Enhancement and Defense of GPS Navigation Using Signal Processing Techniques

by

Nathaniel R. Carson

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 12, 2015

Keywords: GPS Navigation, Spoofing, Signals Processing

Approved by

David M. Bevly, Chair, Albert J. Smith, Jr. Professor of Mechanical Engineering
Andrew Sinclair, Professor of Aerospace Engineering
Dan Marghitu, Professor of Mechanical Engineering

Abstract

In this thesis methods of spoofing prevention are developed to detect, identify, and mitigate an attack against both networked and standalone GPS receivers. A network based detection algorithm is introduced which combines existing network data and GPS receiver outputs to create a dynamic threshold used as an indication of a spoofing attack. Attack mitigation is accomplished in the development of an interference cancellation algorithm. In the event of an attack, correlators are designated to track the attacking signal and extract critical parameters describing its power, phase, and frequency. These parameters are used to create a replica of the incoming signal which is then subtracted from the buffered raw data. This process removes the interfering signal allowing recovery of the authentic signal and computation of true receiver position. The anti-spoofing routines evaluated in this thesis have an advantage over other methods due to their robustness in a wide variety of situations and their ability to mitigate an attack without any prior knowledge of the spoofer or the spoofed signal characteristics.

Testing of the algorithms developed in this thesis is accomplished using various types of simulated GPS data since live-sky testing in the GPS frequency band is restricted by the Federal Communications Commission. Actual GPS measurements are collected and modified to simulate spoofing in tests of the detection algorithms. Sets of simulated GPS data files are combined in software to simulate spoofing at the signal level. These data sets are used to test the interference cancellation algorithm's effectiveness at removing a spoofed signal in the intermediate frequency (IF) stage. The detection and suppression algorithms are demonstrated to effectively alert the user to an attack and mitigate its effect in IF stage generating a cleaned data set for acquisition and tracking of the authentic GPS signal.

Acknowledgments

My aim and hope is that this work is done for the glory of God and that He is honored in it. It is to Him that I am primarily grateful for His sovereign leading and generous blessings in every aspect of life. He has blessed me with wonderful parents who have always been a steady support and a constant source of advice. I would like to thank them for their wisdom and encouragement over the years. I would also like to thank my precious wife Annika for her amazing sacrifice, encouragement, patience, love, and support through the graduate education process. Finally, I would like to thank my advisor, Dr. David Bevly who has been a source of leadership and guidance through my time here at Auburn.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the growing level of dependence on the Global Positioning System (GPS), it is critical to protect its integrity and ensure its robustness against a variety of threats. Many critical civilian infrastructures rely on both GPS positioning and timing. Financial institutions utilize precise GPS time to time stamp transactions, airlines are becoming increasingly reliant on GPS for navigation, and hundreds of everyday users rely on hand held GPS units to get them to their destination.

Along with increased use has come a variety of threats. Due to the structure of the GPS signal and its relative weakness compared to local background noise, GPS is susceptible to both jamming and spoofing attacks. Jamming operates by blanketing a region in GPS frequency noise to prevent receivers from detecting authentic signals. Spoofing is a more sophisticated method of attack in which receivers are deceived into tracking false signals and calculating an incorrect position solution. Methods of detecting such attacks have been researched on several fronts mostly in the signals processing arena where signal power and other parameters can provide indications of spoofing [1]. This work explores spoofing prevention by looking at ways to detect, identify, and mitigate a spoofing attack on both networked and standalone GPS receivers.

1.2 Prior Work

Signals processing techniques and the defense of GPS against both known and rising threats have been researched extensively in recent years. The work presented here builds

off this research and contributes to the development of more reliable and defensible navigation technologies. In the realm of signals processing detection methods for jamming and spoofing, much work has been done both with GPS as well as other wireless technologies. In 2005, Wenyuan Xu explored methods of mounting jamming attacks in general wireless networks and proposed authentication detection methods to alert the user to such attacks [2]. Multiple jamming detection and localization methods have been proposed, developed, and tested [3–5]. More recently, android technology has enabled the development of detection and localization technology using dead reckoning in pedestrian applications [6]. With the more recent developments of spoofing methods, there has been a rise in research conducted characterizing spoofing attacks and proposing various detection schemes. Gunther recently published an article which provides a very detailed and thorough overview of current spoofing technologies, their dangers, and possible vulnerabilities that would allow detection [1]. In the past few years, Todd Humphrey's has conducted significant research highlighting the vulnerabilities of GPS to more sophisticated methods of spoofing [7]. His work has also aided in the development of more efficient and effective detection schemes. Humphrey's work, in combination with work by Psiaki, has shown multi-antenna detection and phase monitoring to be effective at alerting users to advanced attack forms [8, 9]. This work will be described in section 3.3.2. In 1997 John Cooper and associates proposed a method of interference suppression for modulated jammers [10]. Recently, with the development and implementation of spoofing, this work has been applied to spoofing suppression [11]. This is just a brief overview of a few of the publications relevant to this thesis. GPS jamming and spoofing defense is a rapidly expanding field and there are many other publications that are pertinent to this work. Throughout the thesis, more detailed attention will be given to some of this research.

1.3 Contributions

This work offers several contributions to the field of spoofing detection and mitigation. First, this thesis provides a comprehensive overview of spoofing techniques and reviews several existing detection methods specifically focusing on the signals level of the attack. Second, a new detection method utilizing network information is described and tested in simulation. This method leverages existing network data common in many GPS applications giving it an advantage of low overhead since much of the detection architecture already is in place. Third, this work describes the development and implementation of an interference suppression scheme which is used in combination with a detection routine to effectively cancel the effects of a spoofing attack at the signals level. The suppression scheme was developed around the same time Brounmandan published his parallel work in reference [11]. It utilizes the same concept of subtracting a signal in the intermediate frequency stage of processing but the focus in this work is on advanced spoofing suppression. In this work, the interference suppression scheme is developed and applied specifically for signals which align closely in space and time as would be encountered in a sophisticated attack. This introduces complications to the suppression scheme which is developed in 3.4.

The specific contributions of this work, then, are as follows:

i) Comprehensive overview of attack methods and current detection schemes

ii) Introduction and development of a network detection scheme

iii) Development and application of an attack suppression scheme specifically for sophisticated forms of spoofing attacks

iv) Introduction of the concept of an inline spoofing detection and suppression module for use with commercial receivers

1.4 Thesis Outline

This work will focus on spoofing detection and suppression schemes for GPS using signals processing and analysis techniques. This chapter has described some of the previous work conducted in this field. It has also provided motivation for the research and outlined the specific contributions. Chapter 2 provides detailed and necessary background information important to understanding the attack prevention techniques developed and implemented in the following chapters. Background information is provided on the operation and structure of the Global Positioning System. The weaknesses of GPS are described as they apply to the attack forms addressed here. Chapter 2 will also briefly describe GPS network application which will aid in understanding the detection schemes developed in later sections. In chapter 3, current detection methods will be described and a new detection scheme will be developed. This chapter will also detail the development and implementation of the suppression scheme. Chapter 4 will contain the simulation development, testing methods, and results. The final chapter will describe future directions for the work and possible improvements.

Chapter 2

Background

2.1 Global Positioning System

2.1.1 GPS Architecture and Operation

The GPS architecture is divided into two segments: the Ground Segment and the Space Segment. The Ground segment, also called the Operation Control System (OCS) , consists of the following components. The Master Control Station is responsible for all control of the GPS constellation. In the event of an emergency, the Backup Master Control Station will replace the OCS. Four ground antennas supply an interface for the GPS satellites and the OCS. The final component is the network of monitoring stations which supply measurements and data to the OCS.

The Space Segment maintains a minimum of 24 satellites and currently consists of 32 operational block II/IIA/IIR/IIR-M space vehicles [12]. Each satellite is capable of broadcasting three ranging signals: the C/A (coarse/acquisition code) which is the civilian ranging signal, the precision (P) code, and the Y-code which replaces the the (P) code when anti-spoofing is turned on. The Y-code is encrypted to limit access to only authorized United States Government users while the C/A code is not encrypted and available to all users. The satellites orbit in six (6) approximately circular orbital planes. The orbital planes are evenly rotated about the earth's rotational axis and inclined at an elevation of 55 degrees [13].

GPS navigation is based on a principle known as trilateration. Known positions and distances to the unknown location are used to determine position. The space segment, together with the ground segment, provide world-wide accessible satellites with known locations which

serve as reference points. The messages which the satellites broadcast provide timing information in addition to satellite parameters known as ephemeris data. The user computes his position using the known satellite locations as well as the ranges to the satellites determined from the signal travel time and speed of light.

2.1.2 Signal Structure and Properties

The satellite message is a combination of binary codes and a carrier wave. The C/A code mentioned above, also referred to as a Pseudo Random Noise (PRN) or Satellite Vehicle Number (SVN), is unique to each satellite. It is a 1023 bit long code which repeats every millisecond. This code is used to identify each satellite, determine the doppler shift induced by relative motion between the user and the satellite, and pin-point the phase shift of the signal. The P(Y) code is the precise version of the C/A code. It accomplishes the same purposes but is 6.187110^{12} bits long and repeats every week. The navigation message is the bit sequence which contains the critical information (ephemeris data) for computing satellite positions, transmit time for computing ranges to each satellite, and other health and status information on the satellites. The navigation message is transmitted at 50 bit/s or one bit every 20 ms. For every bit of the navigation message, this means that the C/A code repeats 20 times as shown in Figure 2.1.

The navigation message is broken into frames which take 30 seconds to transmit. Each frame is divided into five sub-frames which each take 6 seconds to transmit. Sub-frames 1-3 are unique to each satellite and contain clock corrections, health indicators, age of data, and satellite ephemeris parameters. Sub-frames 1-3 repeat every 30 seconds. Sub-frames 4-5 contain ionospheric model parameters as well as almanac information and other data. These frames repeat every 12.5 minutes. A complete 12.5 minute cycle is called as Master Frame and consists of 25 frames [15]. The complete transmitted satellite message is created by adding the navigation message to both the C/A code and P(Y) code which are ninety degrees out of phase. The GPS signal generation is shown in Figure 2.2.

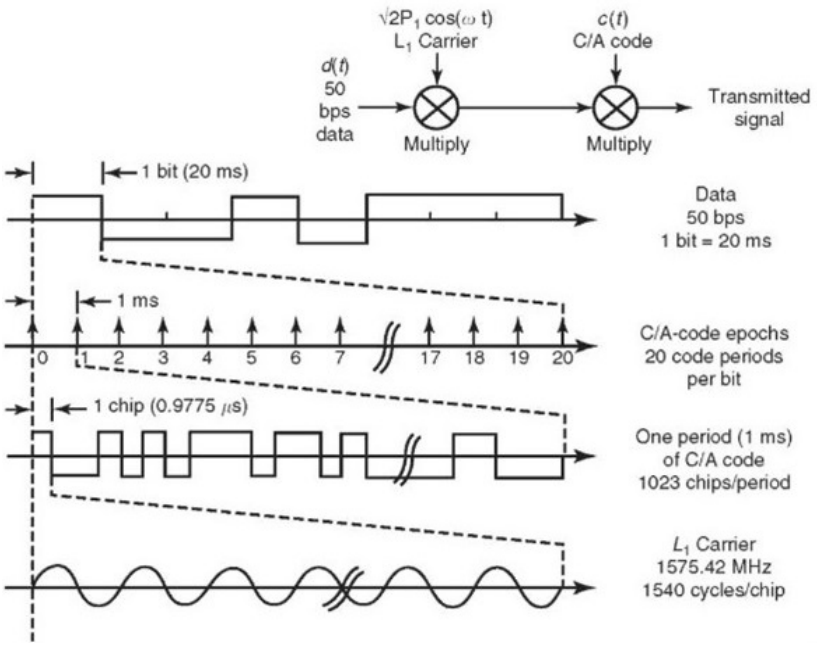


Figure 2.1: Structure of the GPS signal [14].

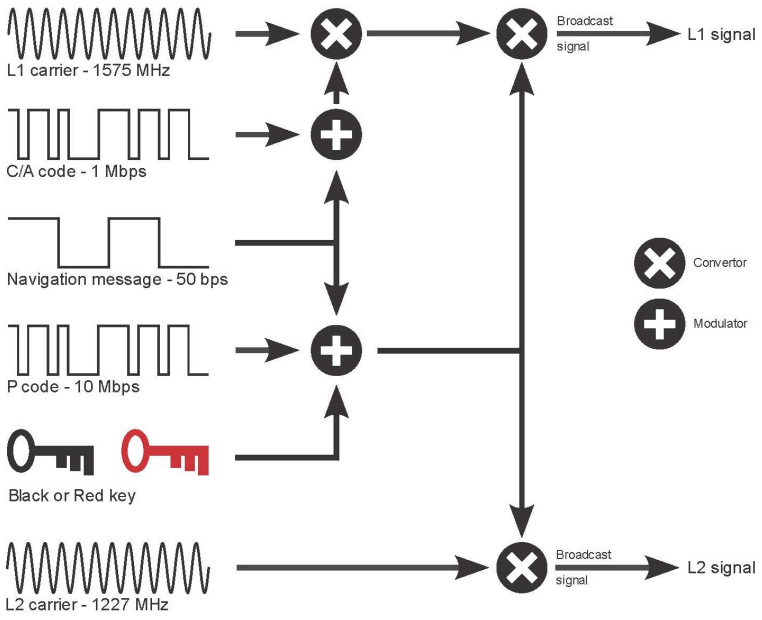


Figure 2.2: Generation of the GPS signal [16].

The C/A and P(Y) are orthogonal due to the 90^o phase shift allowing the receiver to distinguish between the two codes. The resulting code and navigation message is modulated onto the GPS L1 carrier with a frequency of 1574.42 MHz. The C/A code is modulated onto the in-phase carrier while the P(Y) code is modulated onto the quadrature phase carrier maintaining orthogonality. The P(Y) and navigation message are additionally modulated onto a second carrier at 1227.6 MHz referred to as L2. In this work, the focus will be on the L1 1575.42 MHz civilian C/A code modulated GPS signal since it is not encrypted. The combined code and navigation message are modeled by the Equation (2.1) [15].

$$S_{L1}^{(k)}(t) = \sqrt{2P_{C1}}x^{(k)}(t)D^{(k)}\cos(2\pi f_{L1} + \phi_{L1}) + \sqrt{2P_{Y1}}y^{(k)}(t)D^{(k)}\sin(2\pi f_{L1} + \phi_{L1}) \quad (2.1)$$

Where P_{C1} and P_{Y1} are the power of the signals carrying C/A and P(Y) respectively, $x^{(k)}$ and $y^{(k)}$ are the C/A and P(Y) code sequences consisting of a binary sequence where 0 is represented by 1 and 1 represented by -1, $D^{(k)}$ is the navigation data bit, f_{L1} is the L1 carrier frequency, and ϕ_{L1} is the phase offset.

The C/A code has a spreading spectrum effect on the GPS carrier resulting in the power-frequency spectrum shown in Figure 2.3 below. The power is concentrated in the main lobe with residual power lobes spread symmetrically about the center frequency. The GPS minimum signal power specification at the surface of the earth is -158.5 dBW for C/A and -161.5 dBW for P(Y) [15]. The thermal background radiation noise level at the surface of the earth is roughly -204 dBW/Hz meaning the GPS received signal power is significantly below the thermal noise floor. Nominally, the Signal to Noise Ratio (SNR) is -22dB.

Since the GPS signal power is below the thermal noise floor, the correlation properties of the C/A are used to acquire the signal. Since the signal power is spread across a wide bandwidth as shown in Figure 2.3, so is the noise accumulated by the receiver. When the incoming signal is multiplied by an exact, perfectly aligned replica of the C/A code in a process known as despreading, the bandwidth of the signal is reduced to about 100 Hz (this comes

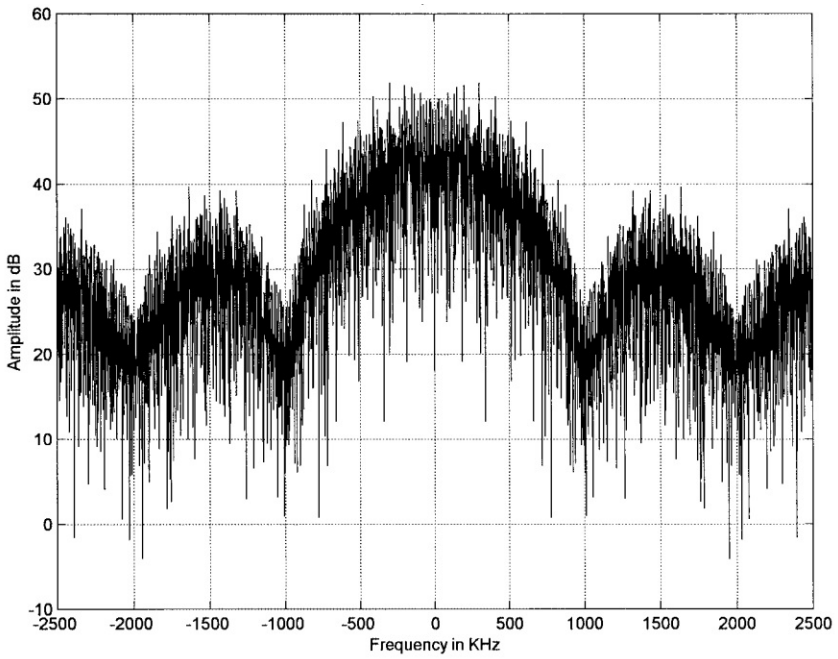


Figure 2.3: Frequency power spectrum of C/A code [17].

from the 50 bit/s navigation message) while the noise power remains spread over the whole spectrum. Subsequent filtering can remove the majority of the noise allowing acquisition of the GPS signal. The C/A autocorrelation property is what allows this despreading. Since the PRN codes only correlate with themselves at one point, the replica will only despread the authentic signal at exactly the correct Doppler and code phase. The auto and cross correlation properties of the C/A PRN sequences are shown in Figure 2.4 below.

2.1.3 Acquisition of GPS Signals

In the previous section, it was mentioned that the strong auto-correlation properties of the GPS C/A code are used to despread the signal and pull it above the thermal noise floor. This process is known as signal acquisition and is the initial step in decoding the satellite signals. Acquisition is performed in a receiver by creating a replica PRN identical to that transmitted by a particular satellite. This PRN is matched against the incoming raw signal to determine if that PRN is present. Since GPS satellites are traveling at thousands of kilometers per second, the transmitted signal experiences extension or compression due

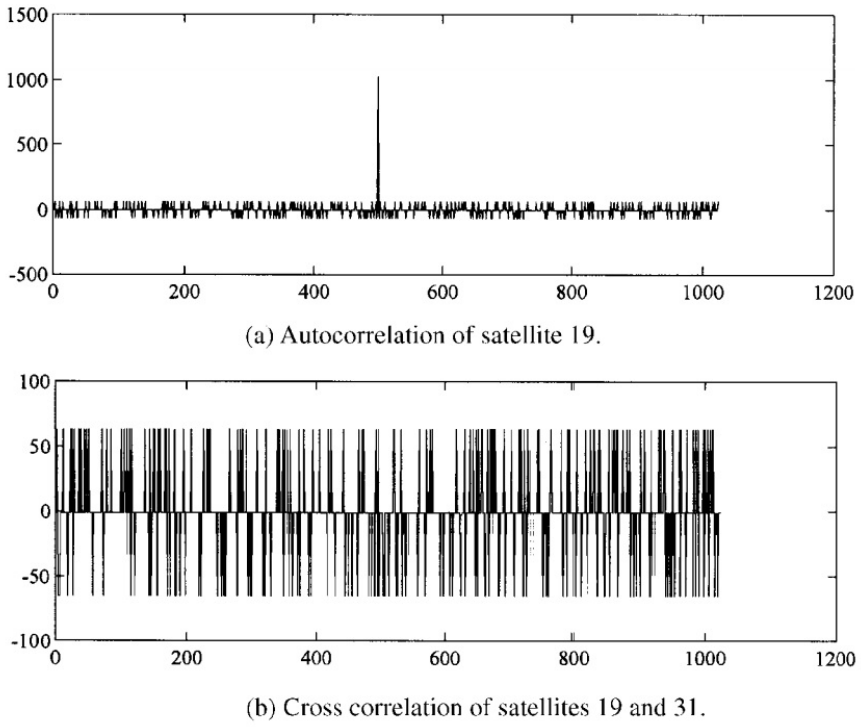


Figure 2.4: (a) Auto-correlation magnitudes of the C/A code. (b) Cross-correlation magnitudes of PRN 19 and PRN 31.

to unknown Doppler shift. Additionally, the start position, or shift, of the code is also unknown. This results in a two dimensional search space or acquisition plane. The process of acquisition is shown in Figure 2.5 below, and is carried out as follows. The raw incoming signal is down converted in the receiver front end to an intermediate frequency (IF) for processing in the receiver body. The raw IF is split and multiplied by a local oscillator. One arm is multiplied by a sine wave and the other is multiplied by a cosine wave producing in-phase and quadrature phase arms for processing. At this point there are two methods of proceeding. The first is a direct, or brute force approach in which a correlation is computed for every possible combination of code phase and Doppler shift. This technique, referred to as serial acquisition, is effective but computationally expensive. A more common approach in software type receiver is shown in Figure 2.5.

This approach is called parallel acquisition. The in-phase and quadrature signals are combined in a Fourier Transform. The replica PRN is pushed through a Fourier Transform

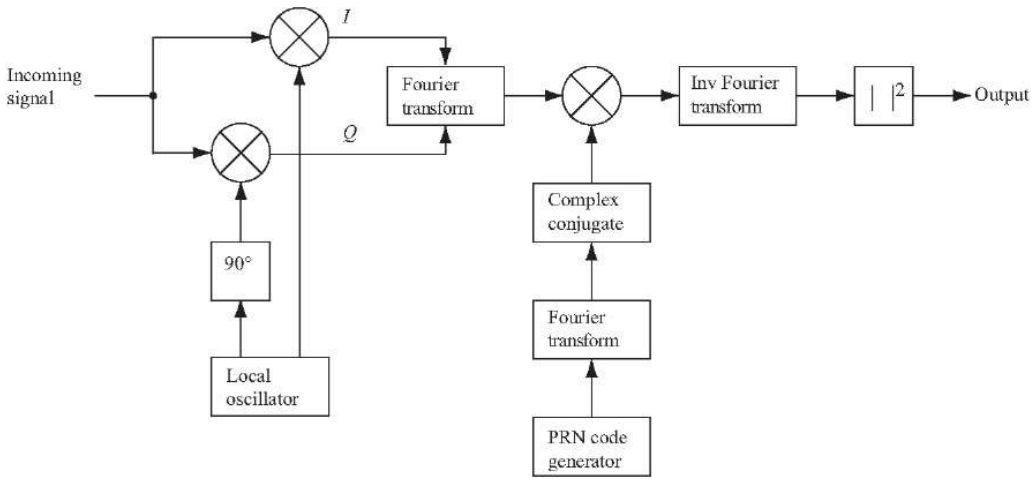


Figure 2.5: Block diagram of acquisition on a single PRN [18].

and the complex conjugate of the result is computed. This result is multiplied by the I and Q combination. The inverse Fourier Transform is taken before squaring the absolute value of the result. This method significantly reduces the computation time as all code shifts are computed in parallel for every Doppler shift [15].

The output of the acquisition computations is a two-dimensional matrix of correlation values which can be evaluated to determine if a signal is present. A typical acquisition matrix is plotted in Figure 2.6 below. This process accomplishes the despreading of the GPS signal and pinpoints the Doppler and code shift of the incoming signal allowing the user to begin tracking and decoding.

The process described above outlines the most basic form of acquisition. The process has several variables which allow the user to specialize acquisition for particular conditions. One such variable is the integration time. In Figure 2.5 above, just before the inverse Fourier Transform, an integration or accumulation block is present. This block integrates multiple segments of signal to increase the power of the correlation. Increased integration time is used extensively to acquire signals that have been degraded. Typically, the integration time ranges from 1-10 ms resulting in an integration over 1-10 C/A codes [19]. The 10ms limit is a result of the navigation message modulation. Every navigation data bit transition "inverts" the C/A code modulated onto the carrier in the span of that data bit. Summing over both

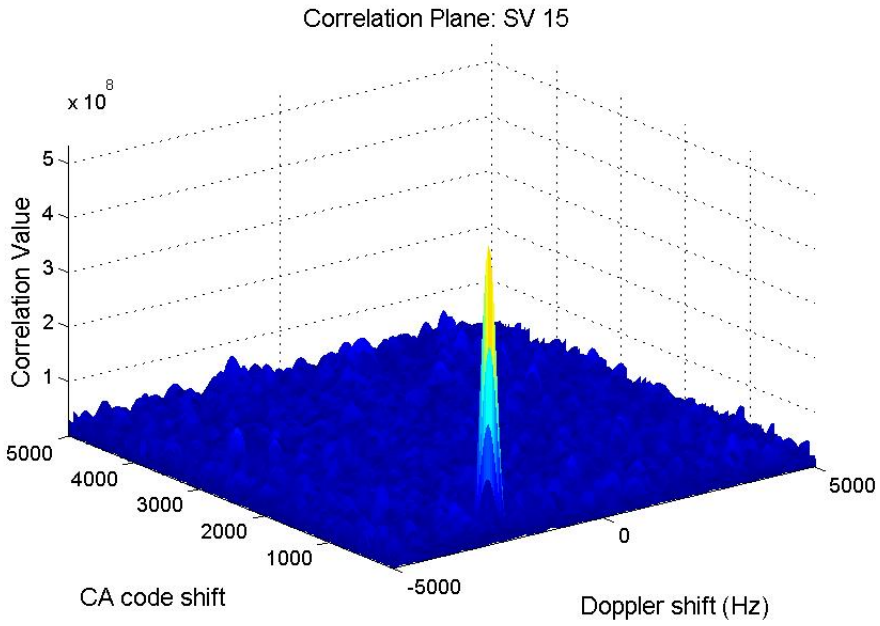


Figure 2.6: Typical acquisition plane showing a correlation peak for satellite 15.

negative and positive versions of the C/A code results in a reduction in the correlation power. Thus, to avoid summing over a data bit transition, a 10ms limit is imposed. With this limit, it is ensured that one of two successive integration periods will contain a transition free segment of 10 C/A codes.

A more complete description of the acquisition process, then, includes the integration segment. The correlation process is performed on two successive segments or integration periods and the result with a higher power is chosen for use in the remainder of the acquisition process. From here, the Doppler shift and code phase are extracted from the acquisition matrix and stored for use in the tracking loops.

2.1.4 Tracking of GPS Signals

The purpose of tracking loops is to increase the precision of the estimates provided in acquisition, track changes in the signal, and demodulate data bits. Tracking loops have many different structures and variations, each suited to its specific purpose. Looking at Equation (2.1) we can see that the satellite signal at an instant is composed of the C/A code

value, the navigation data bit value, and the carrier wave value. By eliminating the C/A code and the carrier, we are left with the data bit value as desired. The most basic form of a demodulation loop is shown in Figure 2.7. The incoming signal is first multiplied by a replica carrier wave generated by a numerically controlled oscillator (NCO). The NCO is controlled by outputs from the carrier loop discriminator and filter which will be described in more detail below. The signal, with the carrier wiped off, is now multiplied by a replica PRN code generated at the particular Doppler and code shift in a process called code wipe off or despreading [17].

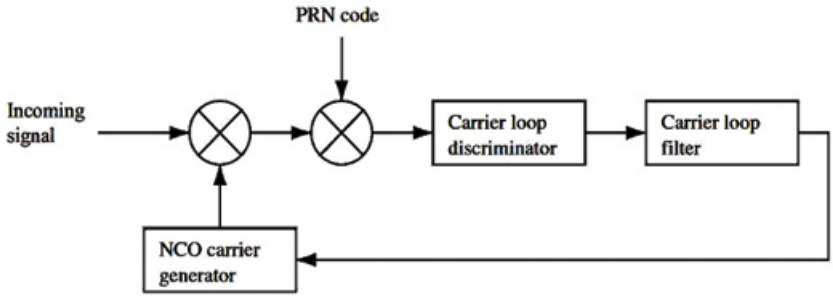


Figure 2.7: Most basic form of a demodulation tracking loop [18].

Tracking is broken into stages since both carrier and code wipeoff must be accomplished to demodulate the data bits. To perform wipeoff, feedback loops are used which track changes in Doppler shift and code phase. For carrier removal, phase lock loops (PLL), frequency lock loops (FLL), or some combination of the two are used. To achieve code tracking, a delay lock loop (DLL) is used to track the changes in the code phase.

For carrier tracking, it is vital for the NCO to be able to generate an accurate replica of the incoming carrier wave. The Costas Loop is perhaps most commonly used to track carrier frequency since it is data bit transition insensitive. The aim of a Costas Loop is to drive all the power into the In-phase (I) arm based on feedback from a loop discriminator and filter. Figure 2.8 below shows the block diagram for a typical GPS application of a Costas Loop. The Costas loop is a form of PLL and can be used with a number of different

discriminators. The most precise is the traditional arctan discriminator shown below. Other common discriminators are also shown in the Table 2.1 from reference [18].

Table 2.1: Phase Lock Loop discriminators

Discriminator	Description
$\phi = tan^{-1}(Q^k/I^k)$	Arctan discriminator with output of phase error
$D = sign(I^k)(Q^k)$	The output is proportional to $sin(\phi)$
$D = I^k Q^k$	The output is proportional to $sin(2 * \phi)$

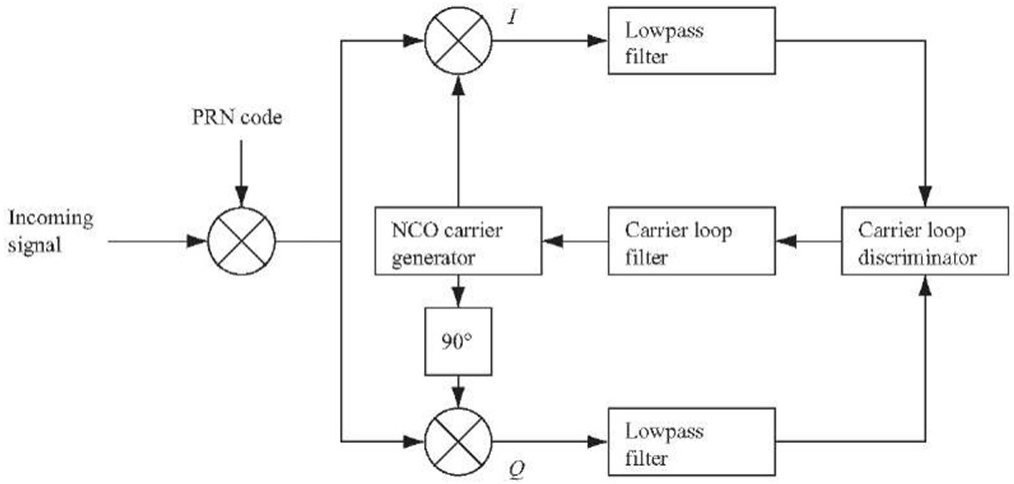


Figure 2.8: Costas tracking loop for carrier tracking [18].

The phase lock loop alone is not enough to demodulate the navigation message. To accomplish this, the C/A code must be tracked and removed as well through the use of a code tracking loop such as a DLL. The goal of a code tracking loop is to monitor the phase of the code and output a perfectly aligned replica of the incoming PRN. A delay lock loop takes a base-band signal as an input since the carrier has been wiped off in the carrier loop. The base-band signal is then multiplied by three C/A code replicas. The replicas are shifted by some nominal chip width value. A precise delay lock loop may only shift the replicas by 1/5 of a chip width while a more common and versatile DLL shifts the replicas by 1/2 of a chip width. One version is shifted forward relative to the estimated code phase, one remains at the estimate, and one is delayed. These multiplications produce three correlation

outputs: early, prompt, and late. The correlation power levels are compared to determine a correction term for the PRN generator. The aim of the DLL is to keep the power concentrated in the prompt branch. The corrections then adjust the phase of the early, late, and prompt correlations to ensure that the prompt code replica aligns as closely as possible with the true code. Graphically, the DLL can be described as shown in Figure 2.9. The DLL seeks to straddle the peak seen in the figure as closely as possible to maintain accurate replicas of the C/A code.

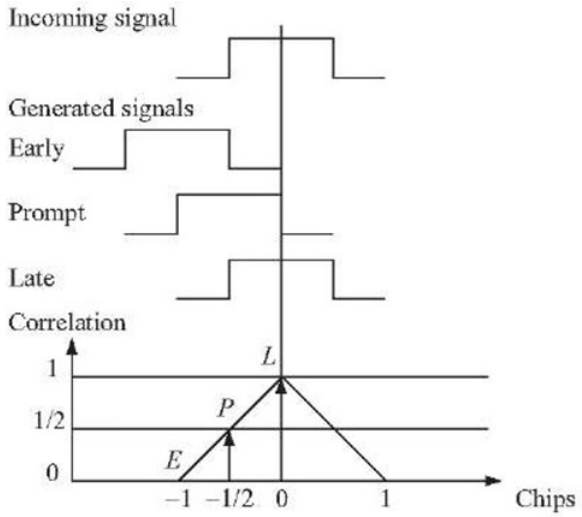


Figure 2.9: Early, Prompt, and Late correlation values [20].

As with the PLL, the DLL employs a discriminator and loop filter to update the PRN generator. The discriminator leverages the power level differences between the early, late, and prompt correlations. Common DLL discriminators are shown in Table 2.2.

Table 2.2: Delay Lock Loop discriminators

Discriminator	Description
$\frac{1}{2}((I_E^2+Q_E^2)-(I_E^2+W_E^2))$	Early minus Late Power
$\frac{1}{2}(I_E-I_L)I_P)+(Q_E-Q_L)Q_P]$	Dot product discriminator
$\frac{\sqrt{(I_E^2+Q_E^2)}-\sqrt{(I_L^2+Q_L^2)}}{\sqrt{(I_E^2+Q_E^2)}+\sqrt{(I_L^2+Q_L^2)}}$	NormalizedEarly minus Late envelope

The complete tracking loop is a careful combination of the PLL and DLL described above. This combined tracking loop maintains frequency and phase lock on the signal using the Costas Loop PLL which is data bit insensitive while the replica PRN generator is updated based on outputs from the DLL. The total block diagram of the tracking loops is shown in Figure 2.10 below.

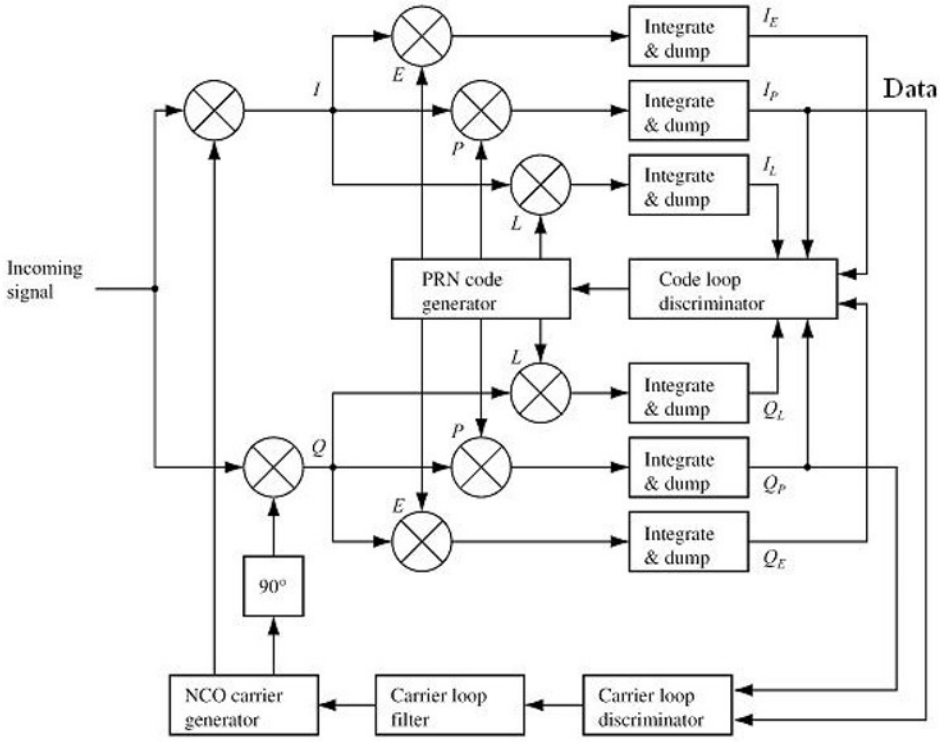


Figure 2.10: Tracking loop block diagram [18].

The output of this tracking loop will be the I_P arm accumulations. Depending on the length of the integrate and dump, these accumulations will represent 1-10ms of data. Most commonly, a 1-2ms coherent integration period is used in the accumulations until bit transitions have been identified. The coherent period is then lengthened to 10ms when the transitions have been detected. Thus, for a single data bit, there will be anywhere from 2 to 20 separate accumulations. During a positive navigation data bit (e.g. a value of +1), these accumulations will be positive. For a negative (-1) data bit, these accumulations will be

negative. From the I_P arm then, the data bits can be extracted and the goal of demodulation has been achieved [15].

2.2 GPS Weaknesses and Vulnerabilities

2.2.1 Degradation of GPS Signals

As mentioned in the prior section, the GPS signal is extremely weak, -22dBm below the thermal noise floor on average. In many ways, this is the Achilles heel of GPS navigation. In clear sky environments, the low signal power does not pose a problem. However, in urban environments, under heavy foliage, indoors, or in the presence of interference, acquisition and tracking of GPS signals can become very difficult or impossible. The geometric diversity of the satellite constellation is one method used to address the less malicious forms of signal degradation. In an urban environment or under heavy tree cover, line of sight (LOS) to one or several satellites may be interrupted causing that satellite to be dropped from the receiver's position computation. The minimum number of satellites needed to determine position is 4, and at any given time, a receiver in open sky will have a LOS to 5-12 satellites allowing for redundancy. Thus when one satellite is lost, the receiver will still be able to compute a position. Figure 2.11 shows the probability distribution of satellites visible satellites at the surface of the earth.

Even with multiple satellites available for redundant measurements, the signals can be weakened by tree cover or interference, requiring signals processing techniques to raise the signal power above the noise floor. Increased integration time can greatly enhance the ability of a receiver to track in a low signal to noise environment as can other network type enhancement techniques. Particularly in urban settings, foliage is not the only degradation that the signals will experience. Receivers moving quickly in and out of buildings or through urban canyons experience a loss of lock, forcing the receiver to spend time reacquiring the signals. Environments in which the GPS signal is weakened either by foliage, buildings, or other obstructions, are here referred to as harsh environments.

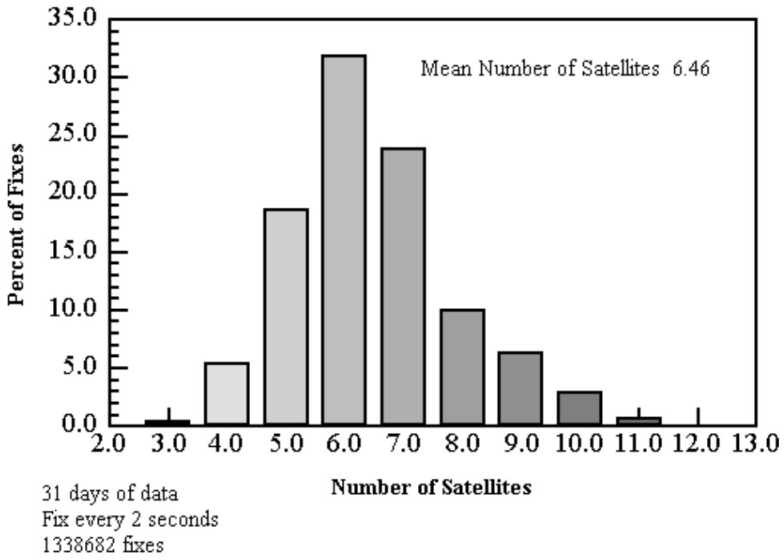


Figure 2.11: Histogram of satellites in view at any time on the Earth's surface [21].

Users who travel in urban canyons or near large buildings will experience an additional form of degradation called multipath. Multipath, shown in Figure 2.12 below, is a result of the receiver locking onto a signal reflected off a nearby structure [22]. In an urban environment, buildings will block the LOS to several satellites while reflecting the signal off their surfaces. This leads to a delay of the signal proportional to the additional path length of the reflected signal. If the receiver locks onto a reflected signal, the additional path length will be added to the pseudorange calculation resulting in a skewed position solution.

Natural obstructions and degradation are not the only sources of interference that may be faced by a GPS user. Some users may be subject to attacks of various forms which block or falsify the signal and cause a loss of true position solution. These sorts of attacks are referred to as jamming and spoofing attacks respectively.

2.2.2 Jamming Attacks

Jamming attacks are a relatively easy or "dumb" form of attack. A jamming attack works by blanketing a region in signals in the GPS spectrum at a higher power than the authentic GPS signal. The target receiver is saturated with noise and loses the ability to

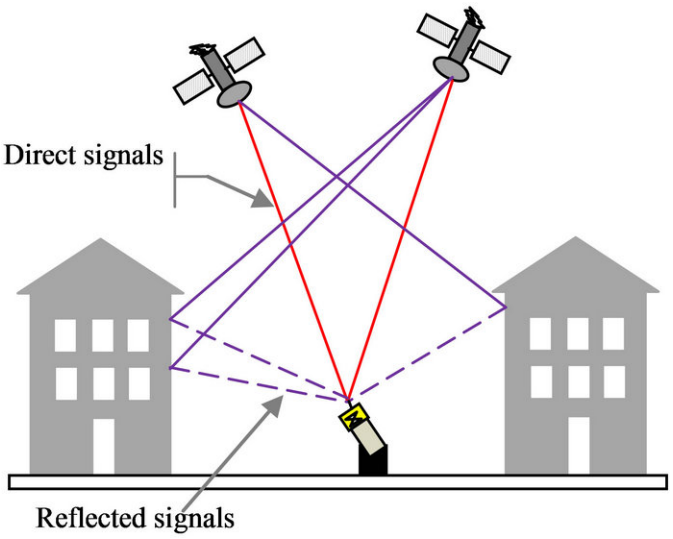


Figure 2.12: Multipath in an urban environment.

acquire or track the true signal cause a loss of signal and jamming of the receiver. Jamming can be both malicious or unintentional. The FCC attempts to limit the use of signals in the GPS band reducing unintentional interference significantly. A jamming attack is designated here as a simple form of attack because this sort of attack leaves plenty of recognizable signals alerting the user to an attack. For example, the noise levels in a receiver will rise significantly and abruptly in a jamming attack and the receiver will struggle to maintain tracking and quickly loose lock on all channels even in an open sky setting. Current technology allows, in many cases, the identification and localization of a jamming type attack with comparative ease. However, jamming attacks still pose a significant threat to GPS users. The simplicity with which the attack can be conducted and the low power of the GPS signals combine in making this threat dangerous on many levels. A simple search for GPS jamming technologies will reveal that cheap, easy to use, jammers are readily available to purchase online. While the purchase, marketing, and sale of any such device is prohibited in the US and may result in tens of thousands of dollars worth in fines, the complete elimination of these devices is impossible.

In August of 2013, the issue of jamming became very pertinent to the air traffic industry when a truck driver inadvertently jammed the Newark NJ airport ground based augmentation system (GBAS) which was in a testing phase at the time. The driver was using a jamming device he had purchased to block the GPS tracker his company installed in his truck. As he drove by the Newark airport on a regular basis, his device blocked GPS signals on multiple occasions before an FCC Enforcement Bureau officer identified the truck as the source of interference [23]. Since a jamming attack results in a complete loss of position solution, there is no danger of an autonomy or partial autonomy algorithm being deceived into using falsified GPS positions and will instead either fail completely or throw out the GPS measurements depending on its design. However, jamming still posse a serious threat to GPS users who rely heavily on constant access to GPS signals for positing and precision timing.

2.2.3 Spoofing Attacks

In recent years, more sophisticated and dangerous forms of attacks have been developed which involve deceiving a receiver into tracking a false set of signals rather than just blocking the authentic signals. This type of attack is called spoofing. Unlike a jamming attack, spoofing attacks do not always leave easily recognizable indicators of an attack. While the noise levels jump significantly in a jamming attack, a spoofing attack may involve little to no rise in noise power as the attacker may tune the spoofing power to simulate that of the authentic signal. Since the receiver is still reporting a position solution, autonomy or partial autonomy algorithms may incorporate spoofed GPS positions without recognizing that there is an issue. Many other indicators of an attack that are present in jamming do not appear in a spoofing type of attack. While it is recognized as a threat, there are very few cases of a successful, malicious, spoofing attack. However, there are several academic demonstrations of the feasibility of such an attack as well as actual reported and verified events of accidental spoofing. One such event occurred at a European international airport

in which a GPS repeater used for testing fooled aircraft GPS units into believing the plane was on the ground setting off the ground proximity alarm during takeoff [24]. In another demonstration, researchers showed the ability to capture a flying drone in autonomous mode and control its motion by hijacking the on board GPS receiver [7].

The purpose of a malicious attack in the real world could vary significantly in scope. An adversary could utilize such an attack to confuse or ambush a more advanced security force relying on GPS for navigation or target acquisition. A domestic spoofer could utilize the technology to commit fraud falsifying time stamps in the financial sector or adjusting timing by increments on the order of milliseconds. Or, as described in the spoofing scenario above, a truck driver could use spoofing to create a false trajectory or alter the timing of events to cover up criminal behavior. No matter its use, spoofing is a dangerous and realizable threat, one that could, especially in a wartime environment, even endanger human life.

During the initial development and introduction of GPS navigation technologies, spoofing was not significantly discussed. The limited knowledge available about the operation of GPS and the small scope of application made it an insignificant threat. With the massive growth in application, and subsequent growth in available information on technical details of GPS operation, has come the threat of spoofing.

Spoofing attacks may take one of several forms. Each type of spoofing varies in complexity and, inversely, in ease of implementation. Spoofing attacks can be broken into two basic categories: those which involve rebroadcasting of real signals and attacks in which the signals are synthetically generated. The first is a more basic type of attack known as meaconing [1]. In this type of attack, realistic signals are broadcast at slightly higher power than the authentic GPS signals and delayed by some amount. Receivers which are not locked onto authentic signals will acquire and begin tracking the higher powered spoofed signals. If unprotected civilian receivers enter the acquisition stage during an attack, they will likely be captured. Even in the tracking stage, it is known that a receiver may lock onto a spoofed signal, or "jump" signals if the code phase and Doppler shift are aligned very closely and

the power difference is significant [1]. In a successful attack, since the attacker likely is rebroadcasting signals received from its antenna location, this is the position that will be computed by any captured receiver. The receiver calculates the psuedoranges as shown in Equation (2.2) from Gunther's article on spoofing technologies [1].

$$\rho_S^k = (\vec{e}^k)\vec{r}' + c(\delta + \tau_S) + ||\vec{r} - \vec{r}'|| + \eta_S^k \tag{2.2}$$

Here ρ_S^k is the spoofed pseudorange to the k^{th} satellite where $\vec{e}^k$ is the unit vector to the k^{th} satellite, $\vec{r}'$ is the spoofer's position, $\vec{r}$ is the victim's position, c is the speed of light, δ^k is the clock offset of the k^{th} satellite, τ_S is the rebroadcasting delay and η_S^k is the noise. These pseudoranges lead to a state solution given by Equation (2.3) below.

$$\zeta = (\vec{r}', c(\delta + \tau_S + ||\vec{r} - \vec{r}'||) \tag{2.3}$$

The receiver calculates a position solution $\vec{r}'$ which is the attacker's antenna location and a clock-offset of $(\delta + \tau_S + ||\vec{r} - \vec{r}'||)/c$. In this type of attack, the spoofer is limited to inducing a delay common to all satellites. A variation on this approach separates each signal by channel and individually delays each signal. This results in a position solution which is neither the user or the spoofer location giving the attack more variability [1].

The second class of attacks involves the broadcasting of manipulated or synthetic signals. This type of attack, referred to here as a synthetic type attack since it involves the artificial manipulation of the underlying GPS signal, is much more complex from the attackers perspective. It requires high precision information on the target's location and trajectory, precise timing, and careful execution to be accomplished successfully. Again, an attack during the cold start of a receiver is the most difficult to avoid. Since a receiver will lock onto the strongest signal present, the attacked receiver will lock directly to the spoofed signal upon acquisition. In a warm start situation where information on predicted satellite positions and approximate user position is available, there is less predictability. Depending on

how close the spoofed position is to the predicted location, the receiver may lock onto either the spoofed or authentic signals. In a tracking situation, the receiver is in its most hardened state posing the most difficult situation for a subtle spoofing attack. It is this case that the synthetically generated spoofing attack proves most effective.

Since the attacker can fully manipulate the signals and the data content, the attack may begin by broadcasting synthetically simulated data replicating the authentic signal as it would be received at the target location. As the signals align, they begin to interfere slightly which will be evaluated in a later section. This phenomenon is known as "beating." The authentic and spoofed signal are practically indistinguishable in this stage. As a result, when the spoofed signal power is raised above that of the authentic signal, the target receiver correlators transition smoothly from the authentic to the spoofed signal. The target receiver is now tracking the spoofed signals but still reporting an authentic position since the spoofer is still broadcasting the user position. However, having full control over the target receiver, the attacker begins the drag-off stage of the attack. The simulated position solution is changed slightly dragging the user away from the authentic position solution. The receiver is now hardened to the authentic signals and will not reacquire them unless the attacker ends the attack or drops the signal. The attacker can control the victim at will unless the attack is detected and suppressed. Since the transition from authentic to spoofed tracking was a seamless transition, there will be very few indications in the receiver outputs.

An additional variation on this approach involves manipulating just the timing aspect of the signal rather than the actual position solution. This is much simpler in many ways but still requires very precise knowledge of the user location and trajectory. A situation like this may arise if an attacker desires to manipulate a precise time tracker such as those used by financial institutions. By broadcasting the precise antenna location, the attacked receiver will transition to the higher powered spoofed signal. The attacker can then manipulate the timing bias by minute amounts to delay or advance transaction timing by microseconds in an effort to commit fraud.

2.3 GPS Networks

GPS, while commonly used as standalone hardware, is also often used in the context of a network. Cellphone GPS units are tied into the Cellular network, aircraft GPS units are connected to a variety of aviation networks, and vehicular GPS units used in vehicle convoying are tied to the local area network (LAN) on the convoy. Each of these networks passes or contains an array of information in addition to GPS data. In the vehicle convoy network, sensor data from inertial measurement units (IMU) is often available as well as ranging data between network nodes provided by Ultra Wide Band Radio (UWB) or radar.

In this work, a leader-follower network is used as the basis GPS network for proof of concept, algorithm development, and testing. In a leader follower network, one vehicle, designated as the leader, travels along a trajectory usually driven under direct human control, though, in more advanced systems, this vehicle may be automated. The second vehicle, designated as the follower, trails the leader along the same trajectory autonomously or semi-autonomously. GPS waypoints, radar data, vehicle sensor data, and environment information are combined in a control algorithm which directs the follower along the optimal path behind the leader. In a multiple vehicle convoy, several vehicles occupy the follower position. GPS data is critical to the operation of a leader follower network. The precise positions provided by GPS allow the vehicles in the network to maintain safe trajectories and velocities. An undetected error of just a few meters could have catastrophic consequences. For this reason, it is vital to have robust methods of detection which will alert the user to possible spoofing and warn control algorithms to ignore spurious GPS information. If the spoofed GPS positions are incorporated into the control algorithm measurement updates, the resulting control commands will drive the attacked vehicle off the desired course as demonstrated in spoofing of an unmanned aerial vehicle [7]. However, if spoofing is rapidly detected, then the false GPS data can be ignored in the measurement update or the vehicle can be halted until the spoofing is suppressed.

Chapter 3

Spoofing Detection and Suppression Algorithm

3.1 Spoofing at the Signals Level

In order to fully explore defensive measures to detect and suppress spoofing, it is necessary to develop an understanding of the effects of spoofing at the signal level. How a spoofing attack may be conducted as well as its effects in the signals processing realm are critical to effectively detecting an attack.

3.1.1 Basic Spoofing at the Signals Level

The most basic forms of a GPS spoofing attack involve the broadcasting of realistic signals at a higher power than the authentic signals. The signals can come from rebroadcasting actual data sets as in the case of the accidental spoofing of civilian receivers at the Newark airport. The signals can also be generated in simulation by a user with knowledge of the GPS signal structure. Many commercial simulators exist which allow the creation of realistic GPS data sets at any defined location and time in the past.

Figure 3.1 shows the basic setup of a simple spoofer rebroadcasting collected data sets.

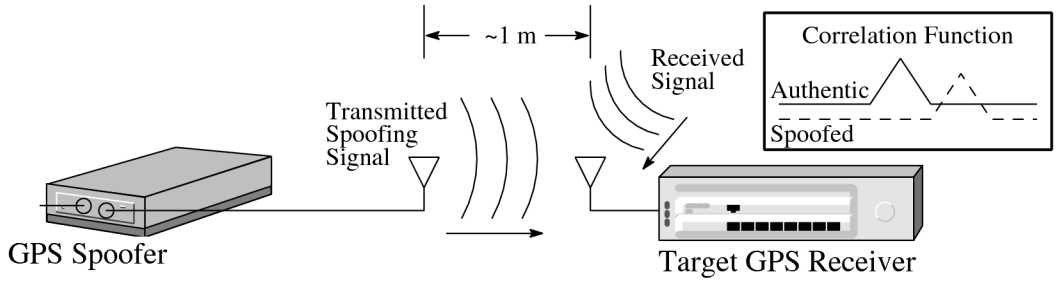


Figure 3.1: Basic block diagram of a simple spoofer.

In a recent publication of Navigation, Gunther explores the implementation methods and relative danger of various spoofing methods [1]. In the Navigation article, the initiation time of the attack is broken into three categories which will be helpful in the present work. The categories are cold start spoofing (CS), re-acquisition spoofing (RA), and tracking spoofing (TR). The designations come from the stage in which the receiver is attacked: either during a cold start, re-acquisition, or its tracking stage. A CS attack is, in general, the most easily mounted since the target has no knowledge of the authentic signals or its estimated current location. In a cold start attack, the receiver will lock onto whichever signals have the highest power, which in a properly coordinated attack will be the spoofed signals. In a re-acquisition attack, the receiver is somewhat, but not totally, hardened to the attack given the information stored in the receiver about estimated location, predicted satellites visible, approximate time, and other identification information. The final case presents the opportunity to defend the receiver. In this case, the receiver is in the tracking stage and hardened to any external sources. The receiver will not switch to spoofed signals unless the true signals are suppressed or the spoofed signal power is raised well above that of the true signal. This is due to the fact that in a basic spoofing attack, it is not possible to align, even remotely closely, the authentic and spoofed signals. The tracking delay lock loop (DLL) and frequency or phase lock loop (FLL/PLL) are locked onto the authentic signal and insensitive to the spoofing signals which are delayed by a minimum of several microseconds. Based on the signal structure of the GPS signal, shown in Figure 3.2 below, a microsecond of delay corresponds approximately to one C/A code chip width or 1575.42 carrier wave cycles.

In a basic attack, it is not possible for the spoofed signals to align or overlap with the authentic signals since the spoofer is limited to broadcasting signals with some finite delay. This limitation works to the defending receiver's advantage. In the acquisition sequence, a spoofing attack will be readily apparent. Since the spoofing peak is not aligned in at least one of the two axes in the acquisition plane, two different peaks will appear during an attack.

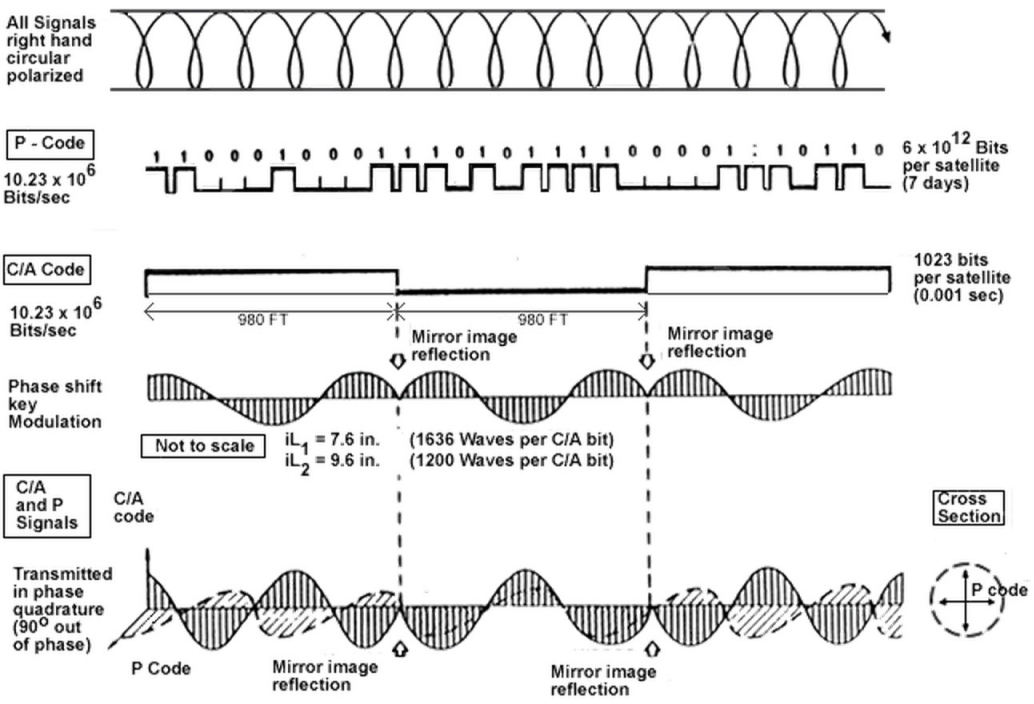


Figure 3.2: Signal structure of the GPS signal [25].

This plays a roll in the suppression method discussed in Section 3.4. An acquisition plane during a basic spoofing attack is shown in Figure 3.3 below.

In this form of attack, the spoofed signals act as noise when the receiver is locked onto the true signals. The same is true of the authentic signals when the receiver is locked onto the spoofed signals.

3.1.2 Synthetic Spoofing at the Signals Level

The signals analysis becomes more involved when more advanced spoofing methods are considered. The advanced techniques are lumped into a category referred to as synthetic spoofing for the purposes of this analysis, as all of the advanced methods rely on artificially manipulating or generating the signal to conduct an attack. In synthetic attacks, the aim of the spoofer is to capture the target receiver in a seamless manner leaving behind fewer traces of an attack. The attacker accomplishes this by simulating as closely as possible, the authentic signal at the target receiver's location.

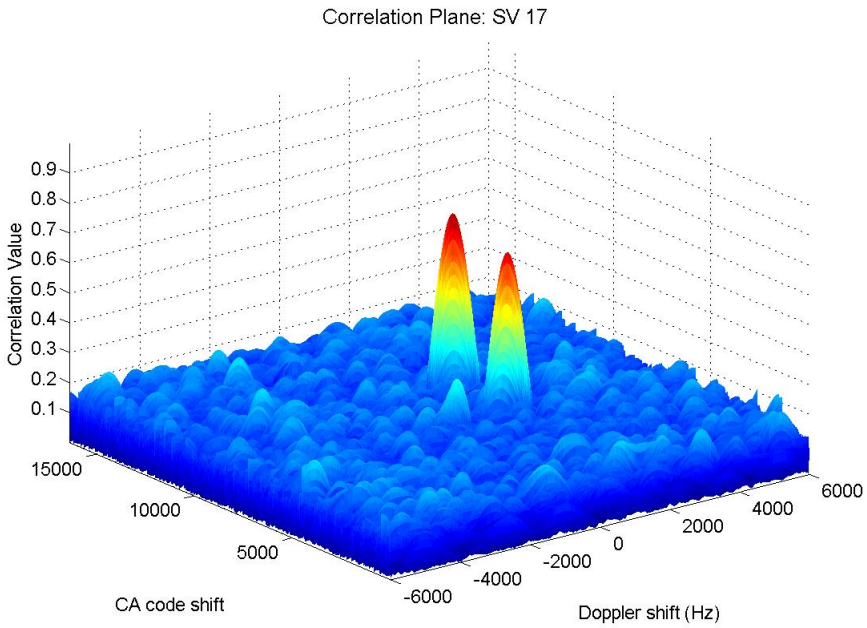


Figure 3.3: Dual peak in the acquisition plane caused by a spoofing attack.

When signals of identical or similar frequency are closely aligned in time, a well-known phenomenon known as beating occurs. This effect is shown in Figure 3.4 below. As the phases of the signal align, the magnitude of the combined signal is amplified while misalignment causes a decrease in amplitude.

Since the aim of the attacker is to align the true and spoofed signals, a synthetic spoofing attack will often induce this sort of beating in the resultant signal. Until the signals are closely aligned, beating is not an issue even if the frequencies are closely matched. At first, this seems unlikely but recalling that the signals are modulated by the C/A and navigation messages explains why this is the case. Although the signals will beat over short spans, the effect of the misaligned bit transitions mitigates this effect in the tracking loops and acquisition. The signals will not interfere constructively, or beat against one another, unless they are aligned by a delay of less than approximately $1\mu s$. This is equivalent to one C/A code chip. In reality, the separation may be slightly greater, due to noise and spreading effects, while still causing slight constructive interference.

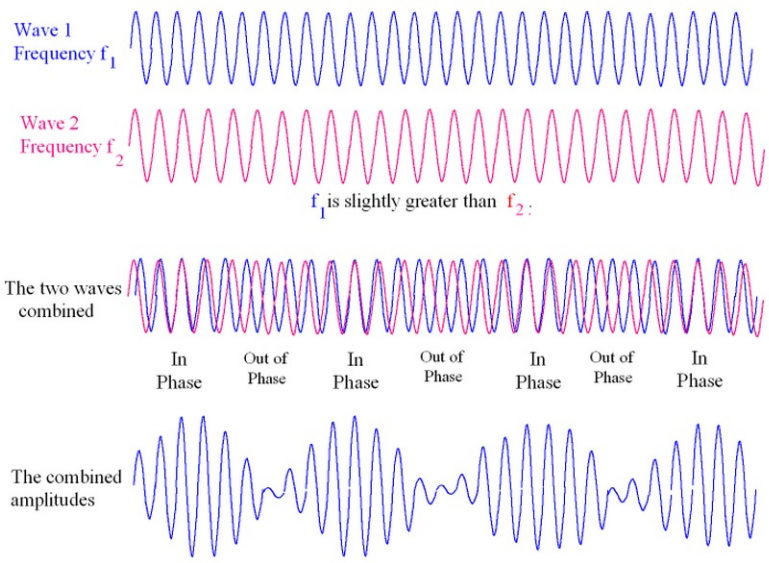
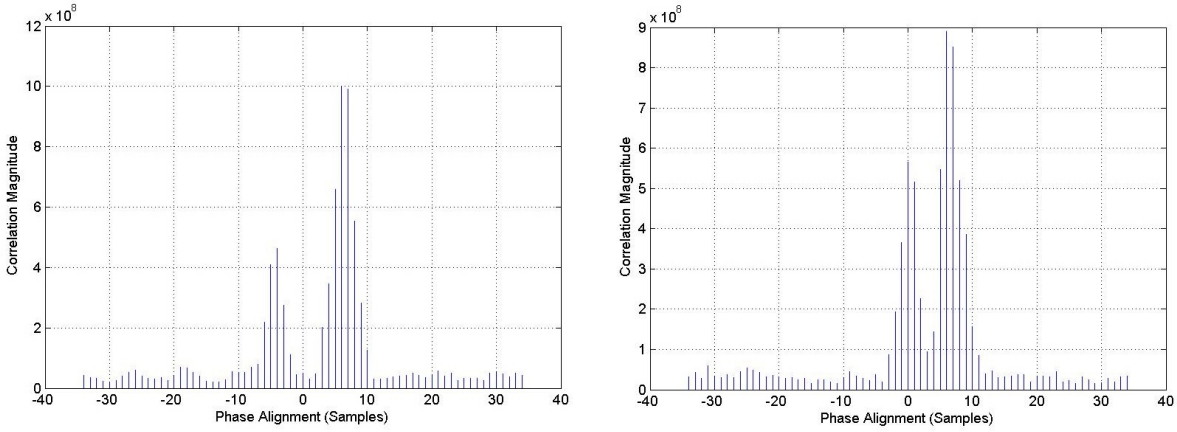


Figure 3.4: Beating phenomenon seen in two closely aligned waveforms.

The $1\mu s$ time separation between the real and spoofed signals is a critical spacing in many regards. It is at this separation that the real and spoofed signals are theoretically completely distinguishable in the acquisition plane as well as the the DLL correlation plane. If a signal is aligned inside this $1\mu s$ window, the tracking loops will struggle to distinguish between the real and spoofed signals. It is this weakness that the spoofer takes advantage of by hiding the spoofed peak under the authentic peak. Close alignment of dual peaks is shown in Figure 3.5 below. The figure was generated using two sets of simulated GPS data combined in software. The data generation process is described in more detail in Chapter 4 on testing and analysis. In the acquisition correlations shown in Figure 3.5, the spoofed peak can be seen approaching the authentic signal. The higher power spoofed signal will capture the receiver when alignment is within a single C/A chip or $1\mu s$.

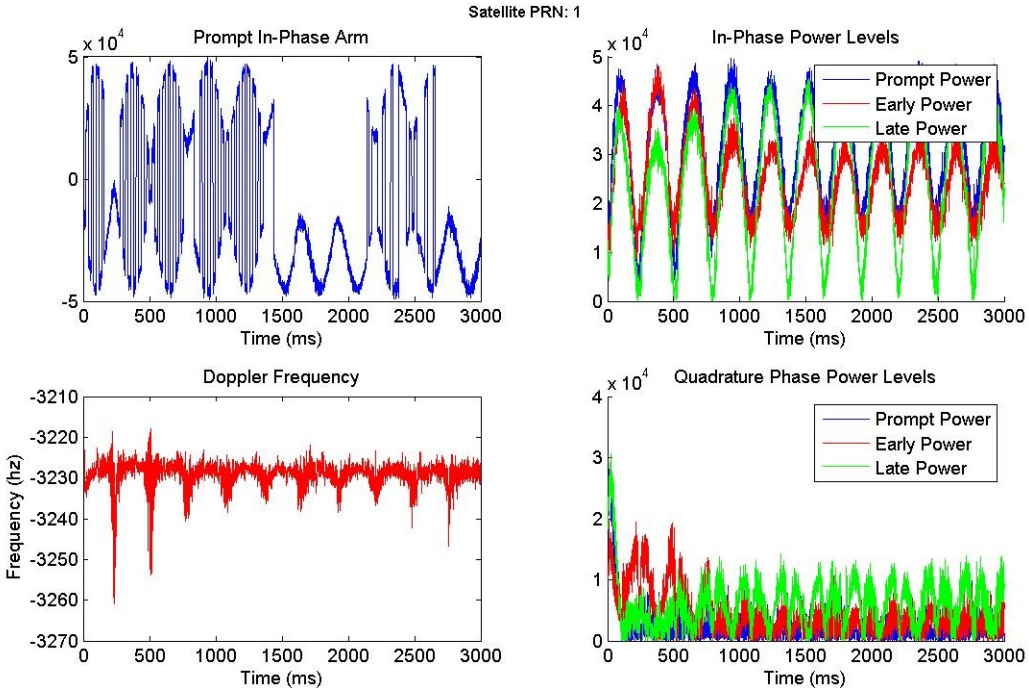
In the correlator outputs, beating can be clearly seen when the signals are aligned. If the power levels and corresponding amplitude A_R of the real and fake signals are similar, then the combined amplitude maximum is $2A_R$ while the minimum amplitude becomes nearly zero. As the spoofed power and corresponding amplitude A_S is raised, the maximum amplitude becomes $A_S + A_R$ while the minimum becomes $A_S - A_R$. This phenomenon is



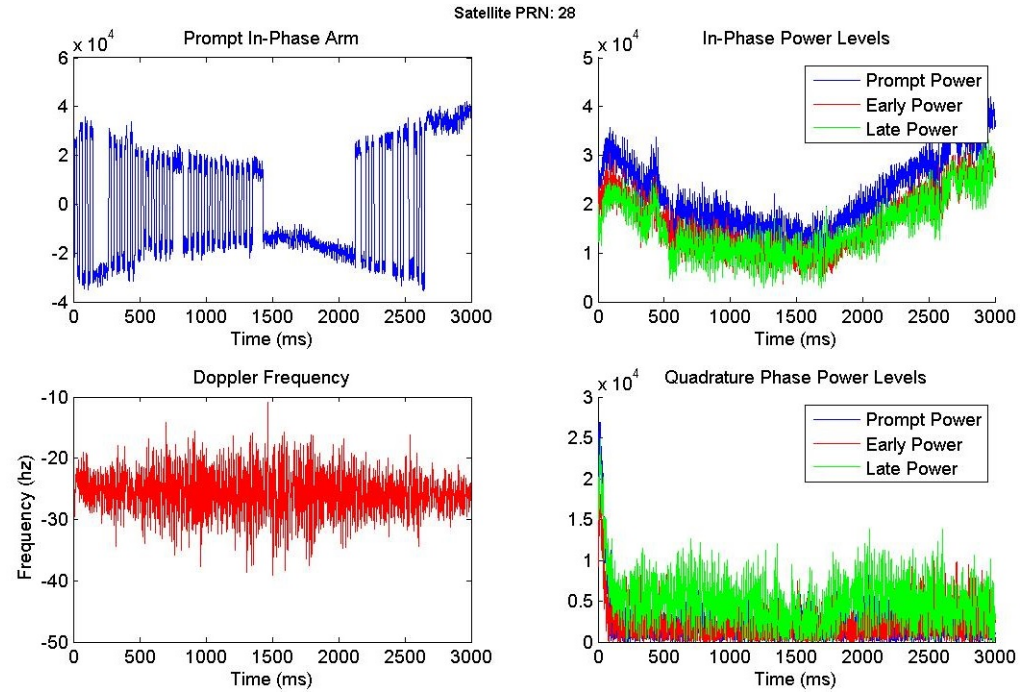
(a) (b)

Figure 3.5: Acquisition peaks in the code phase plane. In (a) the peaks are separated by a $2\mu s$ delay, in (b) the delay is decreased to $1\mu s$.

seen in Figure 3.6. It can also be seen that the beating frequency on each channel may be significantly different. This is due to the varying frequency present on each channel. In Figure 3.6(a) the beating pattern occurs much more quickly while in 3.6(b) the frequencies are more closely aligned leading to a slower beating pattern.



(a) Tracking loop outputs for PRN 1.



(b) Tracking loop outputs for PRN 28.

Figure 3.6: Tracking loop outputs for a closely aligned signal.

3.2 Mounting a Synthetic Attack

Having developed an understanding of the effects of a spoofing attack at the signals level, it is possible to describe the process of mounting an advanced spoofing attack and its limitations. Understanding how an attack is mounted will allow more precise and effective detection and mitigation methods.

A synthetic attack may be mounted using a number of variations. However, all synthetic attacks, as they are classified in this thesis, rely on the ability of the spoofer to generate replica GPS signals in order to capture the target receiver correlators subtly. A receiver can be considered captured if one of two criterion is true [7] .

1) The target receiver correlators are locked onto a non-authentic signal that is separated from the true signal by at least a $2\mu s$ delay.
2) The spoofed signal power is at least 10dB above that of the corresponding authentic signal.

As a result, there are a couple options open to a spoofer. Either the fake signal power can be raised well above the authentic signal power before manipulating the underlying navigation data bits and C/A code, or the C/A code can be shifted away from the true signal. The former is a combined jamming and spoofing attack while the latter is known as "drag-off" since it has the effect of dragging off the target receiver correlators. The drag-off method has a couple significant advantages. First, it does not leave noticeable power level variations which could indicate spoofing. Second, the interference of the underlying authentic signal is almost completely mitigated as the signals become separated in space. As a result, the focus will be on a drag-off type attack since this is more difficult to detect using more traditional power level monitoring.

Signal Alignment

As mentioned in prior sections, the aim of a synthetic attack is to align the spoofed signal closely with the authentic signal. In reference [7], the authors describe in detail the

development and implementation of a synthetic drag-off type spoofer. The attacker first collects authentic signal information and develops a data base of information on the signals. The attacker then uses the known target location and the information library to generate signals identical to those at the target position. The power of the spoofer remains low until the signals are aligned within a few meters to avoid detection. When alignment has been achieved, the attacker slowly raises the spoofing power slightly above that of the authentic signals. At this point, beating may be present offering the target receiver an opportunity for detection. When the spoof power is above the authentic power, the attacker initiates drag-off. This is achieved by shifting the fake signals forward in time by at least a few microseconds. Since the attacker has a stored data base of almanac and satellite information, predicting the incoming data bits is not very difficult allowing for signal generation and propagation forward in time. The forward shift is desirable since many receivers, if they detect two closely aligned signals, will track only the earlier one assuming that the second is multipath.

Correlator Drag Off

As the spoofed signal is shifted forward, the beating between the signals subsides and the target correlators are dragged off the true peak as shown in the diagram in Figure 3.7. When the critical 1-2μs separation between the signals is achieved, the target receiver is under the control of the attackers and the true signal now begins to act as a mild noise source.

In the process of dragging off a receiver's correlators, there are several issues that the attacker must consider. First, the beating of the signals discussed in detail above will tend to throw the tracking loops off both signals or back and forth between the two. This can be avoided by shortening the time that the receiver aligns the signals. By aligning the spoofed signal while the power is low and then quickly raising the power to slightly above that of the authentic signal the attack ensures a fast transition to the spoofed signals. When

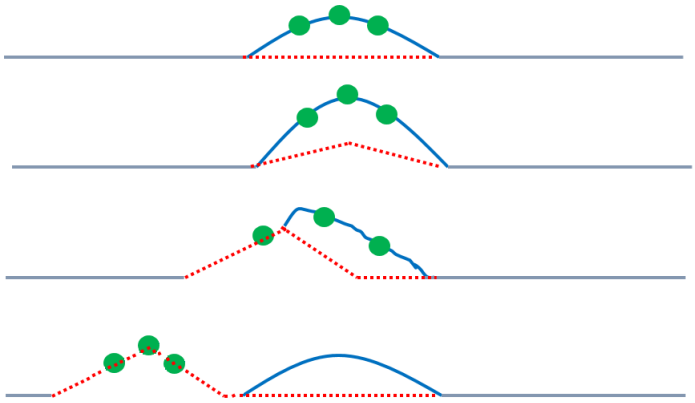


Figure 3.7: Drag off initiated in a DLL. Green Dots: tracking loop correlators. Solid blue line: combined authentic and spoofed peak at the true peak location. Red Dotted Line: Spoofed peak.

the transition has occurred, the signals must be separated as quickly as possible to avoid throwing off the tracking loops or alerting the user to an issue.

The second consideration to look at is how drag-off is initiated. If the attacker is not careful, the receiver will compute a false position solution, possibly quite far from the true receiver location, alerting the user to an attack. The aim of a synthetic attack is to make the transition from authentic to spoofed as smooth as possible in both at the signals level as well as at the measurements level. A sudden jump in position is a tell tale sign of a spoof attack and must be avoided by the attacker. To drag the target correlators, the C/A code must be shifted forward in time as discussed above. This can be accomplished most directly by simply shifting the entire signal forward in time. An advanced spoofer may desire to manipulate the C/A shift separately from the rest of the signal giving more variability in the form of attack. However, the effect of these two approaches is the same; the C/A code is shifted forward some small, finite amount greater than about $2\mu s$. At first, this may seem impossible to achieve without changing the position computed by the receiver since the pseudoranges will be shifted by at least $300 - 600m$. However, the spoofer here takes advantage of the fact that the ranges are not actual ranges to the satellite but rather relative ranges and a timing bias. As long as the attacker maintains a constant shift and

shift rate across all channels, the added range tacked on to each computed pseudorange will not affect the final position solution. The additional error gets rolled into the bias estimate and does not change the position computation. While it is not explored in detail in this thesis, this could pose an additional detection method if implemented properly. While the pseudorange change does not affect the position solution, a constant shift across all channels uncorrelated with receiver motion could be used to alert the user to a drag-off situation and allow detection of the attack.

Position Manipulation

Having captured the correlators, the target receiver is now completely under the attacker's control. The process of modifying the computed position can be accomplished using one of a few methods or a combination of the two. The primary and most effective method is by a simple manipulation of each individual channel. By delaying or advancing individual channels, the receiver computes new pseudoranges that drive the position solution away from truth. This process can be accomplished as slowly or as quickly as the attacker desires. In a rapid position manipulation, the attacker again runs the risk of alerting the user to a possible attack. Receivers that are coupled with an external device such as an inertial measurement unit (IMU) can still be captured by maintaining the induced position error rate withing the error bounds of the IMU error growth.

The attacker may additionally induce false motion by manipulating the underlying data bits. This can change the almanac information leading to a computation of false satellite positions or it could involve changing timing parameters, perhaps on a channel by channel basis to induce predictable error. To properly mount an attack using this method, the attacker must coordinate the changes in navigation message information across channels to avoid inconsistencies. The attacker must also ensure that the authentic and spoofed signals do not interfere or beat against one another alerting the user to an issue.

3.3 Detection of Spoofing

Having developed an understanding of the attack methods, their effect at the signals level, and intricacies of mounting a sophisticated attack, it is possible to begin analysis and development of various detection schemes. Given the reliance of many systems on GPS positions and timing, it is critically important to develop and maintain architecture capable of detecting spoofing attacks. If an autonomy algorithm which relies heavily on GPS inputs includes falsified inputs, the results may be disastrous. In this section, two methods of detecting and identifying spoofing will be presented. The first relies on external input from a local network. This has the advantage that deep access to the receiver is not needed. However, it requires a network with additional ranging information being passed across it. This is a common scenario in vehicle networks using vehicle to vehicle communications. The second approach utilizes an antenna array to detect changes in the angle of arrival of the signal. This method can be used independent of a receiver network but requires the use of an antenna array, at least two receiver cores, and an additional processing unit.

3.3.1 Network Anomaly

The concept behind the first spoofing detection method is a comparison of network data. Many types of data could be used in this sort of detection routine. Any sort of ranging information produced by ultra wide band radio (UWB), lidar, sonar, or radar could easily be used in this detection method. In addition, with modification, information provided by an IMU, visual odometry, or dead reckoning could also be used in this routine. In this thesis radar generated ranging information between nodes is used as the comparison data for spoofing detection. In a spoofing attack, only the GPS reported measurements are impacted. Other network information, most significantly the ranging information, is unaffected by a GPS spoofing attack. The spoofing detection algorithm leverages this fact to detect anomalies caused by an attack. In this section, several key aspects of the detection algorithm and its development will be developed.

Relative Position Vector Comparison

The first detection algorithm is based off of relative position vector (RPV) comparison. Given the reported GPS positions of each vehicle in the network, RPVs between each node can be calculated. In work by Auburn University, a dynamic base real time kinematic (DRTK) algorithm was developed to calculate RPVs between moving vehicles to centimeter level precision using GPS [10]. This moving base RTK system has been successfully used to provide truth systems for networked vehicles and is being explored for use in autonomy algorithms. The high precision enabled by the use of RTK technology makes its use advantageous to the precision of this detection method. If one or more GPS nodes have been captured, the reported GPS RPVs will not be accurate when compared to other ranging measurements. A threshold comparison of radar and GPS RPVs will reveal anomalies in GPS position solutions that indicate spoofing. The detection algorithm developed here calculates the GPS RPV at each iteration, compares it to the radar reported RPV, measures the error against a dynamic threshold and determines if spoofing has occurred.

To reduce the probability of false alarm (PFA), noise, other sources of error, and singular anomalies must be accounted for. A single measurement difference between the GPS RPV and radar RPV does not necessarily indicate spoofing immediately. GPS positions have inherent error as does the radar measurement. These errors must be accounted for as well as single measurement anomalies. For example, if something briefly moves in front of the radar obstructing the view to the leading vehicle, the radar will report no range or an error range. With no threshold against which to compare measurements, such an event will result in a false alarm, i.e. a detection of spoofing when in fact none has occurred. To decrease the PFA, a threshold of detection was developed.

Threshold Development

In a perfect detection setting, the radar and GPS reported ranges would contain no errors besides those induced by spoofing. Any difference between the two measurements then

would immediately indicate spoofing. In reality, noise affects both the radar and GPS measurements. The noise affecting each can be characterized statistically. For GPS, the standard method of reporting estimated or expected errors is in terms of a measurement called the Dilution of Precision (DOP) and the user range error. A more useful measurement for this application is the expected standard deviation of horizontal position. It should be noted that while receivers report DOP values with high precision, these values are statistically derived as there is no truth against which to compare the calculated position. The DOP values are measurements of expected error not actual error. As a result, the standard deviations mathematically related to the DOP values are estimated standard deviations. The standard deviations in a local ENU reference plane are dependent on the two factors named above: (i) variance of the user range error, (ii) the dilution of precision term. The DOP is dependent entirely on satellite geometry and accounts for errors due to satellite locations [15]. The variances on position are given below.

$$\sigma_E^2 = \sigma_{URE}^2 EDOP^2 \tag{3.1}$$

$$\sigma_N^2 = \sigma_{URE}^2 NDOP^2 \tag{3.2}$$

$$\sigma_V^2 = \sigma_{URE}^2 VDOP^2 \tag{3.3}$$

Where σ_E^2, σ_N^2, and σ_V^2 are the variances in the east, north, and up direction respectively, $EDOP^2$, $NDOP^2$, and $VDOP^2$ are the dilution of precision terms in the east, north, and vertical directions respectively, and $sigma_{URE}^2$ is the variance of the user range error. For the purpose of threshold development, the vertical or up direction is not particularly important. There are larger errors in the vertical direction of GPS measurements and including these error estimates in the threshold would not benefit us since the motion of ground vehicles if primarily in the horizontal direction. Combining the east and north components into a horizontal error measurement, the following equations are developed.

$$E_H = \sqrt{\sigma_E^2 + \sigma_N^2} = \sigma_{URE} HDOP \tag{3.4}$$

$$HDOP = \sqrt{EDOP^2 + NDOP^2} \tag{3.5}$$

Where E_H is the root mean squared error in the horizontal, East-North, plane [15]. To determine the variance on the GPS RPV magnitude, the root mean square (RMS) horizontal errors of each vehicle are combined.

$$E_G = \sqrt{E_{H1}^2 + E_{H2}^2} \tag{3.6}$$

Here E_G is the GPS RPV error and E_{H1}^2 and E_{H2}^2 are the root mean squared errors in the horizontal plane for vehicle one and vehicle two respectively. For networks with more than two vehicles, the GPS RPV error is calculated on the RPVs between every vehicle. The GPS RPV error is calculated in units of meters and represents the one sigma expected error of the vectors calculated between each vehicle. The variance of the radar RPV magnitude is determined based on the unit used, most commonly, meters. Combining the GPS RPV error with the radar RPV error using the root mean squared method yields the following equation.

$$E_C = \sqrt{E_{H1}^2 + E_{H2}^2 + \sigma_r^2} = \sqrt{E_G^2 + \sigma_r^2} \tag{3.7}$$

In Equation (3.7), E_C is the combined one sigma deviation of the GPS and radar RPVs and σ_r^2 is the variance of the radar RPV. From here, the determination of the threshold is dependent on the desired probability of false alarm (PFA) and strictness. Using E_C directly as a threshold against which to compare the difference RPV will lead to high PFA. E_C is the one sigma deviation of the difference between the GPS and radar RPVs thus, there is a 68% probability that this threshold will not be exceeded for each measurement. Conversely, the threshold will be exceeded 32% of the time. Using this tight of a threshold will result

in a false detection of spoofing often. In the development of the algorithm, the two sigma deviation was used as the basis threshold. For a single measurement, the PFA is less than 5%. However, as mentioned above, there are many events which could cause anomalies in the range measurements for a single iteration or a few in a row.

To address this, several successive measurements are combined to determine if spoofing is occurring. Choosing the number of measurements to include is dependent on the update frequency and desired rapidity of detection. Three to five successive measurements at an update frequency of 1 Hz were in the development of the algorithm. This yields a PFA of $(3.13 \times 10^{-5})\%$ according to the well-known Equation (3.8).

$$PFA = \prod_{i=1}^{N} P_i \tag{3.8}$$

Where P_i is the probability of false alarm for a single measurement and N is the total number of measurements.

Summary of Detection Algorithm

The detection portion of the algorithm operates, as stated before, by comparing radar produced ranges between nodes to the equivalent ranges calculated from the reported GPS positions. If a difference is detected between these two ranges that is greater than the threshold developed and persists for more than a few measurements, then the algorithm determines that spoofing of the GPS position has occurred and designates specific correlators to track the spoofed signals. Data describing the spoofed signal is extracted from the correlators and passed to the spoofing wipe off module described in Section 3.4. The algorithm is summarized as follows.

Detection Algorithm

i) Take in radar range between nodes and GPS positions of each node and calculate the GPS RPV.

ii) Determine threshold for that measurement using the deviations of the RPVs.

iii) Compare the difference between the GPS RPV and the radar RPV to the threshold.

iv) If the threshold is not exceeded, then return to the first step. If the threshold is exceeded, then evaluate the succeeding measurements.

v) If multiple measurements exceed the threshold successively, then alert the user to spoofing and designate correlators to track the spoofed signal.

vi) Proceed to analysis of the spoofed signal so it can be removed from the incoming data.

3.3.2 Phase Difference Convergence

The second method outlined here makes use of common receiver outputs for detection, eliminating the need for access to the receiver hardware. However, to implement this method, two separate receivers and antennas are needed. For some applications, this is not practicable. In other cases, though, this poses no problem at all. In a receiver network, multiple receivers and antennas are already present. In a large scale stationary application such as might be found in a precise timing application, the addition of an antenna and receiver does not impose a huge burden compared to the benefit is supplies.

The phase convergence method is described in reference [8] and thoroughly developed in reference [26]. Variations on this method which require a more complex setup but only one antenna are described and analyzed in references [9,27]. The phase convergence method relies on geometric manipulation of the incoming signals. An antenna array leverages the fact that signals coming from different directions will reach the two antennas different times since there is a defined space separation between them. Figure 3.8 shows how the antenna array relies on geometry to manipulate the incoming signals.

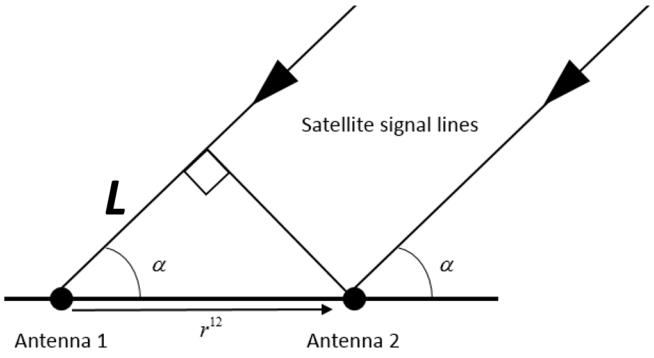


Figure 3.8: Dual antenna vector projection of LOS.

The signal arriving from a particular satellite will have to travel an extra distance denoted by L to reach the first antenna. As the figure shows, the magnitude of L is a function of the angle of arrival α. The added distance L results in a delay that is a function of α and the speed of light c_L. Since the satellite signals reaching the antenna arrive at different times, the phase of the signal at each antenna will differ. It is this variation that the phase difference method uses to detect an attack. The unspoofed phase difference between the antennas is given by the Equation (3.9).

$$\delta\phi^i_{21} = \phi^i_1 - \phi^i_2 = -\frac{2\pi}{\lambda}(\vec{r}^{\,i})P^T b_{BA} + \beta + 2\pi\delta N^i_{21} + n^i_{mp21} + n^i_{rcvr21} \tag{3.9}$$

In the case of spoofing, the equation is slightly changed. Equation (3.10) below shows the phase differences for the spoofing scenario.

$$\delta\phi^i_{21} = \phi^i_1 - \phi^i_2 = -\frac{2\pi}{\lambda}(\vec{r}^{\,sp})P^T b_{BA} + \beta + 2\pi\delta N^i_{21} + n^{sp}_{mp21} + n^i_{rcvr21} \tag{3.10}$$

In the phase difference equations above, $\delta\phi^i_{21}$ is the single difference carrier phase observable between antennas 1 and 2 on channel i. In the geometric term λ is a single wavelength of the carrier signal, $\vec{r}^{\,i}$ is the vector from satellite i to the antenna array centroid, b_{21} is the vector from antenna 1 to antenna 2, and P^T converts from reference coordinates to body coordinates. β is the line-bias-plus-fractional-differential phase term, n_{rcvr21} is the receiver thermal noise, and n_{mp21} is the multipath error. The only difference between the

two equations can be observed in two terms. First, in the spoofed case, the LOS vector becomes $\vec{r}^{sp}$ denoting the LOS to the spoofing module. Second, the multipath noise term becomes n^{sp}_{mp21} denoting the fact that all the multipath error have become the same since all channels are being propagated from the same source.

In a clear sky, diverse satellite geometry will result in many different α values, one for each satellite. As a result, there will be a diversity of L values and corresponding differences in phase between the two antennas. In a spoofing environment, the angle of arrival for all channels will be identical since the spoofer is broadcasting from a single location. In a very advanced attack (and one which is extremely difficult to mount) the attacker may utilize two or even three spoofing modules. In this case, the phase differences between the two antennas will converge two two or three values corresponding to the AOA of each spoofing element.

Figure 3.9 below shows the phase difference detection hardware setup utilized by researchers at Cornell and described in detail in [26].

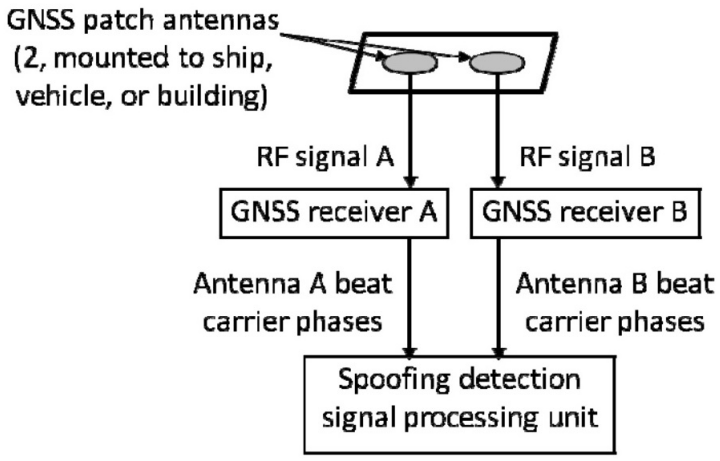


Figure 3.9: The phase difference detection setup used by Psiaki et al.

In this hardware configuration, two antennas mounted side by side feed into two separate receivers. The beat carrier phase measurements supplied by the receivers are fed into the spoofing detection software module which performs the difference computations and analysis.

An effect to that seen in the differenced carrier phase can also be seen in the difference between the pseudorange measurements on each antenna. The pseudoranges are differenced on every channel to determine if the signal is arriving from the same direction. In a clear sky environment, the differences will take on diverse values. In a spoofing environment, the measurements will collapse to roughly the same value on every captured channel. This method is particularly effective on antennas with relatively large line of sight distances between them since the pseudorange differences will vary more significantly when authentic signals are being tracked.

Data collected during a spoofing attack demonstrates the effect of spoofing on the differenced carrier phase and pseudoragne measurements. Figure 3.10 below shows the differenced carrier phases and differenced pseudoranges during a clear sky unspoofed scenario. The diversity of the measurements demonstrates that an attack has not occurred and that the signals are arriving from different sources.

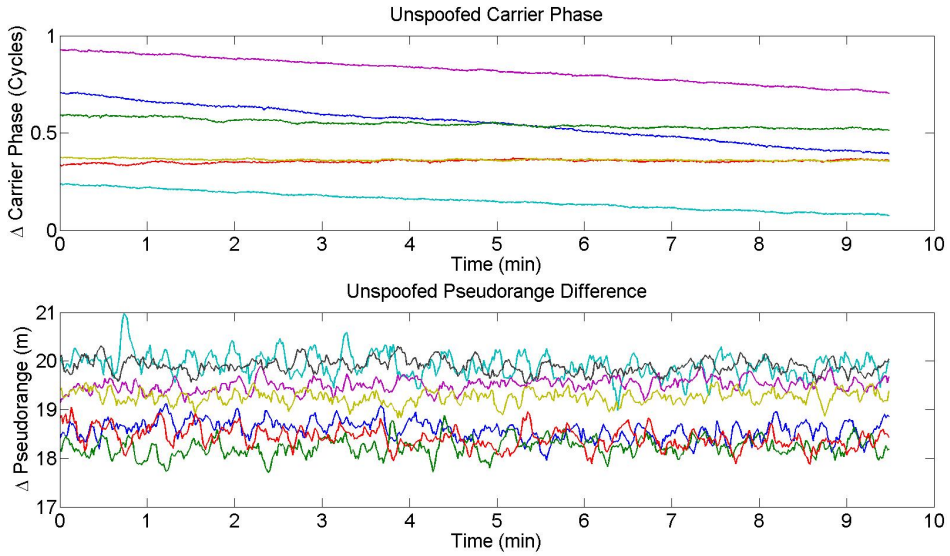


Figure 3.10: Carrier phase and pseudorange difference for multiple channels during clear sky.

When an attack is mounted, the differences across antennas collapse to approximately the same value on every channel as the receivers begin tracking signals all coming from the

same source. Figure 3.11 shows the carrier phase and pseudorange differences during an attack.

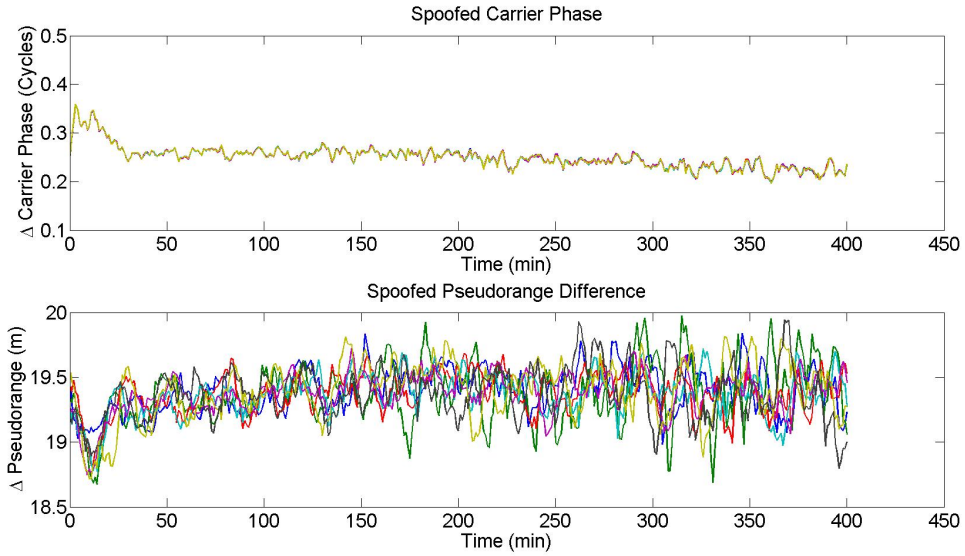


Figure 3.11: Carrier phase and pseudorange difference during a spoofing attack.

In applying the phase and pseudorange difference method, the detection algorithm tests a hypothesis of diversity versus a hypothesis of singularity to the measurments. The diversity hypothesis corresponds to an authentic, live-sky scenario while the singularity hypothesis corresponds to a detected spoofing attack.

Variations on the Phase Difference Detection Method

The detection setup shown in Figure 3.9 is not the only configuration for the dual antenna phase difference detection scheme. A similar result can be achieved by using a single receiver and dual antennas. This method is also described by Psiaki in [26]. The concept of using a phase difference is still the same, however, the measurements are generated by the same receiver connected to a switching antenna. The measurements for one antenna are generated, the antenna input is switched at the RF switch and the phase measurements are computed again. The predictable switching pattern can be used to test the spoofing hypothesis. This configuration is shown in Figure 3.12 below.

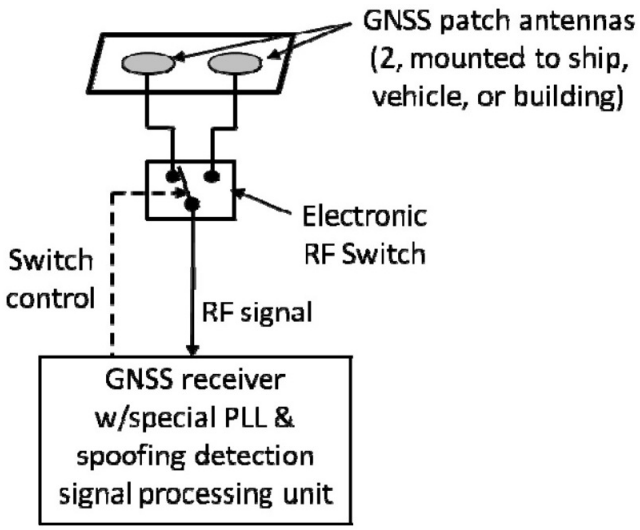


Figure 3.12: The switched antenna phase difference detection setup.

3.4 Successive Interference Cancellation

Having detected and tracked the spoofed signal, the next step is to prevent the spoofer from maintaining control of the receiver. As soon as spoofing is detected, the false position information can be ignored from the control algorithms in the network. This prevents the spoofer from driving the captured vehicle off course or along a dangerous trajectory. However, simply ignoring the bad GPS information is often not sufficient to solve the spoofing problem. If possible, an attempt must be made to recover the authentic GPS signals and suppress the spoofed signals. By ignoring the spoofed signal, the user may be able to track the true signals at least temporarily, however, a complete suppression scheme is much more robust and offers several advantages. First, a signal cancellation scheme allows the user to extract a significant amount of information from the attack which may allow for localization and suppression of the spoofer. Second, by canceling the signal, the user is maintaining lock on the false signal which will provide constant monitoring preventing accidental re-acquisition of the fake signal or loss of the authentic signal. Third, a suppression scheme at the signal level can allow for the development of an independent spoofing prevention module for use inline with a commercial of the shelf (COTS) receiver. Such a module would take receiver

outputs and, possibly using additional external sensor measurements, determine if spoofing has occurred. Designated correlators in the module would be assigned to track the spoofed signal allowing for suppression as will be described in the following sections. The output of the module would be a clean, RF signal that could be fed into a COTS receiver through the RF port.

Successive interference cancellation (SIC), as its name implies, iteratively suppresses an incoming interference signal on a channel by channel basis. Each acquired PRN is assigned an individual channel in the tracking stage which maintains lock on the spurious signal extracting descriptive parameters to allow suppression. From these parameters, a complete spoofed replica is created. The replica is then subtracted from the incoming raw signal to completely remove the effects of spoofing.

In the next sections, the important components of the SIC suppression method are described in detail. The network detection method outlined in 3.3.1 is combined with SIC and and a complete detection and suppression algorithm is developed in Section 3.5.

3.4.1 Tracking of Spoofed Signal

In Section 2.1.4 typical GPS tracking loops were described in detail. The correlators here are similar to the standard tracking loops but have a sole purpose of retrieving only enough information to recreate the spoofed signal.

The incoming spoofed signal has exactly the same parameters as an authentic signal and can therefore be acquired and tracked in the same way. As with any sine wave, three parameters fully define the spoofed carrier: amplitude A_s, frequency, f_s, and phase ϕ_s. Two additional parameters describing the C/A code bit value, C_s, and the navigation data bit value, D_s, complete the definition of the incoming spoofed signal. The aim of the designated SIC tracking loops is to extract these five parameters from the incoming signal thereby fully describing the signal at every successive iteration. The process is performed on every channel broadcast by the spoofer to enable complete suppression.

The tracking loops used in this thesis for parameter estimation are costas-type loops with small modifications for easy exctraction of the defining parameters. The carrier discriminator is an arctan discriminator given by Equation (3.11).

$$\phi_i = tan^{-1}(Q_i^k / I_i^k) \tag{3.11}$$

Where ϕ_i is the carrier phase error at iteration i, I_i, and Q_i are the i^{th} in-phase and quadrature accumulations over the tracking integration period respectively. The tracking integration periods are typically $1ms$ but can be extended in weak signal environments. The loop filter takes the phase error generated by the discriminator and produces an NCO output. The loop filter used is described by Equation (3.12).

$$NCO_i = NCO_{i-1} + \frac{\tau_2}{\tau_1}(\phi_i - \phi_{i-1}) + \phi_i \frac{T_{int}}{\tau_1} \tag{3.12}$$

Where NCO is the NCO update, i is the iterator, T_{int} is the integration accumulation period, usually 0.001 seconds, τ_1 and τ_2 are the loop coefficients calculated based on desired filter response as described in the following equations.

$$W_n = \frac{8BW_\gamma\zeta}{4\zeta^2 + 1} \tag{3.13}$$

$$\tau_1 = \frac{k}{W_n^2} \tag{3.14}$$

$$\tau_2 = 2\frac{\zeta}{W_n} \tag{3.15}$$

Where W_n is the natural frequency, BW_γ is the loop noise bandwidth, ζ is the desired damping ratio, k is the loop gain, τ_1 and τ_2 are the loop filter coefficients which are fed into the loop filter.

With this discriminator and loop filter, the phase of the signal is locked to the incoming signal allowing generation of $1\mu s$ segments of the carrier. The PRN replica is updated by a delay lock loop of the form described in Equation (3.16).

$$Code_{err} = \frac{\sqrt{(I_{E(i)}^2 + Q_{E(i)}^2)} - \sqrt{(I_{L(i)}^2 + Q_{L(i)}^2)}}{\sqrt{(I_{E(i)}^2 + Q_{E(i)}^2)} + \sqrt{(I_{L(i)}^2 + Q_{L(i)}^2)}} \tag{3.16}$$

$Code_{err}$ is the error in the code estimate, $I_{E(i)}$ and $Q_{E(i)}$ are the in-phase and quadrature accumulations as before over the integration period for the i^{th} interval.

Combined, the loop filters described above can provide the information needed to fully describe the signal at every interval. Figure 3.13 below shows how the information is extracted from the tracking loops. The in-phase and quadrature prompt arms provide power level information that can be used to compute the amplitude of the signal. The absolute value of the in-phase prompt arm provides the sign of the navigation data bit at that point. The PRN code generator produces a replica of the C/A code for that integration period. The code phase and carrier frequency can be extracted from the carrier loop filter commands to the NCO. Information defining all five parameters that describe the spurious signal are available from this tracking loop. Additional computations described below use the information available from the tracking loop to reconstruct the spoofed signal.

The power levels from the prompt in-phase and quadrature arms are used to compute the signal amplitude as mentioned above. This is accomplished using the following equations. The I_P and Q_P accumulations are computed as a integral of the replica prompt arm PRN multiplied by the estimated carrier wave over the accumulation interval. In discrete domain, this is a sum given by Equation (3.17).

$$A_P = \sum (PRN_{P(i)} Carr_i) \tag{3.17}$$

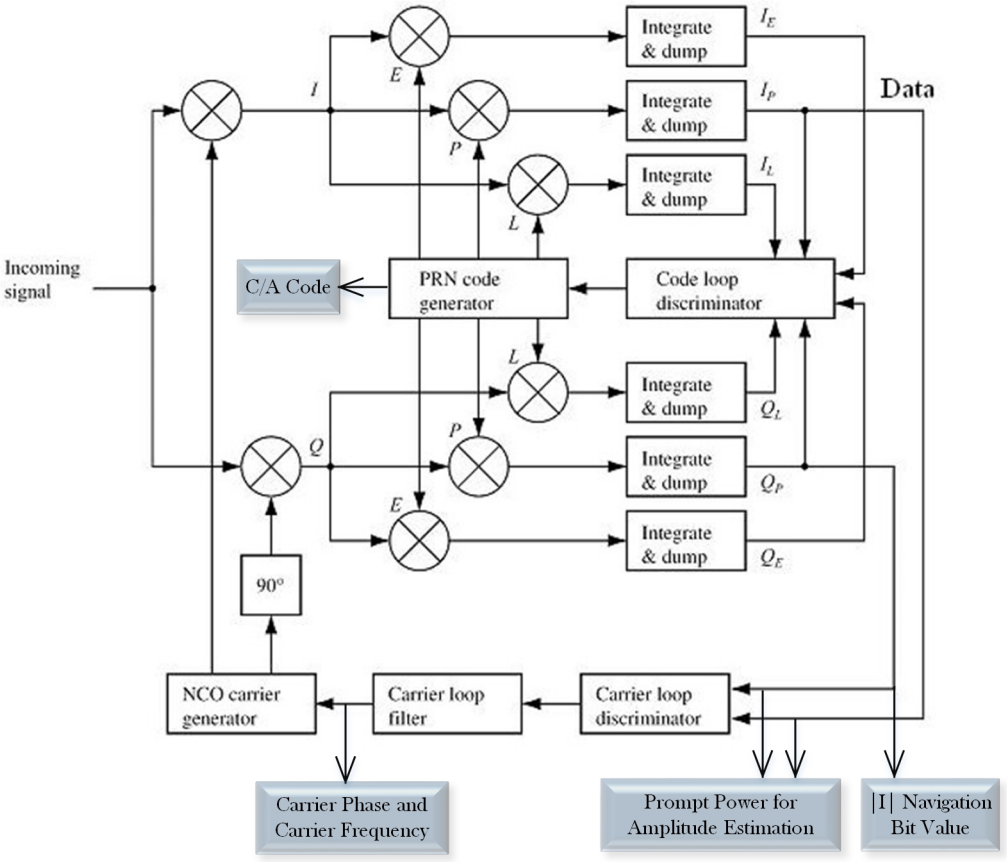


Figure 3.13: Parameter extraction loop for SIC.

Where A is the arm which could be either in-phase or quadrature, $Carr_i$ is the carrier replica, $PRN_{P(i)}$ is the prompt C/A estimate. From the I_P accumulation, normalized estimates of each sample are extracted according to Equation (3.18).

$$I_{ij} = 2\frac{|I_P|}{J} \tag{3.18}$$

The quadrature arm normalized estimate is calculated similarly according to Equation (3.19).

$$Q_{ij} = 2\frac{|Q_P|}{J} \tag{3.19}$$

Where i is the iterator denoting successive integration periods, j is the iterator denoting samples within each integration period, J is the total number of samples in an integration period, I_P and Q_P are the I and Q accumulations as before. Combining the normalized estimates, an amplitude estimation can be achieved using Equation (3.20).

$$A_s = I_{ij} + Q_{ij} \tag{3.20}$$

All the components necessary to reconstruct the signal are now present. The process of reconstructing and removing the spoofed signal from the raw incoming data is described in the next section.

3.4.2 Spoofed Signal Reconstruction and Removal

Having assembled all the necessary variables to replicate the incoming spoofed signal, it is possible to remove it from the raw IF data. The individual signal is reconstructed according to Equation (3.21) for every channel k.

$$S_i^{(k)} = A_i^{(k)} C_i^{(k)} D_i^{(k)} sin(2\pi f_{s(i)}^{(k)} + \phi_{s(i)}^{(k)}) \tag{3.21}$$

Where $S_i^{(k)}$ is the k^{th} channel, i^{th} iteration signal replica, $A_i^{(k)}$ is the amplitude of the replica, $C_i^{(k)}$ represents the C/A code for that segment, $D_i^{(k)}$ is the navigation message bit value, and the sine term is the carrier. The reconstructed signals, carefully aligned in time, are summed to recreate a complete replica of the total spoofing signal. This is shown mathematically by Equation (3.22).

$$S_{tot(i)} = \sum_{k=1}^{K} S_i^{(k)} \tag{3.22}$$

Where $S_{tot(i)}$ is the i^{th} segment of total reconstructed spoofed signal, k represents the channel counter, K is the total number of acquired channels which may change through time, $S_i^{(k)}$ is the i^{th} iteration of the signal on channel k. The successive interference cancellation algorithm is developed by combining all the elements above. Figure (3.14) below shows the complete block diagram for SIC.

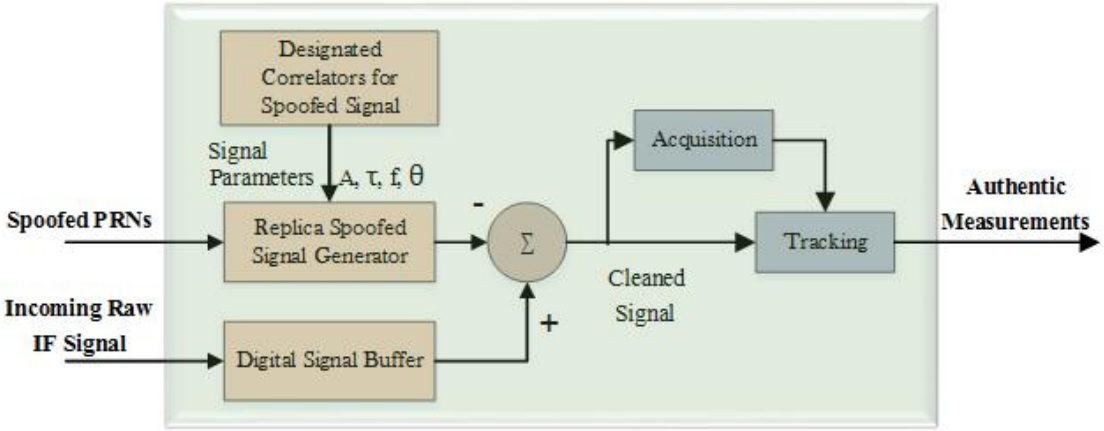


Figure 3.14: SIC algorithm block diagram.

As shown in the figure above, the designated SIC correlators maintain lock on the spoofed signal extracting the five critical parameters. The spoofed PRNs along with the parameters are fed to the replica generator which recreates estimates of the false signal for the segment. The incoming raw signal is fed into a digital signal buffer where it is aligned with the replica generated of the spoofed signal. The individual replicas for each channel are added together and the combined replica is subtracted from the buffered signal. The

cleaned signal is then fed to acquisition and tracking loops allowing for a corrected position solution computation. The total algorithm is summarized below.

Suppression Algorithm

i) Designated tracking loops calculate spoofed signal parameters A_s^q, C_s^q, D_s^q, f_{dS}^q, and ^q_s.

ii) The replica spoofing generator creates a replica signal according to Equation (3.21) and Equation (3.22).

iii) The replica signal R_s^q is subtracted from the raw IF data.

iv) The resulting cleaned signal is searched for the authentic GPS signals.

v) Tracking is performed when the authentic signals have been recovered.

3.4.3 Distinguishing Authentic and Spoofed Signals

In this thesis, the main focus is on detecting spoofing and developing an effective suppression technique. Less attention is given to the many methods which could be used to continuously distinguish the two signals when an attack has been mounted. Although not exhaustively explored, a couple methods are identified in this section which allow this distinction to be made. In future work, this is an area that should be expanded due to the high potential improvement it offers to effective suppression. In the signals realm, there is little to set the two signals apart particularly in an advanced attack. In Figure 3.15 below, data taken from a real attack scenario demonstrates that the real and spoofed signals look nearly identical in the acquisition plane. As a result, distinguishing which is authentic and which is spoofed must be accomplished by another method.

Although an attack may be detected by any of the methods described, there is always a possibility that not all channels on the receiver were captured. In this case, the identification of an attack must be on a channel by channel basis. Having detected an attack, the number of channels that have been captured must be determined. This is important information for

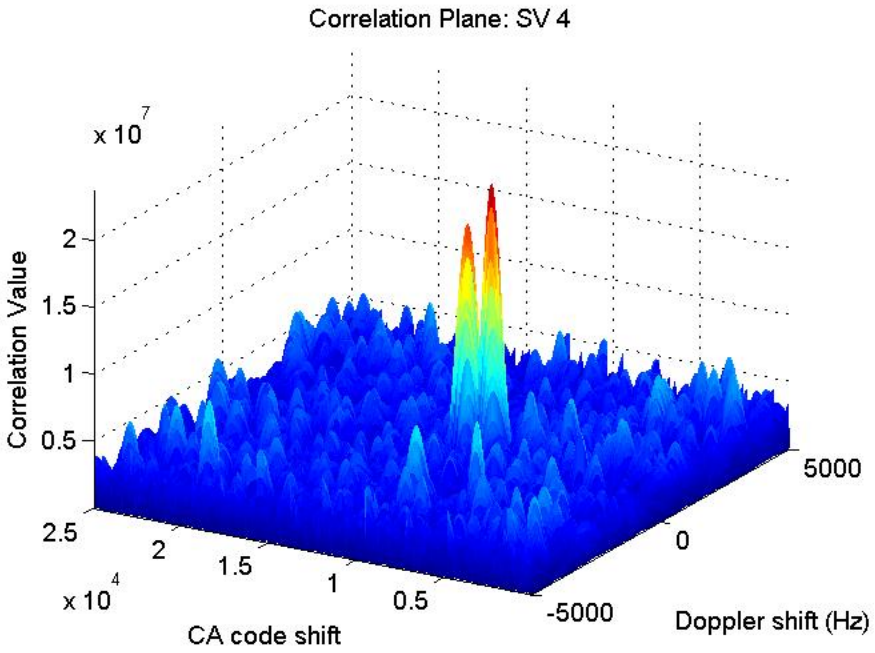


Figure 3.15: Correlation plane generated using real, live-sky data during an attack.

the SIC algorithm described above as it acts on a per channel basis. If a spoofing attack is detected but the attacker is only operating on half of the active channels, then activating the SIC algorithm indiscriminately will result in the removal of some authentic signals in addition to the false ones. Practically, such a scenario is unlikely. First, the aim of the spoofer is to capture all channels. Second, the power levels of the spoofer must be maintained above those of the authentic signal if the attack is to succeed. Both of these items together mean that the tracking loops in the SIC algorithm will track only the spoofed signal. However, in an uncoordinated attack or a unsophisticated attack, the possibility of capturing only a few channels exists. Therefore, having detected an attack, it important to be able to distinguish the authentic and fake signals on a per channel basis.

Identification of the false signal and differentiating it from the authentic signal can be achieved by several methods. The first method is one proposed by Broumandan and is based on correlating information across multiple channels with spatial movements of the receiver antenna [28]. Since the spoofer will be transmitting his signals from a single, or possibly two or three locations, motion of the receiver antenna relative to the spoofing antenna will

result in measurable changes in the receiver outputs which correlate across multiple channels. In a live-sky, diverse transmitter scenario, the outputs should not be correlated channel by channel. This allows the user to distinguish between channels which have been spoofed, and channels that have not. When no spoofing is present, the variations are uncorrelated. Using correlations of cross-channel measurements is an effective method of identifying spoofed and authentic signals on each channel. Those which show no correlation to any of the other signals are likely authentic while those showing correlation are spoofed. Although this is an effective method of distinguishing signals, it relies on the assumption that the receiver antenna is moving relative to the spoofer. If the receiver is stationary, there is no correlation of the signal variations across channels.

A second method relies on a similar principle to the first. Correlation across channels is used as the metric for identification but rather than using antenna motion to generate the patters, an antenna array is used. This method builds off the detection approach described in Section 3.3.2 and uses the results of the antenna processing to distinguish authentic and spoofed signals. In this method, the carrier phase difference between the two antennas will take on diverse values in an authentic signal environment. During spoofing, however, the phase differences will converge to a single value. Any channels that do not converge are likely still authentic signals. Even in a multiple spoofer environment, the values will converge according to the number of spoofers present.

One other method that, though computationally expensive may prove effective, is iterating through possible combinations of spoofed and unspoofed channels. This is not an optimal solution, but may be effective particularly in a network type setting where computational load may be distributed across multiple nodes. With knowledge of a spoofing attack based on information from one of the detection methods described before, the SIC algorithm may be activated removing all signals that the designated SIC loops are currently tracking. This will result in a new position solution and receiver outputs which may be checked by

testing them against the detection routine. In the network detection scheme, several iterations may be performed eliminating the signals most likely to be spoofed until a position solution which matches external network data is computed. This sounds complex but even with a completely random guessing scheme, this results in only 16 possible scenarios to test. There are two possible signals to choose from and only four channels are needed for a position solution. After a correct position solution has been computed, additional signals from the other channels may be added one at a time by testing each signal to see if it matches with the position solution. In a real attack, the signals will be able to be ranked according to whether they are likely authentic or likely spoofed based on several factors including origination time (time they appeared in the data), power levels, power variations, data bit authentication, and other factors. The origination time, or the time at which the signal is identified in the raw data, is a key piece of information in ranking. A signal which appears in the middle of data collection is very likely spoofed, reducing the number of iterations which must be tried before identifying the spoofed signal.

The methods outlined above are capable of identifying and maintaining a distinction between authentic and spoofed signals. The thorough development and testing of these methods is beyond the scope of this thesis, but each offers the ability to enhance detection and mitigation. In the next section, the network detection method is combined with SIC into a algorithm for detection and suppression of spoofed signals.

3.5 Algorithm Development

By combining any of the described detection methods with SIC, a complete anti-spoofing methodology can be developed. An effective system will offer robust detection in any design environment and suppression of the attacking signal. In cases where suppression has not been effective or is not possible, the user must be alerted to the fact that the incoming data may not be reliable. Taking all these requirements into account, the full development of a spoofing detection and suppression algorithm may be completed.

Any detection routine may be chosen based on the systems requirements or limitations. In this thesis, the network detection scheme is combined with SIC to create a complete anti-spoofing routine. This detection method offers highly reliable detection capabilities within its design parameters. It is intended for use in a GPS network similar to those used in vehicle convoying networks. It is directly applicable to any other network with contains GPS units and an additional ranging device such as radar, lidar, or ranging radios. For the development of the algorithm, a two-vehicle vehicle convoy network with radar measurements between vehicles will be investigated.

Detection was performed according to the network routine detailed in Section 3. A threshold was chosen to reduce the probability of false detection. As a result, the response time was increased according to the PFA equations describing the algorithm response. The probability equations are plotted in the figures below showing how the thresholds were selected. The figures show the probability of falsely detecting an attack, which, because the threshold is straight forward mathematically, is also the probability of missed detection. Figure 3.16 shows the full probability curves.

Figure 3.16: Probability of false alarm for various thresholds and sample windows.

Expanding the graphs to see the portion of the curves below the one percent level gives a better understanding of the thresholds. This is shown in Figure 3.17.

Looking at these figures, it is important to note that the x-axis is restricted to discrete integer values. Based on these figures, a two-sigma threshold was used as it allows for fairly rapid detection without tightening the threshold excessively. A one sigma threshold does not present a viable option as the probability of false alarm is still high even after five successive measurements. The two-sigma threshold offers variability to the user in allowing a cost-benefit trade off between speed of detection and probability of false alarm. The user may select the rapid detection and still have very low PFA or they may increase the detection time and reduce the PFA to statistically minuscule values.

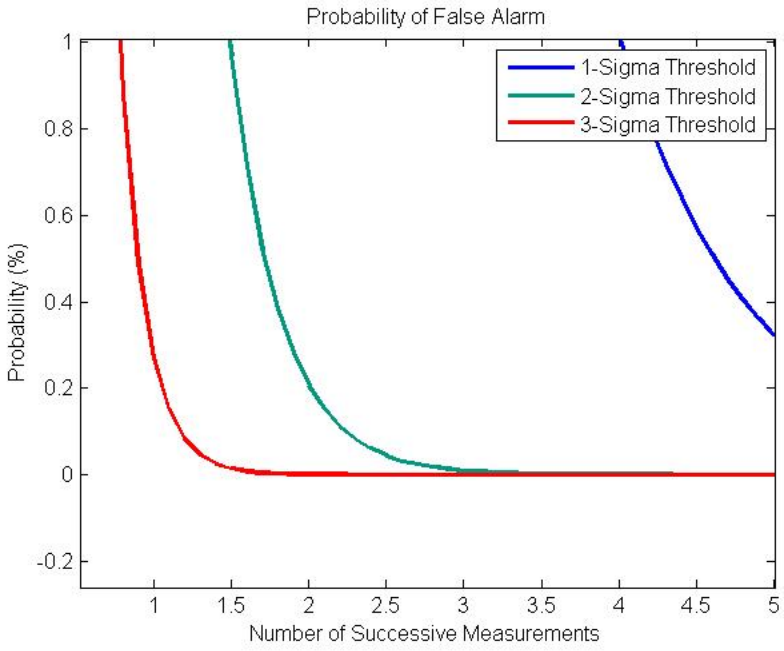


Figure 3.17: Probability of false alarm expanded graph.

Combining the systems described above, we have the completed detection and suppression scheme. Figure 3.18 shows a block diagram of the combined detection and suppression algorithm.

The detection module (denoted by the blue and gray blocks) takes in information from both vehicles where relative position vectors are computed and evaluated for indications of spoofing. The blue blocks represent the on board receiver and the network node on the first vehicle. The "second vehicle data" includes GPS positions and GPS dilution of precision estimates. In this system, the radar unit is on the first vehicle. If radar was present on the second vehicle, this would be passed over the network and included in the second vehicle data. The control outputs of the spoofing detection are used to determine if SIC (denoted by the tan blocks) should be activated. If spoofing is detected, the designated correlators are fed signal tracking parameters and a switch is activated changing the receiver input to cleaned data rather than the raw incoming signal. This algorithm gives anti-spoofing capability to users and comprises a complete detection and suppression routine.

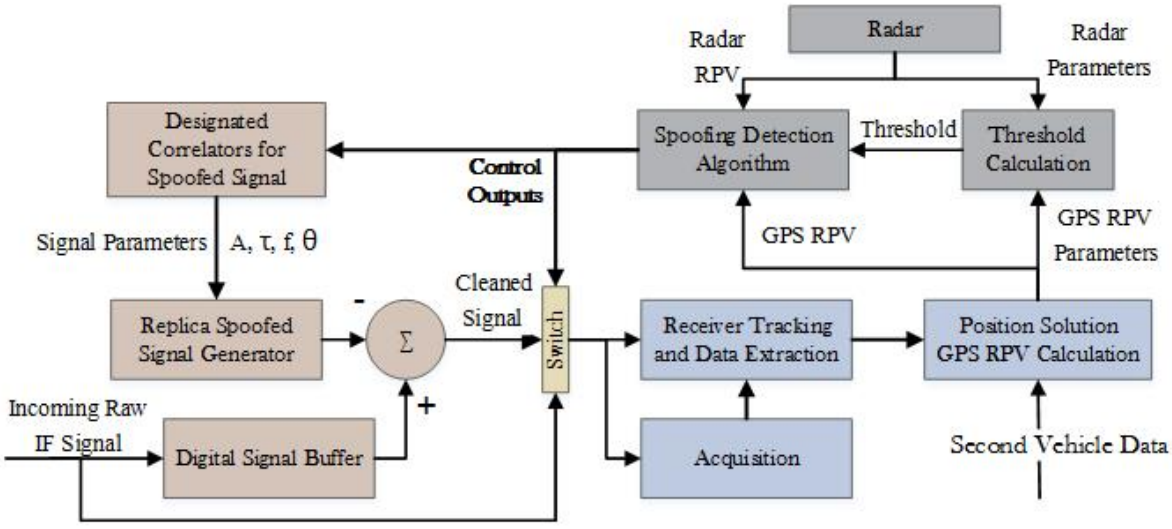


Figure 3.18: Detection scheme and SIC algorithm block diagram.

Chapter 4

Simulation, Testing, and Results

Testing an anti-spoofing routine is highly difficult on several levels. First, broadcasting spoofing signals with a power level high enough to capture a test receiver is against the law. This restricts testing to simulation or sanctioned test operations with government involvement or oversight. Second, the nature of a spoofing attack limits research capabilities. Since the attack variables are highly dependent on the type and location of the attack, it is difficult to generate a highly accurate simulation or model of the attack. It is possible to attempt live-sky simulation without broadcasting spoofed signals, but this is practically very difficult. At any point on the earth, there will be authentic GPS signals present. To simulate a spoofing attack in an open sky environment, the attacking signals must be perfectly synced with the authentic signals as they enter the receiver RF ports (direct injection is necessary to avoid broadcasting false signals) or the authentic signal must be suppressed. Both of the options are highly difficult. As a result, the best option available for research is complete simulation in which both the authentic and spoofed signals are generated. The effectiveness of simulation relies on the accuracy of the assumptions made. Since GPS is well characterized it is not difficult to develop precise simulated data reducing the amount of error in testing.

Given the difficulty of live sky dynamic testing, particularly in the case of a vehicle network, simulation environments were developed to test both the detection and suppression routines. Given the testing restrictions, all algorithms were written in Matlab and tested using post-processed data and code. However, the working prototype can be used to develop a real-time application since none of the software relies on future inputs. In the following section, methods of testing the detection routine are described along with results of each test. The suppression scheme is then evaluated in simulation and results are discussed.

4.1 Detection Methods Testing

To test the network detection scheme, a simulation was developed that models two vehicles in a convoy or platoon scenario. In this simulation, each vehicle has a GPS unit tied to a central network. The following vehicle is also equipped with a radar unit or ranging device. To created the simulated spoofing situation on two vehicles, real GPS data was collected on two vehicles traveling along a chosen trajectory. The radar data from these scenarios was simulated since this allowed for more variability in testing. A final iteration of the testing and development should include actual radar measurements and actively modified GPS positions on one vehicle to simulate spoofing.

4.1.1 Testing Scenarios

There are a variety of situations in which spoofing can occur. Below, each scenario in which the algorithm was tested is described briefly followed by the results and a discussion on the algorithm response. GPS data was recorded on two vehicles traveling along a two-lane road in an urban environment. The two vehicles were separated by 10 to 200 meters for the duration of the run. Vehicle positions were recorded at a 1Hz update rate and estimated deviations on the two positions were computed using the dilution of precision (DOP). Testing was performed to evaluate the response of the detection method in each scenario.

Single Follower Captured

In this scenario, a single follower in the network is captured by a single spoofer. This vehicle, and only this vehicle, is under the effect of the spoofer. The leader, and any other followers in the network are not directly affected by the spoofing. Such an attack is difficult since the spoofed signals must be directly propagated towards only the target vehicle without affecting other vehicles in the network, but it is not impossible. To simulate spoofing of one vehicle captured by a single spoofer, a simulation generated deviation from the original path was added to the data. The resulting position data shows the spoofed vehicle deviating from

the course during the duration of spoofing. Figure 4.1 shows this trajectory which results in one vehicle reporting deviation from the path only for a short segment. This poses a challenge to the spoofing detection module since the spoofing only occurs over a short time period on a single vehicle. The true vehicle path follows the lead vehicle marked in blue.

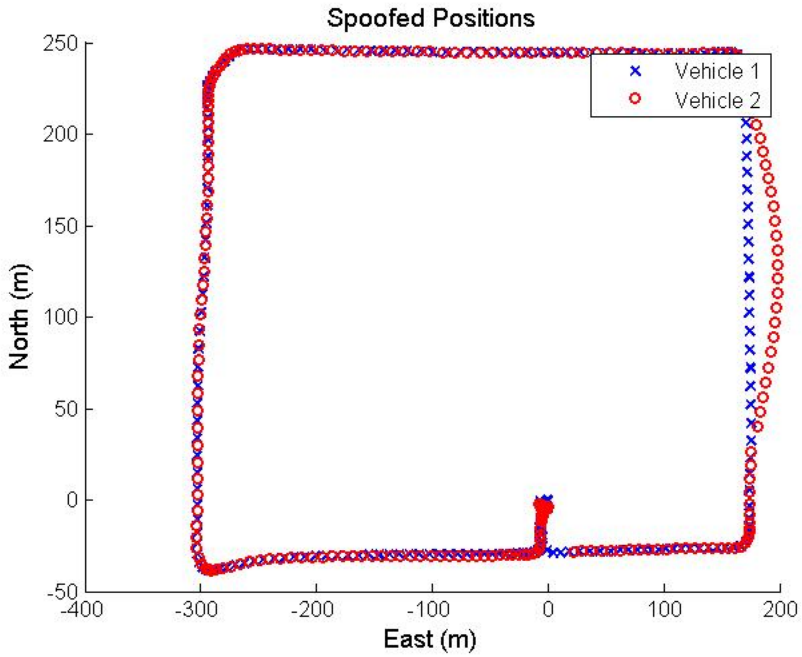


Figure 4.1: Spoofed trajectories of vehicles with one vehicle captured by a single spoofer.

During this scenario, the module detected spoofing within five samples of the first point where spoofing caused a deviation that exceeded the threshold. Figure 4.2 shows the ranges between the vehicles reported by the simulated radar measurements as well as those calculated based on the reported GPS positions.

About 180 seconds into the run, the variance between the two measurements can be seen to rise rapidly. This is visible in the first plot as the two values begin to spread. The lower figure shows the difference in the nominal range measurements compared to the dynamic threshold denoted by the green line. The deviance as spoofing is initiated is much more visible in the second figure as the range error rises sharply above the threshold and remains there for the duration of the spoofing. As the spoofing is deactivated, the algorithm returns to the original state indicating that spoofing is no longer present.

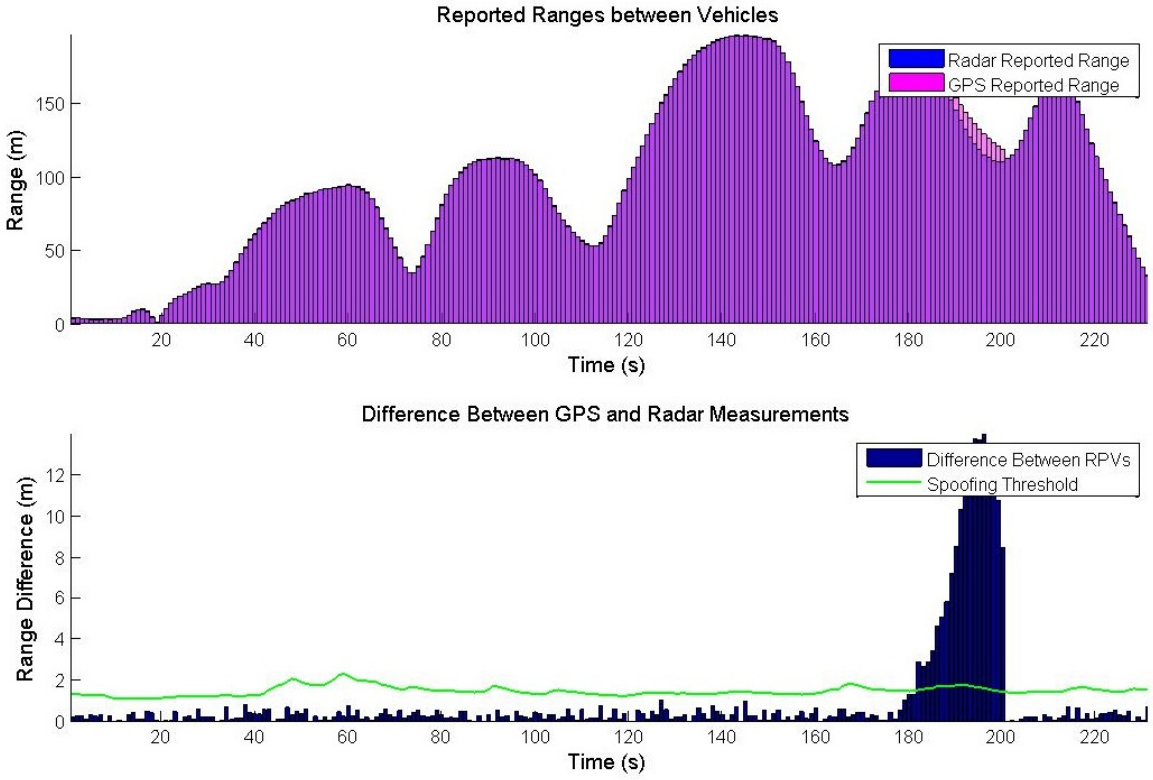


Figure 4.2: Relative position vectors and dynamic threshold outputs for first spoofing scenario.

The detection method worked well in the first test in which a single vehicle is captured by a single spoofer. In this case, the following vehicle was captured and deceived into thinking it was traveling a parabolic trajectory somewhat parallel to the true path.

Multiple Vehicles Captured Independently

For this second scenario, more than one vehicle in the network is targeted and captured in the spoofing attack. Each vehicle is captured separately. This sort of attack would require high levels of sophistication but is theoretically possible, and so, it is assessed as a possible threat to be mitigated. Such an attack would require multiple spoofers individually targeting vehicles. Each vehicle would initially be sent its own position to capture the correlators before being gradually dragged off its true position as the attack progressed. The trajectories used to test this scenario are shown in Figure 4.3.

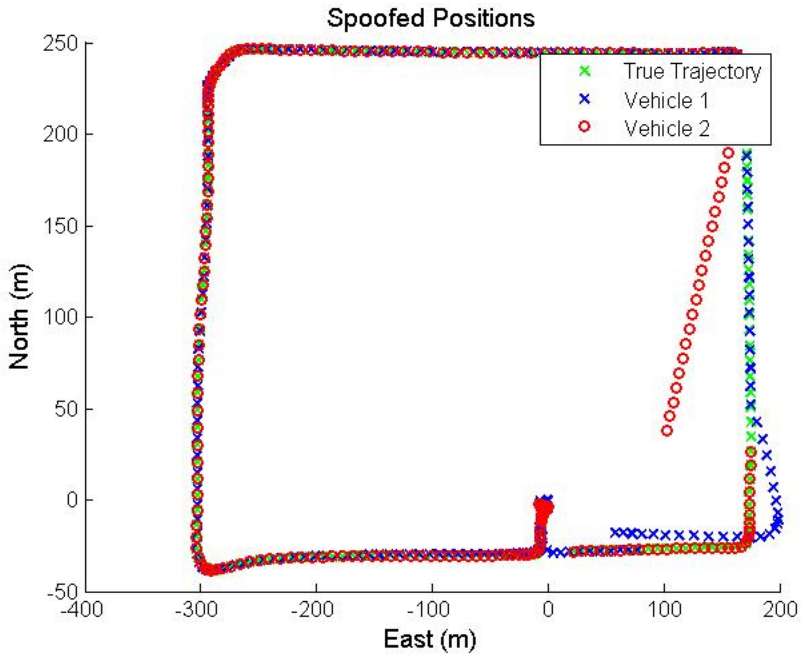


Figure 4.3: Trajectories of two vehicles captured by separate spoofers.

About 170 seconds into the run, a spoofer gains control of both vehicles. The spoofed trajectory of the first vehicle is semi-parallel to the true trajectory while the second swings back and forth across the true path. Figure 4.4 shows the ranges between the vehicles reported by the simulated radar measurements as well as those calculated from the reported GPS positions.

Once again, as spoofing is initiated, the variation between the radar reported ranges and the GPS computed vectors can be seen to rise. However, in this case, although the positions clearly deviate from the actual path as seen in Figure 4.3, the range difference does not significantly reflect this. The range difference does indicate spoofing for the entire segment of spoofing, however, for the first portion of the attack, the indicator is not very far above the threshold. This is a result of the fact that the indicator is an absolute or magnitude comparison. Even though the paths are drifting apart, the difference in the range reported by the radar and that reported by the GPS units is not rising significantly. Although the detection routine clearly indicated an attack, the scheme would be greatly enhanced in this scenario by the inclusion of a heading parameter. By comparing the heading indicated by

the radar to that indicated by the GPS, the range comparison would become a total vector comparison increasing the detection capability. Even without this addition, the scheme is quite robust as it is not possible for an attacker to maintain a trajectory which holds the radar and gps ranges within the threshold parameters for any length of time.

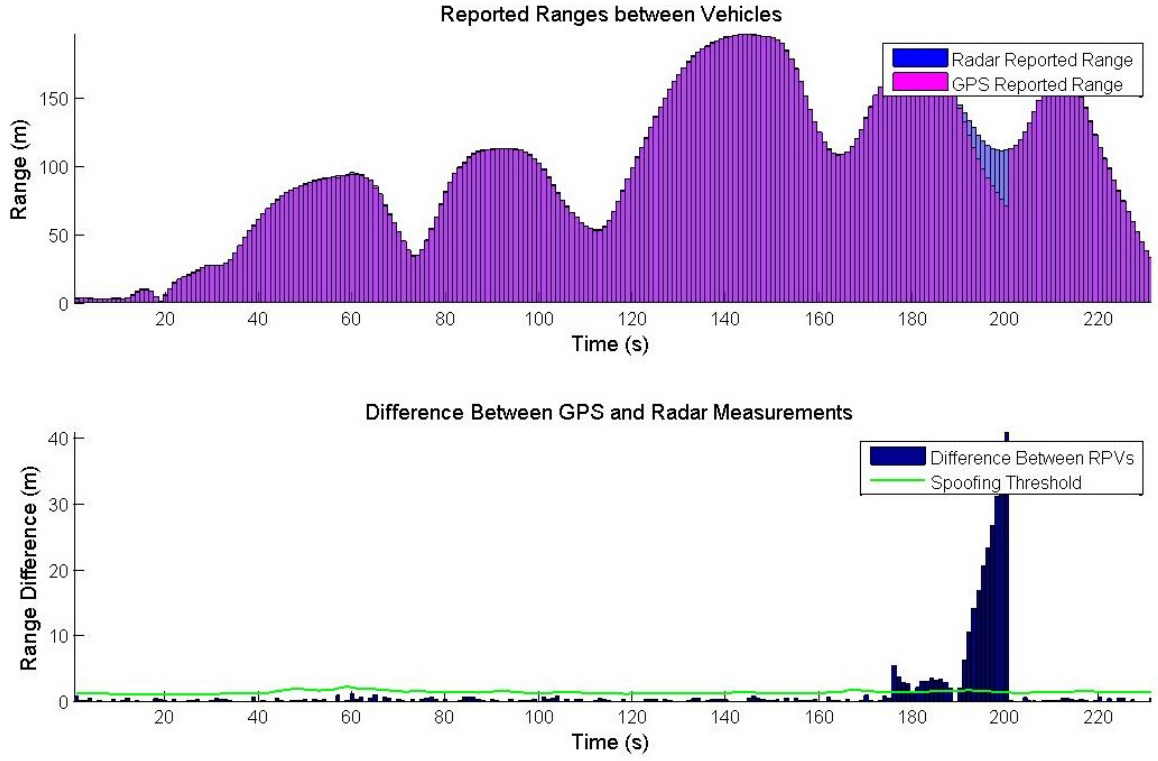


Figure 4.4: Relative position vectors and dynamic threshold outputs for second spoofing scenario.

Multiple Vehicles Captured Simultaneously

In the third scenario, multiple or all vehicles in the network are captured by the same spoofer. This results in all vehicles being dragged to the same position. Figure 4.5 shows the trajectory used to test this situation. The path of both vehicles can be seen to converge to roughly the same trajectory when spoofing is imitated. The spoofing is again only activated for a short portion of the path to test the response of the algorithm.

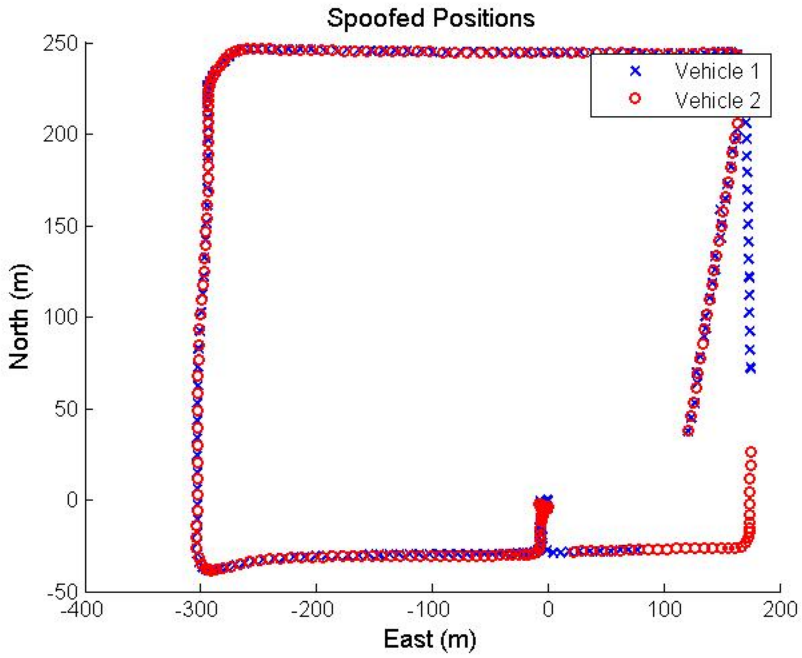


Figure 4.5: Trajectories of two vehicles captured by a single spoofer.

This sort of attack is more obvious to the user as at least one vehicle will have a sudden jump in reported position as the receiver is captured. The receiver may be unaware of the capture since the code phase and Doppler shift of the spoofed signal will still be relatively well aligned with the authentic signal. This event is likely to occur in many spoofing situations unless the attacker uses a highly directional antenna. The signal that is broadcast, even if it is highly sophisticated and calculated to target only a single node in the network, is likely to affect any other undefended GPS unit in the area. As a result, the target receiver as well as any other receiver that has been captured will begin reporting identical positions. The range difference and spoofing indicators are shown in Figure 4.6.

For this scenario, spoofing is detected rapidly and the certainty of spoofing remains high for the entirety of the run. The difference between the range magnitudes is over 100 meters for the duration of spoofing. This will always be the case for this type of spoofing attack since both vehicle are reporting the same position solution. The radar will not report a zero range so the magnitude of the spoofing indicator will be roughly the range between the

vehicles. The detection method works extremely well on this scenario, one likely to appear in a spoofing environment.

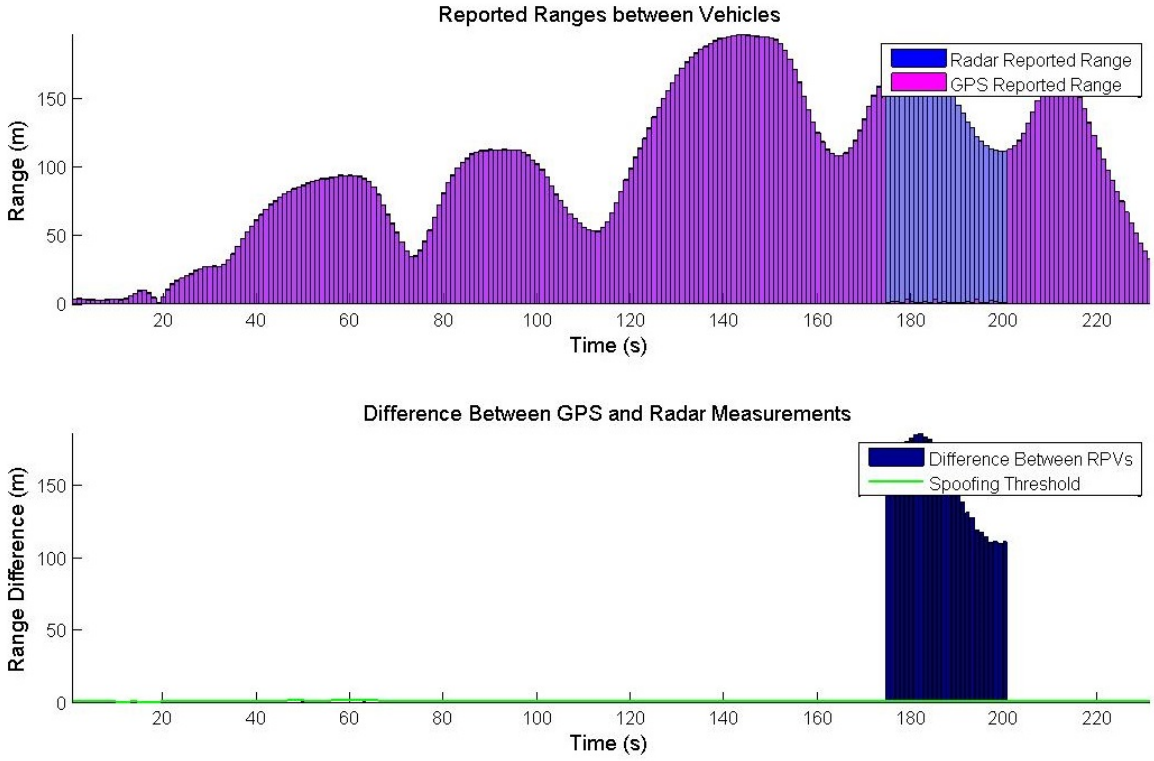


Figure 4.6: Relative position vectors and dynamic threshold outputs for second spoofing scenario.

Table 4.1 below compares the response of the algorithm in each of the testing scenarios described above. The network detection method proved highly effective in both the first scenario in which a single vehicle is captured as well as in the third scenario where as single spoofer captured both vehicles. The likelihood of a missed detection in these cases is low. The network scheme proved least effective in the scenario where multiple spoofers are able to independently capture both vehicles. The likelihood of a missed detection is low to medium since it is possible for the attackers to maintain the vehicle trajectories in a manner consistent with the radar measurements. The effectiveness of the scheme can be greatly improved in this scenario by including vector directions or heading in the threshold computations.

Table 4.1: Network detection evaluation table.

Scenario	Effectiveness	Likelihood of Missed Detection
1. Single Spoofer Single Capture	High	Low
2. Multi Spoofer Multi Capture	Nominal	Low-Mid
3. Single Spoofer Multi Capture	High	Very Low

4.2 Suppression Testing

The suppression scheme presented previously in Section 3.4 was coded in Matlab and built into a software defined, post-processing, GPS receiver. Evaluation of the suppression method was accomplished using a combination of a couple methods and data types. The successive interference cancellation (SIC) algorithm was first evaluated using purely Matlab simulated closely aligned GPS-type data to determine how effectively GPS signals could be forced below the noise floor. The data for this method was not actual GPS data. The navigation message was a random binary message and not and actual GPS navigation code. This allowed complete definition of the signal including exact knowledge of the true amplitude and other parameters to measure how effectively wipe-off was performed. In addition to using this matlab defined signals, a Spectracom GNSS simulator was used to generate simulated but realistic signals. This data generation method is discussed in more detail under each section. Results are discussed and the algorithm evaluated for each type of test data used. These testing methods, by using simulated data, allow evaluation of algorithm response to signals that are very closely aligned in code phase. This allows the establishment of limits on the ability to distinguish authentic and false signals and, correspondingly, limits on the application of SIC.

4.2.1 Evaluation with Matlab Simulated Data

The first method by which the SIC algorithm was evaluated used Matlab simulated GPS-type signals generated to represent spoofing on a single channel or PRN. The data

was not authentic GPS data but was instead randomly generated binary data modulated to a carrier along with authentic C/A code for a single PRN. This setup gives maximum variability in testing allowing the precise definition of amplitude, phase shift, code phase and, perhaps most importantly, noise parameters. By eliminating noise entirely from the simulation, the ideal response of SIC can be established. This method of data generation was used in the development of SIC to tune the tracking algorithms and wipeoff filters.

Using Matlab generated data, the authentic and spoofed signals can be closely aligned and even overlaid to evaluate the limits of SIC in terms of signal alignment. Simulated signals were generated at successively closer increments in terms of code phase providing insight into impacts on the acquisition plane as the signals align. Recall in Section 3.1.2, the beating effect of closely aligned signals was discussed. A similar pattern is seen in the acquisition plane as the signals begin to encroach on each other. To allow improved definition and more detailed analysis, the signals were generated in Matlab at a sampling frequency a factor of 32 times greater than the intermediate frequency. The IF was set to 1MHz resulting in a sampling frequency of 32MHz which yields roughly 32 samples for every C/A code chip. The correlation peak generated in acquisition of a PRN sequence appears, when sliced along the code phase axis to view only a single frequency bin, as a triangle or pyramid. This is a result of the fact that, although the code replica is not precisely aligned with the incoming PRN, there is still a residual correlation until the shift exceeds a chip width. The lower magnitude correlations that are not at the precise code phase are referred to as residual peaks. The residual peaks extend a chip width on either side of the primary peak. This effect necessitates the $2\mu s$ separation between signals to avoid interference. In actuality, because of the effects of noise and quantization, the actual limits are between 1 and $3\mu s$. Signals this closely aligned will be sure to interfere but only slightly unless the gap is closed closer than $1 - 2\mu s$.

Figure 4.7 below shows the correlation planes generated using the Matlab simulated data. Two identical signals were generated and aligned successively closer beginning at a

separation of roughly 2 chip widths or precisely $2\mu s$ and drawing closer until the two are precisely aligned. No noise was added to the data yielding perfect correlation peaks. In this process, because the data is noiseless, the interesting effects of interference can be clearly seen.

The initial offset of roughly 2 C/A code chip widths or $2\mu s$ results in almost no interference between the two signals. The peaks are clearly separated with only a small segment between the signals where the residual correlations add to create residual correlation peaks. As the chip offset is decreased to 1.75 chips, the correlation pattern changes as expected. However, when the separation is further reduced to 1.5 chips, the interference pattern becomes more defined. The tops of the peak appear as expected but there is a sort of trough between the two peaks that reveals how the signals interfere. At 1.5 chip width separation, the portion of the signals that is overlapping is also exactly out of phase resulting in direct cancellation between the signals at that point. This results in the trough seen in the figure. As the gap closes further to 1.5 chips, the portion of the signals that overlaps is now 90° out of phase resulting in no interfering effect. The two peaks taper off as expected resulting in no cancellation or subtraction of the residual peaks. As the gap decreases to exactly 1 chip offset, the portion of overlapping C/A code is exactly in phase and therefore interferes constructively. This results in a filling of the gap between the peaks as the correlations caused by the residual peaks add to exactly the magnitude of the true peaks which causes a sort of flat topped correlation peak. The pattern of constructive and destructive interference continues on a smaller scale as the gap is further reduced until the peaks are exactly overlaid resulting in a much higher magnitude single correlation peak.

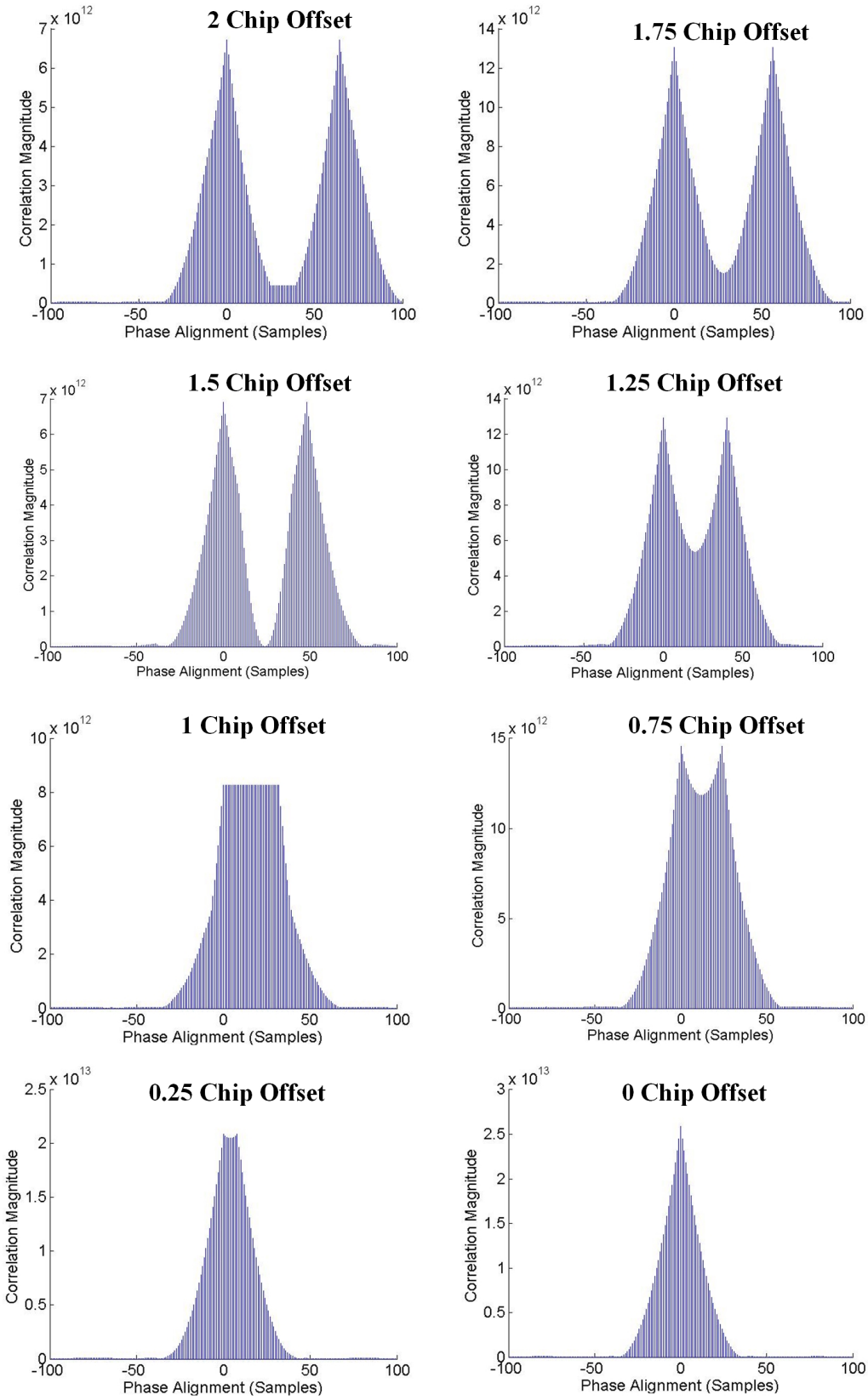


Figure 4.7: Correlation peaks for signals successively drawing closer in code phase.

Having generated signals closely aligned and precisely defined, it is possible to perform SIC to determine how effectively the algorithm is able to suppress signals as they interfere and beat against one another. First, SIC is performed on the signals separated by $2\mu s$ shown in the first pane of Figure 4.7. The software receiver easily tracks the noiseless signal producing the tracking plots shown in Figure 4.8. This figure shows the outputs of the tracking loops in the software receiver. The top left plot shows the prompt code, in-phase (I) arm outputs revealing the random data bits. The bottom left plot shows the doppler frequency converging quickly to zero where it remains for the duration of the test. The top and bottom plots on the right show the power levels of the in-phase and quadrature arm of the correlators respectively. The plots look unusual at first because of the perfect tracking as a result of no noise. But they show that the SIC tracking loops were able to perfectly track the spoofed signal.

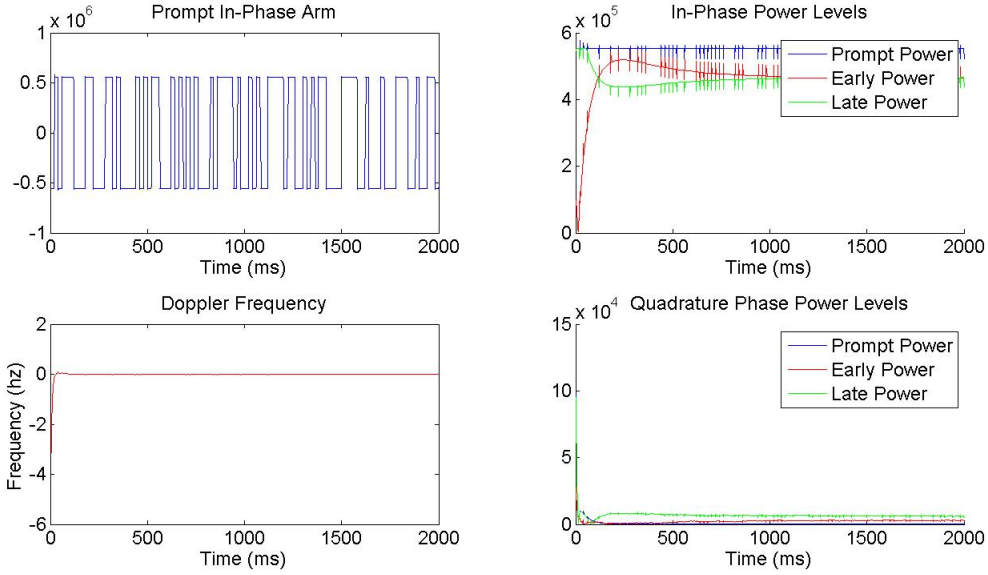


Figure 4.8: Tracking of noiseless Matlab generated signals.

This indicates that SIC will likely work on signals separated by at least a $2\mu s$ delay. Looking at the acquisition planes after wipeoff confirms this. Figure 4.9 shows the acquisition plane after performing SIC. The figure clearly shows that SIC has totally suppressed one peak without affecting the other at all. Tracking was performed on the remaining signal and

no negative effect was caused by the interference suppression. This result confirms that SIC is completely effective against signals with separation greater than $2\mu s$.

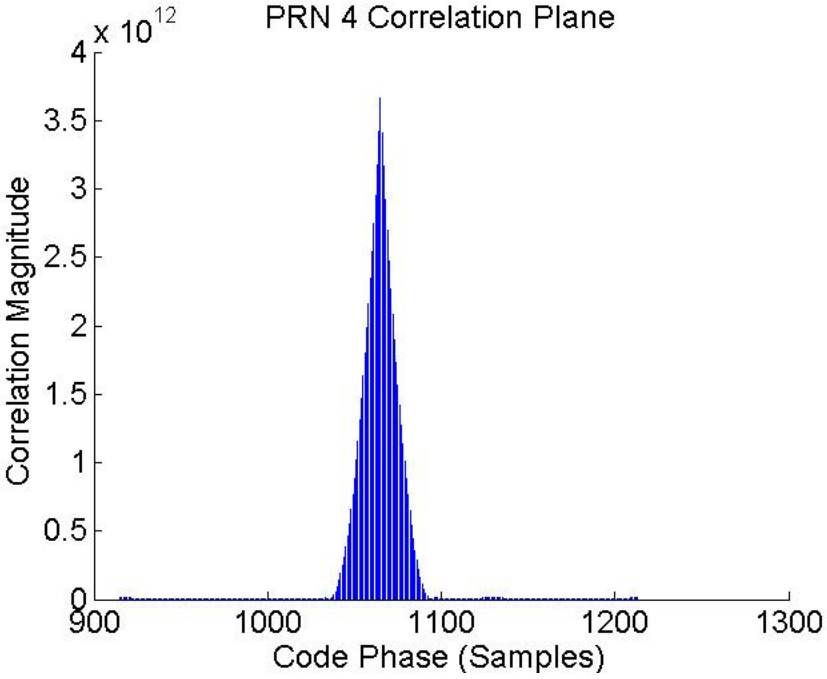


Figure 4.9: Correlation plane after performing wipeoff on signals separated by $2\mu s$.

The algorithm was then tested on signals aligned with an offset of $1.5\mu s$. The result is shown in Figure 4.10. Again, one signal is completely suppressed while the second remains totally intact. The tracking results also revealed that no unwanted damage to the remaining signal was caused by the SIC algorithm. SIC was also performed on signals with both 1.75 and 1.25 chip offsets with the same effect: complete suppression of one signal with no interference caused to the other.

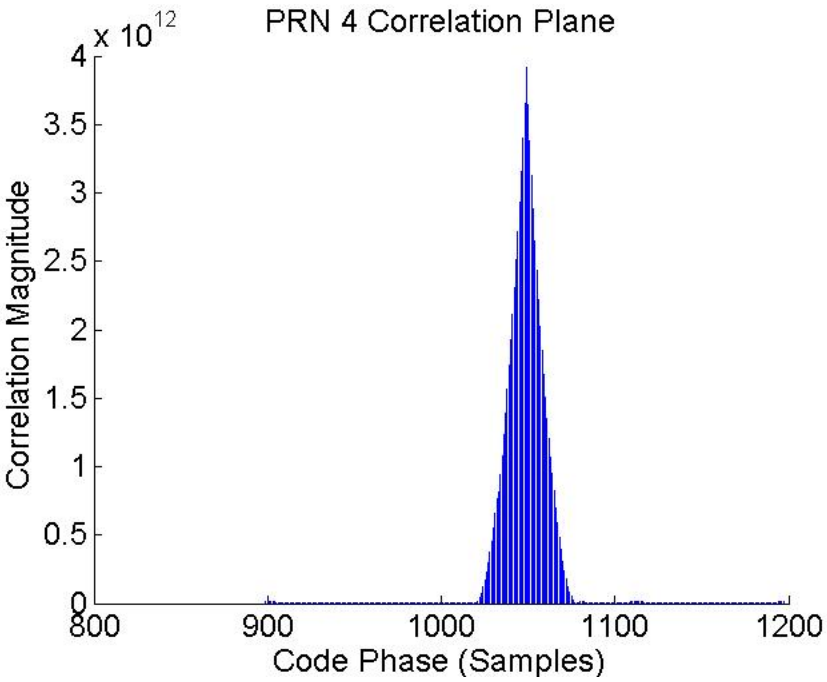


Figure 4.10: Correlation plane after SIC is performed on signals with a 1.5μs separation.

When the separation between signals is less than 1μs the interference between the signals becomes more noticeable and SIC becomes more difficult, however, it is still surprisingly effective. SIC was performed on signals aligned to 1μs and the resulting acquisition plane is shown in Figure 4.11.

Although SIC nearly completely suppressed the interfering signal, there are still some residual correlation peaks indicating that complete wipeoff of the spoofed signal was not achieved. The close alignment of the signals resulted in a partial removal and not total mitigation as was achieved in previous trials. However, although the signal was not totally eliminated, the suppression achieved was more than sufficient to drive an actual signal well below the noise floor.

SIC was then performed on signals separated by only 0.5μs. The correlation plane after suppression is shown in Figure 4.12.Again, SIC performs surprisingly well but is not completely effective at removing the encroaching signal. The residual peaks can be seen left in the place of the suppressed peak. The remaining signal is largely unaffected in the correlation plane and the software receiver was able to acquire and track it without trouble.

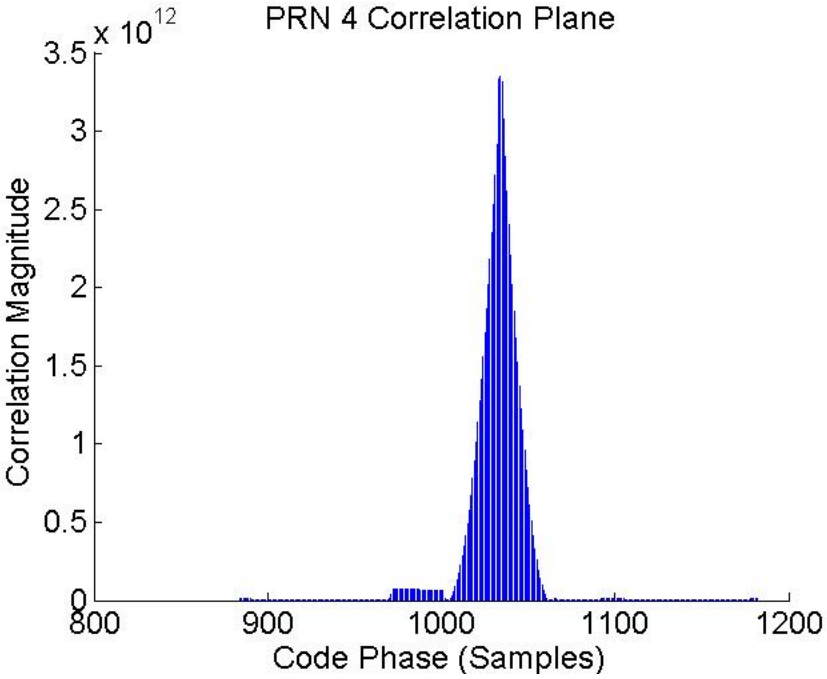


Figure 4.11: Acquisition plane after SIC on signals separated by $1\mu s$.

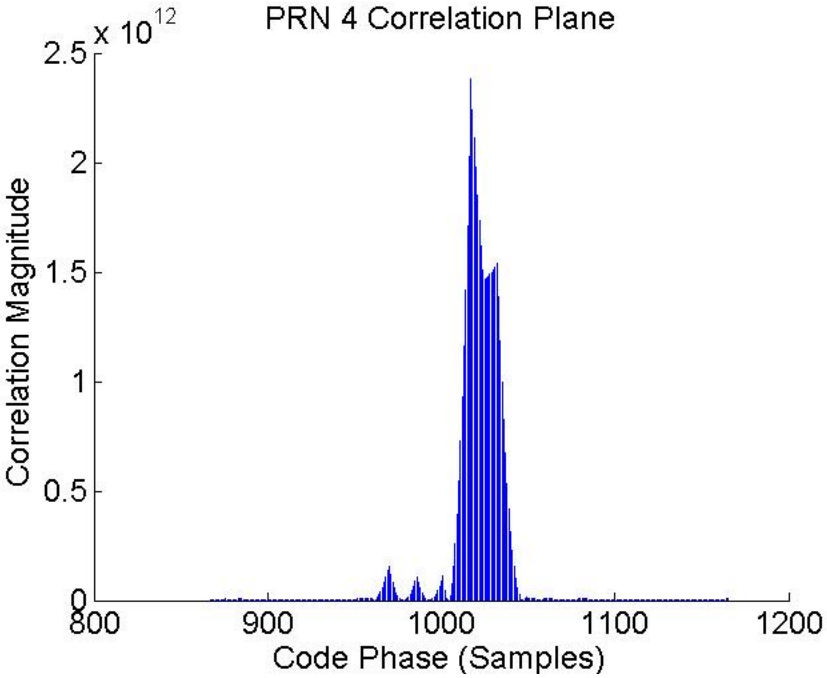


Figure 4.12: Acquisition plane after SIC was performed on signals separated by only half a chip width.

Testing SIC on these data sets demonstrated that it is effective even against signals that are extremely closely aligned. The $1 - 2\mu s$ limit discussed before prevents SIC from completely removing signals that are separated by a smaller delay. Signals with separation greater than $2\mu s$ experience no destructive interference at all and SIC is completely effective. For delays between $1\mu s$ and $2\mu s$, SIC works effectively on perfect noiseless signals. For delays less than $1\mu s$ the interference patterns between the signals prevent SIC from working perfectly but in situations where the close alignment is not prolonged, it may still prove effective. However, in an actual spoofing situation, the signals will not be noiseless. The noise and variations in the amount by which a signal is delayed result in beating patterns of interference. As a result, from the Matlab simulated data, it can be concluded that SIC is effective against signals with delays greater than at least $2\mu s$. Depending on the scenario, SIC can be effective on signals more closely aligned but this will be considered the alignment limit for which SIC is reliable.

4.2.2 Evaluation with Spectracom Simulated Data

Having established the ability of SIC to operate on signals which are aligned closely in time, complete sets of GPS data were created to evaluate the ability of the SIC algorithm to act across multiple channels simultaneously allowing the computation of an authentic position solution in a spoofing environment. Simulated spoofing data is ideal for this evaluation since it allows significant variability in the scenario generation. The power levels can be adjusted, both the spoofed position and the authentic position are completely user defined, and the timing delay can be precisely manipulated.

Simulated Data Generation

The diagram in Figure 4.13 below shows one method by which simulated spoofing data was generated. Two Spectracom simulators were programmed to output different position solutions at nearly identical times. The outputs were attenuated to achieve desired power

levels and then combined using a hardware combiner/splitter. The output was recorded using N210 USRPs with WBX daughterboards and written to IF data files. The data was recorded at either 5MHz or 25MHz and written as 8-bit integers.

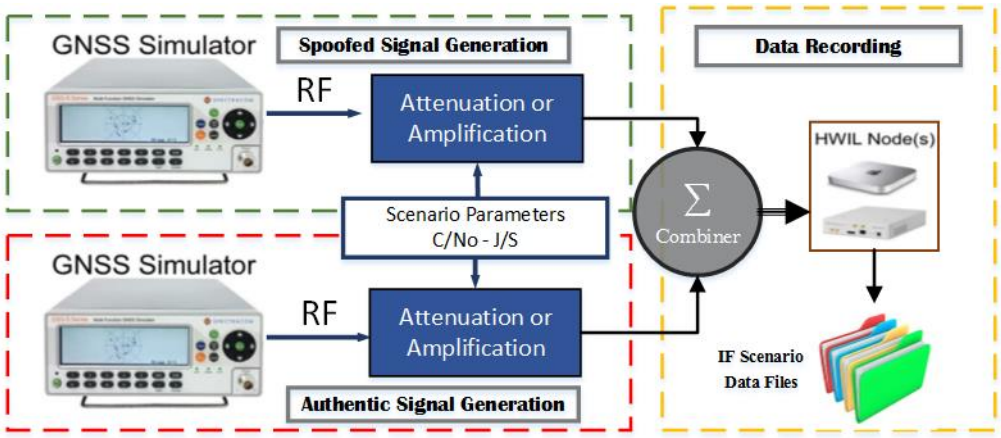


Figure 4.13: Generation of simulated spoofing data scenarios.

Using the method shown in Figure 4.13 introduced several problems. Primarily, timing the simulators to generate data simultaneously was not possible. Syncing the simulators to the same network eliminated the majority of the timing errors. However, to generate data at the precision levels needed to align signals to within a microsecond, this method proved to be ineffective. The start up delays varied too significantly between the simulators to effectively time the scenario generation. Replacing one of the simulators with a live-sky feed and setting the other simulator to "real-time" was also attempted. The delays in the simulator processing core proved too significant to make this a viable solution.

As a result, the method shown in Figure 4.14 was used as the primary simulated spoofing scenario generation method. In this setup, a single Spectracom simulator is used to generate both an authentic and spoofed data set. Recording on the USRP is triggered by an on-board GPS disciplined oscillator (GPSDO) which allows precise alignments of the data files. First, an authentic data set is generated in the simulator software package. The simulation is initiated and recording is triggered when the GPSDO determines that the broadcast time is equal to the chosen start time. A second data set representing the spoofer is generated using

the same time parameters as the first data set. Recording by the USRP is triggered by the GPSDO and the result is two data sets aligned precisely in time with different embedded position solutions. These two data sets can then be aligned in software with any desired delay. The precision of the delay is limited by the sampling rate. As discussed in prior sections, an actual spoofer is limited to delays or advances greater than at least $1 - 2\mu s$ to avoid beating patterns caused by signal interference. The software combination allows a resolution of $0.04\mu s$ with a sampling frequency of 25MHz and a resolution of $0.2\mu s$ with a sampling frequency of 5MHz.

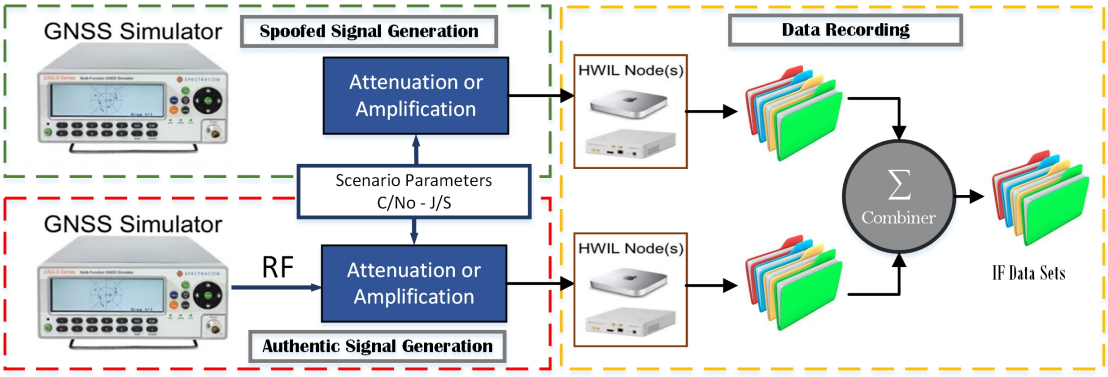


Figure 4.14: Software generation of simulated spoofing data files.

Scenario Testing and Results

The first data set generated using the combined data files incorporated a large spread in distances between the authentic and spoofed signals as well as a realistic delay accompanying that range. This is a scenario that would be encountered in a basic attack where the attacker is several kilometers from the target. Figure 4.15 shows the programmed positions designated as "authentic" and "spoofed."

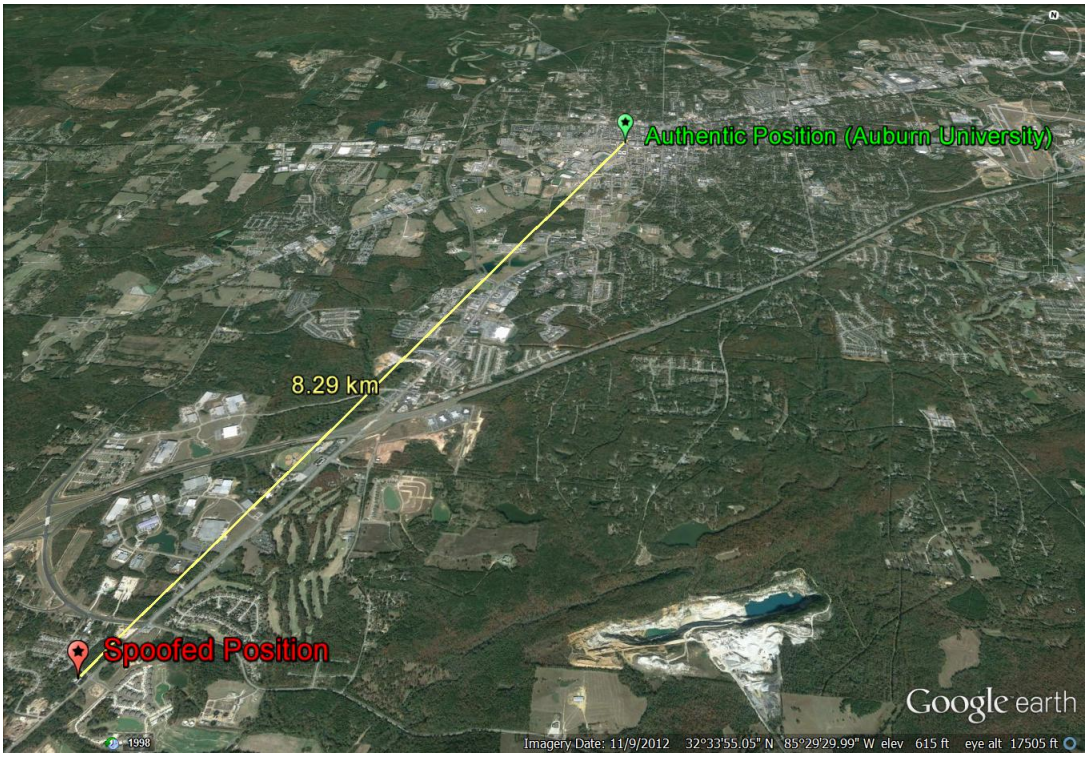


Figure 4.15: Map of positions used to create spoofing scenario near Auburn, Alabama.

The spoofed signal was given a power slightly higher than the authentic signal on every channel to ensure that the software receiver would lock onto the attacking signal without a detection and prevention scheme. Two IF data sets with positions near Auburn University in Auburn, AL were generated according to the map shown above. A sample rate of 5MHz and $int8$ format was used. The data sets were combined in software with a $30\mu s$ delay which is an approximation of the delay that would be induced by a spoofer combined with the transmit delay along the line of site between the two locations. The combination of the data sets resulted in acquisition planes like the one shown in Figure 4.16. Two peaks are clearly visible and distinguishable. The spoofing peak is noticeably more powerful than the authentic peak which causes the software receiver used in testing to acquire and track the false signals rather than the weaker authentic signals. For comparison, Figure 4.17 shows an expanded view of the code phase axis of the correlation plane.

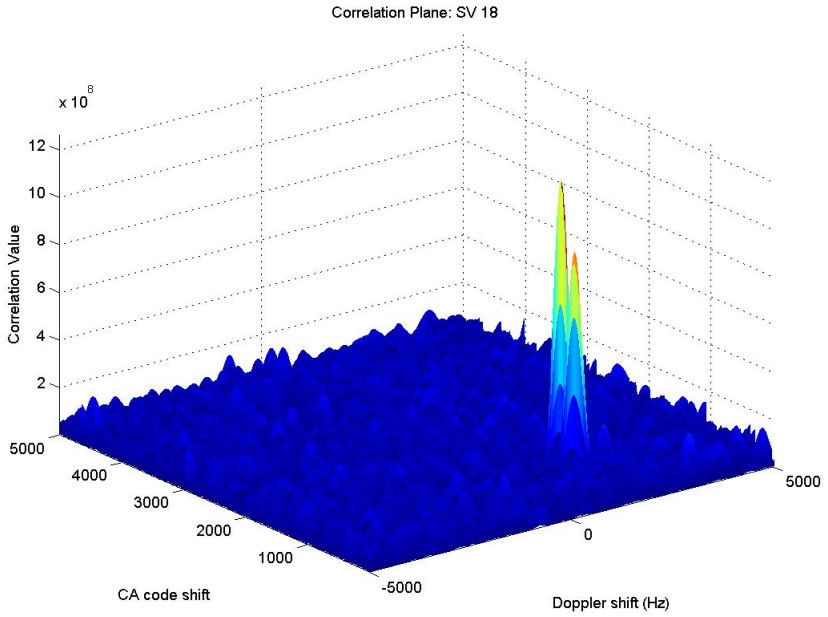


Figure 4.16: Representative correlation plane during a spoofing simulation.

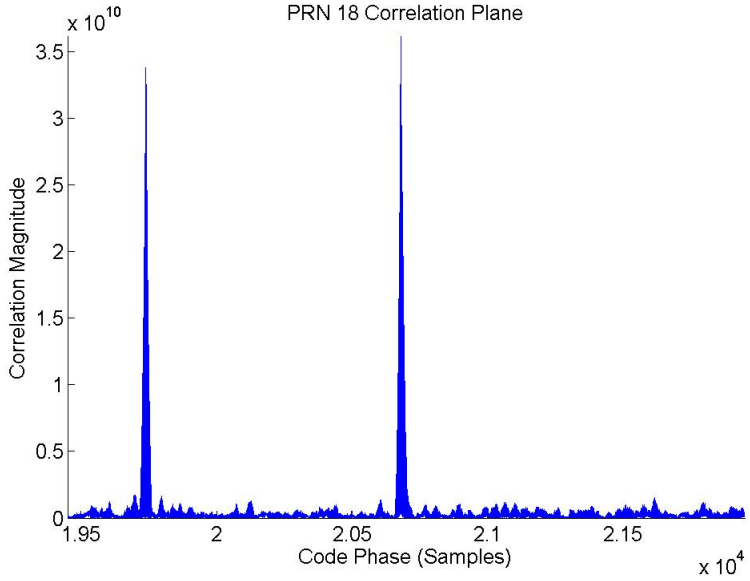


Figure 4.17: Representative correlation plane: expanded code phase axis.

Compared to the figures shown in the previous section describing Matlab simulated data, the large separation between the peaks caused by the $30\mu s$ delay can be clearly seen in Figure 4.16 and Figure 4.17. The spoofed peak (code phase 1000 samples) is clearly higher than the authentic peak 100 samples to the left. If acquisition is conducted at this point in the signal, the receiver will detect and track the spoofed signal. The integration period for this acquisition was set to $3ms$.

The histogram, frequency plot, and time domain plot in Figure 4.18 show that the simulated data parameters align well with those from a live-sky collection. The histogram shows a normal distribution across the full $int8$ spectrum with a noticeable peak at the center. The frequency domain plot shows a slight power peak around the intermediate frequency of 1.25MHz. The time domain plot reveals no abnormalities in the signal.

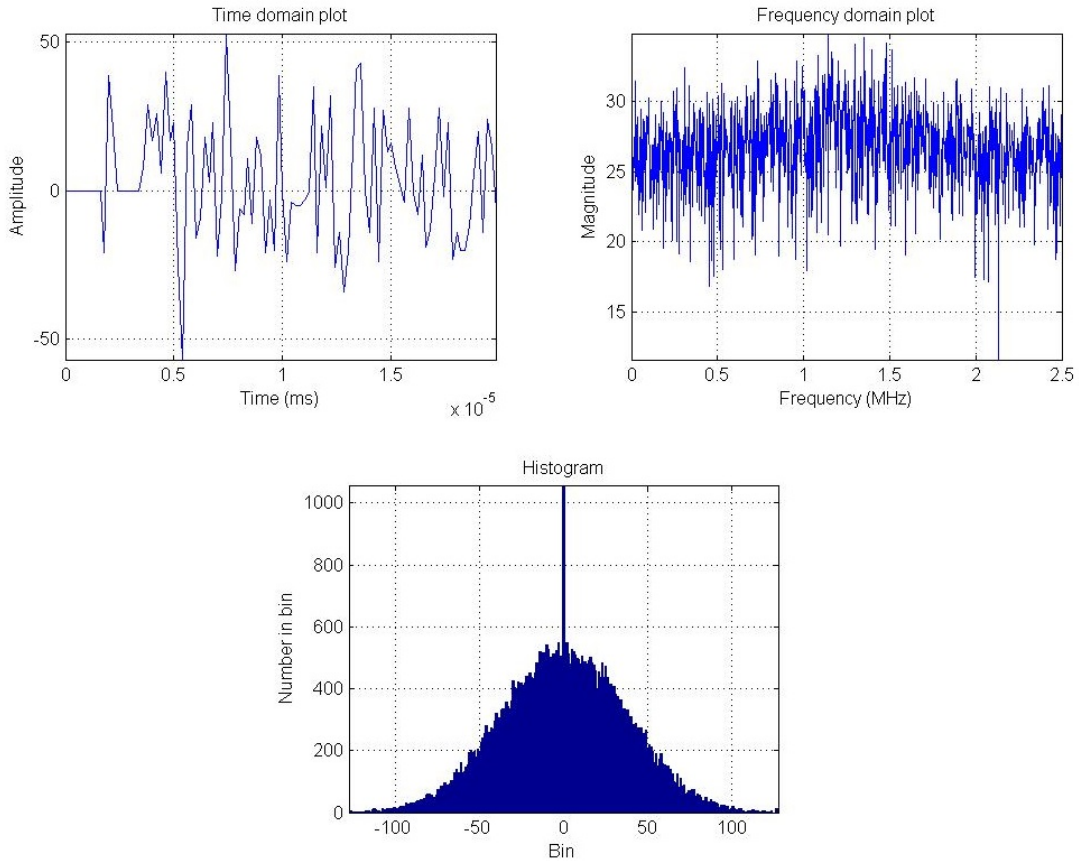


Figure 4.18: Histogram, time domain plot, and frequency domain plot of simulated spoofing data.

The software receiver developed in Matlab was run using the raw incoming data containing both the spoofed and authentic signals. The post process receiver locked on to the spoofing signals as expected and computed the position solution shown in Figure 4.19.

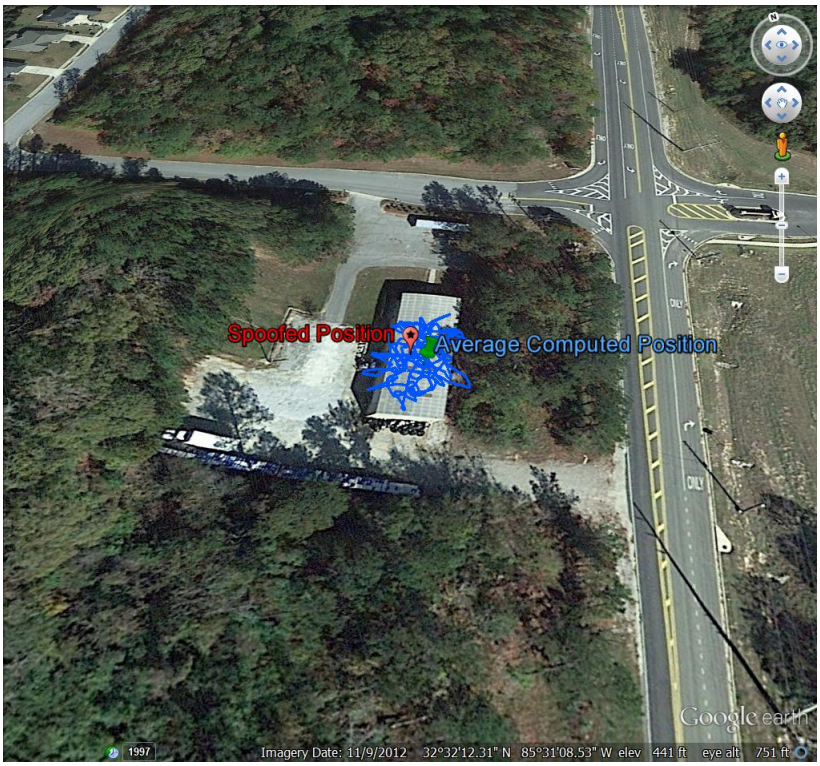


Figure 4.19: Spoofed position solution computed by software receiver.

Based on the position solution, the authentic signals were completely ignored on every channel and acted only as noise in the receiver. If the position solution was a combination of authentic and spoofed signals, the computed path would be a drifting trajectory far from both the authentic and spoofed positions. Since this is clearly not occuring, the receiver has been completely captured on every channel. Successive interference cancellation was then activated with the following effects. The results of wipeoff are clearly visible in the acquisition planes. The secondary spoofing peak is completely suppressed, disappearing into the noise floor. This is shown in Figure 4.20. Expanding again along the code phase axis in Figure 4.21, the spoofed peak is clearly driven below the noise floor. No residual effect of the spoofing or cancellation is left in the correlation plane.

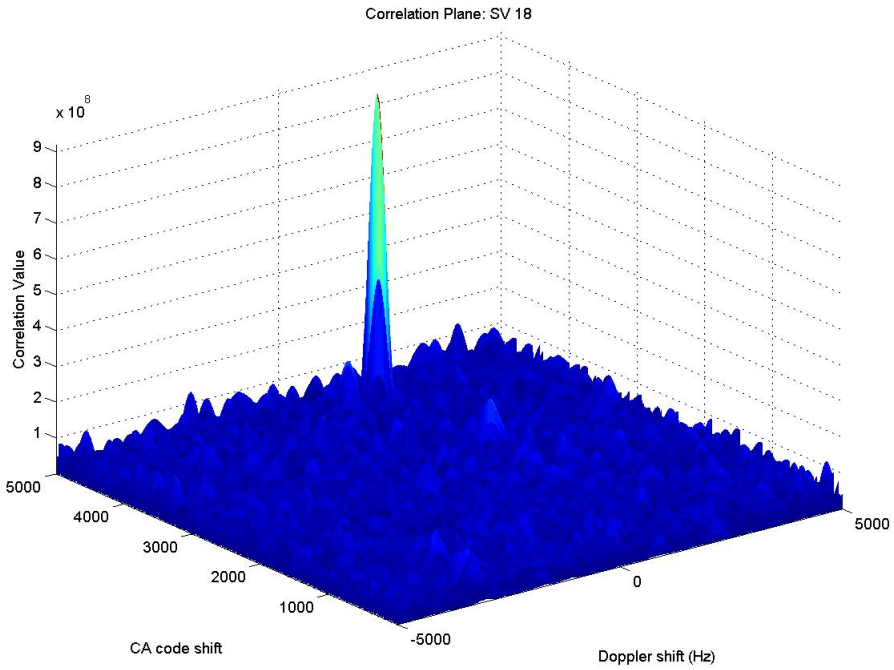


Figure 4.20: Acquisition plane after spoofed signal is removed.

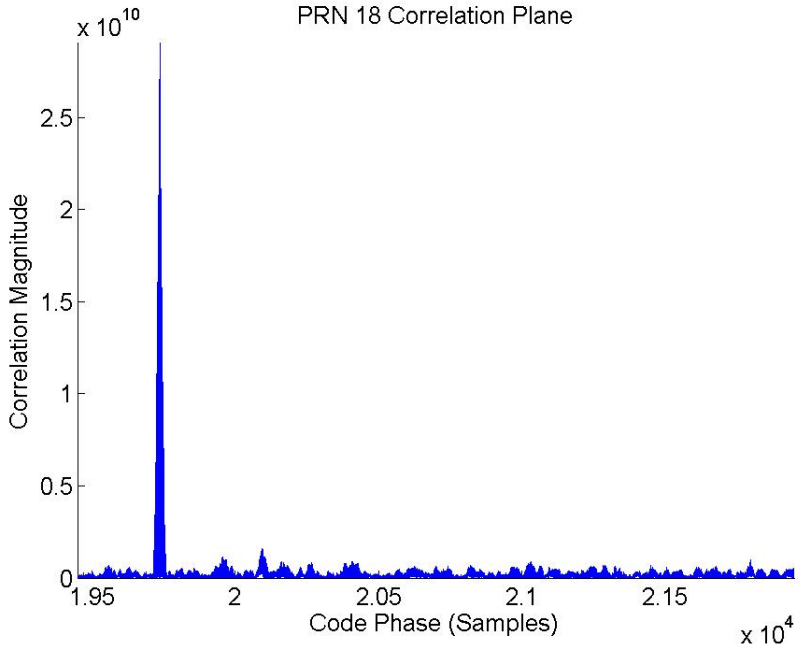


Figure 4.21: Correlation plane after wipe off - expanded code phase axis.

After removing spoofing, the cleaned signal was fed back into the software receiver. The histogram, frequency analysis, and time domain plot look the same as before. The receiver computes the authentic position solution shown in Figure 4.22.

Figure 4.22: Authentic position solution computed after cancellation of the interfering signal.

This scenario demonstrates the effectiveness of SIC against a simple attack. There is no apparent residual error introduced to the position solution after performing wipeoff and the average noise is actually decreased as the spoofed signals are removed. The software receiver easily acquires, tracks, and computes an authentic position solution using the true signals after SIC has suppressed the spoofed signals.

The second simulation tests the ability of SIC in a situation where the authentic and spoofed signals are more closely aligned in both space and time. The authentic position is a static location while the encroaching spoofing signal is a motion trajectory with the

authentic position as its start location. This represents, in some ways, an advanced spoofing attack in which the attacking signals are initialized exactly in line with the authentic signals and then slowly dragged off to some spoofed position. The map in Figure 4.23 shows the planned positions and trajectories for this simulated spoofing attack. For this scenario, after performing SIC the results are evaluated using both the software receiver as well as a commercial Ublox receiver. This demonstrates that the SIC algorithm is effectively removing the spoofed signal such that any receiver will be able to compute the correct position solution using a SIC corrected data file.

Figure 4.23: Map of trajectories used to create second spoofing scenario near Auburn Alabama.

As before, the data files were generated in the Spectracom simulators and recorded using N210 USRPs. The files were combined in software with a $2\mu s$ delay and stored as 8-bit integers with a sample frequency of 5MHz. This delay tests the theoretical limits of the SIC algorithm. Recall that delays significantly less than this will result in considerable beating of the signals and an inability to distinguish between the authentic and spoofed signals.

Acquisition and tracking was performed on the raw data first using the Matlab developed software receiver. The acquisition planes appeared as expected with dual peaks emerging,

however, due to the close alignment of the signals, the peaks are not visibly distinguishable in the full acquisition plane. An expanded view reveals two peaks, the spoofed peak having a greater magnitude than the authentic. The acquisition plane shown in Figure 4.24 is typical of the acquisitions performed on this data set.

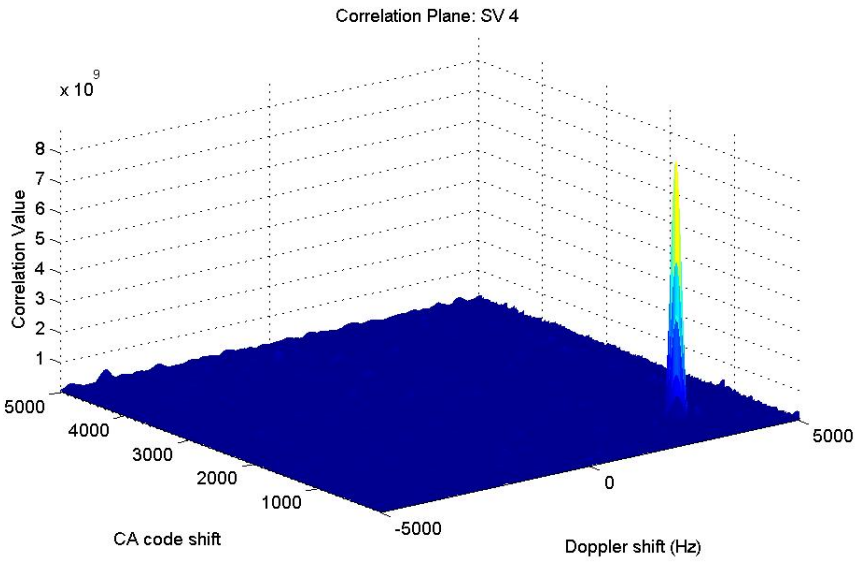


Figure 4.24: Acquisition peak with two closely aligned signals.

To demonstrate that two peaks are indeed present in the correlation plane, the data was plotted along the code phase axis and expanded. Figure 4.25 shows this zoomed plot revealing the dual peaks. The close proximity of the peaks distinguishes this type of attack from the previous tests.

In section 4.2.1 testing with randomly generated GPS-like data showed that SIC will work on perfect representations of closely aligned signals. This scenario tests whether SIC is effective on actual GPS signals which are extremely closely aligned in code phase as well as position. Running the software receiver on the first minute of raw data, the map shown in Figure 4.26 was generated. The first minute of data is enough to demonstrate that the receiver has been captured. The position solution, as expected, locked to the trajectory traced by the spoofed data set. The positions were computed as averages of the 100 ms position solutions to create a smoothed trajectory.

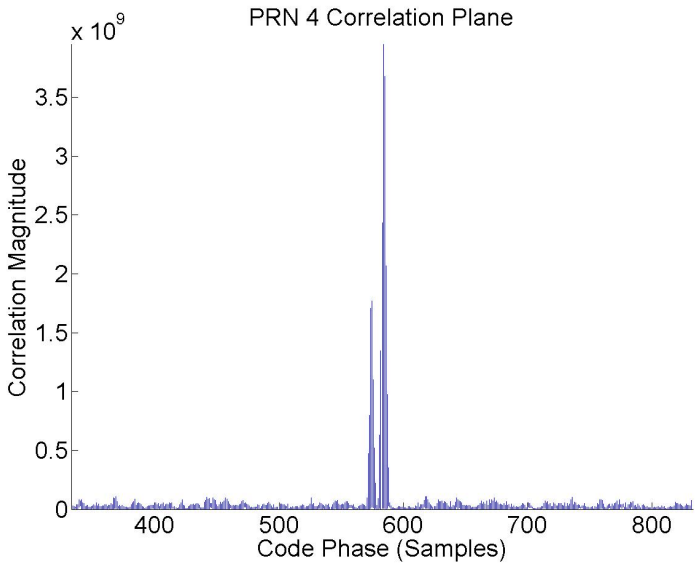


Figure 4.25: Expanded acquisition plane showing the code phase axis.

Figure 4.26: Computed trajectory for the second scenario with spoofing present.

Based on the position solution computed, the spoofing attack was successful in capturing the tracking loops of the receiver. At this point, SIC was performed on all the active channels to remove the encroaching signal. A cleaned signal was generated which contained only the remaining authentic signal. The cleaned data set was run through the software receiver to determine if SIC had effectively suppressed the attack. Figure 4.27 shows the acquisition plane after running SIC. Again, the full acquisition plane, while useful, does not have enough resolution to show how effectively SIC removed the spoofed signal. Figure 4.28 shows the same acquisition plane expanded along the code phase axis. The spoofed peak can be seen to be totally suppressed while the authentic signal remains.

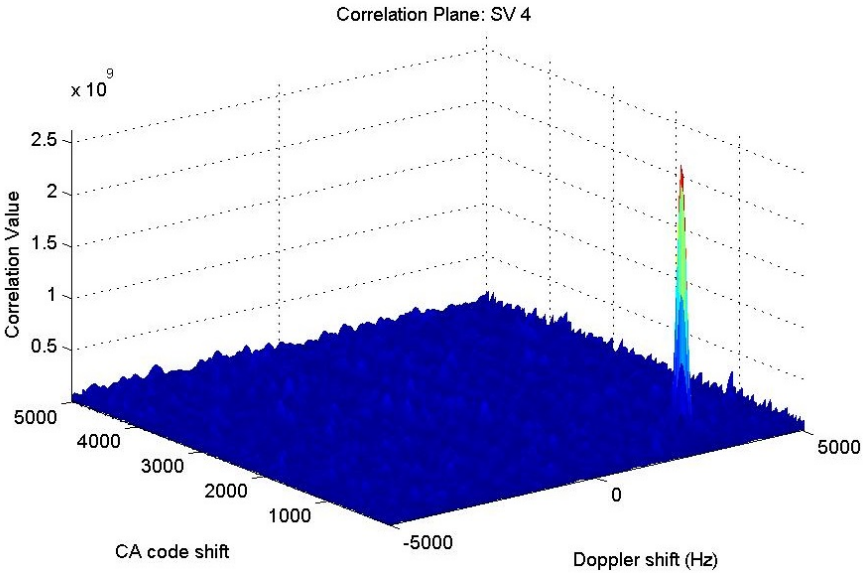


Figure 4.27: Acquisition plane for PRN 4 after SIC has been performed.

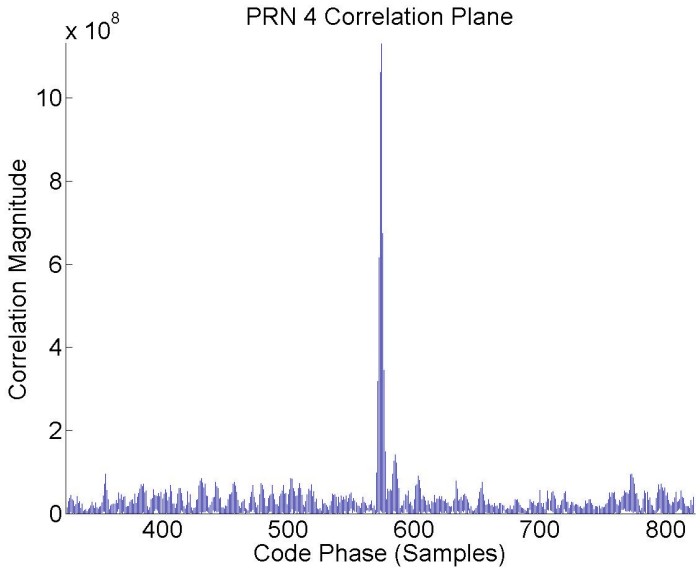


Figure 4.28: Expanded acquisition plane for PRN4 after SIC has been performed.

Next, a position solution was computed by the software receiver using the cleaned data (i.e. after SIC removal) to determine the cross-channel effectiveness of the suppression. The position solution snapped back to the authentic position at the intersection center as shown in Figure 4.29.

Figure 4.29: Map of position solution computed after SIC.

To verify the results seen in the software receiver, the cleaned data set was further processed and replayed through the Ettus USRPs and into a commercial Ublox receiver.

Processing was performed using Python and Matlab code developed by the author to convert the cleaned data sets into the format required by the USRPs for replay. The files were downconverted from the selected 1.25 Mhz intermediate frequency to 0 IF and filtered to remove the resulting high frequency mirror signal. The result was normalized to fill the 256 bins of the 8 bit integers. The resulting position solutions are shown in Figures 4.30 and 4.31 below. Figure 4.30 shows the position solution computed by running the raw data set through the Ublox receiver. A longer set of data was played into the Ublox compared to the segment processed in the software receiver resulting in several minutes of position solutions. As expected, the solution is defined by the higher powered spoofed signal demonstrating that the Ublox receiver was fully captured. After performing SIC, the cleaned and processed signal was played through the Ublox resulting in the position solution shown in Figure 4.31. The position has snapped back to the correct location at Toomer's Corner showing that SIC effectively removed the spoofed signal leaving the authentic signal intact.

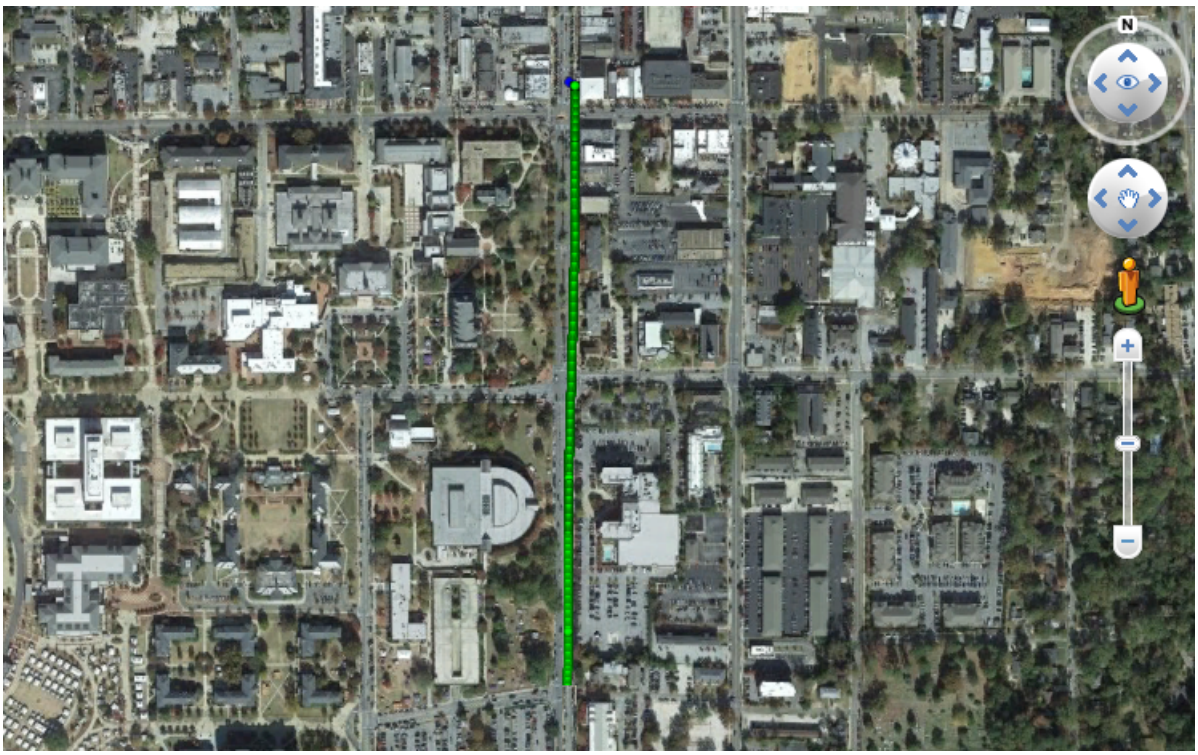


Figure 4.30: Position solution computed by Ublox receiver before applying SIC.

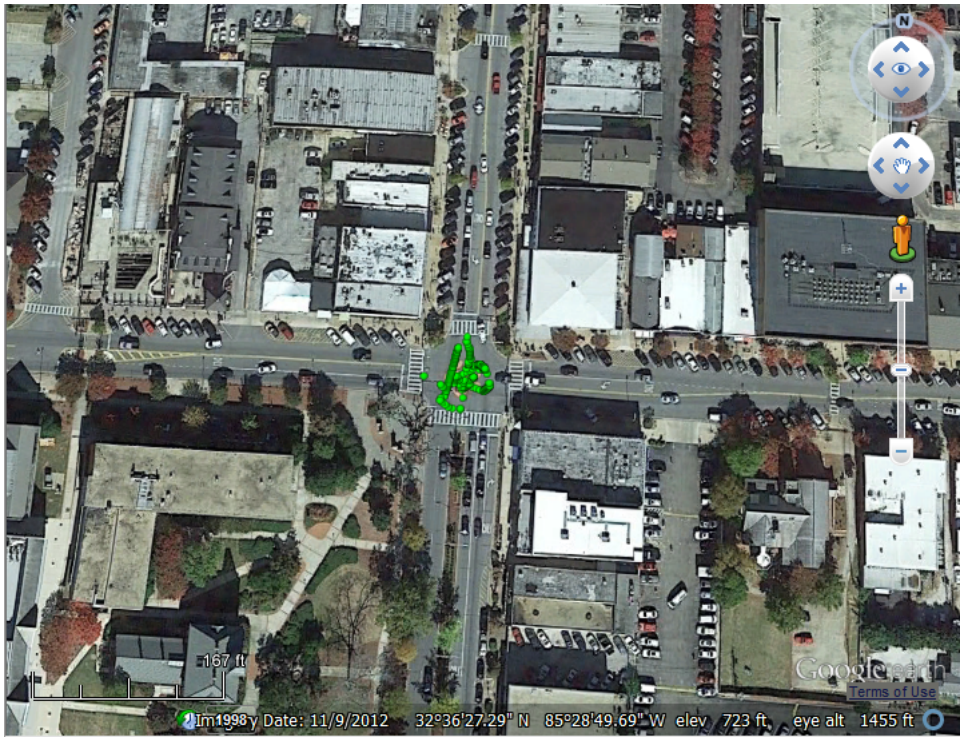


Figure 4.31: Position solution computed by Ublox receiver after applying SIC.

By replaying the signals through the Ublox receiver, it can be seen that all spoofed signals have been totally removed. In cases where all signals have not been removed, the position solution will be very unpredictable. The reasons for this are explained in Appendix B. This scenario demonstrated that SIC is very effective even against signals which are closely aligned. The $1 - 2\mu s$ limit in separation between signals is one which will very likely be met or exceeded in any type of attack. In a simple attack, it is not possible to induce delays less than this due to processing time as well geographical separation between the attacker and the target. In an advanced or synthetic type attack, the attacker must either forward propagate or delay the signal after capturing the correlators in order to prevent the effects of beating from throwing the tracking loops. As a result, any type of attack will result in a separation between the signals which allows the application of SIC to suppress the attack.

4.2.3 Summary of Results

The network spoofing detection scheme developed and evaluated in this work proved to be very successful in detecting an attack relatively quickly. Depending on the desired detection speed, the sample rate and detection threshold can be adjusted in a trade off between faster response or more accurate detection. The only situation in which the detection scheme has any likelihood of a missed detection is in a scenario where multiple advanced spoofers are present. This would allow the independent capture of the nodes and possible manipulation to maintain the RPV variations within the range expected by the algorithm. The possibility of this being accomplished is extremely remote due to its complexity and the difficulty of implementation. Adding an additional layer to the detection algorithm by including the radar heading in the threshold would prevent even the multi-spoofer attack from being missed by the detection routine.

In evaluating the effectiveness of SIC, simulated data representing a GPS signal with no noise was used to determine the limits of SIC application. Authentic GPS data sets generated by a Spectracom simulator were then used, showing that the SIC algorithm could successfully suppress a more powerful spoofing signal leaving the authentic signal intact for determination of the actual position. Using the Matlab simulated data, it was demonstrated that SIC is highly effective against signals separated by delays of greater than $2\mu s$. This number is a function of the C/A code interference patterns as the two signals begin to align. It was also demonstrated that SIC is effective against signals aligned with delays between $1 - 2\mu s$. In this case, environmental parameters and interference patterns between actual noisy GPS data will determine how effectively SIC can suppress the attacking signal. For signals delayed by less than $1\mu s$ SIC still works but will only be intermittently effective. This is due to the fact that in an actual attack, the precise alignment between the authentic and spoofed signals will vary with time. For a nominal delay of $1\mu s$, there will be points at which the actual delay is much less. Thus, while SIC may be effective in these cases, it will

depend entirely on the situation parameters such as noise, the code phase rate change, and relative power of the two signals.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Many sectors of industry rely on the GPS L1 civilian signal and unprotected commercial receivers which are susceptible to spoofing attacks. The trucking industry is increasingly incorporating GPS in convoying applications and individual vehicle monitoring. The financial sector relies on precise timing available through GPS to time transactions down to fractions of seconds. In the public sector, many every day users trust the navigation algorithms in hand held devices without a second thought. Even the most basic form of spoofing attack is capable of deceiving these unprotected GPS receivers. With the advent of more complex and advanced forms of spoofing, the danger is growing of an attack which leaves little or no indicators in the target receiver. in an advanced attack, by forming and broadcasting a synthetic replica of the signals which would be received at a target's location, the attacker has the ability to "lift" the receiver correlators off of the true signal and smoothly transition them to tracking the spoofed signal. All of this combines to pose a vulnerability to many off the shelf receivers.

In this thesis, techniques were explored and developed for the detection and mitigation of spoofed GPS signals. While detection of a basic spoofing attack is not trivial, it is also not extremely complex when the user has knowledge of what attack indicators to look for. The methods of detection discussed and developed in this thesis focused specifically on detection of advanced forms of attack in which there may be fewer obvious indications of an attack. The methods have application to simple attacks but were developed specifically to focus on an advanced attack. Detection based on phase differencing was looked at briefly and results from one test were shown. This method, while requiring additional hardware, offers

an easy opportunity to distinguish between authentic and spoofed signals on every channel since the detection is on a channel by channel basis. In a case where only a few channels are captured, this will be obvious in the correlations between the delta phases. The network detection scheme was thoroughly developed and explored as a robust detection method using networked GPS receivers. Such networks are common in many sectors of industry. This method offers the benefit of incorporating existing hardware and data already passed on the network reducing the need for additional hardware. In simulation testing, this method reliably detected and alerted the user to a spoofing attack in every situation tested.

Finally, having detected an attack, a method of successively suppressing the attack based on successive interference cancellation (SIC) in the intermediate frequency (IF) stage was developed for application to advanced attacks. This algorithm was explored using simulated data on a channel by channel basis as well as on an authentic scenario multi-channel simulation. In single channel testing, SIC was demonstrated to be effective even against signals that were extremely closely aligned in code phase. In the scenario simulation testing, SIC effectively wiped off the spoofing attacks producing cleaned data sets with only the authentic signals remaining. This cleaned IF data could be sent to a GPS receiver for computation of a corrected position solution.

5.2 Future Work

Although the methods of detection and suppression proved to be highly effective in the environments in which they were tested, there are several areas in which they can be improved or expanded. First, in the network detection routine, although detection is relatively rapid, the detection may still be 1-3 seconds after an attack has actually succeeded. In many cases, this may be sufficiently rapid. However, in cases where the network or individual nodes are moving rapidly, 1-3 seconds may not be fast enough. At highway speeds 1-3 seconds would translate to hundreds of feet in distance traveled. So the first item in future work is the improvement of detection time in the network detection scheme.

Second, in performing SIC, it is highly beneficial to have detailed and accurate knowledge of which channels have been captured and which are still tracking the authentic signals. Theoretically, in a successful attack, all channels will be tracking the spoofed signals. However, in an actual spoofing environment, the spoofer may not have a line of sight to all the same satellites visible to the target receiver. In this case, the target will likely be tracking several spoofed signals and one or two authentic signals. The position solution will diverge rapidly from both the authentic and the desired spoofed position. An indiscriminate application of SIC would remove both the authentic as well as the spoofed signals. In the testing performed in this thesis, the spoofed signals were distinguished based on timing delays as well as power levels. In an actual spoofing environment, a more robust classification scheme will be extremely beneficial. The phase detection scheme provides this sort of classification but requires the implementation of additional hardware. Another such scheme would involve the correlation of changes in the Doppler shift or pseudoranges across multiple channels as the receiver antenna moves. In a diverse sky environment, the changes across channels would be uncorrelated since the angle of arrival of each signal is different. In a spoofed situation, the changes in Doppler or pseudoranges would be correlated since all the signals are arriving from the same direction. These correlations would allow the classification of signals as either spoofed or authentic ensuring SIC acts on only the spoofed signals.

The final future work item is the combination of a detection, classification and suppression scheme into a single module. The proposed module would work inline with an existing commercial-off-the-shelf receiver (COTS). The module would take in raw RF data and down-convert it to IF for processing. Detection and classification of spoofed signals would be performed to determine if an attack has occurred. In the case of an attack, the module would alert the user to an attack and activate the SIC algorithm wiping off the spoofed signal in the IF stage. The cleaned signal would then be re-mixed to the RF GPS frequency and fed into the receiver RF port. Such a module would be able to work with any COTS receiver having an RF port and would incorporate both detection and suppression

of spoofing in a single unit. The detection routine used could vary based on the available hardware. For a completely stand-alone module, a multi-antenna detection routine could be incorporated into the module. For implementation into a network of GPS receivers, the module could take as additional inputs measurements from other receivers or devices and utilize the network detection scheme developed in this thesis.

Bibliography

[1] C. Gunther, "A survey of spoofing and counter-measures," *Navigation*, vol. 61, no. 3, pp. 159–177, 2014.

[2] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '05. New York, NY, USA: ACM, 2005, pp. 46–57.

[3] A. Brown, D. Reynolds, C. D. Robers, and M. S. Serie, "Jammer and interference location system - design and intial test results," in *Proceedings of the Institute of Navigation GPS '99*, 1999.

[4] G. Vinothkumar, G. Ramya, and A. Rengarajan, "Lightweight decentralized algorithm for localizing reactive jammers in wireless sensor network," in *Advanced Computing (ICoAC), 2012 Fourth International Conference on*, Dec 2012, pp. 1–5.

[5] D. Liu, J. Raymer, and A. Fox, "Efficient and timely jamming detection in wireless sensor networks," in *Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on*, Oct 2012, pp. 335–343.

[6] I. Kraemer, P. Dykta, R. Bauernfeind, and B. Eissfeller, "Android gps jammer localizer application based on c/n0 measurements and pedestrian dead reckoning," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, 2012, pp. 3154–3162.

[7] D. Shepard, J. Bhatti, and T. Humphreys, "Drone hack: Spoofing attack demonstration on a civilian unmanned aerial vehicle," *GNSS World, The Business and Technology of GNSS*, 2012.

[8] M. L. Psiaki, B. W. OHanlon, S. P. Powell, J. A. Bhatti, T. E. Humphreys, and A. Schofield, "Spoofing detection with two-antenna differential carrier phase," *GPS World, The Business and Technology of GNSS*, 2014.

[9] M. L. Psiaki, S. P. Powell, and B. W. OHanlon, "Gnss spoofing detection using high-frequency antenna motion and carrier-phase data," *Proceedings of ION GNSS 2013*, 2013.

[10] J. Cooper and P. Daly, "Preprocessing of gnss signals subject to interference," *International Journal of Satellite Communications*, vol. 15, pp. 247–257, 1997.

[11] A. Broumandan, A. Jafarnia-Jahromi, and G. Lachapelle, "Spoofing detection, classification and cancelation (sdcc) receiver architecture for a moving gnss receiver," *GPS Solutions*, vol. 19, no. 3, pp. 475–487, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10291-014-0407-3

[12] D. o. D. GPS NAVSTAR, "Global position system precise positioning service performance standard," Print, 2007.

[13] J. S. Subirana, J. J. Zornoza, and M. Hernndez-Pajares, *Global Navigation Satellite Systems: Volume I: Fundamentals and Algorithms*. European Space Agency Communications, 2013.

[14] What when how, tutorials and information structure of gps signals. [Online]. Available: http://what-when-how.com/space-science-and-technology/global-positioning-system-gps-2/

[15] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2011.

[16] Gps signal modulation scheme. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/thumb/3/3b/GPS_signal_modulation_scheme2.svg/220px-GPS_signal_modulation_scheme2.svg.png

[17] J. B.-Y. Tsue, *Fundamentals of Global Positioning System Receivers: A Software Approach*. John Wiley and Sons Inc., 2000.

[18] K. Borre, D. Akos, N. Bertlesen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver: A Single Frequency Approach*. Birkhauser, 2007.

[19] F. V. Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.

[20] What when how, tutorials and information code tracking replicas. [Online]. Available: http://what-when-how.com/a-software-defined-gps-and-galileo-receiver/carrier-and-code-tracking-gps-and-galileo-receiver-part-3/

[21] Crowtracker satellites in view. [Online]. Available: http://crowtracker.com/hdop-gpsposition-errors

[22] G. Kumar, G. Rao, and M. Kumar, "Gps signal short-term propagation characteristics modeling in urban areas for precise navigation applications," *Positioning*, vol. 4, no. 2, pp. 192–199, 2014.

[23] G. Gibbons, "Fcc fines operator of gps jammer that affected newark airport gbas," *Inside GNSS*, 2013.

[24] E. Steindl, W. Dunkel, A. Hornbostel, C. Hattich, and P. Remi, "The impact of interference cuased by gps repeaters on gnss receivers and services," *Proceeding of the European Navigation Conference*, 2013.

[25] Kulshreshta. (1997) Structure of gps signals. [Online]. Available: http://nptel.ac.in/courses/105104100/lectureB_6/images/5.gif

[26] M. L. Psiaki, B. W. OHanlon, S. P. Powell, J. A. Bhatti, T. E. Humphreys, and A. Schofield, "Spoofing detection with two-antenna differential carrier phase," *GPS World, The Business and Technology of GNSS*, 2014.

[27] M. L. Psiaki, S. P. Powell, and B. W. OHanlon, "Correlating carrier phase with rapid antenna motion," *GPS World, The Business and Technology of GNSS*, 2013.

[28] A. Broumandan, A. Jafarnia-Jahromi, V. Dehghanian, J. Nielsen, and G. Lachapelle, "Gnss spoofing detection in handheld receivers based on signal spatial correlation," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, April 2012, pp. 479–487.

[29] J. Ray, , S. Deshpande, R. Nayak, and M. Cannon, "Gnss radio: A system analysis and algorithm development research tool for pcs," *GPS World, The Business and Technology of GNSS*, 2006.

Appendices

Appendix A

Weak Signal Acquisition using Pre-Integration Data Wipeoff

A.1 Introduction

The GPS signal at the earth's surface is a very low powered signal. Although it is transmitted at a much higher power, it experiences decay caused by the extreme distance as well as scattering in the earth's atmosphere. By the time the signal reaches the earth, it has a power of about - 160 dBW. This is significantly lower than the thermal background radiation power. As a result, simple demodulation of the signal is not possible directly. Instead, as described in Section 2.1.2, the signal is modulated with a coarse acquisition (C/A) code which is a pseudo random signal unique to each satellite that correlates strongly only with itself. The navigation message is decoded in the receiver through a process of correlating to remove the C/A code and determine descriptive signal parameters. Under normal conditions, this process works well and allows rapid, unaided acquisition of the GPS signals. However, in scenarios where the signal power is significantly reduced relative to the the background noise, the correlation power of the C/A acquisition process is not be sufficient to pull the signal from the noise floor.

The relative strength of the GPS signals is often described in terms of "signal to noise ratio" or SNR which is a ratio of the signal power to the noise power. This is a helpful descriptor because both the received signal power and the noise power are variables dependent on environmental factors. It is only the relative strength of the signal compared to the noise that will determine if it can be acquired. When the SNR falls below a certain threshold (this is somewhat dependent on the acquisition hardware) the signal cannot be acquired or tracked. It is buried too far below the noise floor for correlation of the C/A code to be effective.

The SNR can be reduced by many variables. The following are several factors which act on the SNR by weakening the the GPS signal (lowering the S value). Weather patterns, while not independently able to lower the SNR enough to prevent acquisition, can weaken the GPS signal. Foliage can block or inhibit line-of-sight from the receiver to the satellite causing significant degradation of the signal. A pedestrian who walks under a thick tree canopy can observe this effect on any hand-held GPS device as the computed position solution will become more uncertain or even unknown. Buildings and other natural or man-made obstruction can weaken or block the GPS signals coming to the receiver. Indoor use of GPS is highly difficult because of this. Any GPS signals that are able to pass through a building are so significantly weakened that ordinary receivers are unable to acquire or track them. All of these factors reduce the SNR by lowering the signal power. It is also possible to reduce SNR by raising the noise floor. This can be accomplished accidentally as well as intentionally. The federal government has strict rules in place governing the broadcast of signals in the GPS band. As a result, interference in this band is kept to a minimum. However, there are many scenarios in which accidental transmission in the L1 band is possible. Many sectors of industry use equipment which could, if not used correctly, accidentally transmit in the L1 band. A malfunctioning transmitter operating in a band near the L1 frequency can easily "leak" L1 band noise. It is also possible that someone could maliciously transmit GPS frequency noise to intentionally raise the noise floor knocking out GPS capabilities in a particular area. All of these factors lower the SNR effectively burying the signal in noise preventing its acquisition or tracking.

The low signal power of the GPS is in many ways the greatest weakness of GPS. It prevents or impedes its use in many environments from urban canyons to forested areas to indoor settings. It also gives malefactors a better opportunity to degrade civilian GPS navigation technologies. As a result, methods of improving the weak signal performance of GPS are a subject of focused research. In this appendix, a methodology is introduced that

allows extension of the coherent integration period (CIP) across multiple navigation bits without external aiding.

A.2 Extended Integration

In Section 2.1.3, the process of acquisition or correlation is developed in detail. That information is necessary for the development of the extended integration method but will not be covered again completely here. Recall the typical acquisition architecture shown in Figure A.1. In acquisition, the incoming signal is processed in segments with a length referred to as the integration period. The integration period is always a factor of the C/A code length (i.e. 1,2,...,n ms in length). The incoming segment is split into in-phase and quadrature branches through a multiplication by reference carrier sine wave generated by a local oscillator. In parallel acquisition, a Fourier transform is performed and the result is multiplied by the conjugated Fourier transform of the C/A replica.

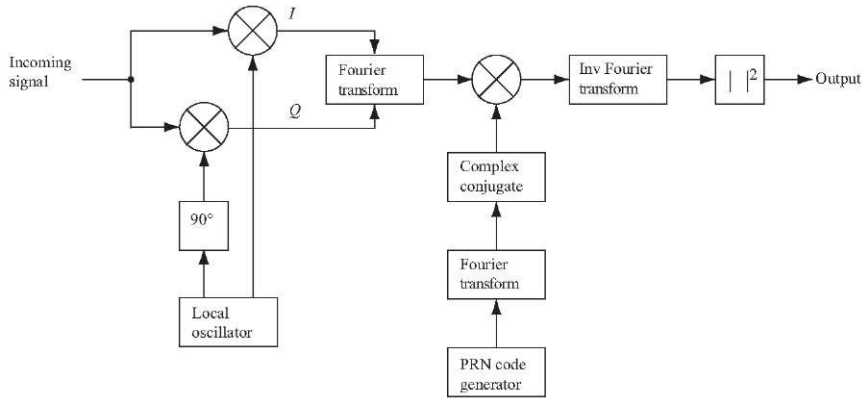
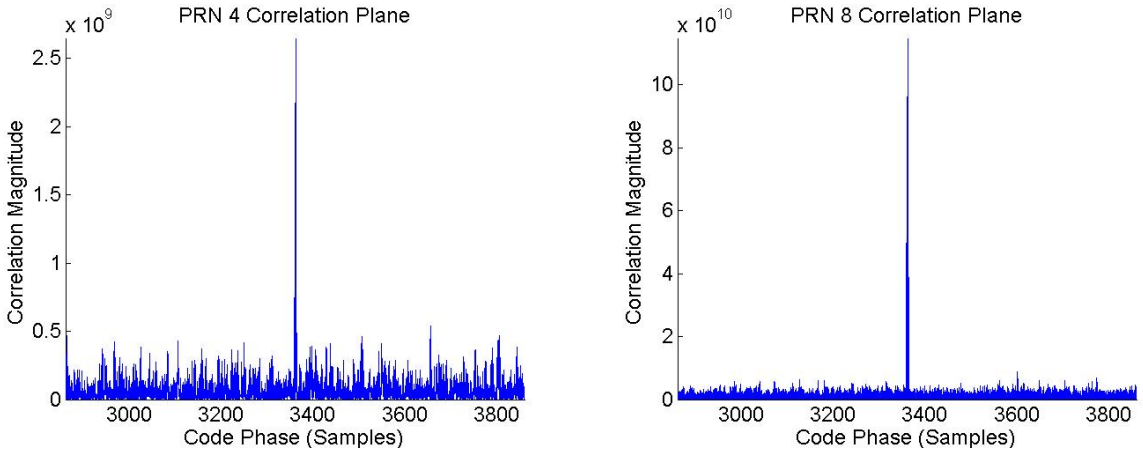


Figure A.1: Typical acquisition architecture in a GPS receiver [18].

The C/A replica must have the same length as the incoming signal. As the integration period is increased in increments of milliseconds, the replica C/A code is lengthened by the same amount. By increasing the integration period, the magnitude of the correlation is increased. In a strong signal environment, a single millisecond integration period, or correlation of the signal over one C/A code, will be sufficient to generate a strong peak indicating acquisition. As the signal power decreases, longer integration periods can be used

to increase the magnitude of the correlation peak. The noise is assumed to be white or Gaussian. As a result, by increasing the integration period, the peak from the correlation of the incoming C/A and replica C/A is increased while the noise peaks on average cancel reducing their magnitude. This is shown in Figure A.2 below. The left image shows the correlation plane in the code phase axis for a single millisecond integration period. On the right, the integration period is increased to 10 ms. The relative peak to floor distance is far greater in the 10 ms integration compared to the 1 ms integration.



(a) (b)

Figure A.2: Acquisition peaks in the code phase plane. In (a) the integration period is 1 ms. In (b) the integration period is 10 ms.

In a weak signal environment, increasing the integration period is one of the most powerful tools available. If the signal is weakened due to line-of-sight obstructions, it allows correlation across multiple C/A codes increasing the power of the correlation peak. On the other hand, if the signal strength is still good but the local background noise is raised due to interference, it allows the suppression of the background noise under the assumption that the noise is uncorrelated. Although extended integration is a powerful tool in weak signal acquisition, it has a few limitations. The extent that the integration period can be extended is severely limited by several factors. Primarily, the navigation data bits which modulate the carrier every 20 ms have the effect of inverting the C/A code any time the bit value changes. Thus, every 20 ms, there is the possibility of a data bit flip and in unaided acquisition, the

precise location and information on the bit value (1 or -1) is not externally available. As a result, the coherent integration period is theoretically limited to 20 ms. Practically, since the location of the transition is unknown, the CIP is limited 10 ms. This ensures that one of two successive correlations does not contain a bit transition. To increase the integration period beyond 10 ms effectively and consistently, the data bits must be accounted for. If a transition is included in the integration, the inverted C/A will be summed along with the "upright" C/A codes resulting in a decrease of correlation power. In an environment where the signal is already weak or obscured by noise, this will result in a non-detection.

Variation in the Doppler over time also affects the ability to increase the integration period. If the Doppler is not close enough to constant over the integration, then the change in Doppler will also have the effect of inverting some of the C/A codes. This is shown in Figure A.3 below.

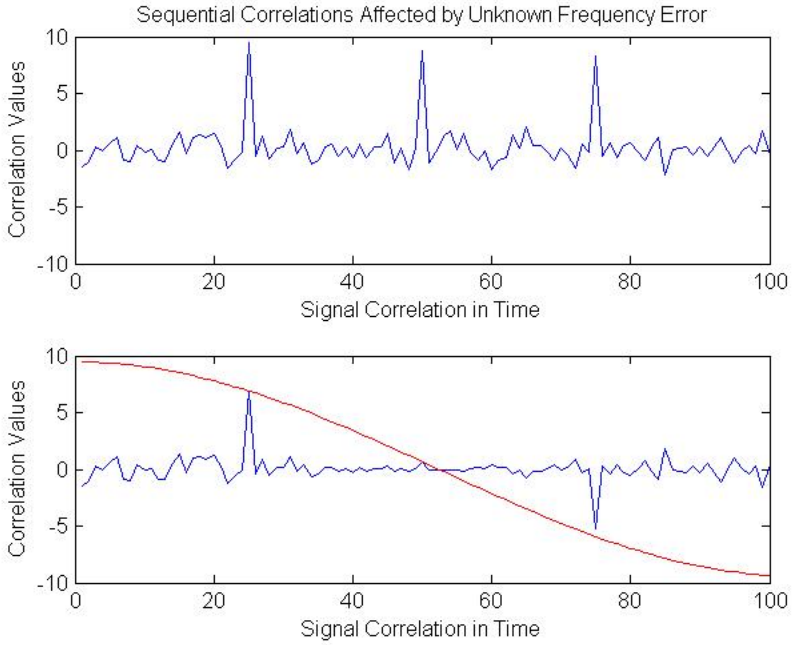


Figure A.3: Frequency variations (shown in red) due to changes in Doppler invert successive C/A codes.

The change in Doppler is related to the receiver velocity as shown in Figure A.4 below. By modeling the receiver velocity, much of the Doppler dynamics can be accounted for. This

greatly reduces the error introduced by frequency variations. For the purposes of this thesis, the receiver is assumed to be stationary eliminating the need to model receiver dynamics. As seen in Figure A.4, it is still possible to integrate up to 150 ms for receiver that is moving at a walking pace. However, the aim of the present work is to address the limitations caused by data bit transitions and assuming a stationary receiver eliminates additional variables that need to be accounted for. If this work is to be applied to rapidly moving receivers, the inclusion of an inertial measurement unit (IMU) or other external velocity sensor would allow modeling of the receiver velocity and the corresponding Doppler variations.

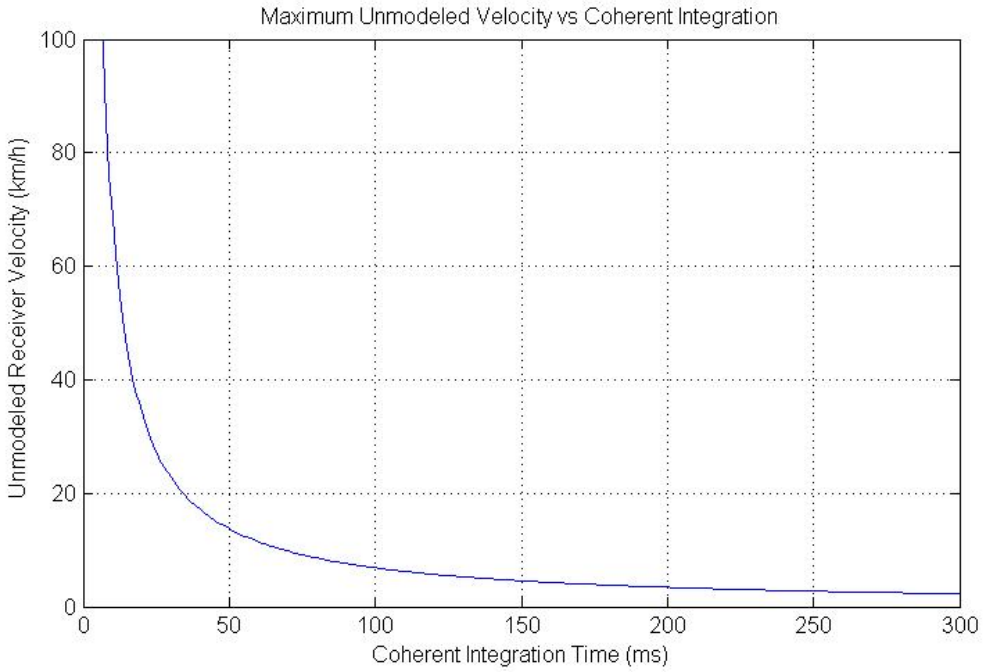


Figure A.4: Unmodeled receiver velocity vs. the permissible coherent integration time.

There are additional variables that must be considered in the implementation of an extended integration scheme. The two primary restrictions are the data bit transitions and the change in Doppler over time. The type of noise, whether or not it is correlated, also has an impact on the effectiveness of the scheme. Correlated noise will not experience the same cancellation as white noise. Under the assumption that the receiver is stationary and the background noise is uncorrelated, the major obstruction to increased coherent integration periods is the unknown data bit transitions.

A.3 Data Bit Guessing Scheme

Several methods have been proposed to address integration over bit transitions. One method often employed is non-coherent integration. Small coherent integration intervals (usually 1-5ms) are accumulated over a longer non-coherent interval. Summing the intervals raises the correlation peak, however, the noise power average also increases due to an effect known as squaring loss. Holding the coherent window constant and doubling the non-coherent integration time results in a doubling of the correlation peak and noise floor with a $\sqrt{2}$ increase in the noise floor standard deviation. The resulting signal to noise ratio also increases by a factor of $\sqrt{2}$ or about 3 dB [29]. As the integration period is increased non-coherently, the squaring loss becomes a significant factor. Because of this, coherent integration presents a more effective option if the obstacles preventing its implementation can be overcome.

For an assisted GPS receiver, the incoming data bits can be somewhat known or easily predicted. This allows the user to multiply the incoming signal by the known data bits in a wipeoff process. In effect, this aligns all of the C/A codes allowing coherent integration over a significantly longer period. However, if the receiver is unassisted, the data bits are unknown and even the transition points are unknown. To overcome this, a data bit guessing scheme is proposed in which an unassisted receiver generates a series of random bit sequences in an attempt to replicate the actual bit sequence. The incoming signal is multiplied by each random bit sequence and a coherent integration is then performed over the entire interval. For integration periods between 20 - 100 ms, this does not add an impossible amount of processing load. For an integration period of 100 ms, there are a total of 5 possible bits (each with a chip width of 20 ms). This results in $2^5 = 32$ possible combinations of ones and negative ones. From an acquisition standpoint however, half of these combinations are redundant. Since a single inverted C/A code will result in the same correlation as an upright C/A code, in terms of correlation, a bit value of 1 is equivalent to -1. For example, in correlation terms, the multiplying by the sequence 1,1,-1,1,-1 is the same as multiplying by

the sequence -1,-1,1,-1,1. Thus, the total number of actual combinations which must be tested in a 100 ms window is $(2^5)/2 = 16$.

This combination allows integration over 100 ms with a single guess for each incoming bit. Since the transition point is unknown, the guess sequence is unlikely to align well with the actual sequence resulting in a loss of correlation power. This effect is shown in the rough graphic shown in Figure A.5 below. The top bit sequence is the actual incoming bit sequence while the two sequences below it are shifted guess sequences. On the right, the individual and total correlation values are represented with various colored bars. The green represents a positive correlation, red represents a negative correlation and blue represents the total correlation power achieved by integrating over the entire period. In the first row, an integration period of 10 ms is used resulting in an all positive correlation. In the second row, the guess sequence is misaligned by about 1/3 of a data bit width. This results in some positive correlation and some negative correlation as portions of the bit transitions are included in the extended integration. In the bottom pane of the graphic, the bit guess sequence is perfectly aligned resulting in all positive correlation and a high correlation magnitude.

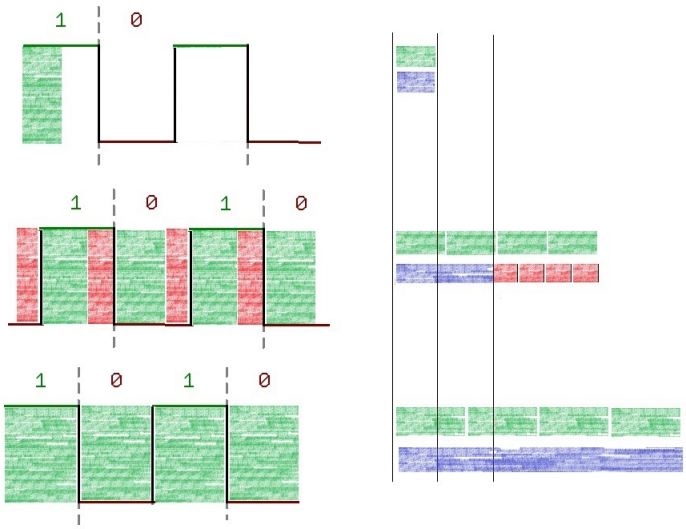


Figure A.5: Graphic showing the effect of misaligned guess sequences. The blue bars on the right represent the total correlation value achieved for each alignment with the actual sequence in the top row.

Since the bit transitions are unknown, it is unlikely that the bit guess transitions will be very closely aligned to the truth. If the bit guesses are centered across a bit transition, then the correlations will completely cancel out over the integration period negating the purpose of guessing the bits in the first place. Even with this difficulty, it is possible to use this sort of coarse guessing scheme. If no correlation is achieved in the first loop through the guess sequences, all the sequences can be shifted by 1/2 of a navigation data bit and the correlations performed again. In at least one of the sequential correlation loops, the guesses will be aligned closely enough to achieve an increase in the correlation magnitude. Another solution is to divide the guessed bits into segments. By splitting each guessed bit into two or more guesses, a more precise estimation can be generated. If there are two data bits in an incoming signal segment, then there will be a sequence of four guessed bits if the guess scheme is doubled. Each guess bit will span a segment approximately 10 ms long. This gives more precision in accounting for bit transitions and guarantees that the positive and negative correlations will never totally cancel out. In a worst case scenario, with two guessed bits for every authentic bit, at least every other guessed bit will result in a complete positive correlation. In doubling the number of guessed bits per segment, there are now a total of $(2^{(5\times2)})/2 = 512$ guess sequences. From a processing standpoint, this is not a feasible number of iterations to perform. When dealing with integration times of 60 ms, which corresponds to $(2^{(3\times2)})/2 = 32$ guesses, this seems more computationally reasonable. However, the first method of shifting the guess sequences is a more efficient approach. As a result, this was the method chosen and applied in the work for this thesis.

A.4 Testing and Results

To test the extended integration data bit guessing scheme, radio frequency (RF) data was generated using a Spectracom GNSS simulator. The output RF signal was fed into an Ettus Research USRP where it was downconverted and sampled. For processing, the data files were upconverted to an intermediate frequency (IF) of 1.25 MHz and sampled at a

frequency of 5 MHz. These data files were processed in Matlab on an HP EliteBook 8570w with an intel core i7 processor. The acquisition and extended integration algorithm was developed by the author and employs a parallel acquisition scheme.

A static simulation was programmed into the spectracom simulator with a programmed position solution of Toomers Corner in Auburn, AL. The data files were recorded at full signal power and noise was subsequently added to create weak signal data files. To evaluate the impact of increased integration time, a signal to noise ratio was computed based on the peak correlation value compared to the mean and standard deviation of the noise. The SNR equation is given here.

$$SNR = \frac{P_c - \bar{N}}{\sigma_N} \tag{A.1}$$

Where P_c is the peak correlation value, $\bar{N}$ is the mean of the noise, and σ_N is the standard deviation of the noise. To convert this into a more conventional decibel scale, the following equation is used.

$$SNR(dB) = 20 \times log_{10}(\frac{P_c - \bar{N}}{\sigma_N}) \tag{A.2}$$

The use of this statistic is helpful in direct comparison of integration times as it demonstrates both the increase in the peak power and the decrease in the noise. It should be noted that the bandwidth, receiver parameters, and processing or antenna gains are not taken into account in this metric. This is beneficial because the improvement achieved by just increasing the integration time can be more accurately evaluated. A better antenna, power injection, and filtering can all result in a more improved signal but a relative comparison of the improvement achievable due to increased integration time is the focus. As a result, SNR was chosen as a metric of comparison rather than C/No which incorporates other factors as well. To develop a baseline for reference, a recorded file was processed in the matlab acquisition and tracking loops with no noise added. The integration time was increased in increments of

10ms from 10-100ms without using the data bit guessing scheme. The integrations include the data bit transitions as well as any unaccounted changes in frequency or other errors. The results are shown in Figure A.6 below. The initial 10 ms integration produces a signal to noise ratio of about 65 dB. As the integration time is increased, this rises about 4.5 dB to just below 70 dB at a 40 ms integration. After this, the peaks taper off reaching about 59 dB at an integration period of 100 ms. This demonstrates the ineffectiveness of increasing the integration period beyond about 10-20 ms. In this case, the data bits and frequency errors worked out such that there was some slight improvement up to about 40 ms but none after that.

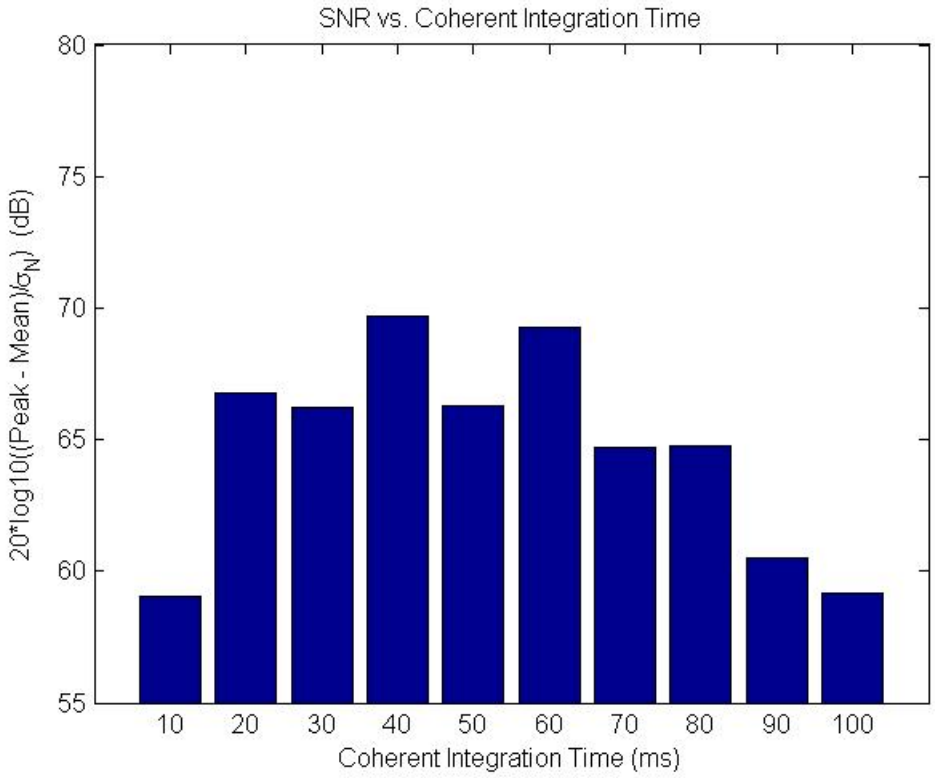


Figure A.6: Signal to noise ratio for increased integration times without accounting for data bit flips.

The tracking loop outputs in Figure A.7 below show the actual underlying data bit transitions that occurred as well as other tracking parameters. The I-arm and Q-arm power levels are shown in the right panes of the figure. The top left pane shows the I-arm outputs

revealing the data bits. The bottom left pane shows the Doppler frequency output by the tracking loops. From the I-arm where the data bits are visible, it can be seen that increasing the integration time incorporates several data bit transitions. These transitions are the main driving force behind the reduction in SNR visible in Figure A.6.

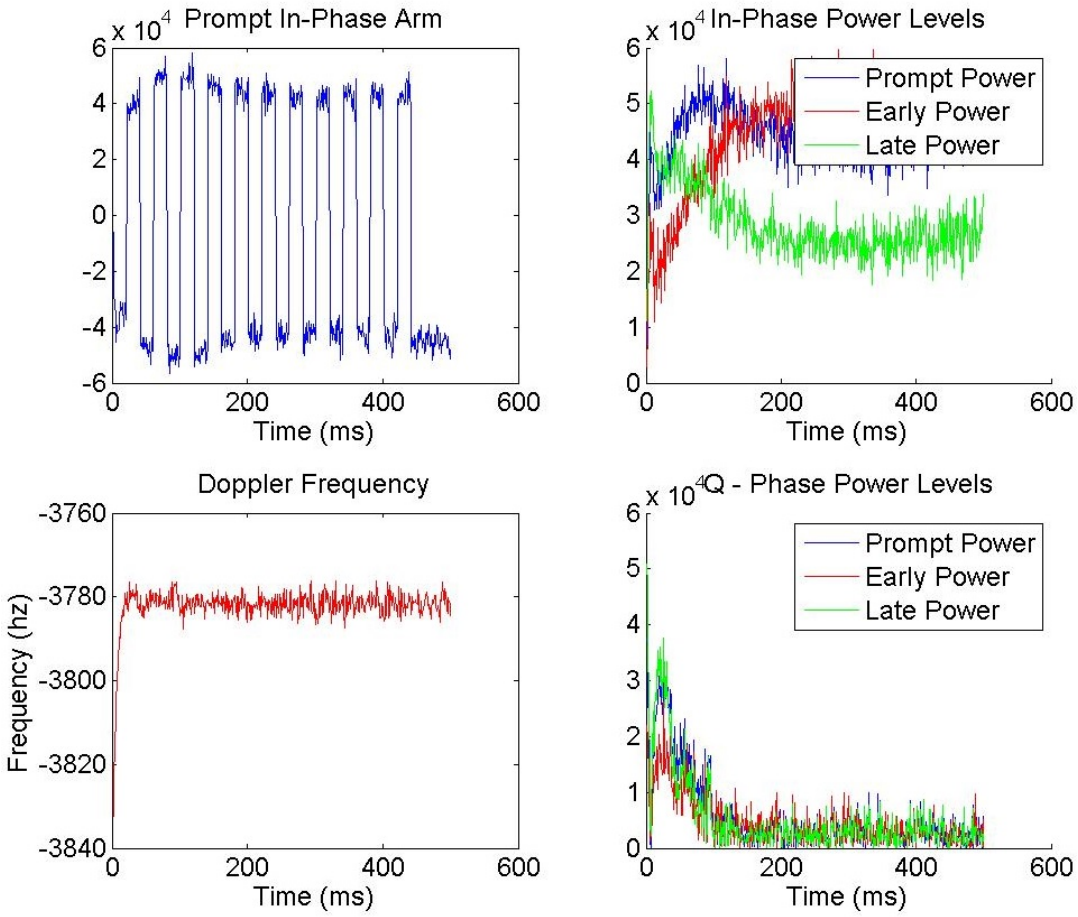


Figure A.7: Tracking loop outputs for the extended integration period.

For the same signal, the data bit guessing scheme was activated. For each integration time beyond 20 ms, sequences of data bit guesses were computed. The correlation values for each sequence were developed and the maximum chose from each integration time. These results are plotted in the bar chart in Figure A.8 below. The initial integration power is the same as before and remains approximately equal until the integration period is increased beyond about 40-50 ms. At this point, the data bit guessing scheme enables a significant increase in the correlation power compared to the noise floor. Rather than an overall decrease

in the correlation power as seen in Figure A.6, the increased integration time allowed an increase of over 15 dB. This is a significant improvement on signal strength demonstrating the effectiveness of data bit wipeoff using a guessing algorithm. Given that a 3 dB gain corresponds to an approximate doubling in the power ratio, this represents a nearly five fold increase in the power ratio.

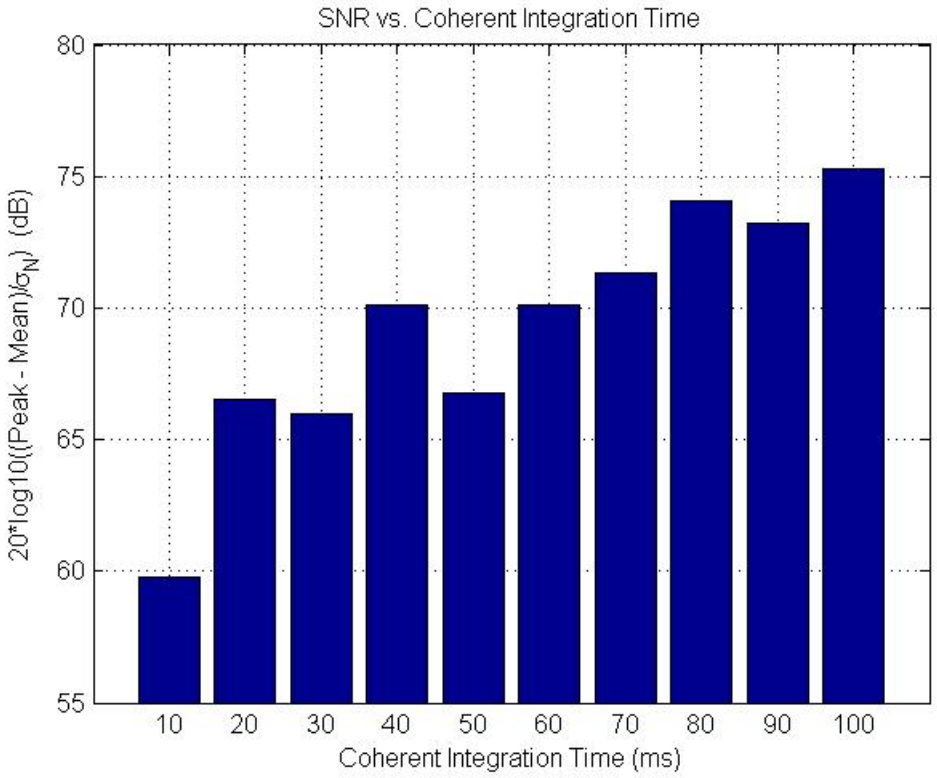


Figure A.8: Bar chart showing the correlation power achieved in increasing the integration time using a data bit guessing scheme.

Weak signal data files were then tested to determine the ability of the algorithm to pull a weak signal out of the noise floor for acquisition. The data file created contained a signal that was too weak to be detected with a 10 or even 20 ms integration period. The correlation peak for a 10 ms integration is shown in Figure A.9 below. The peak is clearly buried well below the noise floor and is not acquirable. Application of an increased integration period should have the effect of driving the noise floor down (since it is random uncorrelated noise)

and pulling the peak up. A post-extended integration correlation plane will reveal if this is true.

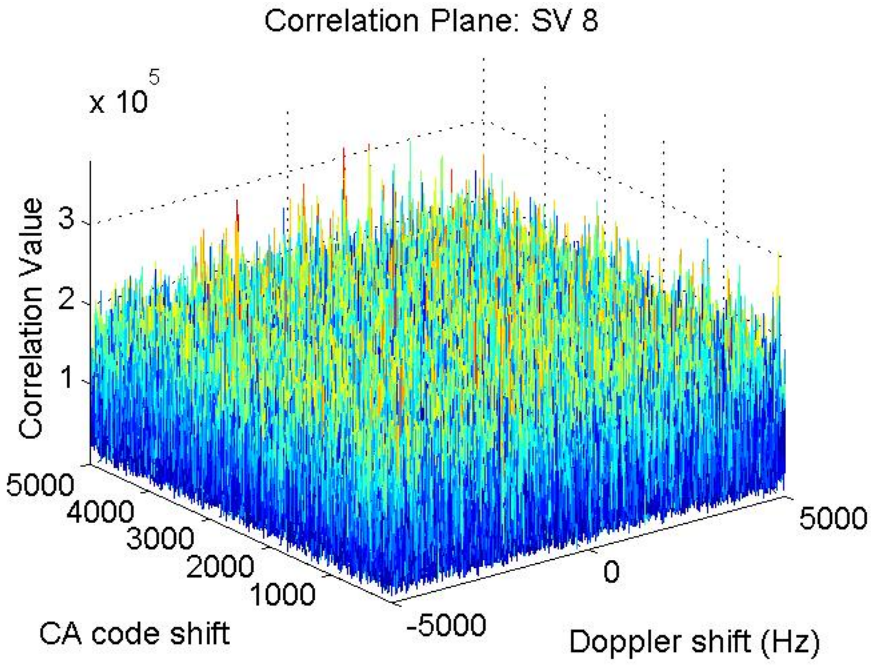


Figure A.9: Correlation plane for a 10 ms integration. The peak is buried well below the noise floor.

The data bit guessing algorithm was applied to the file with the following results. The bar chart in Figure A.10 on the next page shows the SNR for each successive integration period. It should be noted that the SNR is significantly lower for every integration period than the previous trials. The red line shows the threshold of acquisition. The signal is easily acquired in all integrations longer than 40 ms with the data bit guessing activated. The correlation plane also shown on the next page contains the acquisition results from the best guess sequence of the 100 ms integration period. The peak is clearly pulled above the noise floor and is easily acquired.

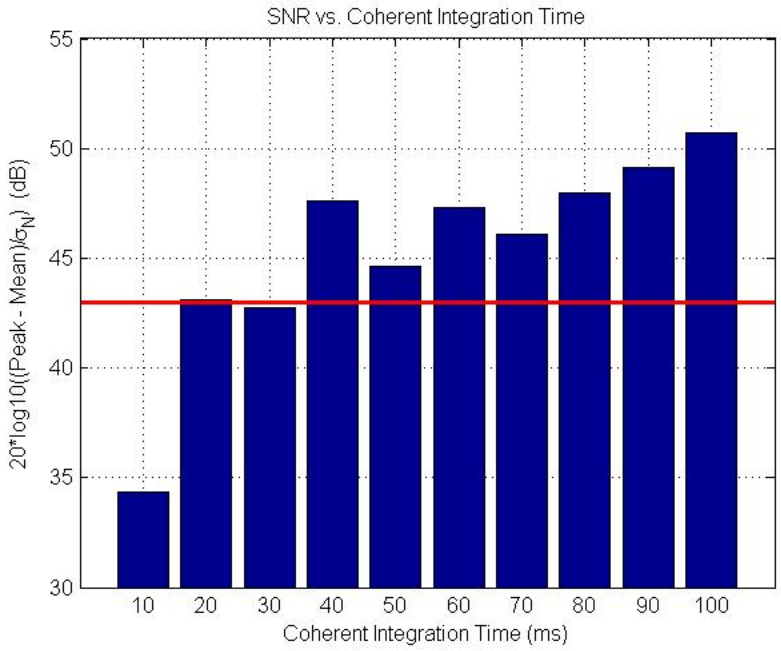


Figure A.10: Results of extended integration with data bit guessing on a weak signal.

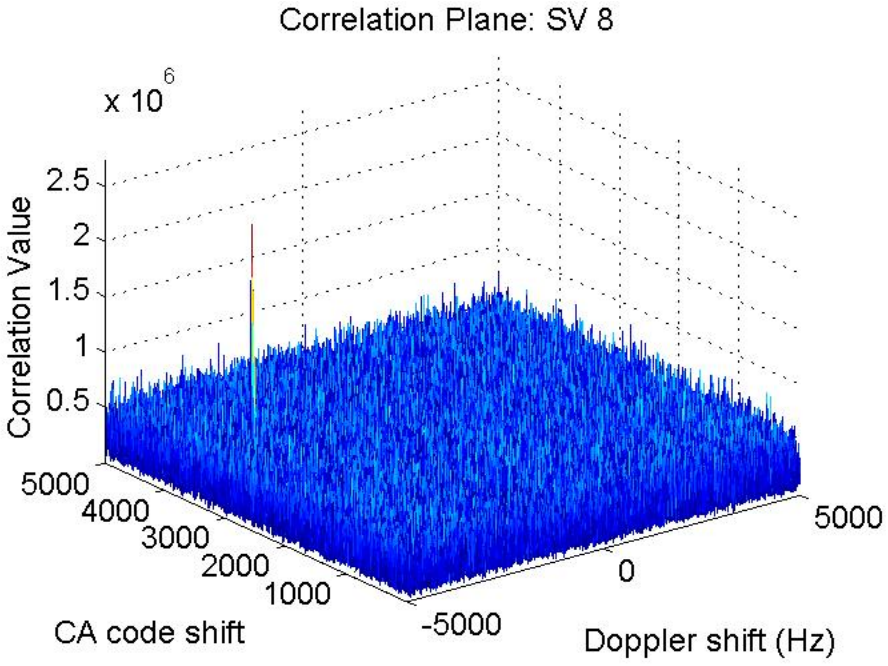


Figure A.11: Correlation plane from a 100 ms integration with data bit guessing to wipeoff navigation bits.

The acquisitions that produced Figures A.10 and A.11 were performed on the same underlying signal as that shown in Figure A.7. The data bits lined up well to allow extended integration to be effective. If the signal segment is shifted forward about a half of a chip width, the extended integrations become less effective since the guessed bits are bisected by the actual bit transitions. Figure A.12 shows the coherent integration times when the bit guesses do not line up well with the bit transitions. The scheme is still effective but the correlation power levels are reduced.

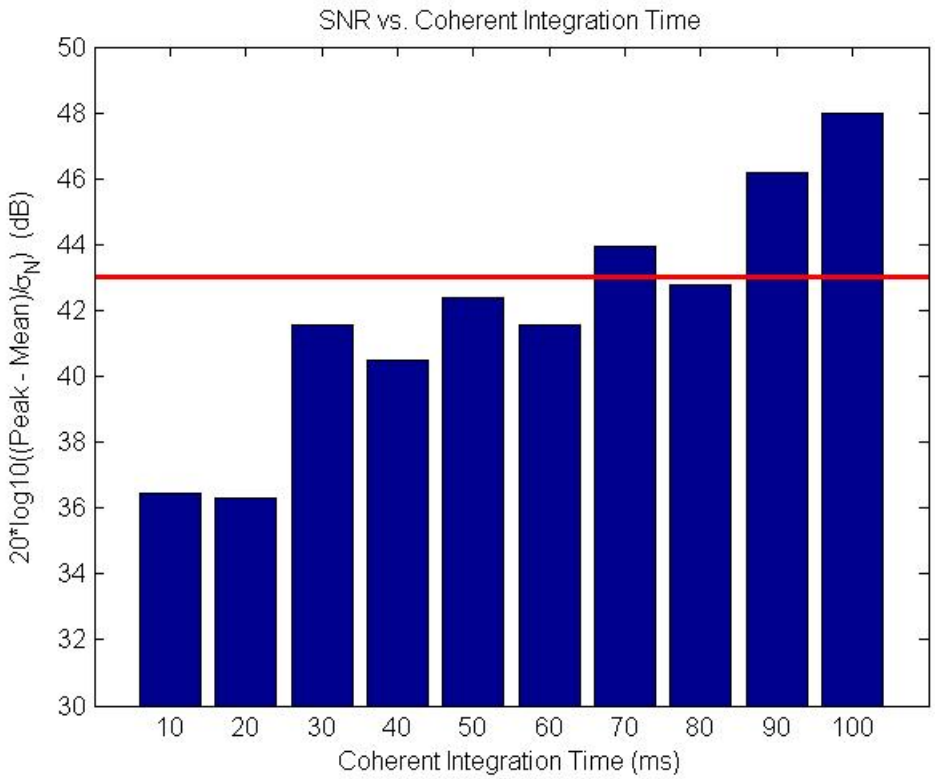


Figure A.12: Correlation SNR for various integration periods when the guessed bits are not well aligned.

A.5 Conclusions

In this appendix, an extended integration method was proposed in which an unassisted stand alone GPS receiver can attempt a cold acquisition using a completely random navigation data bit guessing algorithm to at least partially wipeoff the data message to allow

longer segments of coherent integration. Algorithm development was performed in Matlab and testing was conducted on IF data files recorded using Ettus USRPs. The results demonstrate that a guessing algorithm is effective for increasing the integration time. There are several possible applications for this sort of system. Civilian users who often find themselves traveling in heavily forested areas, densely populated urban areas with high rise buildings, or other weak signal environments would benefit from the implementation of such an extended integration method. Users who are out of cell phone service and unable to access assistance data would particularly benefit from an unassisted acquisition method. There are some challenges which must be overcome in order to fully implement the algorithm.

Primarily, the method is somewhat computationally intensive. On every channel on which acquisition is attempted, the number of acquisition correlations that must be performed is defined by the equation, $N = 2^{M-1}$ where N is the total number of correlations performed and M is the number of navigation data bits included in the integration segment. There are, however, ways of reducing the computational load. One obvious method is, rather than performing an acquisition on every possible combination of guessed bits, use the first guess that exceeds the acquisition threshold. If the total integration period is 100 ms, there are 16 possible bit combinations. However, if correlations are performed on each guess combination successively, there is a high probability that if an acquisition is possible, the peak will be found well before all guesses have been tried. This will statistically reduce the average computational load in half since a correlation peak will be found on average half way through the guess sequences. Another way to reduce the computational load is to use a partial guessing scheme. Rather than integrating coherently over the entire period, the coherent integrations could be limited to 40 or 60 ms. This reduces the iterations to 2 and 4 respectively. These coherent intervals could then be summed to create a much longer incoherent integration period. This introduces a squaring loss, however, the loss is significantly reduced due to the extended coherent integration times.

The testing and results summarized here demonstrate the viability of a blind guessing algorithm. There are several additional challenges which must be addressed to make this a more robust method. First, the frequency affects will need to be accounted for if the guessing scheme is to be applied to receivers with a significant velocity. Second, the computational load will require more processing power than a traditional receiver. This will need to be addressed with the designation of several correlators and additional processing units. In all, the guessing method produced good result and promises to be a viable solution to GPS navigation in weak signals environments.

Appendix B

Position Solution Error when Authentic and Spoofed Signals are Combined

In any spoofing scenario, there will be at least two signals present at the target receiver's antenna location. The aim of the prevention systems presented in the body of the thesis is to identify a spoofing attack and allow recovery of the authentic GPS signals. If this is properly accomplished, the receiver will be locked onto a single coherent set of signals. However, a spoofing environment is unpredictable and may impact the target receiver in a number of ways. One possible result of a spoofing attack is a position solution that is actually a combination of both spoofed and authentic signals. This is not necessarily a result that is desired by the attacker as the aim is to completely capture the target receiver with a coherent set of false signals. However, in the process of an attack, there are several scenarios which could result in a combined position solution. If the attack is not well tuned, power levels across channels could vary significantly. If the attacker has not properly adjusted power levels, the target receiver could maintain lock on some authentic signals while also acquiring several channels of the spoofed signal. It is also possible that a new satellite could rise during the attack. In this case, if the new channel is not immediately broadcast by the attacker, the previously captured target could lock onto the new authentic signal resulting in a combined position solution. Finally, it is also possible that the process of using successive interference cancellation could leave some spoofed signals present for the target receiver to track. If SIC is not properly coordinated or tuned, a spoofed channel may remain in the IF data causing a combined position solution. The purpose of this appendix is to briefly look at the position solution impact of including both authentic and spoofed signals in computations.

To determine the impact of including both authentic and spoofed signals in a position solution computation, it is important to know what stage of the receiver is affected by the

signal combination. The front end and IF processing portions of the receiver do not directly impact the position solution as it relates to a combined solution. Instead, it is in the computation of pseudoranges and the least squares position solution that the combination of authentic and spoofed signals has its impact. In most receivers, the pseudoranges used in determining a nominal distance to each satellite are computed on a relative basis. The earliest arriving signal is used as a reference from which pseudorange differences are computed. The time difference of arrival between signals is used to compute the relative difference in length between the pseudoranges. The pseudoranges are updated and combined with the ephemeris information on satellite positions to compute a position solution in a least squares algorithm. Because of this process, the combination of authentic and spoofed signals has a larger and more unpredictable impact on the final position solution than what might be expected. A spoofed signal will either be delayed or advanced in time relative to the authentic signal. This is necessary to avoid damaging interference as discussed in Section 3.1.2. Since the pseudoranges are computed based on the time of arrival referenced off the earliest arriving signal, the combinations of spoofed and authentic signals introduces two separate time references which are indistinguishable to the receiver. If the spoofed signal is delayed, the receiver will choose the earliest arriving authentic signal as the original reference signal. All of the signals, both authentic and spoofed which are used in computing the position solution will be referenced off this earliest arriving authentic signal. Since the spoofed signals have a delay due to actual travel time as well as processing and possibly programmed delay, the computed pseudorange will be significantly longer for any spoofed signal. As discussed in several places in the body of the thesis, the spoofed signal must be separated by a minimum of $1 - 2\mu s$ from the authentic signal. This translates to a minimum of 300-600 meters in additional length added to any spoofed pseudorange. In most cases, there will be significantly more delay particularly if the attack is not an advanced attack in which the spoofed signal is aligned with the authentic signal. In these cases, the delays could increase to kilometers or tens of kilometers. These delays are added to the computed pseudorange lengths which

are then incorporated to the least squares position solution computation. The least squares computes a best estimate for position based on the estimated ranges to each satellite as well as the timing bias. By including massive delays between spoofed and authentic signals, large and unpredictable position errors will appear in the solution.

Figures B.1 and B.2 below show three dimensional results of including three authentic signals and two spoofed signals. The sphere represents a to-scale globe and the blue vectors represent the computed ranges to each satellite as reported by the least squares best guess algorithm. The black X marks the computed position solution. Figure B.1 shows the position solution where the spoofed signals on both channels delayed by 2 ms relative to the authentic signals. The authentic signal corresponds to a position on the earth's surface about 1 km from the position corresponding to the spoofed signal.

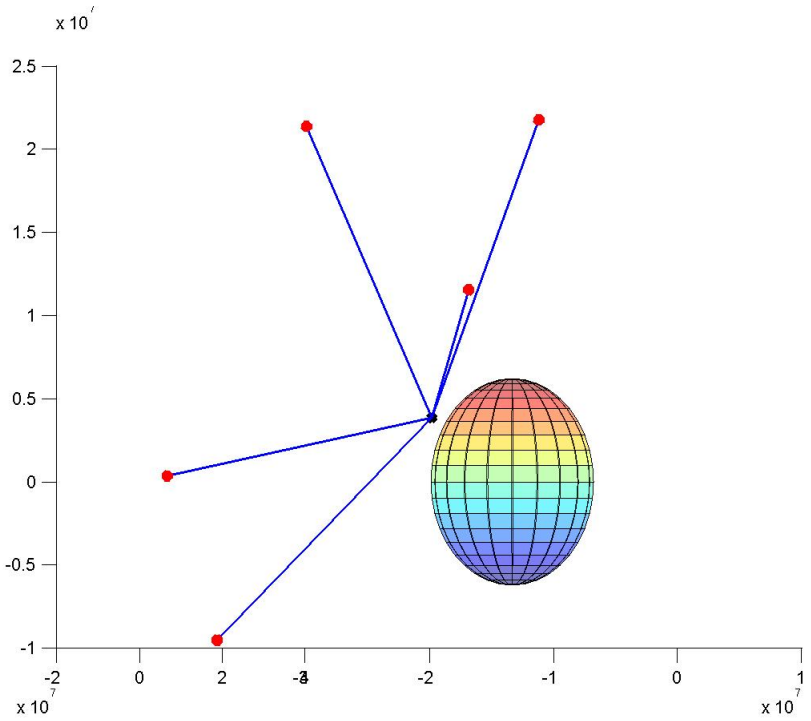


Figure B.1: Three dimensional plot demonstrating the effect of including both authentic and spoofed signals in the position solution. The spoofed signal is delayed by 2 ms.

Figure B.2 below shows the same scenario except the delay between the spoofed and authentic signal has been reversed. The authentic signal is now 2 ms behind the spoofed signal.

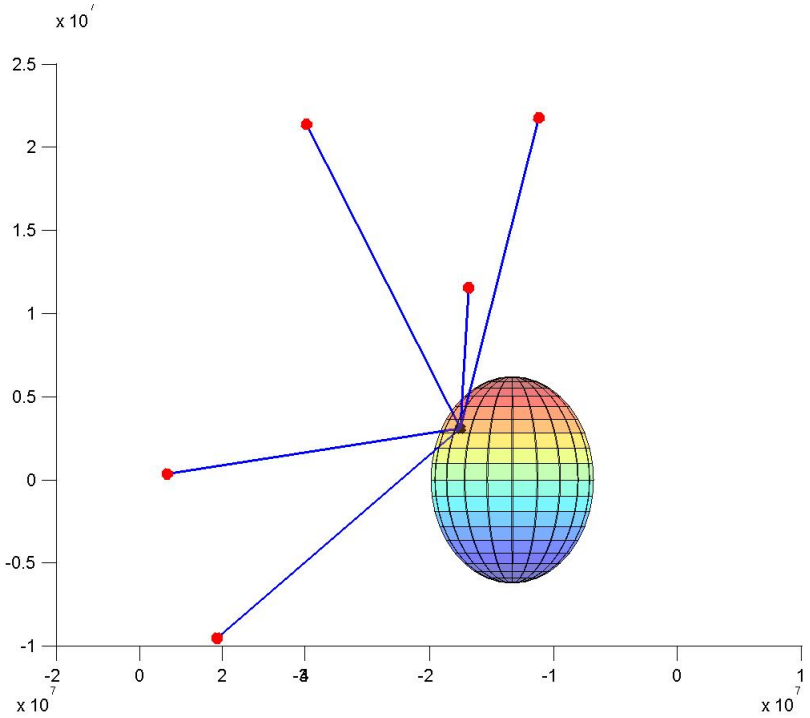


Figure B.2: Three dimensional plot showing the effect of changing the delay between the combined authentic and spoofed signals.

By looking at the results in a latitude-longitude plot, the impact on local horizontal error can be seen. The plot in Figure B.3 below includes the same two points shown in the above figures with the addition of a couple data points. Two points represent the authentic and spoofed locations included in the data sets. An additional point shows the result of combining the authentic and spoofed sets with no delay between them. The position solution, as expected, is somewhat between the authentic and spoofed positions. The error in this case is significantly less that that seen when there are large delays between the authentic and spoofed signal sets.

It may seem at first glance that the computed position when there is a delay between the signals should still lie somewhere between the authentic position and the spoofed position.

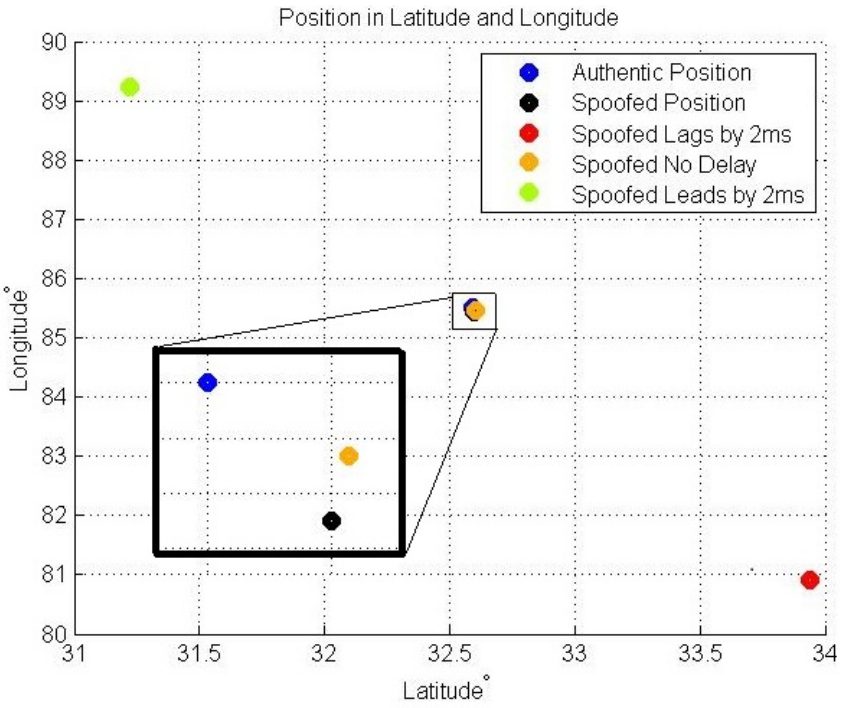


Figure B.3: Latitude and longitude plot showing error caused by combining authentic and spoofed signals.

Because of the least squares regression used in the position computation, this is not the actual result. Instead, the error is highly unpredictable and based on many factors. The time delay between the authentic and spoofed signals is the most significant factor as it results in dramatic changes to pseudorange computations. The number of included signals both spoofed and authentic also has a significant impact on the resultant position solution. The satellite geometry has an impact on the solution as well since the least squares solution relies on precise locations of satellites. If the timing and ranges are both off, significant errors are introduced. All of these factors impact the resultant position solution. The number of variables and their infinite range of possible values makes the impact of combining authentic and spoofed signals highly unpredictable.