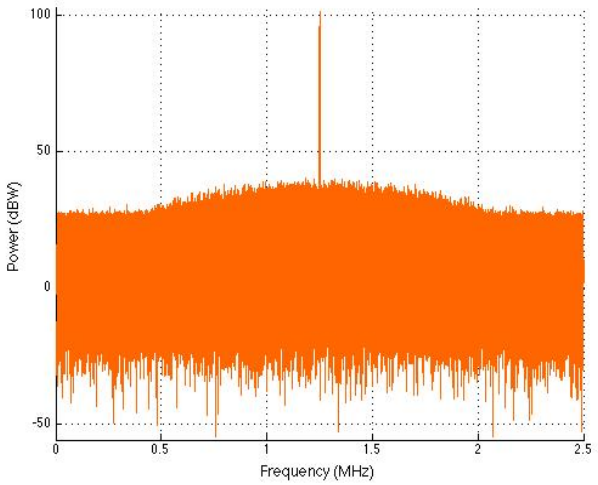

Figure 5.36: Frequency Domain of CRPA Simulation - Antenna 1

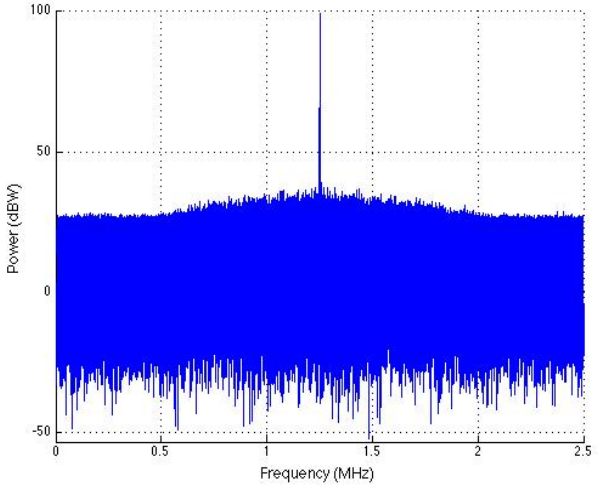

Figure 5.37: Frequency Domain of CRPA Simulation -TXRX - Antenna 1

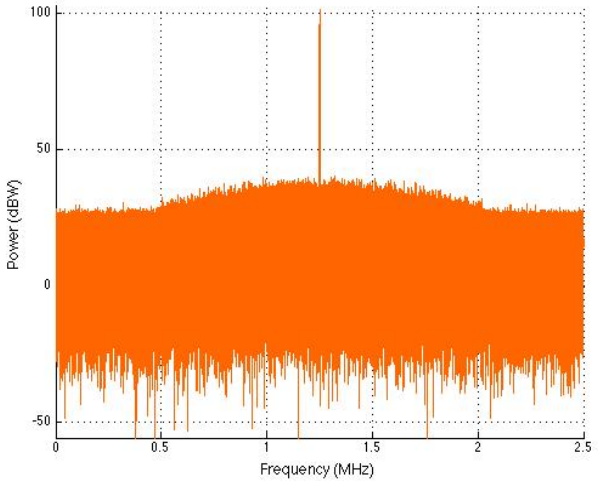

Figure 5.38: Frequency Domain of CRPA Simulation - Antenna 2

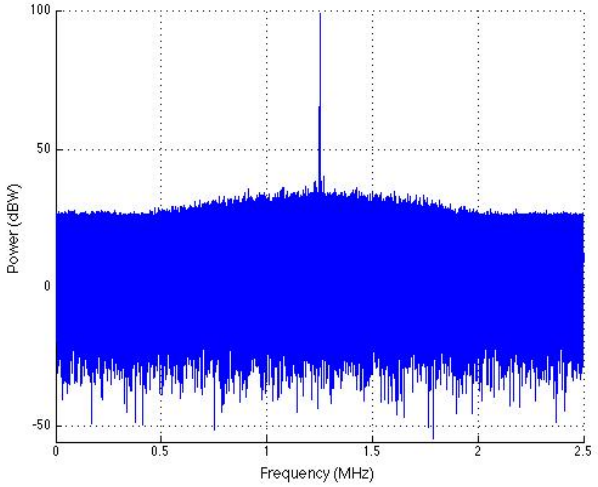

Figure 5.39: Frequency Domain of CRPA Simulation -TXRX - Antenna 2

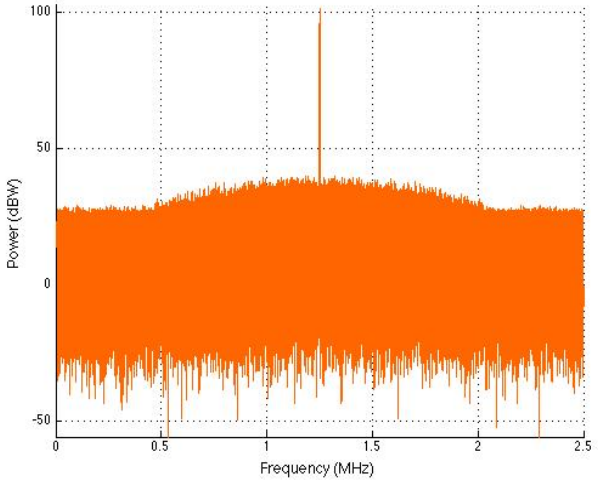

Figure 5.40: Frequency Domain of CRPA Simulation - Antenna 3

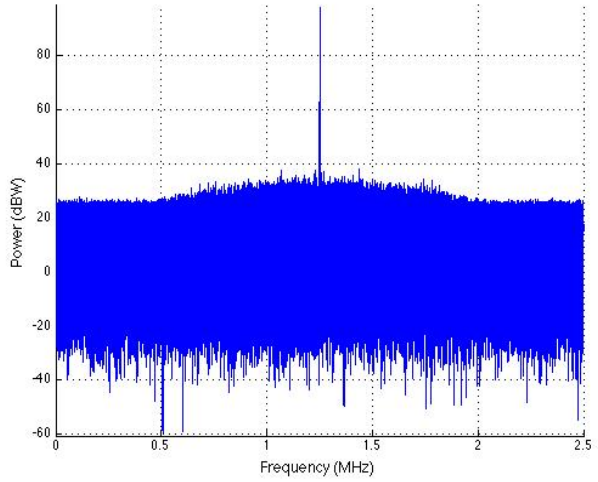

Figure 5.41: Frequency Domain of CRPA Simulation -TXRX - Antenna 3

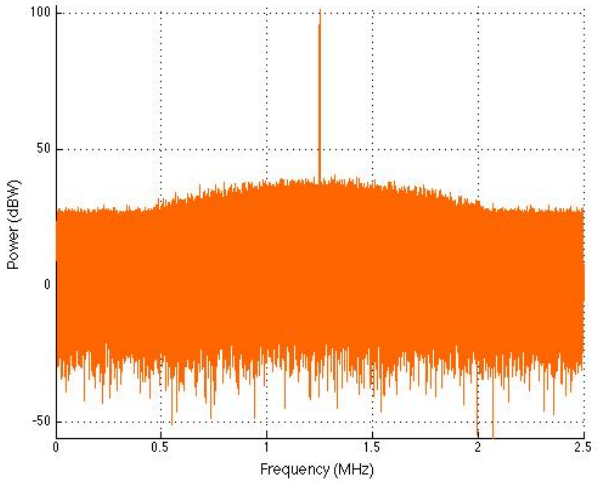

Figure 5.42: Frequency Domain of CRPA Simulation - Antenna 4

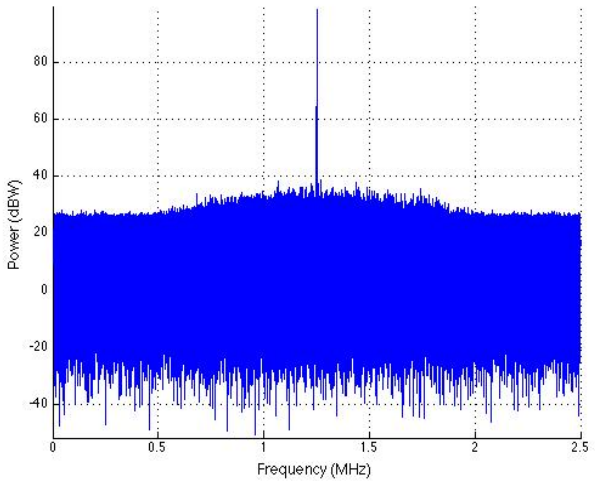

Figure 5.43: Frequency Domain of CRPA Simulation -TXRX - Antenna 4

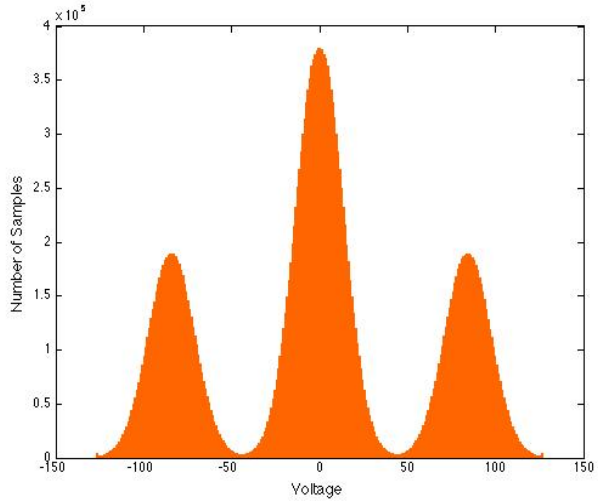

Figure 5.44: Histogram of CRPA Simulation - Antenna 1

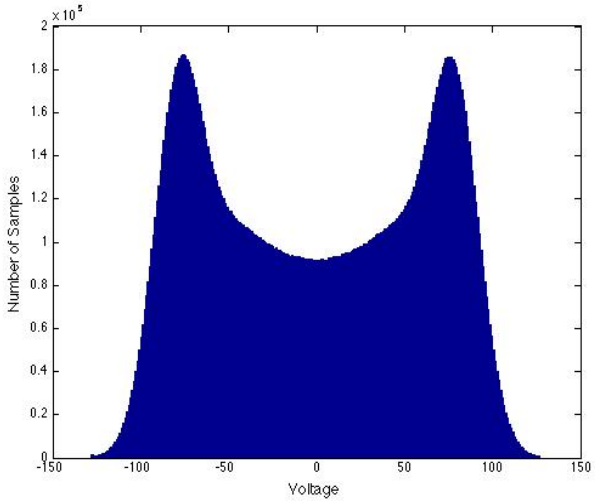

Figure 5.45: Histogram of CRPA Simulation -TXRX - Antenna 1

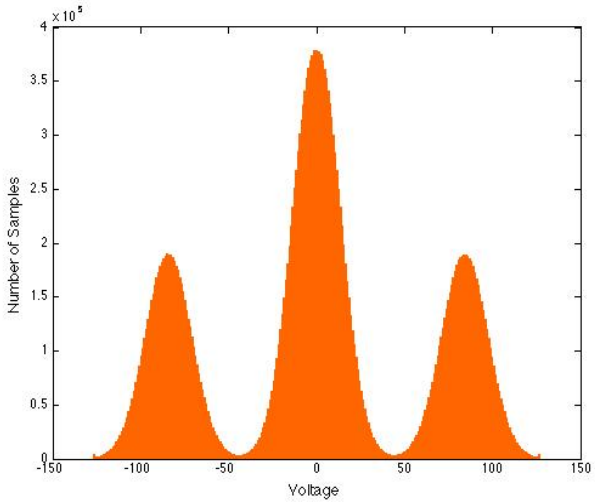

Figure 5.46: Histogram of CRPA Simulation - Antenna 2

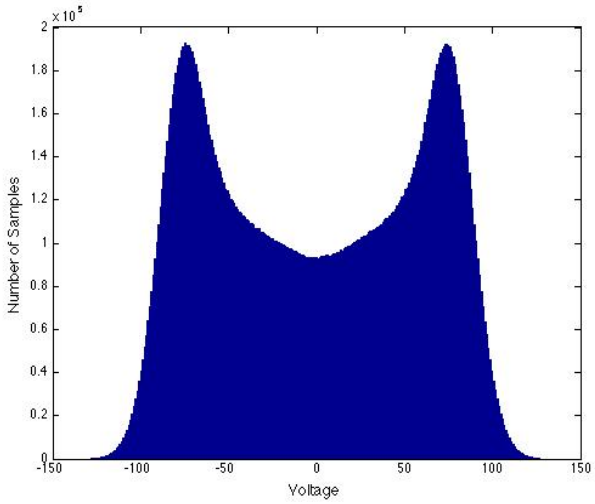

Figure 5.47: Histogram of CRPA Simulation -TXRX - Antenna 2

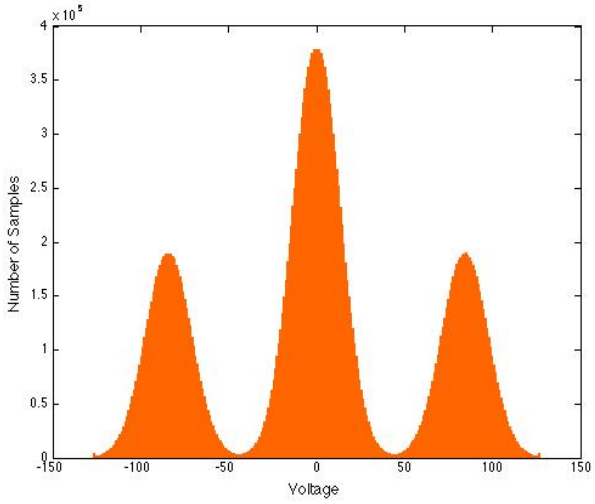

Figure 5.48: Histogram of CRPA Simulation - Antenna 3

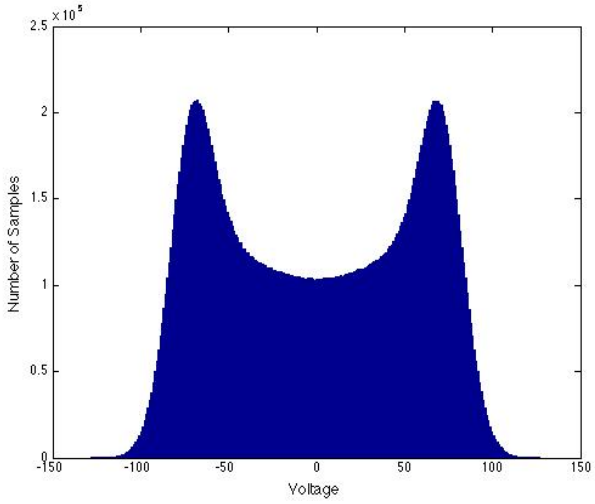

Figure 5.49: Histogram of CRPA Simulation -TXRX - Antenna 3

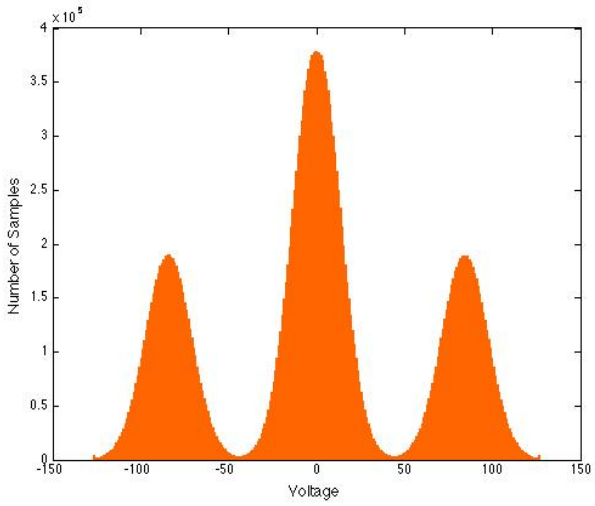

Figure 5.50: Histogram of CRPA Simulation - Antenna 4

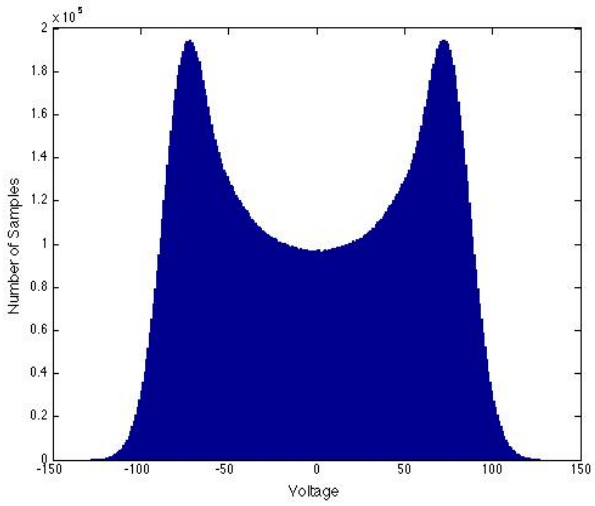

Figure 5.51: Histogram of CRPA Simulation -TXRX - Antenna 4

It can be seen that the histograms of the played-back and recorded signal do not match the ones of the original simulated signals exactly. The portion of the histogram representing the true GPS signal is not completely lost like in Figure 5.21 but it is not as prevalent as the original simulated signal. To illustrate a possible cause of this effect, time-domain plots illustrating the jammer-start period of the simulation are provided below in Figures 5.52 through 5.59.

It can be seen from the histograms and time-domain plots that the simulated signals were generated as close to saturation as possible to ensure the maximum USRP playback accuracy was reached. Even so, power fluctuations can be seen in the time-domain plots of the played-back and recorded signals. These fluctuations are likely the cause of the histogram discrepancies. Essentially, because of the power fluctuations, the jammer samples are squeezed closer to the true GPS signals, causing the loss of the distinctive WGN-like shape in the histograms.

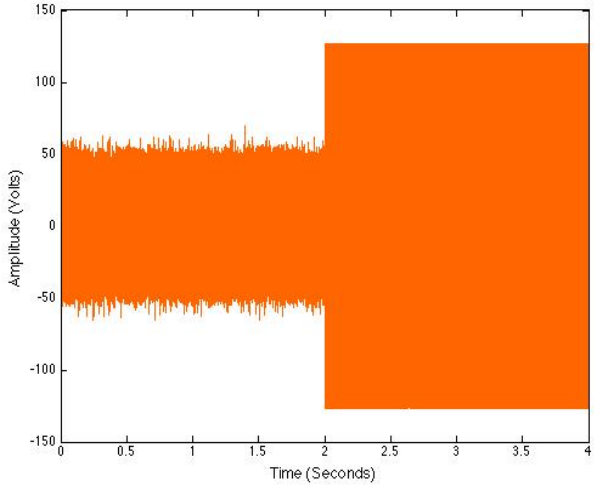

Figure 5.52: Time Domain of CRPA Simulation - Antenna 1

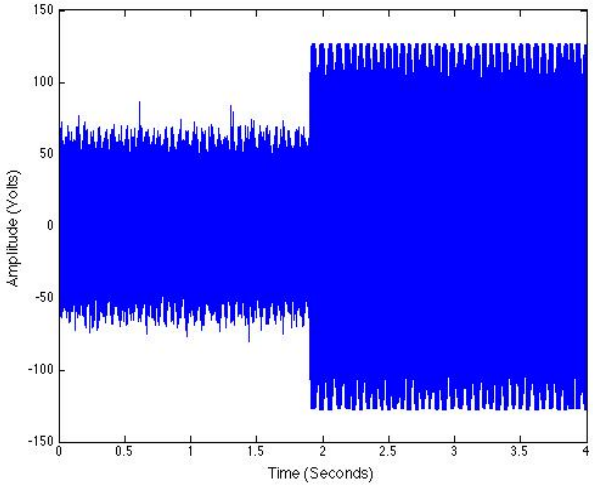

Figure 5.53: Time Domain CRPA of Simulation -TXRX - Antenna 1

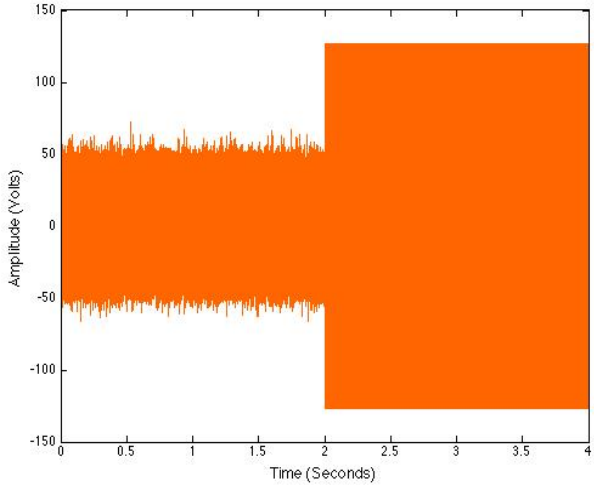

Figure 5.54: Time Domain of CRPA Simulation - Antenna 2

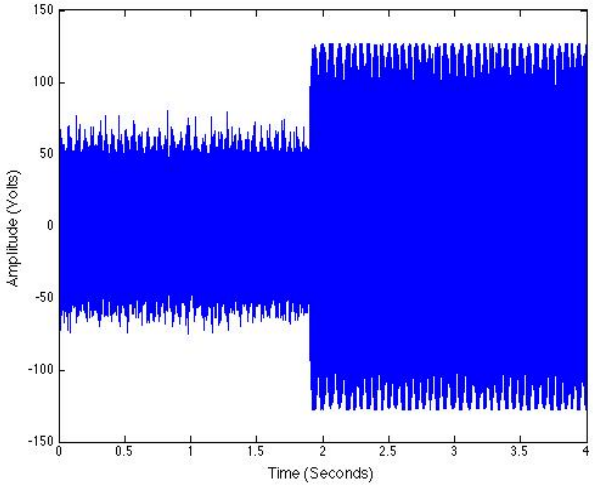

Figure 5.55: Time Domain CRPA of Simulation -TXRX - Antenna 2

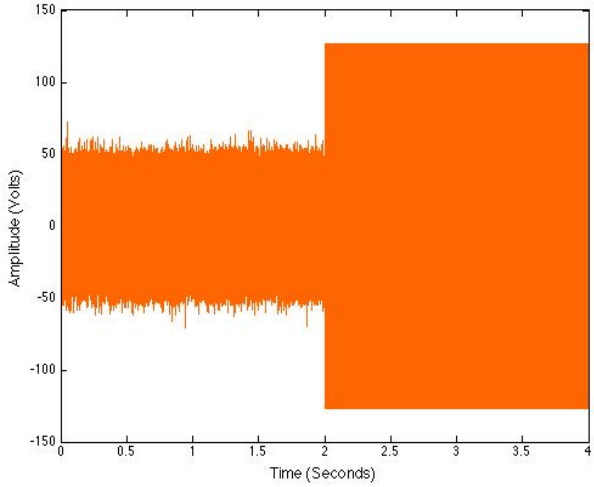

Figure 5.56: Time Domain of CRPA Simulation - Antenna 3

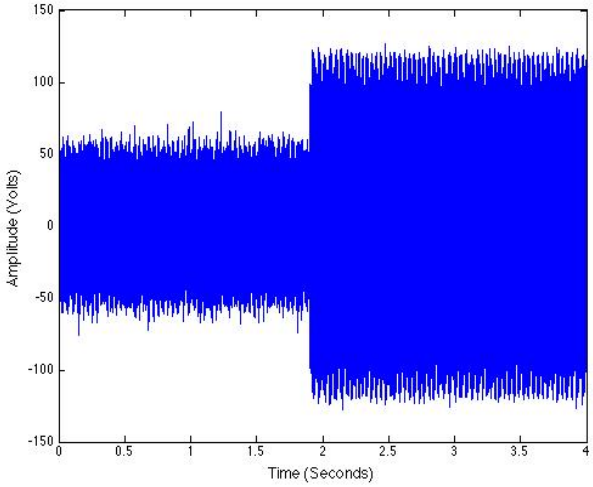

Figure 5.57: Time Domain CRPA of Simulation -TXRX - Antenna 3

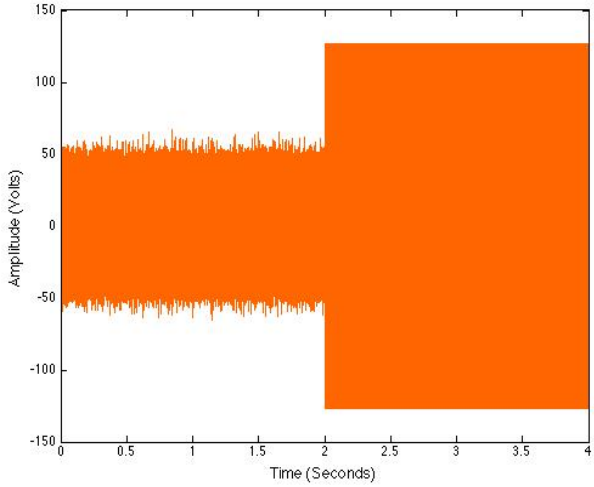

Figure 5.58: Time Domain of CRPA Simulation - Antenna 4

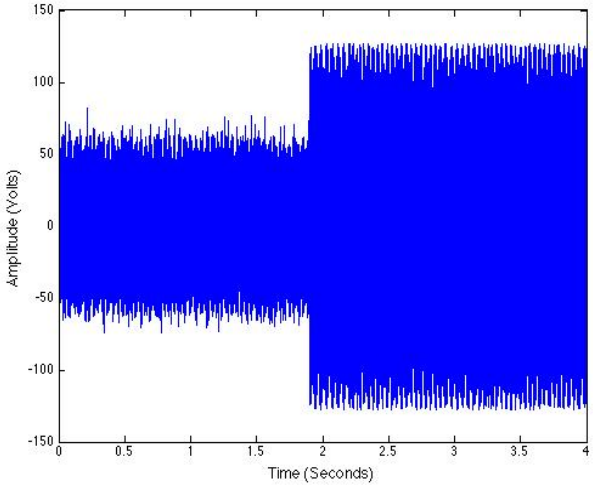

Figure 5.59: Time Domain CRPA of Simulation -TXRX - Antenna 4

The simulated and played-back and recorded CRPA signals were analyzed using the multiple-antenna software receiver with the power-minimization, beam-forming, and null-steering techniques. In-phase, early, prompt, and late tracking correlators and Doppler frequency estimates of selected satellites provided by power minimization are illustrated below in Figures 5.60 through 5.65.

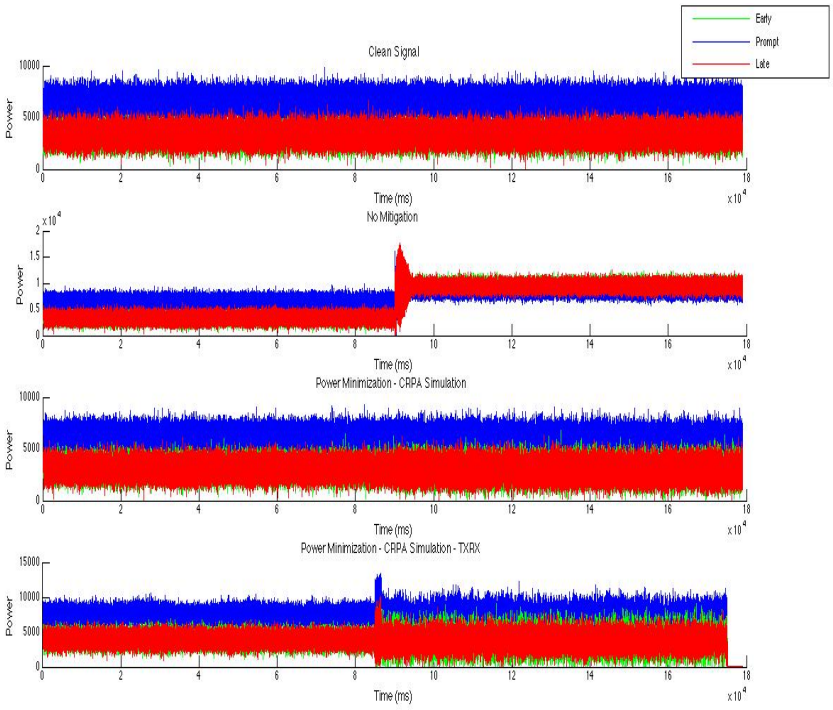

Figure 5.60: In-Phase Tracking Correlators - Power-Minimization - PRN 26

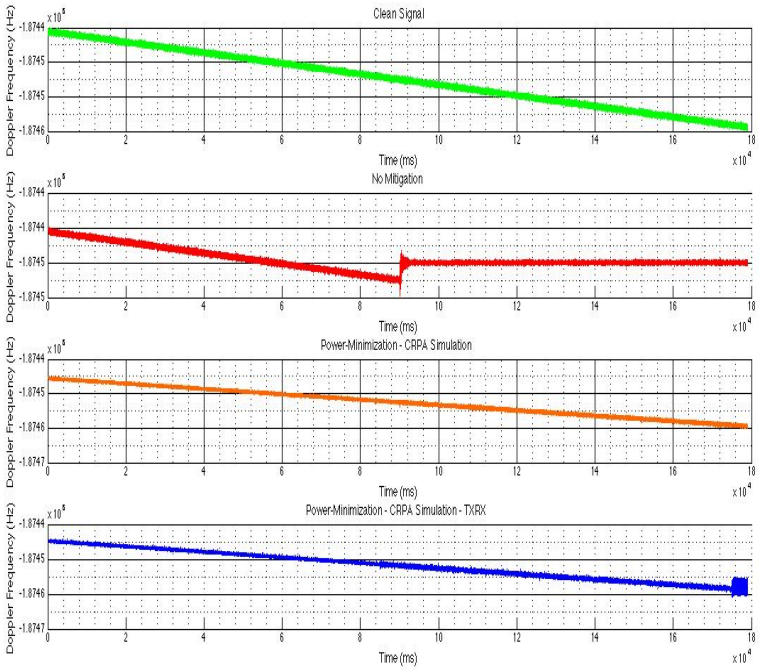

Figure 5.61: Doppler Frequency - Power-Minimization - PRN 26

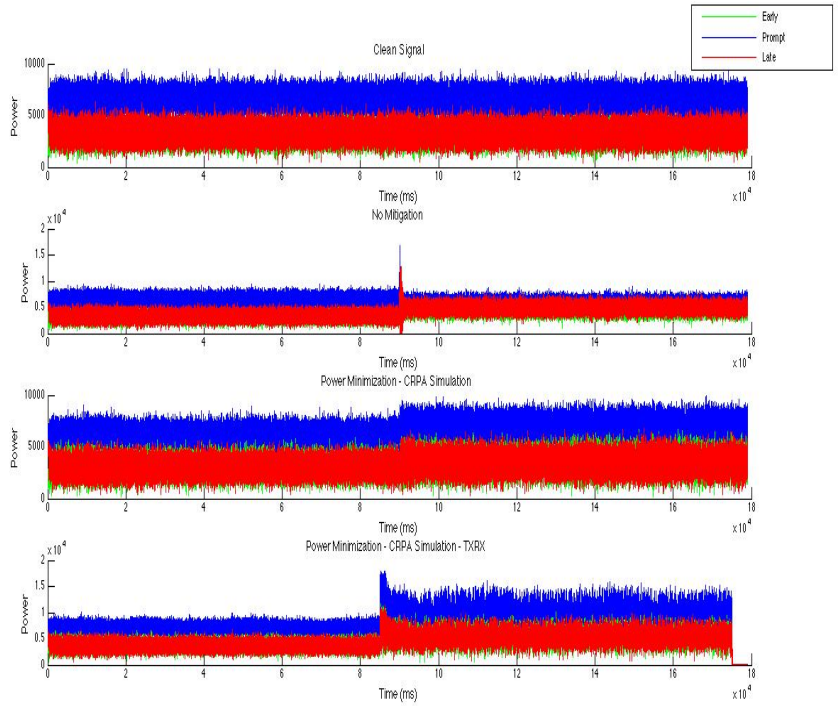

Figure 5.62: In-Phase Tracking Correlators - Power-Minimization - PRN 18

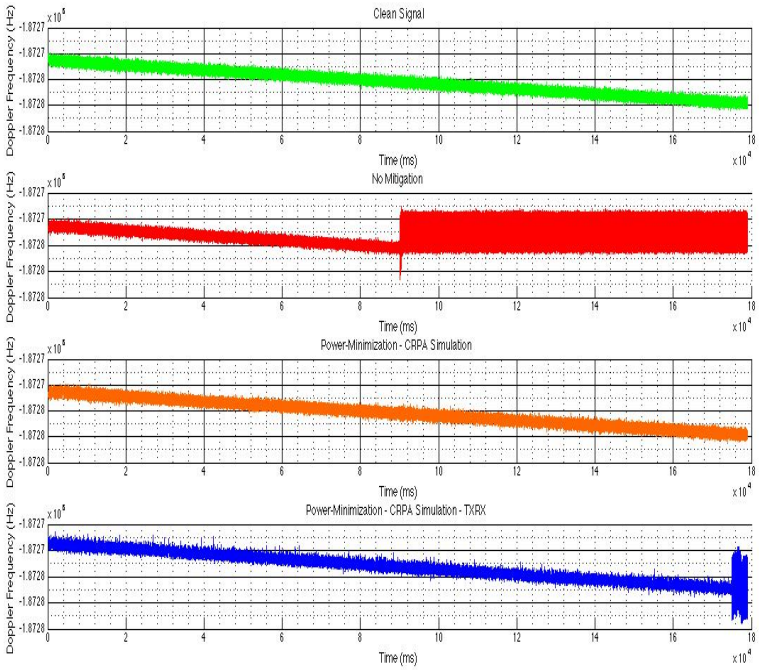

Figure 5.63: Doppler Frequency - Power-Minimization - PRN 18

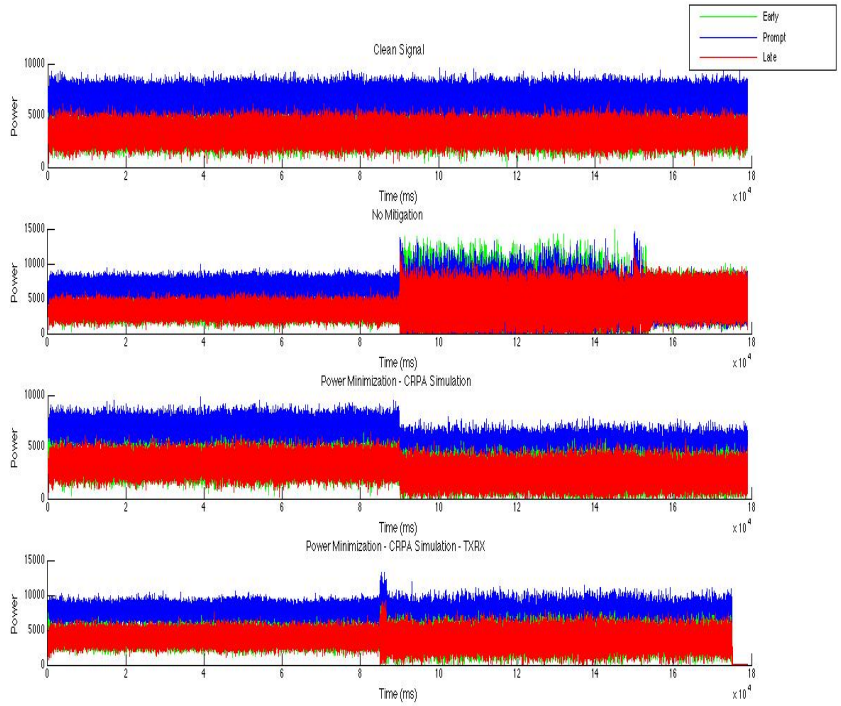

Figure 5.64: In-Phase Tracking Correlators - Power-Minimization - PRN 5

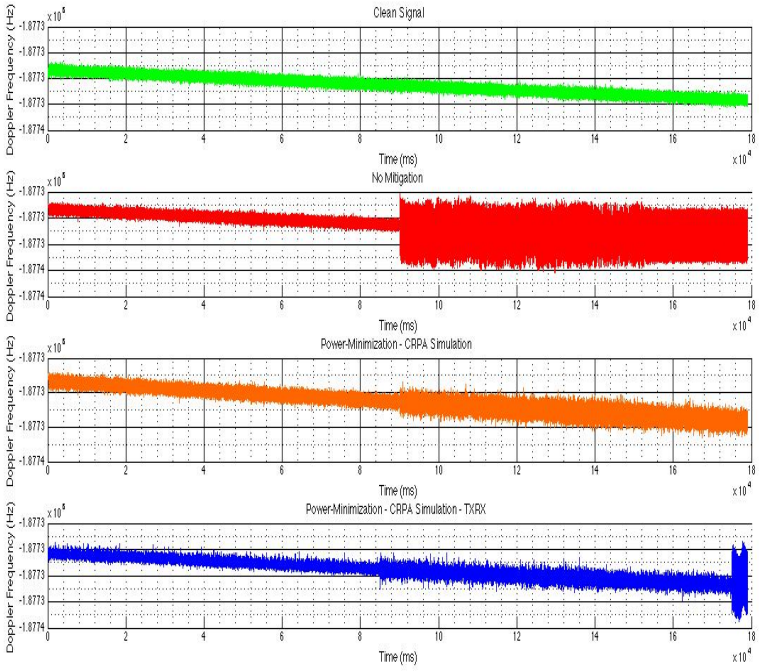

Figure 5.65: Doppler Frequency - Power-Minimization - PRN 5

It can be seen that power-minimization provided better tracking results and Doppler estimates for both the simulated and played-back and recorded signals than that provided with no mitigation. The prompt in-phase correlators (blue) remain above that of the early (grean) and late (red) for each satellite provided. Also, the Doppler estimates have trends similar to that of the clean signal even when the jammer is engaged. Basic position results provided by power-minimization are illustrated by Figures 5.66 and 5.67.

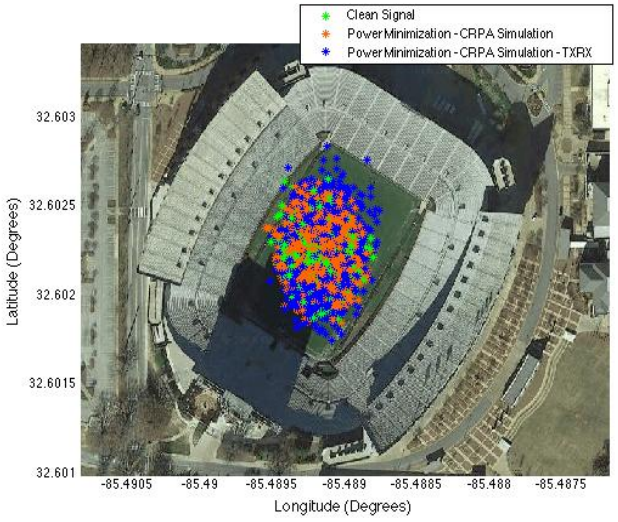

Figure 5.66: WGS-84 Positions - Power Minimization

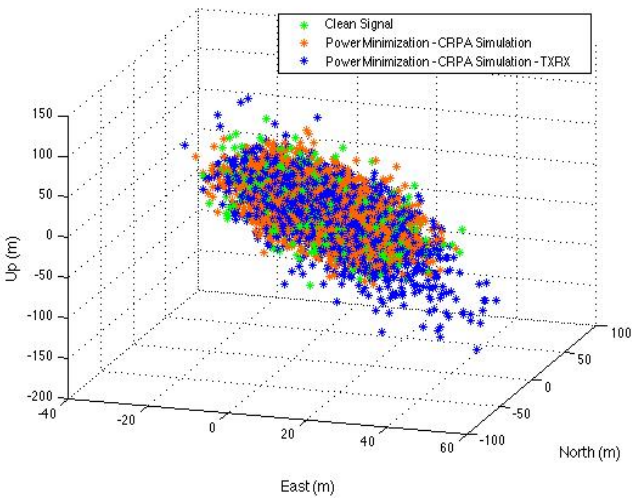

Figure 5.67: ENU Positions - Power Minimization

Power minimization provided positions when the jammer was engaged similar to ones calculated using the clean signal . Additionally, the playback and record did not significantly degrade the final positions calculated by the software receiver. Antenna gain patterns provided by beam-forming for PRN # 5 are illustrated by Figures 5.68 and 5.69. The 2D gain pattern corresponds to the gain pattern at the initial elevation of PRN #5.

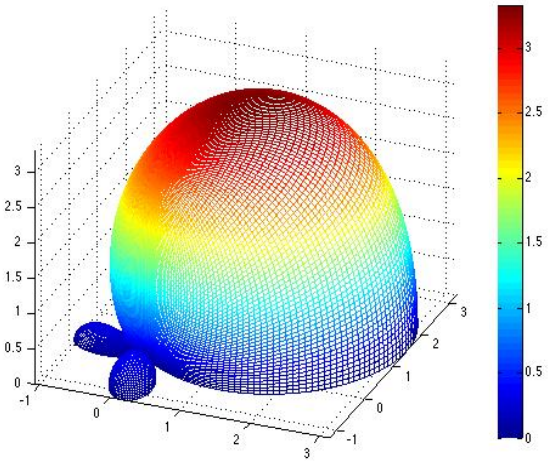

Figure 5.68: 3D Gain Pattern - Beam-Forming - PRN 5

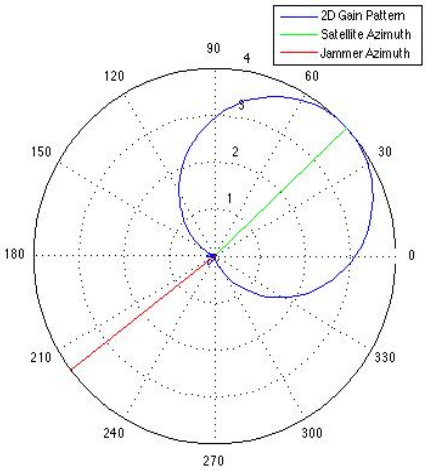

Figure 5.69: 2D Gain Pattern - Beam-Forming - PRN 5

A digital beam or gain was placed in the direction of PRN # 5. In-phase, early, prompt, and late tracking correlators and Doppler frequency estimates provided by beam-forming of selected satellites are provided in Figures 5.70 through 5.75.

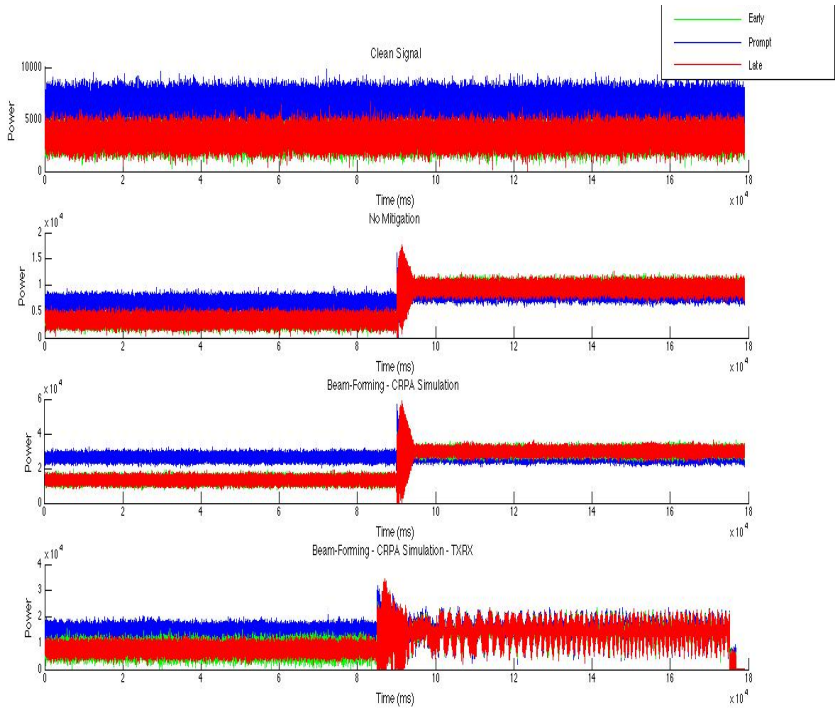

Figure 5.70: In-Phase Tracking Correlators - Beam-Forming - PRN 26

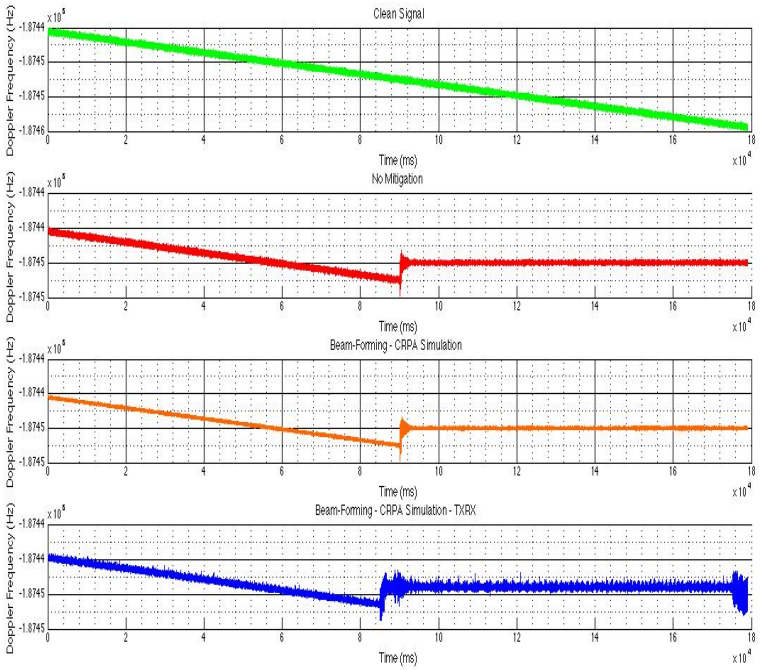

Figure 5.71: Doppler Frequency - Beam-Forming - PRN 26

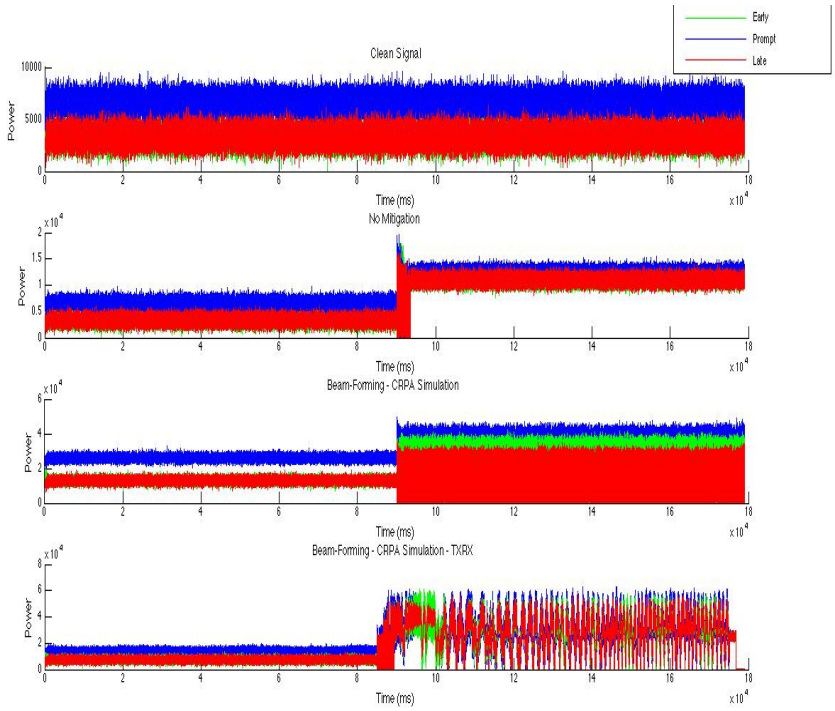

Figure 5.72: In-Phase Tracking Correlators - Beam-Forming - PRN 25

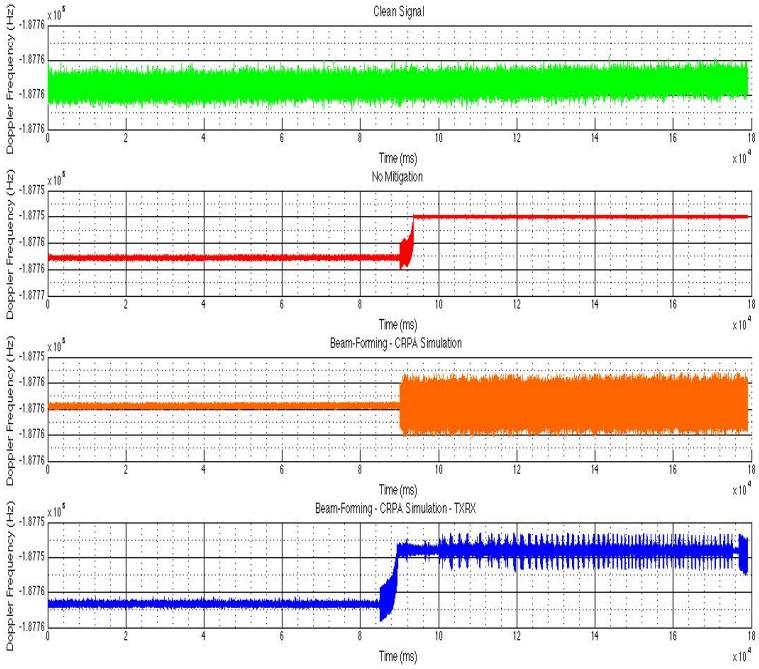

Figure 5.73: Doppler Frequency - Beam-Forming - PRN 25

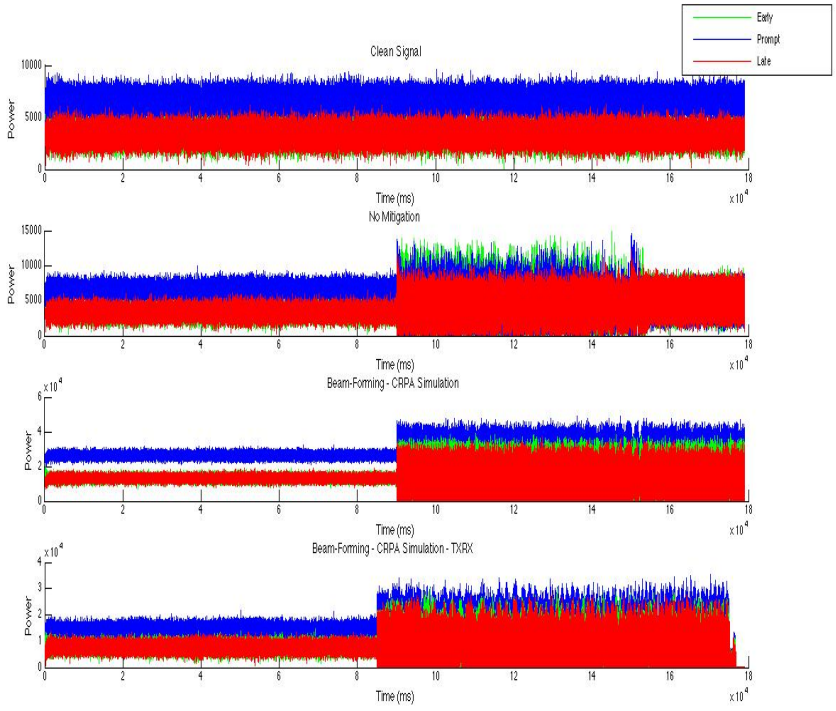

Figure 5.74: In-Phase Tracking Correlators - Beam-Forming - PRN 5

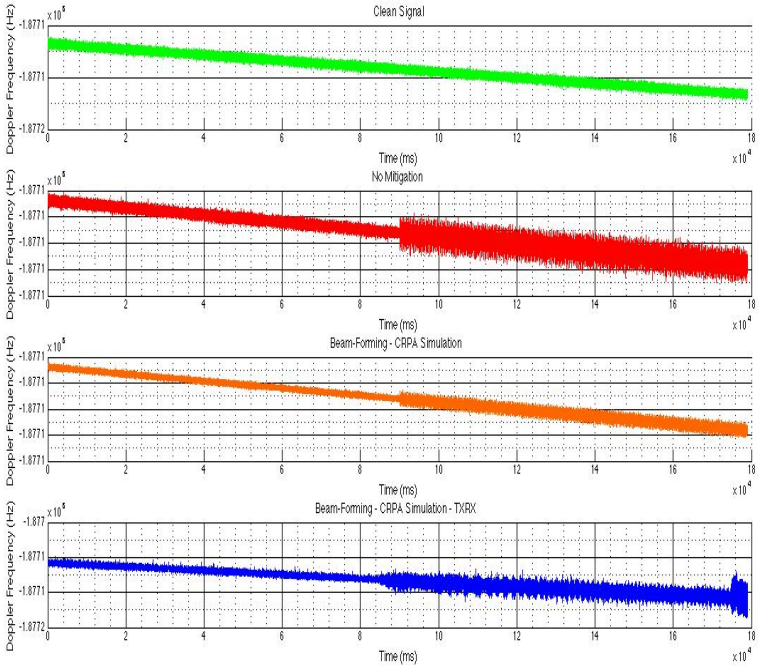

Figure 5.75: Doppler Frequency - Beam-Forming - PRN 5

It can be seen that beam-forming did not provided better tracking results and Doppler estimates for PRN # 26 and that a similar trend is followed between the simulated and played-back and recorded scenarios. For PRN # 25, beam-forming provides better tracking results and Doppler estimates using the simulated scenario and drastically worse results using the played-back and recorded scenario. For PRN # 5, beam-forming was able to provide slightly better tracking and Doppler estimates for both the simulated and played-back signals. Final position results provided by beam-forming are illustrated by Figures 5.76 and 5.77.

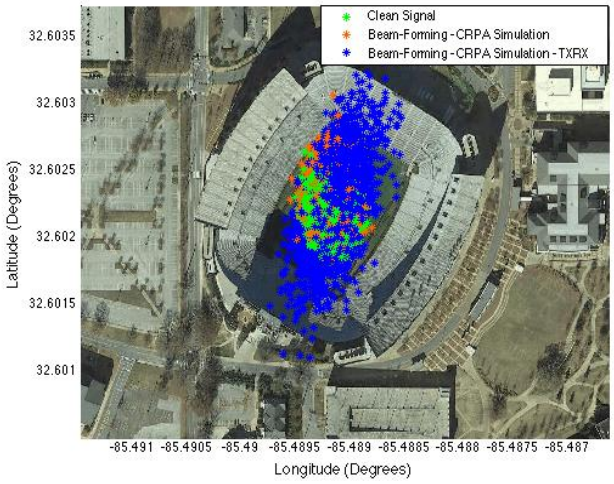

Figure 5.76: WGS-84 Positions - Beam-Forming

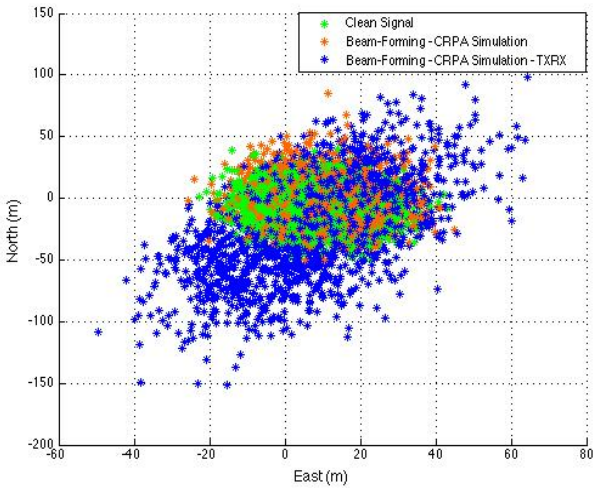

Figure 5.77: ENU Positions - Beam-Forming

Beam-Forming was able to mitigate the jammer and provided a position solution similar to that of the clean signal. Despite the deteriorated tracking and Doppler estimates, the played-back and recorded simulation was still able to provide a position solution but with greater error. The gain pattern formed by null-steering is provided by Figures 5.78 and 5.79.

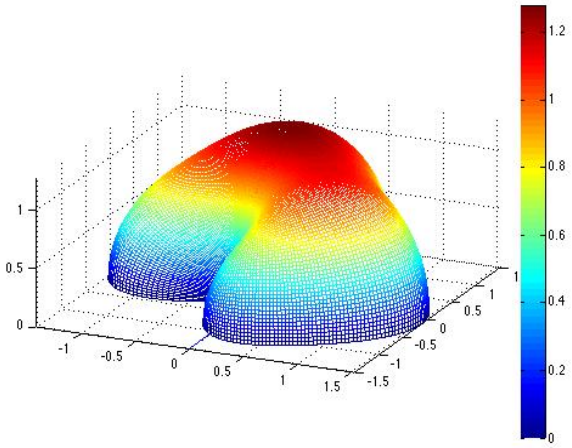

Figure 5.78: 3D Gain Pattern - Null-Steering

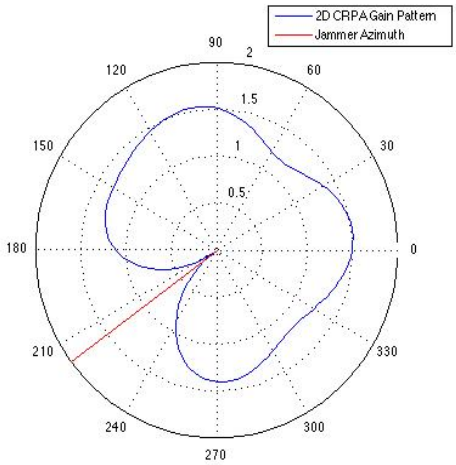

Figure 5.79: 2D Gain Pattern - Null-Steering

A null was successfully placed in the direction of the jammer. In-phase, early, prompt, and late tracking correlators and Doppler frequency estimates provided by null-steering of selected satellites are provided in Figures 5.80 through 5.83.

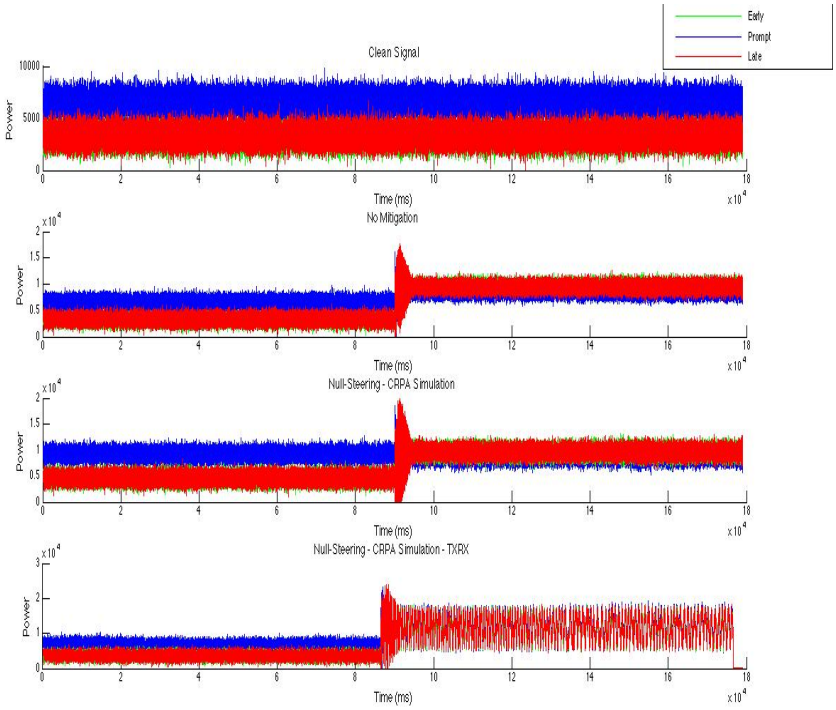

Figure 5.80: In-Phase Tracking Correlators - Null-Steering - PRN 26

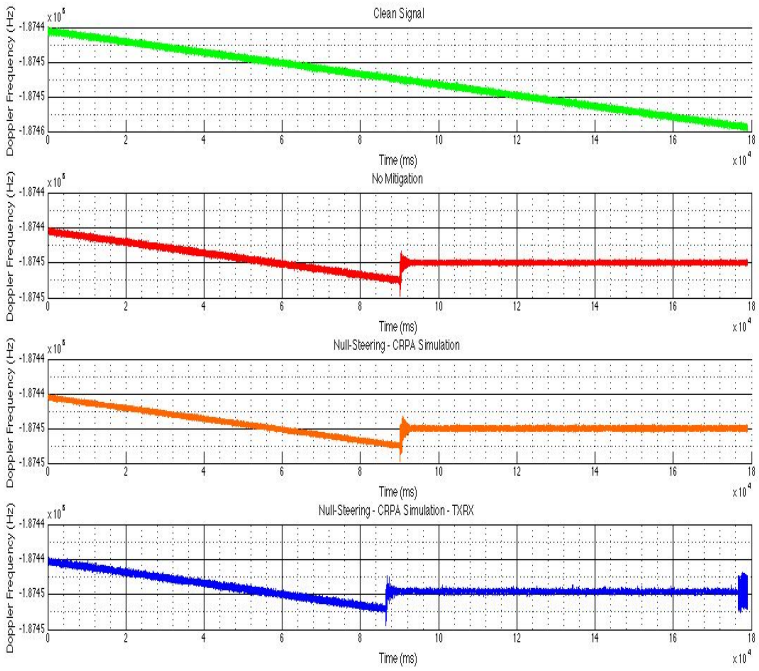

Figure 5.81: Doppler Frequency - Null-Steering - PRN 26

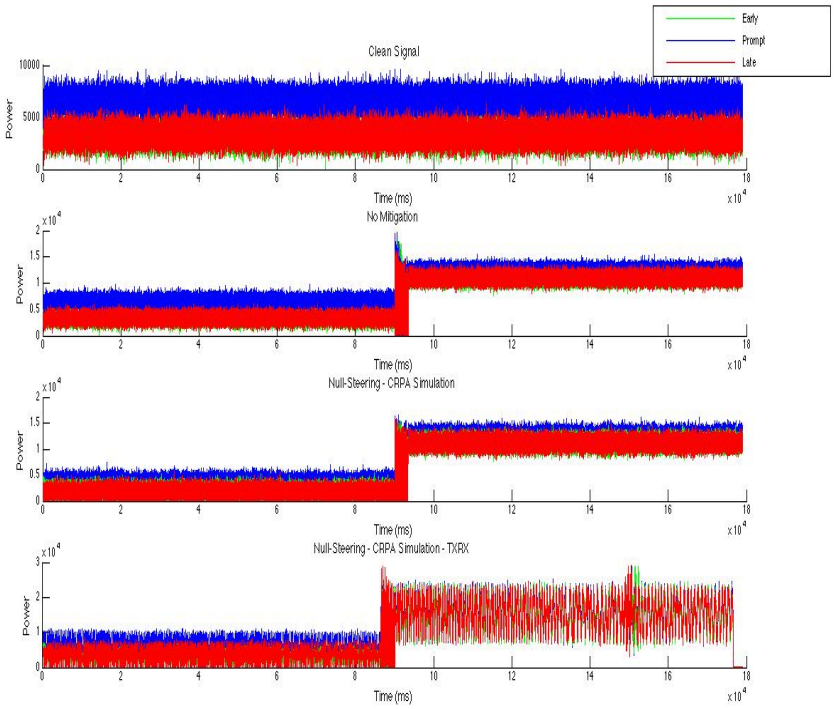

Figure 5.82: In-Phase Tracking Correlators - Null-Steering - PRN 25

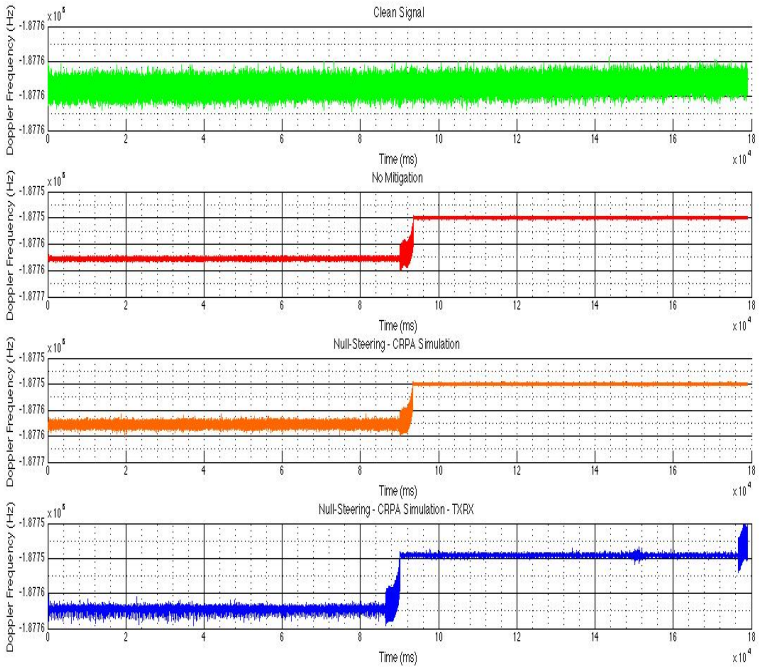

Figure 5.83: Doppler Frequency - Null-Steering - PRN 25

It can be seen that null-steering provided similar results as beam-forming using both the simulated and played-back and recorded signals for PRN # 26. Null-steering using the simulated CRPA signals was not able to track PRN # 25 as well as beam-forming did. Final position results provided by null-steering are provided by Figures 5.84 and 5.85.

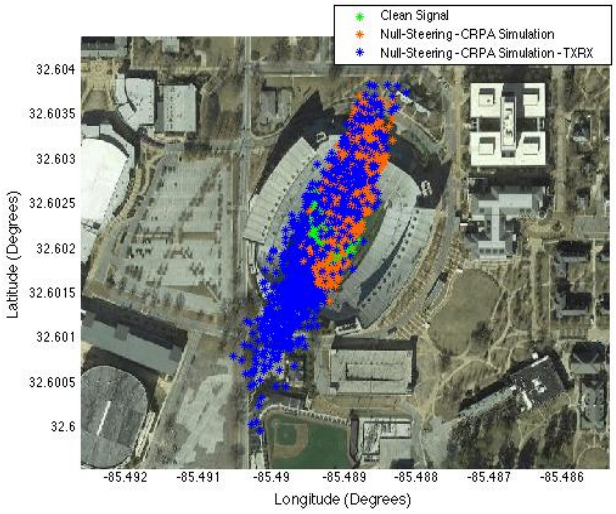

Figure 5.84: WGS-84 Positions - Null-Steering

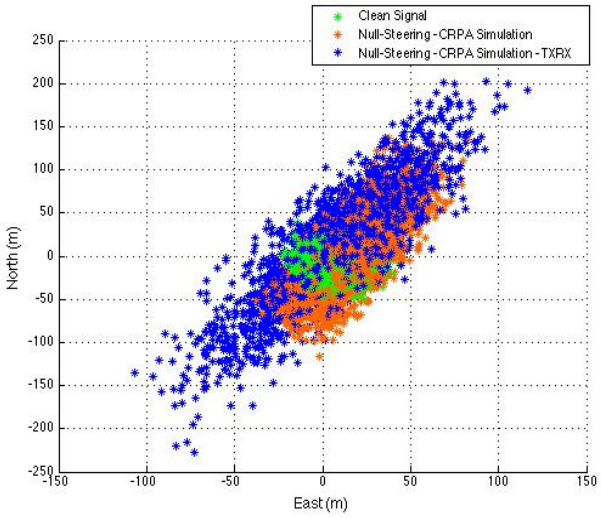

Figure 5.85: ENU Positions - Null-Steering

In this simulation, null-steering under-performed both power-minimization and beam-forming. Also, power minimization provided the best tracking results, Doppler estimates, and position results for this simulation. Additionally, power minimization performed well with the signals that were played-back and recorded with the USRPs. The performance of beam-forming and null steering suffered when using the played-back and recorded signals.

The tests presented in this section illustrate that the software simulator is a valuable tool for developing and testing different types of interference mitigation techniques. The simulator provides full control, allowing a user to simulate custom interference scenarios to test different aspects of mitigation techniques. Additionally, this section has illustrated that a USRP setup similar to the one pictured in Figure 5.35 is capable of providing useful simulation data for the power minimization technique. More analysis is needed to determine the exact cause of the less-accurate results provided by the USRP setup in beam-forming and null-steering.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The software simulator detailed in this thesis is capable of providing single-antenna data to software and hardware receivers via the USRP. It provides complete control of the simulation and the capability to test novel receiver technologies. Also, the software simulator can generate multiple-antenna interference simulations for rapid testing of novel interference mitigation techniques. Additionally, a multiple-USRP setup can provide useful data to test robust mitigation techniques like power minimization. The multi-USRP setup was not able to provide data with the same level of accuracy for more sensitive techniques like beam-forming and null-steering. Moving forward, importance must be placed on mitigating the errors in the USRP simulation that caused the inconsistencies in the beam-forming and null-steering techniques.

6.2 Future Work

This thesis has left opportunities for improvement. First, the software simulator could be modified to include simulated P(Y) code, and the L2 signal. This will allow for a more complete simulation environment and provide more testing capabilities. Additionally, since the GPS community is expected to eventually move to M-code, it will be very beneficial to add this capability to the software simulator. This will allow for testing of future M-code receivers. The software receiver is currently capable of generating jamming scenarios, and it would be beneficial if spoofing simulation capability can be added to the simulator. Lastly,

the MATLAB-based simulator needs to be converted to to a more efficient and faster programming language. Specifically, the scenario simulation needs to be modified to take a desired position and velocity as input and generate the corresponding code and carrier phase based on user specified simulation settings. This functionality could be paired with a field-programmable-gate-array device and used to generate hardware-in-the-loop (HWIL) simulations. HWIL simulation allows for extensive testing of functional hardware in a laboratory setting. HWIL is a very powerful tool but is very expensive. It would be very beneficial if inexpensive software GPS simulation techniques can be used to generate a HWIL simulation. Moving forward, the GAVLAB is interested in creating a software GPS simulation-based HWIL capabilities. A block diagram of a possible HWIL configuration is provided by Figure 6.1.

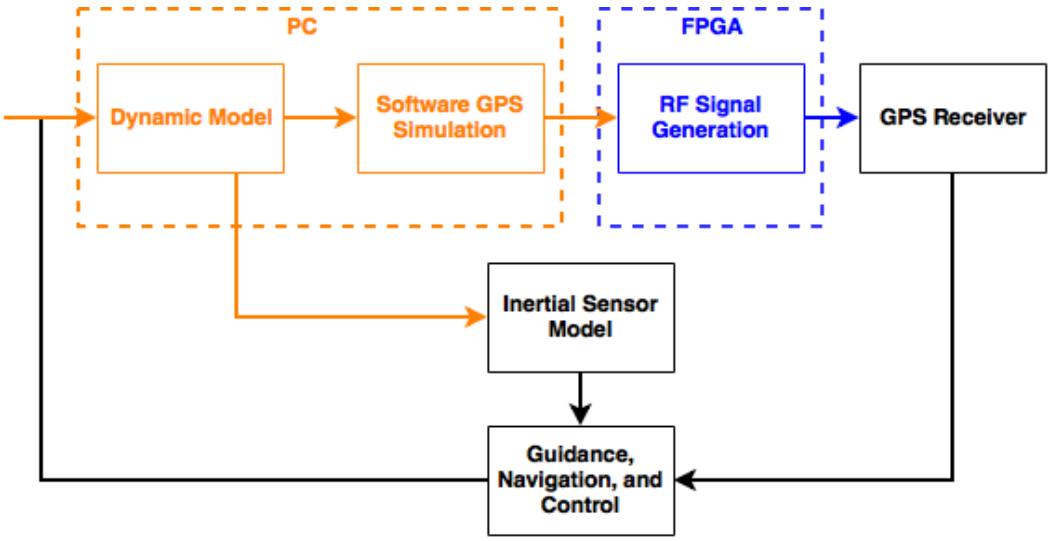

Figure 6.1: HWIL with Software GPS Simulation

In this configuration, a dynamic model is used to calculate the position and velocity to be simulated by the software GPS simulator. The GPS simulator block takes position and velocity as input and outputs the corresponding code and carrier phase. The code and carrier phase are then sent to a FPGA, which generates an IF GPS signal using pre-loaded CA code and data message libraries. The FPGA also upconverts the IF signal to the desired GPS RF. The RF signal is sent to the hardware GPS receiver under test and the outputs are

sent to the guidance, navigation, and control (GNC) algorithms under test. Additionally, inertial measurements can be simulated based on the dynamic model and sent to the GNC algorithms. The output of the GNC algorithms are sent back to the dynamic model and the process is repeated. This configuration would allow for relatively inexpensive HWIL simulation in a laboratory environment.

Bibliography

[1] David G. Alciatore and Michael B. Histand. *Introduction to Mechatronics and Measurement Systems*. McGraw Hill, Boston, MA, 2007.

[2] John W. Betts. *Engineering Satellite-Based Navigation and Timing*. Wiley, Hoboken, New Jersey, 2016.

[3] Kai Borre, Dennis M. Akos, Nicolaj Bertelsen, Peter Rinder, and Soren Holdt Jensen. *A Software-Defined GPS and Galileo Receiver - A Single Frequency Approach*. Birkhauser, Boston, Massachesetts, 2007.

[4] Alison Brown and Neil Gerein. Advanced GPS Hybrid Simulator Architecture. In *Proceedings of the ION 57th Annual Meeting*, page 1, 2001.

[5] Alison Brown, Neil Gerein, and Keith Taylor. Modeling and Simulation of GPS using Software Signal Generation and Digital Reconstruction. In *Proceedings of the ION National Technical Meeting*, page 646, 2000.

[6] Alison Brown, Jarrett Redd, and Michael Dix. Open Source Software Defined Radio Platform for GNSS Recording and Simulation. In *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation*, page 1508, 2013.

[7] Nathaniel Carson. Enhancement and Defense of GPS Navigation Using Signal Processing Techniques. Master's thesis, Auburn University, Auburn, Alabama, December 2015.

[8] NASA CDDIS. note =.

[9] Josh Clanton. Software-Defined, Usrp-based, RF Record, Playback, and Analysis System (SURRPAS). Integrated Solutions for Systems (IS4S), 6 2016.

[10] Phillip Corbell and Mikel M. Miller. Design and Analysis of a Matlab Based Digitized IF GPS Signal Simulator and a Simulink Based Configurable Receiver. In *Preceedings of ION GPS*, page 1906, 2000.

[11] David DeLorenzo. *Navigation Accuracy and Inteference Rejection for GPS Adaptive Antenna Arrays*. PhD thesis, Stanford University, Stanford, CA, 8 2007.

[12] Ruihui Di. A USRP-based Flexible GNSS Signal Recording and Playback System: Performance Evaluation and Study. Master's thesis, Miami University, Oxford, Ohio, 2013.

[13] Lei Dong. If GPS Signal Simulator Development and Verification. Master's thesis, University of Calgary, Calgary, Alberta, Canada, December 2003.

[14] Fabio Dovis. *GNSS Interference Threats and Contermeasures*. Archtech House, Norwood, Massachesetts, 2015.

[15] Per Enge and Pratap Misra. *Global Positioning System - Signals, Measurements, and Performance*. Ganga-Jamuna Press, Lincoln, Massachesetts, 2011.

[16] Jerry Ginsberg. *Engineering Dynamics*. Cambridge University Press, Cambridge, United Kingdom, 2008.

[17] Paul D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Archtech House, 2013.

[18] Per Gustavsson. *Development of a MatLab-based GPS Constellation Simulation for Navigation Algorithm Developments*. PhD thesis, Lulea University of Technology, Lulea, Sweeden, April 2005.

[19] Andrew Hennigar. Analysis of Record and Playback Errors of GPS Signals Caused by the USRP. Master's thesis, Auburn University, Auburn, Alabama, December 2014.

[20] National Instruments. What is I/Q Data?, 3 2016. `http://www.ni.com/tutorial/4805/en/`.

[21] John A. Klubuchar. Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users. *IEEE Transactions on Aerospace and Electronic Systems*, 23(3):325–331, 5 1987.

[22] Thyagaraja Marathe, Saeed Daneshmand, and Gerard Lachapelle. A High Fidelity Multi Antenna Software Simulator for Array Processing in GNSS. In *Proceedings of the 2016 International Technical Meeting*, page 705, 2016.

[23] Scott Martin. *GPS Carrier Phase Tracking in Difficult Environments Using Vector Tracking For Precise Positioning and Vehicle Attitude Estimation*. PhD thesis, Auburn University, Auburn, AL, 5 2017.

[24] Oliver Motenbruck and Eberhard Gill. *Satellite Orbits - Models, Methods, and Applications*. Springer, Berlin, Germany, 2012.

[25] NAVSTAR GPS Joint Program Office, 2004. IS-GPS-2000D.

[26] Heitor David Pinto, Jennifer E. Valdez, Luke M. B. Winternitz, Munther A. Hassouneh, and Samuel R. Price. Development and Test of a Digitally Steered Antenna Array for the Navigator GPS Receiver. In *Proceedings of the 27th Space Simulation Conference*, Annapolis, MD, 11 2012. American Institute of Aeronautics and Astronautics.

[27] Ettus Research. https://www.ettus.com.

[28] Ettus Research. Synchronization and MIMO Capability with USRP Devices, 2 2015. `https://www.ettus.com/content/files/kb/mimo\_and\_sync\_with\_usrp.pdf`.

[29] Ettus Research. USRP Hardware Driver and USRP Manual, 9 2016. http://files.ettus.com/manual/page_usrp2.html.

[30] spectracom. GSG - 5/6 Advanced GNSS Simulators. https://spectracom.com/sites/default/files/document-files/GSG-5_and_GSG-6_Series_Advanced_GNSS_Simulators_revA.pdf.

[31] Joshua Starling and David Bevly. Error Analysis of Carrier Phase Positioning for Controlled Reception Pattern Array. In *Proceedings of the 2017 International Technical Meeting*, Monterey, CA, 1 2017. The Institute of Navigation.

[32] James Bao-Yen Tsui. *Fundamentals of Global Positioning System Receivers - A Software Approach*. Wiley, Hoboken, New Jersey, 2007.

[33] Jong-Hoon Won, Sun-Jun Ko, and Ja-Sung Lee. Design and Test Results of Software Based IF Level GPS Signal Simulator. In *Proceedings of ION GNSS 17th International Technical Meeting of the Satellite Division*, page 305, Long Beach, CA, 9 2004. The Institute of Navigation.

[34] Michael D. Zoltowski and Anton S. Gecan. Advanced Adaptive Null Steering Concepts for GPS. In *Proceedings of the Military Communications Conference*, page 1214.

Appendix A

Appendix

A.1 GPS Ephemeris

A description of the ephemeris parameters brodacast by the GPS satellites that are used to calculate satellite positions and clock corrections is provided in Table A.1.

Table A.1: GPS Ephemeris

Parameter	Name	Subframe
T_{GD_k} (s)	Satellite Group Delay Differential	1
t_{oc_k} (s)	Clock Correction Reference Time	1
a_{f0_k} (s)	Satellite Clock Correction	1
a_{f1_k} (s/s)	Satellite Clock Correction	1
a_{f2_k} (s/s^2)	Satellite Clock Correction	1
M_{0_k} (semicircles)	Mean Anomaly	2
$\triangle n_k$ (semicircles/s)	Mean Motion	2
e_k (unit-less)	Eccentricity	2
$\sqrt{a_k}$ (m$^{\frac{1}{2}}$)	Square Root of the Semi-Major Axis	2
C_{rs_k} (m)	Orbit Radius Correction - Sine Harmonic	2
C_{uc_k} (radians)	Argument of Latitude Correction - Cosine Harmonic	2
C_{us_k} (radians)	Argument of Latitude Correction - Sine Harmonic	2
t_{oe_k} (s)	Ephemeris Reference Time	2
Ω_{0_k} (semicircles)	Longitude of the Ascending Node	3
i_{0_k} (semicircles)	Inclination Angle	3
ω_k (semicircle)	Argument of Perigee	3
$\dot{\Omega}_k$ (semicircles/s)	Right Ascension Rate	3
$idot_k$ (semicircles/s)	Inclination Rate	3
C_{ic_k} (radians)	Inclination Correction - Cosine Harmonic	3
C_{is_k} (radians)	Inclination Correction - Sine Harmonic	3
C_{rc_k} (m)	Orbit Radius Correction - Cosine Harmonic	3

A full description of the complete GPS navigation message can be found in [25].

A.2 Pseudorange, Doppler Frequency, and Carrier-to-Noise Comparison Plots

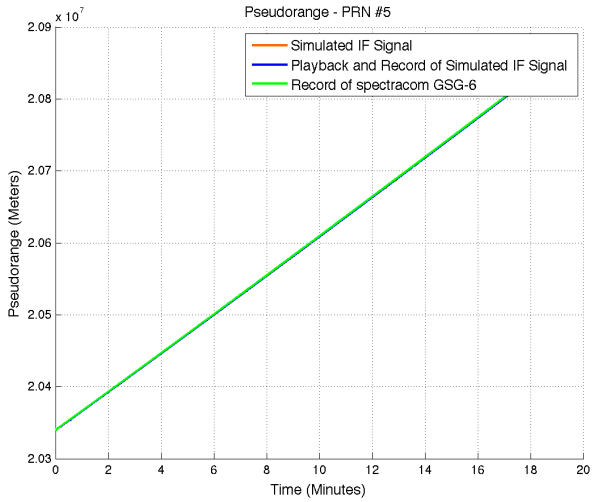

Figure A.1: Pseudorange PRN 5

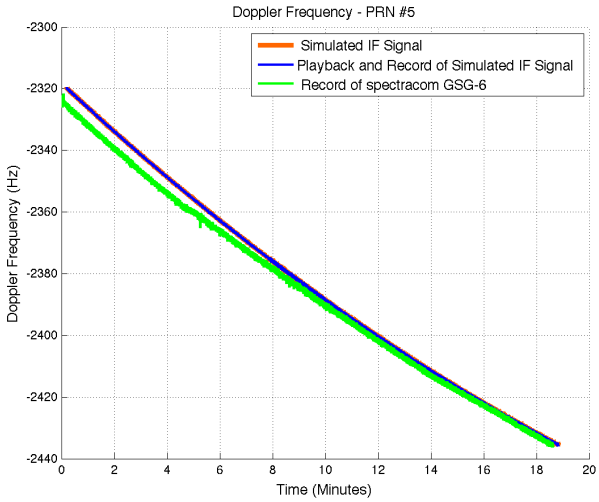

Figure A.2: Doppler Frequency PRN 5

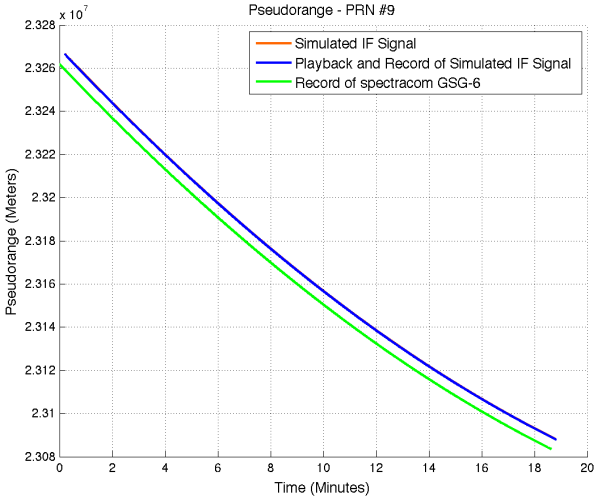

Figure A.3: Pseudorange PRN 9

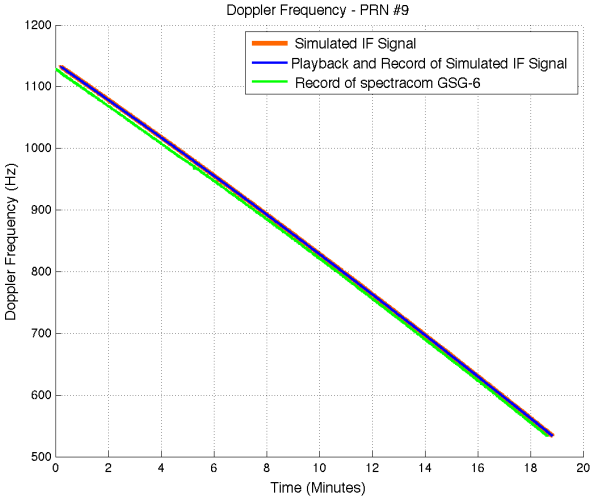

Figure A.4: Doppler Frequency PRN 9

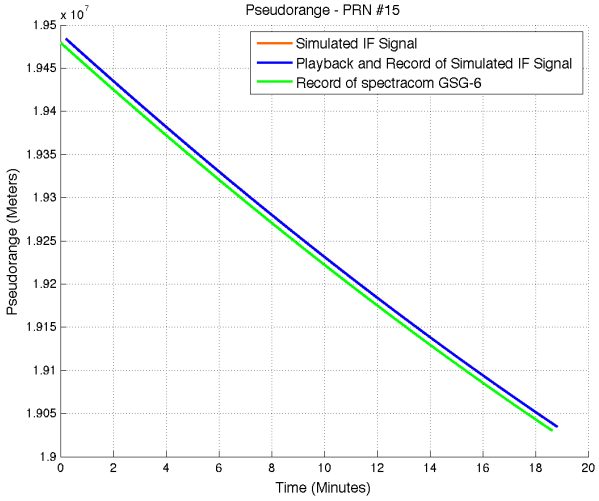

Figure A.5: Pseudorange PRN 15

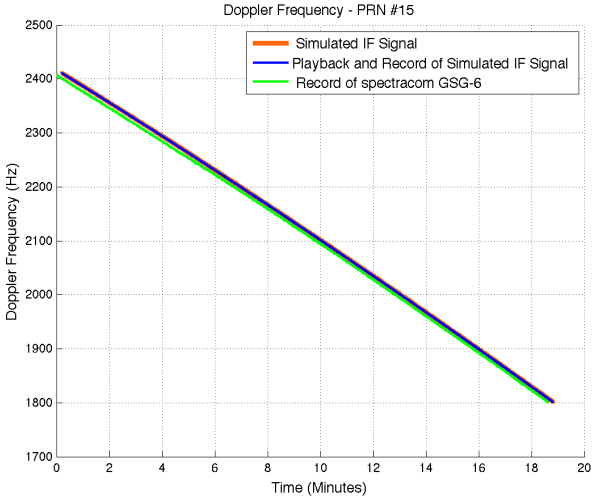

Figure A.6: Doppler Frequency PRN 15

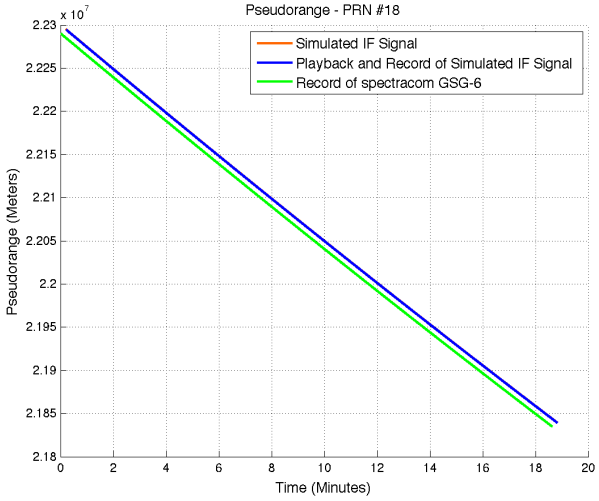

Figure A.7: Pseudorange PRN 18

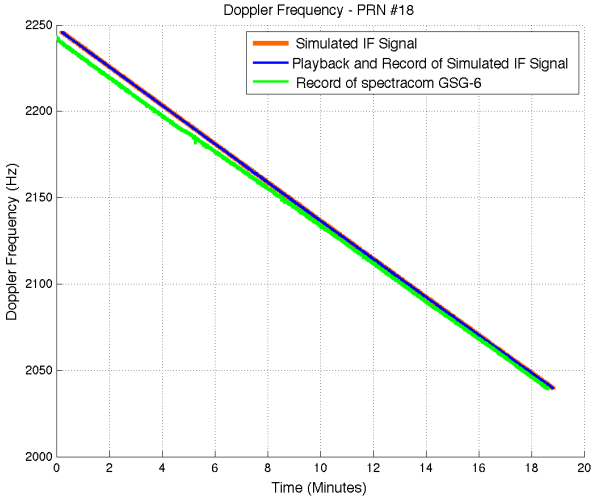

Figure A.8: Doppler Frequency PRN 18

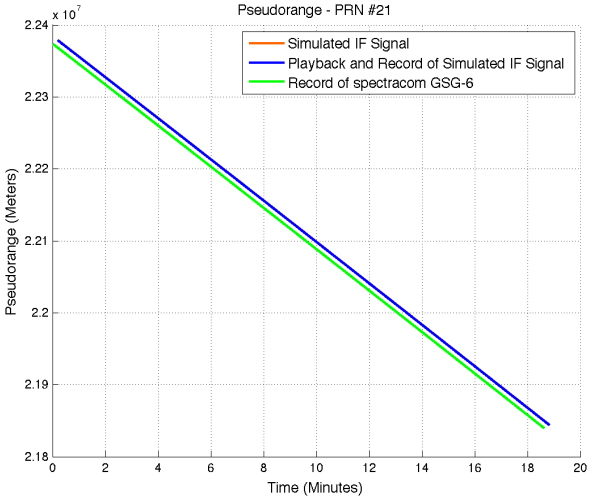

Figure A.9: Pseudorange PRN 21

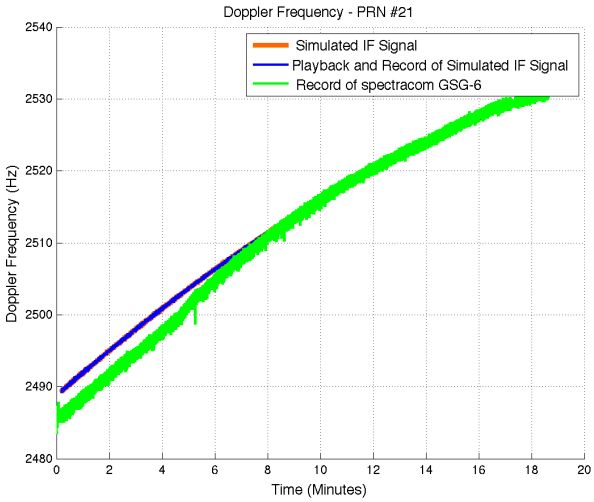

Figure A.10: Doppler Frequency PRN 21

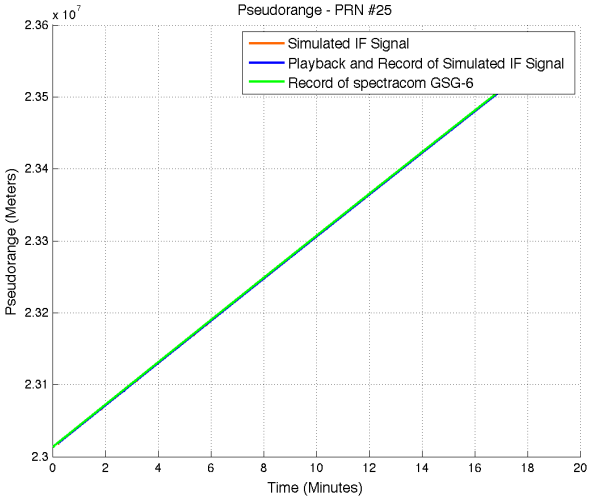

Figure A.11: Pseudorange PRN 25

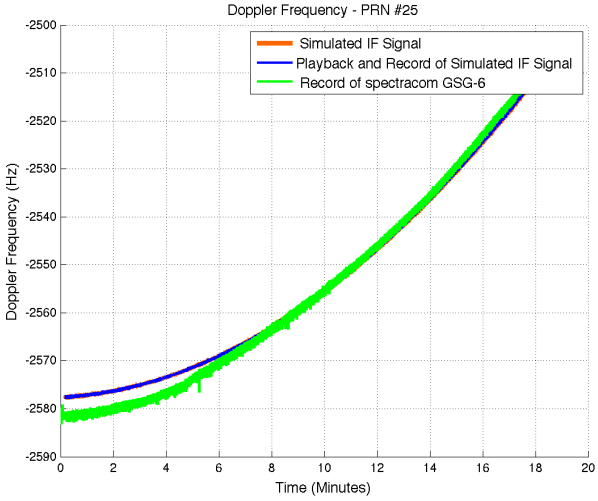

Figure A.12: Doppler Frequency PRN 25

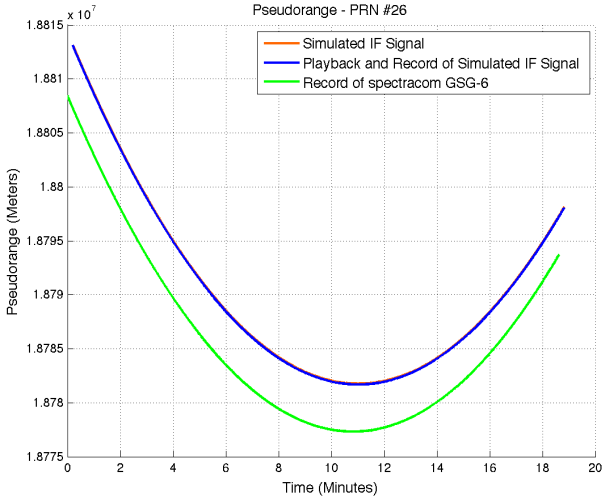

Figure A.13: Pseudorange PRN 26

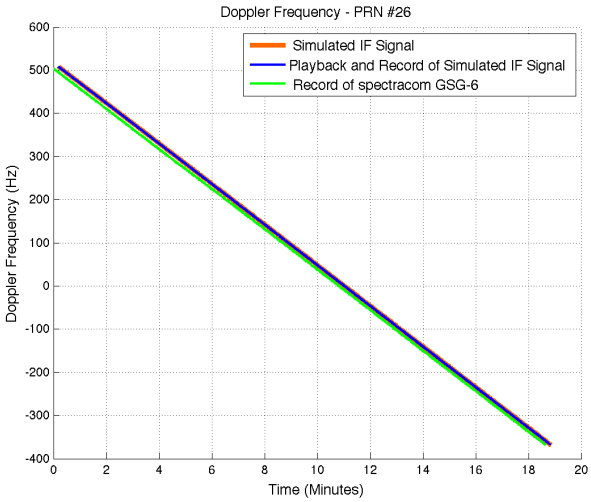

Figure A.14: Doppler Frequency PRN 26

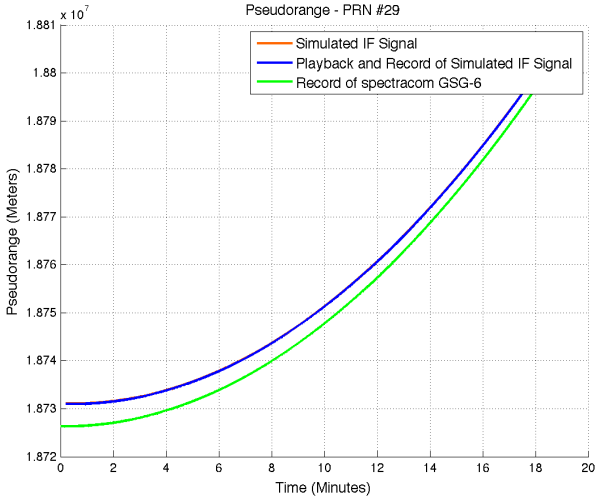

Figure A.15: Pseudorange PRN 29

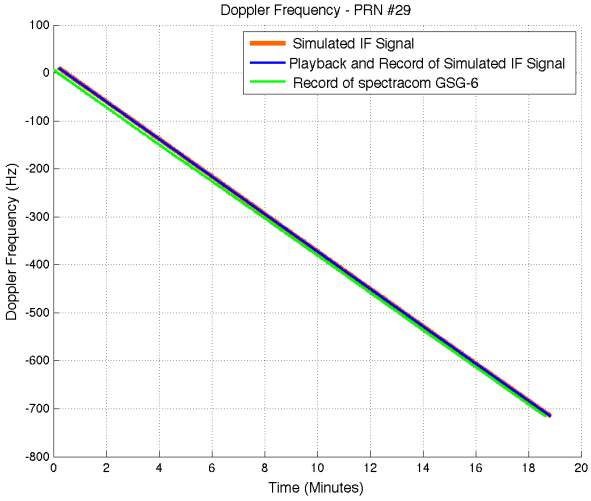

Figure A.16: Doppler Frequency PRN 29

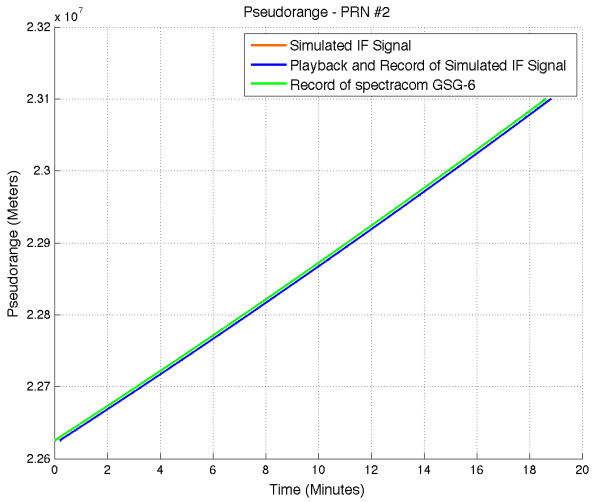

Figure A.17: Raw Pseudorange PRN 2

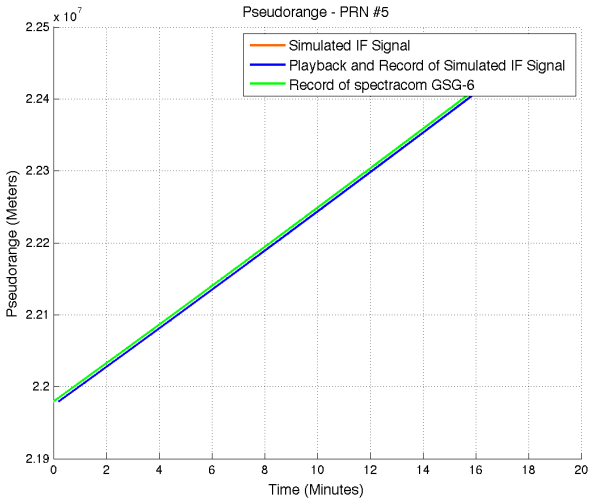

Figure A.18: Raw Pseudorange PRN 5

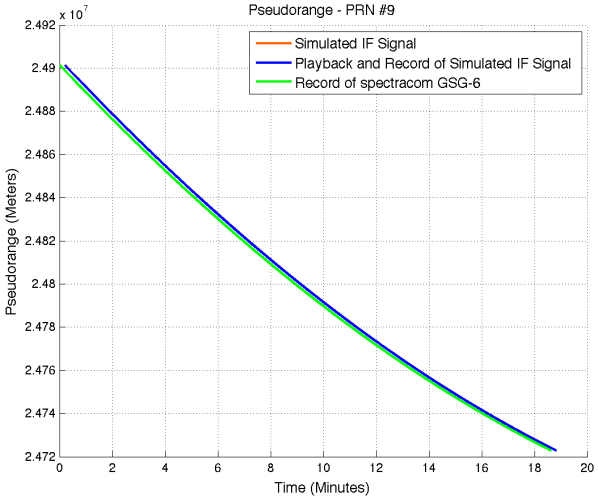

Figure A.19: Raw Pseudorange PRN 9

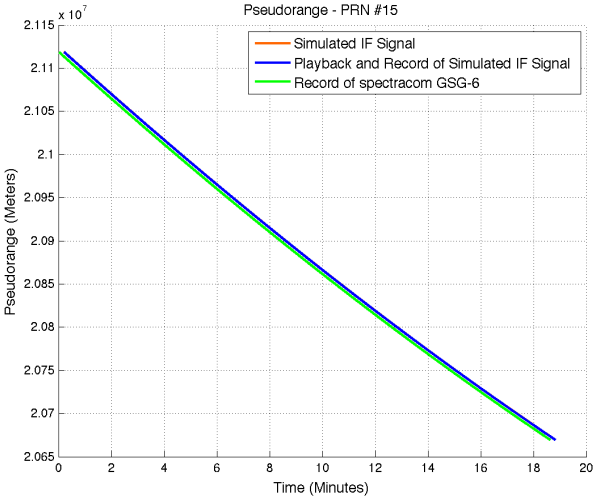

Figure A.20: Raw Pseudorange PRN 15

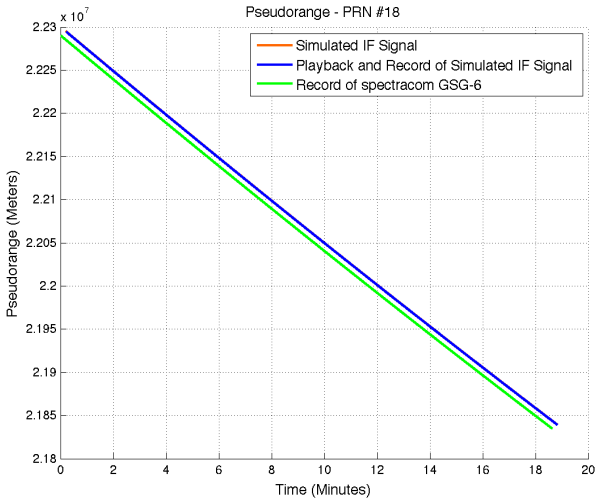

Figure A.21: Raw Pseudorange PRN 18

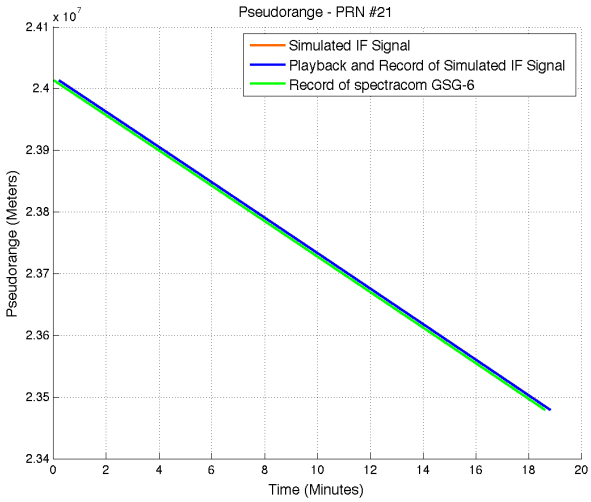

Figure A.22: Raw Pseudorange PRN 21

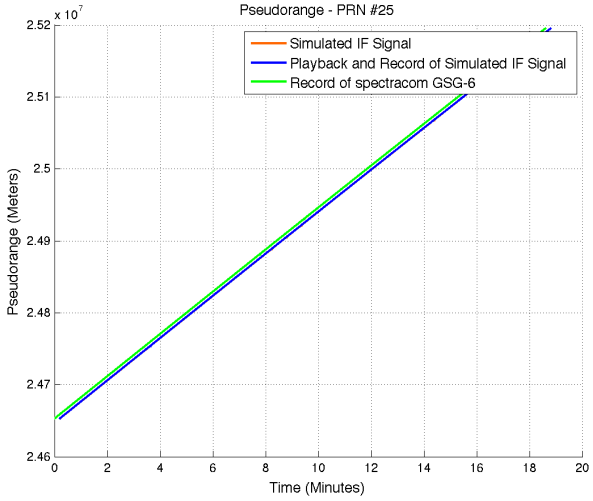

Figure A.23: Raw Pseudorange PRN 25

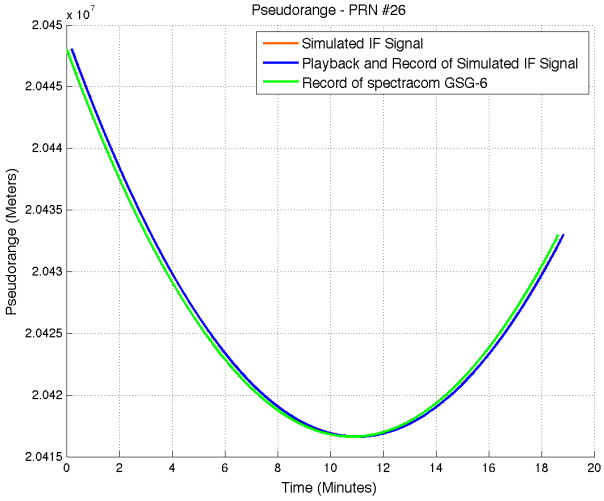

Figure A.24: Raw Pseudorange PRN 26

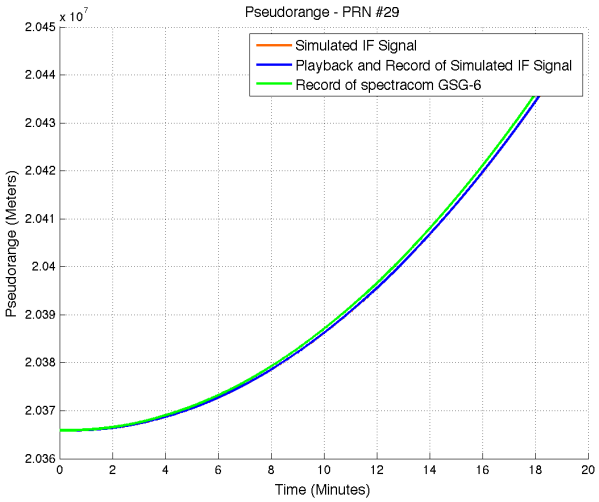

Figure A.25: Raw Pseudorange PRN 29

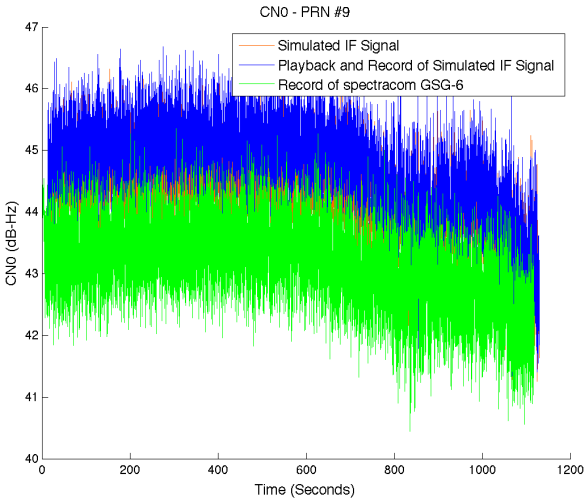

Figure A.26: Carrier-to-Noise Ratio PRN 9

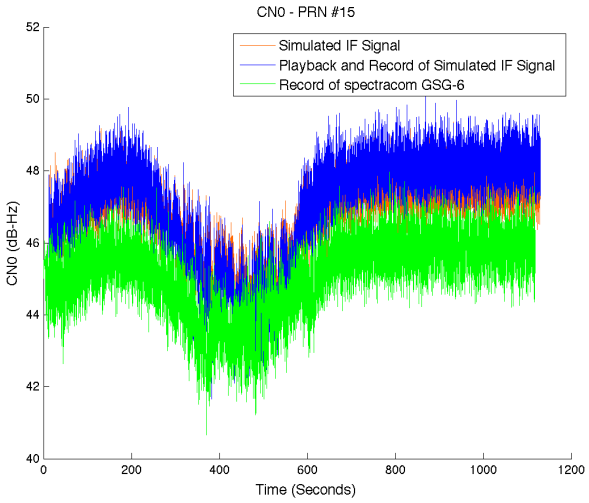

Figure A.27: Carrier-to-Noise Ratio PRN 15

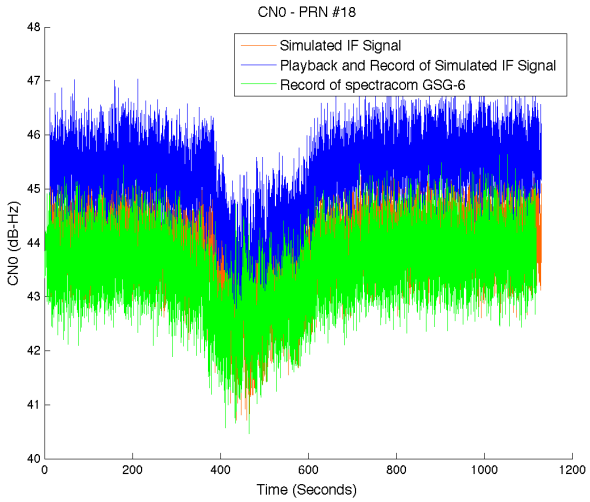

Figure A.28: Carrier-to-Noise Ratio PRN 18

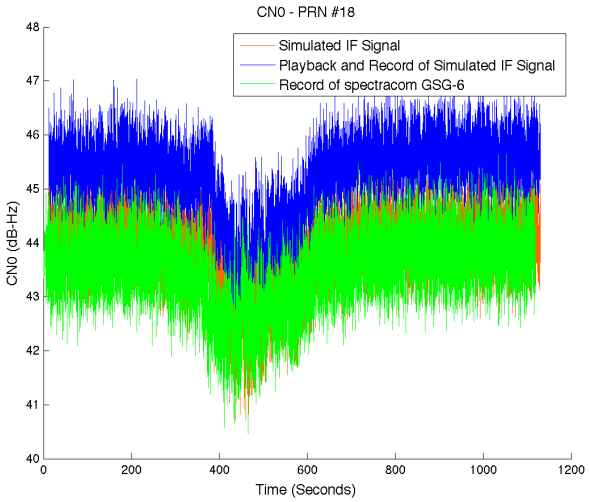

Figure A.29: Carrier-to-Noise Ratio PRN 18

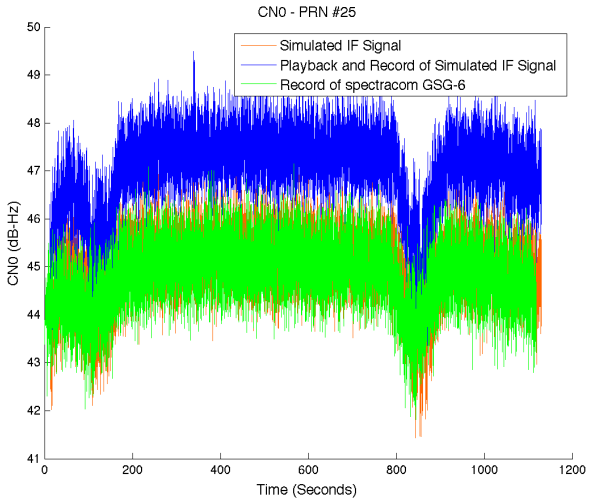

Figure A.30: Carrier-to-Noise Ratio PRN 25

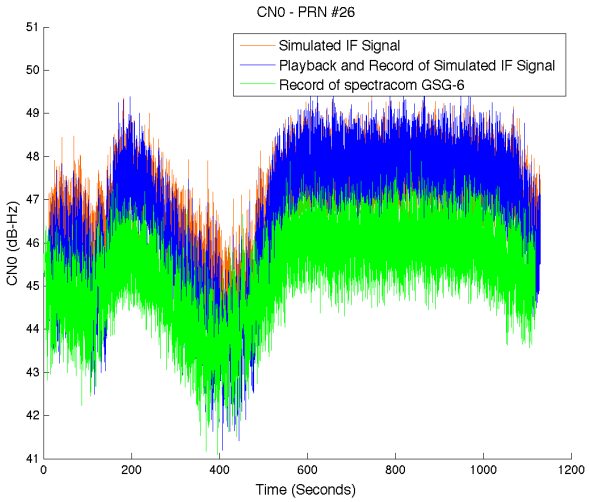

Figure A.31: Carrier-to-Noise Ratio PRN 26

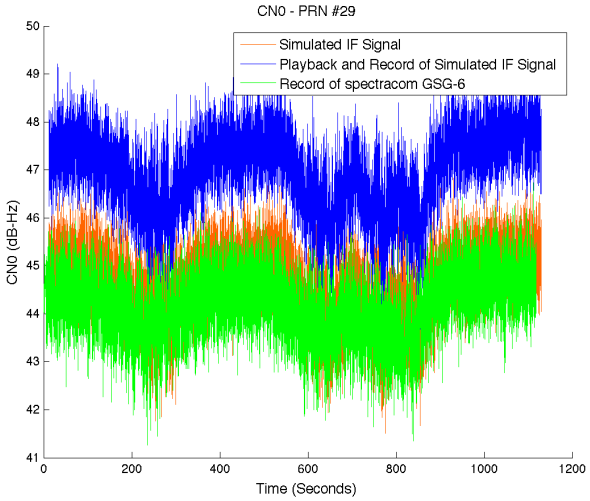

Figure A.32: Carrier-to-Noise Ratio PRN 29

A.3 Software Receiver Sampling Frequency Comparisons

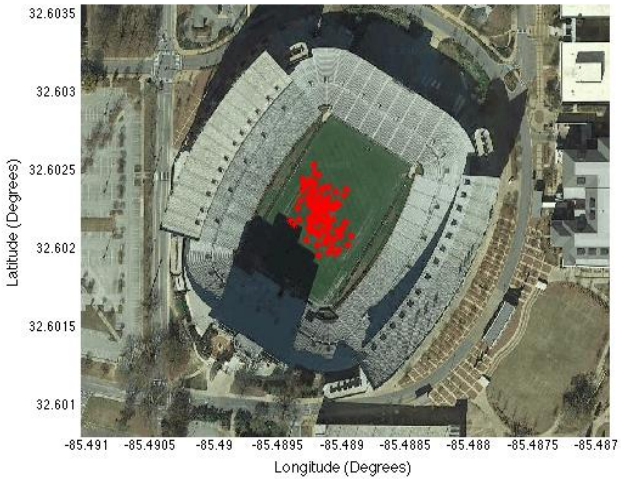

Figure A.33: Software Receiver Position Results - 5 MHz Sampling Frequency

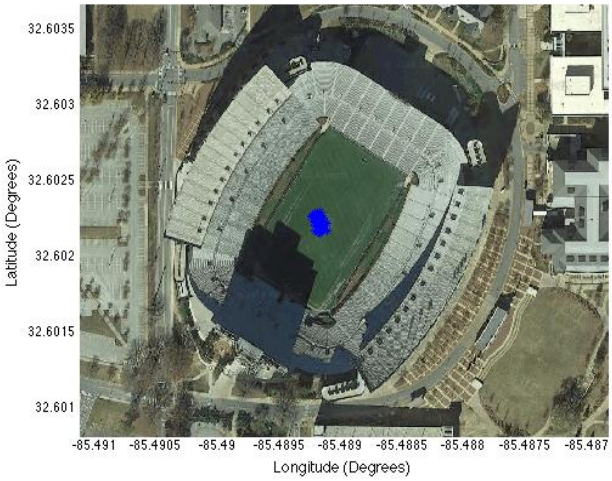

Figure A.34: Software Receiver Position Results - 25 MHz Sampling Frequency