A Multi-Antenna Vector Tracking Beamsteering GPS Receiver for Robust Positioning

by

Scott Burchfield

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
August 6, 2022

Keywords: GPS, CRPA, Vector Tracking, SDR, Beamforming

Approved by

Scott Martin, Chair, Assistant Professor of Mechanical Engineering
David Bevly, Bill and Lana McNair Professor of Mechanical Engineering
Stanley Reeves, Professor of Electrical and Computer Engineering

This thesis is dedicated to my late grandfather, Bro Burchfield, who convinced me to go to grad school when I just wanted to get a job. He inspired me everyday to work as hard as I can to achieve my goals. Without my grandfather's wisdom and inspiration, I would not be where I am today.

Men can wear you down physically, but only God can take away what you learn between your ears - Pawpaw Bro

Abstract

This thesis proposes the coupling of a Global Positioning System (GPS) L1 C/A vector tracking software-defined receiver with controlled reception pattern array array (CRPA) satellite constrained beamsteering, i.e. a multi-antenna vector tracking beamsteering receiver. As technology advances, the number of systems relying on the Global Navigation Satellite System (GNSS) and precise positioning and timing is increasing. Improvements in robustness and overall design are necessary for receivers to estimate position, velocity, and time accurately in all environments. This work is inspired by previous receiver designs by NAVSYS [1] and the German Aerospace Center (DLR) [2]. The proposed receiver conducts pre-correlator beamsteering. Pre-correlator beamsteering is lower SWaP-C as it reduces the number of receiver correlator channels compared to DLR's post-correlator implementation. The vector tracking receiver feeds back attitude corrected satellite geometry to a beamsteering module that updates the beam constraints as the extended Kalman filter (EKF) in vector tracking applies corrections. The proposed receiver is compared to a multi-antenna scalar tracking receiver in all testing scenarios. Comparisons are also made with a single-antenna vector tracking receiver, a single-antenna scalar tracking receiver, and a commercial-off-the-shelf (COTS) receiver. In simulation, receiver performance using a range from one to eight element CRPAs is compared to understand the benefits different arrays offer to receiver design. The simulations show there is a diminishing return in signal quality improvements as the number of elements increase. The simulations also show that a seven and eight element implemented vector tracking receiver can track signals at carrier-to-noise ratios (C/N_0s) as low as 12 dB-Hz, which is much lower than previously documented tracking thresholds [3]. The proposed receiver outperforms the other receivers when the simulation scenarios are noisy, improving PVT estimates considerably. Four element configurations of

the receivers are tested with dynamic live sky collected data. The multi-antenna vector tracking beamsteering receiver outperforms the other receiver designs, aligning closely with the COTS receiver. The performance of the multi-antenna vector tracking beamsteering receiver is followed by that of the single-antenna vector tracking receiver, the multi-antenna scalar tracking receiver, and the single-antenna scalar tracking receiver, respectively. The receivers are also tested in a jamming environment. The proposed receiver offers a four dB improvement in jamming resilience compared to the single-antenna vector tracking receiver.

Acknowledgments

I first want to thank God for strengthening my faith and offering peace throughout this thesis research. I also want to thank the community of believers I have at Lakeview Baptist Church for always brightening up my days and being good friends. Secondly, I want to thank the faculty of the GAVLab for putting in many hours to help me learn and grow in my education. Without the help of Dr. David Bevly, Dr. Scott Martin, and Dr. Howard Chen, I would not have been able to complete this thesis. Dr. David Bevly humbled me in his System Dynamics class, and he continually demanded the best out of me and other students in the GAVLab making us better students and better workers. Dr. Scott Martin was always available for questions, and he taught my favorite class in grad school, Optimal Estimation and Control. Both Dr. Bevly and Dr. Martin send it as professors like they send it on the ski slopes, doing everything 100 percent at 100 mph. Dr. Howard Chen was quick to offer thesis advice, and he was always up for a round of ping pong. I will never forget when we drove to New Mexico, and Dr. Chen and I bought matching cowboy boots. I also appreciate Dr. Stan Reeves for teaching me signal processing, which is the foundation for my current career.

The friendships and connections I made in the GAVLab were invaluable and irreplaceable. I'd like to thank Patrick Carter, Joshua Wood, and Sterling Thompson for their help and advice as I worked on this research. These guys were always around to bounce questions off of, and they were all some of the hardest workers I have been around. They were all superior to me in brains but inferior to me at Bullet Force. Also, Anderson Givhan was instrumental in recruiting me to the GAVLab. Anderson and I carried our friendship from painful nights in the Birmingham-Southern College physics department to working similar vector tracking research in the GAVLab. I appreciate Anderson helping me find direction

in my career by plugging me into a lab at Auburn University and continually being a great friend. Will Bryan and Connor Jones were responsible for the many injuries and embarrassing moments I endured in the GAVLab. They could always convince me to bomb it off a cliff on ski slopes that I had no business being on, and they always were there laughing when I failed 95 percent of the time. I published multiple things while I was in the GAVLab; of those things, the majority of them were videos of my crashes while snowboarding. I'd also like to thank Tanner Watts for the countless times I bounced ideas off of him in the lab. He always had the ability to think one step ahead of me, and he was truly an asset to have as a friend working with GPS.

From IS4S, I would like to thank Josh Starling and Josh Clanton for constantly investing time in me in grad school, advising me on next steps to take in my research. Both Joshes valued my research, and they offered IS4S resources continuously to help support me during my research. I am happy to now call them my co-workers, and it is awesome to have good friends like them in my workplace.

Outside of the lab, the constant support from my family is what kept me driven and motivated to complete grad school. My mom would always send me food, Bible verses, and words of encouragement. My dad always did a good job of getting my mind off of school to go deer hunting or fishing. My brother would always swoop in for an Auburn football game once or twice a year. One of my worst memories as a student at Auburn was Auburn football's collapse against Mississippi State in 2021. Sparingly, my brother did not give me too much grief after the loss. He is a senior at Mississippi State currently.

Last but not least, my soon to be wife Emily Bornstein has been there every step with me as I have worked on this research over the last year. She was always tolerant of me working long hours at IS4S and on my thesis, and she always brought me food when I was working late nights. She also helped me edit this document, and as a horticulturalist, I know reading this paper was quite enjoyable for her. Her pesky dog has also been instrumental to me finishing my thesis. I was able to use my high blood pressure derived from her dog's

incredible defiance to drive many late nights writing numerous pages. All jokes aside, the Lord has blessed me bringing Emily into my life. Without Emily's support over the last year, I would have not finished this research.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADC Analog-to-Digital Converter

AGC Adaptive Gain Control

AJ Anti-Jamming

ANC Adaptive Noise Canceling

AOA Angle of Arrival

AS Anti-Spoofing

AWGN Additive White Gaussian Noise

BPSK Binary Phase Shift Keying

C/N_0 Carrier-to-Noise Power Density Ratio

CDMA Code-Division Multiple Access

COTS Commercial Off-The-Shelf

CRPA Controlled Reception Pattern Array

CW Continuous Wave

dBm Decibel-Milliwatts

DF Direction Finding

DLL Delay Lock Loop

DLR German Aerospace Center

DOP Dilution of Precision

ECEF Earth Centered Earth Fixed

EKF Extended Kalman Filter

ENU East North Up Coordinate Frame

FFT Fast Fourier Transform

FIR Finite Impulse Response

FLL Frequency Lock Loop

FMET First Millisecond of Every Twenty Milliseconds

FPGA Field Programmable Gate Array

FRPA Fixed Reception Pattern Array

GNSS Global Navigation Satellite System

IIR Infinite Impulse Response

IMU Inertial Measurement Unit

J/S Jammer-to-Signal Power Ratio

LNA Low Noise Amplifier

LOS Line-of-Sight

NCO Numerically Controlled Oscillator

NLOS Non-Line-of-Sight

OCXO Oven Controlled Crystal Oscillator

PLL Phase Lock Loop

PPD Personal Privacy Device

PRN Pseudo Random Noise

PVT(O) Position, Velocity, Time, (Orientation)

RHCP Right Hand Circular Polarized

RTK Real-time Kinematic

sc16 Standard Complex int16

SDR Software Defined Receiver

STAP Space-Time Adaptive Processing

SWaP-C Size, Weight, Power, and Cost

TCXO Temperature Compensated Crystal Oscillator

TDOA Time Difference of Arrival

TOA Time of Arrival

TOW Time of Week

USRP Ettus Universal Software Radio Peripheral

VDFLL Vector Delay Frequency Lock Loop

WANF Wavelet-Implemented Adaptive Notch Filter

Chapter 1

Introduction

1.1 Background and Motivation

When the Global Positioning System (GPS) was declared operational in the late 1990s, no one expected the system to be as successful and widely used as it is now. GPS and other Global Navigation Satellite Systems (GNSS) have become integral in everyday civilian life, and the systems have become necessary for military operations. When GPS was declared a U.S. critical infrastructure in late 2004, GPS was granted more funding ensuring the systems would last [4]. Applications such as autonomous vehicles, navigation, banking, the stock market, aviation, farming, and more all rely on GPS or GNSS. Any system that requires precise position, velocity, or time (PVT) can use GPS and/or other constellations in GNSS. With level 3 autonomous vehicles being vigorously researched and readied for mass production [5], GPS usage is going to continue to rise.

The GPS system works well currently, but it does have weaknesses. The satellite signals that are received have to pass through the atmosphere of the earth making the signals very weak when they reach the surface. These signals are susceptible to natural interference (multipath and atmosphere) and man-made interference (jamming and spoofing). Urban environments typically have multipath when signals bounce and bend around buildings before reaching the receiver. Rural environments are more affected by foliage, degrading the signal quality. Atmospheric affects on the signals may also vary from time to time depending on atmospheric weather, but multipath normally is of greater concern. Jamming and spoofing have caused problems in more recent years. As technology advances, jammers and spoofers are becoming more advanced, and the costs of creating jammers and spoofers are decreasing [6], [7]. With many vulnerabilities, poor signal quality directly affects the performance of

GPS receivers. One solution to help receivers adapt to poor signal quality environments is beamforming.

The motivation of this thesis is to leverage beamforming and receiver design to improve a GPS receiver PNT solution, especially in challenging environments. Improvements in navigation robustness help GPS receivers withstand degradation making them more useful in more applications.

1.2 Prior Art

In the early years of GPS, receivers were mostly implemented in hardware. Most early hardware receivers implemented scalar tracking algorithms, and the implementations on the hardware were not very flexible. Now that technology has advanced and faster processing speeds are attainable, some receivers are programmed in software and uploaded to programmable hardware, typically field programmable gate arrays (FPGA). Receivers uploaded to FPGAs are known as software defined receivers (SDR) [8]. Developing software and uploading it to hardware makes receiver design adaptable, encouraging developers to test and optimize different receiver algorithms. SDR performance can be improved by leveraging signal phase in a process called beamforming.

Beamforming is not a new topic of research, and it has been around for over a century. The first implementation came in 1909 when Karl Ferdinand Braun and Guglielmo Marconi used 3 element beamforming to improve the transmit power for a wireless telegraph [9]; they were granted a Nobel Prize in Physics for their contributions. Now, beamforming algorithms are more advanced with applications in cellular systems, radar, GNSS, Bluetooth, and other wireless systems. Beamforming algorithms leverage controlled reception pattern arrays (CRPAs) and use the physics principles of constructive and destructive interference to change the way signals are received to be beneficial for the receiver. By phase shifting and scaling signals, beamforming algorithms can change a CRPA antenna perceived gain pattern, thereby opening the doors for both deterministic and adaptive beam algorithms [10].

Deterministic beamsteering is the simplest form of beamforming, and it is the beamforming algorithm that Braun and Marconi used to increase the transmit power of telegraphy. While other more sophisticated algorithms exist, this thesis explores the effects of implementing deterministic beamsteering to improve receive power.

1.2.1 Constellation Beamsteering

To improve GPS receive power, receivers must beamsteer to the GPS satellites. For this to be done, the satellite geometry must be fed back to the beamsteering algorithm, or the beamsteering algorithm must conduct direction finding (DF) to determine the satellite geometry. NAVSYS developed a High-gain Advanced GPS Receiver (HAGR) that employed beamsteering to improve received signal quality in early 2000. They implemented beamsteering for numerous applications, all of which revolved around the central HAGR idea [1], [11]–[13]. In 2011, a CRPA SDR implementation was created that invoked more advanced minimum variance distortionless response (MVDR) beamforming that beamsteered to orbiting satellites [14]. Both the HAGR and MVDR receivers implemented a scalar tracking architecture. Improvements in the scalar tracking architecture can yield better tracking performance and robustness.

1.2.2 Vector Tracking

Scalar tracking involves GNSS signals that are tracked independently. Positions are then computed by combining all of the separate correlator channel tracking solutions. Vector tracking improves signal tracking capabilities by coupling all of the tracking channels together. This allows stronger channels to aid weaker channels to give an overall better positioning solution. The first vector tracking research and implementation was conducted and patented by James Spilker at Stanford University in 1995 [15]. The research was expanded when Matthew Lashley and David Bevly of Auburn University outlined the benefits of vector tracking in multiple works. They showed that vector tracking survived signal outages and

low carrier to noise power density ratio (C/N_0) tracking better than scalar tracking. Vector tracking outperformed scalar tracking loops by 2-6 dB [3], [16], [17]. Sihao Zhoa and Dennis Akos at the University of Colorado Boulder also showed improvements using vector tracking with tests driving through Colorado canyons with intense multipath [18]. Don Benson at MITRE listed the benefits of using vector tracking receiver architecture [19].

1.2.3 Vector Tracking and Constellation Beamsteering

In 2012, the German Aerospace Center (DLR) created the first digital beamforming vector tracking receiver [2], [20], [21]. The DLR receiver implemented post-correlated beamforming, which means correlators must be created for each antenna stream. In post correlation, satellite signal geometry can be detected, so DLR did not have to feed back satellite geometry from the vector tracking loop. DLR's receiver included algorithms to removed interference and showed a combined vector tracking nulling receiver outperforming a standard vector tracking receiver in an interference scenario.

1.3 Contributions

Coupling vector tracking with multi-antenna beamforming has been proven to be effective in research by DLR. Beamsteering to orbiting satellites has also been proven to improve receiver operation by NAVSYS and Seo et al. While DLR proposed a post-correlation vector tracking beamforming receiver, there is minimal research pertaining to a pre-correlation vector tracking beamsteering receiver architecture, or loosely coupled beamsteering architecture. Several contributions to the field of GNSS signal tracking and beamforming are made in this thesis:

- Implement a multi-antenna vector tracking beamsteering receiver that beamsteers to GPS L1 C/A satellites

- Analyse the performance of the multi-antenna vector tracking beamsteering receiver against single antenna scalar and vector tracking, as well as multi-antenna scalar tracking beamsteering
- Compare the performance of the multi-antenna vector tracking beamsteering receiver and other receivers when the receivers are implemented with different sized COTS arrays ranging from 2-8 elements in simulation
- Compare the performance of the multi-antenna vector tracking beamsteering receiver and other receivers in multiple dynamic live sky scenarios
- Compare the performance of the multi-antenna vector tracking beamsteering receiver and other receivers in an artificially created interference environment created by combining dynamic live sky data and simulated interference
- Introduce low SWaP-C correlator based C/N_0 estimation techniques that are used in all receiver testing
- Show the benefits of the multi-antenna vector tracking beamsteering receiver compared to other receivers

1.4 Thesis Outline

This chapter introduced the concepts that inspired this work and discussed the multi-antenna vector tracking beamsteering receiver. The remainder of the thesis is laid out as follows: Chapter 2 introduces the GPS L1 C/A signal and gives an overview of signal vulnerabilities due to low signal power levels. The chapter also discusses CRPA antennas, including implications of poor calibrations when using CRPA antennas. Chapter 3 introduces a standard GPS L1 C/A software defined receiver. Front-ends, acquisition, tracking, and PVT estimation are all discussed. Chapter 4 outlines carrier-to-noise ratio estimation from a front-end approach to multiple correlator based approaches. Performance of the correlator

based algorithms is shown and compared. Chapter 5 introduces the asynchronous vector tracking receiver with detailed explanation into the asynchronous timing aspects. The maximum range variance is also discussed as a tool to be used in analysis. Chapter 6 introduces the proposed multi-antenna vector tracking beamsteering receiver. Chapter 6 also includes all of the analysis and results from testing the receiver. Chapter 7 includes conclusions that can be drawn and future work.

Chapter 2

GPS L1 C/A Signal

There are numerous global navigation satellite systems in orbit providing precise PNT to the user. The U.S. constellation is known as the Global Positioning System (GPS). Other countries have their own systems (i.e. Galileo (European Union), GLONASS (Russia), QZSS (Korea), and Beidou (China)). Low earth orbit (LEO) satellite systems are also starting to be used for positioning [22]. Some techniques used in this thesis are applicable to other constellations as well as LEO constellations. While many satellite systems are relevant for navigation, the GPS L1 C/A signal is the primary focus of this work. In this chapter, Section 2.1 describes the GPS L1 C/A signal including the carrier, code, data message, and signal modulation. Section 2.2 details the vulnerabilities of the GPS L1 C/A signal, which include jamming, spoofing, and multipath. Section 2.3 introduces common antennas used in capturing GPS signal data as well as an introduction to CRPAs.

2.1 Signal Overview

While there are multiple signals associated with GPS, the GPS L1 C/A signal is the most commonly used civilian signal from the constellation. The L1 C/A signal is comprised of two components: the carrier and code signal (Section 2.1.1). The code and carrier are modulated with the data message that holds information necessary for positioning (Section 2.1.2). The signal modulation is described Section 2.1.3.

2.1.1 Carrier and Code

The GPS L1 C/A carrier signal is a sinusoidal signal centered at the frequency 1575.42 MHz. Other GPS signals that occupy the L1 band are L1C, P(Y) code, and M-Code. The

GPS L1 C/A signal does not use a sub-carrier like L1C and M-Code. There are no plans currently to remove the L1 C/A signal, and nearly all civilian GPS receivers are configured for the L1 C/A signal.

The pseudo-random noise code, also known as the PRN code or spreading code, has a chipping rate of 1.023 Mbps. The signal is a square wave created using a Modulo-2 sum of two binary sequences, which is defined in [23]. The binary signal repeats every 1023 chips, transmitting a full PRN code every millisecond. The PRN code is unique for each satellite, and the signal is known to the receiver, hence, allowing the receiver to correlate the signals with their satellites, which is code-division multiple access (CDMA). The PRN sequences are Gold codes [24]. The signals autocorrelate with identical signals only when they are time aligned, and the cross correlation with the other codes is zero [8]. These characteristics are the driving factors in the noise variance estimation described in Chapter 4.

The ideal autocorrelation function is shown below in Equation (2.1).

$$R(\epsilon) = \begin{cases} 1 - |\epsilon| & \epsilon \leq 1 \\ 0 & \epsilon > 1 \end{cases} \tag{2.1}$$

Figure 2.1 shows the actual autocorrelation function plotted as a function of the chip spacing. When the chip spacing is greater than one, the autocorrelation function is approximately zero as described earlier.

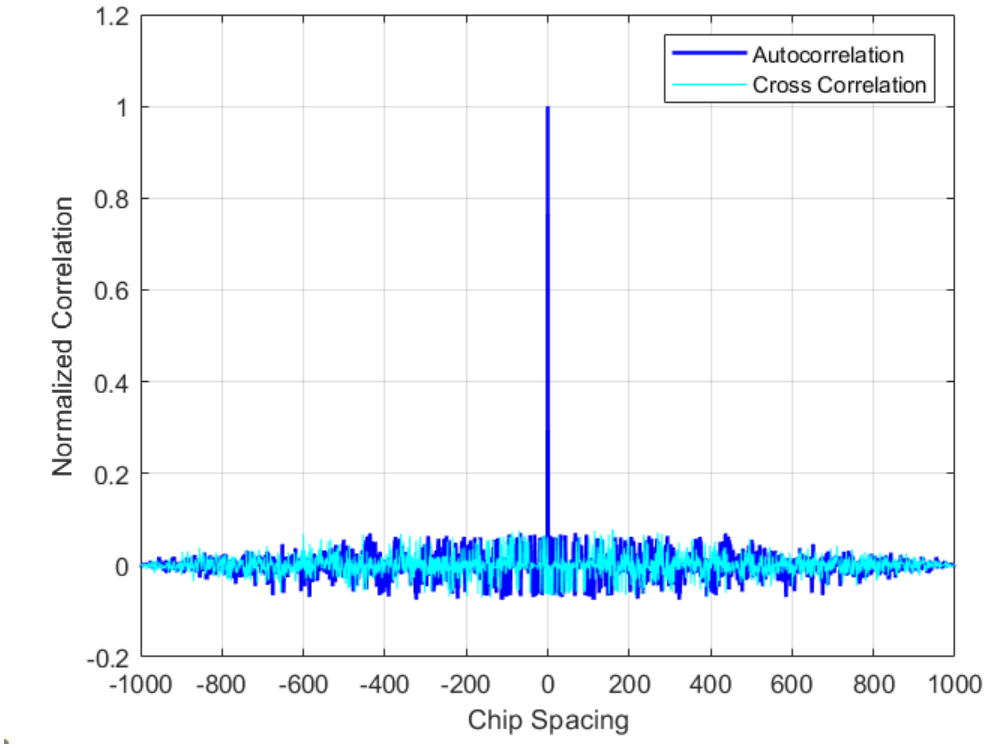

Figure 2.1: Autocorrelation and Cross Correlation of PRN 10 and PRN 33

Figure 2.2 shows a zoomed in plot of Figure 2.1. The triangle shape of the autocorrelation is crucial to the principles of signal tracking that will be described later.

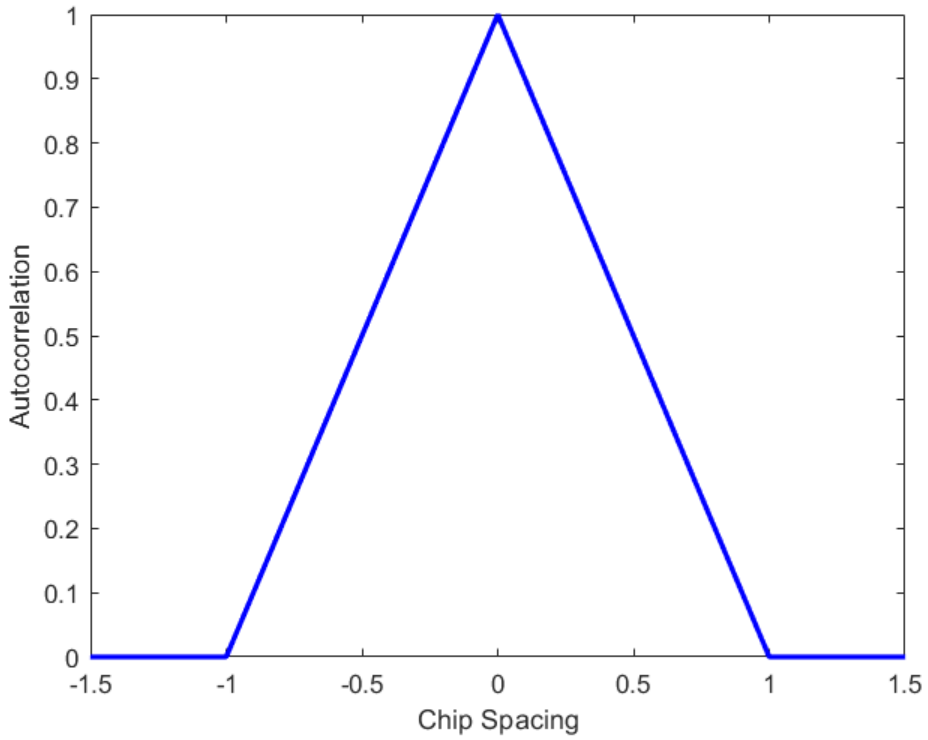

Figure 2.2: GPS Autocorrelation Function

2.1.2 Data Message

The GPS L1 C/A data message transmits important information to the receiver in a binary format at 50 Hz. The data message is modulated onto the carrier signal through binary phase shift keying (BPSK). The data message is comprised of the ephemeris and almanac. The ephemeris relays satellite parameters that the receiver uses to compute position. The almanac relays similar parameters for less accurate position results. Most commercial receivers use the ephemeris as their main source for calculating positions, but they also tend to use the ephemeris, almanac, etc. for integrity checks. Integrity checks are a necessary aspect of a commercial receiver, but they are not included in this work. The process of decoding the ephemeris navigation data message is explained thoroughly in [23] and hence will not be discussed in this thesis.

2.1.3 Signal Modulation

The GPS L1 C/A signal combines the carrier, PRN code, and navigation data using a technique called Binary Phase Shift Keying (BPSK). This technique essentially uses the square wave combined PRN code and data message to flip the carrier signal 180° in phase. BPSK is shown below in Figure 2.3.

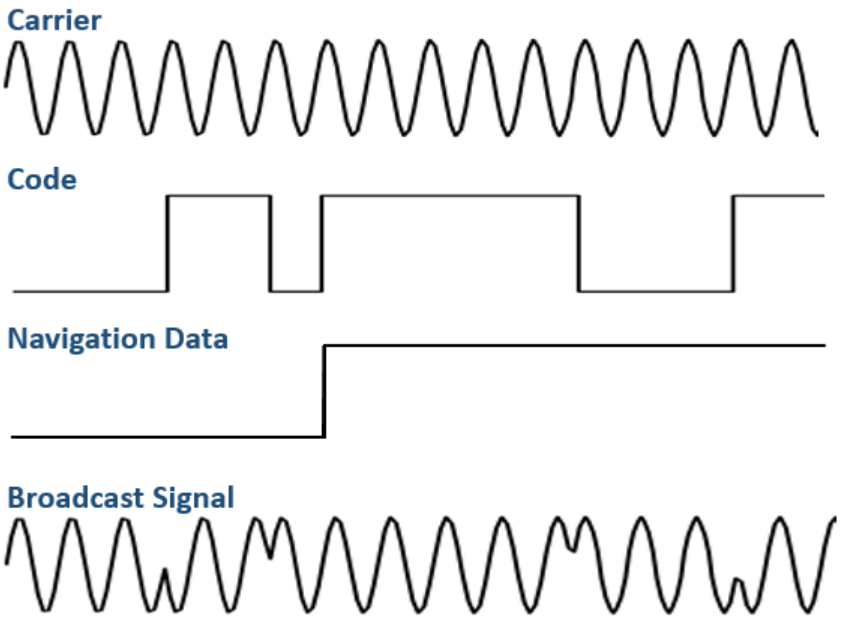

Figure 2.3: BPSK Signal Modulation [25]

The combined broadcast signal is shown at the bottom of Figure 2.3. The receiver is tasked with removing the modulation. This involves tracking the frequency and phase of the signal arriving to determine the precise time of signal arrival (TOA). The frequency spectrum of the GPS L1 C/A signal is shown in Figure 2.4. The GPS L1 C/A signal is typically not observable in the spectrum because the signal is received at -125 decibel-milliwatts (dBm), which is below the receiver noise floor. Receivers utilize adaptive gain controllers (AGC) to ensure the signal is still quantifiable by the receiver. In some cases, when the receivers have their AGCs tuned aggressively, the central lobe of the GPS signal is observed peaking through the noise floor.

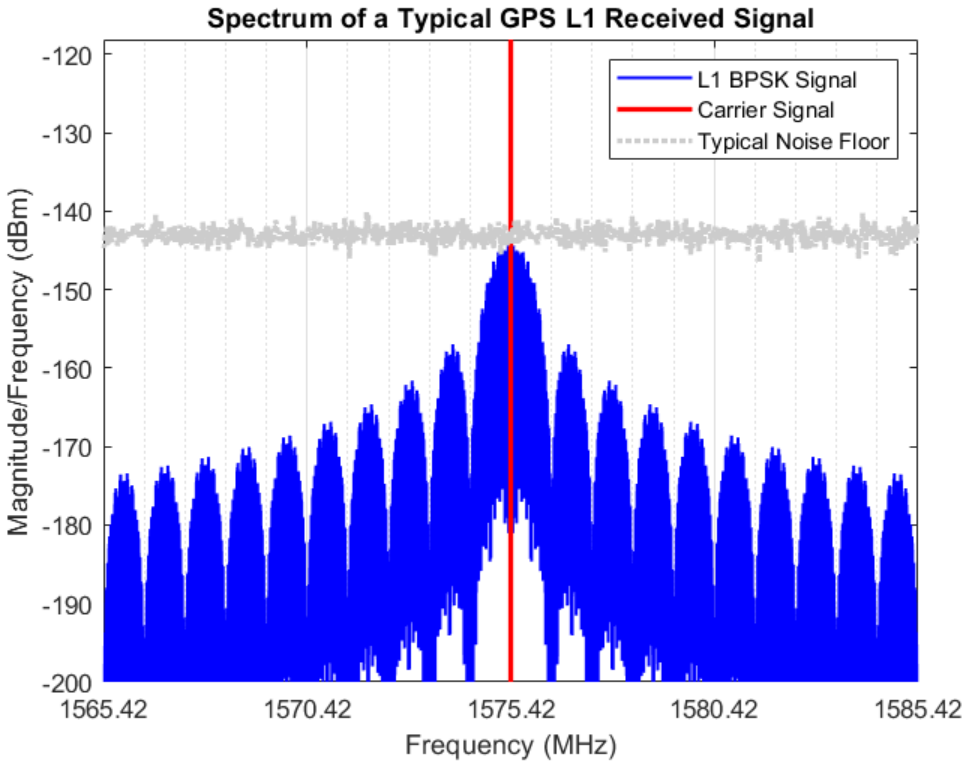

Figure 2.4: GPS L1 C/A Spread Spectrum

2.2 Signal Vulnerabilities

As noted earlier, the GPS signal is received at low power, typically -125 dBm or lower. Signals are transmitted at roughly 50 watts [4]. The drastic loss of power can be simply modeled using the free-space path loss equation shown in Equation (2.2).

$$L_{path} = (4\pi)^2 (d/\lambda)^2 \tag{2.2}$$

L_{path} is the loss of signal gain caused by a signal of wavelength λ traveling in a straight line a certain distance d. The gain loss of the signal can in theory be used to estimate the receive power of the signal, since the receiver knows the satellite locations.

In reality, the signals can be influenced by numerous disturbances that are, at most times, unknown to the receiver. The weakness of the signal leaves it vulnerable to interference such as jamming and spoofing, as well as natural factors such as multipath and blockages.

2.2.1 Jamming

Jamming is the intentional or unintentional blocking of the GPS signal. This occurs when another signal or noise is introduced to the spectrum that masks the already weak GPS signal as shown by Figure 2.5. Jamming is commonly linked with electronic warfare, but it has become more prominent in the civilian sector as infrastructure and daily life becomes more dependent on GPS and precise timing [26]. A prime example of accidental jamming occurred in 2011 when an individual jammed the automatic aircraft landing system at the Newark Airport [27].

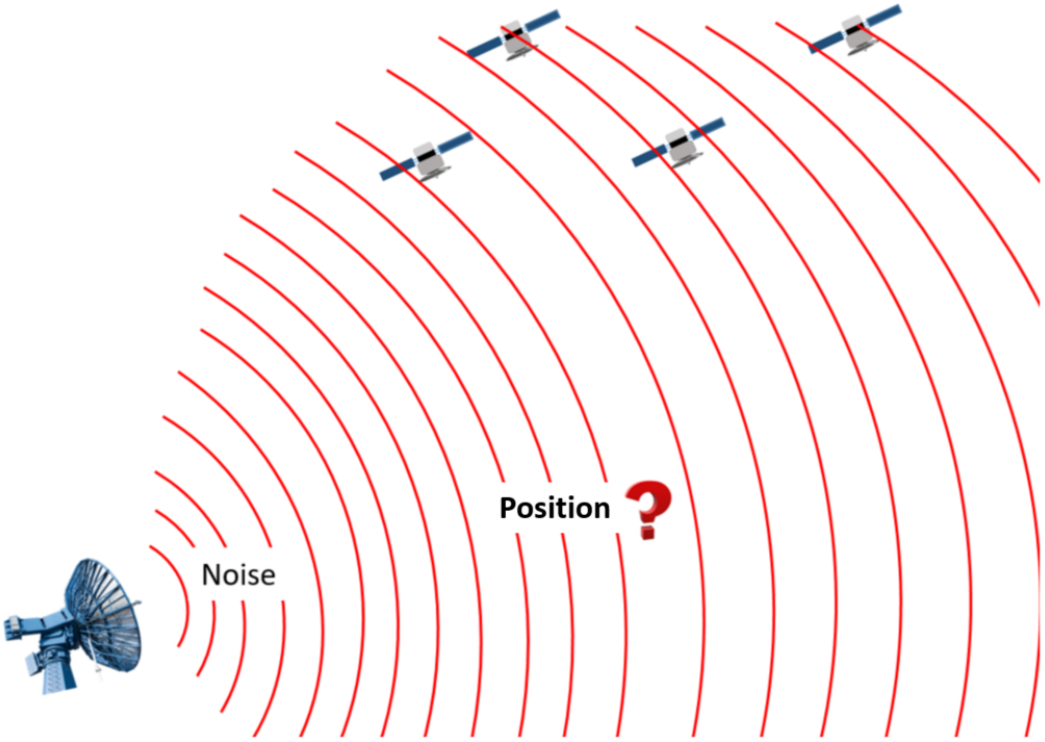

Figure 2.5: Jamming Attack

The simplest form of jamming is with a high power carrier signal or broadband noise (BBN). The jamming signal swamps the spectrum, rendering a simple receiver useless when the signal exceeds a 40 dB Jammer-to-Signal Power Ratio (J/S). This effect can be seen most clearly through signal acquisition, which is explained in detail in Section 3.2. Figure 2.6 shows a successful GPS L1 C/A acquisition on the left and a jammed acquisition search space on the right. The PRN code phase and signal Doppler shift is indiscernible after the new signal is introduced.

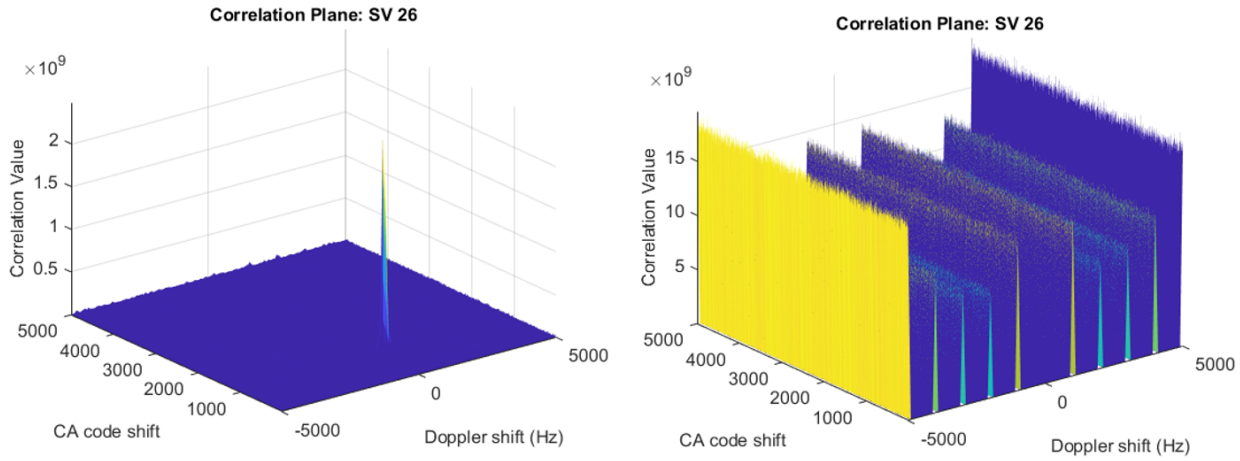

Figure 2.6: Effects of Jamming on Signal Acquisition

Jamming is usually easy to detect. Most modern receivers use simple methods such as power or spectral monitoring to determine when there are erroneous signals in the environment. Jamming effects are shown in Chapter 6. Mitigation methods are also introduced in Appendix B.

2.2.2 Spoofing

Spoofing is harder to detect than jamming. The U.S. Department of Homeland Security (DHS) defines spoofing as "RF waveforms that mimic true signals in some ways, but deny, degrade, disrupt, or deceive a receiver's operation when they are processed" [28]. A

simplified graphic of spoofing is shown in Figure 2.7. Prior to 2008, spoofing was considered an advanced threat that required expensive equipment to fool a receiver [6]. While still advanced, Bhatti and Humphreys demonstrated that a large scale spoofing attack is possible using inexpensive hardware [29]. They were able to gain control of a 65-m yacht in the Mediterranean Sea by gradually dragging its position from its true GPS position. This shows the power and deception of spoofing. More recently, a Tesla Model 3 level 2 autonomous driving system was successfully spoofed to take the wrong exits and veer off of the highway [30]. With autonomous vehicles growing ever more prevalent, engineers will be forced to develop algorithms to be more resistant to spoofing.

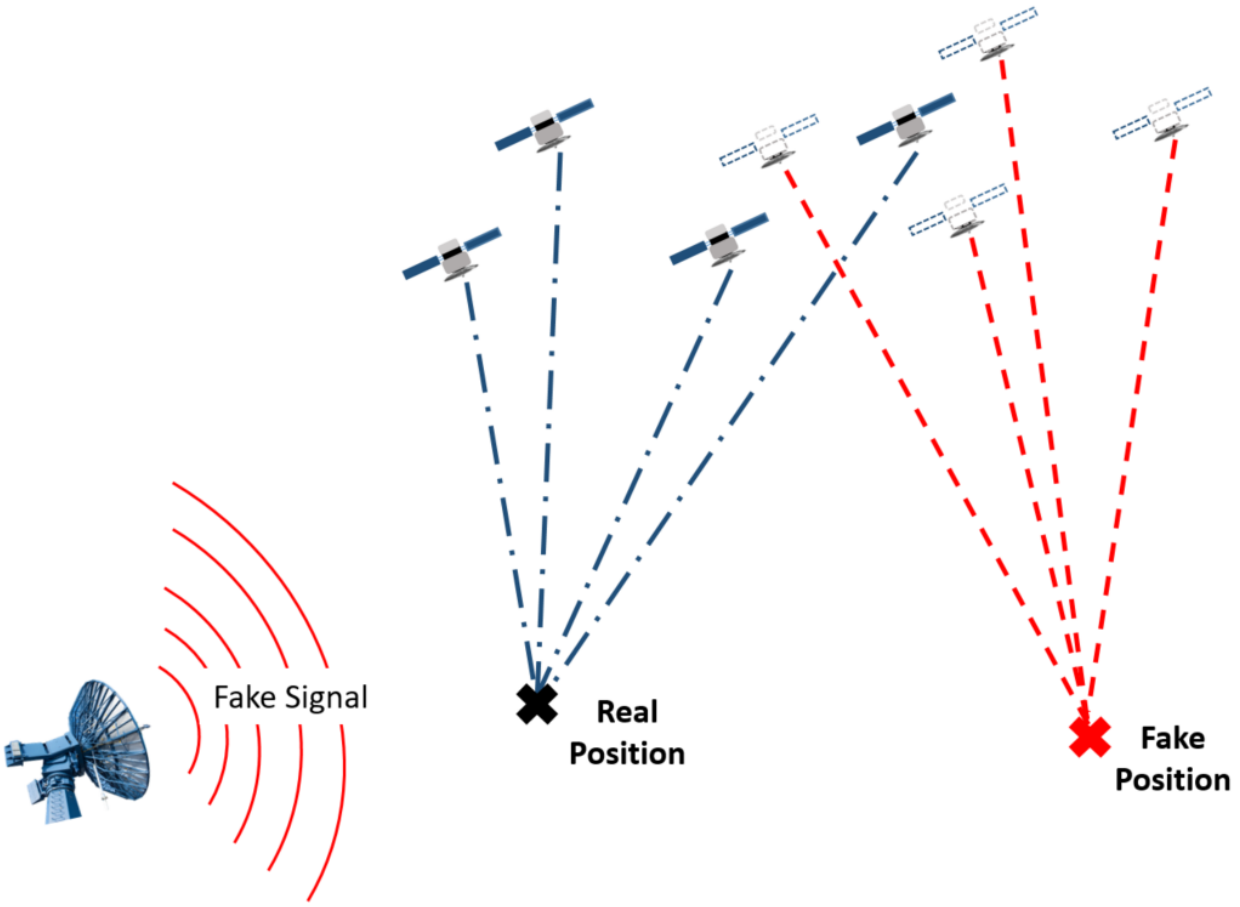

Figure 2.7: Spoofing Attack

Like jamming, spoofing can be detected in the signal acquisition plane. For a simple spoofing attack, Figure 2.8 shows a true GPS signal and a fake GPS signal in the same acquisition plane. The fake signal has a power level that is slightly higher than that of the true signal, and a simple receiver will lock onto the false signal. Smart spoofers align the acquisition peaks initially and then slowly drag the signals apart, which in turn captures the receiver.

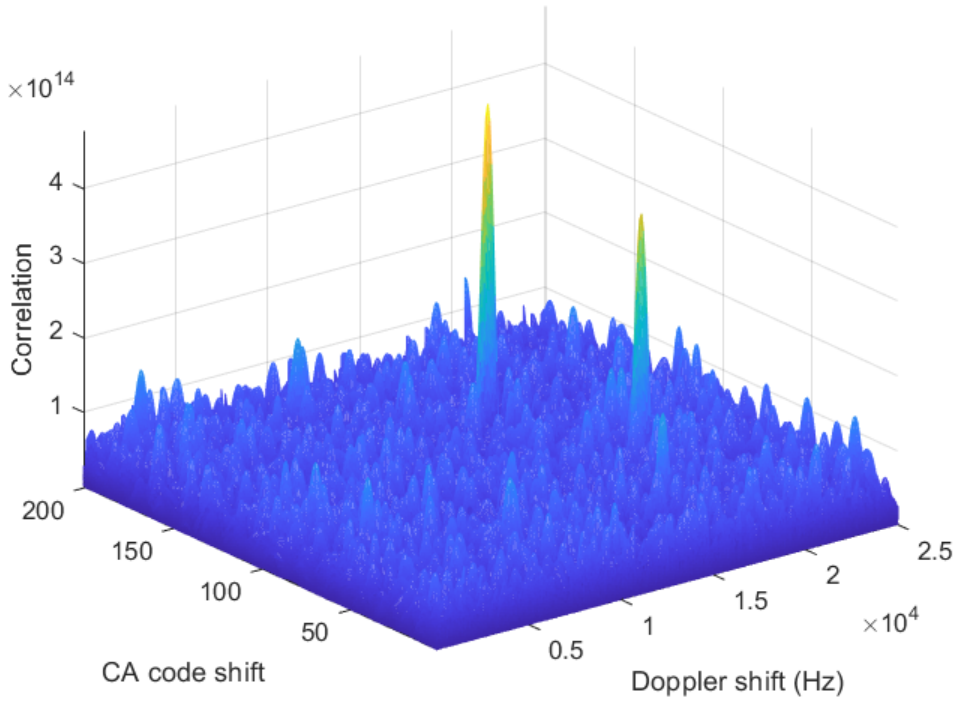

Figure 2.8: Effects of Spoofing on Signal Acquisition

Theoretically, the multi-antenna vector tracking coupled beamsteering receiver presented in this thesis is resilient to spoofing, but it was not tested in this work. The receiver receives more power in the direction of satellites using the beamsteering algorithm. This causes the ground-based spoofer signal receive power to decrease relative compared to the true satellite signal receive power.

2.2.3 Multipath

The most damaging natural vulnerability is from reflections, scattering, fading, etc. that can all be categorized as multipath. When signals reflect off objects and combine at the receiver, destructive interference can occur, decreasing signal power and causing problems for receivers. This idea is shown in Figure 2.9.

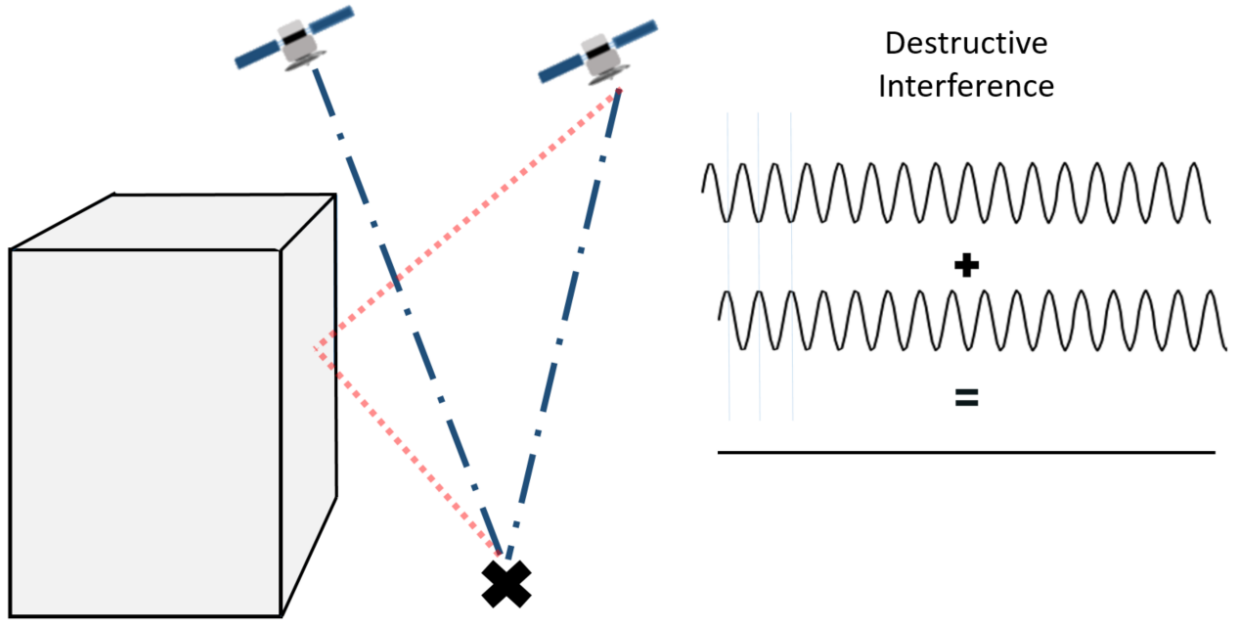

Figure 2.9: Multipath

Multipath occurs to some extent in every environment, but multipath is most prominent in urban canyon environments. An excellent example of an urban canyon environment with the effects of multipath included was presented at ION GNSS+ 2020 by Suzuki et al. [31]. Suzuki showed that non-line-of-sight signals (NLOS) can be tracked by a receiver. The urban canyon from his data collection is shown in Figure 2.10. Suzuki used a neural network to detect NLOS signals, and the percentages next to the signals represent the confidence that the signal is NLOS. In the urban canyon environment, only two signals are clean LOS signals. The rest of the signals will cause adverse effects on the receiver PVT solution.

Like spoofing, this thesis does not specifically address multipath, but the receiver presented should reduce the negative effects of multipath since it will always steer local replicas to align with the GPS L1 C/A signals. All NLOS signals present will be attenuated relative to the LOS signals.

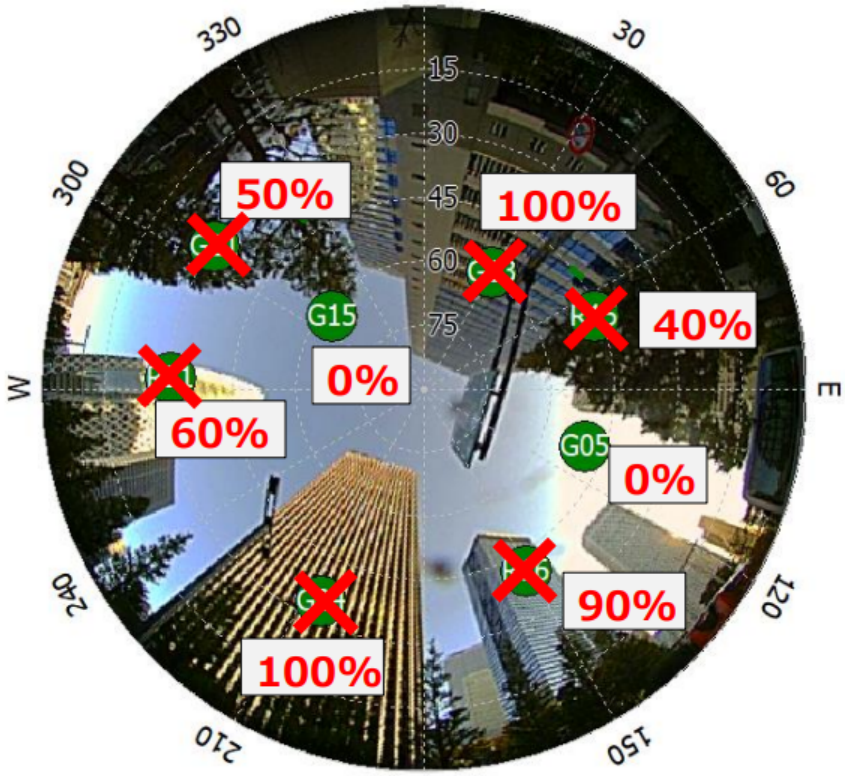

Figure 2.10: Urban Canyon with NLOS Signals Present [31]

2.3 Antennas

Antennas are used to enhance and amplify the low power GNSS signals when they arrive at the receiver. For GPS specifically, patch and helix antennas are the most common. Patch antennas are passive while active helix antennas require voltage to be sent up the line to power the antenna internal low noise amplifier (LNA). The proposed receiver utilizes four active antennas configured in a CRPA to manipulate the way the signals are combined to be beneficial for the receiver.

2.3.1 Controlled Reception Pattern Array (CRPA) Antennas

CRPAs can be used in numerous applications. CRPAs are used to direct cellular signals from towers [32], radar [33], and even, in the looser sense, the James Webb Space Telescope [34]. For radar and cellular applications, the array elements are used to direct antenna transmit gain or power in certain directions. The James Webb Space Telescope directs light in a certain direction (i.e. to a specific focus point) for clear pictures using its mirrors. For this thesis, the beamsteering algorithm aligns all antenna power to specific focal point that is the satellites locations.

To convey how a CRPA manipulates an antenna gain pattern, an ideal conceptional isotropic radiation pattern is shown in Figure 2.11. The pattern is uniform with unity gain in all directions.

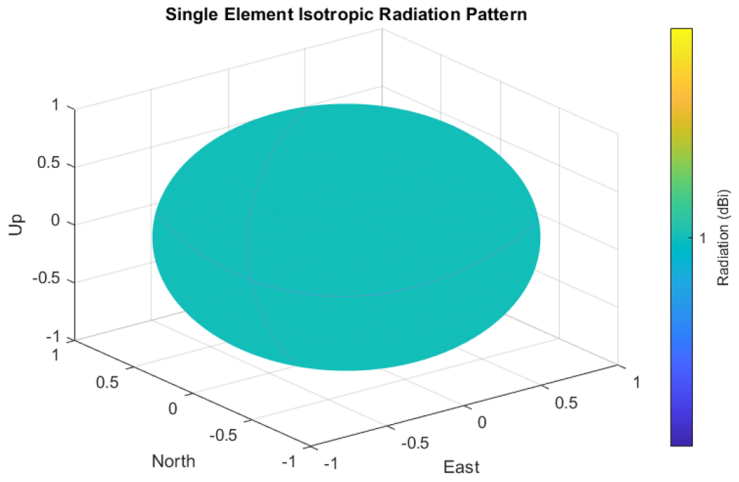

Figure 2.11: Single Element Isotropic Radiation Pattern

When more elements are added to form a CRPA, the signals are summed. The radiation pattern or gain pattern shown in Figure 2.12 is formed with a four element square CRPA of isotropic antennas spaced a quarter wavelength apart. This assumes there is no phase delay between incident signals directly overhead at zenith. With tight element spacing, the shape of the CRPA gain pattern is similar to that of the single element isotropic antenna. Note

the max magnitude of radiation is four at zenith, where the signals perfectly constructively combine to give max amplitude. The magnitude decreases at lesser angles when destructive interference is introduced.

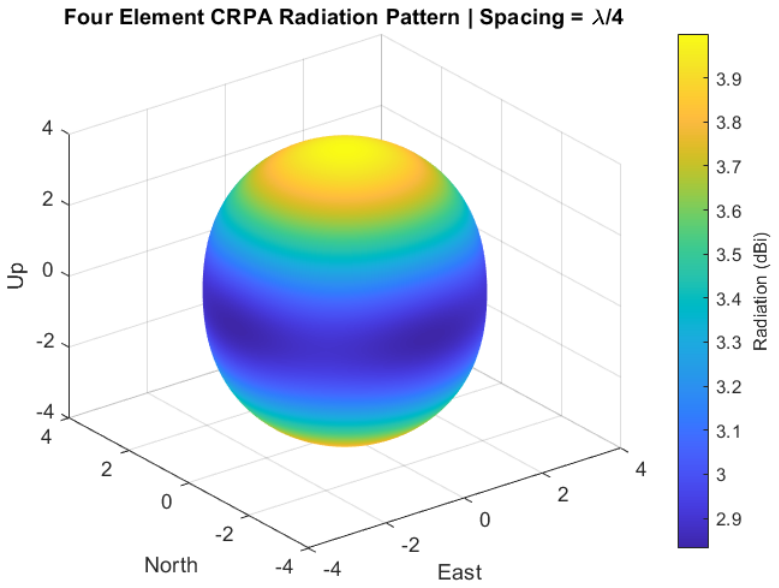

Figure 2.12: 4 Element CRPA Zenith Radiation Pattern ($\frac{\lambda}{4}$ spacing)

For a half wavelength square CRPA, the signal will have almost complete destructive interference at the low elevation angles, as shown in Figure 2.13. This is because the signals are 90 degrees out of phase. The majority of CRPAs leverage this aspect and their spacings are designed for specific frequencies. The antenna used for this work is a half wavelength antenna as shown in Figure 2.14.

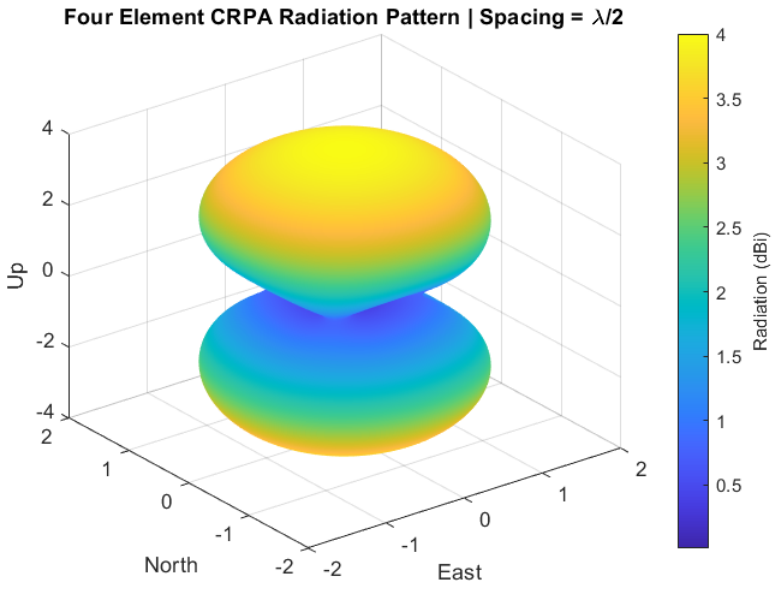

Figure 2.13: Four Element CRPA Zenith Radiation Pattern ($\frac{\lambda}{2}$ spacing)

Figure 2.14: Four Element CRPA Antenna

If a CRPA has a larger spacing than $\lambda/2$, destructive interference is introduced in undesirable array directions as shown in Figure 2.15 due to the reduction in beam width. Additional constructive interference is also introduced in directions where satellites could be present. For this reason, the majority of CRPAs do not increase spacings over a half wavelength.

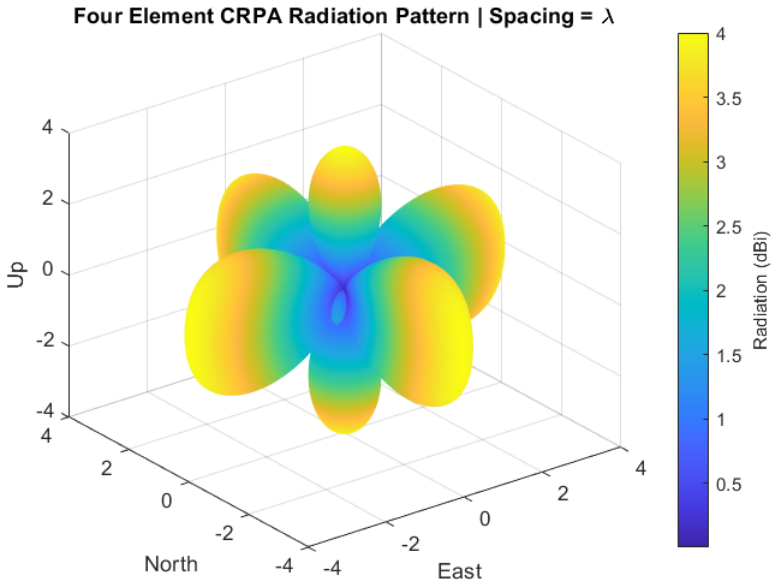

Figure 2.15: Four Element CRPA Zenith Radiation Pattern (λ spacing)

Beamforming

The real benefit of a CRPA is beamforming or the users ability to manipulate each signals phase to get the desired radiation pattern. Various beamforming techniques are described in [35]. Beamsteering is the process of directing antenna power or the antenna beam in a desired direction. For example, Figures 2.12, 2.13, and 2.15 show beamsteering to zenith because the phases enable the signals to constructively add at zenith.

Add general anachoic chamber picture and talk about different was of beamforming. Beamsteering, Nullsteering, and adaptive methods.

Ground Plane Effects

The previous sections made working with CRPAs seem simple but complications arise when applying CRPA algorithms to hardware. Perhaps the most challenging aspect of arrays is calibrating and understanding the array manifold. Calibration involves over the air transmission at the desired frequency to obtain array manifold gain and phase data. Once completed, the calibration process gives the receiver CRPA manifold data that can be used for corrections.

The CRPA needs to be calibrated with a ground plane similar to where it will be installed and used. The ground plane shape affects the gain on the antenna due to reflections. These effects were simulated by Andrew Kintz in his doctoral research at Ohio State University using ANSYS HFSS [36]. In Figure 2.16, Kintz created four ground planes; he created an ideal circular ground plane, the top of an airplane, the top of an airplane with wings and a tail fin, and all of those with angled wings and struts.

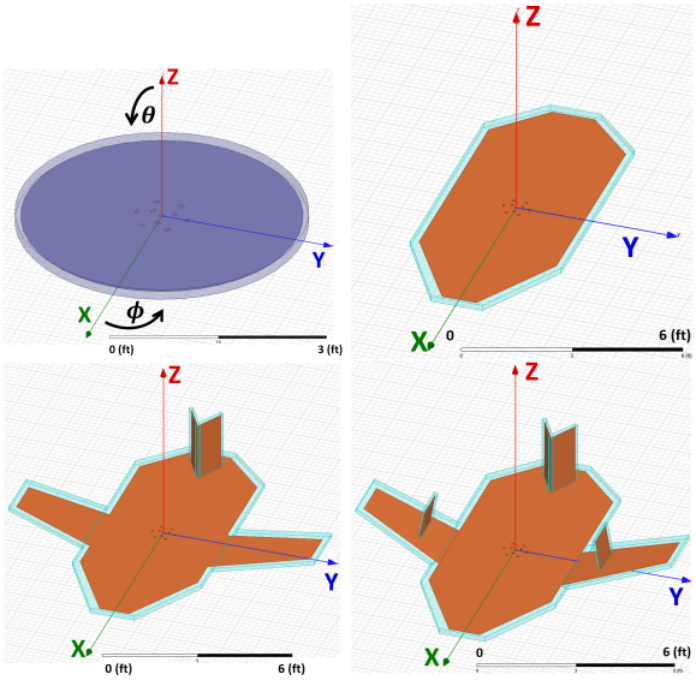

Figure 2.16: ANSYS HFSS Simulated Ground Planes [36]

The antenna element (only one) was simulated as a right hand circular polarized (RHCP) antenna. In Figure 2.17, the left column represents the gain plots for the RHCP antenna and each ground plane (ground planes going from left to right | top to bottom). The right column are the phase plots for the antenna. The RHCP antenna deviates from the ideal isotropic antenna, especially when the ground plane becomes more complex. When looking at gain alone, the variations in the pattern of one element will be compounded as the other CRPA elements are combined in the solution. This is why knowing array manifold data is important for CRPA related algorithms.

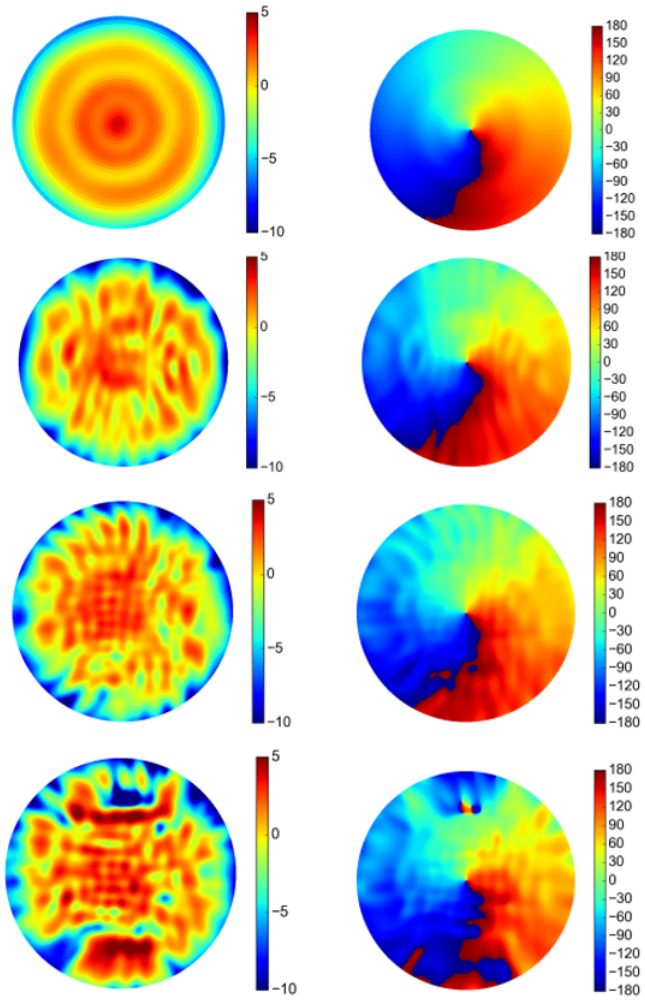

Figure 2.17: ANSYS HFSS RHCP Gain and Phase Effects [36]

For this work, the array manifold was unknown. To account for the array manifold, an acquisition based phase calibration was conducted to determine the combined line-to-line phase delays and array manifold delays induced on each channel. The calibration process is described in Section 6.3.1.

2.4 Conclusions

This chapter included a brief description of the GPS L1 C/A signal, including the carrier and code signal, navigation data message, and signal modulation. Signal vulnerabilities were discussed in detail, including jamming, spoofing, and multipath. Lastly, antenna arrays were introduced and beamforming, including the beamsteering used in this work, was presented. Given a receiver is able to calibrate its array manifold data and phase delays related to the difference in coaxial line lengths or hardware delays, the receiver should be able to perform beamsteering and nullsteering accurately and effectively.

Chapter 3

GPS L1 C/A Scalar Receiver Overview

Software receivers for GNSS were made popular by [8] and [37]. A standard receiver can be divided into four parts: front-end, acquisition, tracking, and PVT estimation. However, a software receiver includes everything after the analog-to-digital converter (ADC), which is acquisition, tracking, and PVT estimation. Section 3.1 describes the baseband front-end. Section 3.2 explains parallel code search acquisition. Section 3.3 outlines software tracking methods. Section 3.4 describes the PVT estimation methods used to obtain positions and velocities.

3.1 RF Front-End

The front-end of the receiver is the first subsystem used after the antenna to interact with the incoming signal. In this thesis, a baseband front-end model is used because all data was recorded with Ettus Universal Software Radio Peripherals (USRPs), which are shown in Figure 3.1.

Figure 3.1: Ettus Hardware Front-Ends: N210 (left), N310 (middle), E312 (right) [38]

An RF front-end is a combination of filters, amplifiers, and mixers that prepare the RF signal for digital conversion. The proper combination obtains the cleanest possible signal for input to the software acquisition algorithm. A block diagram for the front-end is shown below in Figure 3.2. The signal arrives at the antenna and is immediately bandpass filtered to remove the out-of-band spectrum. After bandpass filtering, the signal is amplified and mixed to an intermediate frequency.

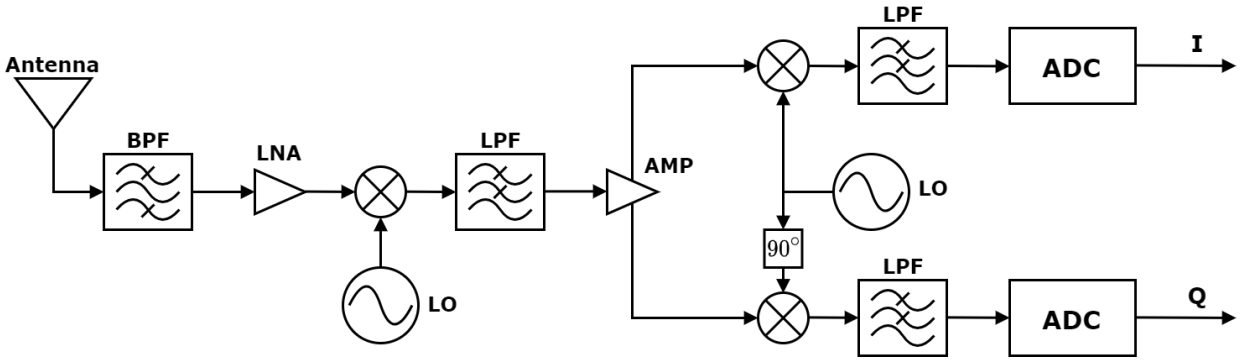

Figure 3.2: Baseband RF Front-End

Once at the intermediate frequency, the signal is filtered with a lowpass filter to remove the duplicate negative signal. Next the signal is amplified and mixed to baseband. At baseband, the signal is complex with the real portion of the signal being the in-phase and the complex portion being the quadrature. With the signal at zero frequency, a lowpass

filter is applied again to remove any remaining out-of-band spectrum. The signals are then passed to the ADC and converted to digital bits. With USRPs, the complex signal is saved as an interleaved signal with $IQIQIQ...$ bit arrangement. This bit interleaving process is not shown in Figure 3.2.

3.2 Parallel Code Search Acquisition

After the GPS signals are converted to a digital sequence, the receiver determines what signals are present in the sequence along with the corresponding signal Doppler frequencies and code phases. To determine a rough estimate of the Doppler frequency and code phase, a replica carrier and code signal are generated and varied in Doppler and code phase until a correlation peak is present. Figure 3.3 shows a failed acquisition plane, or an acquisition plane with no correlation over varying Doppler and code phase. Figure 3.4 shows an acquired satellite signal with a correlation peak at the signal Doppler shift and code phase.

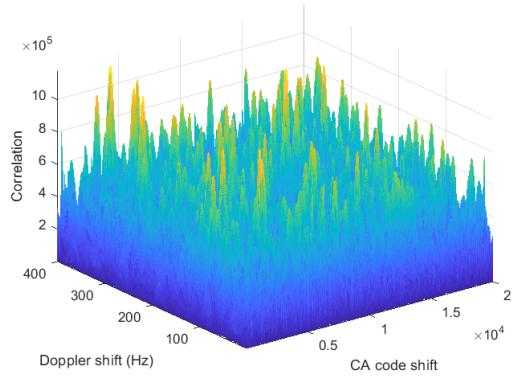

Figure 3.3: Failed Acquisition Plane

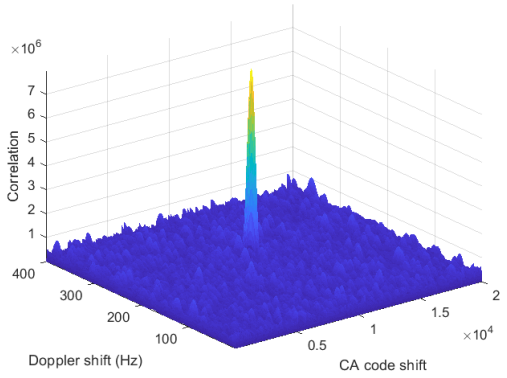

Figure 3.4: Successful Acquisition Plane

Both of the previous figures were generated from a parallel code search acquisition algorithm. There are other methods for acquiring that are described in [39]. Figure 3.5 shows the parallel code search architecture. The incident interleaved signal from the front-end is unwrapped and multiplied by an in-phase and quadrature carrier replica at the desired Doppler frequency. The in-phase and quadrature signal replicas are then summed and the

fast Fourier transform (FFT) is computed. The resultant signal is then multiplied by the complex conjugate of the FFT of the generated PRN code replica. The inverse FFT is then calculated from the multiplied signal. The absolute value of the signal is computed then the signal is squared to generate a vector of code phase correlation data at a specific Doppler frequency. The Doppler frequency is then varied to determine is there is a signal present. For dynamic signals, the Doppler frequency should be searched from -10000 Hz to 10000 Hz. In this work, the Doppler frequency is searched in 50 Hz increments.

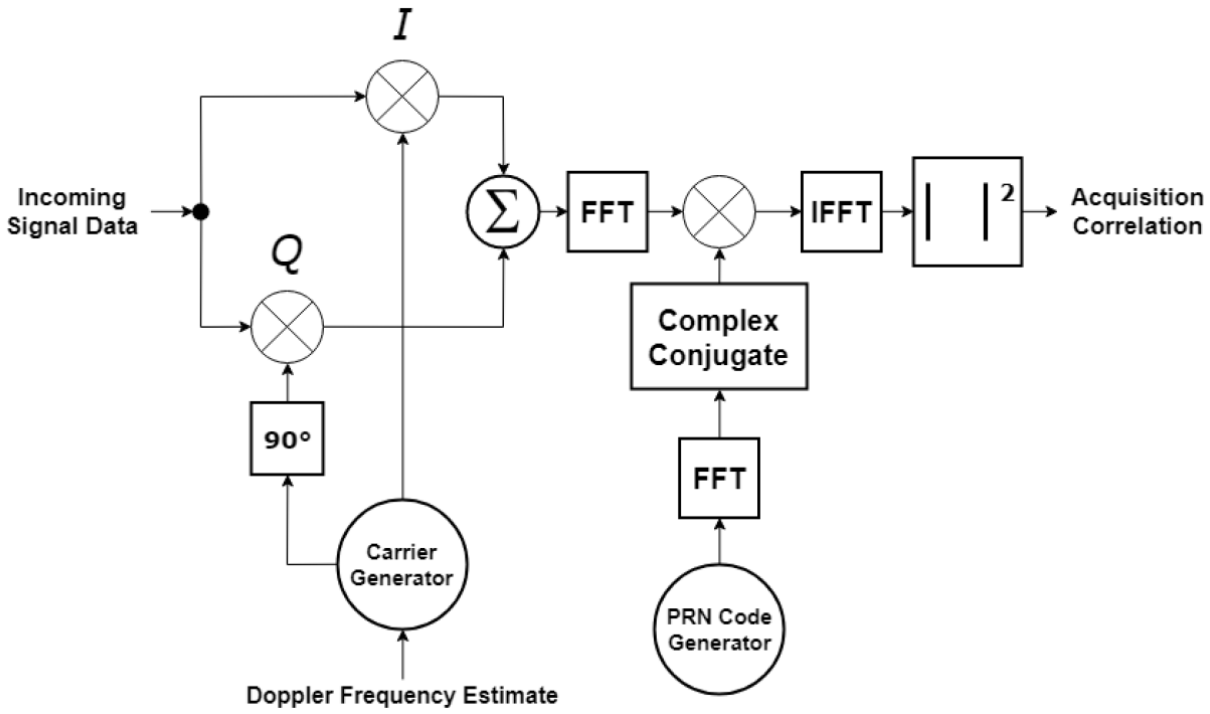

Figure 3.5: Parallel Code Search Acquisition [39]

For most situations, using a one millisecond integration period to perform signal acquisition is sufficient. However, a 10 millisecond integration period offers more robust performance in noisy signal conditions. Note that a navigation data bit flip occurs ever 20 milliseconds, and 10 millisecond acquisition is the highest integration period that is possible without knowing the data bit flip location. Acquisition is performed twice to ensure the data flip does not affect the acquisition solution. It is performed once on the first 10 milliseconds of data and

once on the second 10 milliseconds of data. The acquisition results from each data segment are then combined to obtain an overall group of acquisition results. An example correlation plane for a signal acquired with a 10 millisecond integration period is shown below in Figure 3.6.

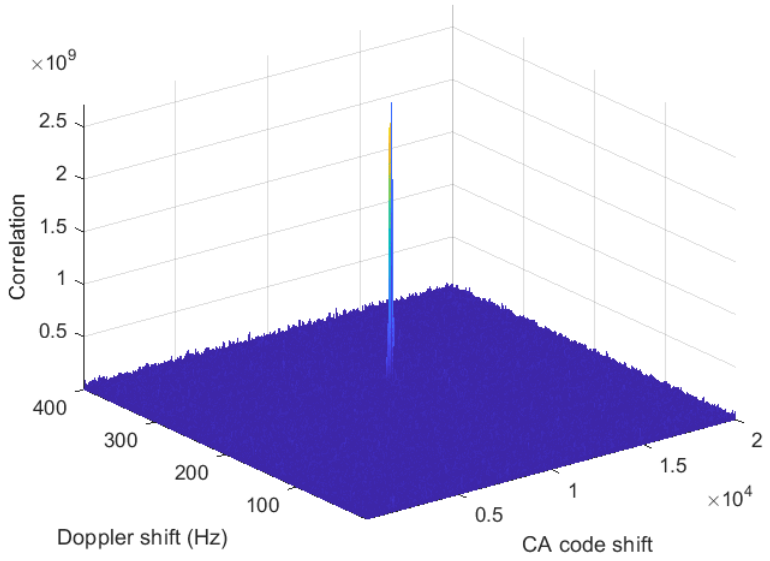

Figure 3.6: 10 ms Successful Acquisition Plane

There is a noticeable difference in the amplitude of the correlation peak between Figures 3.6 and 3.4. 10 millisecond acquisition gives a higher correlation peak that can be acquired in noisier conditions.

3.3 Tracking

After the rough estimates for Doppler frequency and code phase are obtained in signal acquisition, the signal is processed in a tracking control loop that drives signal frequency and phase errors to zero. Signal tracking obtains an accurate replica of the PRN code and carrier that can be used to generate pseudorange and pseudorange rate measurements. The tracking algorithm can be divided into two parts, which are a frequency lock loop (FLL) or phase lock loop (PLL) and delay lock loop (DLL). Tanner Watts describes each one of these

loops in detail, so these algorithms will not be explained as much in this work [39]. The scalar receiver designed for this thesis uses an FLL/DLL tracking loop to drive the carrier frequency into a usable range, and then a PLL/DLL tracking loop is used to generate the final precise replica signals. For most situations an FLL is not necessary, but for robustness, an FLL is implemented to ensure the receiver can obtain lock on dynamic noisy signals.

3.3.1 Carrier and Code Loop Filter Based Tracking

The Costas lock loop is the combination of a FLL/PLL and a DLL. A block diagram for the Costas lock loop is shown in Figure 3.7.

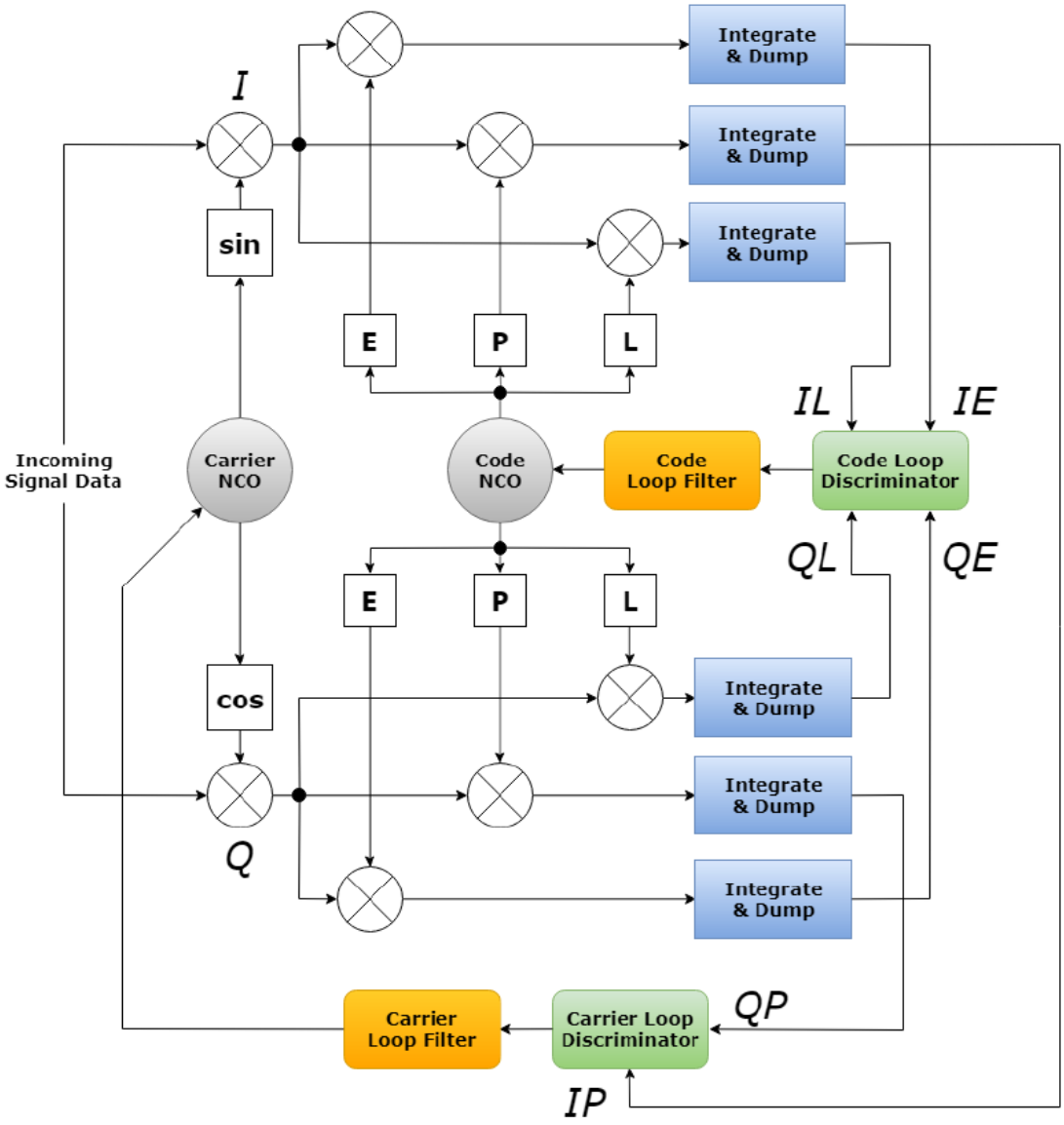

Figure 3.7: Costas Lock Loop [39]

The DLL attempts to obtain an accurate estimate of code frequency and code phase. The code replica frequency is initialized at the chipping rate of the GPS L1 C/A PRN code (1.023e6), and the initial code phase is obtained by acquisition. The incoming signal data is multiplied by the carrier replica. The combined signal and carrier replica are then multiplied by an early, prompt, and late PRN replica code numerically-controlled oscillator (NCO). The early code is offset by -0.5 chips, and the late code is offset by 0.5 chips. After the early, prompt, and late replicas are combined with the signal and carrier replica sequence, the resultant sequence is integrated to determine an early, prompt, and late set of in-phase and quadrature correlators. Since the data is discrete, the integration is described as a summation of all of the data in the sequence. The in-phase early (IE), in-phase late (IL), quadrature early (QE), and quadrature late (QL) correlators are then used to generate a code discriminator, which is a representation of the error in code phase. The equation for generating the code discriminator is shown in Equation (3.1).

$$\phi_{DLL} = \frac{1}{2}\frac{\sqrt{IE^2 + QE^2} - \sqrt{IL^2 + QL^2}}{\sqrt{IE^2 + QE^2} + \sqrt{IL^2 + QL^2}} \tag{3.1}$$

The code discriminator is lowpass filtered and fed back into the code replica in a process called the NCO update. The second order loop filter DLL code NCO update used in this work is from [39] and is shown in Equation (3.2).

$$f_{code_{k+1}} = f_{code_k} - K\left(\omega^2 T \phi_{DLL_k} + \omega\zeta(\phi_{DLL_k} - \phi_{DLL_{k-1}})\right) \tag{3.2}$$

The parameters used for the loop filter are $K = 1$, $\zeta = 0.707$ to 1.414, and $BW_n = 1$ to 5 Hz. Note that the bandwidth BW_n is related to the natural frequency ω and the damping ratio ζ by Equation (3.3) below.

$$BW_n = \frac{\omega(1 + \zeta^2)}{4\zeta} \tag{3.3}$$

The FLL and PLL attempt to obtain accurate estimates of the carrier frequency and carrier phase. The carrier replica is initialized at the acquisition carrier frequency, which is obtained by the Doppler acquisition estimate. The initial carrier phase is assumed to be zero. The in-phase and quadrature carrier replicas are then multiplied by the incoming signal. The resultant sequences are then multiplied by the prompt code replica. The in-phase and quadrature sequences are then summed to obtain in-phase prompt (IP) and quadrature prompt (QP) correlators. The in-phase and quadrature correlators are used to determine a frequency or phase discriminator representing the total carrier phase error between the replica signal and the incoming signal. Equation (3.4) shows the FLL discriminator.

$$\phi_{FLL} = \arctan 2 \left(\frac{IP_1 \cdot QP_2 - IP_2 \cdot QP_1}{IP_1 \cdot IP_2 + QP_1 \cdot QP_2} \right) / \pi T \tag{3.4}$$

IP_1 and IP_2 are the first half and the last half of the in-phase prompt correlator. Likewise, QP_1 and QP_2 are the first and last half the quadrature prompt correlator. T is the signal tracking integration period. arctan2 is the four quadrant arc tangent. The PLL discriminator is shown in Equation (3.5).

$$\phi_{PLL} = \arctan \left(\frac{QP}{IP} \right) / 2\pi \tag{3.5}$$

The FLL and PLL discriminators are filtered, similar to the DLL loop filter. The equation for the FLL loop filter is shown below in Equation (3.6).

$$f_{carr_{k+1}} = f_{carr_k} + TK \left(\omega^2 T \phi_{FLL_k} + \omega\zeta \left(\phi_{FLL_k} - \phi_{FLL_{k-1}} \right) \right) \tag{3.6}$$

The loop filter parameters for the FLL are $K = 1$ to 4, $\zeta = 0.707$ to 1.414, and $BW_n =$ 5 to 18 Hz. After the mean of the frequency discriminator drops below 50 for a period of time, the tracking loop switches to a PLL. The PLL loop filter is shown in Equation (3.7).

$$f_{carr_{k+1}} = f_{carr_k} + K \left(\omega^2 T \phi_{PLL_k} + \omega\zeta \left(\phi_{PLL_k} - \phi_{PLL_{k-1}} \right) \right) \tag{3.7}$$

The loop filter parameters for the PLL are the same as the FLL with $K = 1$ to 4, $\zeta = 0.707$ to 1.414, and $BW_n = 5$ to 18 Hz.

The in-phase prompt and quadrature prompt correlators are shown in Figure 3.8. Notice the in-phase prompt correlator resembles the GPS navigation message, and the quadrature prompt signal resembles zero mean noise. The navigation message is obtained from the in-phase prompt correlator using a parity check algorithm and signal time conversion, which is described in [8].

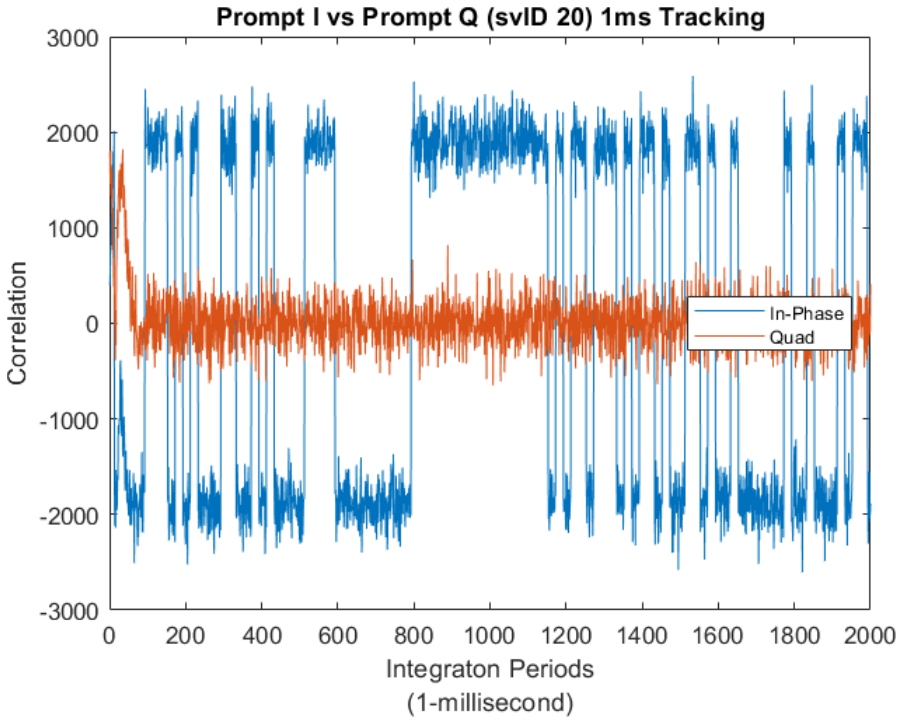

Figure 3.8: 1ms Scalar Tracking

Once the receiver obtains lock using a 1 millisecond tracking integration, the tracking integration period can be increased to 20 milliseconds by starting the loop on an in-phase prompt correlator navigation data flip location. Just like with acquisition, a 20 millisecond integration period yields higher correlation values allowing the tracking algorithm to track noisier signals. The difficulty in implementing a 20 millisecond tracking algorithm is in the handover between 1 millisecond and 20 millisecond tracking. The carrier frequencies, code

frequencies, carrier phase, and code phase corresponding to the bit flip location must be used to initialize the 20 millisecond tracking algorithm.

3.3.2 Alternative Kalman Tracking

An alternative form of the Costas lock loop is a Kalman smoother-based tracking loop. The Kalman filter presented in this section is a steady state Kalman filter, and the theory behind the filter is described in much detail in [40], [41]. Equations (3.8) and (3.9) show the stochastic dynamics model where T is the tracking integration period.

$$\begin{bmatrix} x_p \\ x_v \\ x_a \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ x_v \\ x_a \end{bmatrix}_k - \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \omega_{re_k} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \omega_k \tag{3.8}$$

$$y_{carr_{k+1}} = \begin{bmatrix} 1 & \frac{T}{2} & \frac{T^2}{6} \end{bmatrix} \begin{bmatrix} x_p \\ x_v \\ x_a \end{bmatrix}_k - \frac{T}{2}\omega_{re_k} + \eta_{k+1} \tag{3.9}$$

x_p is the carrier phase difference between the true signal and the receivers reconstructed signal. x_v is the carrier signals Doppler shift. x_a is the rate of Doppler shift. ω_{re} is the reconstruction frequency or the frequency shift in carrier and code due to the Doppler effect, ω_k is the process disturbance vector, and η_{k+1} is the measurement error term. The process noise covariance matrix is shown in Equation (3.10) where q_{ct} is the intensity of an equivalent scalar continuous-time white noise process that models the drift acceleration of the Doppler shift or signal jerk $\dddot{\phi}$.

$$Q_k = q_{ct} \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 & T^5/72 \\ T^4/8 & T^3/3 & T^2/2 & T^4/30 \\ T^3/6 & T^2/2 & T & T^3/24 \\ T^5/72 & T^4/30 & T^3/24 & T^5/252 \end{bmatrix} \tag{3.10}$$

For the DLL implementation, q_{ct} is set to $2\pi/10$. For the PLL implementation, q_{ct} is set to 1×10^{-4}. The values tune the bandwidth of the Kalman filter to be appropriate for signal tracking. The measurement noise covariance for the DLL and PLL are shown below in Equations (3.11a) and (3.11b) [41].

$$R_{d_{DLL}} = d(2-d)\left(\frac{1}{2} + \frac{1}{(C/N_0)T\,(2-d)}\right) \tag{3.11a}$$

$$R_{d_{PLL}} = \frac{1}{2(C/N_0)T}\left(1 + \frac{1}{2(C/N_0)T}\right) \tag{3.11b}$$

d is the early-late correlator spacing in units of chips. C/N_0 is the carrier to noise power density ratio and will be defined the same for the rest of this work. In implementation of static, short-term data where C/N_0 can be considered constant, $R_{d_{DLL}}$ was set to 1 and $R_{d_{PLL}}$ was set to 0.016. The steps of the Kalman filter are the same for the DLL and PLL. Equations (3.12) show the time update.

$$\hat{x}^- = A\hat{x}^+ + B_u\omega_{re_k} \tag{3.12a}$$

$$P^- = AP^+A^T + Q_k \tag{3.12b}$$

The a priori or propagated estimates are represented with a superscript $^-$. The a posteriori or corrected estimates are represented with a superscript $^+$. The measurement update for the Kalman filter is shown in Equations (3.13).

$$L = P^-C^T(CP^-C^T + R_d)^{-1} \tag{3.13a}$$

$$P^+ = (I - LC)P^- \tag{3.13b}$$

$$Z = y - C\hat{x}^- + (T/2)\omega_{re_k} \tag{3.13c}$$

$$\hat{x}^+ = \hat{x}^- + L \cdot Z \tag{3.13d}$$

y depends on whether the implementation is in a DLL or PLL. For the DLL, y is the code discriminator ϕ_{DLL}. For the PLL, y is the carrier discriminator ϕ_{PLL}. Equation (3.14) shows the reconstruction frequency update.

$$\omega_{re_{k+1}} = \left\{ \left[(1-\alpha)^2,\ (1-2\alpha)T + T,\ 0.5\left(-2\alpha T^2 + (2T)^2\right) \right] \hat{x}^+ - (1-2\alpha)\omega_{re_k} T \right\} / T \tag{3.14}$$

$\alpha = 0.7$ for the DLL, and $\alpha = 0.3$ for the PLL. These values yielded desirable performance for the Kalman filter. For the DLL, ω_{re} represents $f_{code} - f_{chip}$. For the PLL, ω_{re} represents $f_{carr} - f_{IF}$ or the Doppler frequency. Both carrier and code phase are updated using Equation (3.15), where the position state $\hat{x}_p^+$ is used to aid the carrier and code phase.

$$\phi_k = \phi_k + \hat{x}_p^+ \tag{3.15}$$

For implementation, Table 3.1 shows all of the tuning parameters.

Table 3.1: Kalman Tracking Tuning Parameters

Param \ Loop	**PLL**	**DLL**
$\boldsymbol{\alpha}$	0.3	0.7
$\mathbf{q_{ct}}$	1×10^{-4}	$2\pi/10$
$\mathbf{L_{ss}}$	$\begin{bmatrix} 0.018239 \\ 0.16903 \\ 0.78329 \end{bmatrix}$	$\begin{bmatrix} 0.0038532 \\ 0.0069603 \\ 0.0054157 \end{bmatrix}$

A steady state Kalman filter was used, so the Kalman gain L was set to the values shown in Table 3.1. Figure 3.9 shows the in-phase and quadrature correlators of the Kalman tracking loop when the correlators are tracking. There is a longer transient response in this implementation compared to the scalar implementation in Figure 3.8 due to the filters having different bandwidths. However, the filter still successfully tracks the signal.

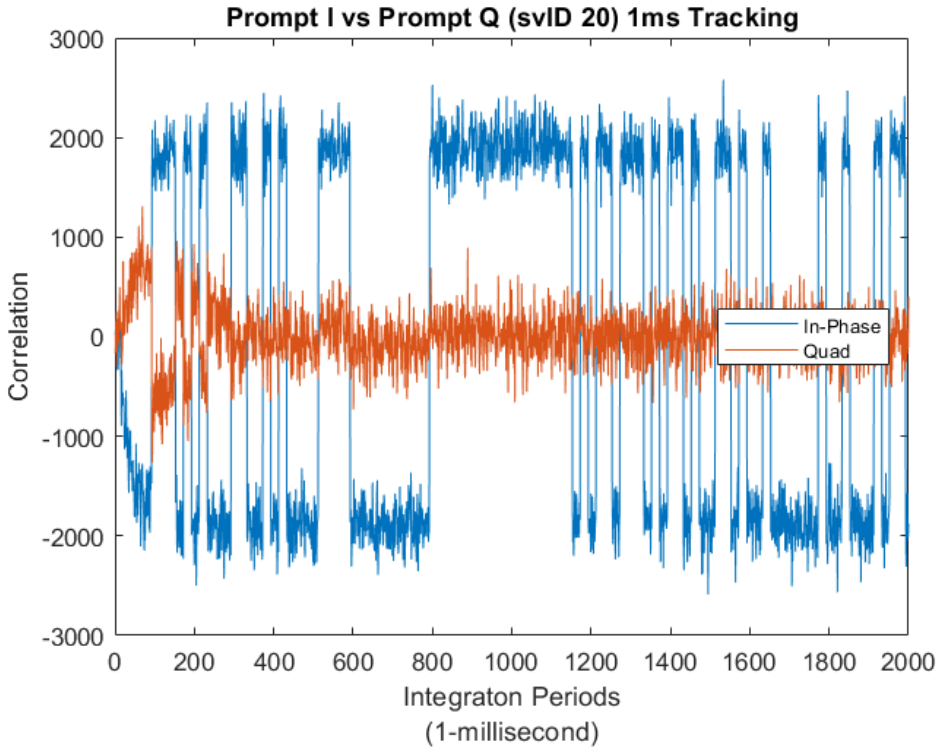

Figure 3.9: 1 ms Integration Period Kalman Tracking

The estimated Doppler frequency and code frequency for two satellites tracked from a static live-sky scenario are shown in Figures 3.10, 3.11, 3.12, and 3.13. The carrier frequency tracking is similar to the scalar tracking loop, as seen in Figures 3.10 and 3.11. The code frequency tracking obtained by the Kalman filter is a lot smoother than the scalar code frequency, as seen by Figures 3.12 and 3.13.

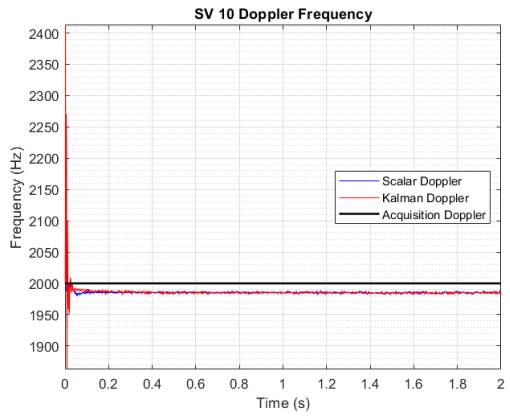

Figure 3.10: SV10 Carrier Tracking

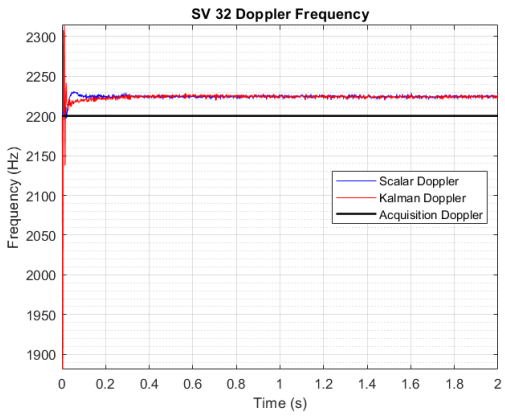

Figure 3.11: SV32 Carrier Tracking

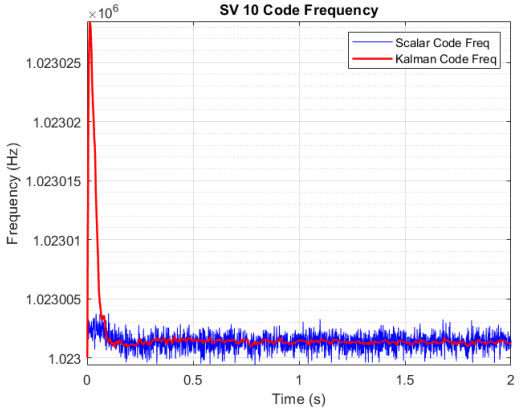

Figure 3.12: SV10 Code Tracking

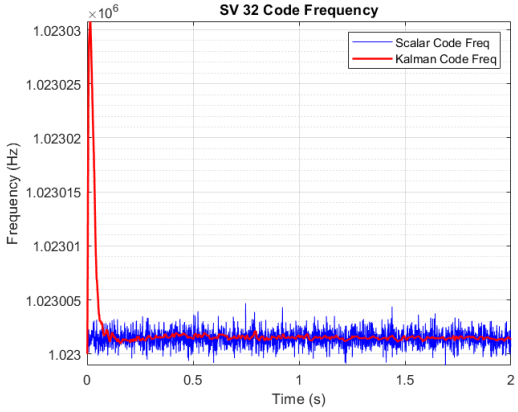

Figure 3.13: SV32 Code Tracking

The differences between the Kalman filter implementation and scalar tracking implementation are minimal. Therefore, when deciding which version to use, either is acceptable. The benefit of the Kalman filter is smoothing and tuning. The Kalman filter smooths the code and carrier frequency estimates and can be easily tuned with the process noise and measurement noise covariances.

When the Kalman tracking loop is configured with a 20 millisecond integration period, the correlators resemble those shown in Figure 3.14. Notice the amplitude difference on the in-phase correlator compared to Figure 3.9. With this amplitude difference, the 20 millisecond tracking implementation can handle noisier signals.

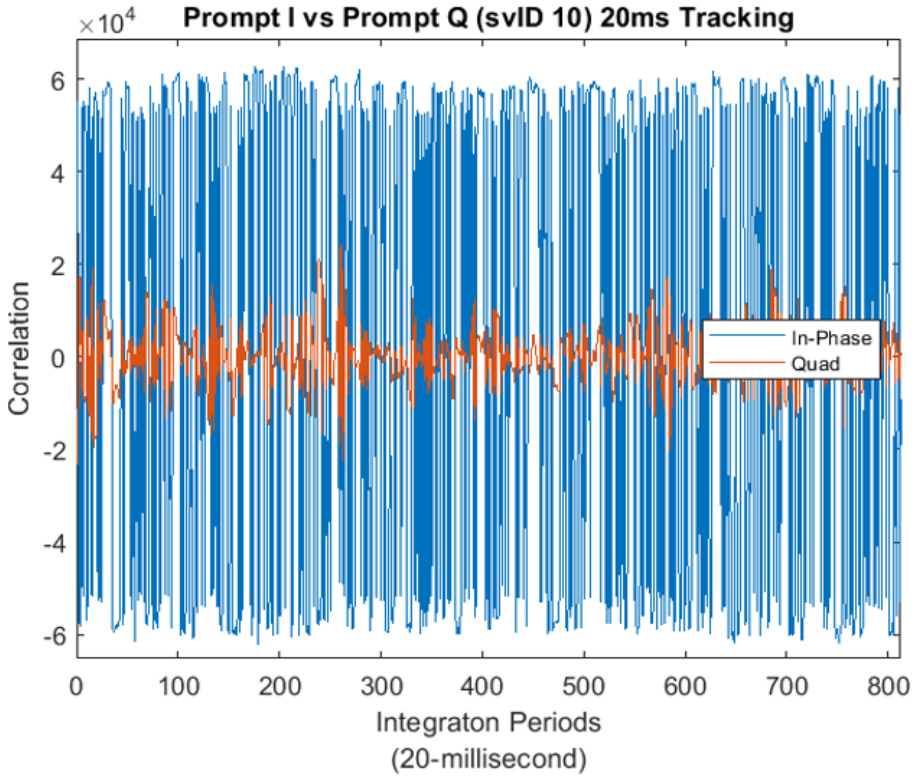

Figure 3.14: 20 ms Integration Period Kalman Tracking

3.4 Estimation of Position, Velocity, and Time (PVT)

After the receiver is locked on the GPS signals, the receiver is prepared to obtain navigation data. A parity check is necessary to determine the start of the navigation data message embedded in the in-phase prompt correlator outputs. After the preamble is found through the cross-correlation parity check, the in-phase prompt is converted from samples to navigation data bits; this conversion process as well as the parity check algorithm can be found in [8].

Once the navigation data bits are obtained, the binary data is decoded using the current format described in the most recent version of the GPS L1 C/A Interface Specification Document [23]. Decoding the navigation data is simple but very tedious.

[23] also outlines the equations used for satellite position and satellite velocity calculation from the decoded navigation data. This calculation is also outlined in [39]. Describing the calculation is beyond the scope of this work.

3.4.1 Pseudorange and Pseudorange Rate

Generation of pseudoranges and pseudorange rates are critical to the receiver positioning capabilities. The pseudorange is similar to the actual range r from the satellite to the receiver; the actual range is unobtainable due to atmospheric effects and clock biases affecting the satellite signals. The pseudorange is modeled in Equation (3.16).

$$\rho = r + c\,(b_r - b_{sv}) + I + T + \eta_\rho \tag{3.16}$$

c is the speed of light in a vacuum at $299,792,458$ [m/s]. b_r is the receiver clock bias, and b_{sv} is the satellite clock bias. The clock bias is caused by the receiver oscillator not being synchronized to the satellite oscillator. Delays caused by the ionosphere and troposphere are captured by I and T. The ionosphere contains particles that cause delays in GNSS signals. The troposphere contains moisture that causes delays in GNSS signals. Figure 3.15 shows how the signal can be delayed, and since the pseudorange is calculated using time, the pseudorange is inherently larger. [42] explains some methods for modeling and correcting atmospheric delays, but atmospheric errors are beyond the scope of this work.

Equation (3.17) represents the pseudorange rate.

$$\dot{\rho} = \dot{r} + c\dot{b}_r + \eta_p \tag{3.17}$$

$\dot{r}$ is the range rate and $\dot{b}_r$ is the receiver clock drift. The satellite clock drift term is omitted because satellites have stable rubidium and cesium clocks.

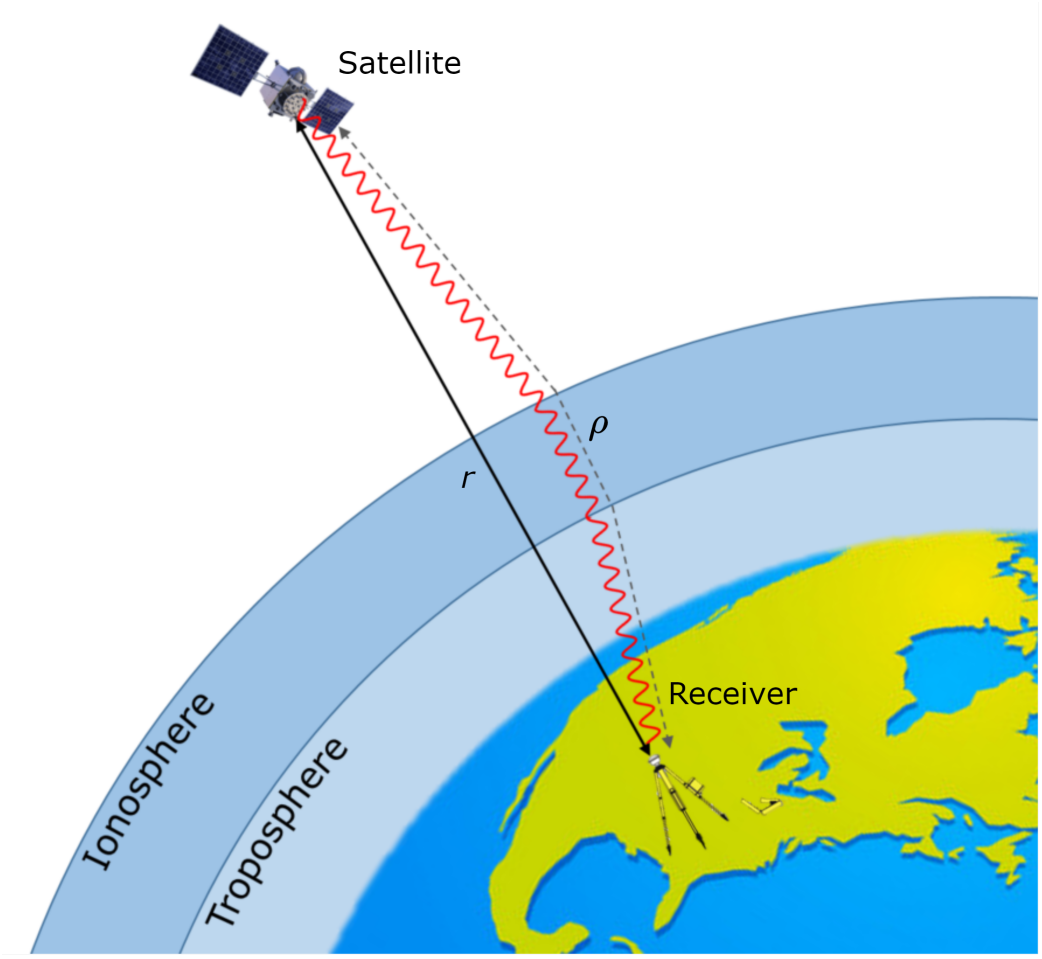

Figure 3.15: Pseudorange with Atmospheric Delays

In positioning, measurements of pseudoranges and pseudorange rates can be obtained from signal tracking. Equation (3.18) shows the pseudorange measurement is obtained from signal tracking.

$$\tilde{\rho}^{(j)} = \left(t_r - t_t^{(j)}\right) \cdot c \tag{3.18}$$

t_r is the user's receive time of the signal, and t_t is the signal transmit time. The transmit time is obtained using the satellite signal time of week (TOW) from the navigation data message, as well as the difference between each channel transmit time $\delta t_t^{(j)}$. The transmit time difference is calculated by logging start buffer sample locations in signal tracking. Equation (3.19) shows the process of obtaining the transmit time where T_{nav} is the position calculation

period.

$$t_t^{(j)} = TOW^{(j)} + (k-1) \cdot T_{nav} + \delta t_t^{(j)} \tag{3.19}$$

With most applications, assuming a common received time for each channel is acceptable. This is due to the receiver tracking time differences (i.e. change in time). Equation (3.20) is used to update the receive time.

$$t_{r_{k+1}} = t_{r_k} + (k-1) \cdot T_{nav} \tag{3.20}$$

The measured pseudorange rate is obtained using the Doppler frequency that comes from signal tracking as shown by Equation (3.21)

$$\tilde{\dot{\rho}}^{(j)} = -\left(f_{carr}^{(j)} - f_{IF}\right) \cdot \lambda_{wave} \tag{3.21}$$

where λ_{wave} is the carrier wavelength of the signal.

The estimated pseudoranges and estimated pseudorange rates can be obtained from the satellite and receiver positions and velocities. The model for the estimated pseudorange $\hat{\rho}$ is shown below in Equation (3.22).

$$\hat{\rho}^{(j)} = \sqrt{\left(x^{(j)} - \hat{x}_r\right)^2 + \left(y^{(j)} - \hat{y}_r\right)^2 + \left(z^{(j)} - \hat{z}_r\right)^2} + c\hat{b}_r \tag{3.22}$$

$x^{(j)}$, $y^{(j)}$, and $z^{(j)}$ are the geometric position states of the j^{th} satellite. $\hat{x}_r$, $\hat{y}_r$, and $\hat{z}_r$ are the estimated geometric position states of the receiver. c is the speed of light. $\hat{b}_r$ is the estimated receiver clock bias. The model for the estimated pseudorange rate $\hat{\dot{\rho}}$ is shown below in Equation (3.23).

$$\hat{\dot{\rho}}^{(j)} = \left(\dot{x}^{(j)} - \hat{\dot{x}}_r\right) a_x^{(j)} + \left(\dot{y}^{(j)} - \hat{\dot{y}}_r\right) a_y^{(j)} + \left(\dot{z}^{(j)} - \hat{\dot{z}}_r\right) a_z^{(j)} + c\hat{\dot{b}}_r \tag{3.23}$$

$\dot{x}^{(j)}$, $\dot{y}^{(j)}$, and $\dot{z}^{(j)}$ are the velocity states of the j^{th} satellite. $\hat{\dot{x}}_r$, $\hat{\dot{y}}_r$, and $\hat{\dot{z}}_r$ are the estimated velocity states of the receiver. $\hat{\dot{b}}_r$ is the estimated receiver clock drift. $a_x^{(j)}$, $a_y^{(j)}$, and $a_z^{(j)}$ are the j$^{\text{th}}$ satellite line-of-sight (LOS) unit vectors. The unit vectors shown in Equation (3.24) are estimated using the estimated range between the satellite and receiver $\hat{r}$, which describes the square root term in Equation (3.22), as well as the difference in satellite and receiver position.

$$a_x^{(j)} = \frac{x^{(j)} - \hat{x}_r}{\hat{r}^{(j)}} \qquad a_y^{(j)} = \frac{y^{(j)} - \hat{y}_r}{\hat{r}^{(j)}} \qquad a_z^{(j)} = \frac{z^{(j)} - \hat{z}_r}{\hat{r}^{(j)}} \tag{3.24}$$

The satellite unit vectors described here are the unit vectors that are fed back into the beamsteering algorithm discussed in Chapter 6.

3.4.2 Weighted Least Squares PVT Estimation

Four satellite signals are required to generate a three-dimensional position in time. The Newton-Raphson least squares solution calculates the position solutions for a standard GNSS receiver. Watts describes in detail the least squares estimation process [39]. The weighted least squares estimation technique is used to determine the scalar position solution in this work. The geometry matrix $\mathbf{G}$ is shown below in Equation (3.25).

$$\mathbf{G} = \begin{bmatrix} -a_x^{(1)} & 0 & -a_y^{(1)} & 0 & -a_z^{(1)} & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_x^{(j)} & 0 & -a_y^{(j)} & 0 & -a_z^{(j)} & 0 & 1 & 0 \\ 0 & -a_x^{(1)} & 0 & -a_y^{(1)} & 0 & -a_z^{(1)} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -a_x^{(j)} & 0 & -a_y^{(j)} & 0 & -a_z^{(j)} & 0 & 1 \end{bmatrix} \tag{3.25}$$

The weight matrix $\mathbf{W}$ is shown as a function of each satellite range variance $\sigma^{2^{(j)}_{\delta\rho}}$ and range rate variance $\sigma^{2^{(j)}_{\delta\dot{\rho}}}$ in Equation (3.26).

$$\mathbf{W} = \begin{bmatrix} \sigma^{2^{(1)}_{\delta\rho}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^{2^{(j)}_{\delta\rho}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^{2^{(1)}_{\delta\dot{\rho}}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^{2^{(j)}_{\delta\dot{\rho}}} \end{bmatrix} \tag{3.26}$$

The range and range rate variances are a function of C/N_0 and the signal signal tracking integration period T as shown by Equations (3.27) and (3.28) from [43].

$$\sigma^{2^{(j)}_{\delta\rho}} = \frac{\beta^2}{2\left(T \cdot 10^{(C/N_0^{(j)})/10}\right)^2} + \frac{\beta^2}{4T \cdot 10^{(C/N_0^{(j)})/10}} \tag{3.27}$$

$$\sigma^{2^{(j)}_{\delta\dot{\rho}}} = \left(\frac{2}{\left(T \cdot 10^{(C/N_0^{(j)})/10}\right)^2} + \frac{2}{T \cdot 10^{(C/N_0^{(j)})/10}} \right) \left(\frac{c}{\pi f_{L1} T} \right)^2 \tag{3.28}$$

β is the chip length of the GPS L1 C/A signal. c is the speed of light in a vacuum. Section 4.2 defined the PRN 33 method of estimating $\mathrm{C/N_0}$. Equation (3.29) shows the measurement vector, with the residuals $\delta\rho = \tilde{\rho} - \hat{\rho}$ and $\delta\dot{\rho} = \tilde{\dot{\rho}} - \hat{\dot{\rho}}$.

$$\delta\mathbf{z} = \begin{bmatrix} \delta\rho^{(1)} \\ \vdots \\ \delta\rho^{(j)} \\ \delta\dot{\rho}^{(1)} \\ \vdots \\ \delta\dot{\rho}^{(j)} \end{bmatrix} \tag{3.29}$$

The final equation for estimating the weighted least squares PVT solution is shown below in Equation (3.30). The positions and velocities are represented in the state vector $\hat{\mathbf{x}}$.

$$\hat{\mathbf{x}} = \left(\mathbf{G}^T\mathbf{W}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{W}\delta\mathbf{z} \tag{3.30}$$

Dilution of precision (DOP) can be estimated along with the state vector to determine the position errors associated with satellite geometry. Equation (3.31) shows DOP derived from the range only measurement observation matrix $\mathbf{G}$ [4].

$$\begin{bmatrix} \text{EDOP}^2 & \bullet & \bullet & \bullet \\ \bullet & \text{NDOP}^2 & \bullet & \bullet \\ \bullet & \bullet & \text{VDOP}^2 & \bullet \\ \bullet & \bullet & \bullet & \text{TDOP}^2 \end{bmatrix} = (\mathbf{G}^T\mathbf{G})^{-1} \tag{3.31}$$

East DOP (EDOP), North DOP (NDOP), Vertical DOP (VDOP), and Time DOP (TDOP) are all DOP parameters in Equation (3.32). Position DOP (PDOP) is defined below as the norm of EDOP, NDOP, and VDOP.

$$\text{PDOP} = \sqrt{\text{EDOP}^2 + \text{NDOP}^2 + \text{VDOP}^2} \tag{3.32}$$

PDOP will be the DOP parameter that will measure all of the position error due to scenario geometry for the rest of this work. Additionally, Horizontal DOP (HDOP) is defined without VDOP as shown in Equation (3.33).

$$\text{HDOP} = \sqrt{\text{EDOP}^2 + \text{NDOP}^2} \tag{3.33}$$

A static GPS L1 C/A scenario was simulated with a Spirent GSS9000 simulator and played into an Ettus USRP N210. As explained by this chapter, the data was then processed using a software defined receiver, and the position results were examined. The satellites

simulated in the scenario are shown in Figures 3.16 and 3.17. There were eight satellites overhead, with four satellites at low elevation. PDOP for the scenario was around 1.86 meters. Note that all satellites were used in the position solution, and an elevation mask threshold was not included.

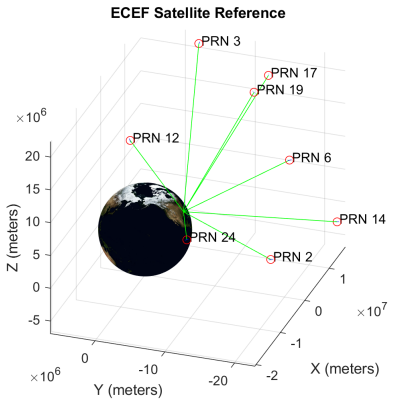

Figure 3.16: Satellite Vehicles Overhead

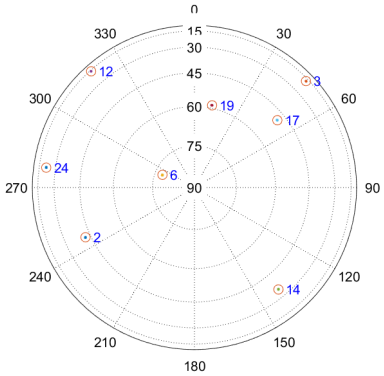

Figure 3.17: Static Roof Scenario Sky Plot

Figures 3.18 and 3.19 show the comparison between the true position in red and the estimated weighted least squares solution in blue.

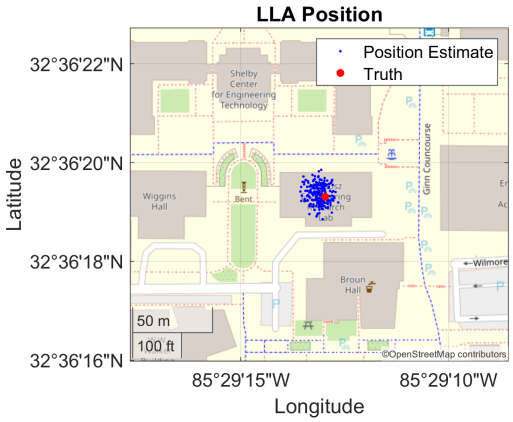

Figure 3.18: LLA Position Estimate with Weighted Least Squares

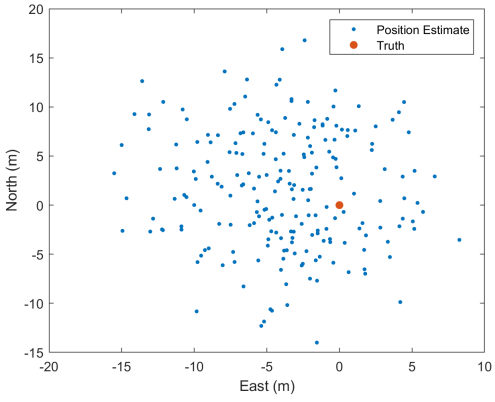

Figure 3.19: ENU Position Estimate with Weighted Least Squares

There is variance in the position solution. This can be attributed to the low elevation mask satellites diluting the solution, since we expect errors closer to the PDOP value. However, even with the variance, the least squares solution is still suitable for initializing vector

tracking, which is described in Chapter 5. For more precise position results, low elevation satellites can be removed and/or an EKF, as described in Appendix C, can be used to obtain a PVT estimate.

3.5 Conclusion

This chapter explains all of the fundamental pieces of a scalar tracking software defined receiver in detail. Receiver front-ends were explained, and a few example front-ends were given. Parallel code search acquisition was explained. Though other acquisition methods can be used, parallel code search acquisition was chosen for this work. A scalar tracking loop was described, as well as an alternative Kalman tracking loop. Weighted least squares positioning was also described with positioning results from a static simulated scenario.

Chapter 4

Carrier-to-Noise Density Ratio (C/N_0) Estimation

The C/N_0 is a measure of signal quality that is used in GNSS receivers. There are numerous ways to estimate C/N_0 [44]–[47]. RF jamming, spoofing, multipath, and blockages can all reduce the C/N_0, which negatively affects the receiver. The ability to estimate / measure C/N_0 is paramount for receivers to adapt filters so the receivers can maintain signal lock.

This chapter is laid out as follows. Section 4.1 defines a front end C/N_0 estimation technique that can be utilized to determine C/N_0 in an interference environment. Section 4.2 presents a known method for estimating C/N_0 in signal tracking, as well as contributes some alternative low SWaP-C methods for estimating C/N_0 in signal tracking. Section 4.3 compares static and dynamic results from the methods outlined in section 4.2. Section 4.4 outlines the conclusions that can be drawn from this analysis.

4.1 Front End Interference Estimation

One technique used to characterize the GPS L1 C/A signals involves determining a jammed C/N_0. This technique was first outline by Phillip Ward in [48] and was extensively used in [3]. The requirements for this method are knowledge of a signal C/N_0 prior to interference and knowledge of the jammer-to-signal power ratio (J/S). By knowing the two variables, the jammed carrier-to-noise power density ratio, $[C/N_0]_{eq}$, can be found. This analytical relationship is shown in Equation (4.1).

$$[C/N_0]_{eq} = -10\log_{10}\left(10^{-\frac{(C/N_0)}{10}} + \frac{10^{\frac{J/S}{10}}}{QR_c}\right) \tag{4.1}$$

C/N_0 is the unjammed carrier-to-noise ratio in dB, R_c is the PRN chipping rate, and Q is the spread spectrum processing gain adjustment factor. Note that Q is 1 for narrowband jamming, 1.5 for wideband spread spectrum jamming, and 2 for Gaussian noise jamming [17]. Figure 4.1 shows the relationship between J/S and $[C/N_0]_{eq}$.

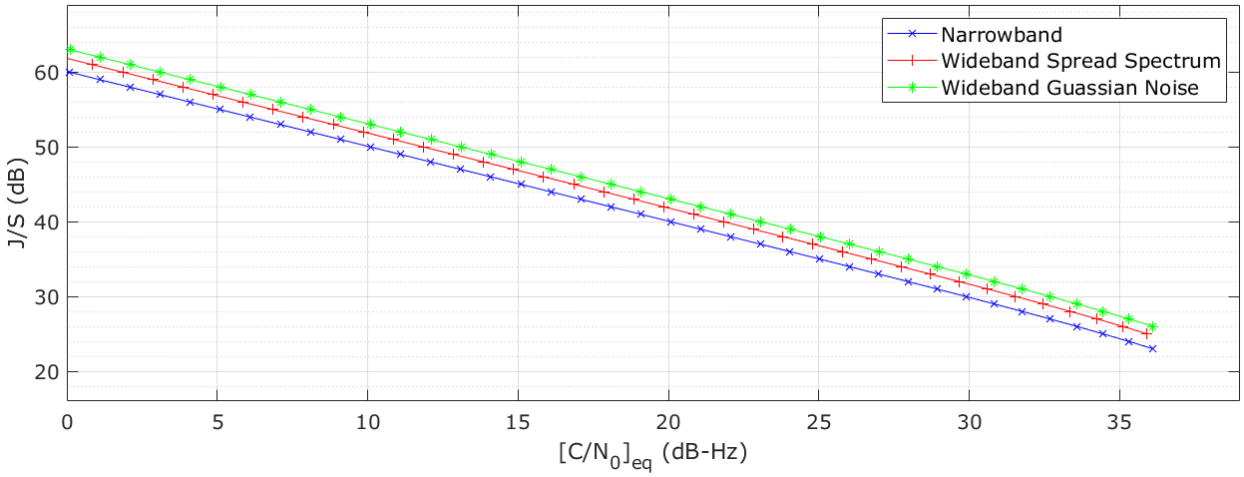

Figure 4.1: J/S Performance for Narrowband and Wideband Jammers [17]

At approximately 25 J/S for all three types of jammers, the $[C/N_0]_{eq}$ is tracking at 36 dB-Hz, which is high enough for scalar tracking receivers. Any higher J/S and the $[C/N_0]_{eq}$ drops below 36 dB-Hz degrading the signal quality.

4.2 Signal Tracking Correlator Based Estimation

A more independent method for estimating C/N_0 is correlator based C/N_0 estimation in the receiver signal tracking loop. One of the goals of the tracking loop is to minimize error between early, prompt, and late PRN replicas so that the receiver can accurately demodulate the signal. When the receiver is tracking, the early, prompt, and late PRN replicas will be higher than the noise floor with the prompt correlator being the highest. To estimate a signal noise characteristics, chip offsets are applied to the correlators to minimize signal correlation resulting in samples on the thermal noise (i.e. noise correlators). By taking the ratio of the power of the tracking correlators and the noise correlators, the receiver can accurately

estimate C/N_0. There are multiple variants of this methodology, and some of the methods require more computation.

Section analyzes a proven 36 correlator method [3], as well as alternative low SWaP-C methods that are a first millisecond of every twenty milliseconds method (FMET), a reduced correlator method, and a combined reduced correlator FMET method. This section also compares using each satellite channel chip offset PRN correlators and uncorrelated PRN 33 correlators.

4.2.1 36 Correlator Method

The 36 correlator method has been implemented in numerous receiver designs by Auburn University and Integrated Solutions for Systems (IS4S) [3], [25], [39], [49]–[51]. The tracking correlators used are the in-phase early (IE), in-phase late (IL), quadrature early (QE), and quadrature late (QL). The chip offsets used to generate noise samples in this method are 180, 120, 240, 700, 660, 740, 1730, 1700, and 1760. The variance is then calculated to give an estimated noise variance $\hat{v}^2$ as shown in Equation (4.2).

$$\hat{v}^2 = \text{var}\left(\begin{bmatrix} \text{IE180} & \text{IL180} & \text{QE180} & \text{QL180} \\ \text{IE120} & \text{IL120} & \text{QE120} & \text{QL120} \\ \text{IE240} & \text{IL240} & \text{QE240} & \text{QL240} \\ \text{IE700} & \text{IL700} & \text{QE700} & \text{QL700} \\ \text{IE660} & \text{IL660} & \text{QE660} & \text{QL660} \\ \text{IE740} & \text{IL740} & \text{QE740} & \text{QL740} \\ \text{IE1730} & \text{IL1730} & \text{QE1730} & \text{QL1730} \\ \text{IE1700} & \text{IL1700} & \text{QE1700} & \text{QL1700} \\ \text{IE1760} & \text{IL1760} & \text{QE1760} & \text{QL1760} \end{bmatrix}\right) \tag{4.2}$$

The estimated signal power is given as a function of the original signal tracking correlators as seen by equation (4.3).

$$\hat{A}^2 = (\mathrm{IE} + \mathrm{IL})^2 + (\mathrm{QE} + \mathrm{QL})^2 \tag{4.3}$$

Both the estimated noise variance and estimated signal power from equations (4.2) and (4.3) are filtered using a complementary filter as shown in Equation (4.4).

$$\hat{x}_k = \alpha \tilde{x}_{k-1} + (1 - \alpha)\tilde{x}_k \tag{4.4}$$

For this implementation, α is set to 0.98. Using the filtered noise variance and filtered signal power along with the signal tracking integration period T, the receiver $\mathrm{C/N_0}$ is estimated in dB using equation (4.5) below.

$$C/N_0 = 10 \log_{10} \left(\frac{\hat{A}^2 - 4\hat{v}^2}{2\hat{v}^2 T} \right) \tag{4.5}$$

This method is computationally intensive due to the number of correlators generated for the noise variance estimation. The next three methods are intended to decrease the computational power needed to estimate $\mathrm{C/N_0}$.

4.2.2 Low SWaP-C $\mathrm{C/N_0}$ Estimation Methods

The three low SWaP-C $\mathrm{C/N_0}$ estimation methods are a FMET method, a reduced correlator method, and a combined FMET reduced correlator method. After the methods are described, another approach is presented by using an uncorrelated PRN 33 to generate noise correlators. Static and dynamic results are compared in a later section.

FMET Method

Instead of estimating $\mathrm{C/N_0}$ every millisecond coherently in signal tracking, the $\mathrm{C/N_0}$ can be estimated using the first millisecond of data from every 20 milliseconds of signal data.

The same 36 correlators are used in this method (i.e. the number of power correlations does not change). However, $1/20^{\text{th}}$ of the signal data is used in the estimation. When using $1/20^{\text{th}}$ of the data to calculate noise variance, the noise variance is approximately $1/20^{\text{th}}$ of the 20 millisecond value. The noise variance is thereby scaled by 20 to obtain accurate estimates of C/N_0. Even though this method is less computationally intensive, this method still requires multiple dedicated receiver channels for computation.

Reduced Correlator Method

The reduced correlator method uses all 20 milliseconds of the signal data to estimate C/N_0. Therefore, no noise variance scaling occurs in this method. The main idea of this method is to reduce the number of chip offset correlators. By only using one chip offset as opposed to using nine, the matrix for noise variance estimation is reduced. Equation (4.6) describes the new noise variance estimation. The only chip offset used is 180.

$$\hat{v}^2 = \text{var}\left(\begin{bmatrix}\text{IE180} & \text{IL180} & \text{QE180} & \text{QL180}\end{bmatrix}\right) \tag{4.6}$$

The C/N_0 estimates become noisier when the number of correlators used in the noise variance estimation are reduced. To counteract this effect, the filtering parameter α was changed to 0.8. The change in the filter parameter makes the filter remove more noise, but it also makes the estimates less responsive to dynamics. The reduced correlator method greatly reduces computation time, but it is inconsistent due to its unresponsive nature to dynamics. This can be seen in the static and dynamic figures later in this chapter.

FMET Reduced Correlator Method

The last method combines the aspects of the FMET method and the reduced correlator method. An offset scaling is applied to the noise variance since the noise variance is using $1/20^{\text{th}}$ of the data. The noise variance is also computed using equation (4.6). Both the noise

variance and signal power estimates are filtered using $\alpha = 0.8$. This method is the least computationally intensive method of all methods.

Uncorrelated PRN 33 Noise Correlators

All of the previous C/N_0 estimation methods can be modified to use PRN 33 instead of using their associated satellite PRN code for noise correlator generation. PRN 33 can be an alternative tool for guaranteeing the noise variance estimate, because it does not correlate with any other signals. The signal tracking correlators used for the power estimate still use their dedicated satellite PRNs to determine the individual channel C/N_0. However, only one set of 4-36 noise correlators needs to be calculated instead of M sets where M in the number of satellites being tracked. This greatly reduces computation for all methods reducing the burden of C/N_0 estimation for the receiver

4.3 Live Sky Correlator C/N_0 Estimation Comparison

For a simple experiment, live sky GPS L1 C/A data was collected with a Nordnav Rxx-2 Frontend and a Ublox EVK-M8T. The hardware setup for the data collection is shown in Figure 4.2.

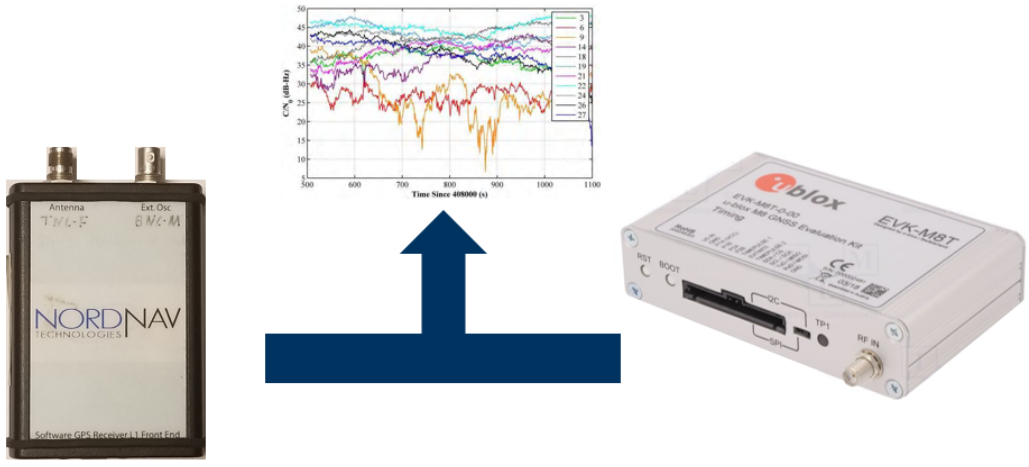

Figure 4.2: Nordnav Rxx-2 and Ublox EVK-M8T

Static and dynamic GPS L1 C/A data was processed with scalar tracking and compared against the Ublox EVK-M8T output C/N_0. The static results are shown in Section 4.3.1. The dynamic results are shown in Section 4.3.2.

4.3.1 Static C/N_0 Results

The C/N_0 estimation results from static GPS L1 C/A data collected from atop the Auburn University Woltosz Engineering Research Building are shown in Figures 4.3, 4.4, and 4.5. The signals with the stronger C/N_0 were selected for this analysis. The left plot in each figure shows the different estimation methods using each channel PRN to create chip offset noise correlators. The right plot in each figure shows using PRN 33 to create noise correlators. Figure 4.3 shows the C/N_0 estimation results from GPS SV 5.

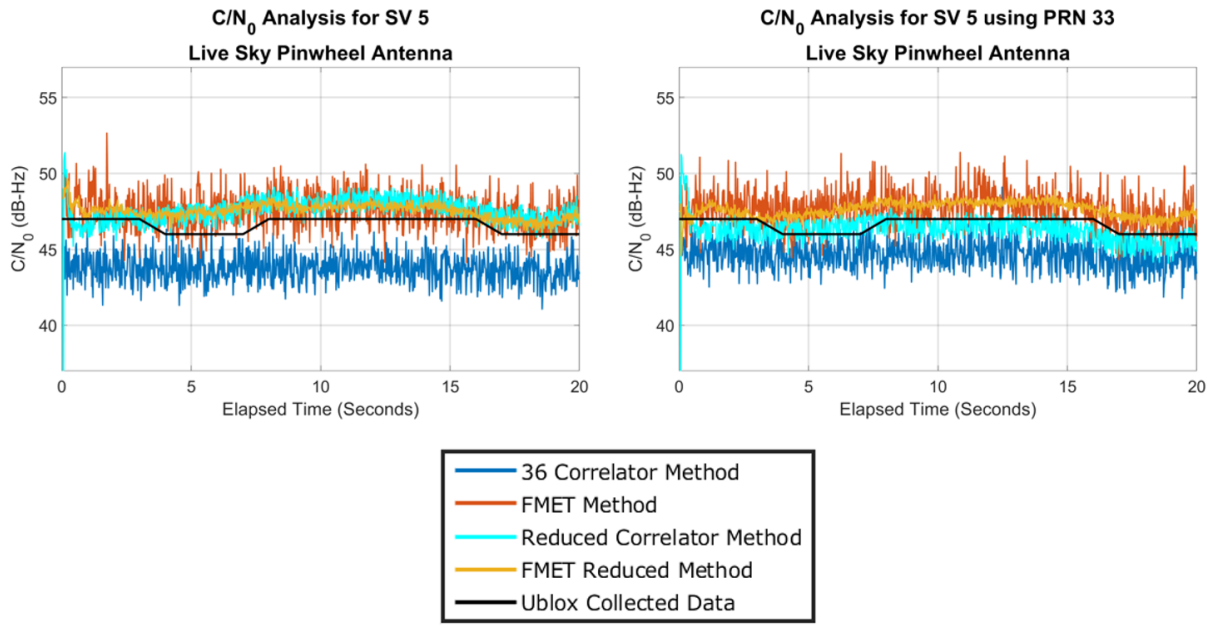

Figure 4.3: Static PRN 5 C/N_0 Results (Left) and Static PRN 33 C/N_0 Results (Right)

The 36 correlator method is negatively biased from the Ublox collected data, which is being used as the truth data. The reduced correlator, FMET, and FMET reduced correlator methods all closely align with the Ublox collected data. There is not a large difference between noise generation with PRN 5 versus noise generated with PRN 33.

Figure 4.4 shows the C/N_0 estimation results for GPS SV 13. In these plots, it can be observed that the reduced correlator method shows a large bias of about 5 dB in both the PRN 13 and PRN 33 plots. This is a attributed to not having enough noise correlators to obtain an accurate noise variance. The other methods more closely align the the Ublox collected data.

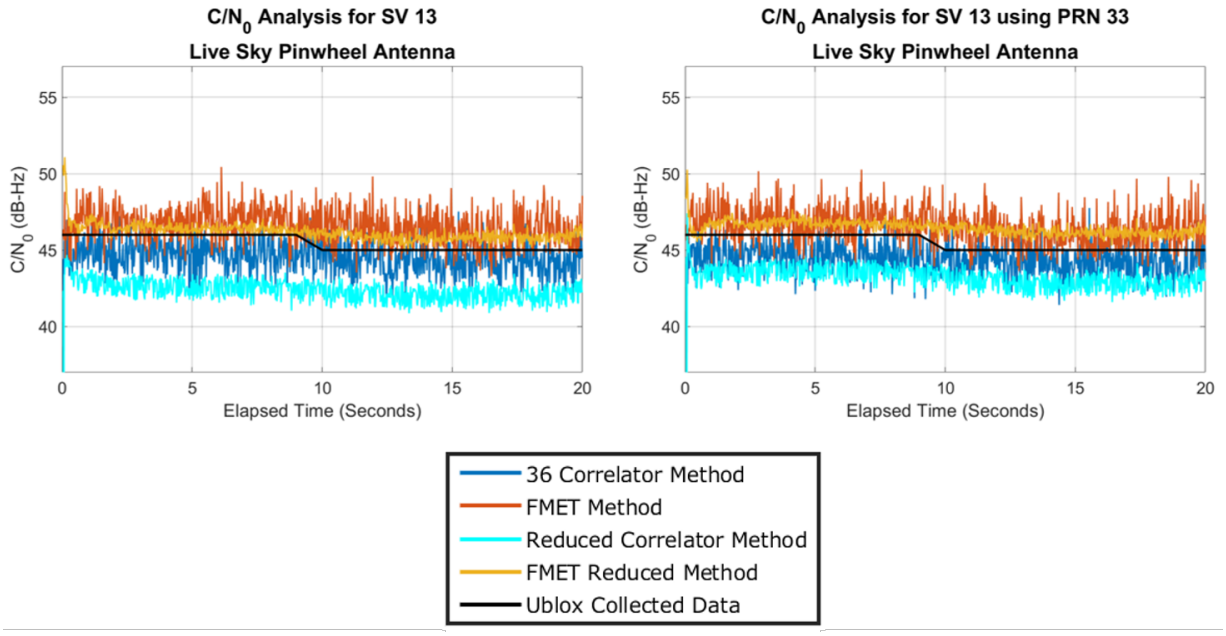

Figure 4.4: Static PRN 13 C/N_0 Results (Left) and Static PRN 33 C/N_0 Results (Right)

The last static data shown is from GPS SV 15 in Figure 4.5. The reduced correlator method completely fails to estimate C/N_0 using PRN 33 to generate noise correlators. However, the reduced correlator method is able to estimate the C/N_0 when the chip offset noise correlators are used. The other algorithms were comparable in their performance.

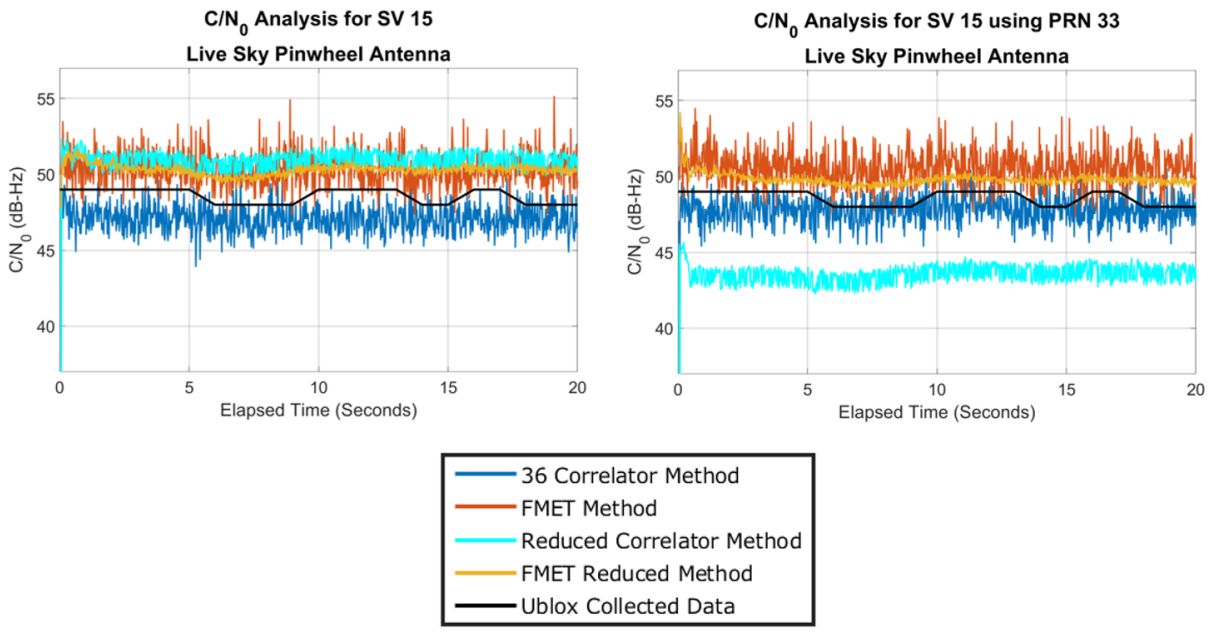

Figure 4.5: Static PRN 15 C/N_0 Results (Left) and Static PRN 33 C/N_0 Results (Right)

4.3.2 Dynamic C/N_0 Results

The C/N_0 estimation results from dynamic GPS L1 C/A data collected in downtown Auburn, Alabama are shown in Figures 4.9, 4.10, and 4.11. The vehicle used to collect the GPS L1 C/A data, as well as the NovAtel antenna used for the collection, is shown in Figure 4.6. Figure 4.7 show the data collection computers used for the experiment.

Figure 4.6: Research Vehicle Setup for Dynamic C/N_0 Data Collection

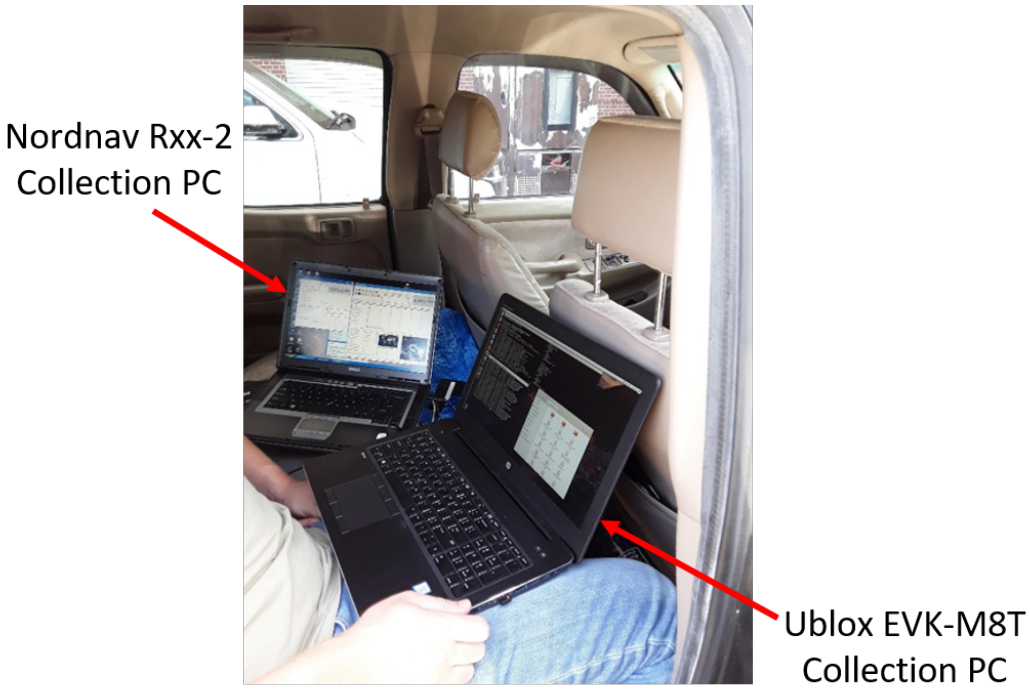

Figure 4.7: Dynamic C/N_0 Instrument Setup

The route for the experiment is shown in Figure 4.8. The route can be characterized as an urban canyon, with foliage, buildings, and other obstructions causing multipath. The route started behind the previously referenced Woltosz Engineering Research Building and proceeded on the streets labeled in Figure 4.8.

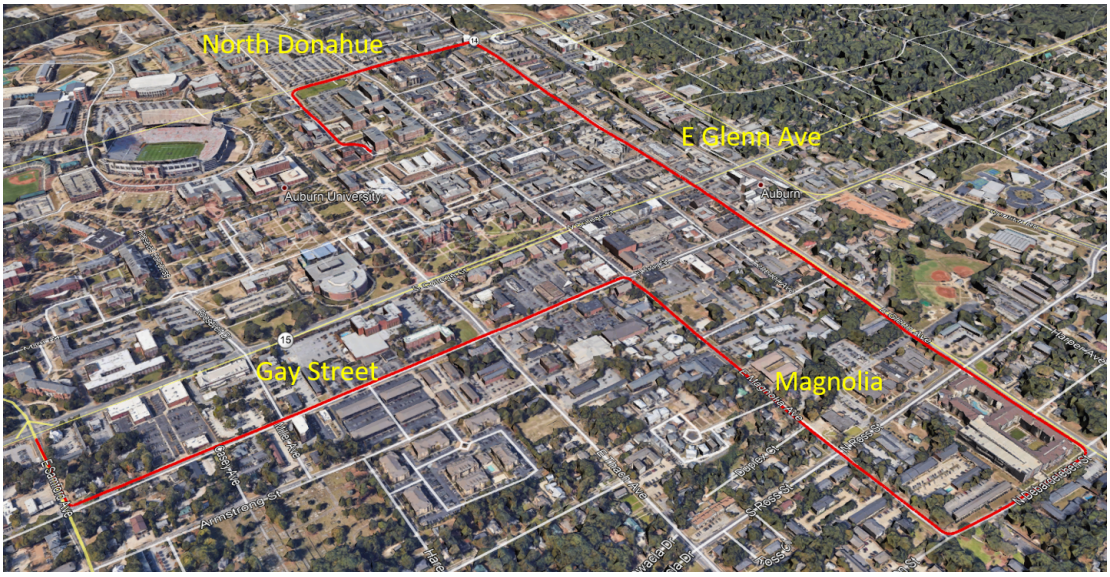

Figure 4.8: Dynamic Route Recorded by Ublox EVK-M8T

Figure 4.9 shows the dynamic C/N_0 estimation results for GPS SV 6. There is a large amount of multipath at approximately the 10 second mark. There is also another drop in the C/N_0 at the 125 second mark. All methods were able to track these outages effectively, which proved the filters used to filter noise variance and signal power were appropriately tuned.

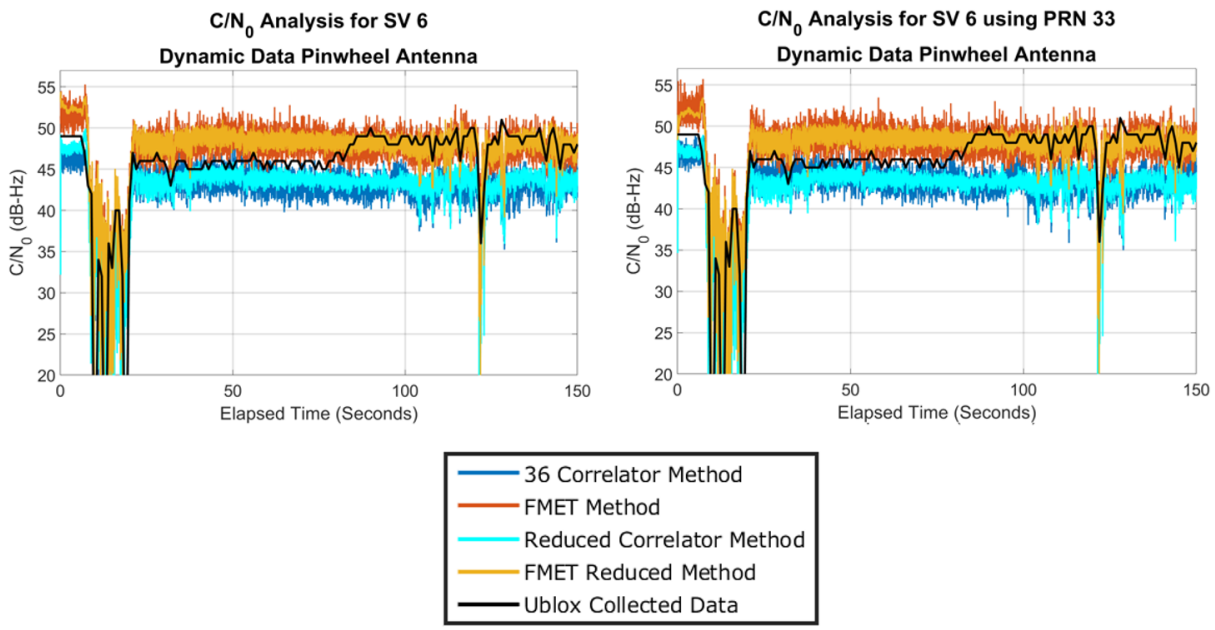

Figure 4.9: Dynamic PRN 6 C/N_0 Results (Left) and Dynamic PRN 33 C/N_0 Results (Right)

GPS SV 17 has a lower elevation angle than GPS SV 6, which is evident by the high number of outages in Figure 4.10. There is a failure to estimate C/N_0 in both the 36 correlator method and the reduced correlator method between the 30 and 95 second mark of the data. Both FMET and FMET reduced method are able to maintain accurate C/N_0 estimates during that time period.

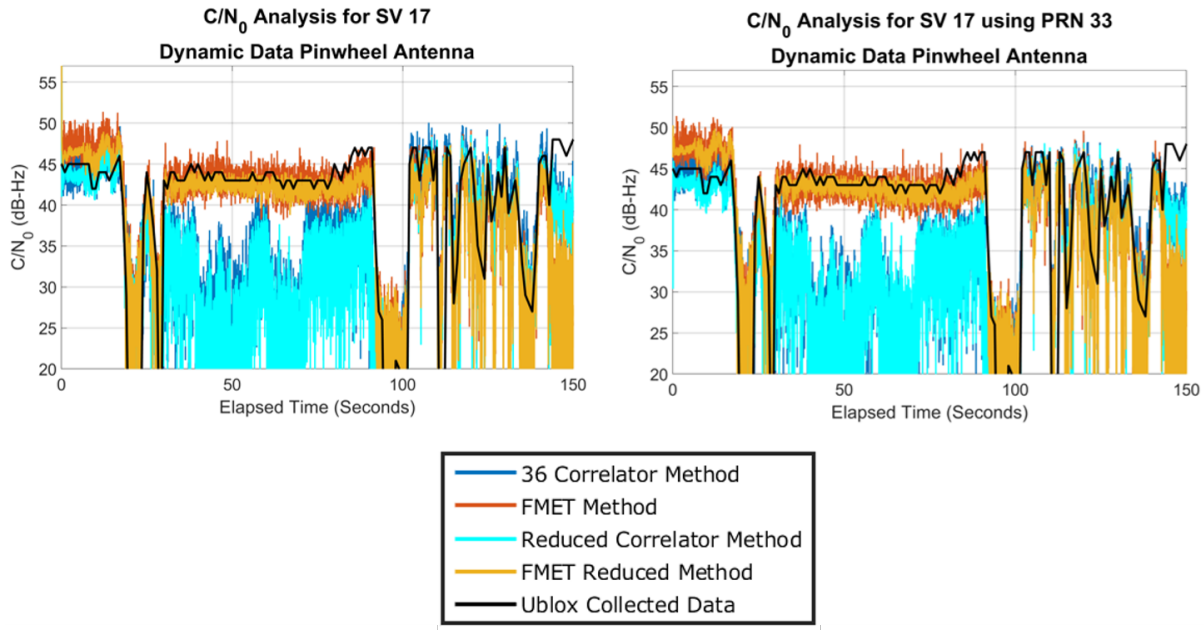

Figure 4.10: Dynamic PRN 17 C/N_0 Results (Left) and Dynamic PRN 33 C/N_0 Results (Right)

Figure 4.11 shows the results from GPS SV 19. All methods are able to track the outages caused by the obstructions and both the FMET and FMET Reduced Method showed small biases in C/N_0.

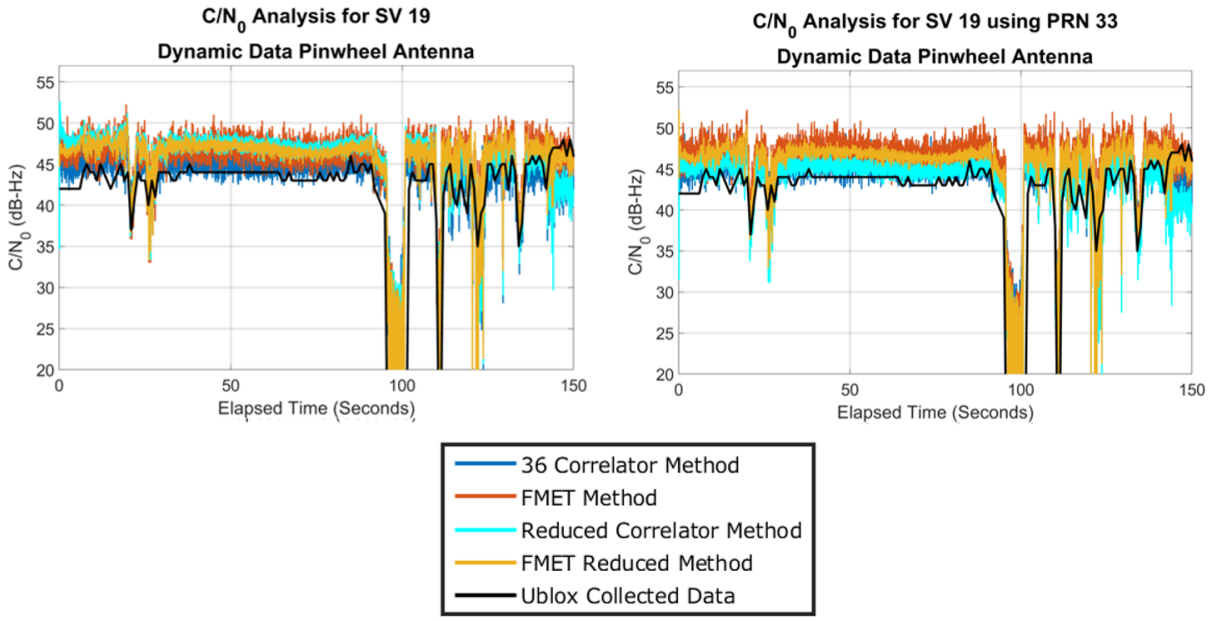

Figure 4.11: Dynamic PRN 19 C/N_0 Results (Left) and Dynamic PRN 33 C/N_0 Results (Right)

4.4 Conclusions

As seen by this chapter, the low SWaP-C correlator C/N_0 methods performed well compared to the widely used 36 correlator method. There were also instances where the low SWaP-C methods outperformed the 36 correlator method as seen by Figure 4.10. Also, there were very few instances where there was a large difference between using PRN 33 and the dedicated PRNs for noise correlator generation. Based on this and the fact that the low SWaP-C methods work well, the method used in this thesis is the FMET reduced correlator method using PRN 33 for C/N_0 estimation. The method is responsive to dynamics and not computationally expensive. Though the method does feature some biases from the true C/N_0 ($+/-$ 2 dB), the method is able to track dynamics and recover from dynamic changes. For implementation in this thesis, the estimated C/N_0 is used to calculate pseudorange and pseudorange rate variances that are used to update a measurement covariance for an extended Kalman filter (EKF).

Chapter 5

GPS L1 C/A Vector Tracking Receiver

Most GNSS receivers use scalar tracking to track code and carrier frequencies, creating signal replicas that are passed to an external navigation processor. Vector tracking is a more integrated approach where the navigation processor is embedded in the tracking loop with an EKF. Numerous software defined receivers have effectively utilized vector tracking in their receiver design [3], [15], [25], [39], [49]–[52]. Vector tracking has been proven to improve receiver robustness by using its ability to track through noisy signal conditions [17], [18]. In this work, vector tracking was selected to add additional robustness to the beamforming receiver.

This chapter is laid out as follows. Section 5.1 defines the vector tracking architecture. Section 5.2 introduces the EKF used in the vector tracking navigation processor. Section 5.3 shows how to implement an asynchronous vector tracking loop. Section 5.4 introduces a tool for determining vector lock characteristics, which is used later to analyze the robustness of the vector tracking receiver.

5.1 Vector Delay/Frequency Lock Loop

The vector delay/frequency lock loop (VDFLL) is a combination of a vector delay lock loop (VDLL) and vector frequency lock loop (VFLL). The combination of the VDLL and VFLL were first shown in [16]. For comparison purposes, a block diagram of the architecture of a scalar tracking receiver, previously derived in Chapter 3, is shown in Figure 5.1. A vector tracking receiver has similar architecture to a scalar tracking receiver. The vector tracking receiver still generates discriminators from correlators originating from a code and carrier replica. However, the vector tracking loop does not include loop filters. Instead,

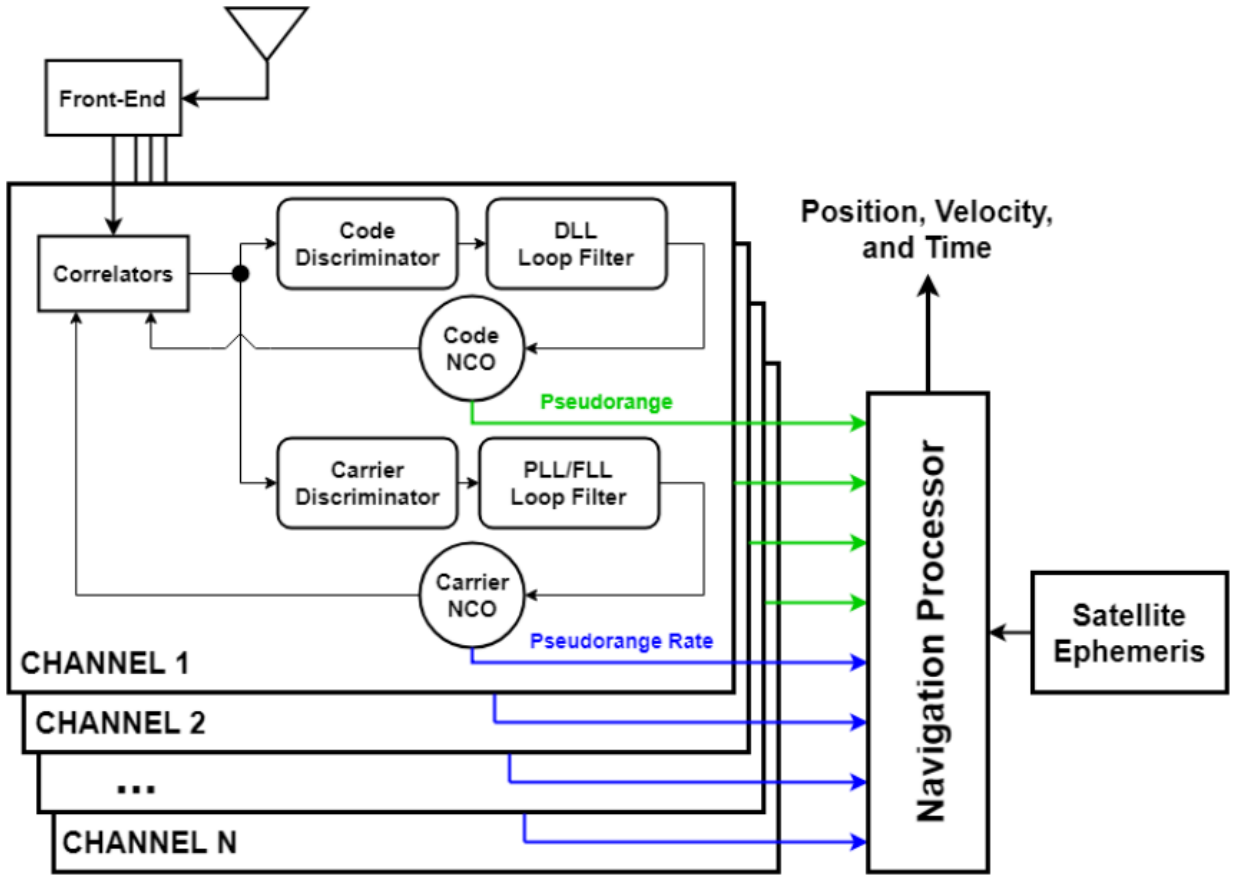

Figure 5.1: Scalar Tracking Receiver Architecture [39]

discriminators are used to calculate range and range rate error terms that are inputs to the extended Kalman filter navigation processor. The EKF then updates the carrier and code replicas to generate new correlators. Instead of the navigation processor being a separate entity, the navigation processor is coupled with signal tracking in the VDFLL. The vector tracking architecture is shown in Figure 5.2. Sections 5.1.1 and 5.1.2 show the how the EKF connects the discriminators and the NCO updates.

5.1.1 Range and Range Rate Error

As stated before in Section 3.4.1, the estimated pseudorange and estimated pseudorange rate equations are shown below in Equations (5.1) and (5.2). These equations are used with the estimated Kalman state vectors to generate estimated pseudorange and estimated

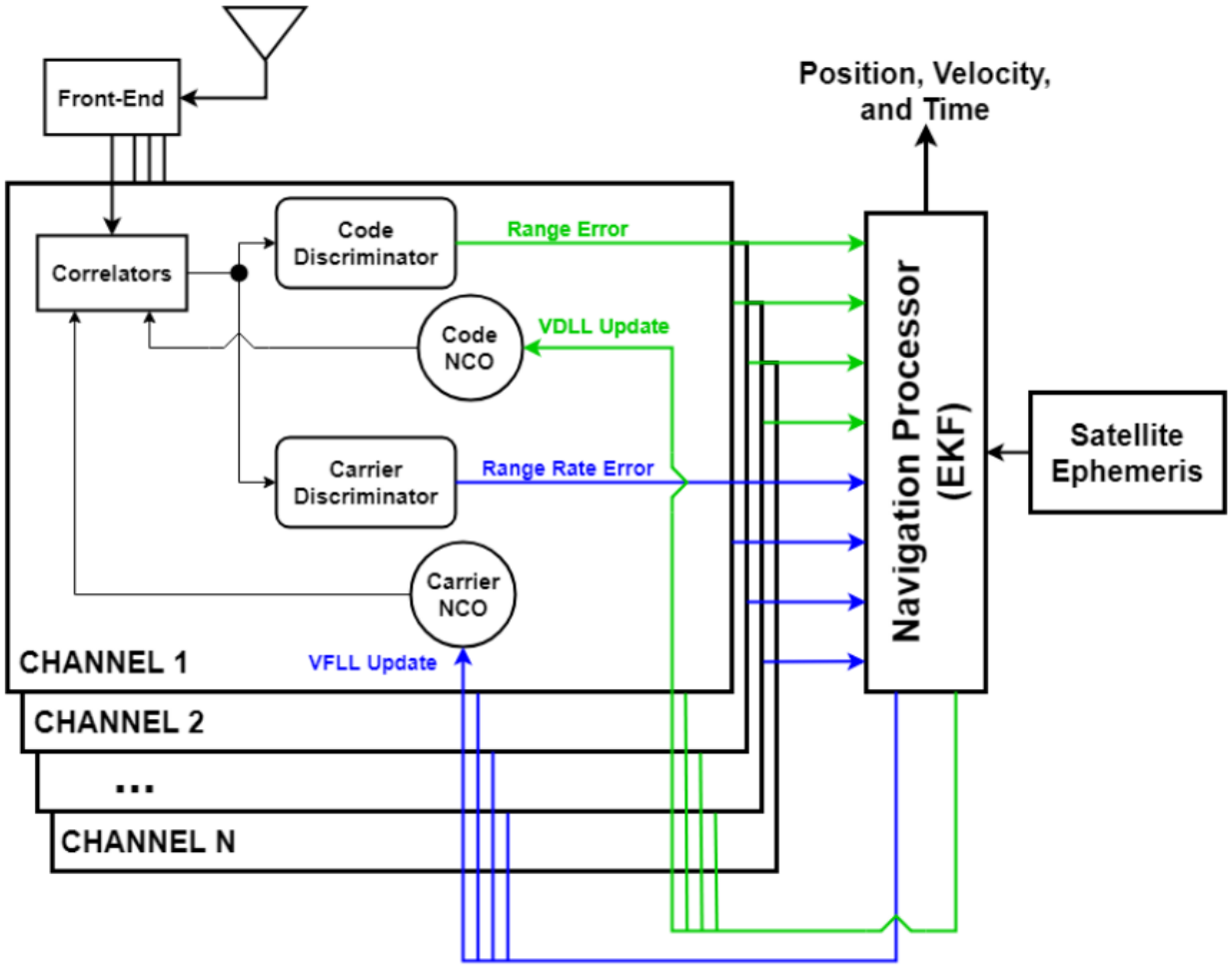

Figure 5.2: Vector Tracking Receiver Architecture [39]

pseudorange rates at each Kalman propagation step.

$$\hat{\rho} = \sqrt{(x_{sv} - \hat{x}_r)^2 + (y_{sv} - \hat{y}_r)^2 + (z_{sv} - \hat{z}_r)^2} + c\hat{b}_r \tag{5.1}$$

$$\hat{\dot{\rho}} = a_x(\dot{x}_{sv} - \hat{\dot{x}}_r) + a_y(\dot{y}_{sv} - \hat{\dot{y}}_r) + a_z(\dot{z}_{sv} - \hat{\dot{z}}_r) + c\hat{\dot{b}}_r \tag{5.2}$$

x_{sv} and $\dot{x}_{sv}$ are the satellite position and velocity in the ECEF x-direction. $\hat{x}_r$ and $\hat{\dot{x}}_r$ are the estimated receiver position and velocity in the ECEF x-direction. The same principles hold true for the y and z-directions. $\hat{b}_r$ and $\hat{\dot{b}}_r$ are the receiver estimated clock bias and clock drift. a_x, a_y, and a_z are the satellite LOS unit vectors.

The measured pseudoranges are generated from the arrival times of the signals, which can be found from the preamble locations in the signal. Equation (5.3) shows how the measured pseudorange is calculated.

$$\tilde{\rho} = c(t_r - t_t) \tag{5.3}$$

t_r is the signal received time, t_t is the signal transmit time, and c is the speed of light in a vacuum. Equation (5.4) shows how the measured pseudorange rate is calculated.

$$\tilde{\dot{\rho}} = -\frac{cf_D}{f_{L1}} \tag{5.4}$$

f_D is the corresponding channel Doppler frequency. Since the signals are mixed to baseband, $f_{DOP} = f_{carr}$. c is again the speed of light in a vacuum. f_{L1} is the frequency of GPS L1. As stated before, the discriminators are used to determine pseudorange error, $\delta\rho$, and pseudorange rate error, $\delta\dot{\rho}$, which are the measurements for the EKF. Equation (5.5) shows the derivation for pseudorange error.

$$\delta\rho = \lambda_{chip}\phi_{DLL} + \tilde{\rho} - \hat{\rho} \tag{5.5}$$

ϕ_{DLL} is the DLL discriminator, and λ_{chip} is the PRN chip width. In theory, the discriminator term should be the only term needed to determine the pseudorange error. However, better results are obtained by adding a residual correction $(\tilde{\rho} - \hat{\rho})$ [39]. Note that the DLL discriminator comes before the Kalman propagation. The residual term comes from the Kalman filters prediction (a priori) step, i.e., both the measured and estimated pseudoranges come from the new Kalman propagated location. The equation for pseudorange rate error is shown in Equation (5.6).

$$\delta\dot{\rho} = -\lambda_{wave}\phi_{FLL} + \tilde{\dot{\rho}} - \hat{\dot{\rho}} \tag{5.6}$$

λ_{wave} is the carrier wavelength, ϕ_{FLL} is the FLL discriminator, and $(\tilde{\dot{\rho}} - \hat{\dot{\rho}})$ is the pseudorange rate residual. The same Kalman calculation timing logic that applies to Equation (5.5) applies to Equation (5.6).

5.1.2 NCO Update

Another non-trivial aspect of asynchronous vector tracking is the NCO update. With the method used in this thesis, the code frequency is updated using a pseudorange estimate that is propagated an integration period beyond the propagation length of the Kalman filter, as shown by Equation (5.7).

$$f_{code_{k+1}} = f_{chip} - \frac{\hat{\tilde{\rho}}_{k+1} - \tilde{\rho}_k}{\lambda_{chip} T} \tag{5.7}$$

$\tilde{\rho}_k$ is the current measurement pseudorange calculated at the end of the Kalman propagation. $\hat{\tilde{\rho}}_{k+1}$ is the measurement estimate of the pseudorange propagated forward ahead of the Kalman filter by one integration period. T is the integration period, f_{chip} is the chipping rate, and λ_{chip} is the chip width.

The carrier frequency NCO update is more direct than the code update. Equation (5.8) shows the carrier NCO update.

$$f_{carr_{k+1}} = f_{IF} - \frac{\hat{\tilde{\dot{\rho}}}_{k+1}}{\lambda_{wave}} \tag{5.8}$$

$\hat{\tilde{\dot{\rho}}}_{k+1}$ is the pseudorange rate measurement propagated one integration period ahead of the Kalman filter, and λ_{wave} is the carrier wavelength of GPS L1 C/A.

5.2 VDFLL Extended Kalman Filter

The VDFLL uses an extended Kalman filter (EKF) to update the receiver PVT. For this work, a position-state VDFLL is used for PVT estimation. Equation (5.9) shows the

Kalman state vector.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} & \hat{\dot{x}} & \hat{y} & \hat{\dot{y}} & \hat{z} & \hat{\dot{z}} & c\hat{b} & c\hat{\dot{b}} \end{bmatrix}^T \tag{5.9}$$

The first six states are the estimated positions and velocities in the x, y, and z directions in the earth centered earth fixed (ECEF) reference frame. $c\hat{b}$ and $c\hat{\dot{b}}$ are the estimated clock bias and clock drift terms, with c being the speed of light in a vacuum. Equation (5.10) shows the discrete state transition matrix A. The state transition matrix changes depending on the Kalman propagation length δT. This is due to the asynchronous implementation of the VDFLL, which is explained in Section 5.3.

$$A = \begin{bmatrix} 1 & \delta T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \delta T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \delta T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \delta T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.10}$$

The measurement observation matrix H is shown in Equation (5.11).

$$H = \begin{bmatrix} -a_x^{(j)} & 0 & -a_y^{(j)} & 0 & -a_z^{(j)} & 0 & 1 & 0 \\ 0 & -a_x^{(j)} & 0 & -a_y^{(j)} & 0 & -a_z^{(j)} & 0 & 1 \end{bmatrix} \tag{5.11}$$

The unit vector corresponding to the channel measurement matching the end of the Kalman propagation is used to populate (5.11). Similarly, the current pseudorange error $\delta\rho^{(i)}$ and pseudorange rate error $\delta\dot{\rho}^{(i)}$ are used to populate the measurement vector $\mathbf{y}$.

$$\mathbf{y} = \begin{bmatrix} \delta\rho^{(j)} \\ \delta\dot{\rho}^{(j)} \end{bmatrix} \tag{5.12}$$

For tuning, the discrete process noise covariance matrix Q represents uncertainty in the velocity model, which is represented by the state transition matrix. Q is shown in Equation (5.13).

$$Q = \begin{bmatrix} Q_{PV} & 0 & 0 & 0 \\ 0 & Q_{PV} & 0 & 0 \\ 0 & 0 & Q_{PV} & 0 \\ 0 & 0 & 0 & Q_c \end{bmatrix} \tag{5.13a}$$

$$Q_{PV} = \begin{bmatrix} \sigma_{PV}^2 \frac{\delta T^3}{3} & \sigma_{PV}^2 \frac{\delta T^2}{2} \\ \sigma_{PV}^2 \frac{\delta T^2}{2} & \sigma_{PV}^2 \delta T \end{bmatrix} \tag{5.13b}$$

$$Q_c = \begin{bmatrix} \sigma_b^2 \delta T + \sigma_d^2 \frac{\delta T^3}{3} & \sigma_d^2 \frac{\delta T^2}{2} \\ \sigma_d^2 \frac{\delta T^2}{2} & \sigma_d^2 \delta T \end{bmatrix} \tag{5.13c}$$

$$\sigma_b^2 = c^2 \frac{h_0}{2} \tag{5.13d}$$

$$\sigma_d^2 = c^2 2\pi^2 h_{-2} \tag{5.13e}$$

T is the tracking integration period, σ_{PV}^2 is the position variance, and σ_b^2 and σ_d^2 are the clock bias and clock drift variances. h_0 and h_{-2} are the power law coefficients found in [53]. Since the USRP used to record data contains a temperature compensated crystal oscillator (TCXO), h_0 is set to 2×10^{-19}, and h_{-2} is set to 2×10^{-20}. Since GPS accuracy is typically around five meters and the receiver in this work improves accuracy with multiple antennas, the position variance σ_{PV}^2 is set to 2 meters^2. Equation (5.14) shows the measurement

covariance matrix R, which is a function of range variance $\sigma^2_{\delta\rho}$ and range rate variance $\sigma^2_{\delta\dot{\rho}}$.

$$R = \begin{bmatrix} \sigma^{2\ (j)}_{\delta\rho} & 0 \\ 0 & \sigma^{2\ (j)}_{\delta\dot{\rho}} \end{bmatrix} \tag{5.14a}$$

$$\sigma^2_{\delta\rho} = \frac{\lambda^2_{chip}}{2\left(\delta T \cdot 10^{(C/N_0)/10}\right)^2} + \frac{\lambda^2_{chip}}{4(\delta T) \cdot 10^{(C/N_0)/10}} \tag{5.14b}$$

$$\sigma^2_{\delta\dot{\rho}} = \left(\frac{c}{\pi f_{L1}(\delta T)}\right)^2 \left(\frac{2}{\left(\delta T \cdot 10^{(C/N_0)/10}\right)^2} + \frac{2}{\delta T \cdot 10^{(C/N_0)/10}}\right) \tag{5.14c}$$

C/N_0 in Equations (5.14b) and (5.14c) is the carrier-to-noise ratio in dB-Hz. Like before, δT is the propagation length of the EKF.

5.3 Asynchronous Timing

Timing synchronization is arguably the most challenging component of implementing vector tracking. Vector tracking loops can be implemented in a synchronized batch [39] or the loops can be implemented asynchronously [17]. The asynchronous implementation better models the time difference of arrival (TDOA) nature of the GPS signals and yields better performance when appropriately tuned.

As stated earlier, vector tracking must be initialized with scalar tracking PVT. In scalar tracking, each channel discriminator is calculated at the same time instance, and those calculations are passed to the navigation processor where pseudoranges are estimated at the same time instance. Vector tracking improves this process by obtaining more accurate pseudorange estimates at the actual arrival times of each channel. The common receive time simplification is not used. This reduces error in the pseudorange estimates and allows for a more precise position estimates. Figure 5.3 shows how asynchronous timing works. When the first channel has obtained a position solution, the channel is immediately propagated a full integration period (20 ms) and a pseudorange estimate is calculated. The extended Kalman filter then estimates a position at the midpoint of the integration period, as shown by the black diamonds. Once the next channel is active, the Kalman filter is propagated to

the end of its integration period. This process continues until all channels are active, and then the process looks like that shown in Figure 5.3.

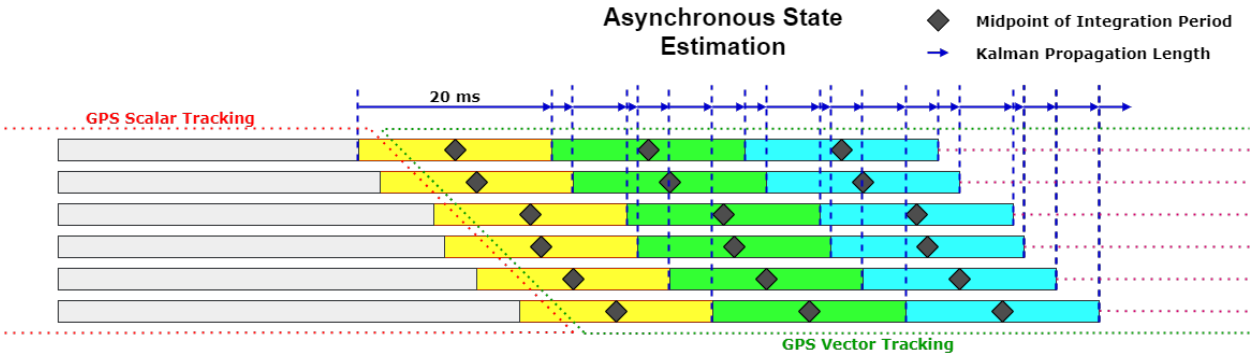

Figure 5.3: Vector Asynchronous Timing Update

As noted previously, the black diamonds represent the midpoint of the integration period. The measurements from each channel calculated at the end of the integration period most closely resemble the position state at the midpoint of the integration period [51]. The blue arrows represent each propagation step. Ensuring the correct propagation lengths is crucial for operating an asynchronous vector tracking loop. The Kalman propagation lengths can be determined by each of the channel preamble locations and the number of samples read into the receiver.

Recall the code frequency NCO update in Equation (5.7). To further explain the NCO update in an asynchronous implementation, suppose only two channels are vector tracking while the other channels are still initializing in scalar tracking. Figure 5.4 shows the code frequency estimation. The key in the upper right explains each step, with the discrete time steps legend key being in the order the steps are implemented. Building off of Figure 5.3, the first channel propagates 20 milliseconds at initialization (shown by the red straight line in Figure 5.4) and a correction is applied, which gives a range measurement $\tilde{\rho}_k^{(1)}$ for channel one. Another propagation is then applied, denoted NCO Update Propagation, that gives an estimate for the next range measurement $\hat{\tilde{\rho}}_{k+1}^{(1)}$. There is no correction for this estimate because the estimate is propagated 20 milliseconds ahead of the current measurement. The

future estimate and the current measurement are then used to update the code frequency, which is best represented at the midpoint of the Kalman propagation denoted by the black diamonds. If only two channels are in vector tracking mode, the process looks like Figure 5.4 with the process following the red, pink, blue, orange, etc. steps. However, the more complicated full process when all channels are activated in vector tracking is not represented Figure 5.4.

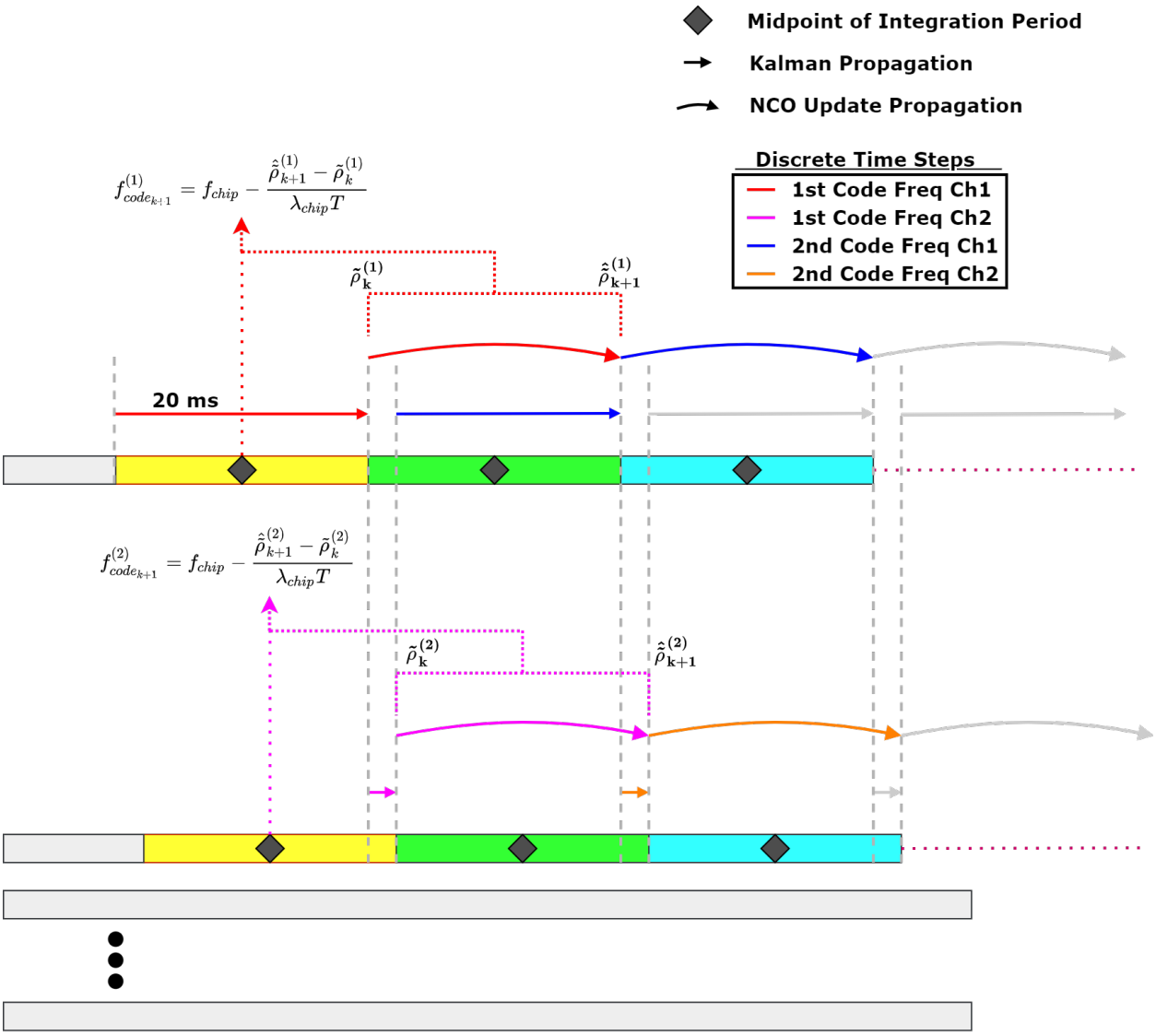

Figure 5.4: Asynchronous Propagation Steps for NCO Update

With more channels, therein lies the difficulty in vector tracking. Visualization of this more difficult implementation is left as an exercise for the reader.

The benefits of asynchronous timing are the ability to obtain more precise position results in certain situations, and, especially for this work, the ability to take full advantage of the arrival time of the satellite signals.

5.4 Maximum Range Variance

The reason vector tracking is used in this thesis is the fact that vector tracking can maintain lock through outages in the GPS L1 C/A signal and maintain tracking at lower C/N_0. On the contrary, a scalar tracking receiver will lose lock, and it takes time re-acquire and track. Vector tracking having higher sustainability through outages than scalar tracking is well known and researched [3], [18]. The research was expanded with work by Benjamin Clark from Auburn University on the uncertainty of vector tracking and vector lock [52]. The uncertainty in vector tracking is difficult to define due to the asynchronous coupling of channels. One channel can fail, but the rest of the channels have the ability to maintain a non-diverged filter. This section introduces Clark's analytical maximum range variance. The analytical maximum range variance will then be used to analyze results from the vector tracking beamsteering implementation.

5.4.1 Analytical Maximum Range Variance [52]

The state covariance matrix, $\mathbf{P}$, is an estimate of the vector tracking algorithm uncertainty. The pseudorange variance is derived from $\mathbf{P}$, and the derivation is shown in Equations

(5.15) to (5.19). The states are transformed using the Equation (5.15) for ease of calculation.

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.15}$$

Equation (5.16) shows the resulting state vector. Note the states are now sorted with the position related states and the velocity related states.

$$\mathbf{Sx} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta cb \\ \delta \dot{x} \\ \delta \dot{y} \\ \delta \dot{z} \\ \delta \dot{cb} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_{rr} \end{bmatrix} \tag{5.16}$$

The same operation with the transformation matrix can be conducted with the state covariance matrix, $\mathbf{P}$. The operation is shown below in Equation (5.17).

$$\mathbf{SPS}^T = \begin{bmatrix} \mathbf{P}_r & \mathbf{P}_{r\times rr} \\ \mathbf{P}_{r\times rr} & \mathbf{P}_{rr} \end{bmatrix} \tag{5.17}$$

The sub-matrices are defined as follows: $\mathbf{P}_r$ is the pseudorange state covariances, $\mathbf{P}_{rr}$ is the pseudorange rate state covariances, and $\mathbf{P}_{r\times rr}$ is the cross-covariances. $\mathbf{P}_r$ can be defined by Equations (5.18).

$$\mathbf{P}_r = \begin{bmatrix} \mathbf{P}_p & \rho_c \\ \rho_c^T & \sigma_{cb}^2 \end{bmatrix} \tag{5.18a}$$

$$\mathbf{P}_p = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y & \rho_{xz}\sigma_x\sigma_z \\ \rho_{xy}\sigma_x\sigma_y & \sigma_y^2 & \rho_{yz}\sigma_y\sigma_z \\ \rho_{xz}\sigma_x\sigma_z & \rho_{yz}\sigma_y\sigma_z & \sigma_z^2 \end{bmatrix} \tag{5.18b}$$

$$\rho_c = \begin{bmatrix} \rho_{xc}\sigma_x\sigma_{cb} \\ \rho_{yc}\sigma_y\sigma_{cb} \\ \rho_{zc}\sigma_z\sigma_{cb} \end{bmatrix}, \tag{5.18c}$$

$\mathbf{P}_p$ is the position covariance, ρ_c is the clock cross-correlations, and σ_c^2 is the clock variance. The sub-matrix $\mathbf{P}_p$ can be separated into a matrix of its eigenvectors $\mathbf{U}$ and a matrix of its corresponding eigenvalues Λ. The largest eigenvalue, λ_{max}, corresponds to the highest variance along a single LOS unit vector. u_{max} is the corresponding eigenvector to the largest eigenvalue λ_{max}. Therefore, [52] shows that the maximum range variance can be calculated by Equation (5.19) below.

$$\begin{aligned} \sigma_r^2 &= \begin{bmatrix} u_{max}^T & 1 \end{bmatrix} \begin{bmatrix} P_p & \rho_c \\ \rho_c^T & \sigma_{cb}^2 \end{bmatrix} \begin{bmatrix} u_{max} \\ 1 \end{bmatrix} \\ &= u_{max}^T P_p u_{max} + 2\rho_c^T u_{max} + \sigma_{cb}^2 \\ &= \lambda_{max} + 2\rho_c^T u_{max} + \sigma_{cb}^2 \end{aligned} \tag{5.19}$$

Clark's findings showed that the filter diverges when the 1σ range uncertainty exceeds $1/6^{\text{th}}$ of a chip, where the chip width of GPS L1 C/A is approximately 293 meters. This means the vector tracking loop is considered unrecoverable if the 1σ range uncertainty exceeds

approximately 50 meters. Note that the vector tracking loop does not diverge if 4 or more satellites are visible. The loop can propagate other ranges forward in time and is able to maintain lock.

5.5 Conclusions

In this chapter, the vector tracking receiver was introduced. The combined VDLL and VFLL or VDFLL was explained in detail. In the VDFLL implementation, carrier phase is not tracked because there is not a PLL, which makes the VDFLL more robust to noise. Initial navigation data is obtained from the scalar tracking PLL and is used during the vector tracking beamsteering process for satellite positions. There are vector tracking implementations with PLLs, such as a PLL aided vector tracking loop [39] and a fully integrated VDLL VPLL [50]. The reasons why a user would want carrier phase is for a GNSS attitude estimate [54], real-time kinematic (RTK) positioning, and navigation data updates. This work assumes attitude is obtained from an external source and the scenarios do not need navigation updates. The VDFLL is the appropriate design for this work due to the assumptions outlined above as well as its robustness to noise. This chapter also explained asynchronous timing, which is required to overcome assumptions from the batched implementation. Also, the maximum range variance analysis tool was introduced to define vector lock.

Chapter 6

Multi-Antenna Vector Tracking Beamsteering

Vector tracking and beamsteering have individually been shown to improve receiver performance [1], [18] in the face of multipath, jamming, etc. With the GPS L1 C/A signal being at such low power, any improvements to C/N_0 carry much merit in receiver design. In 2016, the German Aerospace Center (DLR) presented a combined vector tracking beamsteering receiver architecture [20] that implemented an adaptive eigenspace mitigation method similar to that of [55] in post-correlation. The advantage of post-correlation beamforming is the fact that attitude is not needed because satellite positions are detectable. However, the disadvantages of post-correlation beamforming is a higher computational load on the receiver and a lack of modularity. This thesis couples vector tracking and beamforming before tracking correlators are generated for the vector tracking loop. This reduces the number of correlator channels needed by a factor of the antenna array elements. It also allows for sensor modularity, i.e. the vector tracking loop can be coupled with a NovAtel GAJT or alternative beamforming front-end, and the attitude source can be swapped.

The implementation in this thesis feeds back satellite geometry from the vector tracking navigation processor to create deterministic weights for standard beamsteering. While this method does not actively attempt to protect the receiver from harmful emitters, as in [20], the method presents a simplified and effective implementation of directing more power in the direction of SVs and less power in the direction of the elevation mask, where most undesirable emitters are present. The benefit behind this implementation is simplicity, giving it an edge in low SWaP-C design.

This chapter is laid out as follows. Section 6.1 describes in detail the design of the vector beamsteering receiver. Section 6.2 shows the impact of different array sizes by comparing the

signal improvements from using different array patterns in simulation. Section 6.3 shows how the vector tracking coupling performs in an open sky and multipath environment. Section 6.4 includes the results from injecting interference data on a live sky scenario. Finally, Section 6.5 outlines the conclusions that can be drawn from this analysis.

6.1 Algorithm Architecture

The design of the vector tracking coupled beamsteering algorithm can be defined by three main sections: the vector tracking receiver (Chapter 5), the beamsteering module (Section 6.1.1), and the attitude update (Section 6.1.2). The vector tracking receiver has already been described in detail. The beamsteering algorithm and the attitude update are described in Sections 6.1.1 and 6.1.2.

6.1.1 Deterministic Beamsteering

As described earlier, beamforming is the method of modifying signal phases to obtain a desired radiation pattern on an antenna array. Beamsteering is a subset of beamforming where signals phases are modified to align signals from a desired direction. By aligning the signals in phase, the signals add constructively yielding a higher digital receive power of the cumulative signal. The deterministic algorithm used in this work is described below.

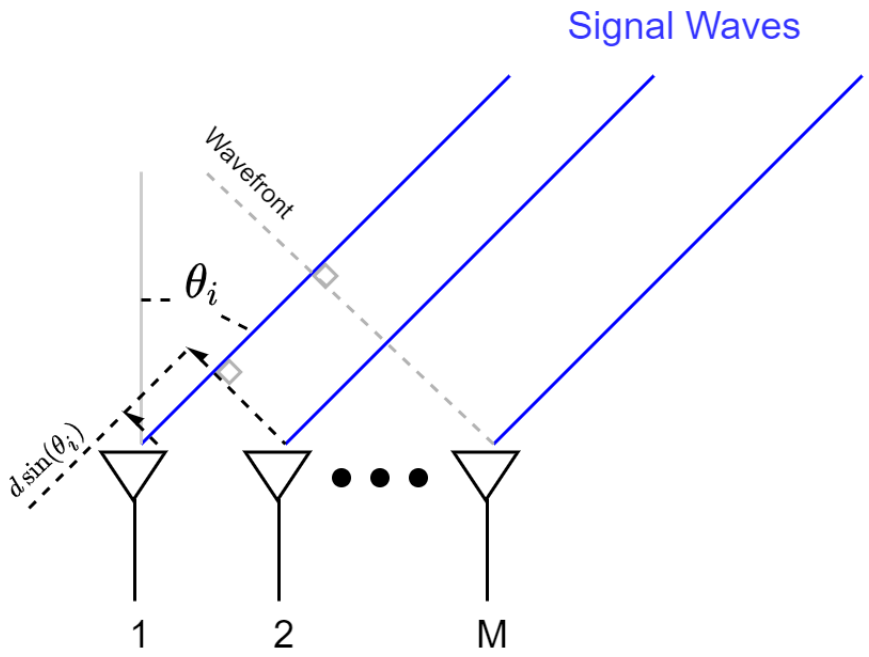

Figure 6.1: Incident Signals on Arbitrary CRPA Array

Using Figure 6.1 as a reference, the desired phase delays (or weights W_m) for the m$^{\text{th}}$ element are shown in Equation (6.1).

$$W_m = e^{-j\frac{2\pi\hat{r}d_m}{\lambda}} \tag{6.1}$$

d_k is the location of each antenna, usually from a selected reference element. λ is the wavelength of the signal. $\hat{r}$ is the boresight vector described in Equation (6.2).

$$\hat{r}\,(Az, El) = \begin{bmatrix} \sin Az \cos El & \cos Az \cos El & \sin El \end{bmatrix} \tag{6.2}$$

The boresight vector is a unit vector directed in the line of bearing given by azimuth and elevation. With the boresight vector as an input to the weight equation, weights can be set to align the antenna beam with a certain azimuth and elevation angle. This simple deterministic beamsteering approach is tested with both a scalar tracking and vector tracking loop implementations for comparison purposes. An example of beamsteering to a signal

at azimuth = 180, elevation = 45 degrees is shown in the top two plots of Figure 6.2. Beamsteering to a signal at azimuth = 320, elevation = 60 degrees is shown in the bottom two plots of Figure 6.2.

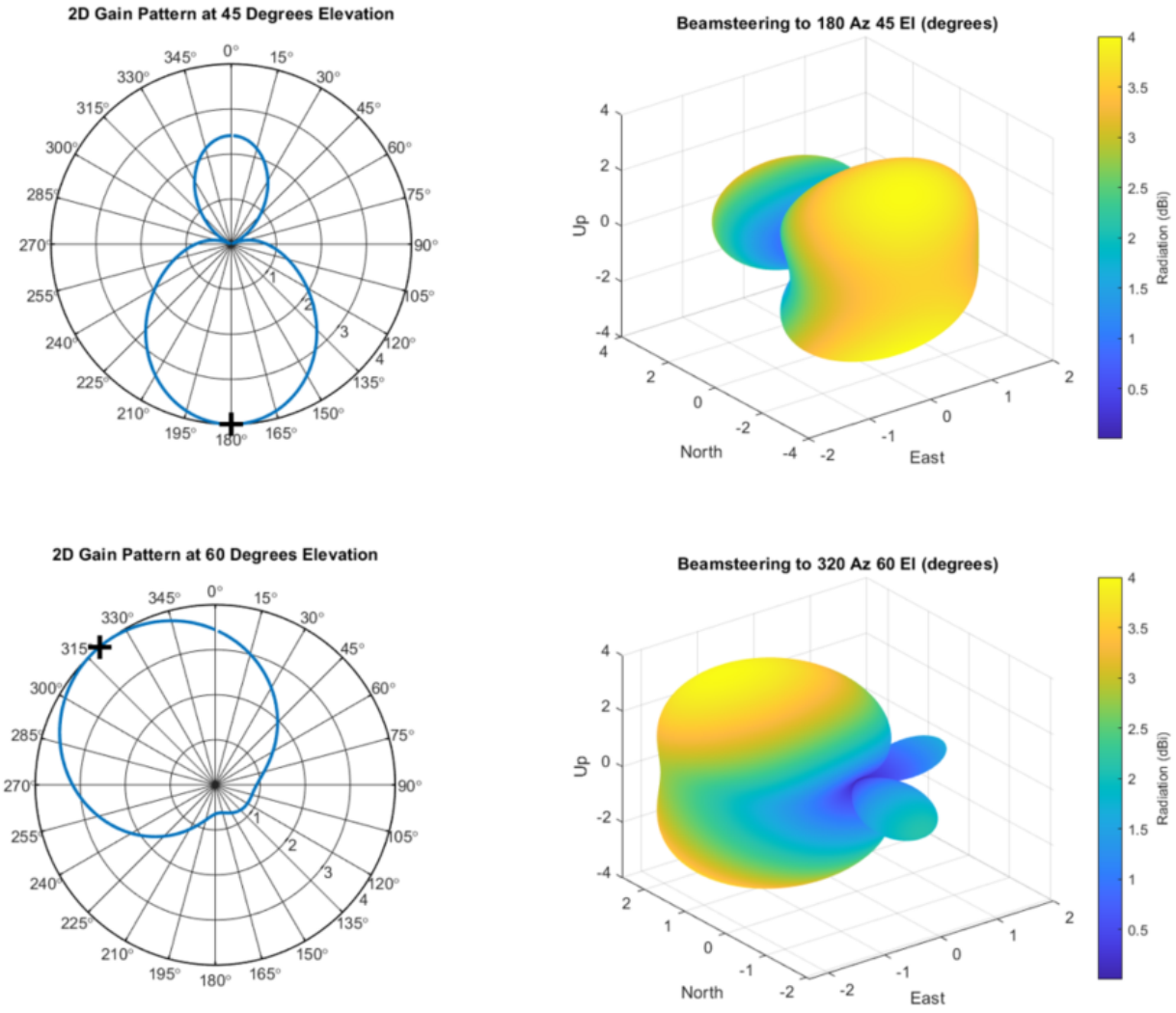

Figure 6.2: 4 Element CRPA Beamsteering ($\frac{\lambda}{2}$ spacing)

Since signals are digitized before the vector tracking loop, beamsteering can be conducted on each tracking channel in parallel. In this form, $N-1$ limitations do not apply. Each tracking channel has four input signal streams that are beamsteered to the single corresponding tracking channel satellite prior to correlator generation.

6.1.2 Course Update

The attitude update is necessary to ensure that the azimuth and elevation angles are accurate in the body frame. Ideally, the user would select an alternative sensor, such as an inertial measurement unit (IMU), to perform the attitude update to improve resilience. However, for this work, attitude is simplified to be course. Since this research is conducted using a ground vehicle on relatively flat terrain in an uncontested environment, this assumption is valid for testing.

The course update was calculated using the velocity output from the navigation processor. For this work, the course was applied from north using Equation (6.3), where v_E is the east component, and v_N is the north component of velocity.

$$\psi_{course} = \arctan\left(\frac{v_E}{v_N}\right) \tag{6.3}$$

Since velocity estimates break down as speed decreases, a course estimate threshold was set to 2 m/s to mitigate false course measurements. This also was accounted for in the data recording process by maintaining a constant course at slow speeds (i.e. not turning the vehicle while driving slow).

6.1.3 Schematic

Figure 6.3 shows an overall block diagram for the multi-antenna vector tracking coupled beamsteering receiver. The receiver initially runs single antenna scalar tracking from a reference element to initialize vector tracking and determine initial estimates of satellite geometry. As soon as the first scalar PVT solution is obtained, the receiver activates single stream vector tracking, which was shown in Figure 5.2. The receiver is given a few seconds to maintain vector lock, characterized by the maximum range variance (Section 5.4), and then the receiver transitions to vector coupled beamsteering.

In the vector coupled beamsteering configuration, all antenna streams are active, and the streams are combined before correlators are generated in vector tracking. Satellite azimuth and elevation angles are estimated in the navigation processor and fed back to update the weights using the boresight vector Equation (6.2) and then Equation (A.1). The weights are then used to make the signals add constructively at the line-of-bearing of the desired SV. Since vector tracking asynchronously processes one SV at a time, the corresponding azimuth and elevations are fed back in a cyclic fashion, aligning with the VDFLL.

Multiple antenna elements can be added to the algorithm with ease based on this design. Section 6.2 shows the advantages of adding multiple antenna elements and the effects the elements have on signal quality.

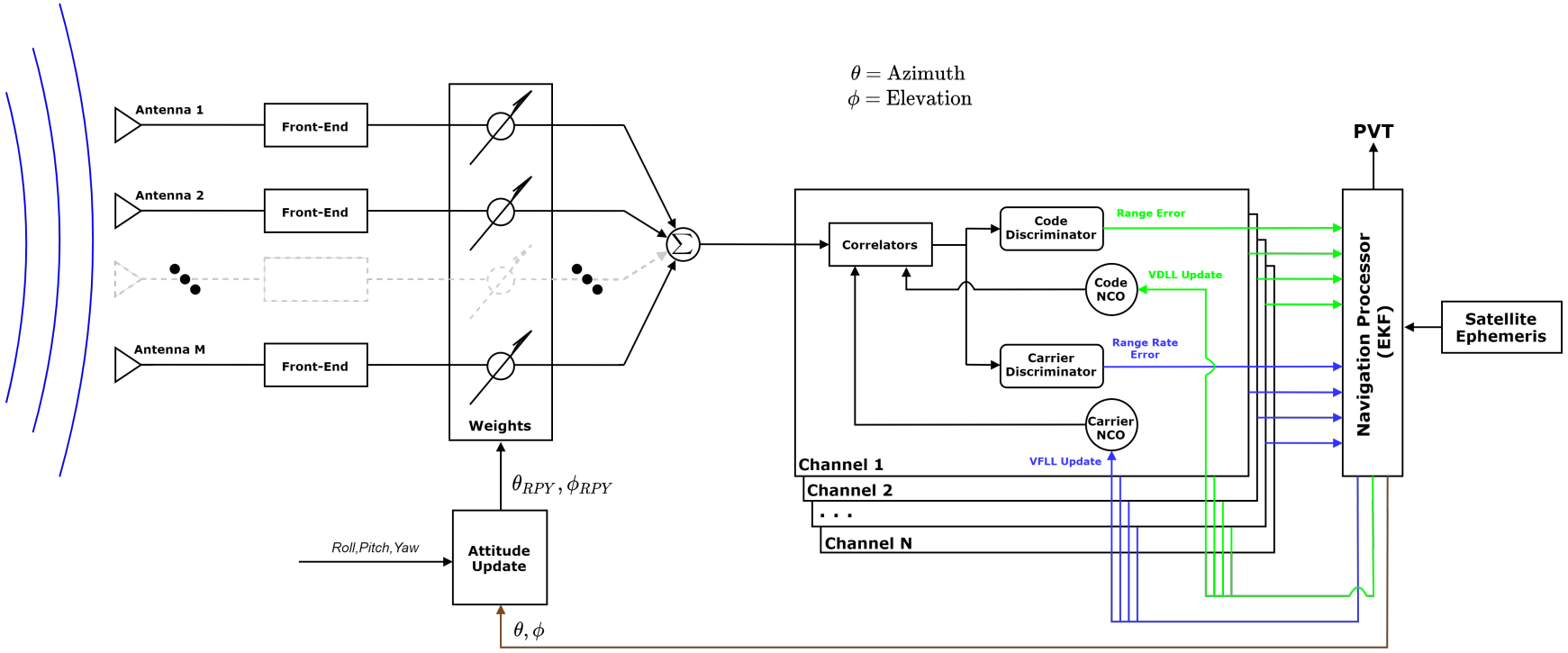

Figure 6.3: Multi-antenna Vector Tracking Beamsteering Receiver

6.2 The Impact of Different Array Sizes

To test the versatility of the multi-antenna receiver, a software GPS signal simulator [56] was used to simulate signal data of different sized arrays. In total, six array geometries were selected for testing. Figure 6.4 shows five of the six geometries that were simulated. Arrays were selected from an Antcom catalog, and the phase centers of the arrays were assumed to be centered on the SMA ports. This thesis does not attempt to make performance claims about Antcom specific arrays.

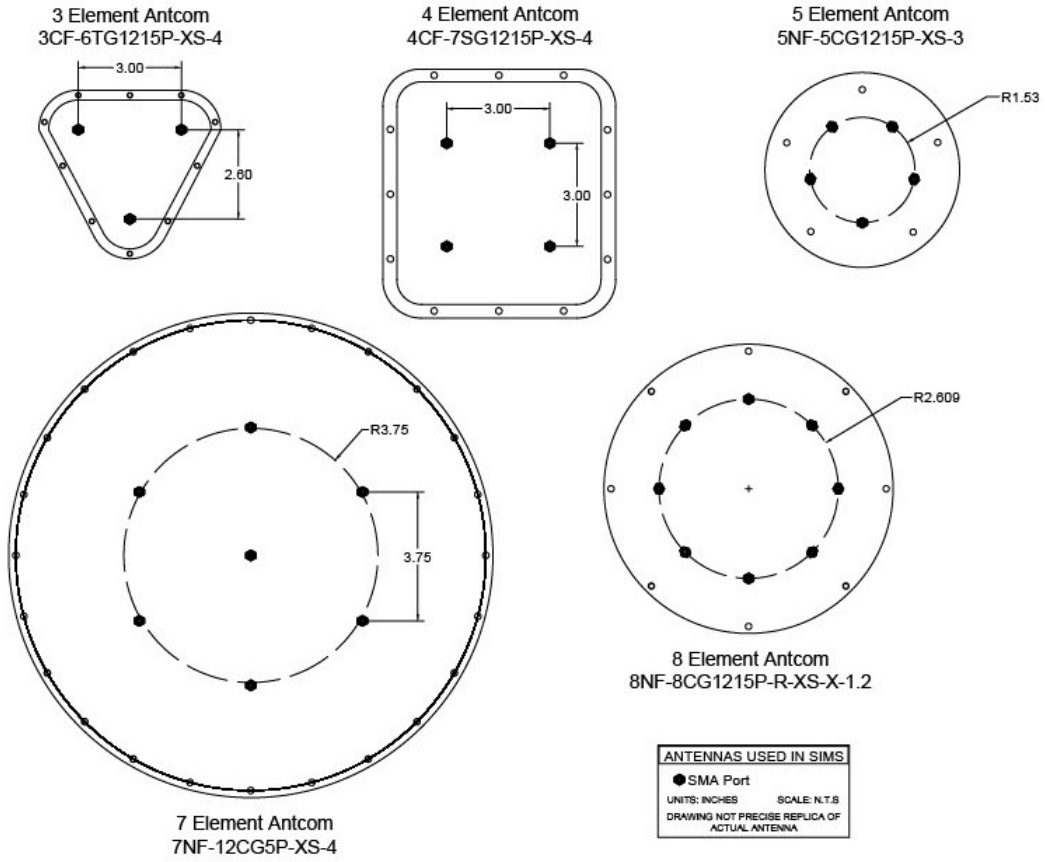

Figure 6.4: Antcom Simulated Array Patterns

In addition to the six arrays shown, a linear two element half wavelength array was also simulated. Section 6.2.1 shows the results from simulating all signals at the same power level. Section 6.2.2 shows how the receiver reacts to a scenario with variable power. A single

antenna scalar tracking receiver, a single antenna vector tracking receiver, and a multi-antenna scalar tracking beamsteering receiver are compared to the proposed multi-antenna vector tracking beamsteering receiver to analyze performance.

6.2.1 Signals Simulated at Same Power

The signals in this section are simulated at constant matching power level (45 dB-Hz C/N_0). With the signals at the same power, the effects of antenna geometry become evident. Figure 6.5 shows the sky plot for the simulated scenario.

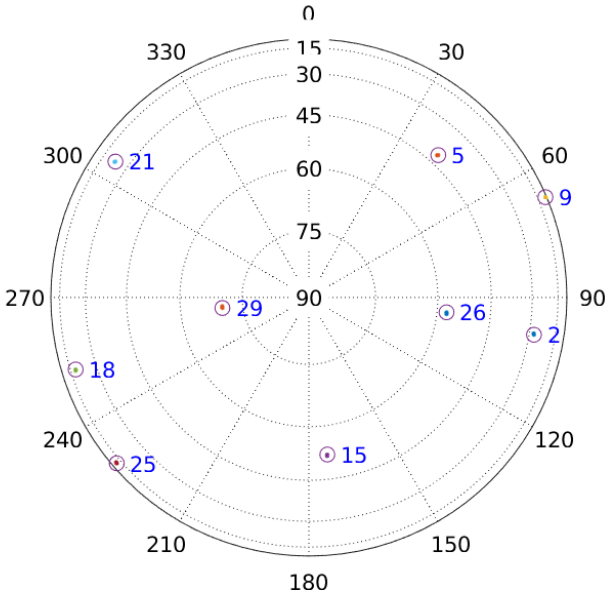

Figure 6.5: Sky Plot for Simulated Scenario

With a PDOP of approximately 1.54, the scenario provides a good measure of the performance of the CRPA arrays. The scenario was also simulated with a stationary receiver to reduce the effects of dynamics on the results. The arrays were oriented north, so the attitude update was unnecessary during the static simulation.

C/N_0 Improvements

C/N_0 is used again as a performance characteristic for the proposed receiver. The C/N_0 is expected to increase once beamsteering is initiated. The signal components add constructively, but the noise does not add constructively if the noise is uncorrelated. In general, if two signals of equal magnitude with uncorrelated white Gaussian noise are added constructively, the signal amplitude doubles while the noise amplitude scales to $\sqrt{2}$. Likewise, if the signal amplitude triples (i.e. three antennas), the noise amplitude scales to $\sqrt{3}$ and so on. This introduces a natural diminishing return in C/N_0 as the number of antenna elements increase, which is seen in plots in this section.

The radiation patterns for each array beamsteered to the highest elevation satellite in the scenario (PRN 29 at el. = 70°) are shown below in Figure 6.6. The units of the radiation pattern are in dB relative to an ideal isotropic antenna (dBi). Note the color scale changing on each plot. As the number of elements increase, the dBi gain value increases. All of the radiation patterns were able to successfully beamsteer to the PRN 29 satellite.

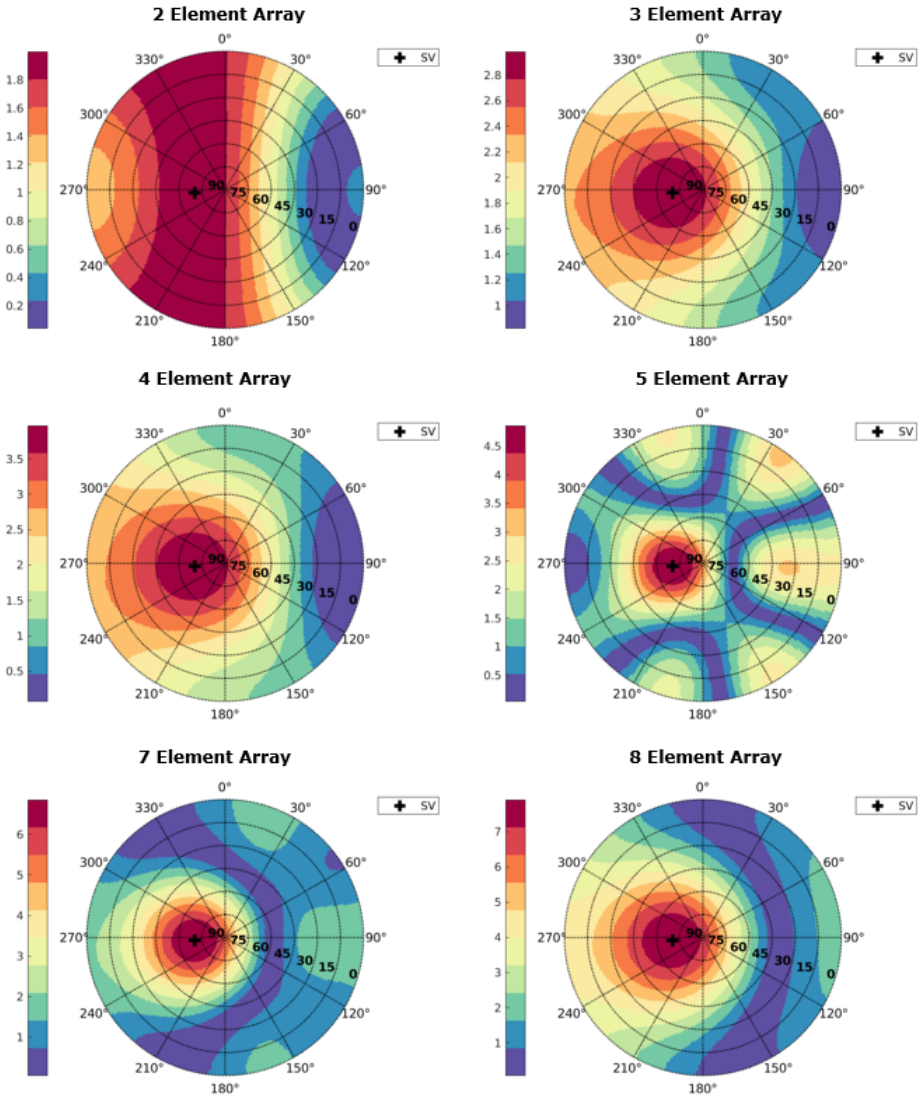

Figure 6.6: Magnitude Radiation Patterns for High Elevation PRN 29 [dBi]

Figure 6.7 shows the C/N_0 of PRN 29 increase in the simulation when beamsteering was turned on at approximately 16 seconds. The mean of C/N_0 over each second is plotted as well as the one sigma bounds of the data. The single element scalar and vector tracking C/N_0 values hold constant at the simulated 45 dB-Hz level. When beamsteering is initiated, the multi-antenna receivers C/N_0 immediately jump to a higher value. For this plot and the proceeding plots in this section, the lightest shaded values correspond to the lowest number of CRPA elements, two. The shading becomes darker as the number of elements

increase. Vector tracking excels in aiding satellites when their signal quality degrades. Since the signal quality is favorable throughout the scenario, vector tracking performs similarly to scalar tracking as expected.

The data shows that the largest jump occurs when the number of antennas is increased from one to two elements. However, immediately after, the jump from two to three elements yields a diminishing return. Surprisingly, the jump from four elements to five elements is larger than some of the previous jumps. This is attributed to the beneficial geometry provided by the array. A whole field of study is devoted to array pattern synthesis where array patterns are geometrically designed to give the optimal results at certain angles. However, the proceeding geometrical array effects are not studied in this work. The five, seven, and eight element CRPAs perform nearly the same for high elevations.

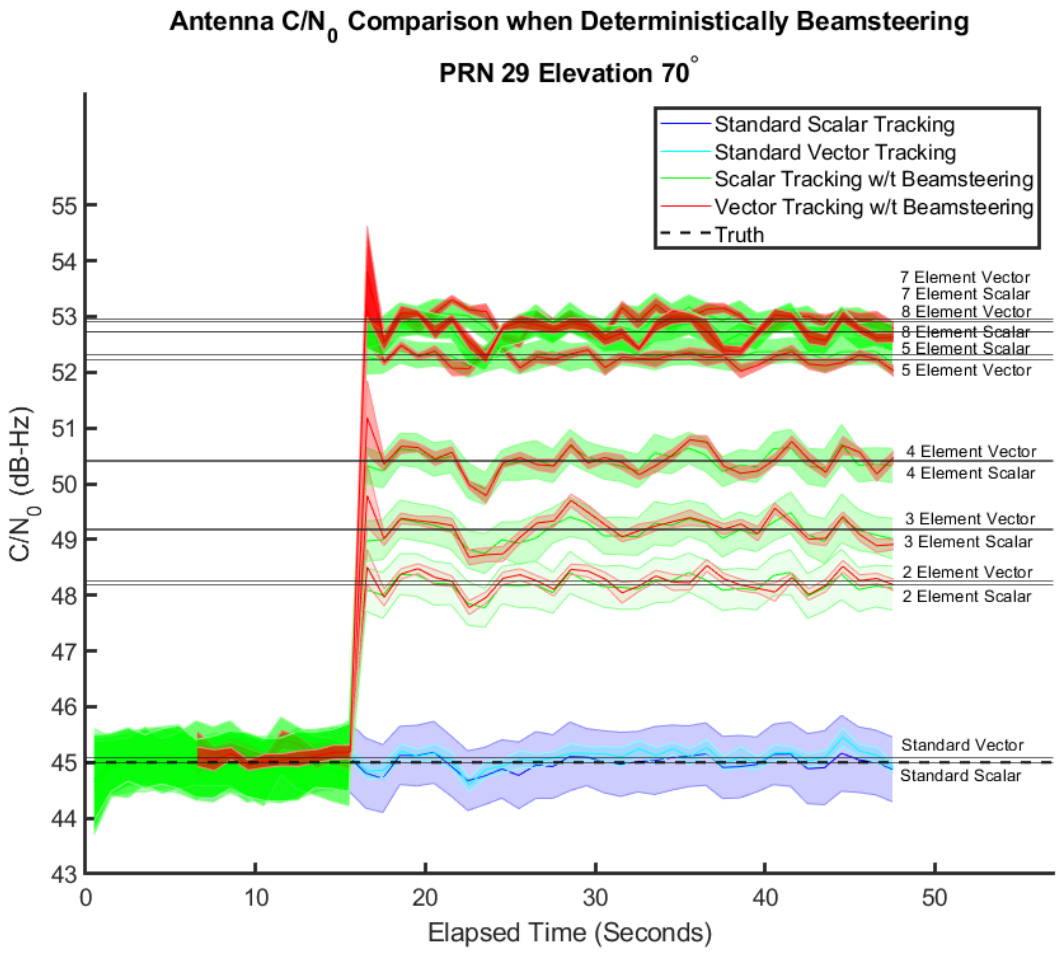

Figure 6.7: Simulated C/N_0 Improvements Beamsteering Toward a High Elevation SV

The prospective signal quality in the receiver has improved with a maximum result of increasing the C/N_0 by 8 dB-Hz for the seven and eight element configurations. This is significant and allows the vector and scalar multi-antenna receivers to push the bounds on track-able low power signals, hence the appeal of multi-antenna receivers.

Figure 6.8 plots the same data in Figure 6.7 after the beamsteering is initiated. Again, since the receiver is stationary, the mean and 1σ are calculated over the last 30 seconds of the data. The overall trend of diminishing return is evident. Also, a general decrease in C/N_0 variance is notable as the signal quality increases. Vector tracking has a lower variance on C/N_0 compared to scalar tracking.

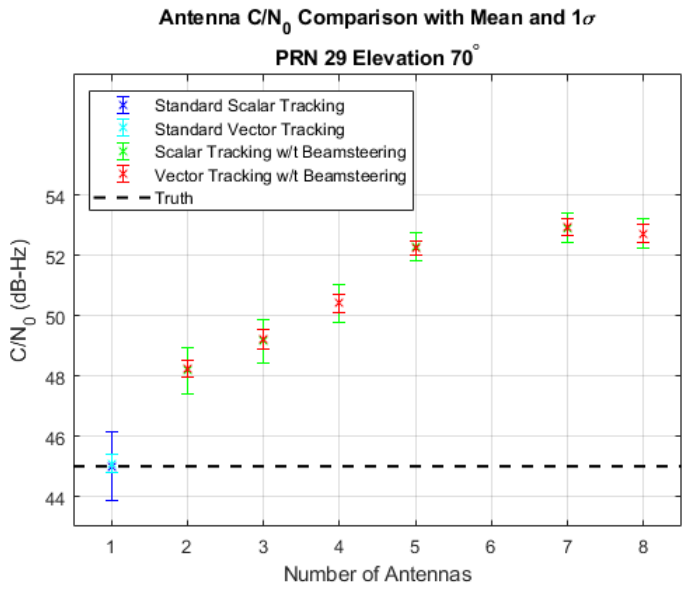

Figure 6.8: C/N_0 Improvements after Initiating Beamsteering Toward a High Elevation SV

Figure 6.9 shows the radiation pattern for each array beamsteering to a mid-level satellite (PRN 15 at el. = 52°). The five element shows quarter wavelength spacing characteristics with multiple high power side lobes. However, it still has its highest lobe aligned with PRN 15. All of the arrays continue to steadily improve the dBi gain associated at the front-end of the receiver, which is expected.

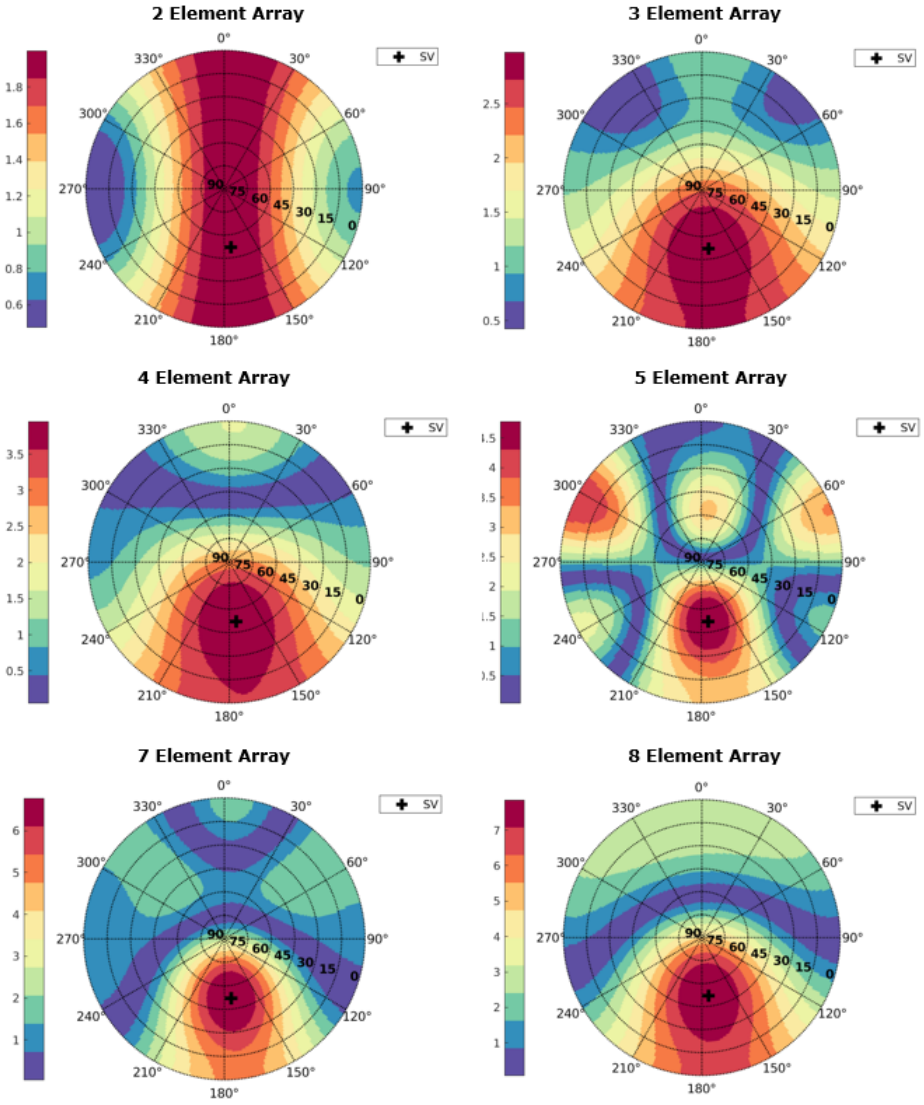

Figure 6.9: Magnitude Radiation Patterns for Mid Elevation PRN 15 [dBi]

Figure 6.10 shows the signal quality improvements when beamsteering to PRN 15 using the antennas under testing. The results are similar to the results in Figure 6.7, with exception to a one dB-Hz decrease in C/N_0 for the five and seven element CRPAs and a one dB-Hz increase in C/N_0 for the eight element CRPA. The eight element array increased the signal quality by eight dB-Hz at the mid elevation level.

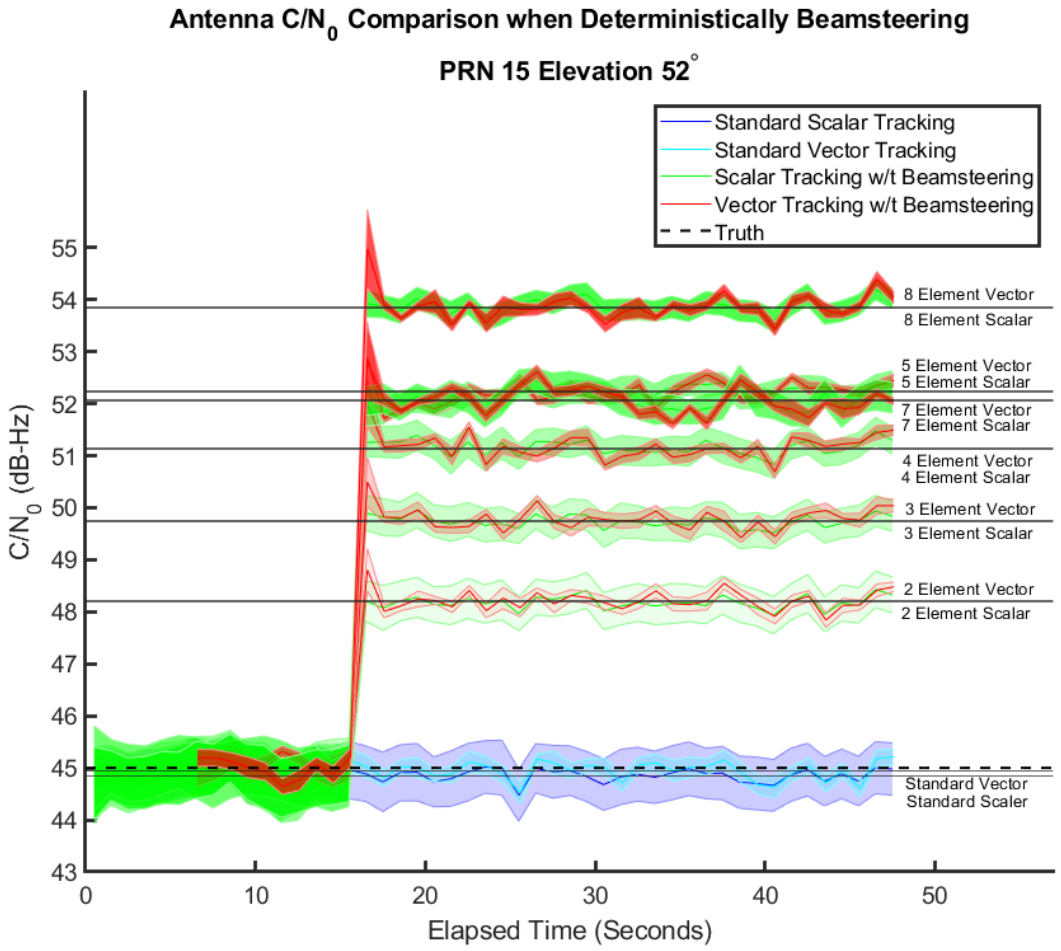

Figure 6.10: Simulated C/N_0 Improvements Beamsteering Toward a Mid Elevation SV

Figure 6.11 shows the mean and 1σ values for the receiver PRN 15 C/N_0 values. Similar to PRN 29 in Figure 6.8, there is a diminishing return in C/N_0 as the number of elements increase. There is a large performance gain by switching from a one to two element array. However, that return lessens as the number of antennas increases (i.e. element 4 to element 5). The data in Figure 6.8 and Figure 6.11 follow an exponential curve. Since the C/N_0 improves by the analytical solution $M/sqrt(M)$ where M is the number of array elements, the data following an exponential curve is expected.

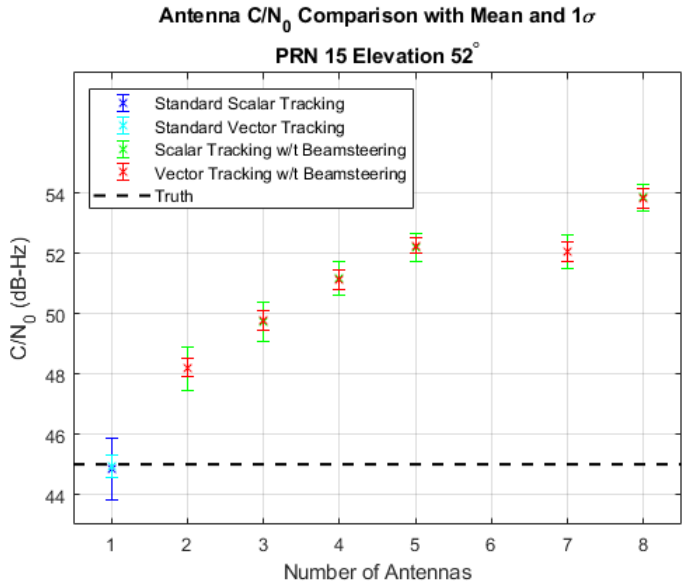

Figure 6.11: C/N_0 Improvements after Initiating Beamsteering Toward a Mid Elevation SV

At lower elevations (PRN 18 at el. = 17°), the antenna array radiation pattern still performs the same as it does at higher elevations. This is a byproduct of simulation and is not the case in practice. The simulation assumes the array elements are isotropic. In practice, most antennas do not have a uniform magnitude distribution, and most antennas have less gain at low elevations.

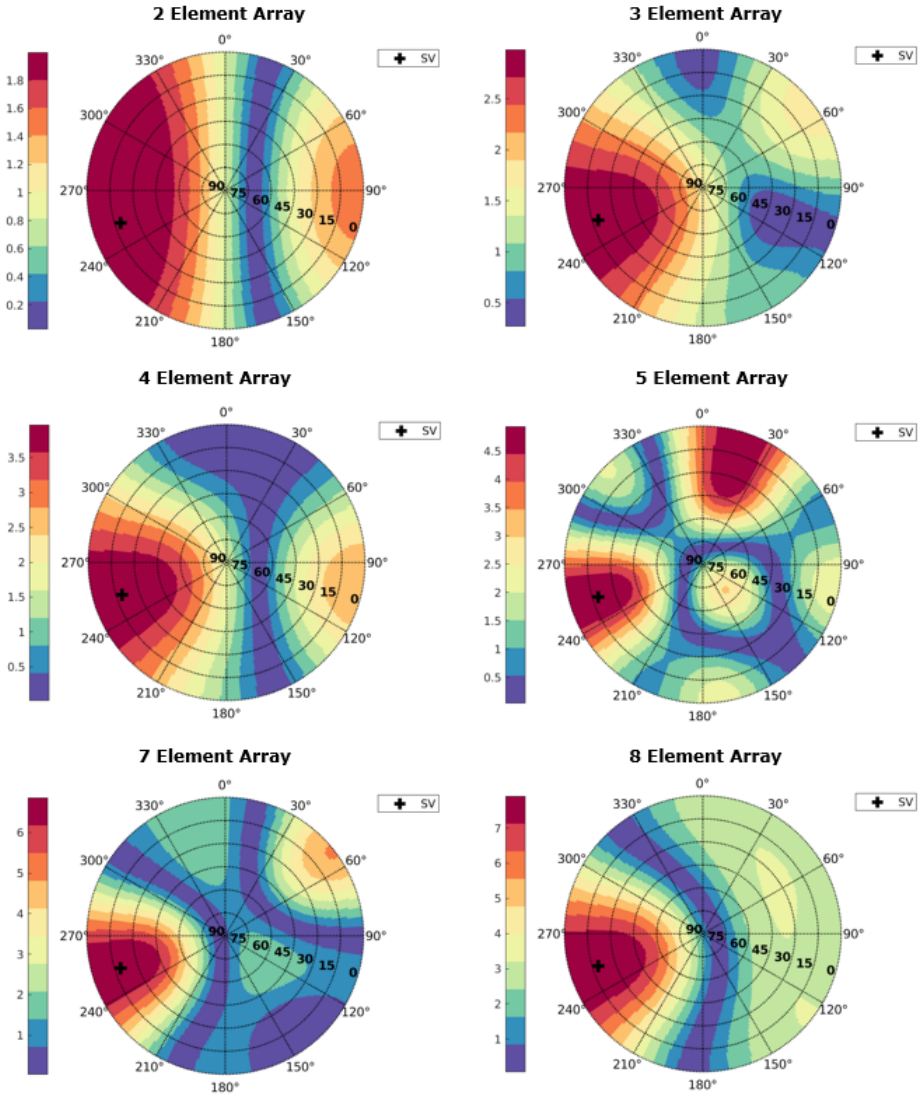

Figure 6.12: Magnitude Radiation Patterns for Low Elevation PRN 18 [dBi]

Figure 6.13 shows the different arrays performance beamsteering to low elevation PRN 18. The seven element array outperforms the other antenna arrays in the simulation, offering almost a 10 dB-Hz increase in C/N_0. This is a deviation from the analytical solution that might be attributed to the geometry of the seven element. The field of selecting the optimal array geometry to achieve specific performance returns is called array pattern synthesis, which is not covered in this work.

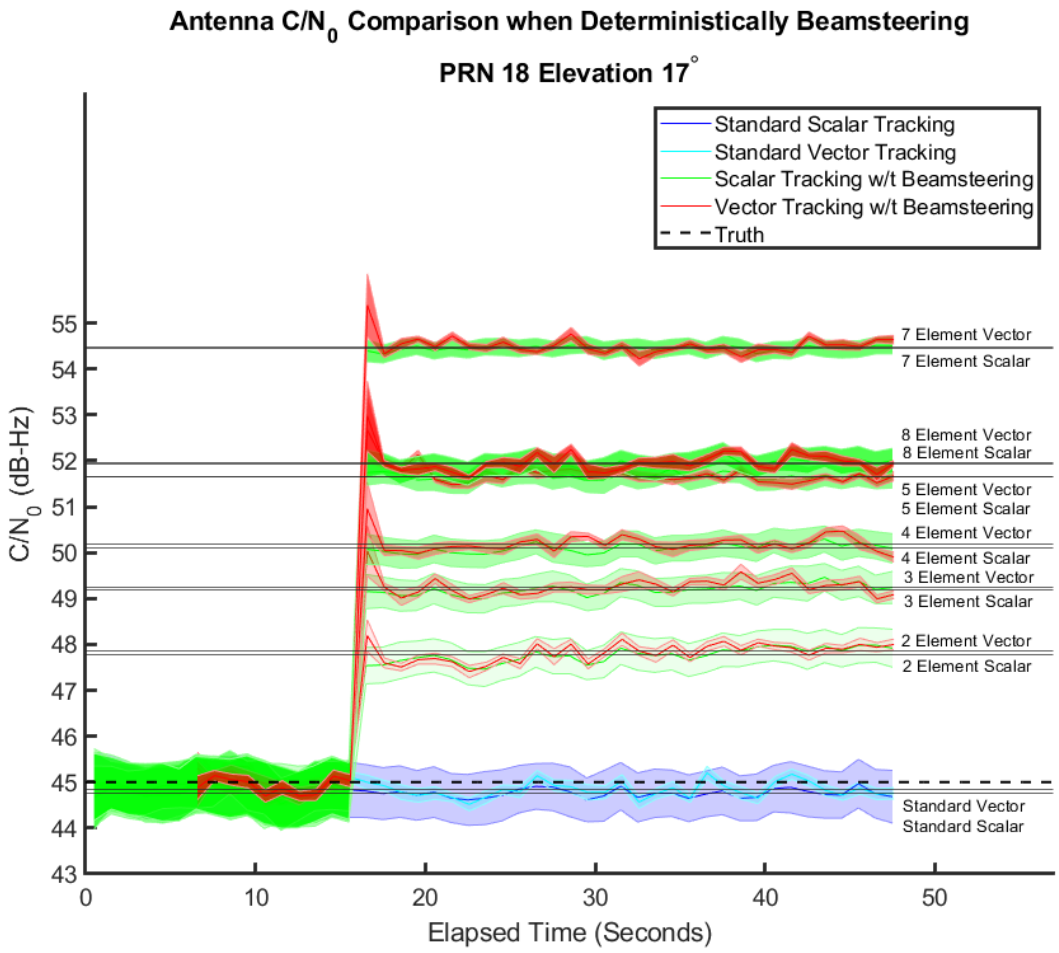

Figure 6.13: Simulated C/N_0 Improvements Beamsteering Toward a Low Elevation SV

From Figure 6.14 outlines the same values from Figure 6.13. Excluding the seven element array, the data still follows an exponential curve. Array pattern synthesis might explain the deviations in the seven element array performance values. It should also be noted that this data is collected from single runs on static scenarios. In other words, the same 48 second scenario was generated for each antenna array, and the scenario was processed once. A monte carlo simulation was considered, but not included in this work due to the significant computation time associated with the simulation.

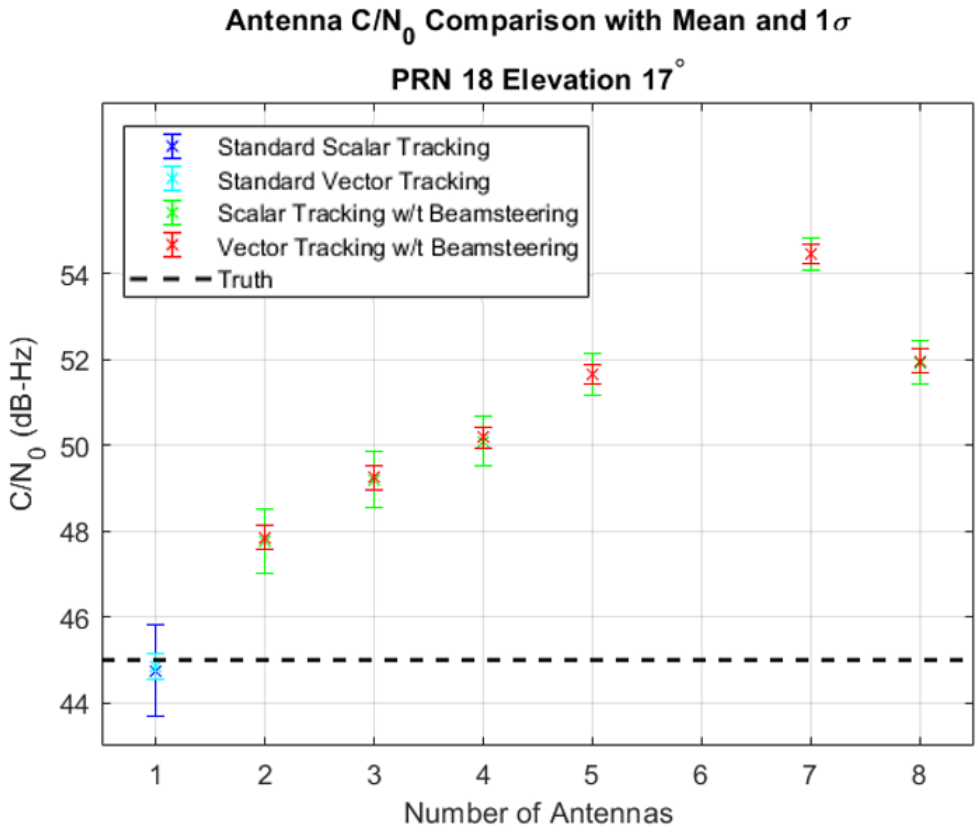

Figure 6.14: C/N_0 Improvements after Initiating Beamsteering Toward a Low Elevation SV

From this data and Equations (5.14b) and (5.14c), it is implied that the range and range rate variances decrease when C/N_0 is increased. Appendix D.1 shows the calculated range and range rate variances for standard vector tracking and multi-antenna vector tracking beamsteering.

6.2.2 Signals at Variable Power

Since vector tracking specializes in aiding low power satellites, the following simulation decreased signal power on low elevation satellites to see the benefits of vector tracking. For this simulation, signals were simulated like in the previous section with all signals having a C/N_0 of 45 dB-Hz. After approximately 50 seconds, signal power on the low elevation satellites was dropped significantly. High elevation satellites maintained their previous C/N_0

values. C/N_0 improvements are shown as well as range and range rate variance calculations are shown in Appendix D.2.

Tracking Improvements

Figure 6.15 shows the C/N_0 dropping to 25 dB-Hz, well below the tracking threshold for scalar tracking. The C/N_0 improvements for PRN 18 are the same over the same power section of the simulation, as shown in Figure 6.13. Note standard scalar tracking loses lock at the power drop. For the receivers being tested, an unlocked loop outputs C/N_0s at a 28 dB-Hz level. This nuance will be fixed in future testing, but it is present and evident in all the results in this thesis. Vector out performs scalar tracking at nearly all antenna levels, showing the benefits of vector tracking. Standard scalar tracking is shaded in gray due to its loss of lock at low power.

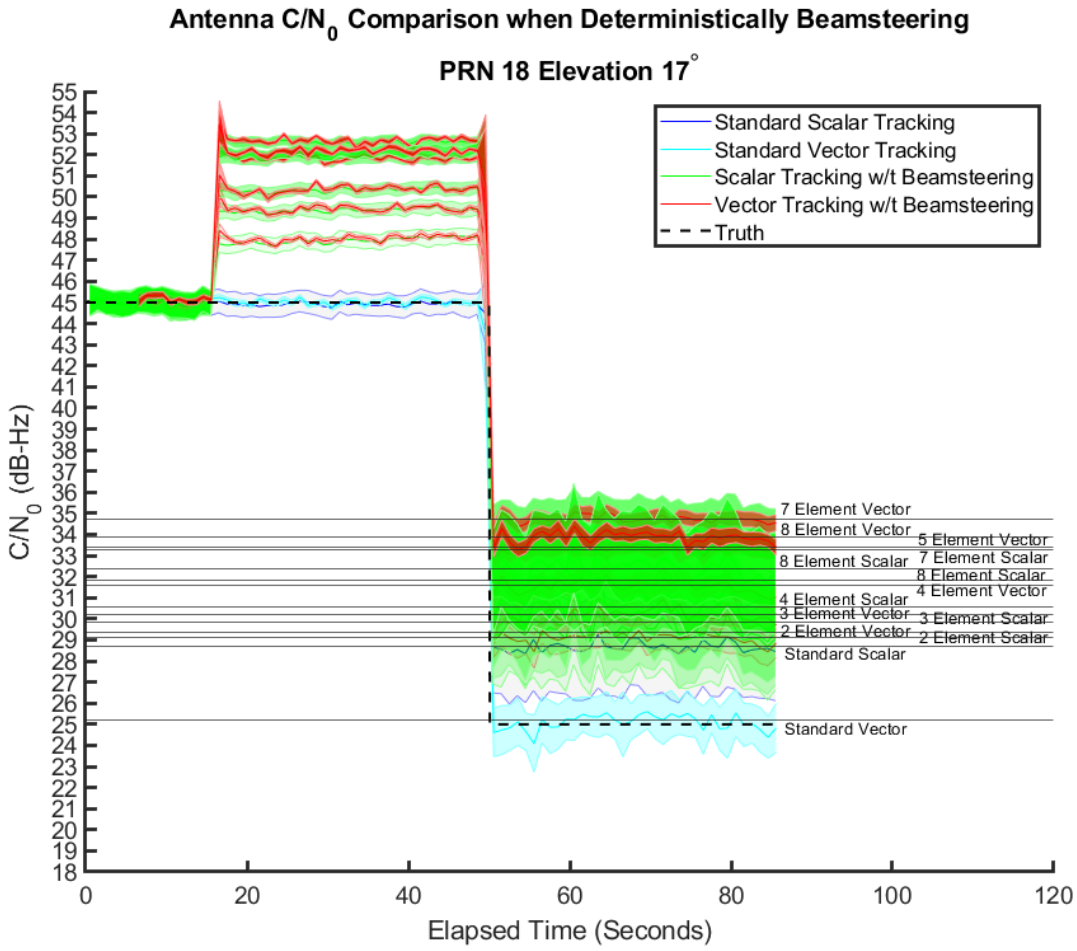

Figure 6.15: Low Power C/N_0 Improvements Beamsteering at 17° Elevation

For a better analysis, Figure 6.16 shows the mean and 1σ data after the drop in signal power. The loss of scalar lock for one antenna is evident as it drifts up to the 28 dB-Hz value. Vector tracking with one antenna is able to maintain lock. Multi-antenna scalar beamsteering shows a significant variance in C/N_0, but can still maintain tracking capabilities. Multi-antenna vector tracking beamsteering shows tracks with a manageable variance and accurate values throughout. As in the previous figure, standard scalar tracking is shaded gray because it lost lock and deviates from the truth simulated power level to the undesirable 28 dB-Hz unlocked value.

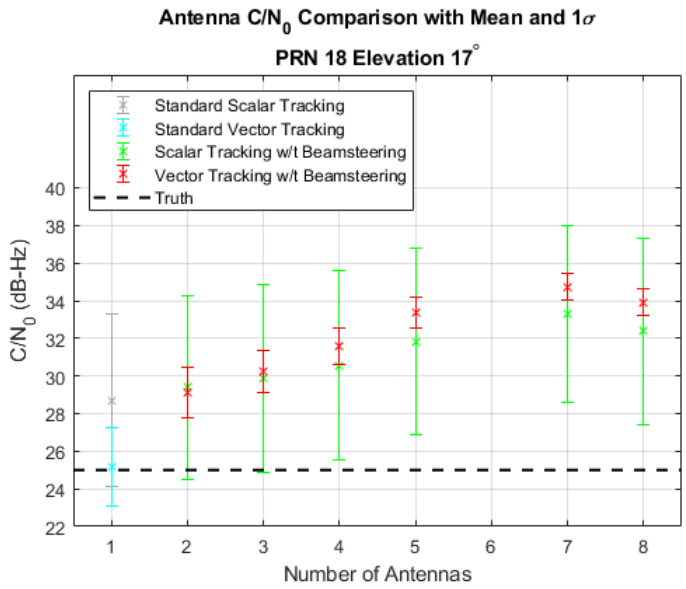

Figure 6.16: C/N_0 Improvements after Power Decrease at 17° Elevation

Extending the comparison to tracking parameters, Figure 6.17 shows the code frequency tracking estimates. The multi-antenna scalar tracking results are plotted in blue, and the multi-antenna vector tracking results are plotted in orange. When the C/N_0 value is dropped to 25 dB-Hz, the variance on the multi-antenna scalar code frequency estimate increases but is able to maintain lock. The variance increase in the code frequency estimate directly affects the code discriminator, which directly affects the pseudorange estimates. Multi-antenna vector tracking is able to maintain a less variant lock compared to multi-antenna scalar tracking.

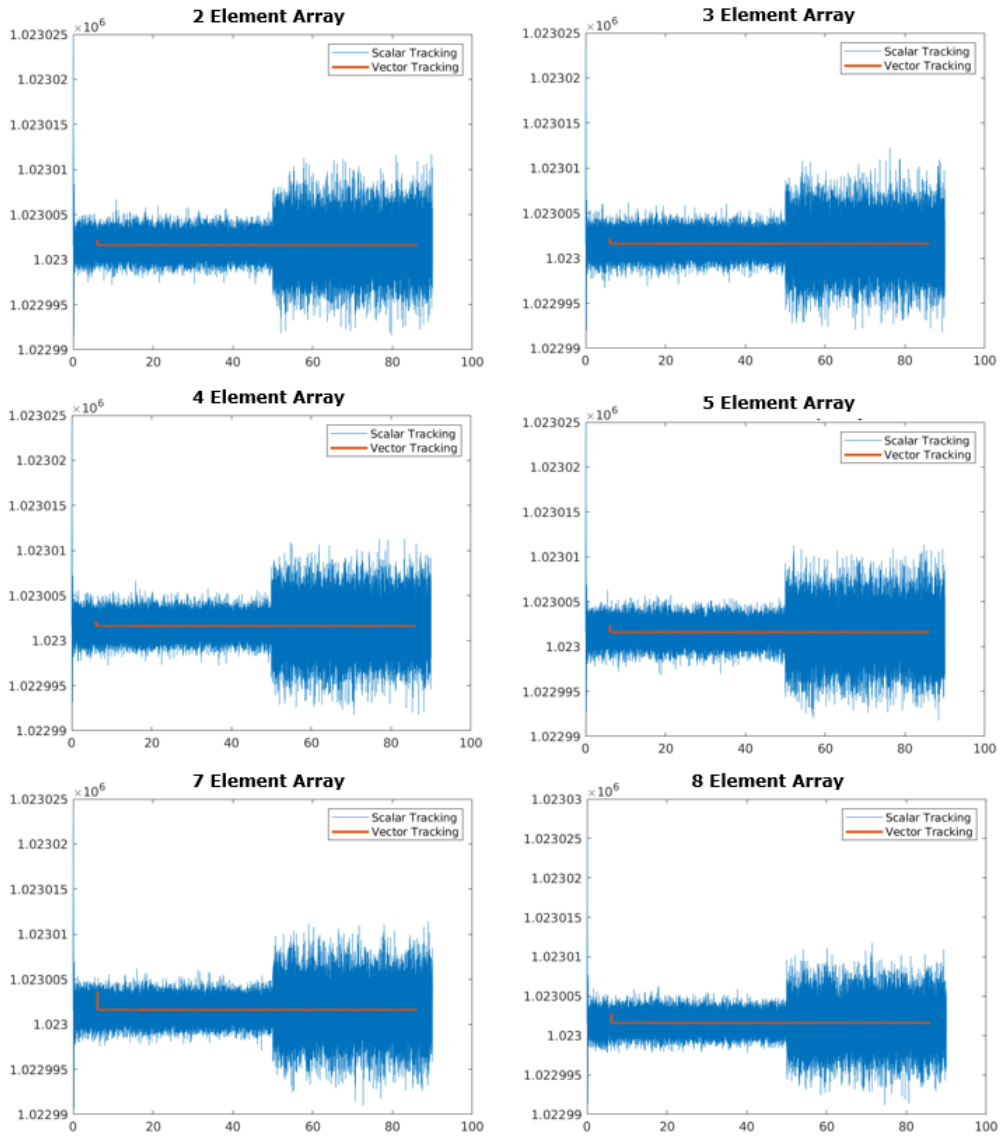

Figure 6.17: Antenna Array Code Tracking Comparison for PRN 18

Figure 6.18 shows the carrier tracking results for PRN 18. Similar to Figure 6.17, the variance increases on the multi-antenna scalar estimates. The multi-antenna vector tracking loop has less variance in its estimates. The variance in the carrier frequency estimate directly affects the pseudorange rate estimates in both tracking schemes.

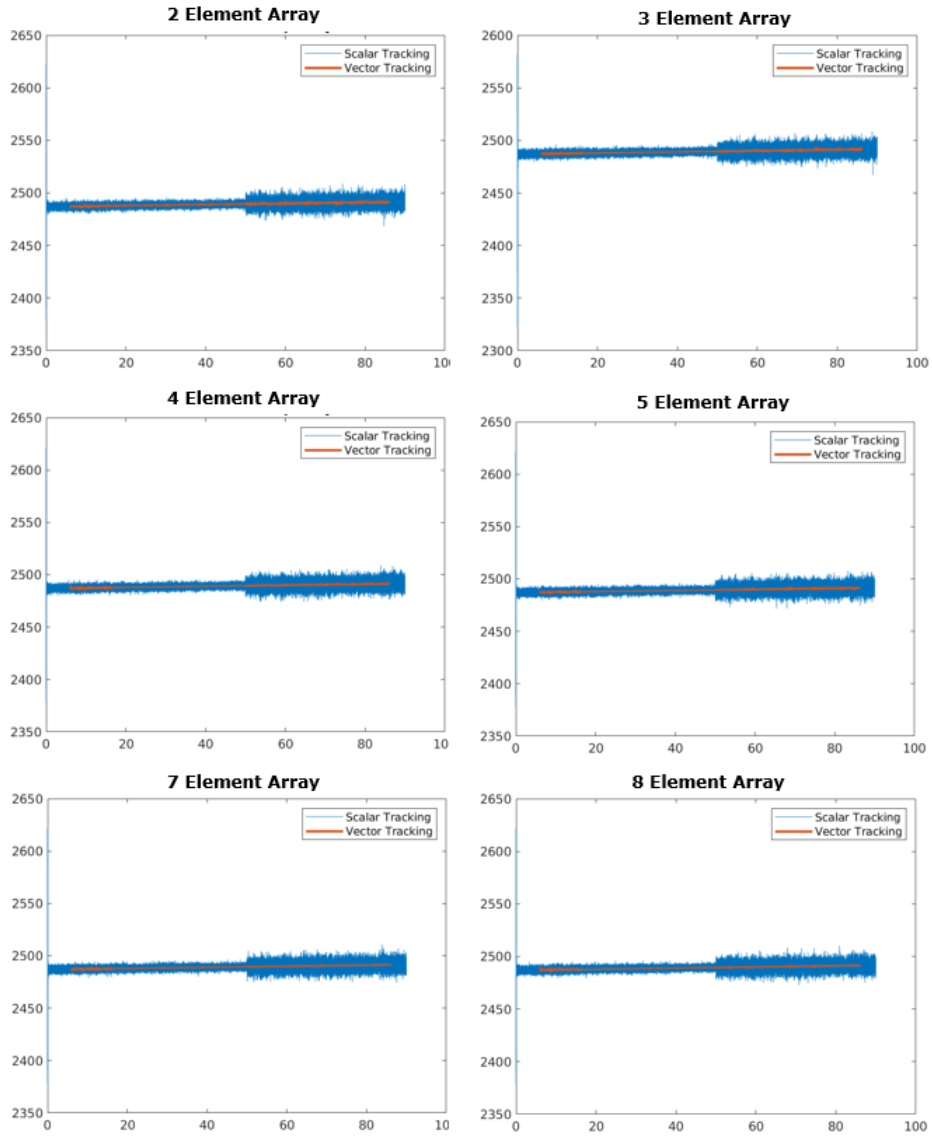

Figure 6.18: Antenna Array Carrier Tracking Comparison for PRN 18

A satellite at 2 degrees elevation was simulated with a true power level of 20 dB-Hz to further test the resilliance of the different receivers. At this level, even vector tracking should struggle. Figure 6.19 shows the resulting signal quality of SV 9. All of the scalar tracking C/N_0s showed loss of lock characteristics with all of them around the same level of 28 dB-Hz. The single antenna vector tracking mean also deviated from the true value of 20 dB-Hz C/N_0. Nevertheless, vector tracking maintains lock for every configuration. For the seven element configuration with a net increase in C/N_0 of approximately eight dB-Hz, the

seven element vector tracking loop can at least handle and track a signal at 12 dB-Hz C/N_0.

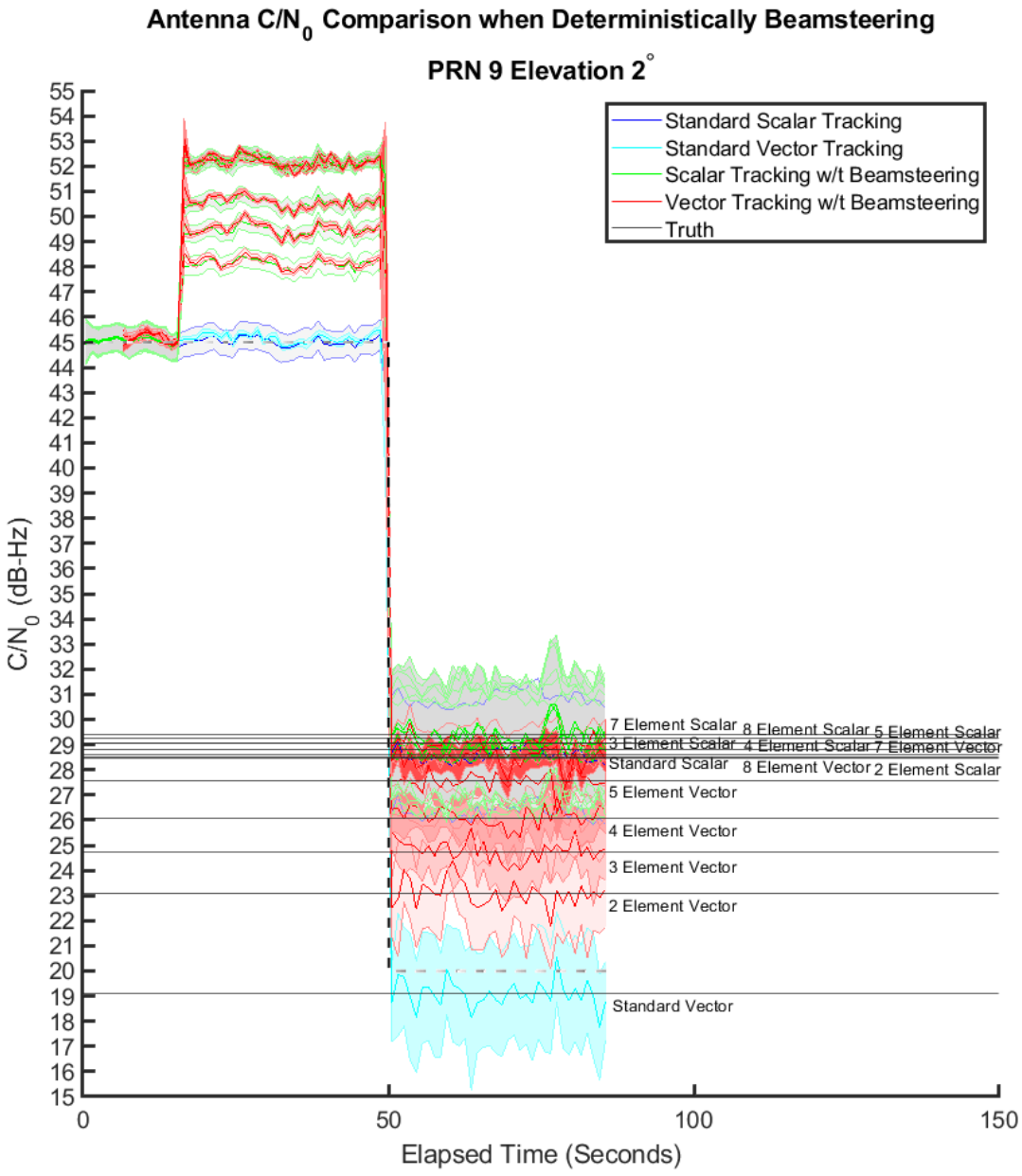

Figure 6.19: Low Power C/N_0 Improvements Beamsteering at 2° Elevation

Figure 6.20 shows vector tracking obtaining valid C/N_0s after the drop in power. All scalar configurations output at the same level indicating loss of lock.

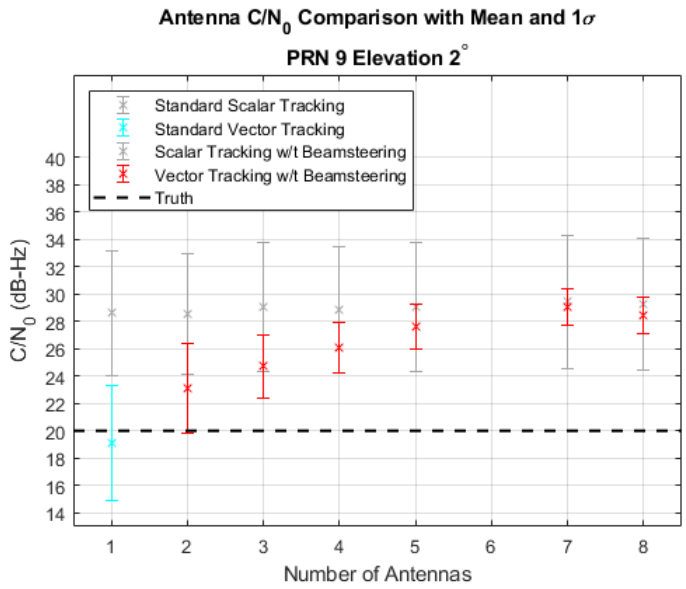

Figure 6.20: C/N_0 Improvements after Large Power Decrease at 2° Elevation

Figure 6.21 also shows the multi-antenna scalar tracking receivers losing lock when C/N_0 was dropped to 20 dB-Hz on PRN 9. The scalar code frequency estimates immediately increase in variance as soon as the C/N_0 was decreased. The five element antenna array loses lock on the code frequency. Other arrays like the three, four, seven, and eight elements are able to maintain lock for short periods of time. Eahc array showed divergent responses at certain instances in the simulation. On the contrary, the multi-antenna vector tracking receivers were able to successfully track PRN 9 after the C/N_0 was dropped. The clean signal channels in the vector tracking loop were able to successfully aid the PRN 9 noisy channel.

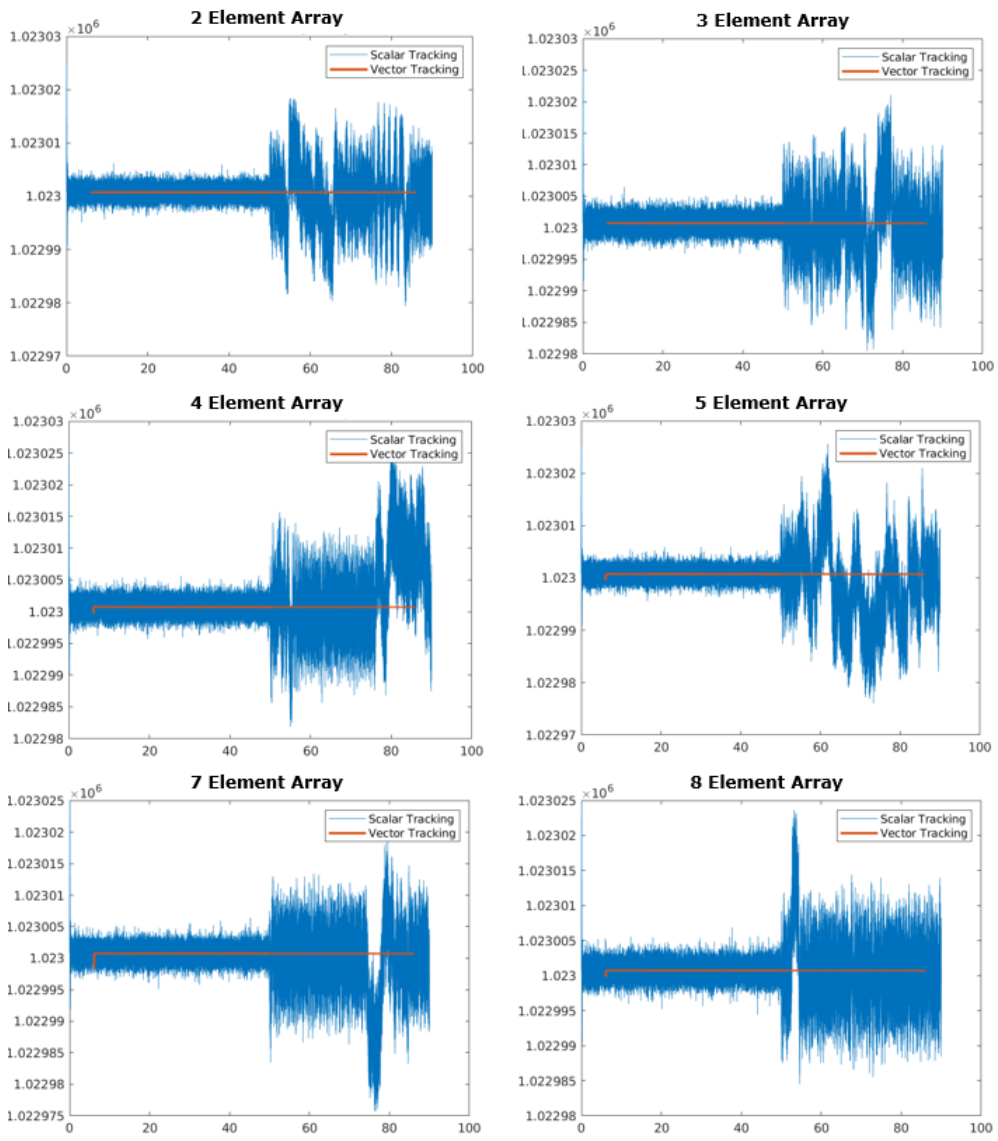

Figure 6.21: Antenna Array Code Tracking Comparison for PRN 9

Figure 6.22 shows the carrier frequency tracking results for PRN 9. All of the multi-antenna scalar tracking receivers diverged from the true carrier frequency. The multi-antenna vector tracking receivers were able to maintain carrier lock during the 20 dB-Hz C/N_0 sequence.

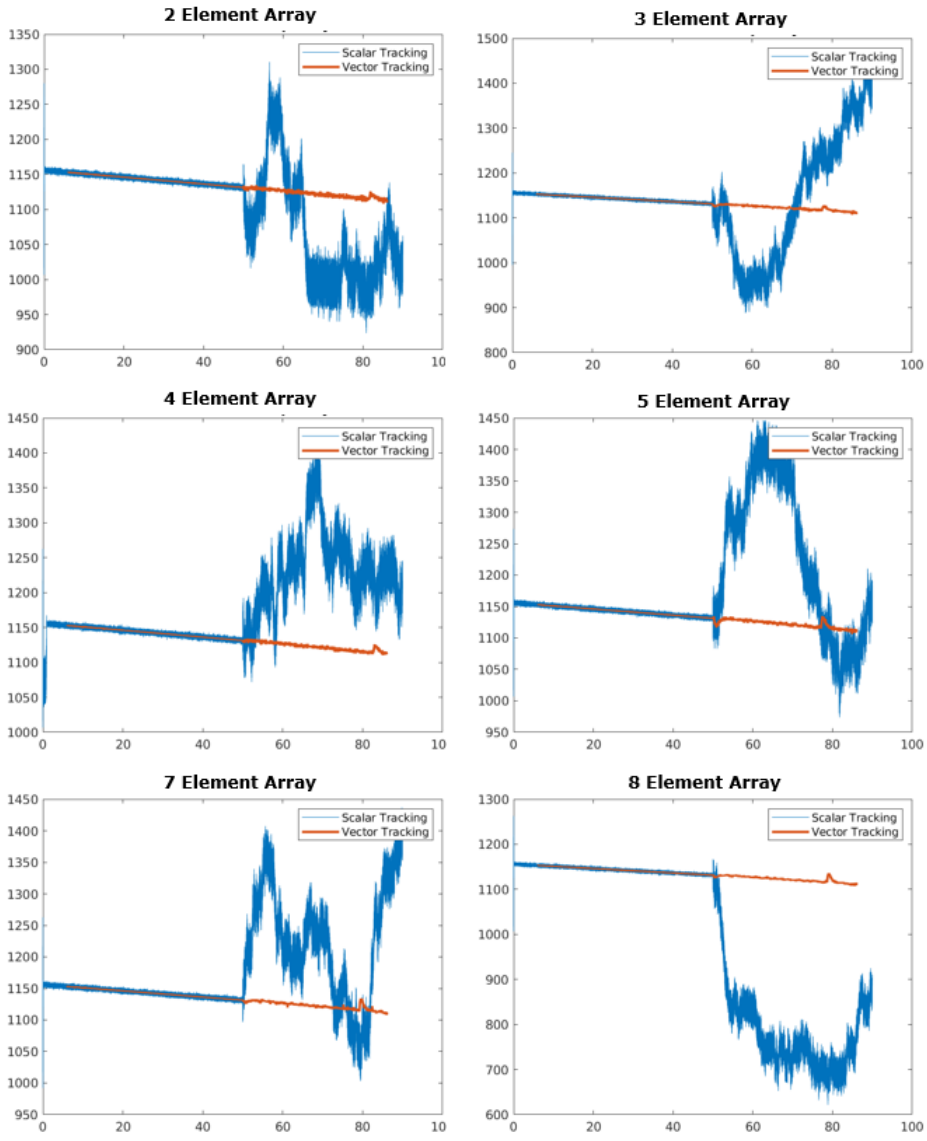

Figure 6.22: Antenna Array Carrier Tracking Comparison for PRN 9

With other SVs at low elevations dropping to a low power level as well, position error from the simulated truth was analyzed to understand the effects on the receiver output.

Position Error Improvements

The position error was analyzed at two locations in the variable power simulation; the locations were before the drop in power after beamsteering was initiated and after the power drop. Table 6.1 shows the absolute mean and 1σ scalar tracking position values in the ENU

frame. The scalar configurations are listed, increasing in the order of antennas. Table 6.2

Table 6.1: Scalar Tracking Position Error from SV Power Simulations

	SVs at Same Power						Low Power SVs Added					
Type (# of e)	μ_E	μ_N	μ_U	σ_E	σ_N	σ_U	μ_E	μ_N	μ_U	σ_E	σ_N	σ_U
Scalar (1)	3.9	6.6	1.0	4.2	6.8	10.0	11.1	10.6	71.2	18.4	16.6	143.0
Scalar (2)	3.7	6.3	1.3	3.9	6.3	9.0	10.0	9.5	63.6	15.3	12.9	111.3
Scalar (3)	3.8	6.4	1.6	3.9	6.2	9.2	10.0	10.2	44.3	13.3	11.0	91.9
Scalar (4)	3.7	6.5	2.4	3.9	6.1	9.0	7.8	9.0	26.0	9.3	9.5	29.0
Scalar (5)	3.7	6.0	0.4	3.0	6.2	8.8	8.5	9.0	27.8	9.9	9.5	32.2
Scalar (7)	3.8	5.9	2.9	3.8	6.2	8.9	7.6	8.9	23.4	9.0	9.5	27.6
Scalar (8)	3.6	6.1	2.9	3.8	6.2	8.9	7.7	8.9	25.8	9.1	9.5	28.6

shows the absolute mean and 1σ results for the scalar tracking position values in the ENU frame.

Table 6.2: Vector Tracking Position Error from SV Power Simulations

	SVs at Same Power						Low Power SVs Added					
Type (# of e)	μ_E	μ_N	μ_U	σ_E	σ_N	σ_U	μ_E	μ_N	μ_U	σ_E	σ_N	σ_U
Vector (1)	2.4	4.9	1.8	0.5	0.3	2.1	2.1	2.6	11.9	2.4	2.3	9.3
Vector (2)	2.3	4.9	2.3	0.5	0.4	2.7	2.3	2.3	10.6	2.7	2.1	8.7
Vector (3)	2.2	4.8	2.3	0.6	0.4	2.7	4.0	1.2	6.5	2.3	1.5	8.1
Vector (4)	2.2	4.6	2.8	0.6	0.4	3	2.7	3.2	13.8	3.2	2.2	10.5
Vector (5)	2.2	4.4	3.0	0.6	0.6	3.5	4.4	1.5	6.8	2.7	1.8	8.9
Vector (7)	2.5	4.5	3.2	0.7	0.6	3.6	3.3	1.9	7.5	2.5	1.8	8.3
Vector (8)	2.1	3.9	3.1	0.6	0.5	3.3	3.6	2.3	7.2	2.6	1.8	8.5

There is a significant amount of position error from the data when using single element scalar tracking at the low power level. Most of the position error is in the up direction with absolute mean of 71.2 meters. The standard deviation in the up direction for single antenna scalar tracking is also high, with the standard deviation being 143.0 meters. The east and north directions also showed significant errors in position. When initiating single element vector tracking, the position error decreases in all ENU. The up direction improved from 71.2 meters to 11.9 meters, along with five times better positions in the East and North directions. For scalar tracking with multiple antennas, the position errors tend to become smaller as the number of antennas increases, with a few exceptions in the data. The same

can be observed by the standard deviation of the error as the variance mostly decreases as the number of elements increases.

Figure 6.23 shows the SVs before power levels were dropped in the simulation. For the SVs at 45 dB-Hz C/N_0, vector tracking has less absolute error than scalar tracking for all directions except for the up direction where vector tracking matches scalar tracking in absolute error. Vector tracking also has a considerably lower standard deviation in position than scalar tracking.

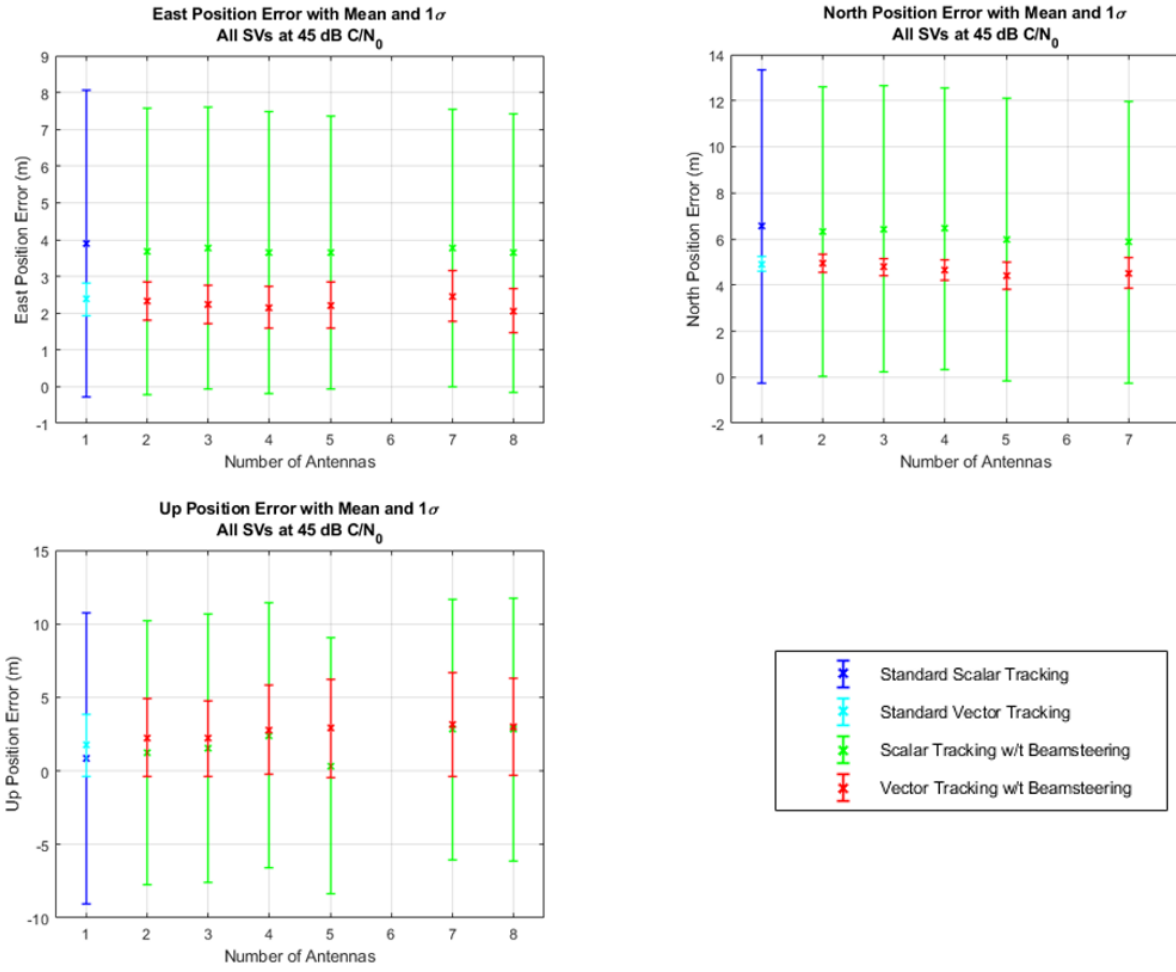

Figure 6.23: Simulated Position Error Comparison with All SVs Simulated at Same Power

For the scenario simulated with lower power SVs in Figure 6.24, the vector tracking position error across all CRPAs is lower than scalar tracking across all of the CRPAs. As

noted in the data in the table, the scalar tracking position variances decrease as the number of antennas increase. Even though the low elevation channels for the scalar tracking loop lose lock, the scalar tracking receiver still positions using only higher elevation SVs.

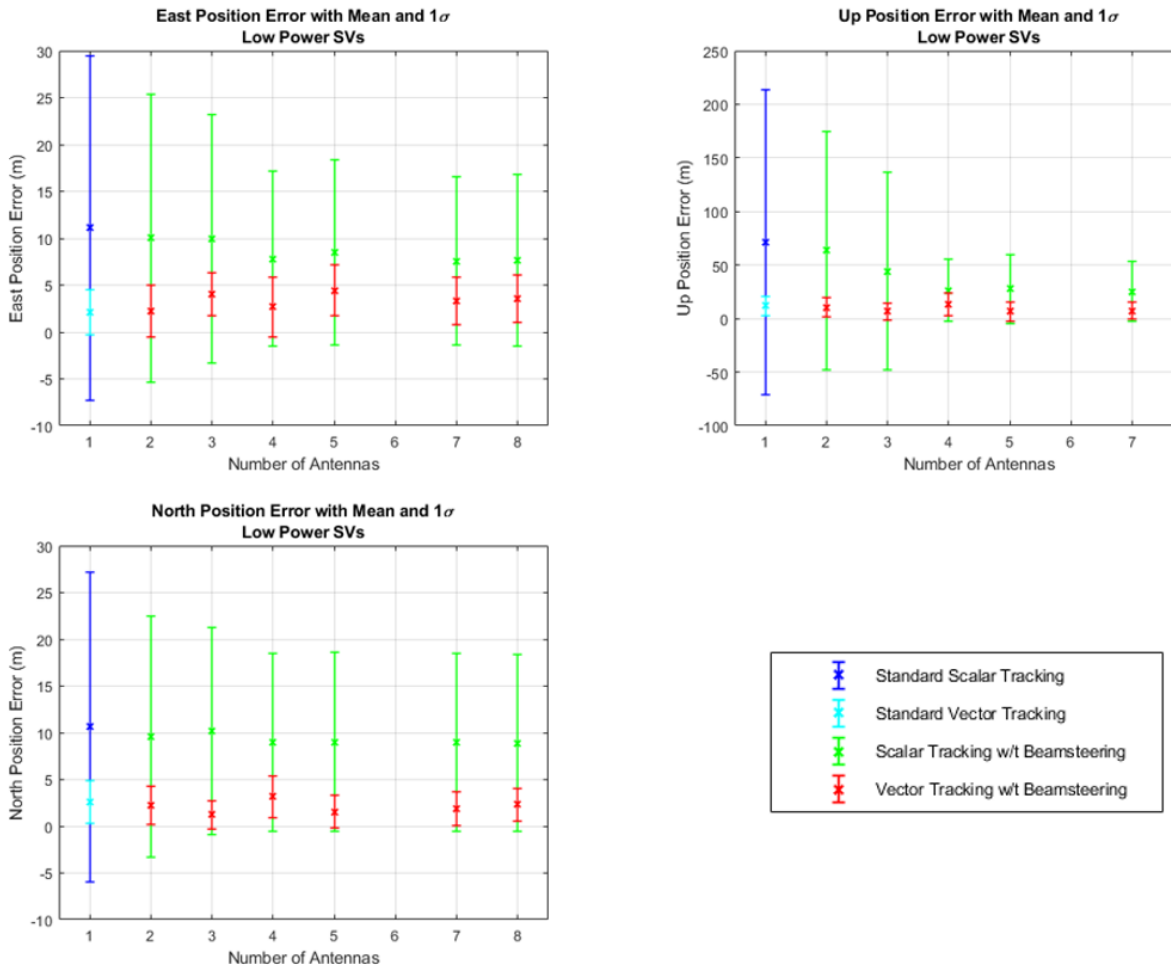

Figure 6.24: Simulated Position Error Comparison with Low Power SVs Ingested

Based on the simulation results, the multi-antenna vector tracking should yield improved results for live sky data. The following section outlines how live sky was collected and processed in the software receivers, as well as the comparison results of multi-antenna scalar and vector tracking receivers.

6.3 Live Sky Testing

Hardware calibration routines are required for live sky processing in a CRPA receiver. Additionally, a course or attitude update is required to keep the beamsteering beams pointing in the desired directions as the receiver changes orientation (Section 6.1.2). Since there are four front-ends with four signal streams going to the receiver, the phase and magnitude difference between the hardware lines must be accounted for in calibration (Section 6.1.2). Typically, array manifold calibrations must be conducted, but an assumption is made in this thesis that does not require array manifold calibrations. The data collection setup is outlined in this section as well as the receiver performance after processing the live sky data. An interference scenario was also created by injecting jamming data on live sky collected data, which is shown in Section 6.4.

6.3.1 Acquisition Phase Calibration

The acquisition phase calibration was conducted in post-processing to remove the line to line phase difference induced by the USRP X310, the front-end selected for live sky data recording, as well as account for delays induced by the array manifold. Since the local oscillator (LO) in the X310 with Twin Rx daughter boards is the same for all channels, the phase offset is the only calibration process that was conducted in this work. An antenna calibration was not necessary in this work since the phase calibration included the antenna. An overview of the phase calibration process is shown in Figure 6.25.

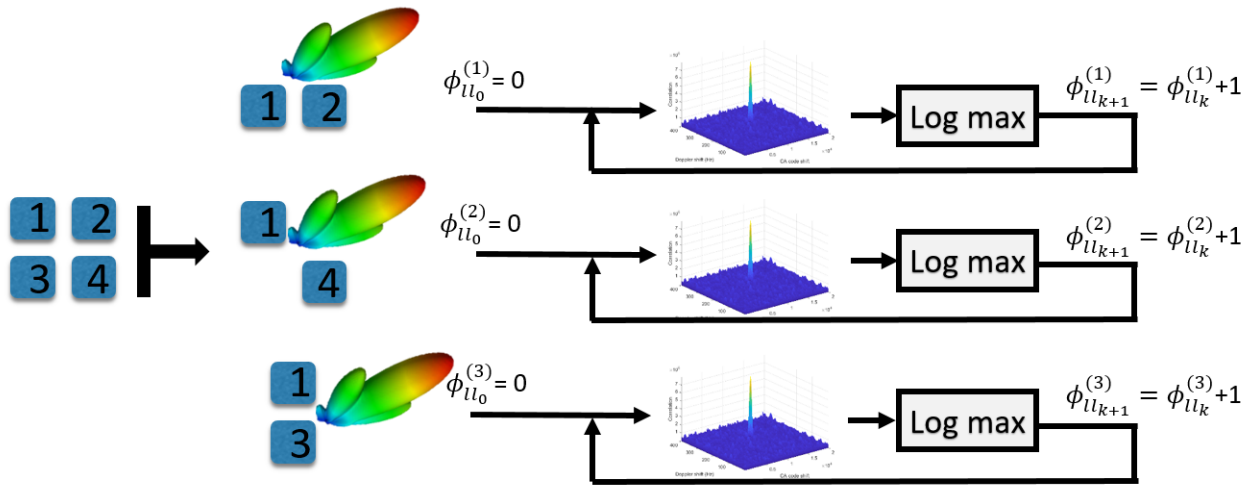

Figure 6.25: Line-to-Line and Array Manifold Phase Calibration

Acquisition phase calibration can be described in three parts: signal combination, acquisition, and max peak detection. For signal combination, signals were combined in this order: signals 1 and 2, signals 1 and 4, and signals 1 and 3. Each combination creates a miniature 2 element sub-array, and the sub-arrays are beamsteered to the satellite. Line-to-line phase delays ϕ_{ll} were added to the beamsteering phase delays, and the line-to-line phase delays were cycled from 0 to 360 degrees with 1 degree resolution, giving a total of 360 different combinations per signal combination. For each signal combination, acquisition was processed, and the acquisition peak magnitude was logged. The same batch of signal data was used, so Doppler was known during the entire process. This reduced acquisition to be a code phase search, which significantly reduced computation time since 1080 acquisitions must be conducted for a four element CRPA. Next, the line-to-line phase delay yielding the maximum acquisition peak from each signal combination was logged. The line-to-line phase delays yielding the highest peaks were taken to be the line-to-line phase delays induced by the hardware.

This process was conducted three times, using the satellites with the three highest elevations and averaging the line-to-line phase delays computed from each satellite. This

process worked well and accounted for array manifold perturbations since the antenna was included in the calibration.

6.3.2 Data Collection Setup

The collection setup for collecting synchronized four channel RF data is described in this section. Figure 6.26 shows a graphical layout of the data collection setup. The CRPA antenna used for the collection was the Antcom Active CRPA 4CF-75G121SP-XS-4. The antenna is shown in Figure 6.26 and referenced in Section 6.2. Even though the antenna is active, quantization noise was a concern for the USRP X310 since GPS L1 C/A data is at low power. With that in mind, a custom JFW signal conditioning box was placed inline between the USRP and the Antcom antenna. The signal conditioning box provided programmable gain and bias tees that powered the CRPA. All cables used in data collection were of equal length yielding minimal difference in signal quality from stream to stream. As stated earlier, an Ettus USRP X310, integrated with a twin Rx daughterboard, was the front-end used for live sky data collection. It was connected downstream from the JFW signal conditioning box, and it offers programmable gains similar to that of the signal conditioning box. To ensure the signal data was not clipped, an AGC was programmed in UHD to continuously adapt the gains to give the best signal quality possible at the receiver. The JFW gains were set manually to raise each signal stream to -2.5 dBFS before the USRP AGC could fine-tune the signal data to better quality. A Twin Rx UHD program recorded 25 MHz sample rate two byte I two byte Q data.

Along with the USRP collecting four channel IQ data, a Ublox F9 receiver was used to record reference data for analysis. The Ublox was connected to a Novatel Pinwheel antenna, identical to the one in Section 4.3.2. A GPS L1 C/A bandpass filter was placed inline to level the playing field for the Ublox F9 and the other receiver configurations tested.

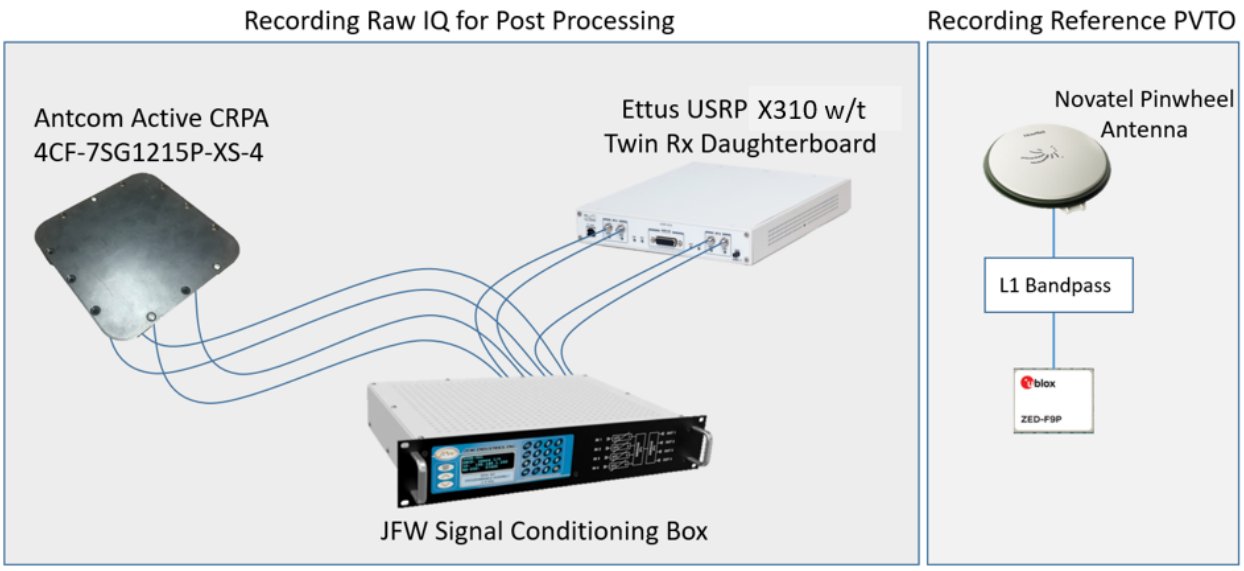

Figure 6.26: Data Collection Setup

A linux based recording computer managed both the Ublox and UHD recording software. The computer logged all of the data to a high-speed Samsung Evo Plus M.2 drive. Dual SFP ports were configured on the USRP to maximize streaming speeds. Unfortunately, the data writing to the M.2 drive turned out to be the bottleneck in the recording process. Time had to be allocated between recording routes to allow the M.2 drive to cool down. Only approximately 7 minutes of complex IQ data could be logged before the M.2 drive started underflowing (i.e. data samples) due to the drive overheating. This shortcoming in the data collection setup is noted as an improvement that should be implemented in future work.

Figure 6.27 shows the data collection vehicle used with the antennas mounted on the roof. The CRPA antenna is mounted 1 meter behind of the Novatel pinwheel antenna, and the antenna offset is considered in post-processing.

Figure 6.27: Data Collection Vehicle

Figure 6.28 shows a top view of the vehicle facing from back to front, with the CRPA antenna in the back and the pinwheel antenna in the front.

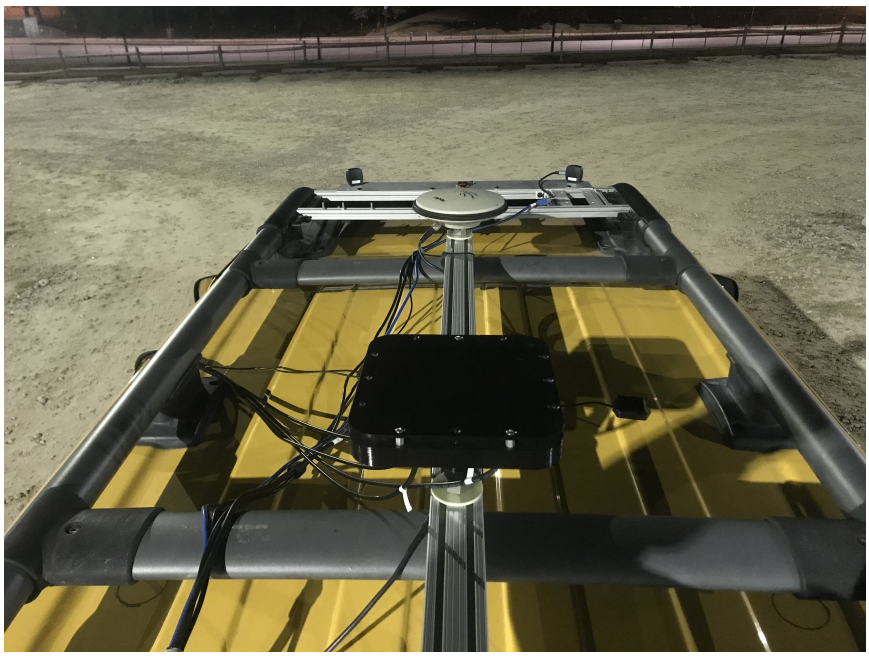

Figure 6.28: Antenna Mounting Configuration

6.3.3 Open Sky Environment

The first route processed was a relatively open sky route. The open sky route was a commute from IS4S to Conway Acres in Auburn, AL. The route included open sky data along the first portion of the route and then mild to medium foliage along the Cox Rd. portion of the route. A picture of the route is shown in Figure 6.29. The route also features an interstate section with high acceleration and deceleration sections in an automotive sense.

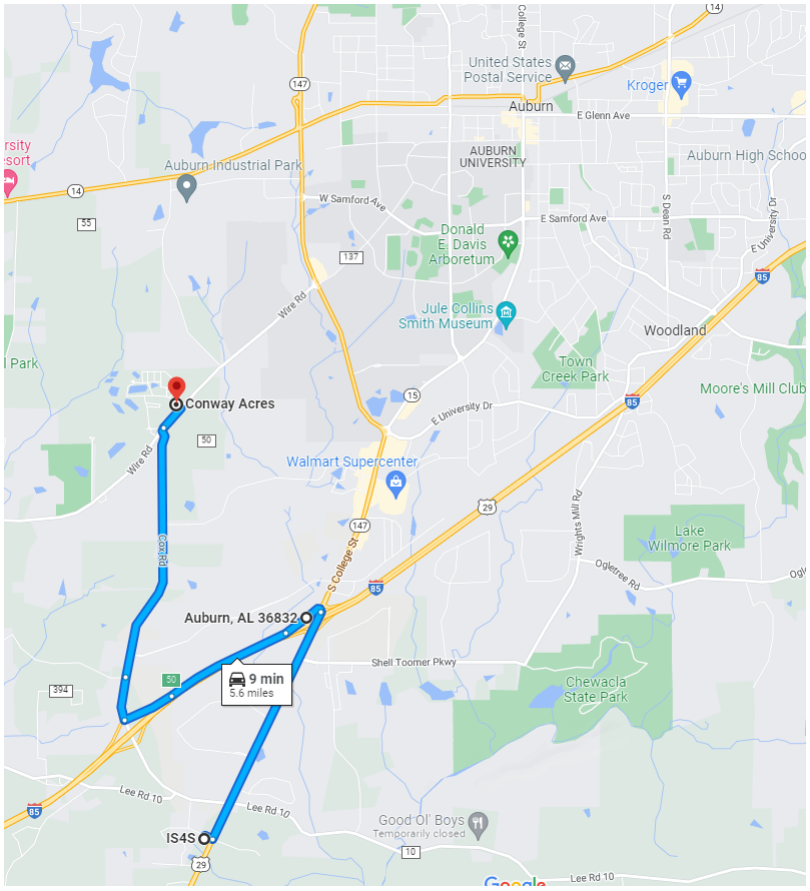

Figure 6.29: Google Maps Commuter Route [57]

The sky plot for the route is shown in Figure 6.30. With only six satellites present, outages in any satellite channel could cause positioning consequences. The satellites present yielded a PDOP value of 3.04 with a HDOP value of 1.52.

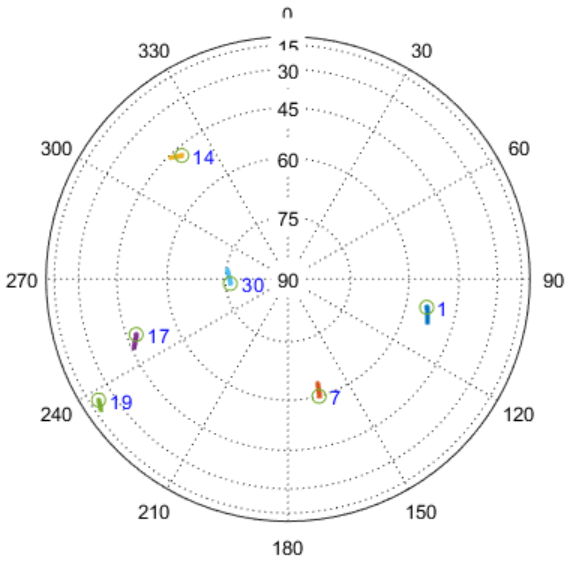

Figure 6.30: Sky Plot Commuter Route

After the route data was processed using a scalar tracking receiver, a scalar tracking receiver with beamsteering, a vector tracking receiver, and a vector tracking receiver with beamsteering, C/N_0 data was analyzed from a high, mid, and low elevation perspective.

C/N_0 Improvements

The SVs chosen for the C/N_0 analysis were SV 30, 14, and 19. SV 30 was at a high elevation of about 75 degrees, SV 14 was at a mid elevation of about 50 degrees, and SV 19 was at a low elevation of about 18 degrees. Figure 6.31 shows the resulting C/N_0 data from the route. For PRN 30, the course was updated correctly as vector and scalar beamsteering showed a roughly constant improvement in C/N_0 across all of the data. The data is also analogous to the simulated data shown in Section 6.2. There was a course ambiguity for vector tracking beamsteering at the beginning of the PRN 19 and PRN 14 data. The C/N_0 decreased initially and then locked to the correct course at approximately 80 seconds. This was determined to be a vehicle attitude estimation problem and does not accurately reflect

the receiver performance using vector tracking with beamsteering. After the initial problem, vector and scalar beamsteering show the appropriate C/N_0 improvements throughout the run with vector tracking coupled beamsteering out performing scalar tracking coupled beamsteering at signal power outages. The same can be said about standard vector tracking compared to standard scalar tracking throughout the run. All C/N_0s did not contradict the Ublox F9 measured C/N_0s, showing similar performance to the F9 for the single antenna algorithms. The drop in C/N_0 at the 580 second mark is the direct cause of an underflow in the IQ recording, where the M.2 drive skipped samples, and all channels on all receivers lost lock.

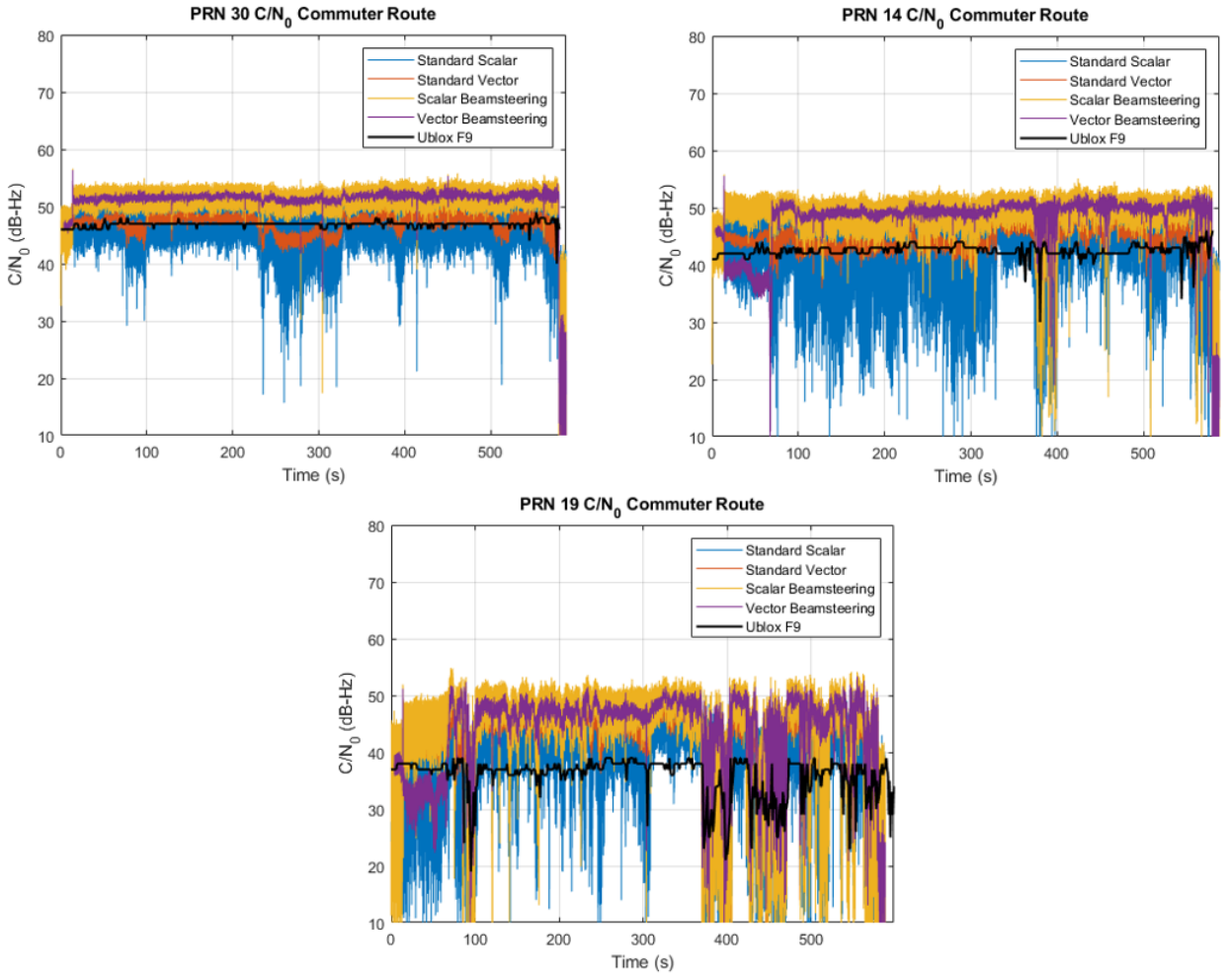

Figure 6.31: C/N_0s from Commuter Route

The other SV C/N_0 plots yielded similar performance to Figure 6.31, but those plots are not shown as they do not add anything in terms of performance characteristics. Pseudorange and pseudorange rate improvements from all routes in this chapter are plotted in Appendix D. The following section shows the positioning improvements from the route.

Positioning Improvements

For the open sky route, positioning improvements were marginal, but there were still improvements in the data. Figure 6.32 shows a comprehensive view of the route with the four different receivers positions being plotted. The Ublox F9 positions are also plotted as a dashed line for reference. The plot was created using google maps and MATLAB in post-processing.

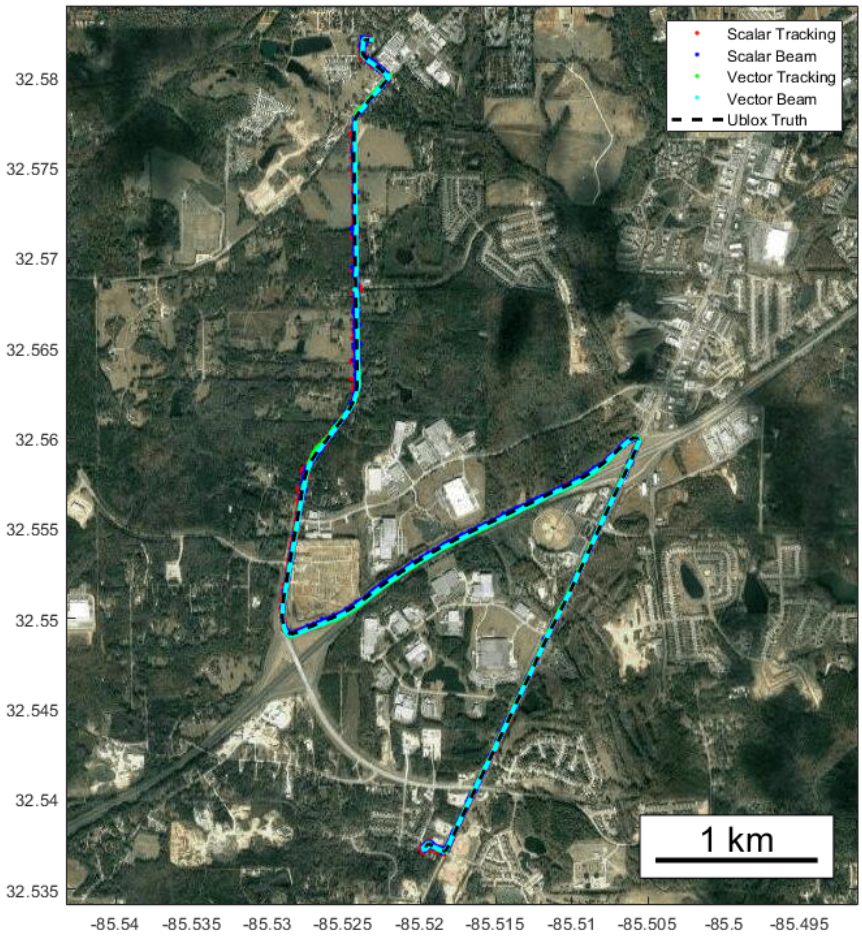

Figure 6.32: Commuter Route Position Results (Wide View) [57]

Figure 6.32 shows that all receiver architectures successfully positioned throughout the whole run. None of the receivers lost lock until the underflow at the end of the run, and that data was cropped from this plot to show the actual performance of the receivers. Figure 6.33 shows a zoomed in portion of the route on South College Street. This plot shows that both scalar tracking and scalar tracking with beamsteering show similar performance in variance and seem unbiased. It is more difficult to see the scalar receiver performance on the plot, but the performance will be analyzed statistically later in this section. Standard single antenna vector tracking shows descent performance but is slightly bias from the Ublox F9, which is considered "truth" throughout this analysis. Vector tracking with beamsteering is unbiased during the route segment and aligns with the F9 positions exceptionally.

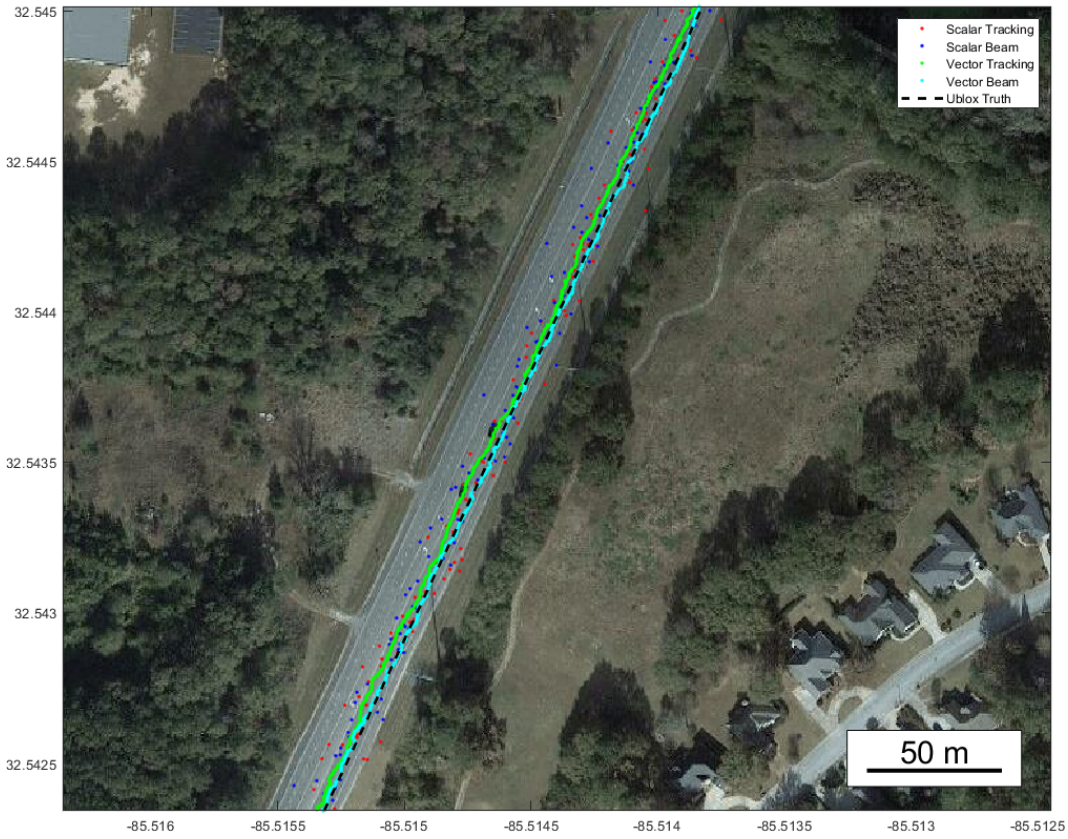

Figure 6.33: South College Positioning Segment [57]

Figure 6.34 shows the same performance crossing the bridge over the interstate as was shown in Figure 6.33. Unfortunately, the position solution degraded when higher dynamics were introduced when the vehicle proceeded onto the interstate. A noticeable bias can be observed by both vector tracking and vector tracking with beamsteering. However, the vector tracking bias is greater than that of vector tracking with beamsteering. This bias can be attributed to the tuning of the vector tracking loop. Vector tracking with beamsteering improves measurement quality. Therefore, since vector tracking with beamsteering has a smaller bias than standalone vector tracking, the bias is attributed to vector tracking over weighting poor measurements arriving at the receiver.

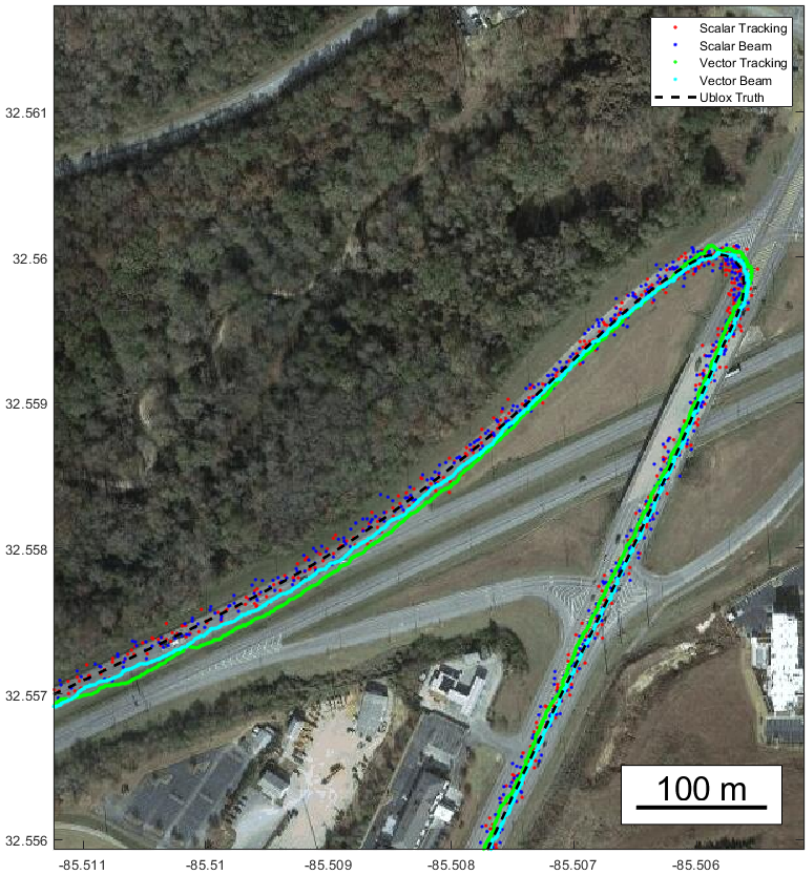

Figure 6.34: Interstate Dynamic Positioning Segment [57]

To analyze the differences, the LLA positions were converted to ENU, and the positions were differenced with the Ublox reference positions to measure position error relative to a COTS receiver. Figure 6.35 shows the position error analysis. Standard scalar tracking and scalar tracking with beamsteering performed relatively similar. The solutions for both scalar tracking algorithms were initially offset from the "truth" position recorded by the Ublox F9. After about 20 seconds, both scalar positions aligned with the Ublox F9. Both vector tracking algorithms suffered due to the scalar algorithm misalignment. Vector tracking relies on scalar tracking to obtain an initial position solution to initialize the vector tracking EKF, and the initial position solution was off by 10 meters in the East direction. The vector beamsteering receiver is still able to perform considerably well until the dynamic portion where the vehicle drove onto the interstate. Vector beamsteering then created a

bias and struggled to maintain an aligned solution for the rest of the run. Standard vector tracking showed similar characteristics to vector beamsteering. However, the errors were higher because the standard vector tracking algorithm did not have the improved signal quality that vector beamsteering created. The biases could be reduced by tightening the process noise filter parameters. Tighter process noise filter parameters (i.e. low Q) provide less correction with the distorted measurements from low elevation, semi-blocked signals.

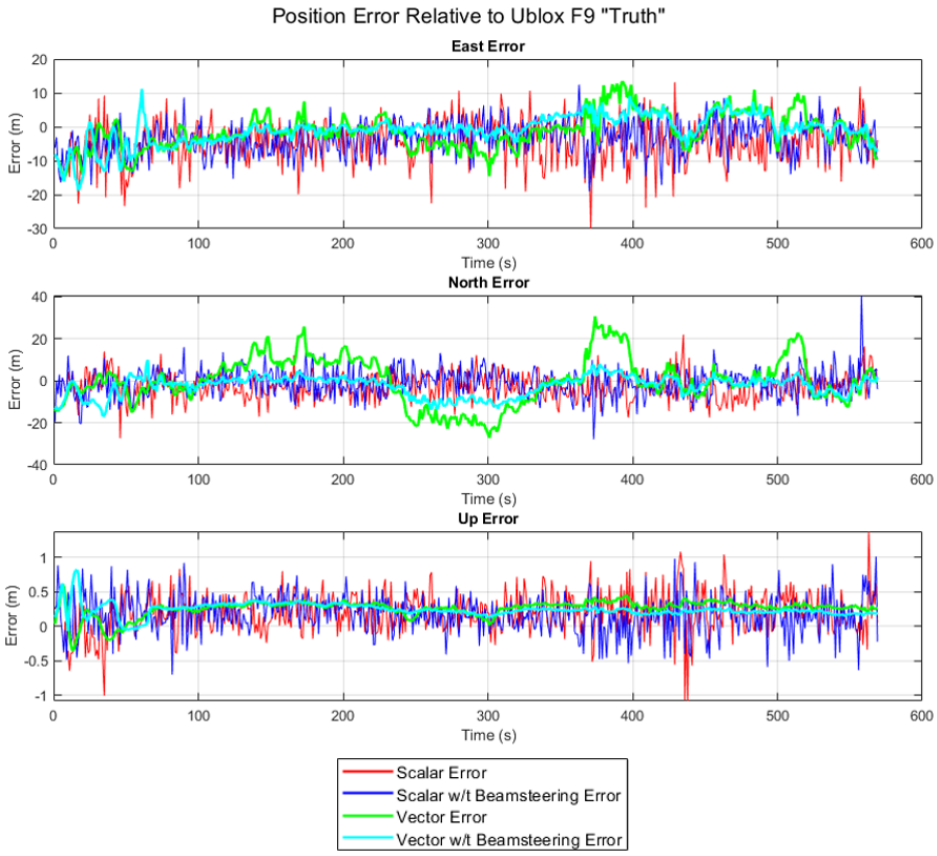

Figure 6.35: Position Error (Relative to Ublox F9) from Commuter Route

The statistics from Figure 6.35 are shown in Table 6.3. The left three columns show the mean error relative to the Ublox F9 in ENU, and the right columns show the 1σ error relative to the Ublox F9 in ENU. From the data, scalar beamsteering performed slightly better than scalar tracking, which is expected because the beamsteering aspect creates better signal quality for the receiver. The mean and standard deviation of the error in the up direction is

minimal across the receivers. This is due to the relative nature to the Ublox F9. Just because the difference between the Ublox F9 and the receivers tested is small does not conclude that the overall position error in the up direction is small. This is merely a comparison to a reputable COTS receiver.

Vector tracking struggled with the large initial bias as evident by the standard vector tracking north error being high. As for all other directions, vector tracking performed as well as scalar beamsteering, if not better. The vector beamsteering receiver performed better than all of the receivers tested, even after the initial bias and dynamics caused problems with its position solution.

Table 6.3: Position Error Statistics (Relative to Ublox F9) from Commuter Route

	μ_E	μ_N	μ_U	σ_E	σ_N	σ_U
Scalar	5.991	5.549	0.286	4.727	4.129	0.221
Scalar Beamsteering	**4.882**	**5.275**	**0.263**	**3.602**	**4.212**	**0.195**
Vector	4.367	8.688	0.274	3.235	7.041	0.084
Vector Beamsteering	**3.181**	**3.383**	**0.245**	**3.071**	**3.749**	**0.089**

With the significant biases in vector tracking and scalar tracking performing relatively well in the open sky environment, the benefits of vector tracking over scalar tracking are not entirely evident. The next route highlights the benefits of scalar tracking with multipath present throughout the route.

6.3.4 Multipath Environment

The multipath route was shorter than the commuter route in the previous section. The route starts static in the Auburn University RV lot, a common tailgating location for football game days. The route then proceeds North toward Jordan Hare Stadium (hence, the name Jordan Hare route), passing tall resident hall buildings and a parking deck. The route is shown in Figure 6.36.

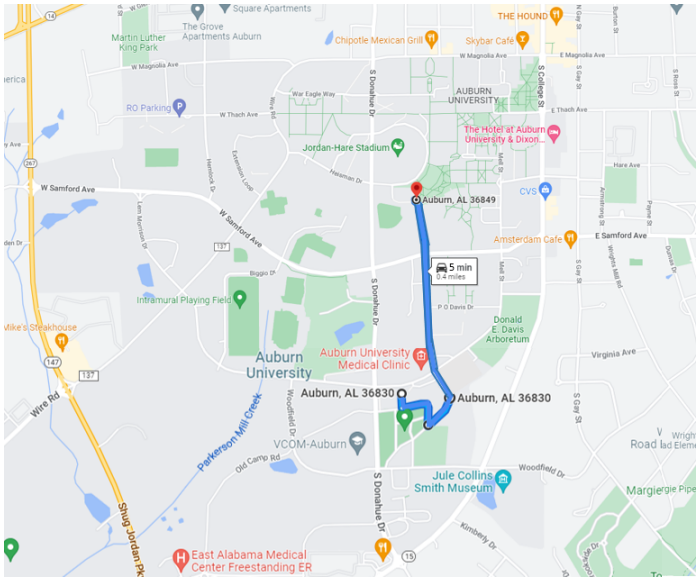

Figure 6.36: Google Maps Jordan Hare Route [57]

Usually, a few buildings and a parking deck would not cause detrimental effects, and the receiver would just see a decrease in signal quality while the vehicle passes the buildings. However, the satellite geometry during the Jordan Hare route was not ideal. Figure 6.37 shows the signals collected by the receiver.

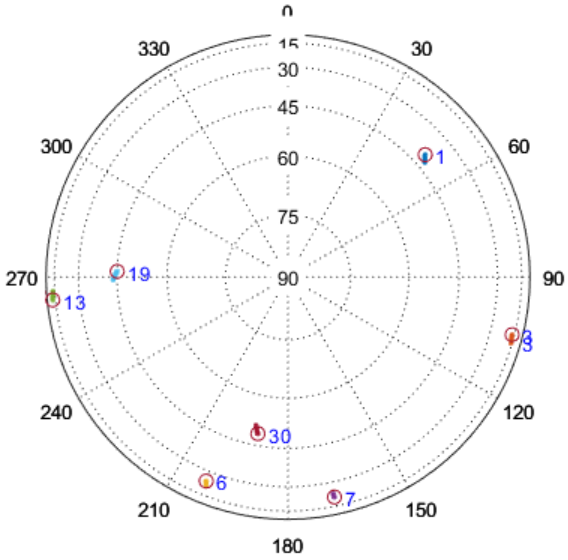

Figure 6.37: Sky Plot Jordan Hare Route

When all satellites are present, the PDOP for the scenario is low at 2.16. However, notice the SV elevations during the route. Only three satellites have descent elevations, and all other signals are at low elevations. With this unfortunate geometry, even small buildings can cause a simple receiver to struggle. The Ublox F9 acquired and tracked two additional satellites giving it better dilution of precision. The failure of the software receivers to acquire the additional signals was not corrected, lending to a comparison of the software receivers in a less than ideal scenario.

C/N_0 Improvements

Four signal channels were evaluated for their C/N_0 performance during this route. SV 30 was the highest elevation satellite at 50 degrees. SV 1 was a mid-level satellite in this scenario at 40 degrees. SVs 13 and 3 were at the lowest elevations, at or below 15 degrees. Figure 6.38 shows the resulting C/N_0s. SV 30 shows clean performance because the line of sight (LOS) is maintained throughout the route. All other SVs showed blockages in their LOS, causing the receivers to experience considerable multipath. For the low elevation satellites, the C/N_0 measurements turn to noise after approximately 200 seconds for the scalar receivers indicating a loss of lock, which will become evident in the position solution later.

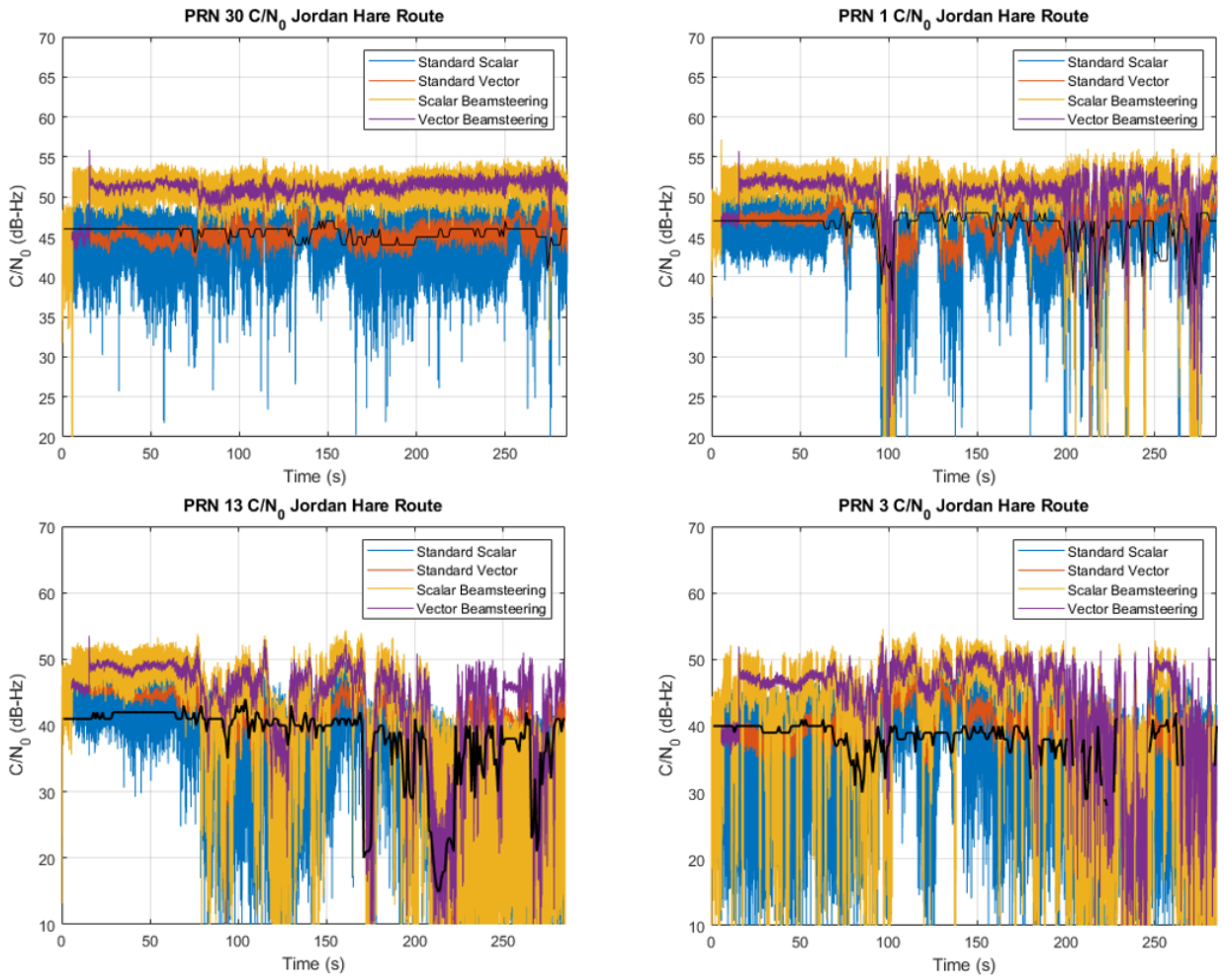

Figure 6.38: C/N_0 from Jordan Hare Route

The performance of the receivers aligned with the Ublox F9 reported C/N_0s for the entire scenario. The Ublox F9 reports a loss of lock on PRN 3 toward the end of the route, as shown by the discontinuities in the black line.

Positioning Improvements

A wide view of the positioning results for the Jordan Hare route is shown in Figure 6.39. The scalar tracking and scalar beamsteering algorithms struggle throughout the route and eventually lose lock as the visible satellites drop to three. Note the errors in the scalar beamsteering solution in the route turns. This is attributed to a course update error in the

beamsteering algorithm causing the beam to lag when the vehicle turns around a corner. It was not present on the previous route.

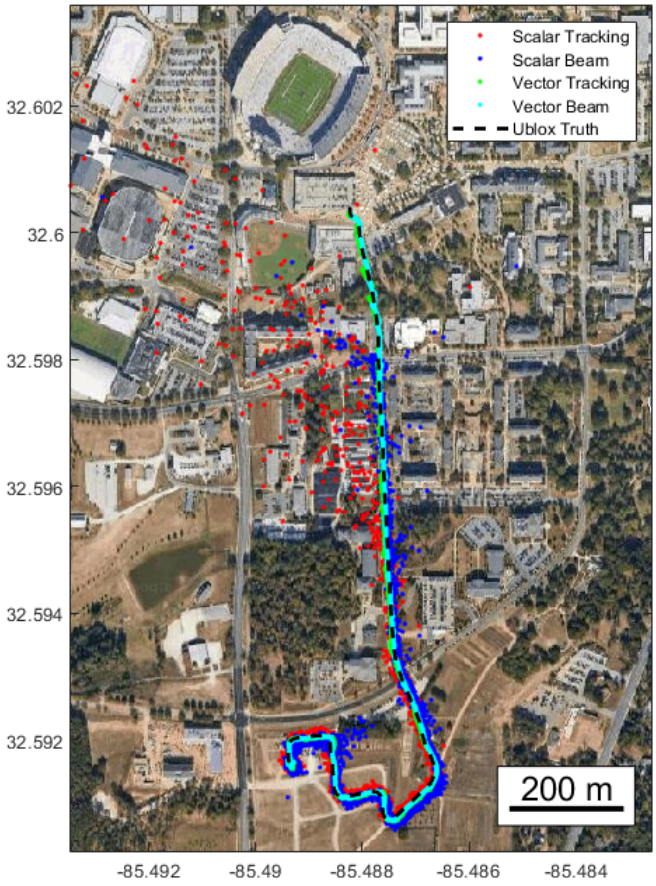

Figure 6.39: Jordan Hare Route Position Results (Wide View)

By zooming in on the position solution where the scalar receivers lose position lock, it can be observed that the scalar beamsteering receiver was able to maintain lock longer than the standard scalar tracking receiver, as shown in Figure 6.40. The scalar beamsteering receiver diverged while passing through the intersection with two buildings and some tree cover. Vector tracking and vector tracking beamsteering maintained lock but did show some multipath effects on their position solutions. Vector beamsteering aligned more closely to the Ublox F9 reported position than standard vector tracking in this view.

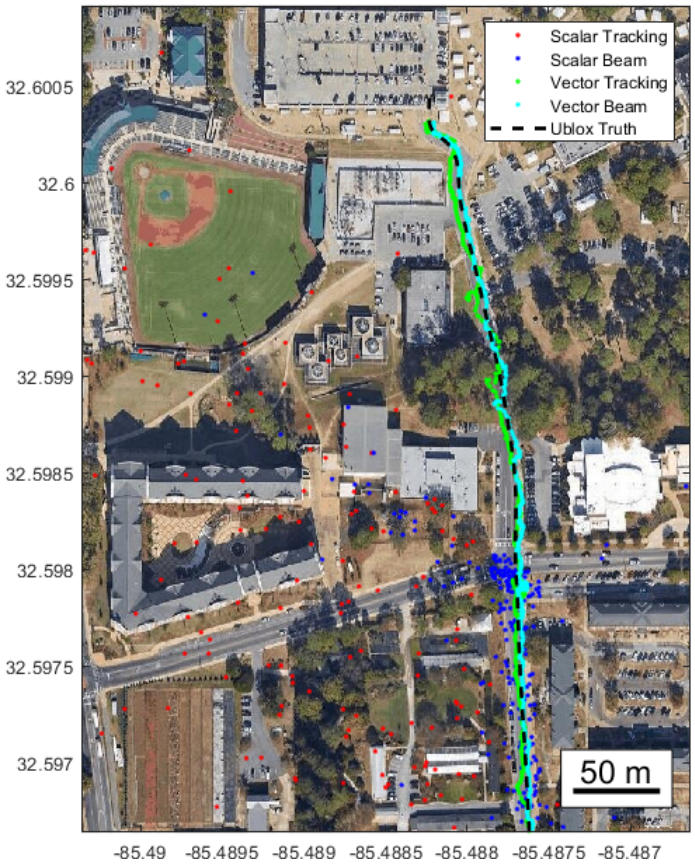

Figure 6.40: Jordan Hare Multipath

The absolute position error relative to the Ublox F9 is shown in Figure 6.41 to see how closely the receivers align with a COTS receiver. The loss of lock in scalar tracking is shown with the 15000 meters of error in the north and east directions. By zooming in, it can be observed that vector tracking with beamsteering receiver outperformed all of the receivers being tested. Even with the poor course updates in the scalar beamsteering receiver, it also shows better performance than the single antenna scalar receiver because it maintains lock for a longer period of time, with quantifiable error in the east and up directions. Vector tracking and vector beamsteering show clear tracking through the multipath to the end of the route.

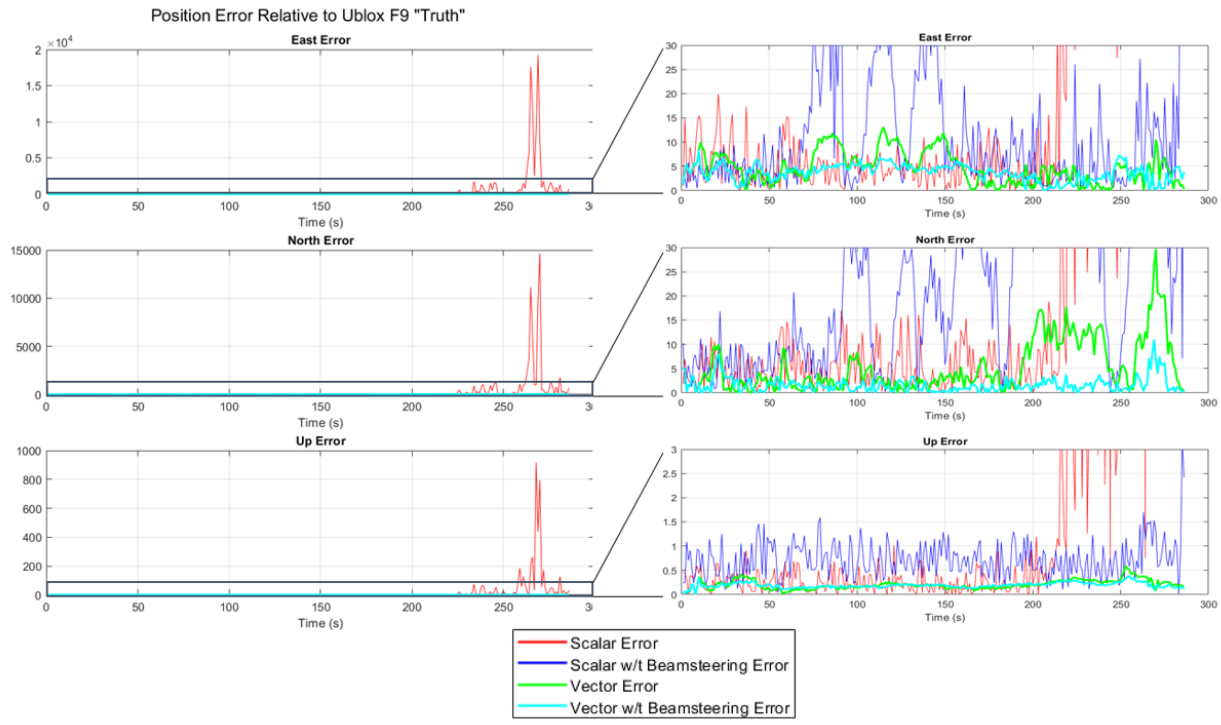

Figure 6.41: Absolute Position Error (Relative to Ublox F9) from Jordan Hare Route

The absolute position error statistics relative to the COTS receiver are shown below in Table 6.4. From top to bottom, each preceding receiver outperforms the previous. The scalar tracking receiver fails catastrophically. The scalar beamsteering receiver fails but performs better than the standard scalar tracking receiver. The vector tracking receiver showed some biases from the Ublox F9 solution, but it survived all of the multipath. The vector tracking beamsteering receiver performed very well considering the signal geometry and the multipath present in the scenario.

Table 6.4: Position Error Statistics (Relative to Ublox F9) from Jordan Hare Route

	μ_E	μ_N	μ_U	σ_E	σ_N	σ_U
Scalar	367.513	253.490	15.570	1865.476	1315.295	80.561
Scalar Beamsteering	**11.400**	**25.129**	**0.725**	**16.410**	**22.487**	**0.384**
Vector	4.670	5.248	0.204	3.263	5.301	0.095
Vector Beamsteering	**3.704**	**1.632**	**0.192**	**1.694**	**1.587**	**0.054**

6.4 Interference Injection Testing

With the vector beamsteering receiver outperforming the other receivers being tested, the last scenario involved overlaying jamming data on the commuter route to test the performance of the receivers in a simulated contested environment. The route with simulated emitters is shown in Figure 6.42. The emitters were activated when the vehicle exited the interstate and turned north. Before the emitters were activated, the vehicle drove through a natural environment, as shown in Section 6.3.3. The emitters consisted of three ground-based wideband noise jammers equally spaced in line of bearing with one emitter facing due south. Once activated, the jammers ramped in power, knocking off all receivers as the C/N_0s dropped. The figure shows that the vector tracking receiver and the vector tracking beamsteering receiver last longer than their scalar counterparts, but eventually, they too fail.

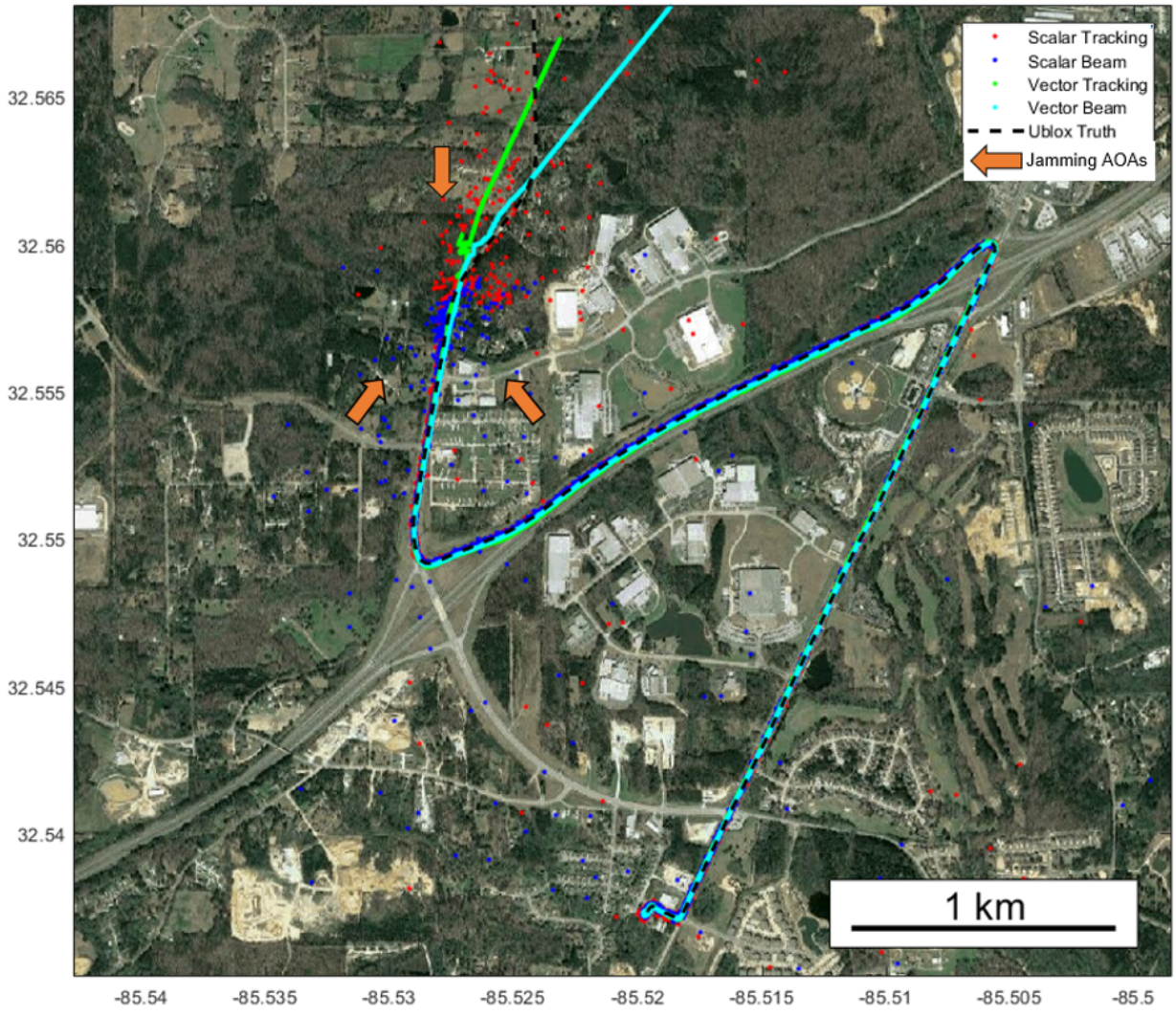

Figure 6.42: Jamming Injected into Commuter Route

6.4.1 C/N_0 Response

When the interference signals ramp in power, the receiver carrier-to-noise ratio is degraded. Figure 6.43 shows the ramp jamming effects on the high, mid, and low elevation satellites. Even though the jamming signals are ground based, all signals are affected at approximately the same time in nearly the same way. Vector tracking with beamsteering performs better than all of the other algorithms, but it does not last much longer than standard vector tracking.

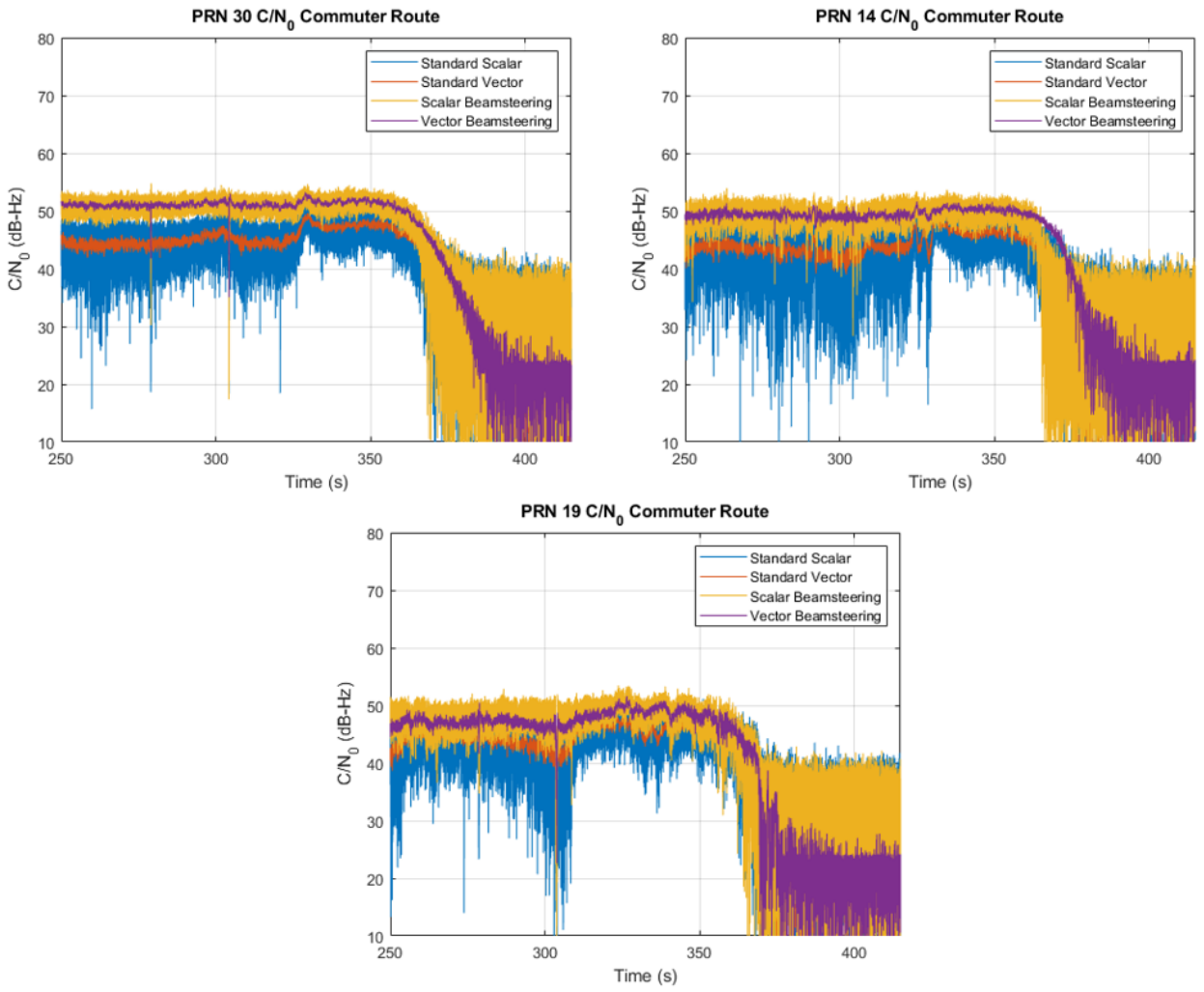

Figure 6.43: C/N_0 from Jamming Injection

Beamsteering improves signal quality, thereby, improving the receiver resilience to interference. Beamsteering also manipulates the CRPA perceived gain pattern reducing gain in non-desired directions as the beam is directed to the target. The improvements in Figure 6.43 are better observed when analyzing vector lock, which was outlined in Section 5.4.

6.4.2 Vector Lock Improvements

The maximum range error defines vector lock or vector health. Figure 6.44 shows the vector lock analysis on the left and the measured signal received power on the right. Vector lock is assumed to be unstable when the 1σ maximum range error exceeds 1/6th of a chip,

or approximately 50 meters. The 50 meter mark is not shown with the vector lock plot for better plot scaling, but both the vector tracking receiver and the vector tracking receiver with beamsteering exceeded the 50 meter threshold. Since vector beamsteering merely scales the signal power, improving data at the receiver, it can be determined that the difference in divergence in the vector tracking and vector beamsteering receiver will be relatively the same from scenario to scenario.

From the vector lock plot in Figure 6.44, it can be concluded that the vector beamsteering receiver lasted about five seconds longer than the standard vector tracking receiver in the jamming scenario. The overall signal receive power incident on the receiver was measured and shown in the right most figure. From this plot, the ramp in the jamming signals are evident as the power is increased every second. From the linear portion of signal power between 370 and 380 seconds, it can be concluded that the vector tracking beamsteering receiver lasting five more seconds in the scenario implies vector tracking beamsteering can handle a jamming signal approximately four dBm higher than the standard vector tracking receiver.

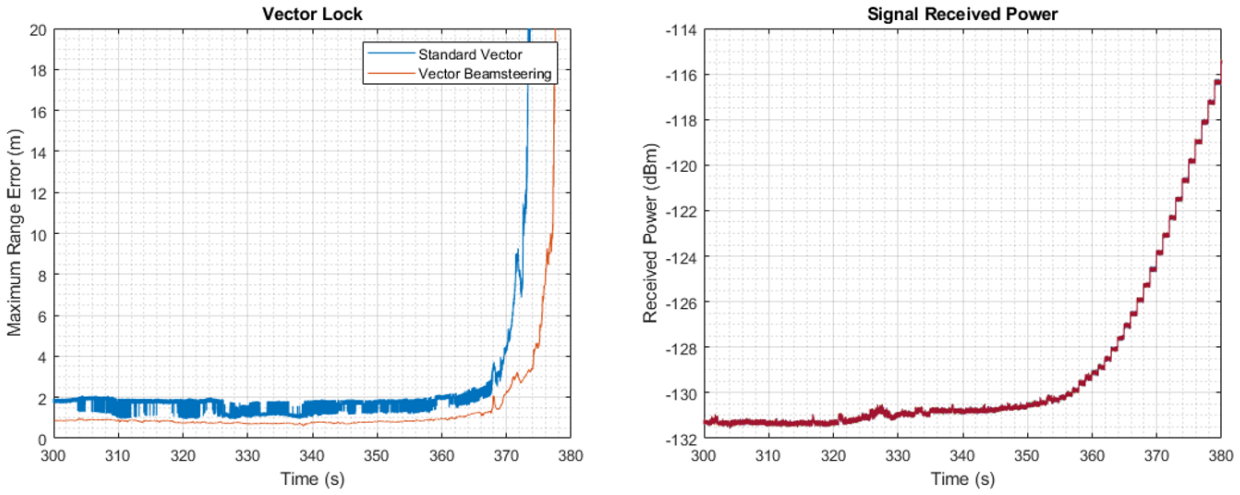

Figure 6.44: Vector Lock with Received Power Comparison

This is an intuitive result since the beamsteering algorithm increases satellite signal receive power. The resilience of the receiver is directly related to the signal quality, so as long as the signal quality is improved, the receiver will perform better with satellite beamsteering. If the jamming signals are incident at the same AOA as the satellite signals, the receiver would not see any improvement in resilience. However, the main assumption is that most interference signals are emitted from the ground making beamsteering yield favorable results since the interference signals were outside of the formed beam lobe.

A constrained nullsteering algorithm could be implemented for better performance. The receiver could leverage satellite positions as constraints to maintain power in the SV directions, and then the algorithm could actively mitigate all signals out of the SV constraints. A crude implementation of this algorithm was attempted for this work. However, the algorithm was unsuccessful due to shortcomings in the calibration process defined initially at the beginning of this chapter. Since the beamsteering lobes are wider in beam width and do not require the calibration accuracy that direction finding algorithms and deterministic nulling algorithms require, the results in this thesis were minimally affected.

6.5 Conclusions

This chapter deduced from the simulations that increasing the number of antenna elements increases received signal quality. However, the signal quality scale does not directly relate to the number of antenna elements, as geometry plays a role. As the number of elements increase, the signal quality improvement rate tends to decrease, because the wavelength of the incident signal constrains the antennas and practically, only a certain number of elements can be placed within that constraint.

The beamsteering algorithm was then ported to live sky data. The live sky analysis showed the vector tracking beamsteering receiver outperforming the other receiver designs, aligning somewhat closely to the COTS Ublox F9. The receiver performance gains using the four element CRPA matched the performance gains deduced from simulation in the

early section. Overall, position error decreased when using the vector tracking beamsteering algorithm, which was shown in simulation and live sky analysis.

Since signal quality improves when the receiver beamsteers to satellite signals, the improvement directly affects interference performance, which was shown in the last section. The improvements were related to the SV signal power increase as long as the interference signals are outside of the beam width. Since most are ground-based, and SV signals are incident at higher elevations, this assumption can be made to improve receiver performance in most scenarios.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis analyzed four different receiver configurations including a multi-antenna vector tracking beamsteering receiver, which was based on a previous high-gain advanced GPS receiver (HAGR) developed by NAVSYS [1] and a multi-antenna vector tracking receiver created by the German Aerospace Center (DLR) [20]. A beamsteering receiver was created applying the HAGR concepts to a vector tracking receiver similar to the DLR receiver, successfully coupling CRPA technology with an asynchronous vector tracking receiver. Unlike the DLR receiver, the multi-antenna vector tracking receiver feeds satellite geometry back to a weighting module. The weighting module then combines the weighted signals before correlators are formed, airing more toward a low SWaP-C design with less resources allocated in multiple correlator streams. The receiver also implements and presents some low SWaP-C correlator based C/N_0 estimation techniques, leveraging a PRN 33 FMET reduced correlator algorithm to reduce computational load on noise characterization.

The multi-antenna vector tracking beamsteering receiver performance was compared to three other software-defined receivers: a single antenna scalar tracking receiver, a multi-antenna scalar tracking beamsteering receiver, and a single antenna vector tracking receiver. From the comparative analysis of all four receivers, the receivers performances ranked as follows: the single antenna vector tracking receiver performed the worst, followed by the multi-antenna scalar tracking beamsteering receiver. The single antenna vector tracking receiver and the multi-antenna vector tracking receiver performed noticeably better than their scalar counterparts in simulation and in live sky testing, especially when signal quality was degraded in the multipath and the interference scenarios.

The simulation section showed how multiple antennas and beamsteering can be combined to create a robust receiver, improving C/N_0 signal quality, which improves position estimates. The simulation section showed how the scalar and vector implementations performed similarly for signals at high power. Once the signals were degraded in simulation, the vector tracking receivers were able to maintain lock while the scalar tracking receivers failed. All of the vector tracking receivers were able to successfully track a signal at low power yielding a C/N_0 of 20 dB-Hz. The seven and eight element CRPAs increased the 20 dB-Hz C/N_0 by roughly eight dB-Hz, implying that the receiver coupled with the seven and eight element CRPAs could possibly track signals yielding a 12 dB-Hz C/N_0. This is well below the tracking thresholds previously defined by Matthew Lashley's groundbreaking work [3]. The four element simulated Antcom antenna, which was later used to record live sky signal data, induced a 3-5 dB-Hz C/N_0 improvement on signals depending on their angle of arrival.

The live sky data analysis presented a post processing acquisition phase calibration to remove line-to-line induced phase delays. The performance of this phase calibration algorithm was not fully realized since an array manifold calibration was not performed, but the acquisition phase calibration is a formidable method for line-to-line phase delay removal. A simple course update was applied instead of using an attitude update. The course update was derived from the receiver velocity. A CRPA, signal conditioning box, and a USRP X310 front-end were required for recording live sky data. No additional sensors were required, but an alternative sensor is recommended for the attitude update if the receiver is going to be subject to contested environments.

Two dynamic live sky data routes were collected, and each receiver processed the data for analysis. For the open sky route with good satellite geometry, the vector tracking receivers were initialized with biases greater than 10 meters in position, which caused problems that the multi-antenna vector tracking receiver was able to overcome. Unfortunately, the single antenna vector tracking receiver struggled to perform equivalently with the multi-antenna

scalar tracking receiver. Of the four receivers, the vector beamsteering receiver offered the best performance with almost a three meter improvement in the east and north direction in position accuracy relative to a Ublox F9 COTS receiver. For the multipath route with poor satellite geometry, the multi-antenna vector tracking receiver offered exceptional performance compared to the three other receiver designs being analyzed. Both of the scalar designed receivers lost lock and failed in the scenario, with the multi-antenna beamsteering receiver surviving longer than the single-antenna scalar receiver. The vector tracking receivers were resilient through the entire scenario, with the multi-antenna vector beamsteering receiver offering an improvement of one meter in east position and 3.6 meters in north position compared to the single-antenna vector tracking receiver, relative to the Ublox F9 receiver.

For more analysis, interference was added to the open sky data before the data was processed by the four receivers. The interference consisted of 3 ramped wideband noise jammers. The time at which the vector tracking receivers lost lock was logged for analysis. Using the principles of vector lock defined by Ben Clarke [52], it was determined that the four antenna vector tracking beamsteering receiver was able to last approximately five seconds longer than the single-antenna vector tracking receiver. With the jamming power logged, this time difference was linked to the receiver surviving a four dB increase in jamming power compared to its single antenna counterpart.

Overall, both hardware and software experiments validated the design of the multi-antenna vector tracking beamsteering receiver and presented it as a formidable receiver architecture. If the application is resilient and precise positioning in the presence of multipath or poor signal quality caused by natural causes then this receiver will work well. However, better CRPA algorithms exist for interference removal that improve receiver resilience in jamming environments. Although it was not tested in this work, the receiver should perform well against low power spoofing attacks. In these attacks, improving GPS signal power could help a receiver maintain lock on the appropriate signals. If computational management is a concern, this receiver might be a good choice due to its simplistic implementation (if vector

tracking can be considered simple) and cutting edge techniques for estimation. From a positioning standpoint, the multi-antenna vector tracking beamsteering receiver performed quite well, highlighting the benefits of CRPA coupling with advanced vector tracking receiver architectures.

7.2 Future Work

The results from the proposed multi-antenna vector tracking receiver analysis showed several areas for potential research and improvement. First, collecting data from dense urban environments might further shed light on the performance of the multi-antenna high gain receiver. Though the poor geometry multipath environment offered problems for the simpler receivers, a dense city might offer problems for a COTS receiver that the multi-antenna vector tracking beamsteering receiver may be able to survive. A data collection in urban Atlanta has already been considered as a plan of action.

The receiver performance in a spoofing environment also needs to be analyzed since spoofing signals are at nearly the same power as the GPS signals. Adding a bit more power the GPS satellites might help the receiver achieve formidable AS performance.

Since the beamsteering attachment is modular and adaptable, improvements or modifications in the beamforming algorithm could be adopted depending on the receiver applications. A constrained nullsteering algorithm could be used to improve the receiver performance in a jamming environment. The feedback of satellite geometry would still be maintained, but array manifold data would have to be recorded for this implementation. Array manifold data can be obtained by manufacturers, calibration companies, or in the do-it-yourself form if access to an anechoic chamber is available. Obtaining array manifold data would open the doors for many other algorithm adaptations that would be interesting in research and application.

Implementing this work with a deeply integrated (DI) vector tracking GPS/INS receiver would yield an incredibly resilient receiver to multipath in urban environments. The

IMU would offer an attitude update to the beamforming algorithm, and the multi-antenna beamsteering would offer better signal quality to the already pertinent position solution a DI receiver offers. Other approaches might be related more to the DLR implementation where the beamforming is applied at the correlator level. In this implementation, there are possibilities of other algorithms that might offer better performance in certain situations.

Bibliography

[1] A. Brown and N. Gerein, "Test results from a digital p(y) code beamsteering GPS receiver designed for carrier-phase time transfer," in *Proceedings of the 57th Annual Meeting of The Institute of Navigation*, E. Powers, Ed., Albuquerque, NM, Jun. 2001, pp. 872–878.

[2] M. Cuntz, A. Konovaltsev, C. Kappen, and M. Meurer, "Vector tracking with a multi antenna GNSS receiver," in *Proceedings of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Nashville, TN, Sep. 2012, pp. 2050–2056.

[3] M. Lashley, "Modeling and performance analysis of GPS vector tracking algorithms," Thesis, Auburn University, Auburn, AL, 2009, 391 pp.

[4] Pratap Misra and Per Enge, *Global Positioning System Signals, Measurements, and Performance*, Revised Second Edition. Jan. 2011.

[5] P. Sigal, "Mercedes hopes to roll out level 3 self-driving in u.s. this year, CEO says," *Automotive News Europe*, Feb. 24, 2022.

[6] T. E. Humphreys, B. M. Ledvina, V. Tech, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," *Proceedings of ION GNSS*, p. 12, 2008.

[7] R. Mitch, R. Dougherty, M. Psiaki, S. Powell, B. O'Hanlon, J. Bhatti, and T. Humphreys, "Signal characteristics of civil GPS jammers," in *Proceedings of the 24th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Portland, OR, Sep. 2011, pp. 1907–1919.

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

[9] Karl Ferdinand Braun, "Electrical oscillations and wireless telegraphy," *Nobel Lecture*, Dec. 11, 1909.

[10] O. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972.

[11] K. Gold, A. Brown, and K. Stolk, "Bistatic sensing and multipath mitigation with a 109-element GPS antenna array and digital beam steering receiver," in *Proceedings of the 2005 National Technical Meeting of The Institute of Navigation*, San Diego, CA, Jan. 2005, pp. 849–857.

[12] D. Reynolds, A. Brown, and A. Reynolds, "Miniturized GPS antenna array technology and predicted anti-jam performance," in *Proceedings of the 12th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Nashville, TN, Sep. 1999, pp. 777–786.

[13] A. Brown and B. Matthews, "Constrained beamforming for space GPS navigation," in *Proceedings of the 20th International Technical Meeting of the Satelite Division of The Institute of Navigation*, Fort Worth, TX, Sep. 2007, pp. 2357–2363.

[14] J. Seo, Y.-H. Chen, D. S. De Lorenzo, S. Lo, P. Enge, D. Akos, and J. Lee, "A real-time capable software-defined receiver using GPU for adaptive anti-jam GPS sensors," *Sensors*, vol. 11, no. 9, pp. 8966–8991, 2011.

[15] J. Spilker, "Vector delay lock loop processing of radiolocation transmitter signals," pat. 5,398,034, Mar. 1995.

[16] M. Lashley and D. M. Bevly, "Analysis of discriminator based vector tracking algorithms," in *Proceedings of the 2007 National Technical Meeting of The Institute of Navigation*, San Diego, CA, Jan. 2007, pp. 570–576.

[17] M. Lashley and D. Bevly, "Comparison in the performance of the vector delay/frequency lock loop and equivalent scalar tracking loops in dense foliage and urban canyon," in *Proceedings of the 24th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Portland, OR, Sep. 2011, pp. 1786–1803.

[18] S. Zhao and D. Akos, "An open source GPS/GNSS vector tracking loop - implementation, filter tuning, and results," *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*, p. 13, 2011.

[19] D. Benson, "Interference benefits of a vector delay lock loop (VDLL) GPS receiver," in *Proceedings of the 63rd Annual Meeting of The Institute of Navigation*, Cambridge, MA, Apr. 2007, p. 8.

[20] M. Cuntz, A. Konovaltsev, and M. Meurer, "Concepts, development, and validation of multiantenna GNSS receivers for resilient navigation," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1288–1301, Jun. 2016. (visited on 09/26/2021).

[21] M. Cuntz, L. Greda, M. Heckler, A. Konovaltsev, M. Meurer, L. Kurz, G. Kappen, and T. G. Noll, "Lessons learnt: The development of a robust multi-antenna GNSS receiver," in *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation*, Portland, OR, 2010, pp. 2852–2859.

[22] S. Thompson, S. Martin, and D. Bevly, "Single differenced doppler positioning with low earth orbit signals of opportunity and angle of arrival estimation," presented at the 2021 International Technical Meeting of The Institute of Navigation, Online, Feb. 23, 2021, pp. 497–509. (visited on 12/05/2021).

[23] R. A. Flores and R. C. Tiznado, *NAVSTAR GPS space segment/navigation user segment interfaces (IS-GPS-200)*, May 2020.

[24] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (corresp.)," *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 619–621, Oct. 1967. (visited on 12/07/2021).

[25] W. Luke Edwards, "Developement of a GPS software receiver on an FPGA for testing advanced tracking algorithms," Thesis, Auburn University, Auburn, AL, 2010, 88 pp.

[26] James V. Carrol, "Vulnerability assessment of the US transportation infrastructure that relies on the global position system," *The Journal of Navigation 56.2*, pp. 185–193, 2003.

[27] I. GNSS, "FCC fines operator of GPS jammers that affected newark airport GBAS," Aug. 31, 2013.

[28] *Improving the operation and development of global positioning system (GPS) equipment used by critical infrastructure.*

[29] J. Bhatti and T. E. Humphreys, "Hostile control of ships via false GPS signals: Demonstration and detection," *NAVIGATION*, vol. 64, no. 1, pp. 51–66, 2017.

[30] R. Mit, Y. Zangvil, and D. Katalan, "Analyzing teslas level 2 autonomous driving system under different GNSS spoofing scenarios and implementing connect services for authentication and reliability of GNSS data," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation*, vol. 1, St. Louis, MS, 2020, pp. 621–646.

[31] T. Suzuki, K. Kusama, and Y. Amano, "NLOS multipath detection using convolutional neural network," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation*, St. Louis, MS, Sep. 2020, pp. 2989–3000.

[32] B. A. Black, P. S. Dipiazza, B. A. Ferguson, D. R. Voltmer, and F. C. Berry, *Introduction to Wireless Systems*. Prentice Hall, May 2008, vol. 1, ISBN: 0-13-244789-4.

[33] R. J. Mailloux, *Phase Array Antenna Handbook*, 3rd. Norwood, MA: Artech House Publishers, 2018, ISBN: 978-1-63081-029.

[34] J. P. Gardner, J. C. Mather, M. Clampin, R. Doyon, M. A. Greenhouse, H. B. Hammel, J. B. Hutchings, P. Jakobsen, S. J. Lilly, K. S. Long, J. I. Lunine, M. J. Mccaughrean, M. Mountain, J. Nella, G. H. Rieke, M. J. Rieke, H.-W. Rix, E. P. Smith, G. Sonneborn, M. Stiavelli, H. S. Stockman, R. A. Windhorst, and G. S. Wright, "The james webb space telescope," *Space Science Reviews*, vol. 123, no. 4, pp. 485–606, Apr. 1, 2006, ISSN: 1572-9672. DOI: 10.1007/s11214-006-8315-7. [Online]. Available: https://doi.org/10.1007/s11214-006-8315-7.

[35] L. C. Godara, "Application of antenna arrays to mobile communications," *PROCEEDINGS OF THE IEEE*, vol. 85, no. 8, p. 51, 1997.

[36] Andrew L. Kintz, "Nullspace MUSIC and improved RF emitter geolocation," Dissertation, The Ohio State University, 2016, 282 pp.

[37] Dennis Akos, "A software radio approach to global navigation satellite system receiver design," Dissertation, Ohio University, Aug. 1997.

[38] (Apr. 2021). "Ettus research," [Online]. Available: https://www.ettus.com/products/.

[39] T. Watts, S. Martin, and D. Bevly, "A GPS and GLONASS l1 vector tracking software-defined receiver," Thesis, Auburn University, Auburn, AL, Feb. 19, 2019, 257 pp.

[40] Mark L. Psiaki, "Smoother-based GPS signal tracking in a software receiver," in *Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Salt Lake City, UT, Sep. 2001, pp. 2900–2913.

[41] C. O'Driscoll and G. Lachapelle, "Comparison of traditional and kalman filter based tracking architectures," presented at the European Navigation Conference, Naples, Italy, May 2009.

[42] Elliott D. Kaplan and Christopher Hegarty, *Understanding GPS - Principles and Applications*, 2nd ed. Boston and London: Artech House Publishers, 1996, ISBN: 978-1-58053-894-7.

[43] R. N. Crane, "A simplified method for deep coupling of GPS and inertial data," in *Proceedings of the 2007 National Technical Meeting of The Institute of Navigation*, San Diego, CA, Jan. 2007, pp. 311–319.

[44] M. Z. H. Bhuiyan, S. Soderholm, S. Thombre, L. Ruotsalainen, M. Kirkko-Jaakkola, and H. Kuusniemi, "Performance evaluation of carrier-to-noise density ratio estimation techniques for BeiDou bl signal," in *2014 Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, Corpus Christi, TX: IEEE, Nov. 2014, pp. 19–25.

[45] K. Ramasubramanian and S. Nadig, "Performance bounds for carrier-to-noise ratio estimation in GPS receivers," in *2006 National Technical Meeting of The Institute of Navigation*, Monterey, CA, Jan. 2006, pp. 953–957.

[46] A. Schmid and A. Neubauer, "Carrier to noise power estimation for enhanced sensitivity galileo/GPS receivers," in *2005 IEEE 61st Vehicular Technology Conference*, vol. 4, Stockholm, Sweden: IEEE, 2005, pp. 2629–2633.

[47] E. Falletti, M. Pini, and L. L. Presti, "Low complexity carrier-to-noise ratio estimators for GNSS digital receivers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 420–437, Jan. 2011.

[48] Phillip W. Ward, *Understanding GPS - Principles and Applications*, ser. Mobile Communication Series. In Kaplan, E. D.: Artech House Publishers, 1996, chapter 6, 209-236.

[49] C. A. Givhan, D. M. Bevly, and S. M. Martin, "Performance analysis of a vector tracking software defined receiver for GPS l5," presented at the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), Oct. 28, 2020, pp. 3163–3179.

[50] Scott M. Martin, "GPS carrier phase tracking in difficult environments using vector tracking for precise positioning and vehicle attitude estimation," Dissertation, Auburn University, Auburn, AL, 2017, 240 pp.

[51] Brian A. Keyser, "Design and implementation of a SoC-based real-time vector tracking GPS receiver," Thesis, Auburn University, Auburn, AL, 2015, 101 pp.

[52] Benjamin J. Clark, "Fault detection and exclusion in deeply integrated GPS/INS navigation," Dissertation, Auburn University, Auburn, AL, Dec. 2012, 175 pp.

[53] Robert Brown and Patrick Wang, *Introduction to Random Signals and Applied Kalman Filtering*, Fourth. John Wiley & Sons, 2012.

[54] P. R. Carter, J. Starling, S. Martin, and D. Bevly, "Baseline impact on geolocation," presented at the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019), Miami, Florida, Oct. 11, 2019, pp. 2588–2597.

[55] M. Sgammini, F. Antreich, and M. Meurer, "SVD-based RF interference detection and mitigation for GNSS," p. 9,

[56] R. Powell, "A multiple-antenna software GPS signal simulator for rapid testing of interference mitigation techniques," Masters, Auburn University, Auburn, AL, Mar. 7, 2017, 143 pp. [Online]. Available: `https://www.ion.org/publications/abstract.cfm?articleID=14922` (visited on 03/12/2022).

[57] *Commuter route*, 2022. [Online]. Available: `https://www.google.com/maps/@32.5774681,-85.5066981,13.54z`.

[58] S. Burchfield, S. Martin, D. Bevly, and J. Starling, "Performance analysis of low SWaP-c jamming mitigation methods for commercial applications," presented at the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), Oct. 28, 2020, pp. 3592–3611.

[59] J. R. v. d. Merwe, A. Rügamer, F. Garzia, and W. Felber, "Wavelet based adaptive notch filtering to mitigate COTS PPDs," presented at the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019), Miami, Florida, Oct. 11, 2019, pp. 3285–3299.

[60] J. Wendel, F. M. Schubert, A. Rügamer, and S. Taschke, "Limits of narrowband interference mitigation using adaptive notch filters," presented at the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), Portland, Oregon, Nov. 8, 2016, pp. 286–294.

[61] J. Starling, "Error analysis of carrier phase positioning using controlled reception pattern antenna arrays," Ph.D. dissertation, Auburn University, 90 pp.

[62] D. S. De Lorenzo, "Navigation accuracy and interference rejection for GPS adaptive antenna arrays," Ph.D. dissertation, Stanford University, 2007, 115 pp.

[63] J. Glover, "Adaptive noise canceling applied to sinusoidal interferences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 6, pp. 484–491, Dec. 1977.

[64] S. A. Hadei and M. lotfizad, "A family of adaptive filter algorithms in noise cancellation for speech enhancement," *International Journal of Computer and Electrical Engineering*, pp. 307–315, 2010.

[65] S. Applebaum, "Adaptive arrays," *IEEE Transactions on Antennas and Propagation*, vol. 24, no. 5, pp. 585–598, Sep. 1976.

[66] B. Widrow and S. D. Stearns, *Adaptive signal processing*. Prentice-Hall PTR, 1985, 474 pp., ISBN: 0-13-004029-0.

[67] Thomas Dean Moore, "Analytic study of space-time and space-frequency adaptive processing for radio frequency interference suppression," Dissertation, Ohio State University, 2002, 209 pp.

[68] I. Gupta and T. Moore, "Space-frequency adaptive processing (SFAP) for radio frequency interference mitigation in spread-spectrum receivers," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 6, pp. 1611–1616, Jun. 2004.

[69] J. Perez-Solano, S. Felici-Castell, and M. Rodriguez-Hernandez, "Narrowband interference suppression in frequency-hopping spread spectrum using undecimated wavelet packet transform," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1620–1629, May 2008.

[70] C. Vonesch, T. Blu, and M. Unser, "Generalized daubechies wavelet families," *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4415–4429, Sep. 2007.

Appendices

Appendix A

Deterministic Nullsteering

Nullsteering has the opposite effect of beamsteering. Instead of aligning antenna power toward a signal, nullsteering attempts to remove all antenna power in a certain direction. By carefully manipulating the phases of each element, the signal can be attenuated by creating destructive interference in the boresight direction. For a four element antenna, three of the signal phases can be shifted by one third wavelength relative to the reference element phase to achieve nullsteering. This is shown by Equation (A.1).

$$b = \begin{bmatrix} 1 & \frac{-1}{M-1} & \cdots & \frac{-1}{M-1} \end{bmatrix} \tag{A.1}$$

M is the number of antenna elements. The phase shift in Equation (A.1) is then multiplied by the beamsteering weights to form nullsteering weights, Equation (A.2).

$$W_{m_{null}} = W_m \cdot b \tag{A.2}$$

By modifying the weights with Equation (A.1), the signal is mitigated using destructive interference in the W_m direction. This allows for deterministic signal removal with a simple modification to the beamsteering algorithm. A nullsteering example is shown similar to Figure 6.2, in Figure A.1.

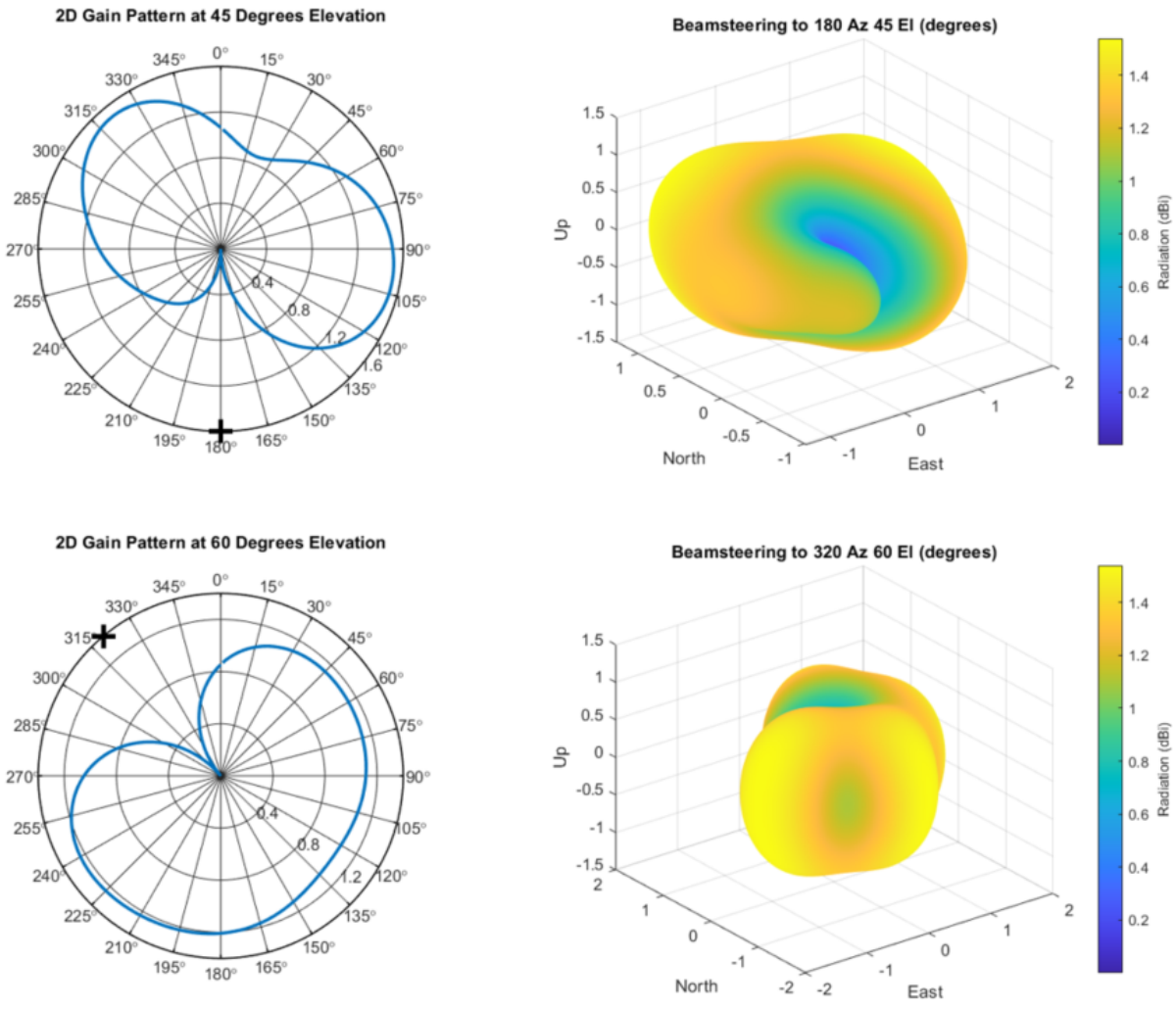

Figure A.1: 4 Element CRPA Nullsteering ($\frac{\lambda}{2}$ spacing)

More nulls can be created by adjusting the boresight weight vector to be an $M \times J$ array and calculating the nullspace, where J is the number of signals present.

Appendix B

Low SWaP-C Front End Jamming Mitigation [58]

GNSS is relied upon heavily in many areas for precise time and location, such as autonomous vehicle control, aircraft landing systems, and the stock market. Maintaining integrity in critical systems is imperative to preserve individuals' and organizations' safety and well-being. The need for more commercial focused interference mitigation methods has increased with the arrival of commercial off-the-shelf (COTS) personal privacy devices (PPDs). COTS PPDs were brought to the forefront of commercial interference research after the Newark Airport incident where an individual inadvertently jammed the automatic aircraft landing system at the airport [27]. While large, multi-element controlled reception pattern array (CRPA) based mitigation algorithms have been proven to be effective and robust at removing radio frequency interference, most individuals and organizations do not want to pay for and handle the large CRPA antennas when implementing anti-jamming hardware. Commercial applications benefit from lower SWaP-C mitigation methods to protect GNSS critical systems.

Though there is a vast amount of research in the area of low SWaP-C interference mitigation methods, few performance comparisons have been conducted. Although these algorithms are not capable of obtaining the same level of interference mitigation performance as the multidimensional array algorithms, they are still valuable for certain applications in the civilian sector.

The methods introduced are mitigation algorithms that use either fixed reception pattern arrays (FRPAs) or two element CRPAs. The single antenna approaches are a wavelet-implemented recursive least means squares adaptive notch filter [59], [60] and a finite impulse response (FIR) power minimization approach [55]. The two element array approaches are a

two-element space-time adaptive processing approach [61], [62] and a two-element adaptive noise canceling approach [63], [64]. The performance of the mitigation methods is outlined in detail in [58].

B.1 Mitigation Methods

The four methods compared in this chapter are a space-time adaptive processing (STAP) power minimization method, an adaptive noise canceling (ANC) method, a SVD-based FIR power minimization method, and a wavelet implemented adaptive notch filter (WANF). These methods are discussed in detail in the following subsections.

B.1.1 2 Element STAP Power Minimization

Spatial methods are widely used for interference mitigation. Beam steering implemented on CRPAs have been proven to be successful in removing radio frequency interference [10], [61], [62], [65]–[68]. [67] compares the space-time adaptive processing (STAP) with space-frequency adaptive processing (SFAP), which was first introduced by Gupta and Moore in [68]. Both methods are shown to achieve low SWaP-C mitigation with fewer antenna elements [67]. The spatial method implemented in this paper is a STAP power minimization algorithm. To ensure low SWaP-C, STAP was implemented with two antenna elements (N) and three taps (K). An outline of the algorithm is shown in Figure B.1.

The input signal states are shown below in Equation (B.1) with k being the current tap.

$$\mathbf{x}_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{1,k-1} \\ x_{2,k-1} \\ x_{1,k-2} \\ x_{2,k-2} \end{bmatrix} \tag{B.1}$$

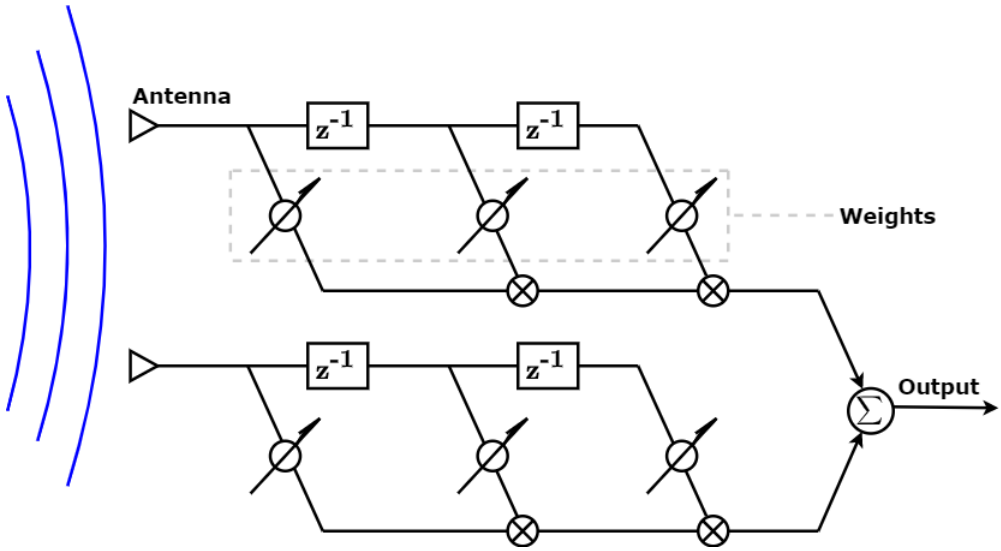

Figure B.1: 2 Element 3 Tap STAP Power Minimization

As with FIR power minimization, the signal state is then used to calculate the autocorrelation matrix, as shown in Equation (B.2). In this case, $M = KN$.

$$\mathbf{R}_{xx} = \frac{1}{M} \sum_{k=1}^{M} \mathbf{x}_k^H \mathbf{x}_k \tag{B.2}$$

Once the autocorrelation matrix is estimated, the optimal weights are estimated using Equation (B.3) below.

$$w_k = \frac{\mathbf{R}_{xx}^{-1} \boldsymbol{\delta}}{\boldsymbol{\delta}^T \mathbf{R}_{xx}^{-1} \boldsymbol{\delta}} \tag{B.3}$$

$\boldsymbol{\delta}$ is the constraint vector, given by $\boldsymbol{\delta} = [1, 0, 0, 0, 0, 0]^T$ for two elements and three taps. Once the optimal weights are calculated, the weights are applied to the signal state to determine the output as shown by Equation (B.4).

$$y_k = \mathbf{w}_k^T \mathbf{x}_k \tag{B.4}$$

B.1.2 2 Element ANC

The theory behind adaptive noise canceling (ANC) can be found in [66]. ANC has also been shown to successfully mitigate interference signals [63], [64]. However, very few implementations of ANC have been created for GNSS. ANC relies on a large noise difference between the interference signal and the carrier signal. At low J/S interference, ANC algorithms tend to become unstable if the algorithm is not adapted correctly. Figure B.2 shows the two element eight tap ANC architecture.

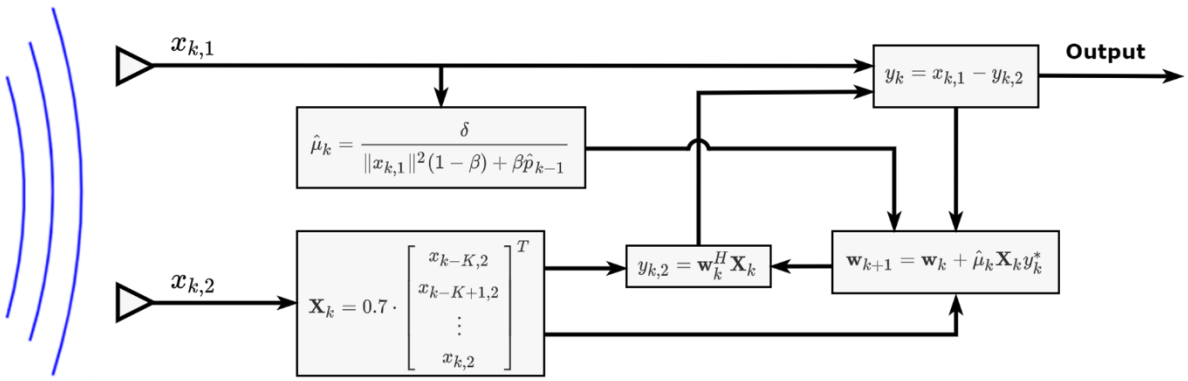

Figure B.2: 2 Element 8 Tap Adaptive Noise Canceling

ANC uses the error between an estimated output signal from the second antenna and the input signal from the first antenna to determine the output signal. In this sense, ANC is unique compared to the other algorithms. ANC uses time delay taps. In this implementation $K = 8$ time delay taps were used. With all algorithms, more time delay taps result in more measurements, which in turn gives better mitigation. However, given the nature of maintained low SWaP-C, eight taps were found to be an appropriate trade off. For two antenna elements, the incoming signal state is shown below in Equation (B.5) and in Figure B.2.

$$\mathbf{X}_k = 0.7 \cdot \begin{bmatrix} x_{k-K,2} & x_{k-K+1,2} & \cdots & x_{k,2} \end{bmatrix}^T \tag{B.5}$$

Notice that this signal state does not include the first signal element, and the signal state is down scaled to ensure the signals are not correlated. The weights are initialized at $\mathbf{w} = [1, 0, 0, \cdots, 0]^T$ and are recursively calculated in the filter. The 2^{nd} element output is estimated in Equation (B.6) below.

$$y_{k,2} = \mathbf{w}_k^H \mathbf{X}_k \tag{B.6}$$

Using the estimated output for signal 2 from Equation (B.6), the overall filter output is the error between the 1^{st} input signal state and the 2^{nd} estimated signal output, as shown by Equation (B.7). Note that $x_{k,1}$ is a scalar value, not a vector.

$$y_k = x_{k,1} - y_{k,2} \tag{B.7}$$

The recursive weight update is shown in Equation (B.8).

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{X}_k y_k^* \tag{B.8}$$

$()^*$ is the complex conjugate, and μ is the learning coefficient. In order to maintain filter stability for different interference signals at different power, the learning coefficient is adaptively updated by using a common power estimation algorithm shown in equation (B.9).

$$\hat{p}_k = \|x_k\|^2 (1 - \beta) + \beta \hat{p}_{k-1} \tag{B.9}$$

β is the forgetting factor, and $\|\|$ is the norm of the input signal. In this implementation, $\beta = 0.95$. μ_k is updated by the power estimate in Equation (B.10) below.

$$\mu_k = \frac{\delta}{\hat{p}_k} \tag{B.10}$$

δ is a filter tuning variable used to control the convergence speed of the filter. For the cases of this study, δ was set to be 0.00001 to ensure robust convergence.

B.1.3 1 Element FIR Power Minimization

The first single antenna algorithm tested was a singular value decomposition (SVD) eigenspace based FIR power minimization filter. In [55], the algorithm uses optimal methods to determine an autocorrelation matrix. Eigenvalue decomposition is performed on the autocorrelation matrix to obtain FIR filter weights. The weights are then applied to the input signal for filtering. At a high level, the FIR power minimization is a mix between a spatial adaptive processing filter and a standard FIR notch filter in its performance. Like the wavelet adaptive notch filter, the FIR power minimization filter damages the spectrum, which can cause degradation to the GPS signal. In this work, the algorithm was implemented with $K = 18$ time delay taps. The architecture for the 1 element 18 tap FIR power minimization filter is shown below in Figure B.3.

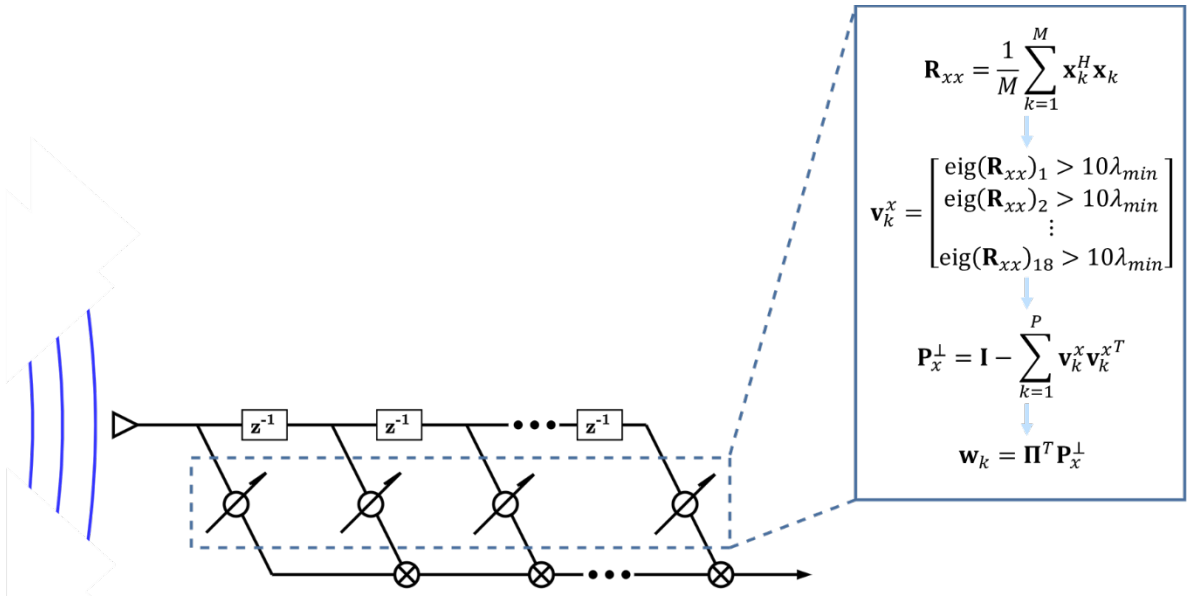

Figure B.3: 1 Element 18 Tap FIR Power Minimization

The incoming signal states are characterized by Equation (B.11), where $\mathbf{x}_k$ is a vector of $K-1$ tap delayed signal states.

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-K} \end{bmatrix} \tag{B.11}$$

The signal states are then used to estimate the autocorrelation matrix $\mathbf{R}_{xx}$, as shown by Equation (B.12) where M is the number of signal samples used to estimate the autocorrelation.

$$\mathbf{R}_{xx} = \frac{1}{M} \sum_{k=1}^{M} \mathbf{x}_k^H \mathbf{x}_k \tag{B.12}$$

After the autocorrelation matrix is estimated, the eigenvectors are calculated from the autocorrelation matrix and compared to a threshold determined by the eigenvalues, λ. The threshold is described in Equation (B.13).

$$\lambda_{thresh} = \lambda_{min} \cdot 10 \tag{B.13}$$

The eigenvectors with corresponding eigenvalues above the threshold are used to create the vector $\mathbf{v}_n^x$. This vector is then used to determine the projector matrix as shown by Equation (B.14).

$$\mathbf{P}_x^{\perp} = \mathbf{I} - \sum_{k=1}^{P} \mathbf{v}_k^x \mathbf{v}_k^{xT} \tag{B.14}$$

The projector matrix is used to determine the weights applied to the signal vector $\mathbf{x}_k$ in Equation B.15.

$$w_k = \mathbf{\Pi}^T \mathbf{P}_x^{\perp} \tag{B.15}$$

$\mathbf{\Pi}$ is defined by [55] to be the $Kx1$ "pinning" vector, or the constraint vector, defined by $\mathbf{\Pi} = [1, 0, \cdots, 0]^T$. The output signal is then defined by Equation (B.16).

$$y_k = \mathbf{w}_k^T \mathbf{x}_k \tag{B.16}$$

B.1.4 1 Element WANF

The use of wavelets in interference mitigation is not a new concept. The wavelet implemented adaptive notch filter aided with pulse blanking was tested by [59] to mitigate jamming signals being transmitted by COTS PPDs. The algorithm can be separated into three phases: wavelet decomposition, mitigation, and wavelet reconstruction, which are shown in Figure B.4. Wavelet decomposition separates the signal into signal packets to be processed separately. Mitigation attempts to remove the jammed signal from the decomposed signal by using adaptive notch filtering. Finally, wavelet reconstruction takes the signal packets and combines them back into a form of the original signal with jamming removed.

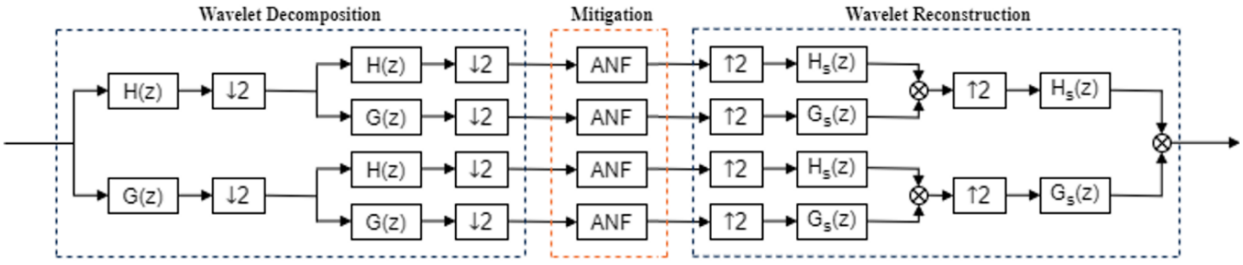

Figure B.4: Level 2 Wavelet Decomposition and Reconstruction Architecture

In the decomposition phase, $H(z)$ is a low-pass filter, and $G(z)$ is the opposite high-pass filter. After the signal is separated with the filters, the signal is down sampled by a factor of two. It is then passed to the next level of wavelet decomposition where it can be separated again. The mitigation phase has independent adaptive notch filters and frequency pulse blanking algorithms acting on each signal packet. The reconstruction phase includes up sampling by a factor of two and then applying equivalent synthesizing low-pass and high-pass filters. Once the signal is filtered, the signals are then combined to move up to the

next level of reconstruction. A more thorough description of wavelet deconstruction and reconstruction can be found in [69]. Figure B.4 shows a level 2 packet decomposition and reconstruction because each wavelet packet is down sampled twice. The algorithm tested in this paper is a level 3 implementation, which adds an additional stage to the example shown. A 24th order Daubechies filter was selected for the low-pass and high-pass filters. More information on Daubechies wavelet filters can be found in [70].

The wavelet filters are used to separate the GPS signal spectrum and are not used to directly mitigate the signal. The adaptive notch filters remove the jamming signals in this implementation. The adaptive notch filter used was first introduced by Wendel et al. in [60]. J. R. van der Merwe et al. improved the adaptive notch filter by adapting the learning coefficient with an average signal power estimator [59]. This paper uses a wavelet-implemented adaptive notch filter (WANF) with the adaptive learning coefficient. The filter itself is a first order IIR filter that estimates the amplitude and frequency of the interference signal. Equation (B.17) shows the filter's transfer function [59], [69].

$$H(z) = \frac{1 - z_0 z^{-1}}{1 - k_\alpha z_0 z^{-1}} \tag{B.17}$$

k_α is the pole contraction factor that determines the width of the notch, and z_0 is the complex estimated state. The equation for z_0 is shown below in Equation (B.18).

$$z_0 = a \cdot e^{2\pi j f_0 T} \tag{B.18}$$

a is the amplitude of the jammed signal, f_0 is the frequency of the jammed signal, and T is the sample period. Figure B.5 shows the discrete block diagram for the adaptive notch filter. Given the signal x_k, the discrete version of the recursive filter can be expressed as difference equations with the IIR portion in Equation (B.19) and the FIR portion in Equation (B.20).

$$r_k = x_k - k_{alpha} \hat{z}_{0,k-1} r_{k-1} \tag{B.19}$$

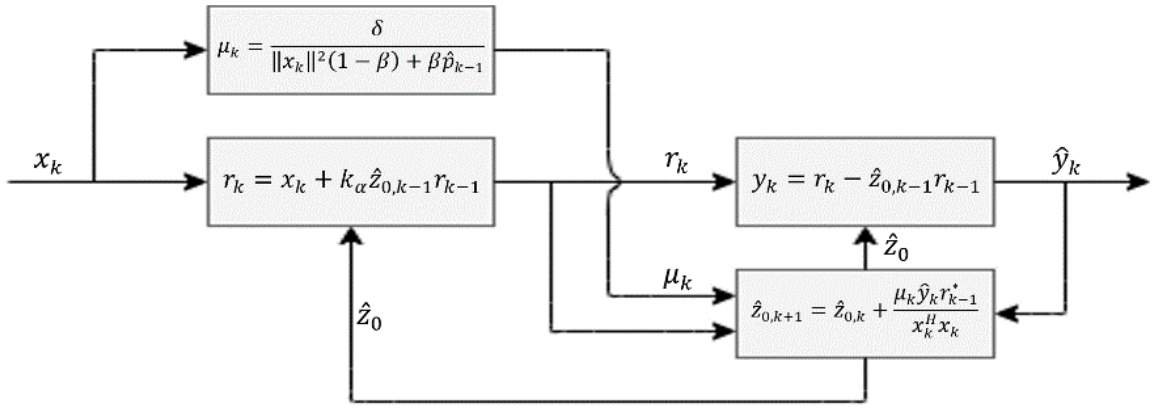

Figure B.5: Adaptive Notch Filter Block Diagram

$$\hat{y}_k = r_k - \hat{z}_{0,k-1} r_{k-1} \tag{B.20}$$

$\hat{y}_k$ is the output of the WANF. $\hat{z}_0$ is initialized by Equation (B.18) and updated using the method shown in Equation (B.21).

$$\hat{z}_{0,k+1} = \hat{z}_{0,k} + \frac{\mu_k \hat{y}_k r^*_{k-1}}{x_k^H x_k} \tag{B.21}$$

x_k^H is the Hermitian transpose of the signal state, r_{k-1^*} is the complex conjugate of r_{k-1}, and μ_k is the learning coefficient. The learning coefficient is adapted with the same power estimation algorithm used for ANC in Equations (B.9) and (B.10). However, $\delta = 50$ for the WANF because the filter requires a much larger μ for robust convergence.

From an observation standpoint, by implementing the adaptive notch filter on the wavelet packets, narrower notches are effectively achieved on the signal spectrum after the packets are reconstructed into the full bandwidth signal. The computational load with this algorithm is attributed to the fact that an adaptive notch filter must be implemented on each wavelet packet. In the simulations, 8 adaptive notch filters were running simultaneously over each wavelet packet. For optimal performance, the adaptive notch filters should be implemented in parallel.

Appendix C

Extended Kalman Filter (EKF) PVT Estimation

In Chapter 3, the outline for the standard least squares positioning algorithm was presented. The EKF positioning algorithm, which was shown in Chapter 5 coupled in vector tracking, is presented in this chapter in a standalone form. When comparing the EKF in Chapter 5 and the EKF in this chapter, very few difference can be found. The differences come from the asynchronous implementation of the vector tracking loop and the batch implementation of the positioning algorithm. The state vector is the same as the vector tracking loop implementation as shown by Equation (C.1).

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} & \hat{\dot{x}} & \hat{y} & \hat{\dot{y}} & \hat{z} & \hat{\dot{z}} & c\hat{b} & c\hat{\dot{b}} \end{bmatrix}^T \tag{C.1}$$

The state transition matrix for the standalone EKF is shown below in Equation (C.2).

$$A = \begin{bmatrix} 1 & T_{nav} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_{nav} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_{nav} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T_{nav} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.2}$$

The measurement matrix H is shown below in Equation (C.3).

$$H = \begin{bmatrix} -a_x^{(1)} & 0 & -a_y^{(1)} & 0 & -a_z^{(1)} & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_x^{(j)} & 0 & -a_y^{(j)} & 0 & -a_z^{(j)} & 0 & 1 & 0 \\ 0 & -a_x^{(1)} & 0 & -a_y^{(1)} & 0 & -a_z^{(1)} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -a_x^{(j)} & 0 & -a_y^{(j)} & 0 & -a_z^{(j)} & 0 & 1 \end{bmatrix} \tag{C.3}$$

T_{nav} is the navigation integration period. In the vector tracking loop, T_{nav} was δT, the step length of the asynchronous loop. The process noise covariance and measurement noise covariance matrices are shown in Equations (C.4) and (C.5). The process noise covariance matrix is the same in the positioning algorithm as it is in the vector tracking loop implementation. The measurement noise covariance matrix updates all of the measurements at once due to the batch implementation in the positioning algorithm.

$$Q = \begin{bmatrix} Q_{PV} & 0 & 0 & 0 \\ 0 & Q_{PV} & 0 & 0 \\ 0 & 0 & Q_{PV} & 0 \\ 0 & 0 & 0 & Q_c \end{bmatrix} \tag{C.4a}$$

$$Q_{PV} = \begin{bmatrix} \sigma_{PV}^2 \frac{T_{nav}^3}{3} & \sigma_{PV}^2 \frac{T_{nav}^2}{2} \\ \sigma_{PV}^2 \frac{T_{nav}^2}{2} & \sigma_{PV}^2 T_{nav} \end{bmatrix} \tag{C.4b}$$

$$Q_c = \begin{bmatrix} \sigma_b^2 T_{nav} + \sigma_d^2 \frac{T_{nav}^3}{3} & \sigma_d^2 \frac{T_{nav}^2}{2} \\ \sigma_d^2 \frac{T_{nav}^2}{2} & \sigma_d^2 T_{nav} \end{bmatrix} \tag{C.4c}$$

$$\sigma_b^2 = c^2 \frac{h_0}{2} \tag{C.4d}$$

$$\sigma_d^2 = c^2 2\pi^2 h_{-2} \tag{C.4e}$$

$$R = \begin{bmatrix} \sigma^{2^{(1)}_{\delta\rho}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^{2^{(j)}_{\delta\rho}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^{2^{(1)}_{\delta\dot{\rho}}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^{2^{(j)}_{\delta\dot{\rho}}} \end{bmatrix} \tag{C.5}$$

The measurement vector $\mathbf{y}$ is shown in Equation (C.6). This implementation is shown for j total satellites. The batch implementation again means that all of the measurements are applied at one time.

$$\mathbf{y} = \begin{bmatrix} \delta\rho^{(1)} & \dots & \delta\rho^{(j)} & \delta\dot{\rho}^{(1)} & \dots & \delta\dot{\rho}^{(j)} \end{bmatrix}^T \tag{C.6}$$

For the EKF to converge, just like with vector tracking, the EKF must be initialized from a least squares position. After the EKF is initialize, the Kalman filter follows the same format with Equations (C.7) showing the time update and Equations (C.8) showing the measurement update.

$$\hat{x}_k^- = A\hat{x}_{k-1}^+ \tag{C.7a}$$

$$P_k^- = AP_{k-1}^+A^T + Q \tag{C.7b}$$

$$L_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{C.8a}$$

$$P_k^+ = (I - L_k H)P_k^- \tag{C.8b}$$

$$\hat{x_k}^+ = \hat{x_k}^- + L_k \cdot \mathbf{y} \tag{C.8c}$$

A static data set was simulated using a Spirent GSS9000 Signal Generator for comparison between the least squares positioning algorithm and the EKF positioning algorithm.

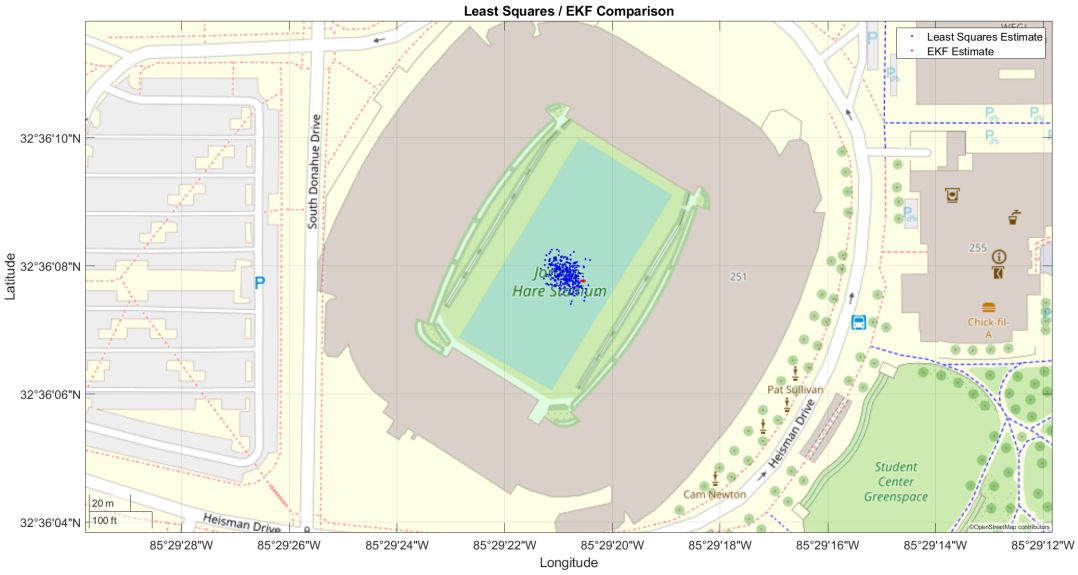

Figure C.1: Static Simulated Least Squares EKF Comparison

Appendix D

Range and Range Rate Vector Tracking Improvements

D.1 Signals at Same Power

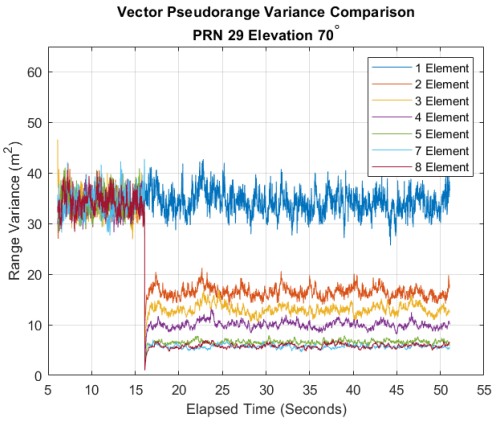

Figure D.1: Same Power Vector Tracking Range Variance with EL 70°

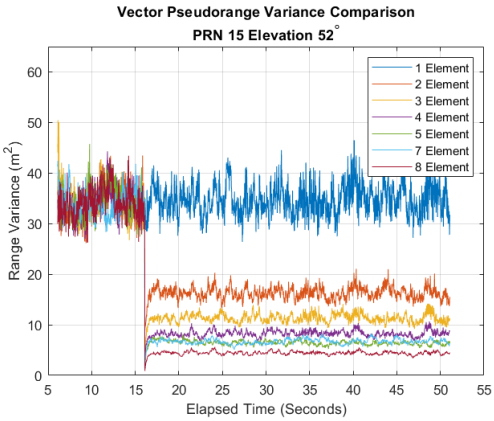

Figure D.2: Same Power Vector Tracking Range Variance with EL 52°

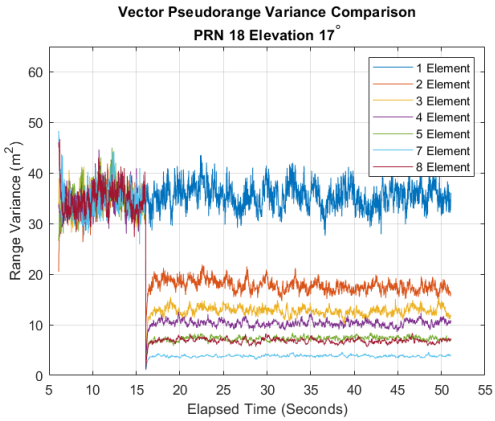

Figure D.3: Same Power Vector Tracking Range Variance with EL 17°

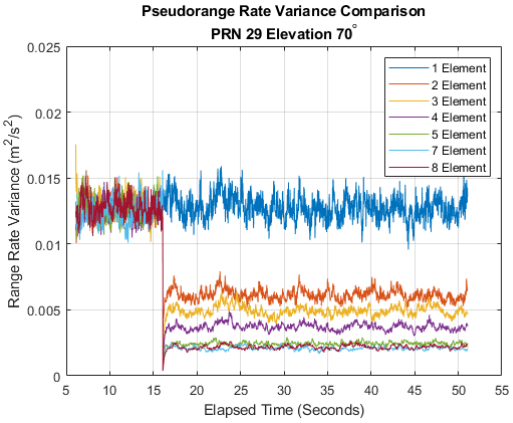

Figure D.4: Same Power Vector Tracking Range Rate Variance with EL 70°

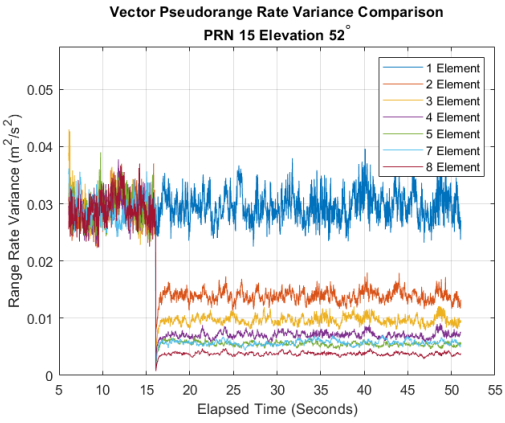

Figure D.5: Same Power Vector Tracking Range Rate Variance with EL 52°

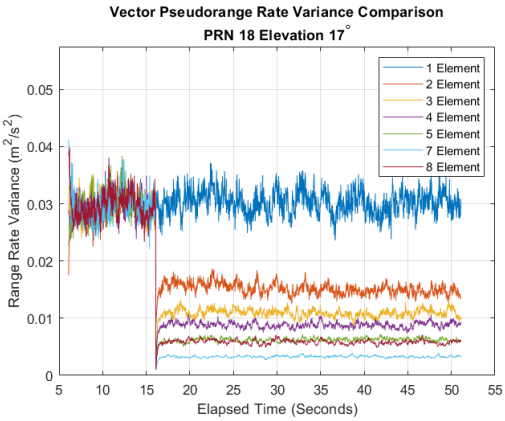

Figure D.6: Same Power Vector Tracking Range Rate Variance with EL 17°

D.2 Signals at Variable Power

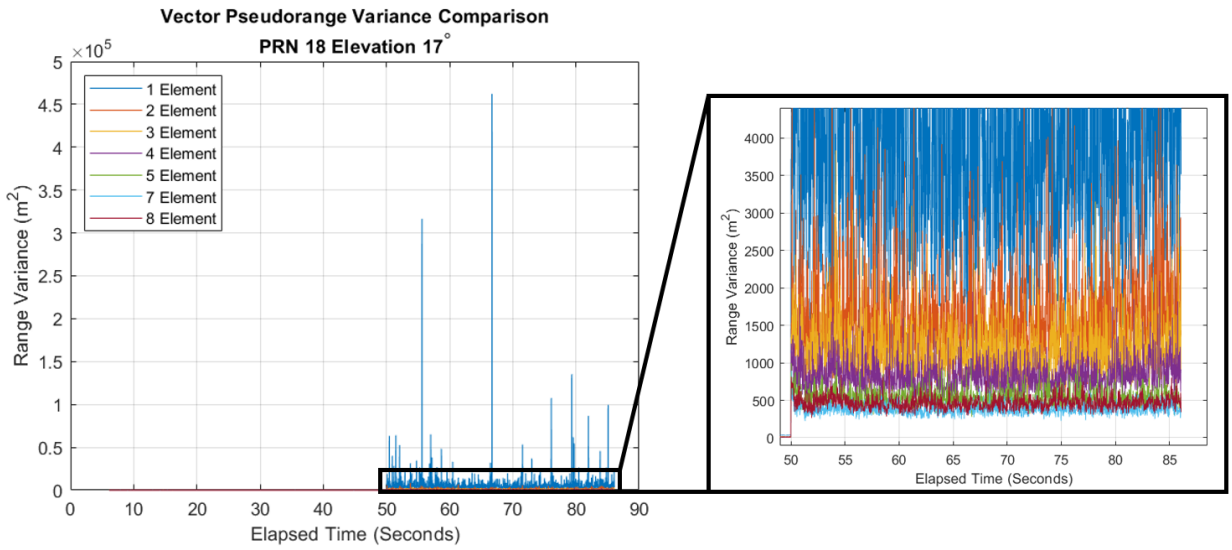

Figure D.7: Variable Power Vector Tracking Range Variance with EL 17°

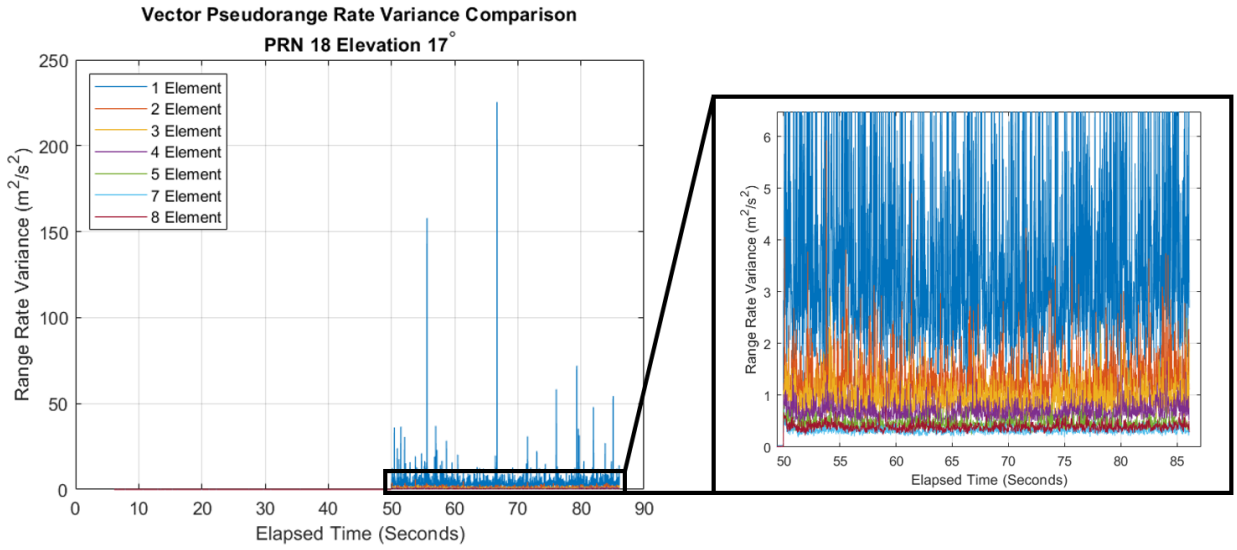

Figure D.8: Variable Power Vector Tracking Range Rate Variance with EL 17°

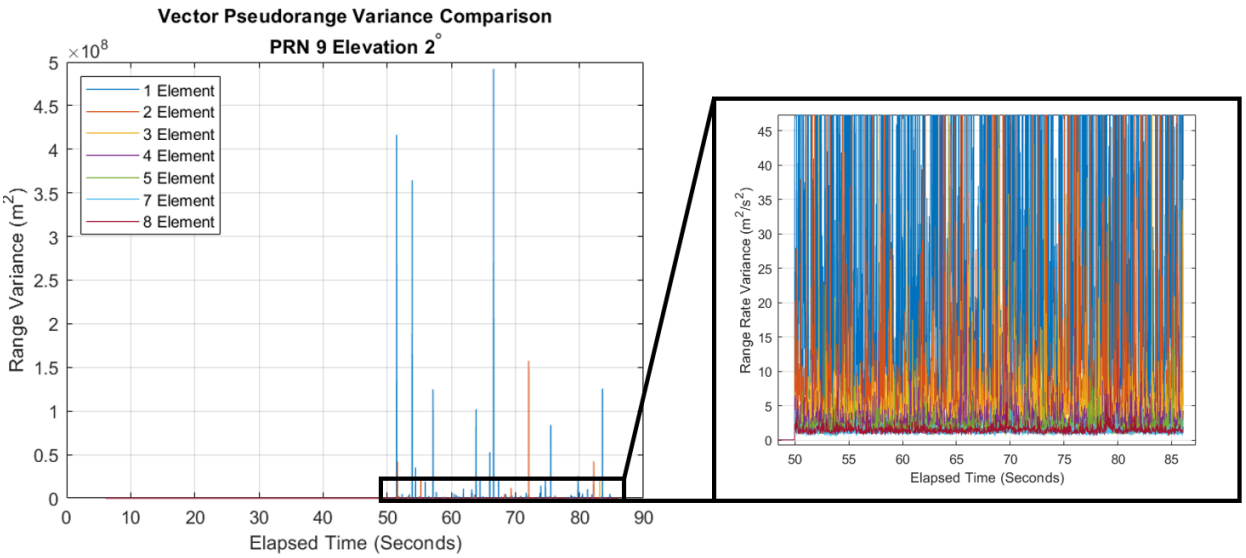

Figure D.9: Variable Power Vector Tracking Range Variance with EL 2°

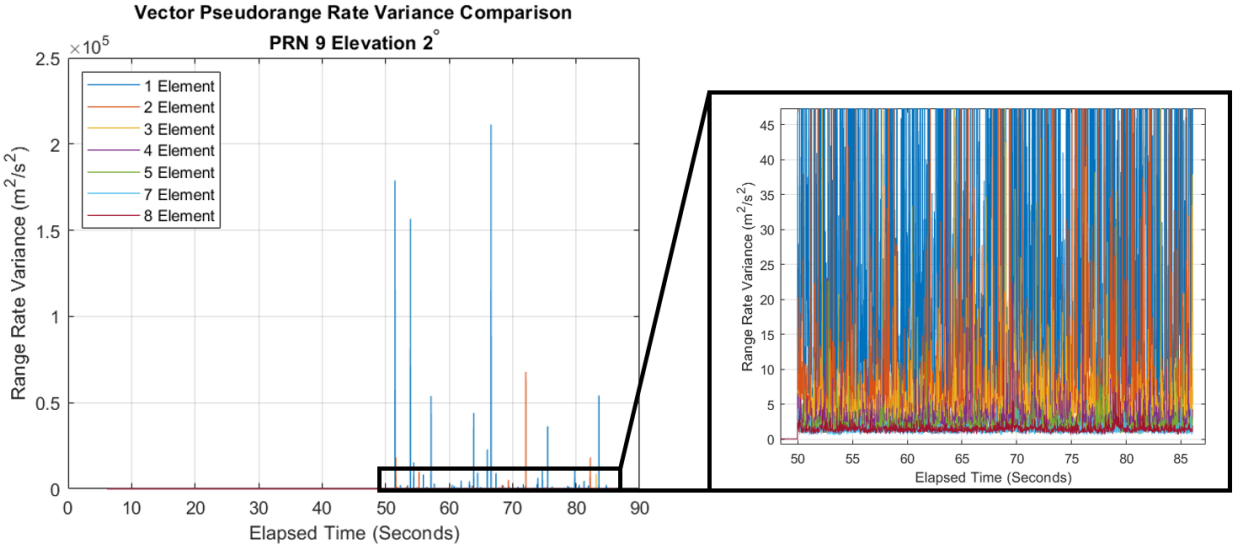

Figure D.10: Variable Power Vector Tracking Range Rate Variance with EL 2°

D.3 Live Sky Commuter Route

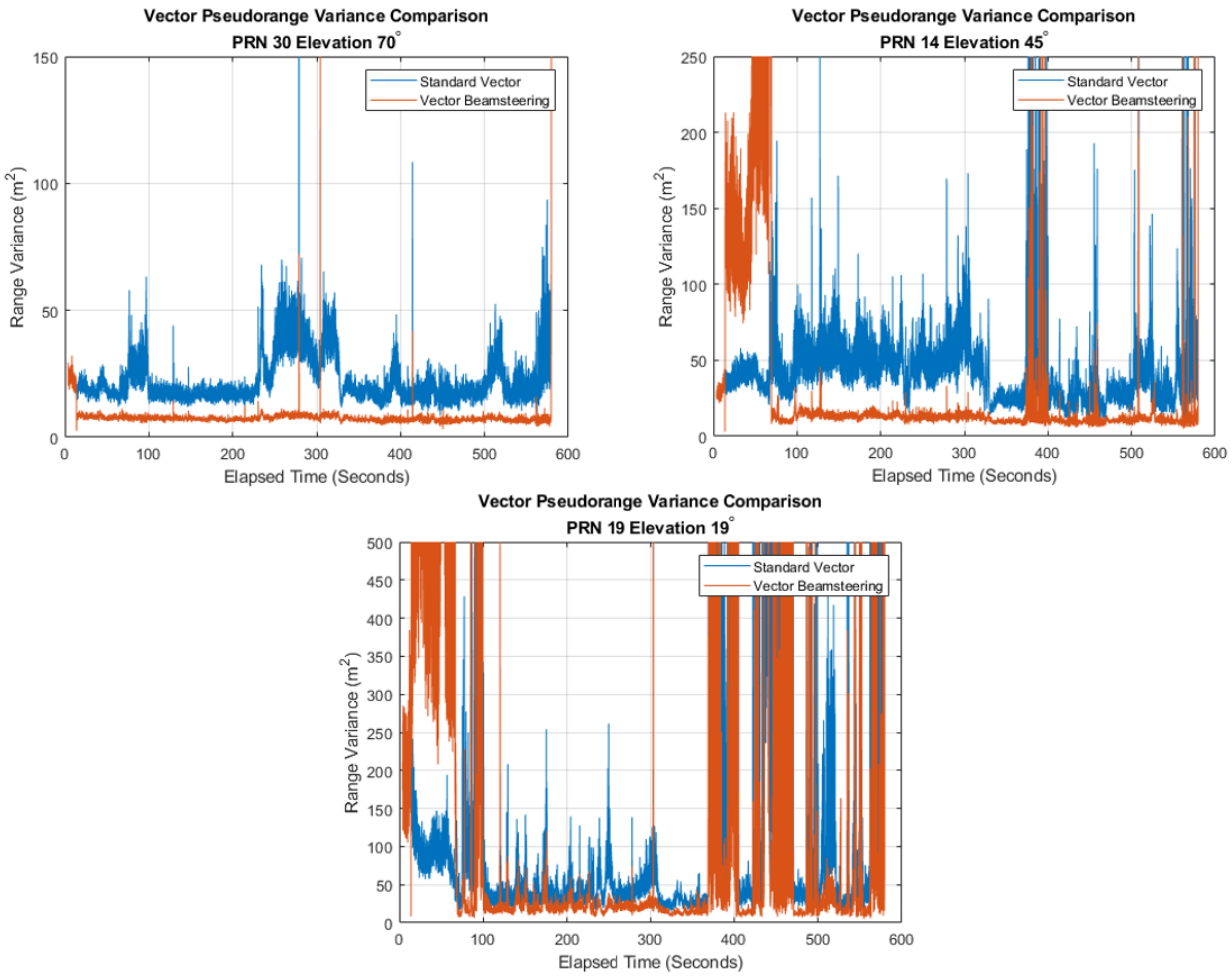

Figure D.11: Range Variances from Commuter Route

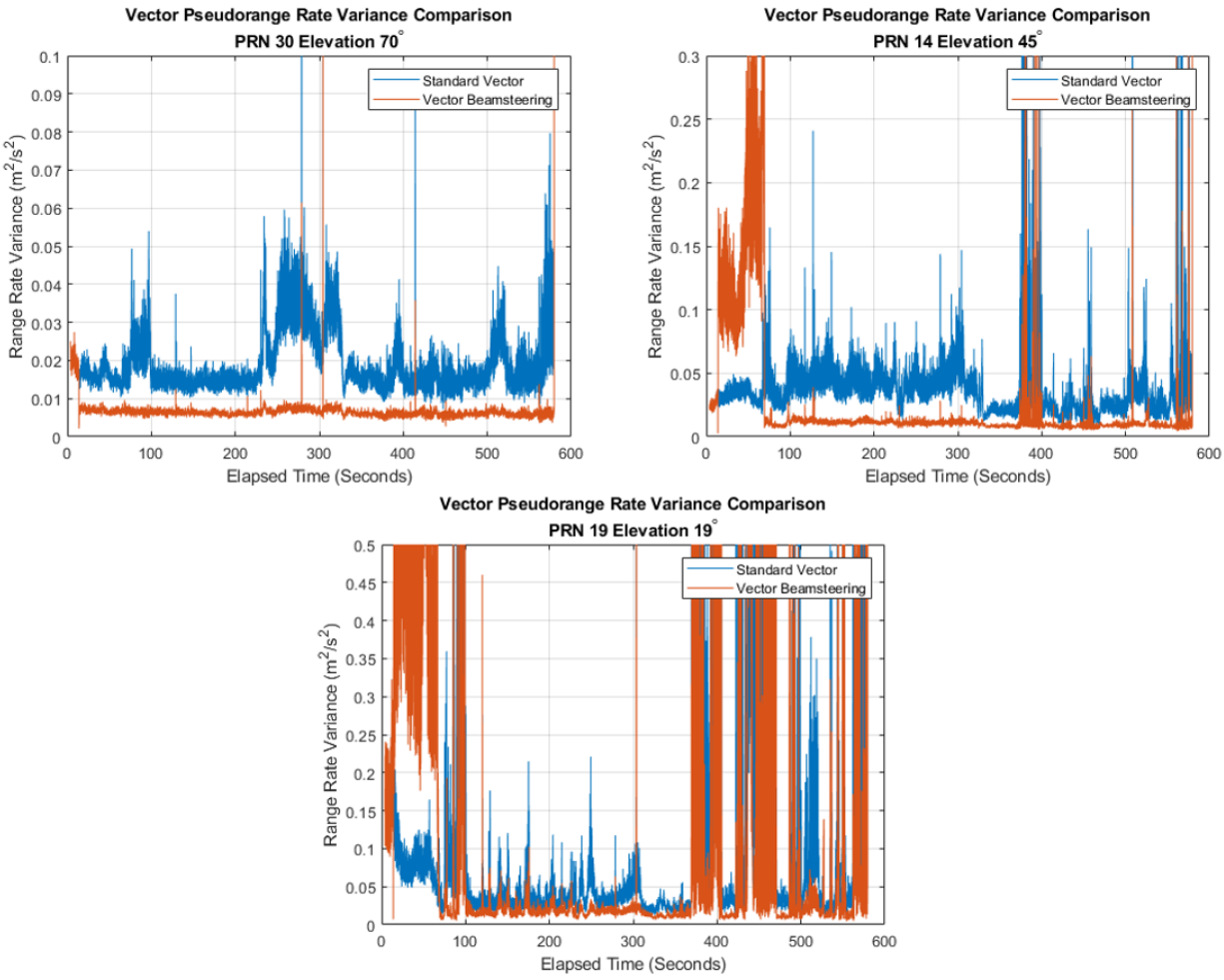

Figure D.12: Range Rate Variances from Commuter Route

D.4 Live Sky Jordan Hare Route

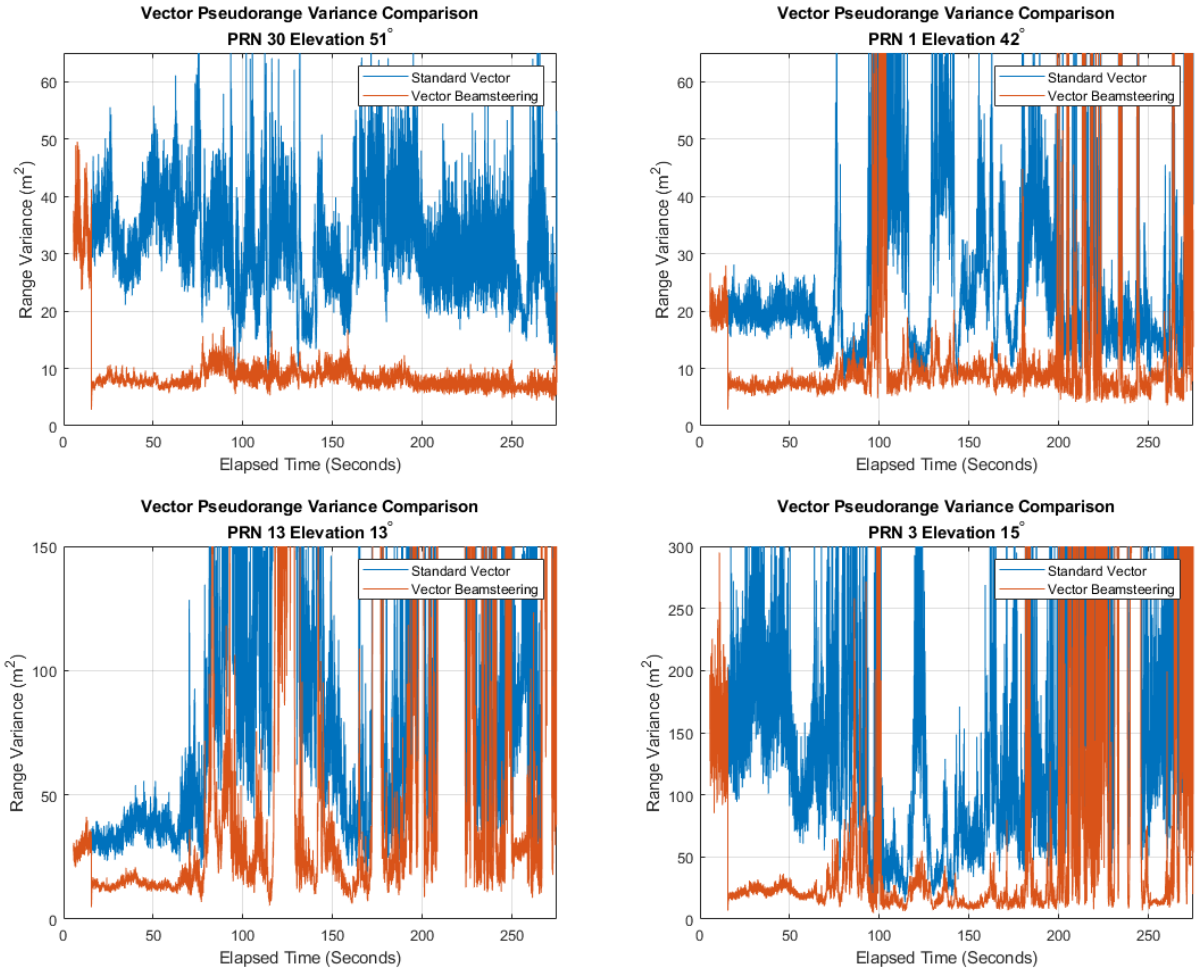

Figure D.13: Range Variances from Jordan Hare Route

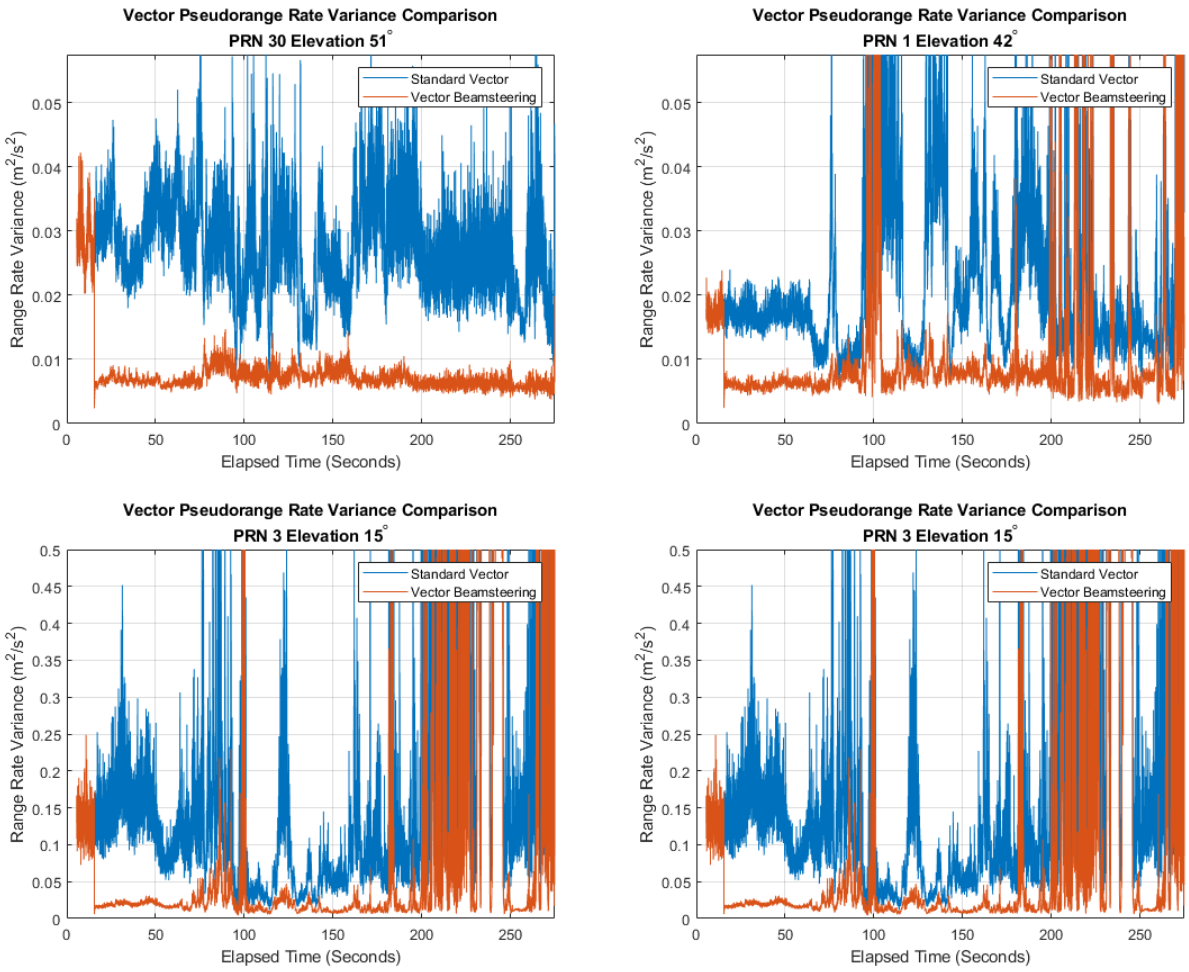

Figure D.14: Range Rate Variances from Jordan Hare Route

D.5 Commuter Route with Injected Interference

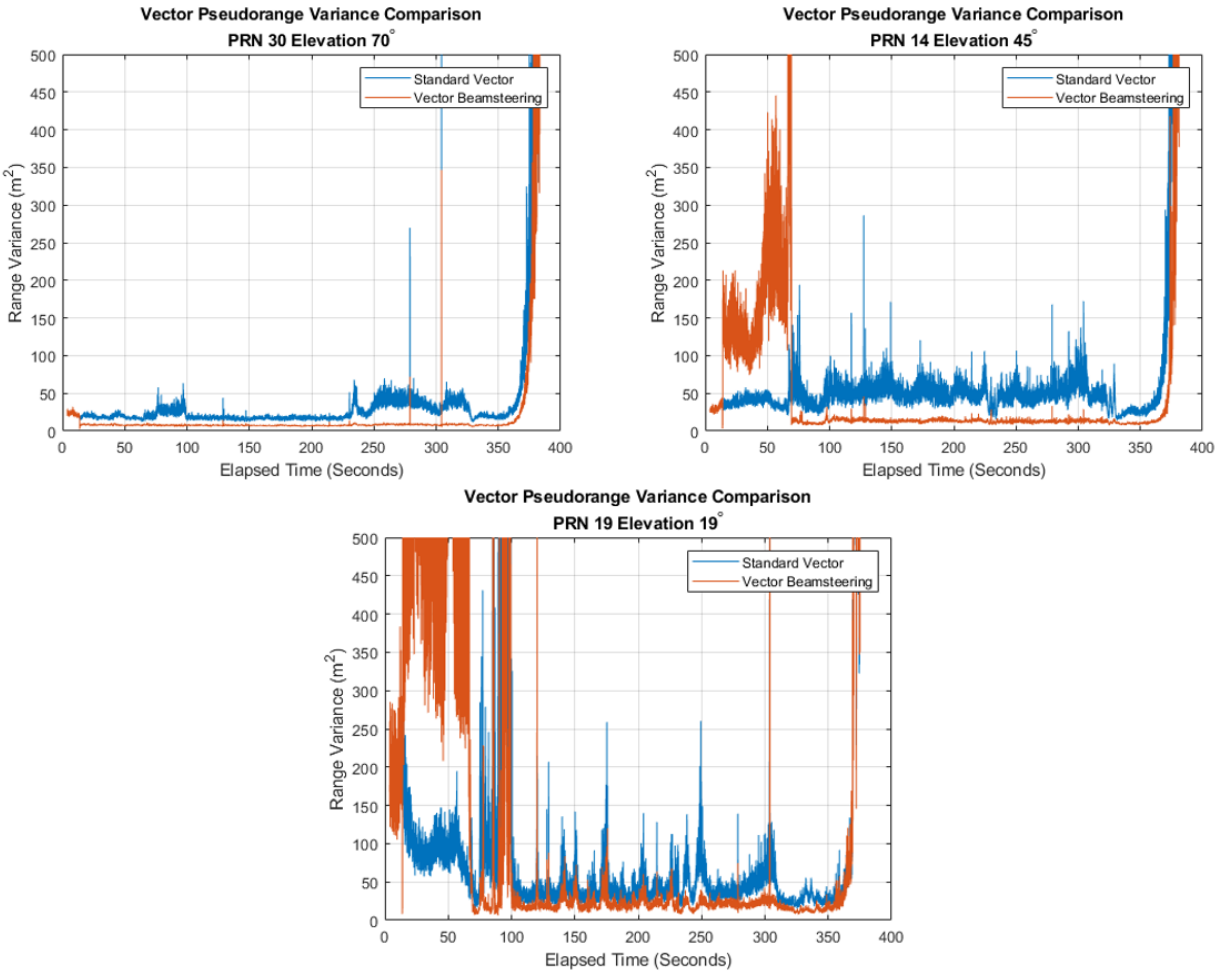

Figure D.15: Range Variances from Jamming Injection

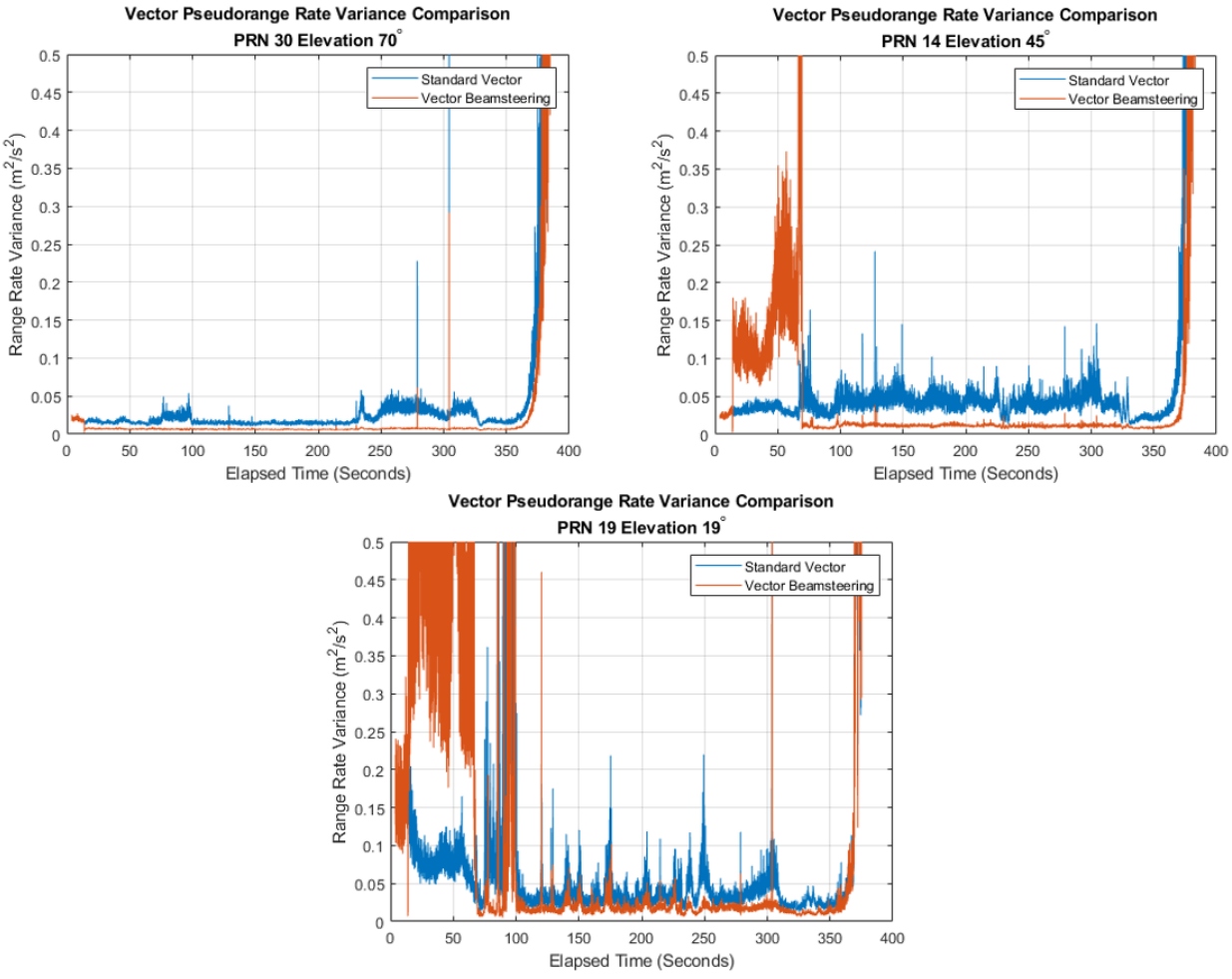

Figure D.16: Range Rate Variances from Jamming Injection