

Fault Detection and Exclusion in Deeply Integrated GPS/INS Navigation

by

Benjamin J. Clark

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 8, 2012

Keywords: GPS, vector tracking, FDE, DI

Copyright 2012 by Benjamin J. Clark

Approved by

David M. Bevly, Chair, Professor of Mechanical Engineering
George T. Flowers, Professor of Mechanical Engineering
Andrew J. Sinclair, Associate Professor of Aerospace Engineering
Dan B. Marghitu, Professor of Mechanical Engineering

Abstract

This dissertation presents a novel fault detection and exclusion method in a centralized deeply integrated GPS/INS navigation system. The method presented is also demonstrated in a centralized vector tracking GPS receiver. Also, a new multipath error model and a range variance parameter are developed to better deal with the additional challenges faced by vector-based receivers. These methods and analysis extend the field of robust navigation, particularly with regards to advanced tracking architectures.

GPS was originally designed to operate in clear line of sight view to the supporting satellite constellation. However, recent advances in receiver hardware and computing power have pushed positioning into more difficult scenarios. New ways of using radionavigation sensors, such as vector tracking and deeply integrated GPS/INS, have been developed to handle these situations. However, integrity has been difficult to maintain in these configurations. Fault detection and exclusion can provide this need by keeping the navigation solution free from erroneous measurements that occur in difficult environments.

This dissertation presents these contributions in four stages. In the first, a multipath model for vector tracking is developed. This model is based on a delayed signal's interaction with the direct signal correlation peak rather than the scalar early and late correlator outputs. To demonstrate both the model and improved performance of vector tracking, simulated results show the vector receiver tracking the signal with 0.015 m less range error. Experimental results show vector tracking performing better in multipath environments by several meters.

Second, the range variance parameter is derived as a means to monitor a vector receiver's tracking situation. Since traditional lock detection does not apply directly to a vector receiver, another approach is needed. The range variance gives an indication of how the receiver's position uncertainty would translate to range uncertainty. This is done in a geometry-free way so the receiver can determine the maximum impact its error would have. The variance parameter is demonstrated in an environment with significant blockage to show its response to the tracking situation.

Third, fault detection and exclusion are applied to a centralized vector tracking architecture. This integrity method is based on the normalized innovation test parameter. When new measurements are provided to the navigation filter, they are normalized by their expected variances. Faulty signals are shown to increase this test parameter and pass the detection threshold. Live sky demonstrations of fault detection and exclusion yield position improvements on the order of one meter.

Lastly, a deeply integrated GPS/INS algorithm is presented. The vector fault detection and exclusion method also applies to this fused navigation system. After dealing with IMU synchronization, results are presented in which an automotive grade IMU is integrated with a vector software receiver. Using the fault detection and exclusion method, positioning performance is improved by several meters.

In total, this dissertation demonstrates two navigation methods: the GPS vector receiver and the Deeply Integrated GPS/INS system. Both of these methods are made more robust by performing fault detection and exclusion. This method is shown to remove velocity drifts and position jumps due to signal errors in difficult scenarios. The result is a highly robust navigation system for continuous positioning in GPS degraded environments.

Acknowledgments

First and foremost, I am thankful to my God for the blessing of this and many other opportunities. I am thankful for my wife, Follin, who has supported me through school here at Auburn University and helped me complete this work. I want to acknowledge my family in helping me to make learning a habit and my professors in equipping me with tools to follow that pursuit. Several members of the GAVLAB were instrumental in this work, particularly Matthew Lashley and Scott Martin. I thank Dr. David Bevly for his advising and growing the GAVLAB as an environment to answer many interesting research questions.

Table of Contents

Abstract	ii
Acknowledgments	iv
List of Figures	ix
List of Tables	xvii
1 Introduction	1
1.1 Contributions	2
1.2 Dissertation Contents	3
1.3 Nomenclature	4
1.4 Summary	4
2 Operation of a GPS Software Receiver	5
2.1 Positioning	5
2.1.1 Multilateration	5
2.1.2 Satellite Positioning	9
2.1.3 Clock Bias	12
2.2 GPS Observables	14
2.2.1 Pseudorange	14
2.2.2 Doppler Frequency	17
2.2.3 Carrier Phase	18
2.3 Channel Tracking	20
2.3.1 GPS Signal	20
2.3.2 Correlation	22
2.3.3 Discriminators	23
2.3.4 Loop Filter	27
2.3.5 Numerically Controlled Oscillator	28
2.4 Acquisition	28

2.4.1	Serial Code Phase Acquisition	29
2.4.2	Parallel Code Phase Acquisition	30
2.4.3	Weak Signal Acquisition	32
2.5	Hardware	32
2.5.1	Front-End	33
2.5.2	Real and Complex Signals	34
2.6	Summary	35
3	GPS Vector Tracking Receiver	36
3.1	Difficult Environments	36
3.2	Vector Tracking	38
3.2.1	GPS Vector Tracking Formulation	41
3.2.2	Equivalent Scalar Tracking Formulation	45
3.2.3	Measurement Generation	49
3.3	Signal Strength	51
3.3.1	Amplitude Estimation	51
3.3.2	Carrier to Noise Ratio	52
3.4	Summary	54
4	GPS Receiver Operation in Multipath	56
4.1	Theoretical Performance	57
4.1.1	Multipath Modification of Receiver Measurements	57
4.1.2	Theoretical Scalar Performance	64
4.1.3	Theoretical Vector Performance	64
4.2	Multipath Simulation	66
4.2.1	Simulation Validation	67
4.2.2	Correlator Models	68
4.2.3	Multipath Model	69
4.2.4	Simulated Performance	70
4.3	Experimental Results	73
4.3.1	Static Performance	73
4.3.2	Dynamic Performance	76

4.4	Summary	76
5	Receiver Lock Detection	78
5.1	Channel Lock Detection	78
5.1.1	Lock Detection Based on Carrier to Noise Ratio	79
5.2	Scalar Lock	80
5.3	Vector Lock	81
5.3.1	Vector Divergence	82
5.3.2	Vector Divergence Experiment	84
5.4	Maximum Range Variance	86
5.4.1	Range Variance Verification	91
5.5	Range Variance Results	92
5.6	Summary	97
6	Vector Fault Detection and Exclusion	98
6.1	Normalized Innovation	98
6.2	Theoretical Threshold Selection	100
6.3	Probability Validation	103
6.4	Vector FDE Results	106
6.4.1	Open Sky in Auburn, AL	106
6.4.2	Moderately Blocked Sky in Auburn, AL	108
6.4.3	Urban Canyon in Atlanta, GA	111
6.5	Summary	116
7	Deeply Integrated GPS/INS	118
7.1	Published Architectures	119
7.2	INS Mechanization Equations	121
7.3	Deeply Integrated Filter	122
7.3.1	State Dynamics	124
7.3.2	State Measurement Relations	126
7.4	Time Synchronization	127
7.5	Fault Detection and Exclusion	127
7.6	DI Results	128

7.6.1	Vector Tracking or Deeply Integrated	130
7.7	Summary	132
8	Conclusion	134
8.1	Future Work	135
	Bibliography	137
	Nomenclature	141
A	Correlator	144
A.1	Fast Correlation	145
A.2	Fine Correlation	146
A.3	Efficient Correlation	147
A.4	Correlation Results	147
B	Simulator	151
B.1	RF Level Simulator	151
B.2	IF Level Simulator	151
B.3	Correlation Level Simulator	152
C	Vector Tracking in a Scalar Formulation	155
C.1	Condition Number	155
C.2	Experimental Results	156
D	Multipath Results	158

List of Figures

2.1	Scalar receiver diagram.	6
2.2	Circle of possible positions at range r_1 from position s_1	7
2.3	Position constrained by multiple ranges to known locations.	7
2.4	Position ambiguity solved with additional range.	8
2.5	Satellite and user position at transmit time, shown with inertial reference ECEF frame.	11
2.6	Satellite and user position at exaggerated reception time with new ECEF frame and original ECEF frame and satellite position.	12
2.7	Scalar receiver diagram.	15
2.8	Pseudorange ambiguity showing receiver clock offset.	16
2.9	Pseudorange formulation from received times.	16
2.10	Illustration of Doppler shift due to relative motion.	17
2.11	Range measurements from carrier phase and code phase.	19
2.12	Tracking channel diagram.	20
2.13	Correlation peak and code tracking discriminators.	24
2.14	Shifted correlation peak and code tracking discriminators that show tracking error.	25
2.15	Linear and pull-in regions for the code discriminator.	26
2.16	Linear and pull-in regions for the carrier discriminator.	26

2.17	Tracking channel diagram.	27
2.18	Generic phase locked loop for code or carrier scalar tracking loops.	28
2.19	Serial acquisition diagram.	29
2.20	Parallel acquisition diagram.	31
2.21	GPS front-end diagram.	33
3.1	Multipath types typical in an urban canyon environment.	37
3.2	Diagram of a generic scalar tracking receiver.	39
3.3	Diagram of a generic vector tracking receiver.	40
3.4	Estimated frequency for scalar and vector receiver during signal loss scenario.	41
3.5	Poor numerical performance of the pseudorange state receiver (left) shown compared to position state vector receiver (right).	48
4.1	Multipath and direct signal correlation peaks (top) and resulting correlation peak with constructive (left) and destructive (right) interference	60
4.2	Multipath range error bounds for possible delays	61
4.3	Induced range rate measurement error for given multipath frequency and phase errors	62
4.4	Steady state error bounds for maximum and minimum range error in the presence of multipath.	63
4.5	Effect of injected bias on single channel tracking error	66
4.6	Simulator Diagram	67
4.7	Simulated multipath induced bias versus relative delay, shown with predicted error bounds.	68

4.8	Diagram showing multipath reflection received by antenna	70
4.9	Scalar and vector tracking error versus multipath delay	71
4.10	Scalar and vector position over simulated multipath range	72
4.11	Scalar and vector receiver position solution in static scenario with moderate proximity to reflective wall	74
4.12	Scalar and vector receiver position errors in static scenario with moderate proximity to reflective wall	74
4.13	Scalar and vector receiver position solution in static scenario with close proximity to reflective wall	75
4.14	Scalar and vector receiver position errors in static scenario with close proximity to reflective wall	75
4.15	Scalar and vector receiver position errors in dynamic scenario.	77
5.1	Error injection vector along line of sight to one satellite.	83
5.2	Divergence experimental results for injected line of sight position and velocity errors on satellite 14.	85
5.3	Divergence simulated results for injected line of sight position and velocity errors on satellite 14.	85
5.4	Found maximum standard deviation direction and value with surface of standard devi- ations for a surface of signal directions.	90
5.5	Range vector bounds on simulated range error for a static scenario.	91
5.6	Range variance parameter along downtown Atlanta, GA route.	93
5.7	Range variance parameter beneath two overpasses in Atlanta, GA.	94

5.8	Range variance parameter in urban canyon in Atlanta, GA.	95
5.9	Amplitude estimate decay for satellite 31 when traveling beneath overpass.	96
5.10	C/N_0 estimate decay for satellite 31 when traveling beneath overpass.	96
5.11	Measurement variance increase for satellite 31 when traveling beneath overpass.	97
6.1	Normalized Innovation Density Function for Valid and Invalid Measurements	99
6.2	Range errors in simulation with 10 m bias and no FDE	101
6.3	Range errors in simulation with 10 m bias and FDE	101
6.4	Position errors in 10 m bias simulation with and without FDE	102
6.5	Theoretical probability of missed detection over a range of pseudorange biases given a threshold of 3.	103
6.6	Simulated probability of false alarm versus velocity variance for a static receiver.	105
6.7	Simulated probability of missed detection versus pseudorange bias for a static receiver.	105
6.8	Vector receiver with and without FDE along College Street in Auburn, AL.	107
6.9	Portion of vector receiver with and without FDE along College Street.	108
6.10	Velocity magnitude profile at static portion along College Street.	109
6.11	Range rate normalized innovations for satellite 28 at static portion along College Street.	109
6.12	Vector receiver with and without FDE along College Street and Magnolia Avenue in Auburn, AL.	110
6.13	Portion of vector receiver with and without FDE along College Street and Magnolia Avenue in Auburn, AL.	111

6.14	Range normalized innovations for satellite 5 at error along Magnolia Avenue.	112
6.15	Vector receiver with and without FDE in Downtown Atlanta, GA.	113
6.16	Portion of vector receiver with and without FDE beneath I-85 overpass.	114
6.17	Range rate normalized innovations for satellite 31 beneath I-85 overpass.	115
6.18	Range normalized innovations for satellite 16 beneath I-85 overpass.	115
6.19	Range rate normalized innovations for satellite 22 beneath I-85 overpass.	116
6.20	Portion of vector receiver with and without FDE beneath I-85 overpass with estimate of lane.	117
7.1	Diagram for loosely coupled (LC), closely coupled (CC), and tightly coupled (TC) GPS/INS architectures.	120
7.2	Deeply Integrated GPS/INS diagram.	123
7.3	DI and DI with FDE positioning solution along semi-urban environment.	129
7.4	DI and DI with FDE positioning solution along Magnolia Avenue.	130
7.5	Range normalized innovations for satellite 5 at DI error along Magnolia Avenue.	131
7.6	Maximum range variance for vector receiver and deeply integrated GPS/INS.	131
7.7	Dead reckoning for vector receiver and deeply integrated GPS/INS in a complete signal outage.	132
A.1	Sawtooth pseudorange error with correlation errors	145
A.2	Alignment of replica code and received code for fast correlation	145
A.3	Alignment of replica code and received code for fine correlation	146

A.4	Alignment of replica code and received code for efficient correlation	148
A.5	Correlation methods for 2 ms correlation	148
A.6	Correlation methods for 10 ms correlation	149
A.7	Correlation methods for 20 ms correlation	149
B.1	Simulator generating IF data file for use in a software receiver	152
B.2	Simulator generating correlator outputs for use in a software receiver	153
C.1	Vector receiver operation in difficult environments	156
C.2	Condition number for vector receiver operation in difficult environments	157
D.1	Tracking error for 9 satellites and multipath-to-direct ratio 0.001.	158
D.2	Tracking error for 8 satellites and multipath-to-direct ratio 0.001.	159
D.3	Tracking error for 7 satellites and multipath-to-direct ratio 0.001.	159
D.4	Tracking error for 6 satellites and multipath-to-direct ratio 0.001.	160
D.5	Tracking error for 5 satellites and multipath-to-direct ratio 0.001.	160
D.6	Tracking error for 9 satellites and multipath-to-direct ratio 0.003.	161
D.7	Tracking error for 8 satellites and multipath-to-direct ratio 0.003.	161
D.8	Tracking error for 7 satellites and multipath-to-direct ratio 0.003.	162
D.9	Tracking error for 6 satellites and multipath-to-direct ratio 0.003.	162
D.10	Tracking error for 5 satellites and multipath-to-direct ratio 0.003.	163
D.11	Tracking error for 9 satellites and multipath-to-direct ratio 0.01.	163

D.12 Tracking error for 8 satellites and multipath-to-direct ratio 0.01.	164
D.13 Tracking error for 7 satellites and multipath-to-direct ratio 0.01.	164
D.14 Tracking error for 6 satellites and multipath-to-direct ratio 0.01.	165
D.15 Tracking error for 5 satellites and multipath-to-direct ratio 0.01.	165
D.16 Tracking error for 9 satellites and multipath-to-direct ratio 0.031.	166
D.17 Tracking error for 8 satellites and multipath-to-direct ratio 0.031.	166
D.18 Tracking error for 7 satellites and multipath-to-direct ratio 0.031.	167
D.19 Tracking error for 6 satellites and multipath-to-direct ratio 0.031.	167
D.20 Tracking error for 5 satellites and multipath-to-direct ratio 0.031.	168
D.21 Tracking error for 9 satellites and multipath-to-direct ratio 0.1.	168
D.22 Tracking error for 8 satellites and multipath-to-direct ratio 0.1.	169
D.23 Tracking error for 7 satellites and multipath-to-direct ratio 0.1.	169
D.24 Tracking error for 6 satellites and multipath-to-direct ratio 0.1.	170
D.25 Tracking error for 5 satellites and multipath-to-direct ratio 0.1.	170
D.26 Tracking error for 9 satellites and multipath-to-direct ratio 0.316.	171
D.27 Tracking error for 8 satellites and multipath-to-direct ratio 0.316.	171
D.28 Tracking error for 7 satellites and multipath-to-direct ratio 0.316.	172
D.29 Tracking error for 6 satellites and multipath-to-direct ratio 0.316.	172
D.30 Tracking error for 5 satellites and multipath-to-direct ratio 0.316.	173

D.31 Tracking error for 9 satellites and multipath-to-direct ratio 1.0.	173
D.32 Tracking error for 8 satellites and multipath-to-direct ratio 1.0.	174
D.33 Tracking error for 7 satellites and multipath-to-direct ratio 1.0.	174
D.34 Tracking error for 6 satellites and multipath-to-direct ratio 1.0.	175
D.35 Tracking error for 5 satellites and multipath-to-direct ratio 1.0.	175

List of Tables

4.1	Mean and variance from simulated scenarios for scalar receiver	72
4.2	Mean and variance from simulated scenarios for vector receiver	72
7.1	System Noise Covariance Matrix Values	126
A.1	Seconds per correlation call for three correlation methods	150

Chapter 1

Introduction

Navigation has seen a boom in usage within the last decade. At the head of this trend is the improvement in size, cost, capabilities, sensitivity and ubiquity of Global Positioning System (GPS) receivers. Having a sophisticated infrastructure built and backed by the US government provided manufacturers with confidence to develop receivers for this system. As more global navigation satellite systems (GNSS) are brought into operation, robust positioning everywhere will become more of a reality. It seems that position information is addictive. In the digital age, there is a desire for more and more information. That information will be integrated into deeper and deeper levels of our lives. Originally designed for use in limited and expensive receivers for military purposes, GNSS receivers have truly proliferated in the past decade. Receivers are located in cars, bikes, phones, and other mobile devices. Now consumers and designers are no longer focused on the how of positioning but on what can be unlocked when position is known. Location based services (LBS) as a field is rapidly expanding with the purchase of smartphones that include GNSS positioning. These systems position with additional network-based information that improves performance [42]. As positioning information is utilized in deeper and deeper levels of system integration, many options and challenges are unlocked for the GNSS receiver. The focus of this dissertation is on some of these advanced ways to improve performance from the same GPS infrastructure by modification of the receiver itself.

Three specific avenues have been pursued:

- Advanced receiver tracking architecture through coupling of the tracking and navigation stages.
- Fault detection and exclusion to improve performance in degraded signal environments.
- Coupling of the receiver with motion information provided by an external sensor.

Of course, there are trade-offs in any design and along with the benefits come other challenges to deal with. On one hand, advanced tracking methods and deep integration of GNSS information and other navigation constraints can improve GNSS receiver tracking performance. This can be accomplished with a vector tracking architecture which can improve not only navigation accuracy but also signal tracking accuracy. The vector architecture was pioneered by techniques published by J. J. Spilker Jr. [29]. Details of a vector architecture will be described in Chapter 3.

However, with the increase in ubiquity of systems like GPS there is also an increase in interferences. Environmental interference has limited the use of GPS in areas such as urban canyons, heavy foliage, and inside buildings. This interference is one of the drivers in advancing tracking methods. Even from the beginning of its design, GPS receivers were made to be operated in adverse and even malevolent environments. In parallel with GPS advances there has also been a segment of development focused on breaking the navigation ability of a GPS receiver. Many ways have been pursued such as jamming [10] and spoofing [27]. With these advances, the fidelity of a receiver has become more critical.

1.1 Contributions

This dissertation develops a robust deeply integrated GPS/INS system for navigation in difficult environments. The specific contributions of this work include the following.

- Details of a GPS software receiver operable in scalar and vector tracking modes.
- Theoretical analysis of superior vector receiver performance in the presence of multipath.
- Simulated and experimental demonstration of improved performance of vector tracking in a multipath environment.
- Analysis of vector tracking limits with maximum error limits as related to discriminator and geometry.
- Novel solution to finding maximum range error from position covariance.
- Development and demonstration of fault detection and exclusion in a centralized vector tracking GPS receiver.

- Algorithm for deeply integrated GPS/INS navigation system with fault detection and exclusion.

1.2 Dissertation Contents

This dissertation catalogs many aspects of GPS operation that can be used in both research and production fields. This section lays out the contents of the dissertation and describes the focus of each chapter. In Chapter 2 the details of a complete GPS scalar receiver are given. This receiver is a software implementation with limited hardware requirements. Complete description of receiver operation provides all relevant background for the rest of the dissertation as well as some help for developers wishing to build and understand a full receiver. A complete software receiver provides flexible means to test new algorithms and data sets. Since the majority of the implementation is in software, development time is drastically reduced. In recent years the fielding of multiple global navigation satellite systems (GNSS) has led to an increase in the use of software receivers. Since software design time is typically shorter than hardware design, a software receiver is highly useful in developing algorithms for new constellations. It can also be useful when investigating constellations whose operation has not been completely defined and therefore limited receivers are available.

Chapters 3 and 4 describe an alternative GPS receiver formulation, the vector tracking receiver. In Chapter 3, the software receiver is extended to operate in a vector tracking mode. Many of the scalar software receiver pieces are used in a vector receiver but significant modifications to the tracking and navigation allow improved performance. As part of the description, a method of comparison between scalar and vector receivers is presented. In Chapter 4, the performance of a vector receiver is compared to that of a standard scalar receiver. This comparison is done with multipath errors in view. Multipath errors arise when reflections of the true signal are received at the same location as direct line of sight signal. Comparison is made on a theoretical basis and validated through simulated and experimental tests.

In Chapter 5 comparison is made between scalar lock detection and the permanent lock of vector tracking. This difference is demonstrated with a divergence experiment in which the vector receiver is able to recover from large position and velocity errors. A new error factor is derived and demonstrated in simulation. This error factor allows position error to range error predictions to

be made without considering the current geometry. This parameter is then demonstrated in a live sky scenario with severe signal blockage.

Chapter 5 presents fault detection in a vector receiver context. Instead of individual channel lock detection, a fault detection and exclusion algorithm is applied to the vector receiver. This novel application of the normalized innovation method to a centralized vector receiver is also demonstrated with live sky results.

What follows in Chapter 7 is the enhancing of a vector receiver with additional sensors. In this chapter, a set of inertial sensors is used to improve performance of both tracking and navigation. Fault detection and exclusion are also included in this framework to further improve navigation robustness. Finally Chapter 8 draws conclusions and indicates possible extensions to this work.

1.3 Nomenclature

Before delving into the remainder of the dissertation, a word about variable layout is appropriate. Consistent use of variables is attempted though there may be occasions when variables may be reused to match usage in other works. In these cases usage differences will be noted. When dealing with variables and the coordinate frame is an important aspect, the frames will be indicated. In general, superscript indicates the coordinate frame of the variable while the two subscript letters indicate the coordinate frames the states refer to. In this work e is the Earth-Centered, Earth-Fixed frame with the z-axis running along the earth spin axis, the x-axis passing through the equator and Greenwich meridian, and the y-axis completing a right-hand coordinate system. A body frame is indicated as b , with the x-axis in the forward direction of the body, the z-axis downward on the body, and the y-axis completing the right-hand coordinate system. Therefore, v_{eb}^e would be the velocity of the body frame (subscript b) with respect to the ECEF frame (subscript e) measured in the ECEF frame (superscript e).

1.4 Summary

This chapter provides a short motivation to the problem faced by GPS receivers. The rest of the dissertation will be spent addressing the contributions to this field. An overview of this sequence is given along with some nomenclature groundwork.

Chapter 2

Operation of a GPS Software Receiver

The software receiver implemented in this work is derived from that presented in [4]. An overview of the stages in a GPS receiver can be seen in Figure 2.1. In this chapter, the process of receiver operation will begin with the positioning aspect of the operation and proceed to the signals that form the solution. First positioning will be described to give the foundation for the reason the receiver operates in the first place. The positioning operation is contained in the Navigation Processor in Figure 2.1. Next, the GPS measurements required to calculate the position will be described. These measurements are generated by the Ranging Processor in Figure 2.1. The next section will describe the GPS signal and how the receiver tracks the signal to generate the desired observables. The acquisition step will then be shown, which kicks off the signal tracking. These operations take place in a Channel context as shown in Figure 2.1. Finally, some comments about the hardware Front End will be given. While this approach is in the opposite direction of the information flow within the software receiver, it is a logical solution-oriented approach.

2.1 Positioning

The purpose of a GPS receiver is the accurate computation of a globally referenced position. Usually this position is reported in geodetic coordinates as latitude, longitude, and altitude above a geoid model of the earth. Positioning of a GPS receiver is analogous to a person navigating based on finding recognizable landmarks and deducing a location based on the known positions of the landmarks. A GPS receiver is able to recognize landmarks (satellites) which have accurate known positions. By using the distances from these landmarks, the receiver can deduce its own position.

2.1.1 Multilateration

The process of calculating a position based on ranges to known positions is known as multilateration. The number of ranges required depends on the dimension of the position to be found. In

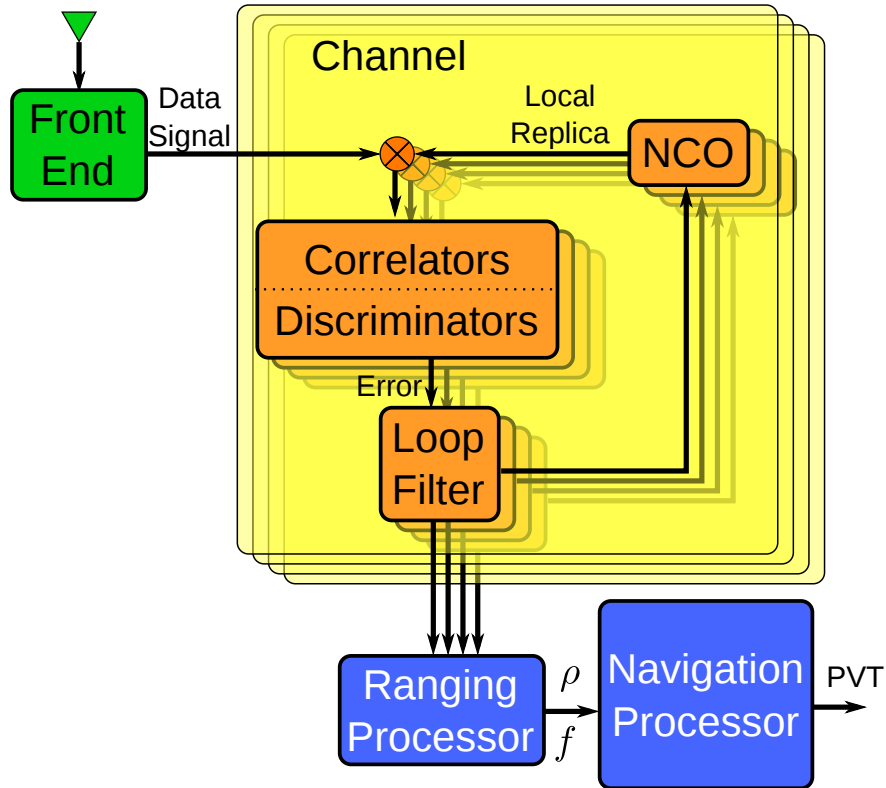


Figure 2.1: Scalar receiver diagram.

the ideal case, to constrain a position in three dimensions requires three known locations. A two dimensional example also illustrates how the multilateration concept works. If a range r_1 is known from a location s_1 then the receiver lies somewhere on a circle centered at the location as shown in Figure 2.2

This infinite set of solutions is still constrained and thus reduces the possible position locations. To further constrain the position, another range r_2 to another location s_2 is required. This case is shown in Figure 2.3.

With these two ranges the position is ambiguously constrained to lie on either of the circle intersections. The geometry of the known locations is important since they must provide the system with independent equations, therefore they should not be co-located. If the transmitters are located near each other, the equations may be mathematically independent but the precision will be degraded. The ambiguity must be resolved in order to determine the location. Therefore a third range will fully constrain the solution to lie on one of the possible intersections, as shown in Figure 2.4

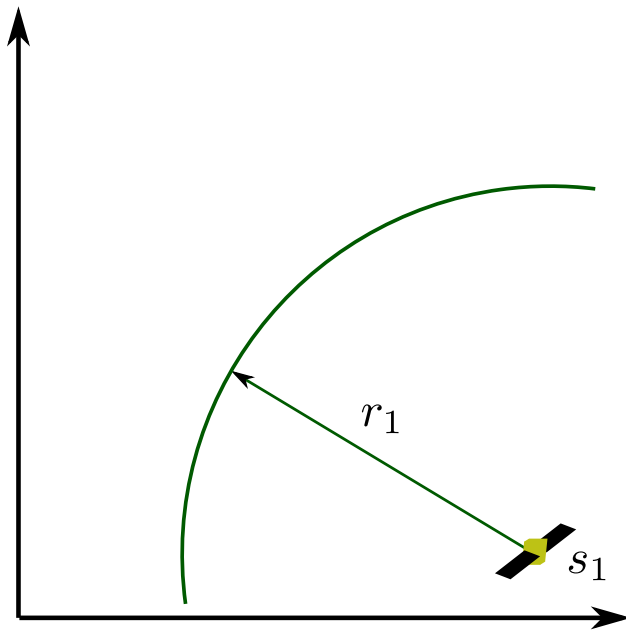


Figure 2.2: Circle of possible positions at range r_1 from position s_1 .

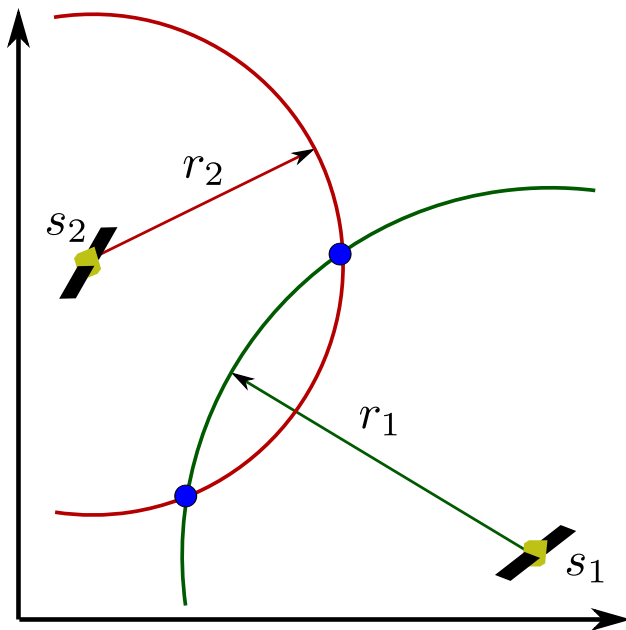


Figure 2.3: Position constrained by multiple ranges to known locations.

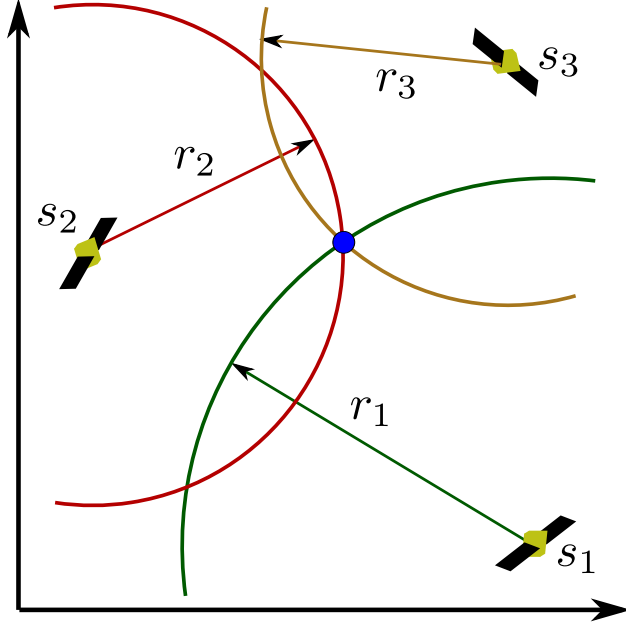


Figure 2.4: Position ambiguity solved with additional range.

This constraint may come from some factor other than an additional satellite range. In Figure 2.3 another constraint is not necessary. Rather, some way to choose from the available solutions is needed. In a GPS receiver, this ambiguity can be solved by assuming the receiver is near the surface of the earth. The other possible solution lies around 20,000 km above the satellites, making the solution nearly 40,000 km above the surface of the earth. In the recursive positioning algorithm described next, this ambiguity is resolved by setting the initial position guess close to the earth's surface. The other solution represents a local minimum and will be ignored in the analysis.

Each range ideally relates the coordinates $(x, y, \text{ and } z)$ of a known location s_i of satellite i to the receiver position u as

$$r_i = \sqrt{(x_u - x_{s_i})^2 + (y_u - y_{s_i})^2 + (z_u - z_{s_i})^2} \quad (2.1)$$

where r_i is the geometric range between the satellite and receiver, and $x_j, y_j, \text{ and } z_j$ are the three dimensional coordinates of j in an arbitrary coordinate frame. This equation can be linearized

about the estimated receiver location $(\hat{x}_u, \hat{y}_u, \hat{z}_u)$ as

$$r_i \approx \hat{r}_i + \frac{\hat{x}_u - x_{s_i}}{\hat{r}_i} (x_u - \hat{x}_u) + \frac{\hat{y}_u - y_{s_i}}{\hat{r}_i} (y_u - \hat{y}_u) + \frac{\hat{z}_u - z_{s_i}}{\hat{r}_i} (z_u - \hat{z}_u) \quad (2.2a)$$

$$r_i \approx \hat{r}_i + a_i \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} \quad (2.2b)$$

where \hat{x} represents the estimate of x , a_i is the unit vector from satellite i to the receiver, and δx is the estimate error for component x .

Since each range r_i has an associated range equation relating the positions of the receiver to the known locations, their linearized versions shown in Equation (2.2) can be put in matrix-vector form as

$$\begin{bmatrix} \delta r_1 \\ \delta r_2 \\ \delta r_3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} \quad (2.3)$$

The solution is solved iteratively by linearizing the range equations about a previous position guess. Since the initial guess is closer to the true position than the alternate solution above the satellite orbits, the iterations will converge to the true solution. However, this iterative linearization assumption can break down when dealing with positions that are close to both possible solutions. This situation appears when localizing with short ranges such as Wi-Fi or other communication radios.

2.1.2 Satellite Positioning

In order to multilaterate a receiver position in the ideal case, the known locations of the signal transmitters must be available. In a GPS receiver, these known locations are the positions of the broadcasting satellites at the time the signal was transmitted. This position information is available as part of the signal transmitted by each satellite. Rather than broadcast the satellite position as part of the data message, the message instead contains orbital parameters to generate the position at any required time. As part of the GPS design, each satellite broadcasts a data message which contains information required to calculate the satellite position at any time. The

standard algorithm for calculating the satellite positions are found in [16]. Another description of this algorithm is shown in [13], which also relates the orbital parameters to the satellite velocities. These parameters were chosen with the trade-offs of solution accuracy as opposed to data length requirements. More accurate parameters would give slightly better satellite positions but would require more time for the individual parameters to be broadcast. Satellite position is very sensitive to these parameters. Only slight changes in the values yield large changes in satellite position. To ensure accuracy of the received signals, there are several checksums inserted in the navigation message.

Although the satellite clocks are highly accurate and stable atomic clocks, they are slightly biased from GPS time. Since timing is such an important part of GPS operation, these clock offsets are also included in the navigation data message. The clock error is modeled as a quadratic equation about a fit time. The quadratic term of the equation is typically zero with the current set of satellites due to the fact that they have such high stability.

A receiver generates the range measurements from estimates of time, as

$$\tau_i = t_{ri} - t_t = r_i/c \quad (2.4)$$

Here τ_i is the transit time for satellite i , t_{ri} is the received time for satellite i , t_t is the transmit time of the signal, and c is the speed of light constant. These two measured times are the estimates at which positioning and ranges are desired.

The algorithm described in [16] yields the satellite solutions in an Earth-centered Earth-fixed (ECEF) frame. This frame is rotating with the earth and provides consistent coordinates regardless of the time of rotation. The satellite position must be calculated for the transmission time t_t . However, when the range is formed later at t_{ri} the ECEF frame has rotated and the satellite position in the current frame is needed. Therefore there is a correction that must be made at the satellite position to compensate for this rotation. Equation (2.1) is really only valid in an inertial frame; one in which the frame itself is not accelerating. Due to the rotational motion of the ECEF frame, the range equation is not strictly true when using positions from ECEF frames at different times. This situation is illustrated below in Figure 2.5. When the satellite transmits its signal, the transmission ECEF (e_t) frame can be considered frozen as an inertial frame. In Figure 2.5, the red

star represents the satellite position at the time of transmission in the transmission ECEF frame, s^{e_t} . The green square is the receiver location near the earth's surface in the same frame, u^{e_t} .

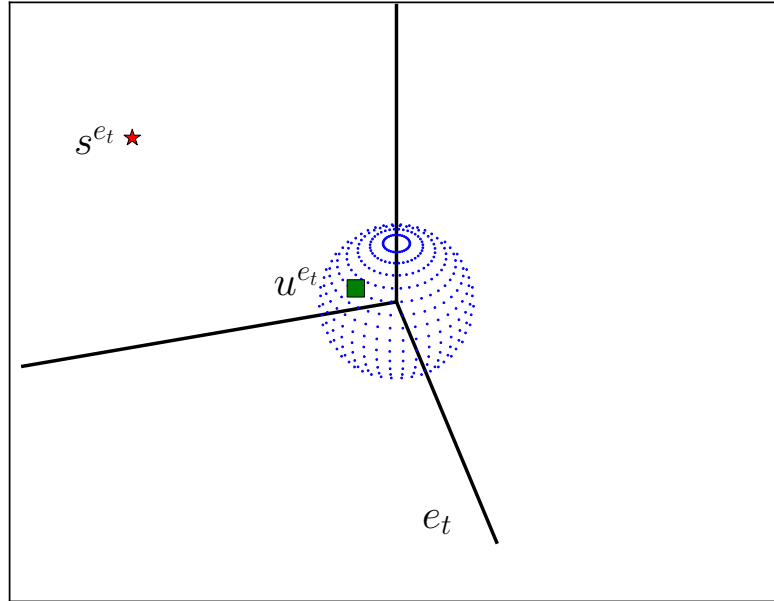


Figure 2.5: Satellite and user position at transmit time, shown with inertial reference ECEF frame.

The satellite signal is travelling at the speed of light but still requires about 70 ms to reach the receiver. During this time the earth has rotated to a new orientation, shown exaggerated below in Figure 2.6. When the signal is received, the ECEF frame has rotated in inertial space to a frame at the reception time (e_r). The transmit satellite position, s^{e_t} , and transmit ECEF frame, e_t , are shown in Figure 2.6 faded. The new red star represents the previously calculated coordinates but applied in the new ECEF frame, the ECEF received frame e_r . This satellite position is referenced as s^{e_r} . The ECEF coordinates for the two red positions are the same; they are referenced to two different ECEF frames. This position difference shows how applying the direct line of sight range equation to the moving ECEF frames will produce erroneous results, effectively biasing each range. This bias is not consistent as ranges are contracted and lengthened on opposite azimuths to the receiver position. Therefore the bias will not be lumped with the clock bias but should be corrected before the position solution is calculated.

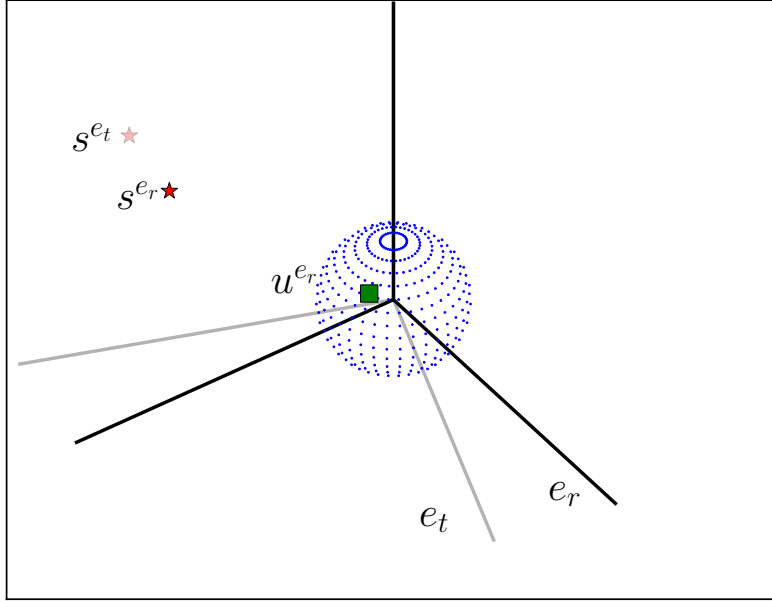


Figure 2.6: Satellite and user position at exaggerated reception time with new ECEF frame and original ECEF frame and satellite position.

A way to correct for this motion between ECEF frame times is to apply a rotation to the satellite positions to bring it into the reception time frame [22]. This gives the coordinates of the faded star in Figure 2.6 in the reception time ECEF frame. This is done as

$$\bar{x}_i = \begin{bmatrix} \cos(\Omega_e \tau_i) & \sin(\Omega_e \tau_i) & 0 \\ -\sin(\Omega_e \tau_i) & \cos(\Omega_e \tau_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{x}'_i \quad (2.5)$$

where Ω_e is the WGS84 defined value for the Earth's rotation rate and τ the difference between the received and transmission times as shown in Equation (2.4).

2.1.3 Clock Bias

Even the highly accurate satellite clocks have time offsets that must be taken into account [16]. This is even more true for a low cost receiver clock. The receiver clock offset from GPS time is completely unknown in the receiver. As in Equation (2.4), the range is highly dependent on

the transit time being accurate. When the receiver does not know its own received time because of the unknown offset between its clock and GPS time, this ambiguity must be solved. When pseudoranges are ready to be formed, the receiver does not know this offset. The severity of the offset depends on many factors but is often taken to be completely unknown. However, once signals are available the receiver can read the transmit time on each of the signals and initialize its own clock. Any residual error will affect all ranges, but by the same amount. Since this clock error is common to all satellite ranges, the range in Equation (2.1) becomes

$$\rho_i = \sqrt{(x_u - x_{s_i})^2 + (y_u - y_{s_i})^2 + (z_u - z_{s_i})^2} + cb \quad (2.6)$$

where b is the clock bias, c is the speed of light, and cb represents the clock bias in meters. Due to the clock bias, the measurement is not actually a range but is called a pseudorange, ρ . The equation can still be linearized about the estimated position and clock bias.

$$\rho_i \approx \hat{\rho}_i + \frac{\hat{x}_u - x_{s_i}}{\hat{r}_i} (x_u - \hat{x}_u) + \frac{\hat{y}_u - y_{s_i}}{\hat{r}_i} (y_u - \hat{y}_u) + \frac{\hat{z}_u - z_{s_i}}{\hat{r}_i} (z_u - \hat{z}_u) + (cb - \hat{c}b) \quad (2.7a)$$

$$\rho_i \approx \hat{\rho}_i + \begin{bmatrix} a & 1 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta cb \end{bmatrix} \quad (2.7b)$$

Since each pseudorange ρ_i has an associated range equation relating the positions of the receiver to the known locations, their linearized versions like Equation (2.7) can be put in matrix-vector

form as

$$\begin{aligned}
 \begin{bmatrix} \delta\rho_1 \\ \vdots \\ \delta\rho_N \end{bmatrix} &= \begin{bmatrix} a_1 & 1 \\ \vdots & \vdots \\ a_N & 1 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta cb \end{bmatrix} \\
 &= C \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta cb \end{bmatrix}
 \end{aligned} \tag{2.8}$$

where there are N ranges. When there are four ranges, the matrix C can be inverted (provided the rows are linearly independent) and the corrections solved for. When more than four ranges are available, the four unknowns are overdetermined and can be solved in a least squares sense using the pseudoinverse $(C^T C)^{-1} C^T$ [13].

2.2 GPS Observables

In order to perform position calculation, measurements must be generated from the incoming signals. These measurements are calculated by a ranging processor as shown in Figure 2.7, repeated here from Figure 2.1. Each observable takes information from the tracking channels and forms values that are meaningful in a navigation context.

2.2.1 Pseudorange

Pseudoranges were introduced in Section 2.1.3 for calculating user position. The formation of these measurements is done by recording and transforming received signal times. Since the satellites are able to transmit at highly predictable times, all in synchronization with one another, the receiver can accurately tell when a signal is received. This transmission stability is due to the highly accurate and stable atomic clocks used to drive the signals. Since the entire constellation is made up of accurate clocks, the satellites can be considered synchronized. The time of transmission is encoded into the satellite signals and is read by the receiver. By doing so, the receiver has a very

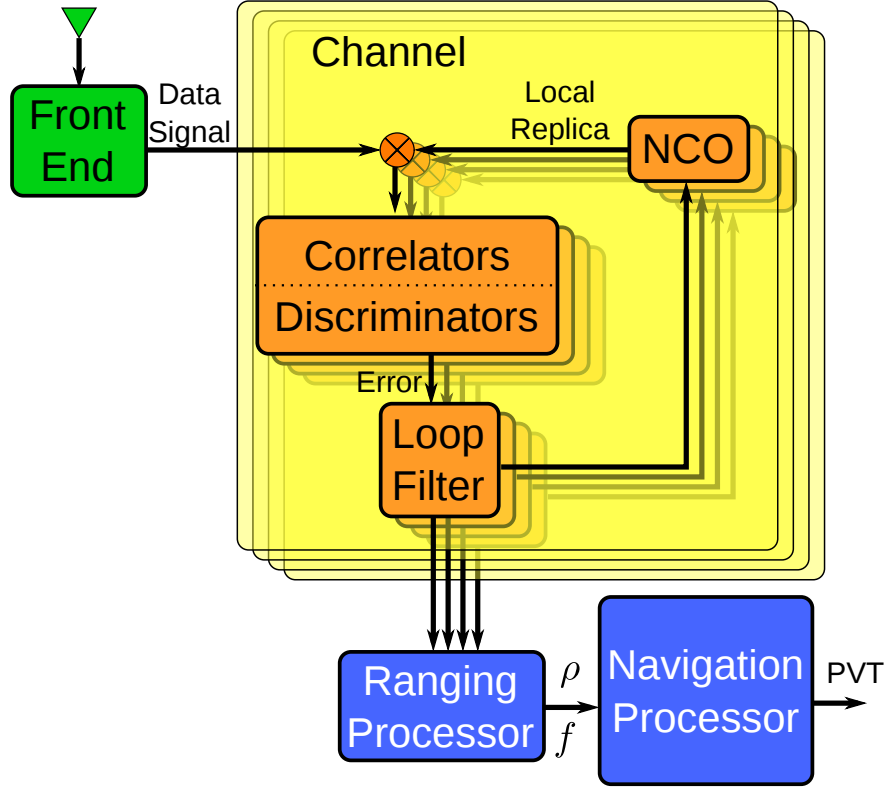


Figure 2.7: Scalar receiver diagram.

accurate record of when the same portion of the different signals was received. From this time and the received time, the travel time can be calculated. Since electromagnetic waves propagate at the speed of light, c , the pseudorange can be calculated as

$$\rho_i = c(t_t - t_{ri}) = c\tau_i \quad (2.9)$$

However, calculation of pseudoranges is slightly more complicated because the received time t_{ri} for satellite i is based on the receiver clock. This clock is typically of low quality and is offset from GPS time as discussed in Section 2.1.3. This bias manifests when forming the pseudoranges from tracking results. The situation is illustrated in Figure 2.8. Since the receiver can tell when a synchronized part of the satellite signal is received, this time can be marked. When all tracked signals have received that part of the signal, the time delay between different received time δ_{ij} can be calculated where i and j are different satellites and

$$\delta_{ij} = t_{rj} - t_{ri} \quad (2.10)$$

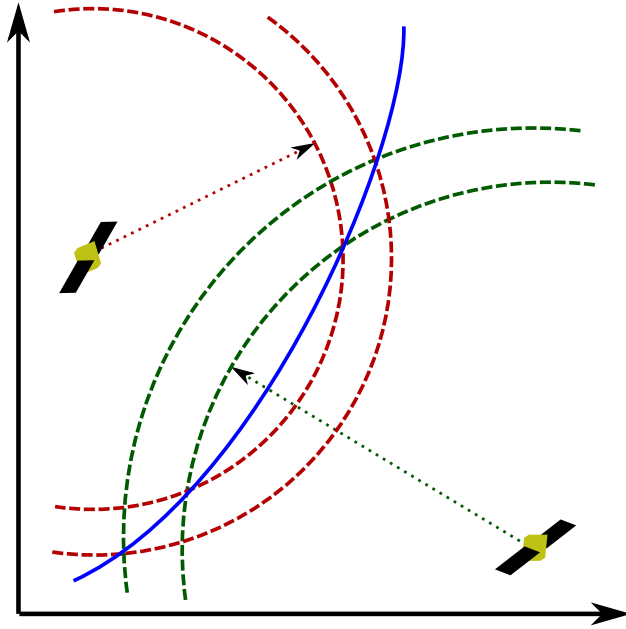


Figure 2.8: Pseudorange ambiguity showing receiver clock offset.

Pseudoranges are formed by assigning some nominal transit time to the shortest pseudorange, typically around 68 ms. With that pseudorange 'set', the other pseudoranges can be formed by adding the δ_{1j} to the nominal pseudorange. This process is illustrated below in Figure 2.9.

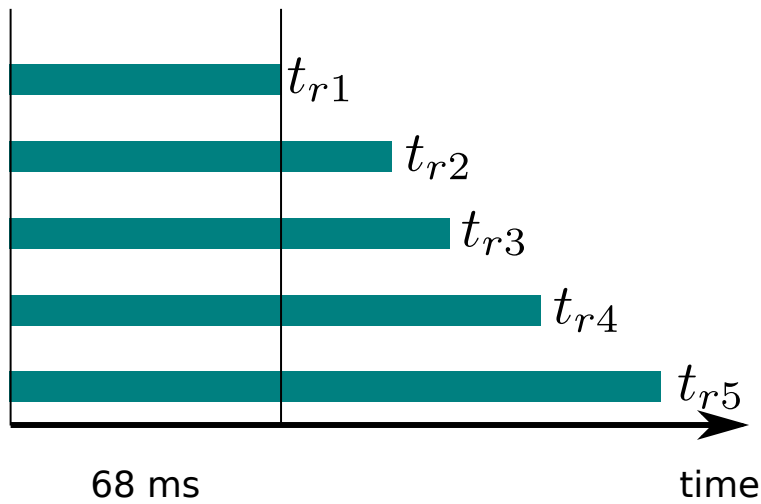


Figure 2.9: Pseudorange formulation from received times.

In Figure 2.9 the bands represent the first estimate of pseudorange, given some initial guess at the shortest pseudorange. Since the receiver clock is able to calculate the received time on its ambiguous clock very well, subsequent pseudoranges are calculated with respect to the arrival

time of this shortest range. After the navigation solution is calculated, these pseudoranges can be corrected and adjusted to match the true range. Some hardware receivers continuously steer their clock bias to zero by adjusting the pseudoranges at each epoch. However, other receivers wait until the clock bias hits a limit before applying a correction to all pseudoranges at once.

2.2.2 Doppler Frequency

While the pseudorange relates to the distance between the receiver antenna and the satellite transmitting antenna, the carrier frequency is related to the velocity of both. When talking about the carrier frequency, the quantity is usually quoted as a Doppler frequency shift from the center frequency of the signal. The Doppler frequency arises because relative motion between the transmitter and receiver actually alters the apparent wavelength received at the antenna. This is shown in Figure 2.10

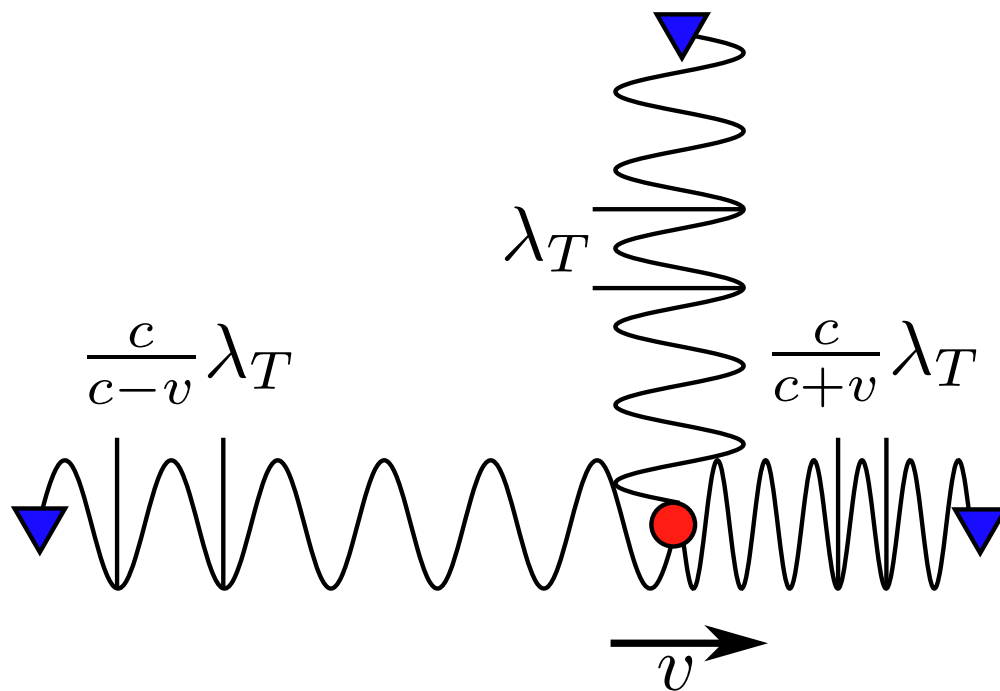


Figure 2.10: Illustration of Doppler shift due to relative motion.

If there is no relative motion between the transmitter and receiver, there is no Doppler shift since the wavelengths are unchanged. If a signal is transmitted, its propagation velocity can be taken as the speed of light, c . If there is a shortening of the distance between the transmitter and receiver at the rate of v , then there is a shortening of the wavelength. If the wave were being

received at a static position, the wave peaks at a consistent period (and distance). The wave period is given by $\frac{\lambda}{c}$. The change in period is a function of the velocity as $\frac{\lambda_R - \lambda_T}{v}$ where R indicates received and T indicates transmitted. Since a signal's frequency is $f = \frac{c}{\lambda}$, this gives the Doppler relation

$$f_R = f_T \left(1 - \frac{v}{c}\right) \quad (2.11)$$

The code component of the signal is also affected by the Doppler shift, being stretched or shrunk by the motion of the receiver and satellite. However, the effect is much smaller on the code because the transmit frequency of the C/A code is much lower (1.023×10^6 samples per second vs 1575.42×10^6 cycles per second). Within a receiver, the Doppler shifts are proportional by the ratio of the transmit frequencies

$$f_{D,\text{code}} = f_D \frac{1.023 \times 10^6}{1575.42 \times 10^6} = f_D \frac{1}{1540} \quad (2.12)$$

where $f_{D,\text{code}}$ is the Doppler frequency shift on the code.

2.2.3 Carrier Phase

The carrier phase of the received signal is another observable that can be used for navigation purposes. The pseudorange is derived from the phase of the code tracking loop in a similar way that carrier phase is found from the carrier tracking loop. Both of these observables are measurements of range but they have some significant differences. These differences are illustrated below in Figure 2.11.

The first difference is the precision of the measurements. Both phase measurements are accurate to about 1 – 2% of the wavelength of the signal. Since the wavelength of a single code chip is

$$\lambda_{\text{code}} = \frac{c}{1.023 \times 10^6} \approx 300\text{m} \quad (2.13)$$

then the code phase (pseudorange) is accurate to about 5 m. Similarly the carrier phase is measurable with a wavelength of

$$\lambda_{\text{carrier}} = \frac{c}{f_{L1}} \approx 19\text{cm} \quad (2.14)$$

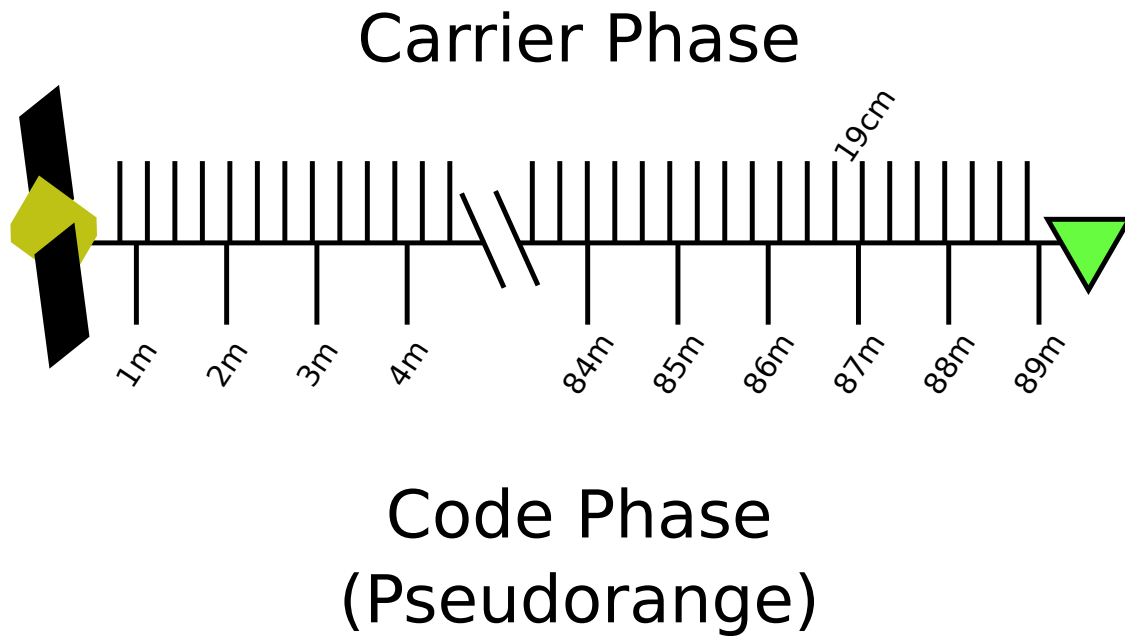


Figure 2.11: Range measurements from carrier phase and code phase.

Then the carrier phase is accurate to about 2 mm. The carrier phase is highly precise and many applications make use of this precision.

The second difference is also illustrated in Figure 2.11. Notice that the carrier phase has only one piece of information that relates the phase to a range measurement. That value is the wavelength. Since the carrier wave is a simple sinusoid, each wave peak is indistinguishable from any other wave peak. While the measurement is very precise, there is an unknown number of wavelengths between the satellite and the receiver. This discrete number of wavelengths is called the integer ambiguity. Any application that uses the carrier phase as a range must solve for this wavelength ambiguity since the fractional phase is the portion that is measured. The pseudorange, however, is not ambiguous since each code period (and code chip) can be uniquely identified by its alignment with the data message. Therefore pseudorange is not an ambiguous measure the same way the carrier phase is ambiguous. The trade-off is that it is a couple of orders of magnitude less accurate.

While both range measurements can be used in a positioning application, the precision of the carrier can be used to propagate the pseudorange. This technique is called carrier smoothed code and is a blending of the benefits of both measurements. More details for this technique can be found in [37].

2.3 Channel Tracking

The observables in a GPS receiver come from the received signals. In order to generate the observables used by the navigation portion of the receiver, the incoming signals must be made available to the receiver. This is done by generating a local replica of the signals and maintaining this replica to match the received signal. The process of updating the local replica to changes in the received signal is called tracking. This process is usually done in a channel structure as shown below in Figure 2.12.

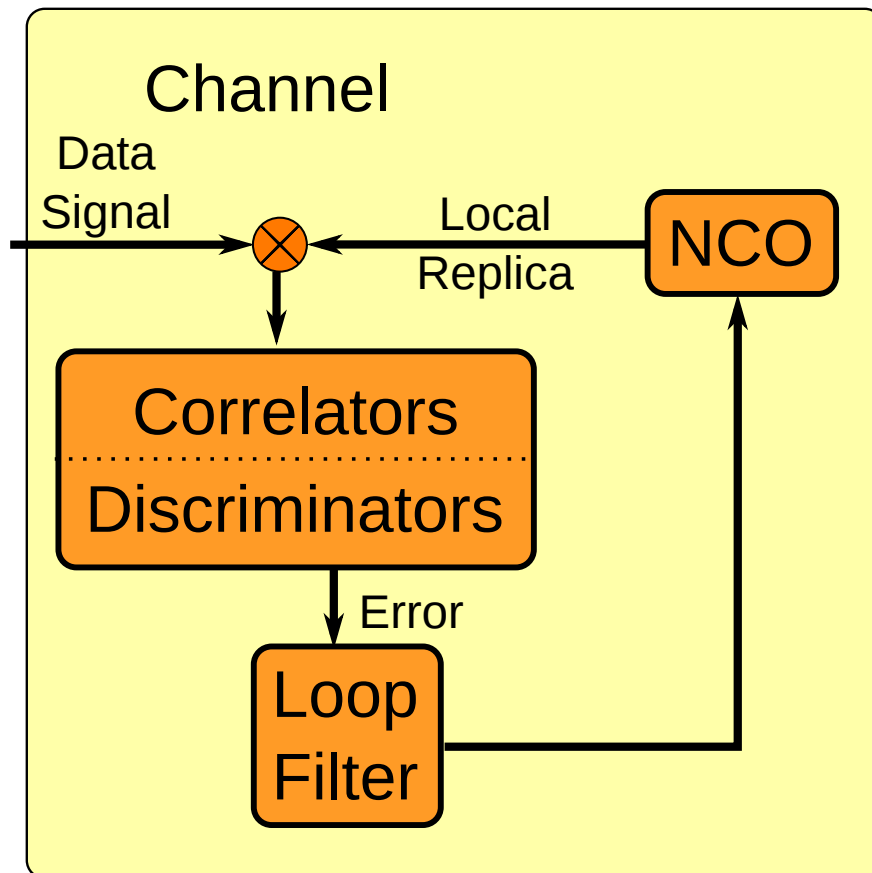


Figure 2.12: Tracking channel diagram.

2.3.1 GPS Signal

The GPS signal is transmitted by each satellite, all synchronized to GPS time. The base of the signal is a basic sinusoidal carrier wave. This carrier wave defines the center frequency of the total signal in the frequency domain. Each satellite uses two different frequencies in the L band. At the

GPS L1 frequency (1575.42 MHz) the satellites transmit two signals in phase quadrature as defined by the GPS interface definition [47]. These two signals contain different code modulations, the course acquisition (C/A) and precise encrypted (P(Y)) codes. At the L2 frequency (1226.7 MHz) only the P(Y) code is modulated. Since the P(Y) code is currently running in encrypted mode, only the C/A code available on L1 is trackable by civilian users. Also transmitted on the signal is a 50 bps data message used to provide constellation information to the user receiver. Newer satellites are being designed to transmit in the L5 band (1176.45 MHz) to give more frequency disparity between signals [18]. There are also other ranging codes being specified such as the L2C code for civilian use of the L2 frequency and the M code, another military ranging code. These codes are not considered here since they are not broadcast by the full constellation at the time of this writing.

The transmitted codes are deterministic but appear as random bit sequences. Therefore they are referred to as pseudorandom noise, or PRN. The generation of the codes is specified in the GPS Interface Control Document, [16] that is freely available to receiver designers. For all users, the C/A code is a 1023 bit sequence assigned to each satellite that repeats every millisecond. For approved military users, the P(Y) code is a truncated sequence that is 6.1871×10^{12} bits long, repeated every week. Since the P(Y) code is so long, the C/A code was originally designed to provide military users sufficient information to begin tracking the signal and then transition to the P(Y) code [16].

The transmitted signal from a single satellite can be modeled as

$$s_i(t) = \sqrt{2P_{C_i}}C_i(t)D_i(t) \cos(2\pi f_{L1}t) + \sqrt{2P_{P_i}}P_i(t)D_i(t) \sin(2\pi f_{L1}t) \quad (2.15)$$

Here i is the i th satellite, P_C is the power level of the C/A portion of the signal, P_P is the power level of the P(Y) portion of the signal, C is the C/A pseudorandom noise sequence, P is the P(Y) pseudorandom noise sequence, D is the 50Hz data message, and f_{L1} is the L1 frequency. When the signal is received, there is a delay due to the transmission time, τ_i , as well as Doppler frequency shifts, phase differences, and measurement noise. Therefore the received signal can be

modeled as

$$s_i(t) = \sqrt{2P_{C_i}}C_i(t - \tau_i)D_i(t - \tau_i) \cos(2\pi(f_{IF} + f_{D_i})t + \phi_i) + \sqrt{2P_{P_i}}P_i(t - \tau_i)D_i(t - \tau_i) \sin(2\pi(f_{IF} + f_{D_i})t + \phi_i) + \eta_i \quad (2.16)$$

where f_D is the Doppler frequency shift, ϕ is the received carrier phase, and η is the measurement noise.

2.3.2 Correlation

It is not directly useful to look at the raw received signal and try to directly estimate the delay and received carrier frequency. This is due to the fact that the received signal is actually below the noise floor. This means that the signal is so weak it cannot be directly detected because its received amplitude is less than that of thermal noise, which is always present on the signal measurements. The ranging C/A code modulated onto the carrier spreads the energy of the signal among the frequency domain, pushing it below the noise floor. Reversing this process lifts the signal above the noise floor. The signal is also shifted down in frequency to baseband, meaning the carrier frequency effect has been removed. These two steps are accomplished by mixing the incoming signal with its local replica in an in-phase (I) and quadra-phase (Q) correlator. The I correlator is generated using a cosine term at the replica frequency (carrier and Doppler) and the Q correlator is generated from a sine term.

After the local replica and received signal are multiplied and accumulated, the model of the correlation can be taken as

$$I(k, \tau) = AR(\epsilon + \tau)D(k) \cos(\pi fT + \theta) + \eta_I(k) \quad (2.17a)$$

$$Q(k, \tau) = AR(\epsilon + \tau)D(k) \sin(\pi fT + \theta) + \eta_Q(k) \quad (2.17b)$$

where A is the correlation amplitude, R is the correlation function, and T is the correlation time.

Since the correlators are functions of the delay τ and frequency f , they are used to solve for these error quantities. Correlation is a very operation intensive process and is typically implemented in hardware for efficiency. In a software receiver, however, this process is typically done on a general

purpose processor in the main software routine. More information about the implementation details of several correlators is given in Appendix A.

2.3.3 Discriminators

The discriminator function is the portion of the tracking algorithm that takes the correlator outputs in Equation (2.17) and generates the error signal. Several different kinds of correlators can be used with differing efficiencies and accuracies [4]. Three general classes of discriminators generate alignment error based on the tracking loop type: code phase, carrier phase, and carrier frequency. To keep the local replica aligned, both code and carrier loops are required but the receiver has the option of maintaining carrier phase or carrier frequency lock. The carrier phase has a much higher accuracy when used as a range measurement but its tracking loop is the first to fail when the signal becomes weak or blocked. On the other hand, the frequency lock ignores the phase of the carrier and attempts to zero the frequency error. This process is much more robust but at the expense of carrier accuracy. Another key observation is that when carrier phase is not tracked, the navigation message bit error rate is higher [11]. The discriminators used in this dissertation are chosen for their highly linear properties in a wide range around perfect tracking (zero tracking error).

Code tracking makes use of the C/A code correlation peak, which is a function of relative delay, τ , as

$$R(\tau) = \begin{cases} \frac{1-|\tau|}{\delta} & -\delta < \tau < \delta \\ 0 & \text{otherwise} \end{cases} \quad (2.18)$$

where δ is the chip width of a single C/A code value. This peak is shown below in Figure 2.13.

When the received signal correlates with a properly aligned local replica, the correlation result is maximized since all samples will properly align in the correlation. However, only using the local replica gives no indication as to which side of the correlation peak the delay is on. The correlation value is the same at -0.2 chips as 0.2 chips. Tracking error is instead generated by intentionally delaying and advancing other local replicas, called the early and late replicas. The best guess of the local replica is called the prompt replica and is desired to follow the maximum value of the correlation peak. The correlator spacing is the distance between the early and late correlators, shown in Figure 2.13 as 1 chip.

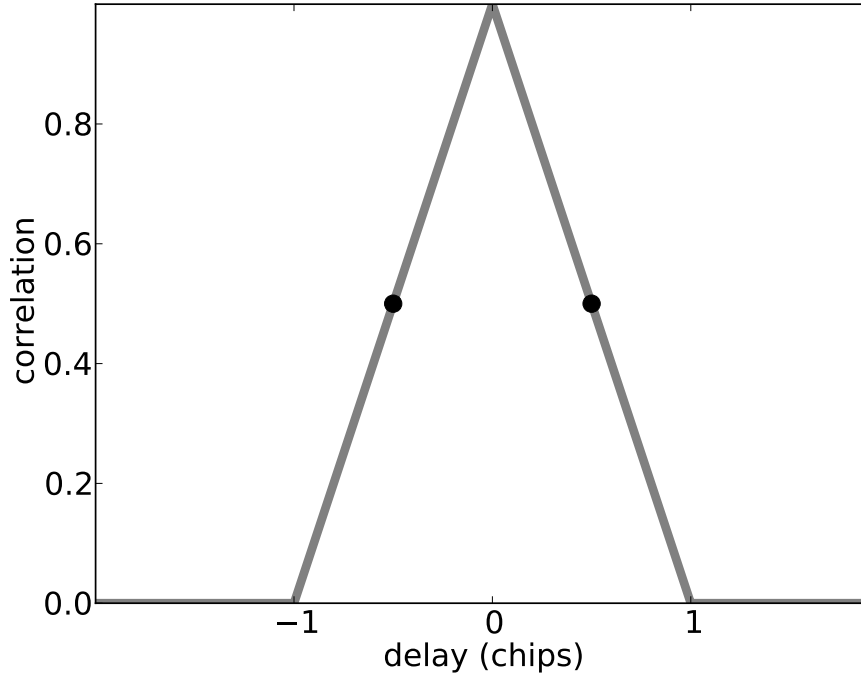


Figure 2.13: Correlation peak and code tracking discriminators.

Since these values are generated on both sides of the peak, any shift between the prompt correlator and the received signal will appear as a shifted correlation peak, shown below in Figure 2.14. The local replica signal may or may not be in lock in the carrier phase, therefore both in-phase and quadra-phase (cos and sin) correlators are used. This allows the channel to span the entire range of possible phase differences, modulo 2π .

The code tracking discriminator generates pseudorange error based on the early and late correlators as

$$\frac{\sqrt{IE^2 + QE^2} - \sqrt{IL^2 + QL^2}}{\sqrt{IE^2 + QE^2} + \sqrt{IL^2 + QL^2}} = \tau \quad -\frac{b}{2} < \tau < \frac{b}{2} \quad (2.19)$$

Here b is the chip width. This discriminator normalizes the early and late powers with respect to the signal amplitude. The output over a range of delays is shown below in Figure 2.15. From this figure, as long as the delay is within half a chip from zero delay, the discriminator output is linear with respect to the delay. This fact is very helpful since the tracking loop filter is designed with linear theory in mind. Notice that if the delay falls outside this range, there is still a meaningful output until the delay is $\frac{3}{2}$ of a chip away (450 m). The fall off occurs when either the early or late

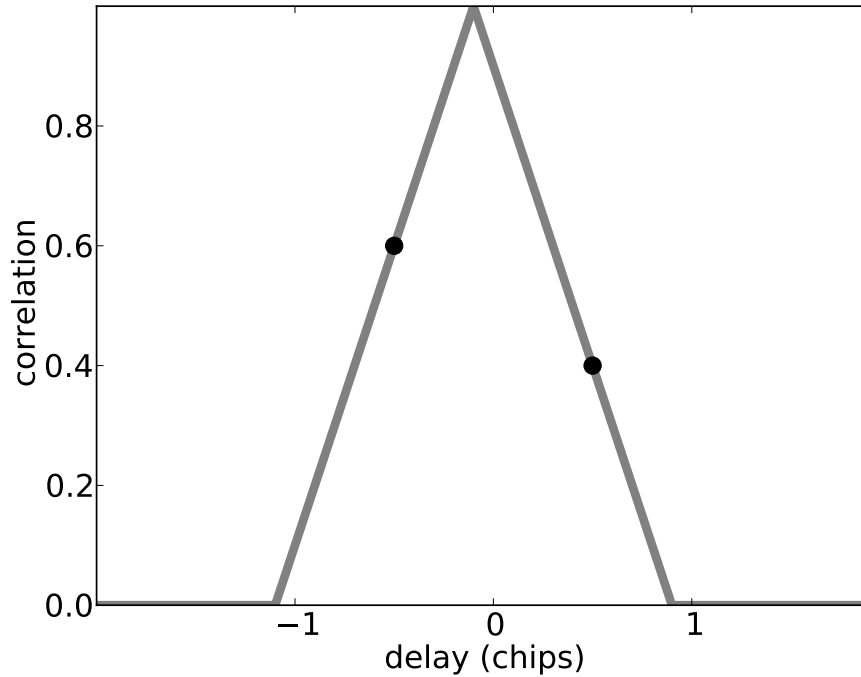


Figure 2.14: Shifted correlation peak and code tracking discriminators that show tracking error.

correlators fall outside the correlation peak shown in Figure 2.13. At that point, their output is essentially zero, meaning it makes no contribution to the correction.

The carrier phase tracking discriminator provides the phase error as

$$\tan^{-1} \left(\frac{QP}{IP} \right) = \phi \quad (2.20)$$

where ϕ is the phase error between the local replica and the received signal. A range of outputs from this discriminator is shown in Figure 2.16. This discriminator is linear within $\pm\pi$ which provides a good measurement of error for the carrier phase discriminator. Outside this linear region the discriminator will cycle slip and begin to track the signal at a different linear region about a different cycle. The linear region spans a full wavelength of the carrier, or 19 cm. This high precision is also the most vulnerable to error when the signal strength gets low.

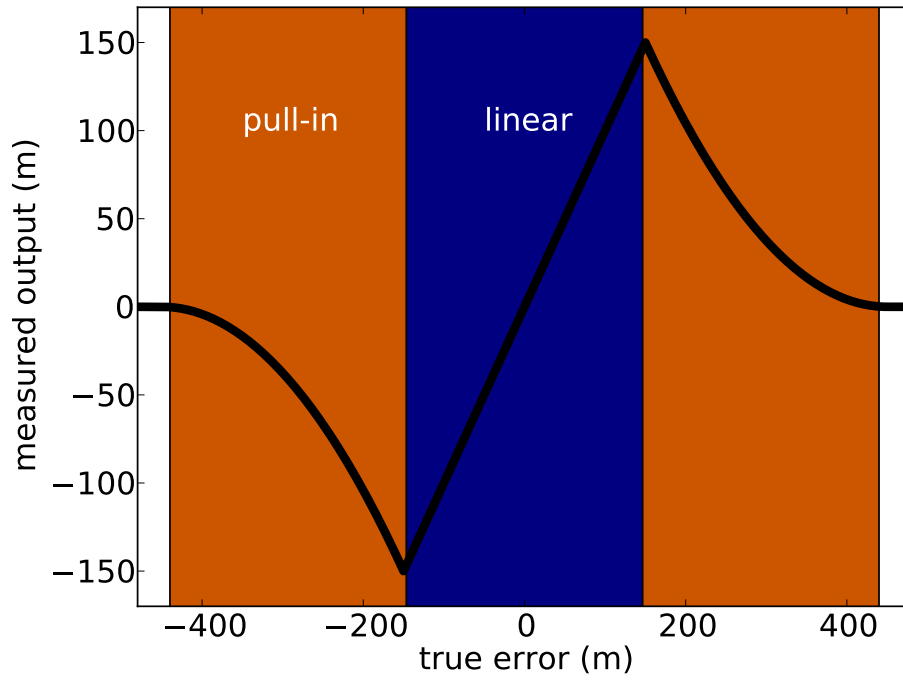


Figure 2.15: Linear and pull-in regions for the code discriminator.

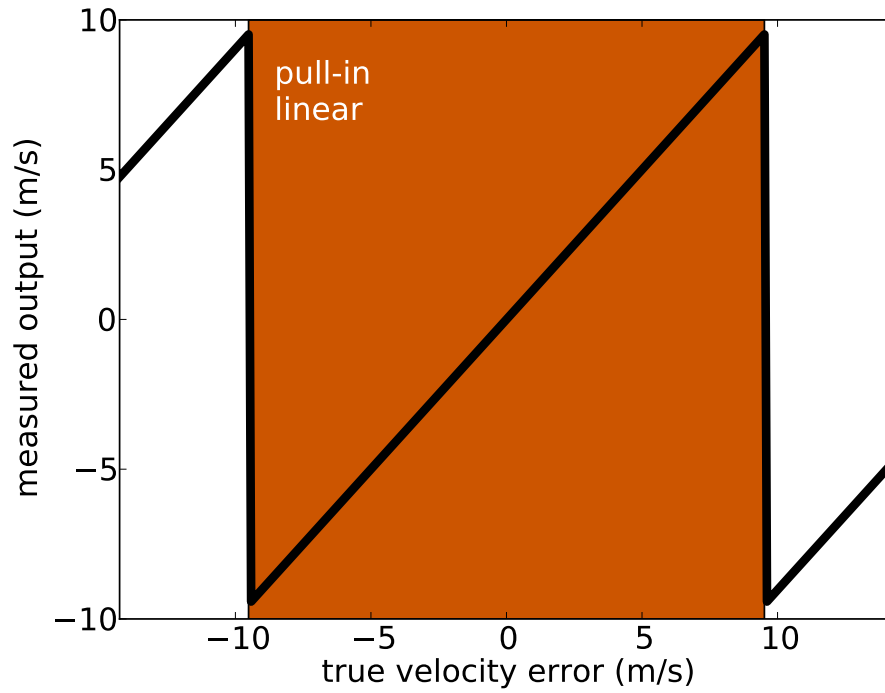


Figure 2.16: Linear and pull-in regions for the carrier discriminator.

2.3.4 Loop Filter

A major determiner of tracking performance is the channel's loop filter as shown again in Figure 2.17. This filter is tasked with the job of transforming a noisy error signal into alterations in the local replica. The ultimate goal of the loop filter is to drive this error signal to zero. When the ranging processor applies to a channel's results, it takes the output of the in-phase prompt signal to represent the best estimate of the signal.

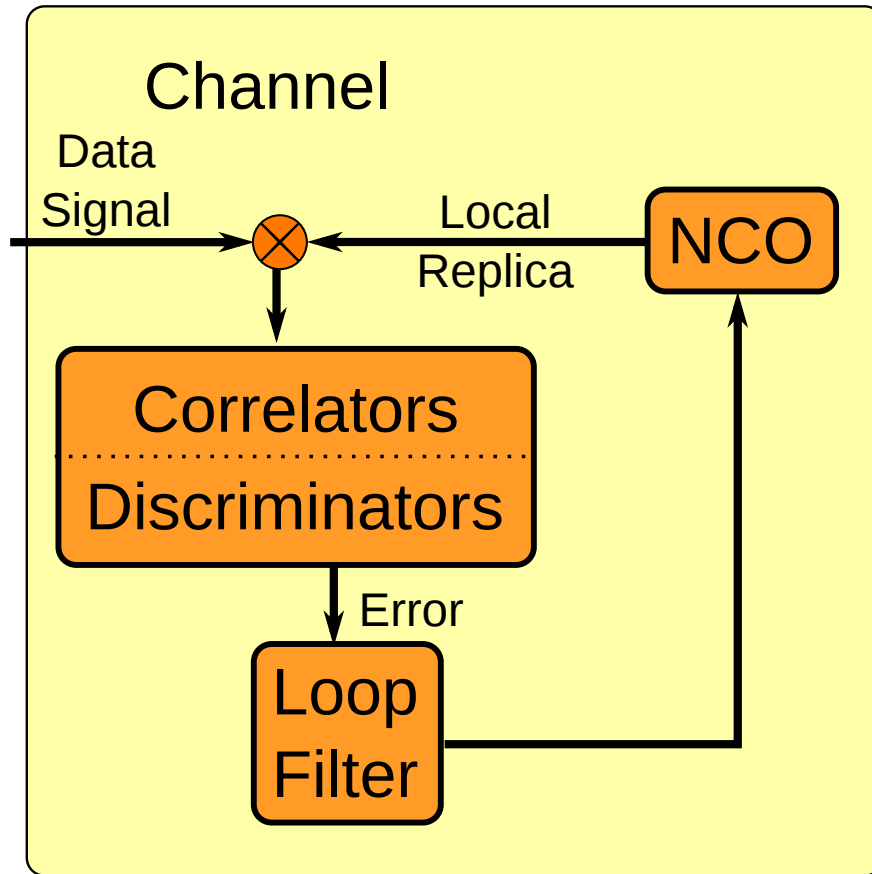


Figure 2.17: Tracking channel diagram.

The loop filter is treated as a SISO system that can be designed to modify the code and carrier frequencies. This allows the local replica to easily track the change in range, or range rate, of the signal over time. The performance of the filter is altered by changing the loop order and associated parameters. Because of the similarity in the design of the code and carrier loop filters, they are often presented with common filter details and then described with different parameters. A particular implementation of the phase lock loop filter presented in [4] is shown in Figure 2.18.

There the performance of the tracking loop is determined by the selection of the parameters c_1 and c_2 . Calculation of these variables is described in [4].

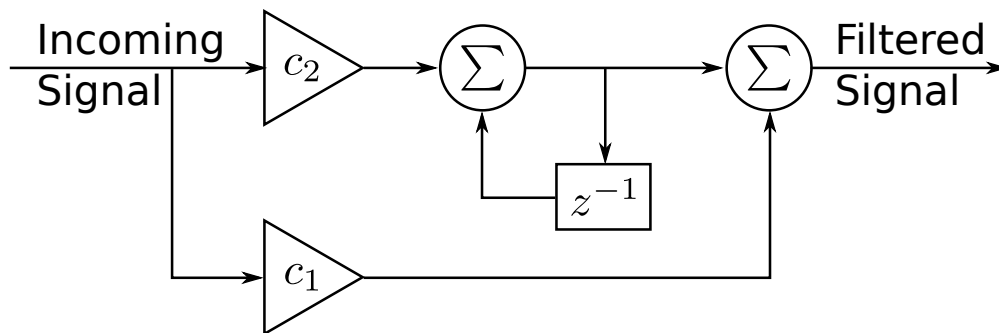


Figure 2.18: Generic phase locked loop for code or carrier scalar tracking loops.

2.3.5 Numerically Controlled Oscillator

After filtering the error signal, the resulting frequency error output is used to drive the numerically controlled oscillator (NCO). The NCO is tasked with generating the local signal replica. This is done by keeping a continuous phase and updating the replica frequency for both code and carrier loops. The NCO acts as a simple integrator by using the current frequency to drive the phase of the code and carrier signal components for the next correlation period. In this case the phase of the signal remains continuous and there are no instantaneous jumps in phase as the channel tracks the signal. However, the shifting of the replica frequency has the effect of slowing or speeding the advance of the phase. In this way the NCO can maintain phase lock (when the loop is designed for this) simply by maintaining continuous phase and updating the frequency. In other receiver architectures such as vector tracking, the replica generation is more elaborate. But in this case keeping track of two frequencies allows the loop to remain in lock, generating an accurate local replica of the received signal.

2.4 Acquisition

In order to kick off tracking described in Section 2.3, an initial local replica guess must be generated. The tracking loops use the replica to generate errors to close the tracking loop. However, these errors are only valid for given ranges about the perfectly aligned codes. This is apparent from considering the discriminator operating ranges shown in Figures 2.15 and 2.16. There are several

methods for searching for signals that can be tracked. Each of these methods must generate an initial value of the code phase and carrier frequency to kick off the channel operation.

2.4.1 Serial Code Phase Acquisition

The first method of acquisition typically implemented is the serial search acquisition. In this method, the search space (code phase and carrier frequency) are generated over a viable range for a single satellite. Since the full C/A code repeats every millisecond, the code search space defaults to the full 1 ms. Smaller search spaces are possible if some information is known about the position and time at the receiver and satellite. For a static receiver, the carrier frequency range is usually given as ± 5 kHz and ± 10 kHz for moderate velocity receivers. The actual search range is immaterial to the algorithm, it only takes longer to search over more frequency slots.

Once the code phase and frequency range are determined, they are cut into ranges for a grid search over the full range of value combinations. The grid search is applied to each PRN in turn to see if that signal is available. Each space of the grid corresponds to a guess at the carrier frequency and code phase offset from the beginning of the signal under consideration. A shifted local replica of the C/A code is generated along with the in-phase and quadrature carrier replica. These samples are multiplied and integrated together to generate the I and Q correlations, which are squared and summed to be the test statistic for that part of the search space. This sequence is shown in Figure 2.19.

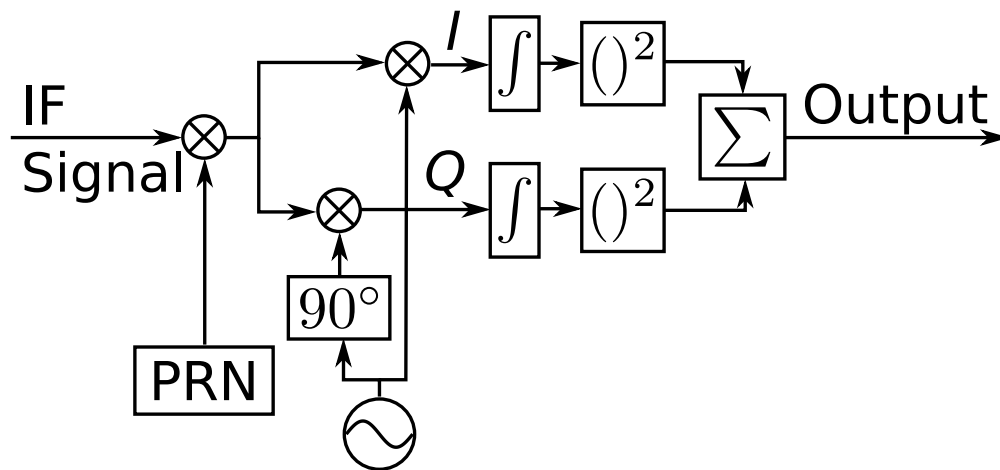


Figure 2.19: Serial acquisition diagram.

This process is repeated until a test statistic is generated for the full grid. The peak is then found and compared to a threshold to determine whether that signal is present. If it passes the threshold test, the code phase and carrier frequency that correspond with the peak are passed off to a channel to track the found PRN. As described in [4] one threshold test is to take the ratio of the peak and the next highest peak and ensure that the peak is at least 2.5 times the next highest. This second peak is restricted to be more than a chip width away from the peak to ensure two correlations are not partially aligned on the same chip.

Note that the correlation is circular since the period of the codes is known to be 1 ms. Thus any code phase offset in the data length is valid since the remaining code will wrap around to the beginning and correlate with the next or previous code period. However, this raises the issue of navigation data bit boundaries. If the next or previous code period is within a different navigation data bit there is a chance that the sign of the phase key will shift. Therefore the correlation would be over two sections of the code with different polarities. This change in phase spreads the power of the correlation out, cancelling the peak that would have been there. For this reason acquisition usually considers two consecutive signal segments of at most half the navigation data bit length. That way if a navigation data bit change occurred in one segment, it would be assured that the other segment would not contain a navigation data bit change.

2.4.2 Parallel Code Phase Acquisition

The longer a pair of sequences gets for correlation, the longer the time required to perform the multiply and accumulate. With long sequences it is hardly ever beneficial to perform correlation in this fashion. The circular cross-correlation sequence can be written as

$$z(n) = \frac{1}{N} \sum_{m=0}^{N-1} x(m)y(m+n) = \frac{1}{N} \sum_{m=0}^{N-1} x(-m)y(m-n) \quad (2.21)$$

where x and y are two sequences of length N . The scaling factor $\frac{1}{N}$ is ignored since the acquisition peaks are compared as a ratio anyway. The Fourier transforms of two sequences $x(n)$ and $y(n)$ are

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad (2.22)$$

$$Y(k) = \sum_{n=0}^{N-1} y(n)e^{-j2\pi kn/N} \quad (2.23)$$

Considering the Fourier transform of $z(n)$ it is seen that

$$\begin{aligned} Z(k) &= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(-m)y(m-n)e^{-j2\pi kn/N} \\ &= \sum_{m=0}^{N-1} x(m)e^{j2\pi km/N} \sum_{n=0}^{N-1} y(m+n)e^{-j2\pi k(m+n)/N} \\ &= X^*(k)Y(k) \end{aligned} \quad (2.24)$$

The parallel code phase search makes use of the rapid executing fast Fourier Transform (FFT) to search over all code phases in a single pass. This is much more efficient than generating the correlation at each possible delay. Therefore, the algorithm calculates the correlations in each frequency slice rather than each frequency-code phase combination. A diagram of this method is shown in Figure 2.20.

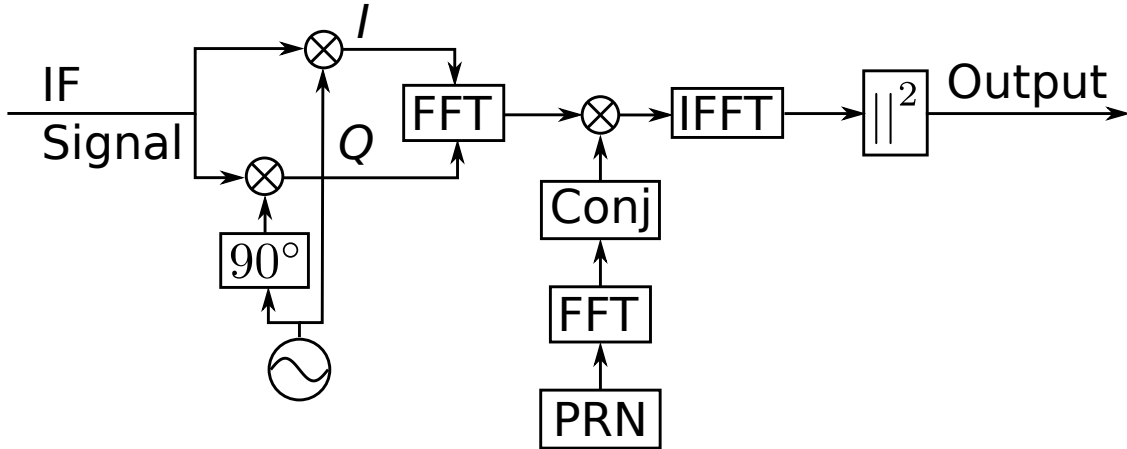


Figure 2.20: Parallel acquisition diagram.

Some additional optimization can be done by computing the complex conjugate of the FFT for the code sequence at the beginning of the search routine. This sequence does not change over

each of the frequency bin searches. The selection of the peak can use the same threshold detection as the sequential acquisition method described in Section 2.4.1.

2.4.3 Weak Signal Acquisition

Acquisition of weak signals can be problematic with the parallel code phase algorithm. Methods are introduced in [48] for searching for these difficult signals. Two key limiting factors in acquisition revolve around the navigation data bit transitions that possibly occur every 20ms. This boundary limits the amount of time to coherently correlate a local replica signal with the incoming signal. Since acquisition also starts out with no previous knowledge about where the navigation data bit transitions may take place, it is also important to take into account this possibility. If a navigation data bit transition occurs within an acquisition data segment, the changed sign will begin to cancel itself out in the correlation. Thus, a signal may be present in the data but cannot be recovered with this acquisition segment. Therefore a great performance increase can be found by acquiring over sequential segments of the incoming data.

A simple method is to use two data segments of the same length (10 ms or less) and acquire on both. The highest peak from the two sets is compared to the threshold since a data transition in one guarantees there will not be a transition in the other. For weak signals this method can be broken down further to make multiple combinations of the acquired data. Real signal energy gain comes from coherently integrating over longer periods. To deal with the navigation data bit transitions, 1 ms segments can be summed by multiplying by a ± 1 factor, effectively searching over possible navigation data bit transitions. With a 20 ms segment, the unsquared I and Q correlations for each possible location of the navigation data bit are accumulated and then summed and squared. This increases complexity in the acquisition stage but can sometimes be the only way to detect weak signals.

2.5 Hardware

While most receivers implement a significant portion of the work in hardware, this is not true for a software receiver. There are dedicated chips that implement the GPS correlators for rapid calculation of the correlator outputs (both for acquisition and tracking) as well as flexible

components for calculating all navigation aspects of the receiver. While the use of a software receiver reduces the development time, there are definite trade-offs in operating such a receiver. The correlation stage of the receiver is particularly computationally intensive although methods have been investigated to improve this performance [26], [27].

2.5.1 Front-End

A front end removes the necessity of sampling the GPS signal at the very high rate of the received signal (1.5GHz). This is done by down-conversion and sampling of the RF signal using a set of signal processing stages. The processes that the GPS front end take to generate these data bits are shown in Figure 2.21.

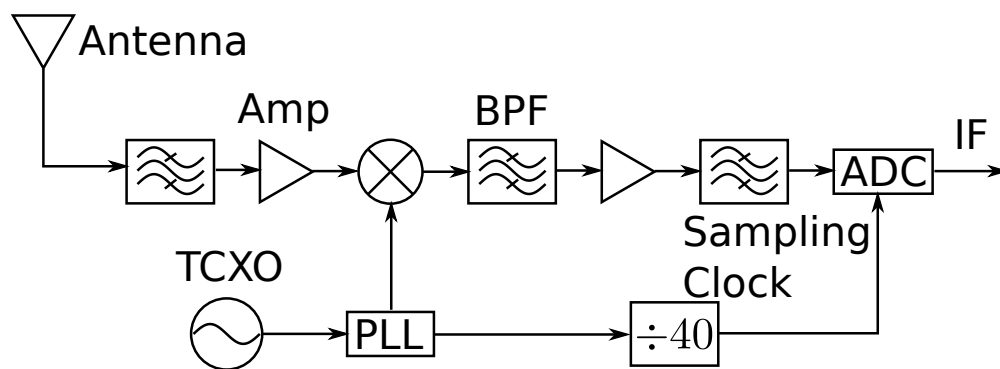


Figure 2.21: GPS front-end diagram.

In a typical front-end, the signal trail will begin with a single active or passive antenna. If the antenna is active, power is supplied to it and some other signal conditioning circuitry is usually included. Passive antennas are usually used when there is a very short distance between the antenna and the signal conditioning circuits, such as in hand held units. Several band pass filters (BPF) are used at various points to condition the signal and reduce interference and aliasing in later processing stages. Several amplifiers (Amp) are used to boost the signal power. An onboard oscillator (such as a TCXO in a low-cost receiver) drives the frequency mixing and sampling portions through a phase lock loop (PLL) and divider. Finally the analog to digital converter (ADC) generates the digital intermediate frequency (IF) bit stream for use in the software receiver.

Part of the processing chain shown in Figure 2.21 is the mixing of the incoming signal with a clock signal. This mixing changes the resulting signal to an intermediate frequency (IF) signal

that retains the Doppler shifts of the original signals. The Doppler shift is an alteration of the carrier frequency from the transmit frequency of the signal. For the signals under consideration, this is the L1 frequency. However, software receivers actually operate at a different frequency, the intermediate frequency, f_{IF} . This comes from the hardware front-end acting on the signal to reduce the high frequency requirements that would come from directly sampling the GPS signal. The Doppler frequency is the frequency deviation from the center frequency of the signal, and this is true for either f_{L1} or f_{IF} . This comes from the fact that the down converted signal is mixed with another frequency

$$f_{mix} = f_{L1} - f_{IF} \quad (2.25)$$

The down-conversion process multiplies a sinusoid at f_{mix} with the incoming signal at the L1 frequency offset by the Doppler frequency f_D as

$$\begin{aligned} & 2 \cos(2\pi f_{L1}t + 2\pi f_D t) \cos(2\pi f_{mix}t) \\ &= \cos(2\pi f_{L1}t - 2\pi f_{mix}t + 2\pi f_D t) + \cos((2\pi f_{L1}t + 2\pi f_{mix}t + 2\pi f_D t) \\ & \qquad \qquad \qquad \approx \cos(2\pi f_{IF}t + 2\pi f_D t) \end{aligned} \quad (2.26)$$

Since there are several low pass filters, the higher frequency terms are filtered and the receiver is left with a signal offset from a new center frequency by the same Doppler. This important operation maintains the signal structure while reducing the sampling requirements for the digital conversion.

2.5.2 Real and Complex Signals

Many different front-end designs have been developed for different applications. One main difference is the type of intermediate frequency signal that is provided to the software receiver. If the signal is mixed with only one oscillator phase, a single signal results. This signal can be considered a real signal when used in the receiver. Alternatively, the incoming signal can be mixed with an oscillator phase and a 90 degree delay in the phase. This yields two components of the same signal. In this case the software receiver can treat the signal as complex and the two phase mixed components are the real and imaginary parts.

2.6 Summary

This chapter provided an overview of GPS receiver operation. The processes were presented beginning with the navigation solution formation. Then a description was given as to how the receiver measurements are formed and how they relate to the received signal. Next the channel structure was detailed to describe how each separate GPS signal is tracked. Finally, some hardware considerations were presented to complete the view of a software GPS receiver implementation. While there are some differences among receivers, these operations remain fairly similar and provide a good background for understanding the rest of this dissertation.

Chapter 3

GPS Vector Tracking Receiver

An operating receiver is subjected to many sources of error. Two particular difficulties that must be handled by any receiver are system dynamics and noise. In a typical receiver such as the one described in Chapter 2, the ability to track through dynamics and noise requires a design trade-off. The loop filters can be tuned to track higher dynamics if they can accept more noise. However to reduce the tracking noise the filter is tuned to reduce the effects of each noisy sample. This is done by reducing the bandwidth of the channel loop filters. This bandwidth reduction, however, can keep the loop from being able to alter the local replica rapidly enough if the signal is changing too quickly. If the receiver had some other way to know how the signal is changing, the loop would be better able to minimize noise while maintaining lock on the signal. Vector tracking has been developed as a unique way to accomplish this using only the information available in a stand-alone GPS receiver.

3.1 Difficult Environments

Before the vector tracking algorithm is described, the environments themselves will be described to motivate the added complexity of a vector receiver. Originally GPS was designed to operate in open sky environments. Design decisions were made with this in mind since the feat of navigating in this way was a new leap in capabilities. The constellation of satellites was chosen to guarantee at least 4 signals available anywhere and at any time. Of course, this requires clear line of sight to the transmitting satellites. As a user moves near obstructions, this guarantee no longer holds since GPS signals are easily shadowed. Operation in areas with large buildings, called urban canyons, induces many blockages. With the proliferation of mobile navigation applications, there is a very high demand for positioning in such locations. There is a high density of users in downtown areas and recently location based services (LBS) have grown in popularity.

Another source of error in these environments comes from signal reflections, or multipath. When the signal is reflected and received along multiple paths, the receiver has more difficulty tracking the true line-of-sight signal. Nearby buildings block the line of sight vector and also create multipath reflections as shown in Figure 3.1. These reflections interfere with the tracking of the direct signals in similar ways. They induce a delay in the code signal as well as constructive or destructive interference from the carrier phase. More description of this influence on vector tracking is described in Chapter 4.

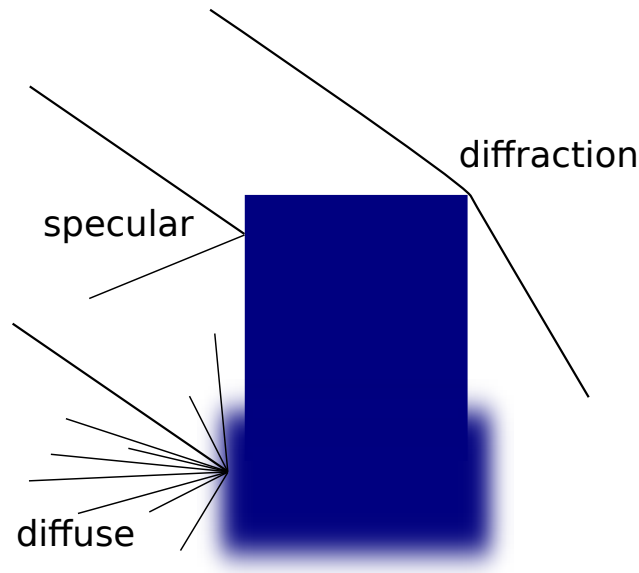


Figure 3.1: Multipath types typical in an urban canyon environment.

However, difficult environments are not only limited to passive error sources. As GPS becomes prolific, so too will those that want to break it. Several types of jamming have been described in the literature, including broadcasting from a simulator, utilizing a receiver-spoofers, and control of multiple phase-locked spoofers [27].

There is a large market for GPS jammers [10]. For example jammers are marketed for purposes such as blocking a vehicle tracker or hiding a fishing location. The weakness of the GPS signal makes them very susceptible to jamming. In mission critical applications such as precision approach, jamming can be a dangerous factor. There are also many cases where receivers must be designed to perform in jamming environments such as in military applications. Not all jamming is intentional, however. Any radio frequency (RF) noise broadcast around the GPS transmission frequencies acts

as noise to a receiver. A case in point would be the LightSquared issue in which a new system was being tested for the public market that unintentionally disrupted most civilian receivers [8].

3.2 Vector Tracking

As described in Chapter 2, a GPS receiver can be designed to operate with independent tracking channels. This receiver uses individual channels to separately monitor the satellite signals in an architecture called scalar tracking. Each channel operates independently of each other and no information is shared among them for tracking purposes. The main operation of a GPS receiver takes the following steps:

- Signals incident on the receiver antenna are passed through a front-end for filtering, down-conversion, and sampling.
- The intermediate frequency (IF) digital signal is then passed to a set of receiver channels which split the tracking to generate information based on a single satellite signal.
- Within a channel, the IF signal is mixed with the local replica signal for the channel's satellite.
- The individual samples are summed during an integration period.
- Discriminators take sets of integrated and dumped values from early, prompt, and late versions of the in-phase and quadrature correlators to generate tracking error measurements.
- The receiver's tracking mechanism uses these tracking errors to modify the generation of the local replica.
- Using the local replica parameters, ranges and range rates are generated to the tracked satellites.
- These measurements are used to generate the position, velocity, and time (PVT) outputs.

This flow of information is shown for a scalar receiver in Figure 3.2.

In this architecture, each channel is tracking the line of sight dynamics of the user antenna as well as the noise on the satellite signal. The code phase and carrier frequency relate directly to the user's position and velocity through the line of sight from the user to the satellite. The tracking is

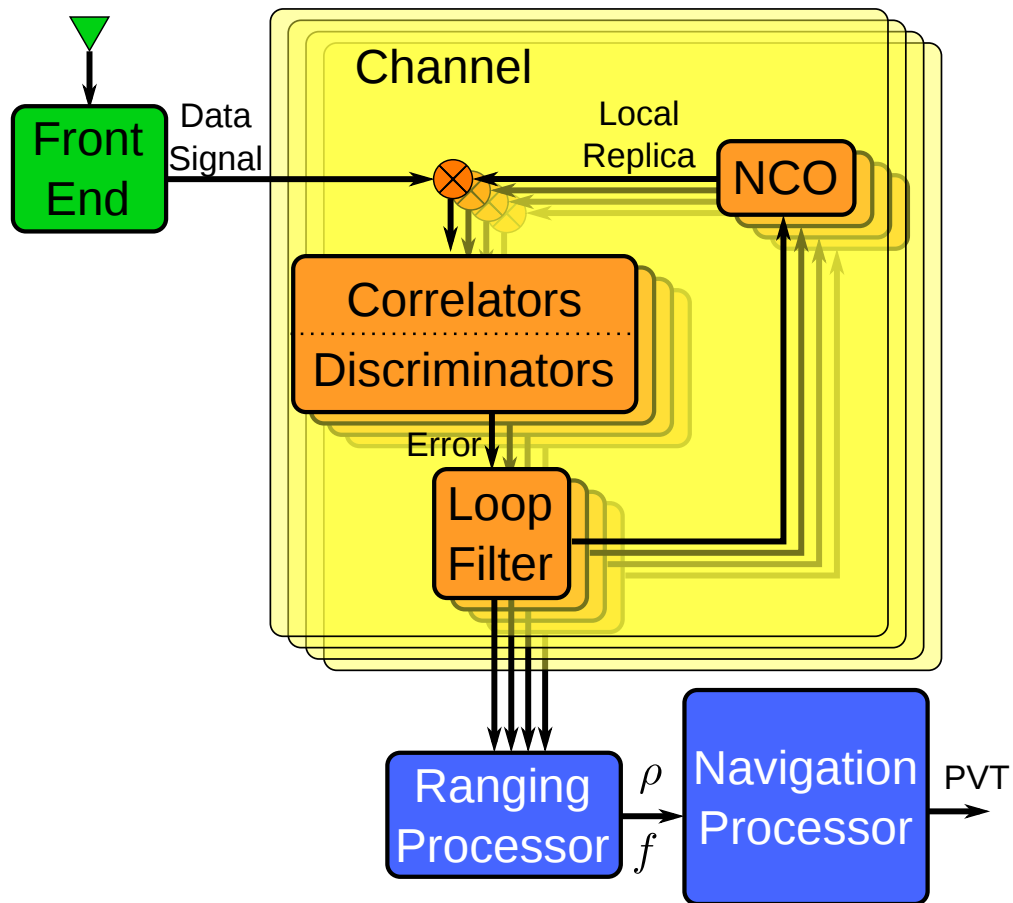


Figure 3.2: Diagram of a generic scalar tracking receiver.

accomplished by driving the code and carrier frequencies into lock with the incoming signal. When in lock, the error from the correlations are kept near zero.

However, it is also possible to combine the signal tracking and navigating stages of the receiver as shown in Figure 3.3. This architecture is known as a vector tracking receiver, or vector receiver. In a vector receiver, the tracked parameters are the position, velocity, and clock terms. These states are used to generate the tracking parameters for all channels. This is accomplished by translating positions and velocities into line of sight ranges and range rates. These line of sight parameters directly go to the channels to track. Tracking in a vector form thus shares the accuracy and power among the signals to improve the tracking of weak signals.

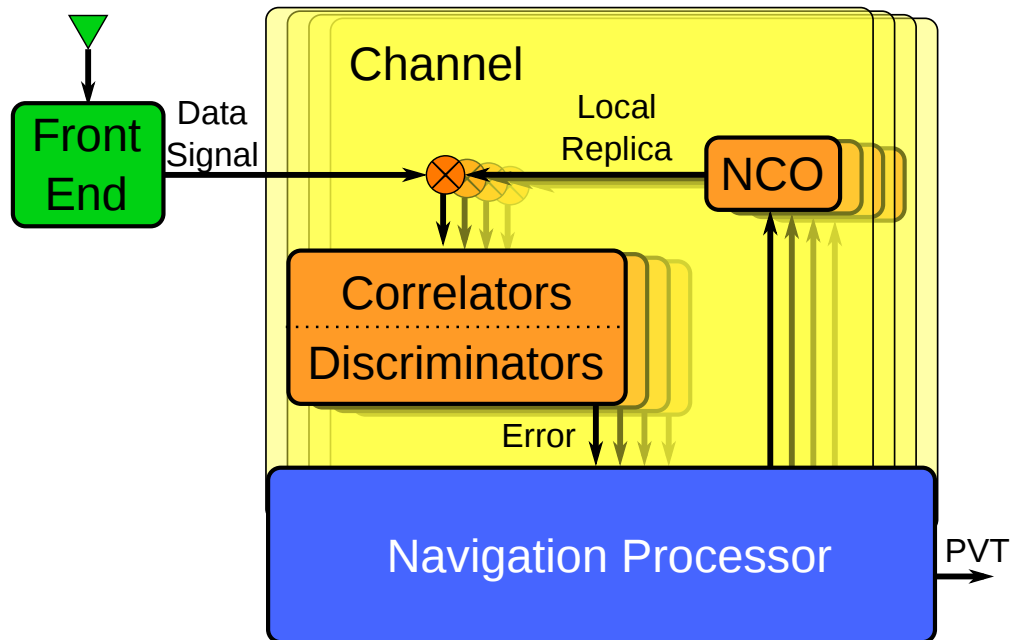


Figure 3.3: Diagram of a generic vector tracking receiver.

A demonstration of this sharing is shown in Figure 3.4. Here a slow turn is simulated and a single satellite signal (satellite 16) is dropped to a C/N_0 of 5 dB-Hz from 5 to 15 seconds. During this time no useful tracking information is generated by this channel. The scalar receiver begins tracking noise and is unable to continue to operate when the signal returns. However, the vector receiver is still estimating for this channel and the frequency continues to update. When the signal returns, the channel is still close to the actual signal so the measurements can continue. This is called the instantaneous reacquisition of signals in a vector receiver.

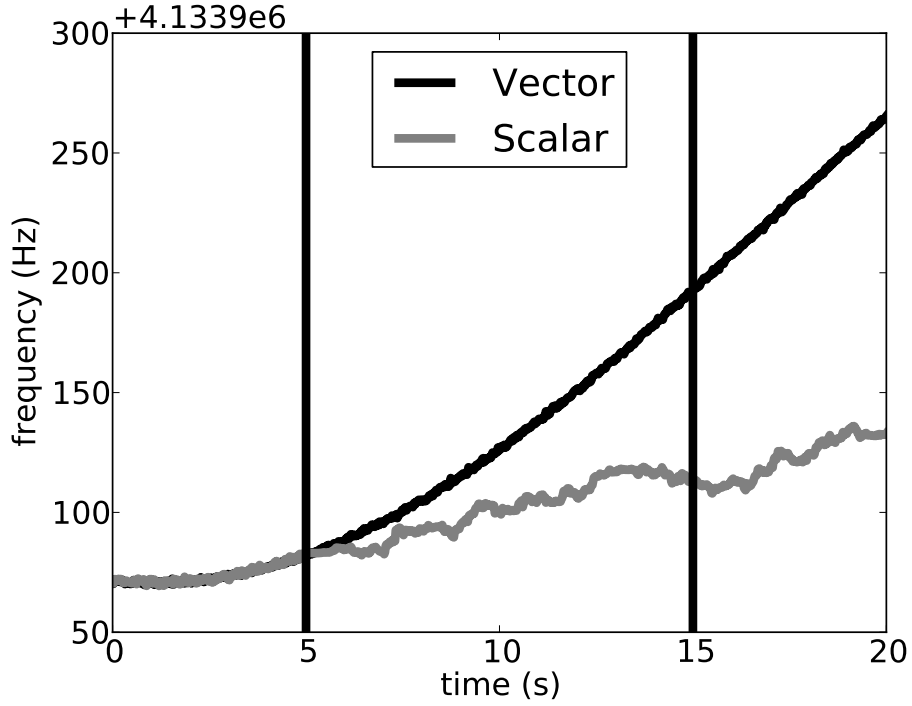


Figure 3.4: Estimated frequency for scalar and vector receiver during signal loss scenario.

3.2.1 GPS Vector Tracking Formulation

The vector formulation used in this work is the position state receiver. The state vector for the filter is

$$\bar{x} = \begin{bmatrix} \delta x \\ \delta \dot{x} \\ \delta y \\ \delta \dot{y} \\ \delta z \\ \delta \dot{z} \\ \delta cb \\ \delta \dot{cb} \end{bmatrix} \quad (3.1)$$

where x , y , z are the three components of the user earth-centered earth-fixed (ECEF) position in meters, cb represents the receiver clock bias in meters, $\dot{*}$ indicates derivative with respect to time,

and δ indicates these are error states. The state dynamics are therefore of the form

$$\dot{\hat{x}} = A\bar{x} + B_d w_d + B_c w_c \quad (3.2a)$$

$$A = \begin{bmatrix} \kappa & 0_{2 \times 2} & 0_{2 \times 2} & 0_{2 \times 2} \\ 1_{2 \times 2} & \kappa & 0_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & \kappa & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & 0_{2 \times 2} & \kappa \end{bmatrix} \quad (3.2b)$$

$$\kappa = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (3.2c)$$

$$B_d = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.2d)$$

$$w_d = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (3.2e)$$

$$B_c = \begin{bmatrix} 0_{2 \times 2} \\ 0_{2 \times 2} \\ 0_{2 \times 2} \\ I_{2 \times 2} \end{bmatrix} \quad (3.2f)$$

$$w_c = \begin{bmatrix} w_b \\ w_r \end{bmatrix} \quad (3.2g)$$

where $0_{2 \times 2}$ is the matrix of zeros, I is the identity matrix, w_x , w_y , w_z are the modeled process noises driving the x , y , z velocity states, respectively. Also w_b corresponds to the clock bias noise

and w_r corresponds to the clock rate noise. The statistics for w_d are

$$\begin{aligned}
 E[w_d] &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
 E[w_d w_d^T] &= \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}
 \end{aligned} \tag{3.3}$$

where σ_x^2 is the variance of the state x . Depending on expected receiver dynamics, the values of σ_x^2 , σ_y^2 , and σ_z^2 can be tuned. The statistics for w_c are

$$\begin{aligned}
 E[w_c] &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 E[w_c w_c^T] &= \begin{bmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}
 \end{aligned} \tag{3.4}$$

The values of σ_b^2 and σ_r^2 are based on rule of thumb analysis. In this work these values are $c^2 \times 10^{-19}$ m and $4\pi c^2 \times 10^{-20}$ m/s, respectively from [7]. When the system in Equation (3.2) is discretized

with Euler discretization, it becomes

$$\bar{x}_{k+1} = A_p \bar{x}_k + Q_p \quad (3.5a)$$

$$A_p = \begin{bmatrix} \alpha & 0_{2 \times 2} & 0_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & \alpha & 0_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & \alpha & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & 0_{2 \times 2} & \alpha \end{bmatrix} \quad (3.5b)$$

$$\alpha = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (3.5c)$$

$$Q_p = \begin{bmatrix} Q_x & 0_{2 \times 2} & 0_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & Q_y & 0_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & Q_z & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & 0_{2 \times 2} & Q_c \end{bmatrix} \quad (3.5d)$$

$$Q_x = \begin{bmatrix} \sigma_x^2 \frac{T^3}{3} & \sigma_x^2 \frac{T^2}{2} \\ \sigma_x^2 \frac{T^2}{2} & \sigma_x^2 T \end{bmatrix} \quad (3.5e)$$

$$Q_y = \begin{bmatrix} \sigma_y^2 \frac{T^3}{3} & \sigma_y^2 \frac{T^2}{2} \\ \sigma_y^2 \frac{T^2}{2} & \sigma_y^2 T \end{bmatrix} \quad (3.5f)$$

$$Q_z = \begin{bmatrix} \sigma_z^2 \frac{T^3}{3} & \sigma_z^2 \frac{T^2}{2} \\ \sigma_z^2 \frac{T^2}{2} & \sigma_z^2 T \end{bmatrix} \quad (3.5g)$$

$$Q_c = \begin{bmatrix} \sigma_b^2 T + \sigma_r^2 \frac{T^3}{3} & \sigma_r^2 \frac{T^2}{2} \\ \sigma_r^2 \frac{T^2}{2} & \sigma_r^2 T \end{bmatrix} \quad (3.5h)$$

The measurement update uses a vector of the channel residuals. These measurements are related to the states by

$$\bar{y} = C\bar{x} + \bar{v}$$

$$C = \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 \end{bmatrix} \quad (3.6)$$

In this case $a_{i,j}$ describes the i th component of the unit vector to the j th satellite. The statistics for the measurement noise vector \bar{v} are assumed to be

$$E[\bar{v}] = 0_{Nx1}$$

$$E[\bar{v}\bar{v}^T] = R_v \quad (3.7)$$

The values for R_v are generated using relations between the estimated channel C/N_0 and the receiver thermal noise.

3.2.2 Equivalent Scalar Tracking Formulation

In order to compare the performance of a scalar receiver to a vector receiver an equivalent scalar filter is required. This can be accomplished by reformulating the vector formulation into pseudorange states. The state vector for the filter is

$$\bar{x} = \begin{bmatrix} \delta\rho_1 \\ \delta\dot{\rho}_1 \\ \vdots \\ \delta\rho_N \\ \delta\dot{\rho}_N \end{bmatrix} \quad (3.8)$$

Where ρ_i indicates pseudorange for satellite i where $1 \leq i \leq N$ when tracking N satellites.

$$\dot{\hat{x}} = A\bar{x} + B_d w_d + B_c w_c \quad (3.9a)$$

$$A = \begin{bmatrix} \kappa & 0_{2 \times 2} & \dots & 0_{2 \times 2} \\ 0_{2 \times 2} & \kappa & & \vdots \\ \vdots & & \ddots & \vdots \\ 0_{2 \times 2} & \dots & \dots & \kappa \end{bmatrix}_{2N \times 2N} \quad (3.9b)$$

$$\kappa = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (3.9c)$$

$$B_d = \begin{bmatrix} 0 & 0 & 0 \\ a_{x,1} & a_{y,1} & a_{z,1} \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ a_{x,N} & a_{y,N} & a_{z,N} \end{bmatrix} \quad (3.9d)$$

$$w_d = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (3.9e)$$

$$B_c = \begin{bmatrix} I_{2 \times 2} \\ \vdots \\ I_{2 \times 2} \end{bmatrix} \quad (3.9f)$$

$$w_c = \begin{bmatrix} w_b \\ w_r \end{bmatrix} \quad (3.9g)$$

When the system in Equation (3.9) is discretized, it becomes

$$\bar{x}_{k+1} = A_\rho \bar{x}_k + q_\rho \quad (3.10a)$$

$$A_\rho = \begin{bmatrix} \alpha & 0_{2 \times 2} & \dots & 0_{2 \times 2} \\ 0_{2 \times 2} & \alpha & & \vdots \\ \vdots & & \ddots & \vdots \\ 0_{2 \times 2} & \dots & \dots & \alpha \end{bmatrix}_{2N \times 2N} \quad (3.10b)$$

$$\alpha = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (3.10c)$$

$$Q_\rho = \begin{bmatrix} Q_{1,1} + Q_c & Q_{1,2} + Q_c & \dots & Q_{1,N} + Q_c \\ Q_{2,1} + Q_c & Q_{2,2} + Q_c & & \vdots \\ \vdots & & \ddots & \vdots \\ Q_{N,1} + Q_c & \dots & \dots & Q_{N,N} + Q_c \end{bmatrix} \quad (3.10d)$$

$$Q_{i,j} = \begin{bmatrix} \beta_{i,j} \frac{T^3}{3} & \beta_{i,j} \frac{T^2}{2} \\ \beta_{i,j} \frac{T^2}{2} & \beta_{i,j} T \end{bmatrix} \quad (3.10e)$$

$$\beta_{i,j} = \left[\sigma_x^2 a_{x,i} a_{x,j} + \sigma_y^2 a_{y,i} a_{y,j} + \sigma_z^2 a_{z,i} a_{z,j} \right] \quad (3.10f)$$

This formulation significantly simplifies the relation between the states and measurements as

$$y = I_{2N \times 2N} \bar{x} + \bar{v} \quad (3.11)$$

where I is the identity matrix, which is $2N$ by $2N$. The measurement Equations (3.6) and (3.11) use the same noise values.

This receiver suffers from poor numerical properties when in this form. An example of this is shown below in Figure 3.5. However, if off-diagonal terms of the process noise in the discrete formulation are zeroed, the filter becomes a stack of channel filters. No information is shared between channels and this receiver then acts like a scalar receiver. The numerical properties of this form are described in more detail in Appendix C.

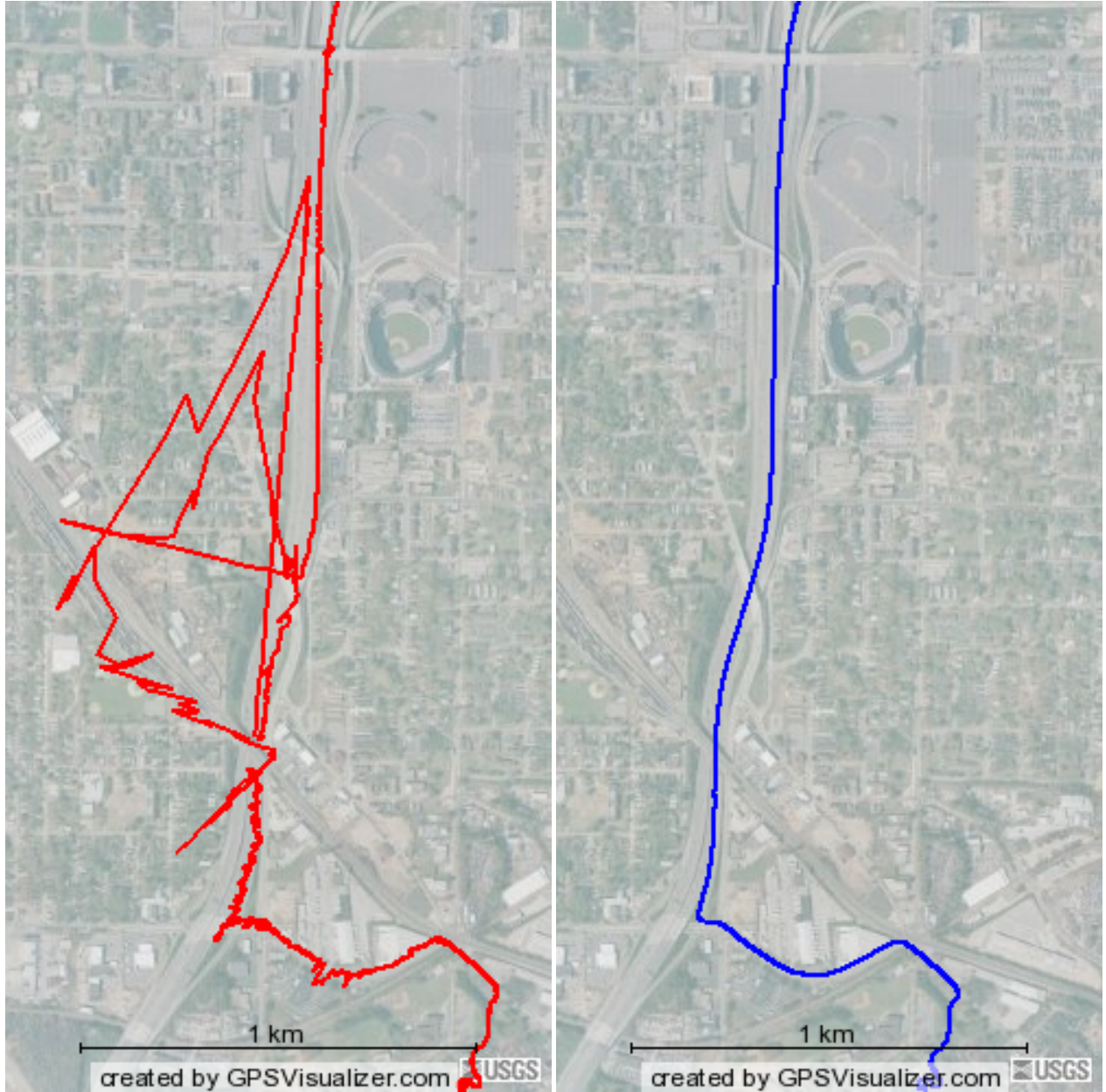


Figure 3.5: Poor numerical performance of the pseudorange state receiver (left) shown compared to position state vector receiver (right).

An alternative way to compare vector and scalar tracking is described in [2]. This is done by recasting the vector receiver to a discrete parametric model, from which transfer functions can be derived. While it is beneficial in understanding the two receiver architectures, the method presented in this dissertation in Section 3.2.2 allows for a more straightforward implementation.

3.2.3 Measurement Generation

Since the system measurement Equations (3.6) and (3.11) relate the error states to the pseudorange and pseudorange rate errors, the generation of these measurements need to be described. At the beginning of an integrate and dump period for the receiver channel, the estimated values of the code phase and carrier frequency are used to generate the channel's local replica. Over the integrate and dump period (20ms in this work) the incoming measured signal is combined with the local replica copies (including in-phase and quad-phase versions of the Early, Prompt, and Late replicas) and summed. These summations, shown below, are used by the receiver discriminators to generate pseudorange and pseudorange rate measurements.

$$I(k, \tau) = AR(\epsilon + \tau)D(k) \cos(\pi fT + \theta) + \eta_I(k) \quad (3.12a)$$

$$Q(k, \tau) = AR(\epsilon + \tau)D(k) \sin(\pi fT + \theta) + \eta_Q(k) \quad (3.12b)$$

Here I is the in-phase replica, Q is the quad-phase replica, A is the signal amplitude, R is the correlation function, ϵ is the range error, τ is the intentional correlator offset ($\pm \frac{\delta}{2}$ for the early and prompt replicas), f is the frequency error, T is the correlation time, and η is the additive noise. Assuming infinite bandwidth, the correlation function is taken to be

$$R(\epsilon) = \begin{cases} 1 - \frac{|\epsilon|}{\delta} & -\delta \leq \epsilon \leq \delta \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

Where δ represents the chip width.

The pseudorange error measurement is generated using the early minus late power discriminator, which in the absence of additive multipath signals is

$$Y_R = IE^2 + QE^2 - IL^2 - QL^2 \quad (3.14a)$$

$$= A^2 \frac{2\epsilon}{\delta} + \eta \quad (3.14b)$$

where

$$IE = I \left(k, \frac{\delta}{2} \right) \quad (3.15a)$$

$$QE = Q \left(k, \frac{\delta}{2} \right) \quad (3.15b)$$

$$IL = I \left(k, -\frac{\delta}{2} \right) \quad (3.15c)$$

$$QL = Q \left(k, -\frac{\delta}{2} \right) \quad (3.15d)$$

While the pseudorange error measurement is generated over a single 20 ms integrate and dump period, the pseudorange rate error measurement uses sequential 10 ms periods. This is done to compare the change in phase over the 20 ms period by correlating the two 10 ms periods separately.

$$Y_{cross} = IP_1 * QP_2 - IP_2 * QP_1 \quad (3.16a)$$

$$= A^2 R(\epsilon_1) R(\epsilon_2) \sin(\theta_2 - \theta_1) \quad (3.16b)$$

$$Y_{dot} = IP_1 * IP_2 + QP_1 * QP_2 \quad (3.16c)$$

$$= A^2 R(\epsilon_1) R(\epsilon_2) \cos(\theta_2 - \theta_1) \quad (3.16d)$$

$$Y_{RR} = \tan^{-1} \left(\frac{Y_{cross}}{Y_{dot}} \right) \quad (3.16e)$$

$$= \frac{\theta_2 - \theta_1}{T} + \nu \quad (3.16f)$$

where

$$IP_1 = I(k, 0) \quad (3.17a)$$

$$QP_1 = Q(k, 0) \quad (3.17b)$$

$$IP_2 = I(k + 10ms, 0) \quad (3.17c)$$

$$QP_2 = Q(k + 10ms, 0) \quad (3.17d)$$

Note that the correlators are not synchronized to begin and end at the same time since the start of the code period depends on the range, which is different for each satellite. Each channel tracks the signal based on the same code period in GPS time. This is one difficulty in using a centralized vector filter [36].

3.3 Signal Strength

As part of a receiver's operation, other processes besides generating tracking errors are needed to provide reliable positioning. One of these operations is signal strength estimation. Signal strength plays a large roll in notifying the receiver of possible failure conditions. In the vector measurement formulation it is also needed since the range measurements are a function of the signal amplitude in Equation (3.14).

3.3.1 Amplitude Estimation

As part of calculating the measurement in Equation (3.14), an estimate of the signal amplitude is required. This estimation is done in two steps. First, a running average filter is used to keep track of the signal noise variance. The new measurement \tilde{v} to the filter comes by taking the variance of noise correlators generated by known erroneous code phase offsets among all channels. Outside the chip width in Equation (3.13), the correlation function is essentially zero. Therefore the correlator values from Equation (3.12) are only noise. Sequential filtering of these values gives a good running estimate of noise variance. The running filter is

$$\hat{\eta}^2 = \alpha_\eta \hat{\eta}^2 + (1 - \alpha_\eta) \tilde{\eta}^2 \quad (3.18)$$

where α is the filtering value. Second, the raw amplitude estimate is made using the in-phase and quad-phase correlators described in Equation (3.15) as

$$\tilde{A}^2 = (IE + IL)^2 + (QE + QL)^2 \quad (3.19a)$$

$$= A^2 + 4\eta^2 \quad (3.19b)$$

This estimate is also filtered so small variations do not drastically change the vector receiver operation. For a further discussion of this amplitude estimation, refer to [33].

Since these amplitude estimates are used in the generation of the measurement Y_R in Equation (3.14), the filtering performance will have an effect on the navigation performance. This is taken as acceptable since the expected amplitude changes slowly as the range from user to satellite has relatively low dynamics. The majority of the amplitude change comes from local environment effects such as signal blockage and shadowing. The vector receiver performance will be affected by the accuracy of the amplitude measurement. However, the full impact and importance of this effect has not yet been consistently demonstrated. In [17] this effect is said to be significant but in [33] it is demonstrated as negligible.

3.3.2 Carrier to Noise Ratio

In a receiver, the estimation of the signal power relative to noise is key in several aspects. The carrier to noise ratio, C/N_0 , is used in measurement variance calculations for the navigation filter. Since the measurement noise changes with the signal power, this information must be included in the filter in either the vector or scalar case. The threshold can also be used to decide a loss of lock situation can be arrived at analytically or empirically [19]. In the case of loss of lock, a scalar receiver would begin the process of acquisition again, as described in Section 2.4. In a vector receiver, the solution is not so simple and will be discussed later in Chapter 5.

Signal Power Estimation

The first part of the C/N_0 is the signal power. When a signal is being properly tracked, as described in Section 2.3, the signal power should be completely in the in-phase prompt correlator. However, this ideal is not realizable as there will always be error in the local replica. Therefore it

is more reliable to take into account motion errors. Since the power is distributed in the I and Q correlators due to phase (and frequency) errors, both correlators should be used. Also, since the peak of the correlation peak is reached in the prompt signal only when the code phase is perfectly known, the early and late signals are instead used [17]. Therefore, a more general idea of the signal power is estimated as

$$C = (IE + IL)^2 + (QE + QL)^2 \quad (3.20)$$

Noise Estimation

The second portion of the C/N_0 is the noise density. As the noise is generally a quantity that affects all receiver channels, it is estimated as a combination of all channels using the correlation property of the C/A codes. When a signal is being properly tracked, the correlators operate near the correlation peak. If a receiver correlates with the incoming signal at a distance that puts it beyond the correlation peak, the correlation is approximately zero. Therefore, using the same data structures to perform correlations at a large offset from the tracked signal allows the receiver to get an estimate of the signal noise. Taking the variance of these intentionally erroneous offsets across channels gives the receiver an estimate of the variance on the receiver noise [17].

Power and Noise Filtering

The signal power and noise variance estimates change very rapidly even in benign environments. To remedy this, a moving-average filter is applied to these quantities, as in [17]. This is similar to the filtering employed in Equation (3.18) for the signal amplitude. With a quantity measurement \tilde{x} , the average measurement \bar{x} is

$$\bar{x}_{k+1} = \alpha \bar{x}_k + (1 - \alpha) \tilde{x} \quad (3.21)$$

where α is the window value. In this receiver, a value of $\alpha = 0.9$ was used throughout. Although this value is hand-tuned it is consistent with the long averaging times (several seconds) mentioned in similar methods [23].

Carrier to Noise Estimation

The carrier to noise ratio uses both the estimated amplitude and the noise estimate. These are combined as

$$C/N_0 = 10 \log_{10} \left(\frac{C - 4\nu^2}{\nu^2} \right) \quad (3.22)$$

The removal of the noise, ν^2 , in Equation (3.22) is necessary since the amplitude estimate (C) includes both the true signal and the additive noise [33]. Since the signal power is found in Equation (3.20) to be the squared sum of four measurements affected by the noise, it must also be scaled by four when generating the true signal power estimate. The estimated C/N_0 can then be used to calculate the thermal noise jitter, used later in fault detection.

Measurement Covariance From Signal Strength

Using the estimated amplitude, C , and noise variance, ν^2 , the measurement covariance for a single satellite is described here. These relations are taken from [33] and [17]. The variance of the range measurement is calculated as

$$E \{v_R^2\} = \frac{8\delta\nu^4 + 4C^2\delta\nu^2 f(\rho_e)}{2C^2} \quad (3.23)$$

$$f(\rho_e) = 2\frac{\rho_e^2}{\delta^2} + \frac{1}{2}$$

where ρ_e is the range error. Similarly the variance of the range rate measurement is calculated as

$$E \{v_{RR}^2\} = \frac{\nu^4}{2C^2} + \frac{1}{4}R^2(\rho_e)\nu^2 \quad (3.24)$$

This equation assumes that the range error is less than half a chip. Otherwise, the correlation function $R(\rho_e)$ would be zero.

3.4 Summary

This chapter concludes the background material with the details of a vector receiver formation. The differences between scalar and vector tracking were given along with some of the motivation for this architecture. An equivalent scalar receiver was also described to facilitate performance

comparison between scalar and vector receivers. The chapter ends with methods for measurement generation and signal amplitude estimation. These scalar and vector receivers will be used in the remaining portion of the dissertation and extended with some novel analysis and enhancements.

Chapter 4

GPS Receiver Operation in Multipath

From the design standpoint, a GPS receiver is assumed to have clear line of sight to the satellites it is tracking. While this assumption holds well in open fields with few obstructions, this is an incredibly limited restriction. In fact, with the increased use of GPS receivers, scenarios presented to the receiver are increasingly filled with line of sight blocking obstacles. These scenarios include heavy foliage areas, urban canyons, and even indoors. In these cases, blocked and reflected signals are the norm rather than the exception. A received signal that has been reflected at least once before being incident on the receiver antenna is referred to as multipath. There are always reflections incident on the receiver's antenna. Any reflected signal will be a delayed copy of the direct signal since the direct path is the shortest path from the satellite to the antenna. While the direct line of sight signal represents the most accurate measure for transit time, a single antenna cannot easily distinguish all of the copies of the signal that have been reflected before they are received. The presence of multiple signals in the same band first increases the noise level on all receiver channels. However, the cross correlation attributes of the GPS codes reduces this effect on other channels. When a received signal is being tracked along with one or more delayed copies, the channel will have errors induced due to the blending of the incoming signals. While there are several types of reflection (specular, diffuse, diffraction [38]), their effect at the antenna is the same. Coverage of multipath and its effects is extensive [5, 44, 6, 28, 36]. Many mitigation strategies have been developed including antenna and algorithm modifications. However, the use of a vector receiver for multipath has had limited work published to date. This chapter investigates a vector receiver's ability to operate in a multipath environment.

A complete assessment of a receiver technique includes performance in environments that are typically difficult for normal operation. Research has been conducted to show the ability of a vector receiver to operate in low signal power conditions. However, there has not been an investigation of

the formulation in typical error-inducing scenarios such as multipath. While changing to a vector implementation to cope with multipath is an extreme response, advantages should be considered.

This chapter will show that vector tracking can improve performance in multipath heavy scenarios. Analytical investigation will show why this is the case by describing how vector formulations better maintain tracking after multipath error has been injected. This will be demonstrated in both simulation and experimentally. Following this analysis, conclusions will be drawn from this work.

4.1 Theoretical Performance

It is proposed that improved performance can be shown by monitoring a receiver's ability to track the incoming signal with minimal error. In this way performance is indicated by the error induced in the tracking of a single channel. A channel that is in error will have its local replica shifted from the true incoming signal. Any noise on the signal will induce some difference between the replica and received signals. However, looking at this error in a controlled situation (such as simulation) shows how the generation of the replica signal performs. This local replica represents the best guess of the received signal and is therefore an indication of a channel's tracking performance.

Analyzing this performance is done by taking a given measurement input and generating corrections dependent on the receiver formulation. Then this newly updated solution is propagated through the receiver model to the next update epoch. The difference between the predicted and actual signals indicates tracking performance. Ideally this difference is minimized from epoch to epoch by the tracking loop mechanism. However, comparisons of these mechanisms will indicate which formulation performs better.

4.1.1 Multipath Modification of Receiver Measurements

When a receiver enters a multipath environment, the correlation plane becomes corrupted by the multiple delayed copies of the satellite signal. This interference will change the measured correlation peak and any change in this peak will drive the receiver with erroneous measurements. These errors do not fit the typical additive noise model used in receiver derivations; the delay error is not an additive bias on the pseudorange. Since the incoming signal is a delayed copy of the true

line of sight signal, its effect is that of a delayed but weaker additive signal that is not negligible due to cross correlation properties. Assuming the antenna is receiving both the direct line of sight signal and a single multipath signal, this can be modeled as an effect on both the in-phase and quad-phase signal components as

$$I(k, \tau) = AR(\epsilon + \tau)D(k) \cos(\pi fT + \theta) + A_m R(\epsilon_m + \tau)D(k) \cos(\pi f_m T + \theta_m) + \eta_I(k) \quad (4.1a)$$

$$Q(k, \tau) = AR(\epsilon + \tau)D(k) \sin(\pi fT + \theta) + A_m R(\epsilon_m + \tau)D(k) \sin(\pi f_m T + \theta_m) + \eta_Q(k) \quad (4.1b)$$

where A_m , ϵ_m , f_m and θ_m indicate the amplitude, delay, frequency error, and phase error of the multipath, respectively. Assuming infinite bandwidth, the correlation function is taken to be

$$R(\epsilon) = \begin{cases} 1 - \frac{|\epsilon|}{\delta} & -\delta \leq \epsilon \leq \delta \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

where δ represents the chip width. Notice that Equation (4.1) contains effects based on the replica delay error ϵ and the multipath delay ϵ_m . It is also noted that the amplitude of the multipath component is

$$A_m = \sqrt{2T \frac{C}{N_0}} \text{sinc}(\pi f_m T) \quad (4.3)$$

where $\frac{C}{N_0}$ is the carrier to noise ratio. Since multipath sources tend to be spatially diverse, this frequency error can often drive the effective amplitude of the multipath signal much lower. Therefore most multipath has too large of a frequency error to affect the channel's tracking ability. This is why, in [41], the multipath reflections are searched for with large frequency differences, almost as if they are separate signals in an acquisition scheme. In the following analysis this multipath amplitude is ignored even though it is a function of the frequency error. This is done to simplify the analysis since the main objective is to look at ranging error due to multipath.

Evaluating Equation (3.14) including an additive multipath signal described in Equation (4.1) yields

$$\begin{aligned}
Y_R &= A^2 \left[R^2 \left(\epsilon - \frac{\delta}{2} \right) - R^2 \left(\epsilon + \frac{\delta}{2} \right) \right] \\
&\quad + A_m^2 \left[R^2 \left(\epsilon_m - \frac{\delta}{2} \right) - R^2 \left(\epsilon_m + \frac{\delta}{2} \right) \right] \\
&\quad + 2AA_m \cos(\Delta) R_{cross} + \eta \\
\Delta &= \pi T(f - f_m) + \theta - \theta_m \\
R_{cross} &= R \left(\epsilon - \frac{\delta}{2} \right) R \left(\epsilon_m - \frac{\delta}{2} \right) \\
&\quad - R \left(\epsilon + \frac{\delta}{2} \right) R \left(\epsilon_m + \frac{\delta}{2} \right)
\end{aligned} \tag{4.4}$$

where δ is the chip width. If both ϵ and ϵ_m satisfy $|\epsilon| \leq \frac{\delta}{2}$ then Equation (4.4) reduces to

$$\begin{aligned}
Y_R &= A^2 \frac{2\epsilon}{\delta} + A_m^2 \frac{2\epsilon_m}{\delta} \\
&\quad + 2AA_m \cos(\Delta) \frac{\epsilon + \epsilon_m}{\delta} + \eta \\
\Delta &= \pi T(f - f_m) + \theta - \theta_m
\end{aligned} \tag{4.5}$$

By including the multipath signal, a bias has been induced in the range measurement. By removing the true correlation peak and the noise from Equation (4.4) and assuming perfect tracking of the true signal before the multipath appearance, the bias effect of the multipath can be seen as

$$Y_m = \begin{cases} A_m^2 \frac{2\epsilon_m}{\delta} & 0 \leq \epsilon_m \leq \frac{\delta}{2} \\ +AA_m \frac{2\epsilon_m}{\delta} \cos(\pi T(f_m) + \theta_m) & \\ \\ A_m^2 \left(\frac{3}{2} - \frac{\epsilon_m}{\delta} \right)^2 + & \frac{\delta}{2} \leq \epsilon_m \leq \frac{3\delta}{2} \\ +AA_m \left(\frac{3}{2} - \frac{\epsilon_m}{\delta} \right) \cos(\pi T(f_m) + \theta_m) & \end{cases} \tag{4.6}$$

This bias is dependent on several things. The multipath amplitude relative to the direct amplitude (called the multipath-to-direct ratio, MDR) determines the magnitude of the bias. Also, the multipath delay relative to the direct signal determines the overlap between the correlation peaks as shown in Figure 4.1.

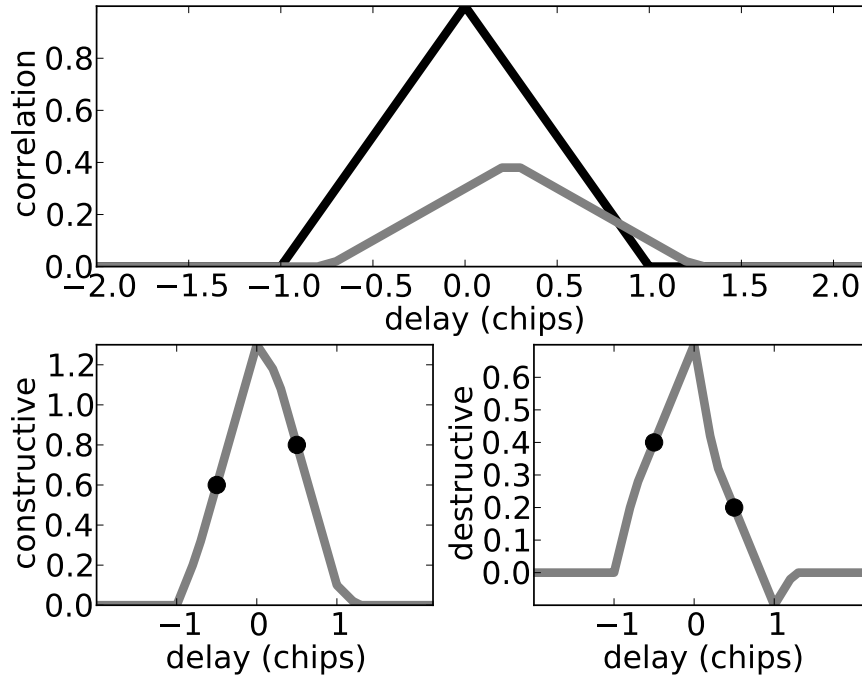


Figure 4.1: Multipath and direct signal correlation peaks (top) and resulting correlation peak with constructive (left) and destructive (right) interference

Finally, the phase error generated by the difference between the direct and multipath frequencies and phases also determine the bias in the measurement. With the same amplitude and delay errors, the multipath can generate either constructive or destructive interference, depending on this total phase error. This is a result of the cosine term in Equation (4.6). Typically $A > A_m$ and so $AA_m > A_m^2$, which means the cosine term can have a larger effect on the bias from multipath. When the phase error is maximum, the resulting bias is larger. But when the phase error is minimum, the bias will actually appear negative from the multipath free measurement. This is illustrated in the bottom subplots of Figure 4.1. If the interference is constructive (left) its effect on the early and late correlators is the opposite from the destructive (right) case. Thus, for a given MDR and relative delay, the actual error lies in the bounds shown in Figure 4.2. The actual error is also dependent on the phase error which can vary quickly.

Another significant source of error from the multipath occurs in the combined phase of the correlation outputs. Measurement formulation is complicated by the introduction of an additional signal in Equation (4.1) which is not zeroed by the correlation. When the multipath inputs are

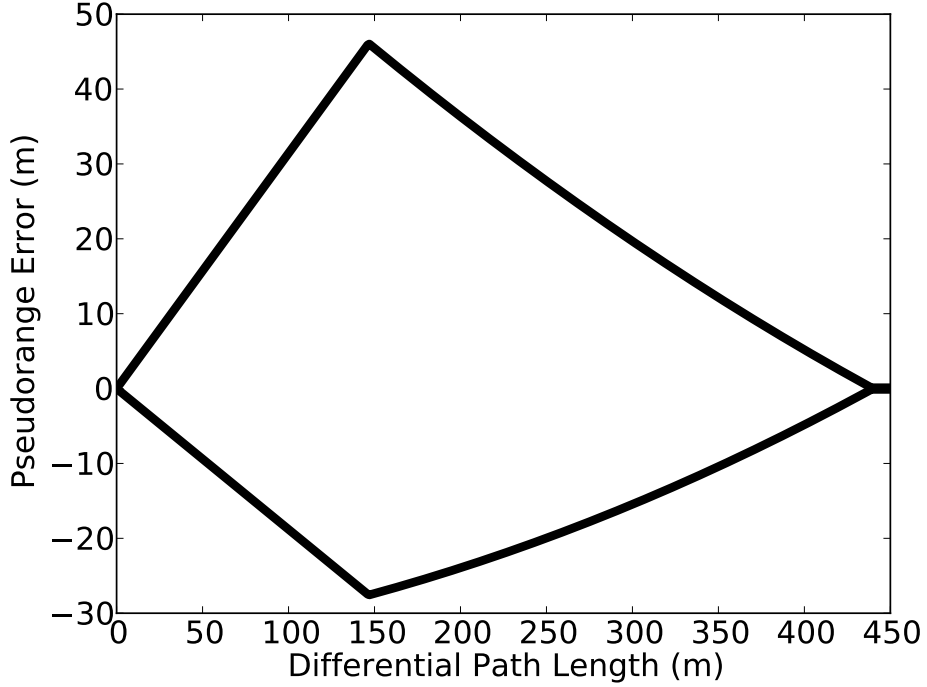


Figure 4.2: Multipath range error bounds for possible delays

operated on in the pseudorange rate discriminator in Equation (3.16), the results appear as shown below.

$$\begin{aligned}
Y_{cross} &= A^2 R^2(\epsilon) \sin(\theta_2 - \theta_1) + A_m^2 R^2(\epsilon_m) \sin(\theta_{m2} - \theta_{m1}) \\
&\quad + AA_m R(\epsilon) R(\epsilon_m) S_1
\end{aligned} \tag{4.7a}$$

$$\begin{aligned}
S_1 &= \sin(\pi T(f - f_m) + \theta_1 - \theta_{m2}) \\
&\quad + \sin(\pi T(f - f_m) + \theta_2 - \theta_{m1})
\end{aligned}$$

$$\begin{aligned}
Y_{dot} &= A^2 R^2(\epsilon) \cos(\theta_2 - \theta_1) + A_m^2 R^2(\epsilon_m) \cos(\theta_{m2} - \theta_{m1}) \\
&\quad + AA_m R(\epsilon) R(\epsilon_m) S_2
\end{aligned} \tag{4.7b}$$

$$\begin{aligned}
S_2 &= \cos(\pi T(f - f_m) + \theta_1 - \theta_{m2}) \\
&\quad + \cos(\pi T(f - f_m) + \theta_2 - \theta_{m1})
\end{aligned}$$

Recall from Equation (3.16f) that the measurement is then generated as the two quadrant arctangent of Y_{cross} and Y_{dot} . Assuming perfect frequency tracking before a multipath error is included,

the induced range rate error is shown in Figure 4.3 for an MDR of 0.1. The main contribution of multipath to a range rate error comes from multipath at a different frequency from the direct line of sight signal. The instantaneous phase of the multipath signal changes how this frequency error induces a range rate error. Traditionally, the theoretical bounds of the multipath error are considered in a scalar steady state situation where the multipath is present and stable until the tracking discriminator zeros the apparent error [5], [44] as shown in Figure 4.4.

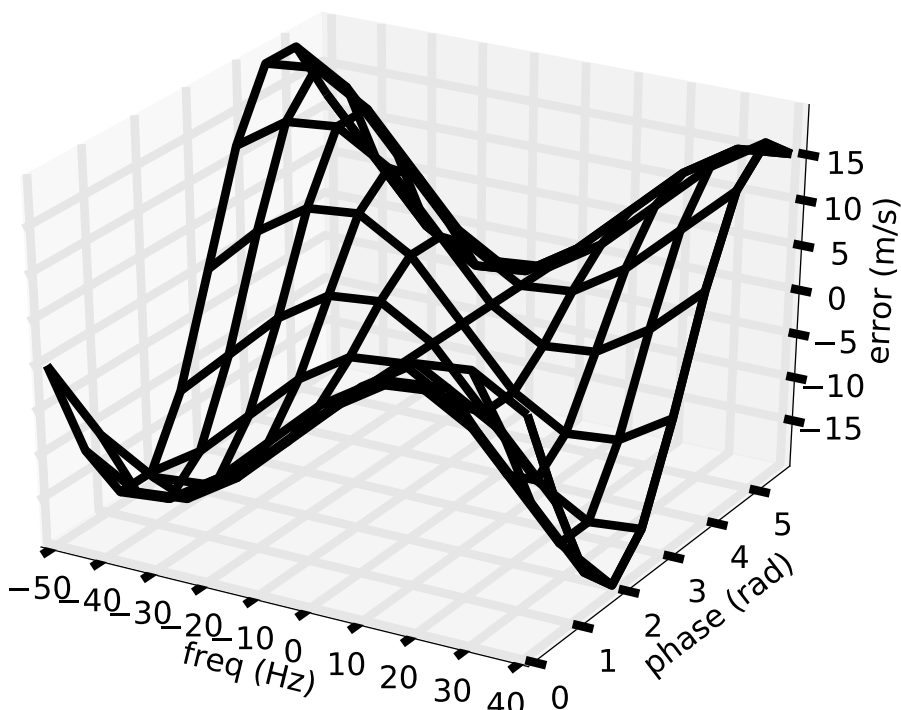


Figure 4.3: Induced range rate measurement error for given multipath frequency and phase errors

The magnitude of this error is also dependent on the various multipath model parameters, MDR, delay, and phase. In this traditional analysis Figure 4.4 is much less symmetric than in Figure 4.2. This is because at steady state, the early and late correlators settle at different delays depending on whether the multipath is constructive or destructive. Therefore the figure has peaks at different locations along the upper and lower bounds. At the initial incidence of multipath, however, whether the signal is constructive or destructive only changes the instantaneous magnitude of the error. Therefore the upper and lower bound peaks are at the same delay, which is in alignment with the late correlator tap setting of half a chip. It is noted in Figure 4.2 that even though the errors peak at the same delay, they are of different magnitudes. This is true because in Equation

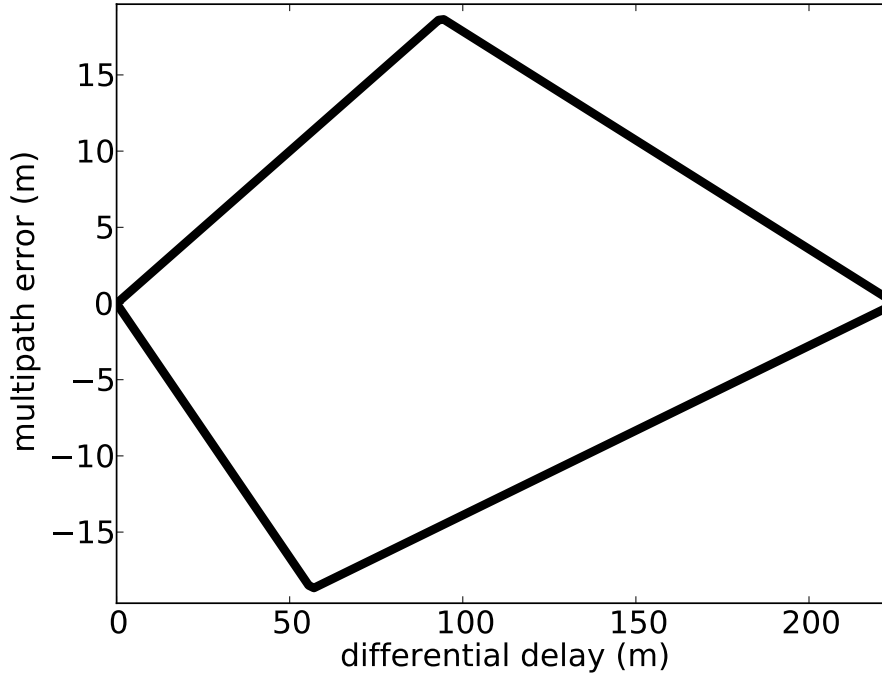


Figure 4.4: Steady state error bounds for maximum and minimum range error in the presence of multipath.

(4.6) the constructive and destructive effects come from the sinusoid term while the error is still biased from A_m^2 term, the square of the multipath amplitude. In Figure 4.4 the actual error is completely dependent on the summed correlation peak and not the discriminator model (assuming same correlator spacing). This is because any unbiased discriminator will eventually drive the channel to equalize the early and late correlator measurements. At that point, the individual discriminator output is zero.

In order to quantify the tracking error induced by the multipath bias, consideration must be taken as to how the measurements are used in the receiver. In steady state, the receiver filter will apply a correction whenever a measurement is made available as

$$\hat{x}_k^+ = \hat{x}_k^- + K_k [\tilde{y}_k - h(\hat{x}_k^-)] \quad (4.8)$$

where \hat{x}_k^+ refers to the state at step k just after a measurement update, \hat{x}_k^- is the state at step k just before a measurement update, K_k is the k th Kalman gain, \tilde{y}_k is the k th measurement, and

$h(\cdot)$ is the measurement prediction function. When multipath is present, the measurement can be modeled with an additive bias as

$$\tilde{y}_k = h(x_k) + b + \nu_k \quad (4.9)$$

where x_k is the true state and b is the multipath induced bias. Introducing this measurement into Equation (4.8) yields

$$\begin{aligned} \hat{x}_k^+ &= \hat{x}_k^- + K_k [h(x_k) + b + \nu_k - h(\hat{x}_k^-)] \\ &= \hat{x}_k^- + K_k [h(x_k) + \nu_k - h(\hat{x}_k^-)] + K_k b \end{aligned} \quad (4.10)$$

This results in the state correction being corrupted by a factor of $K_k b$, which is a function of the steady state Kalman gain and the actual injected bias.

4.1.2 Theoretical Scalar Performance

A scalar receiver uses the incoming measurements to drive the individual estimates of the tracking parameters. Thus a scalar channel has no benefit from information from other channels. In the scalar model, the tracking error generated by the discriminators is an indication that the local replica must be adjusted to drive the error to zero. Therefore there is an immediate compensation by the channel to align the local replica with the perceived tracked signal. In a multipath situation, this drives the single channel replica away from the true signal.

To quantify this temporary divergence, look back at the steady state EKF update described in Equation 3.11. When a new measurement is available, a correction is made with the single erroneous measurement that affects a single channel. Since no information is shared between channels, this error makes a single replica diverge. The scalar measurement update shown in Equation (3.11) gives a direct relation between the local replica errors (the states) and the measurement errors. Therefore multiplication of the steady state Kalman gain with a vector of all zeros except one pseudorange bias produces the effect this error propagates into the state vector.

4.1.3 Theoretical Vector Performance

When performing this same analysis in a vector receiver, it must be noted that the quantity of the measurement error is not the only thing that now affects how a channel is changed. Due to

its very nature, the correction is essentially spread out over all channels by entering the position and velocity estimates. Thus a performance trade-off is made in that any error will propagate to all channels even though the direct effect on the single channel is mitigated. This sharing of information and error is dependent on the satellite geometry, as well as the raw number of signals being tracked.

The vector measurement update shown in Equation (3.6) shows that the matrix of unit vectors C relates the position errors to pseudorange and pseudorange rate errors. That is,

$$x_\rho = Cx_p \tag{4.11}$$

Here, x_ρ is the vector of pseudorange and pseudorange rate errors and x_p is the vector of position state errors. These represent the same errors in different domains, the range domain ρ and the position domain p . These errors are related by the line of sight unit vectors from the receiver to the satellite. Thus, the position error is translated to a range error by taking the dot product of the position error and the unit vector along which the range lies. Similarly velocity errors are related to the range rate errors through the same unit vectors. Therefore the local replica errors are augmented by a bias affecting the vector states as $CK_p b$ where K_p indicates the vector steady state Kalman gain.

Under a steady state scenario, both a scalar and vector receiver are simulated to compare how their measurement updates will affect tracking accuracy. Their steady state Kalman gains are used to multiply the bias. This quantity is the error induced in the states due to the bias. In order to translate this to pseudorange errors, the vector receiver state bias error is multiplied by C to translate it into the scalar receiver domain. The result of these injected biases on a single channel's ability to track are shown below. From Figure 4.5, a consistently greater tracking error is associated with the scalar receiver than the vector receiver. The vector receiver performs with less tracking error due to the linking of the channels through the position and velocity solution. The magnitudes of these errors are low because they are single steps in the filter after the introduction of the multipath error. Therefore the full bias is not appearing in the filtered solution. This sample scenario is taken as representative of multipath cases. Further variations are covered in Section 4.2.

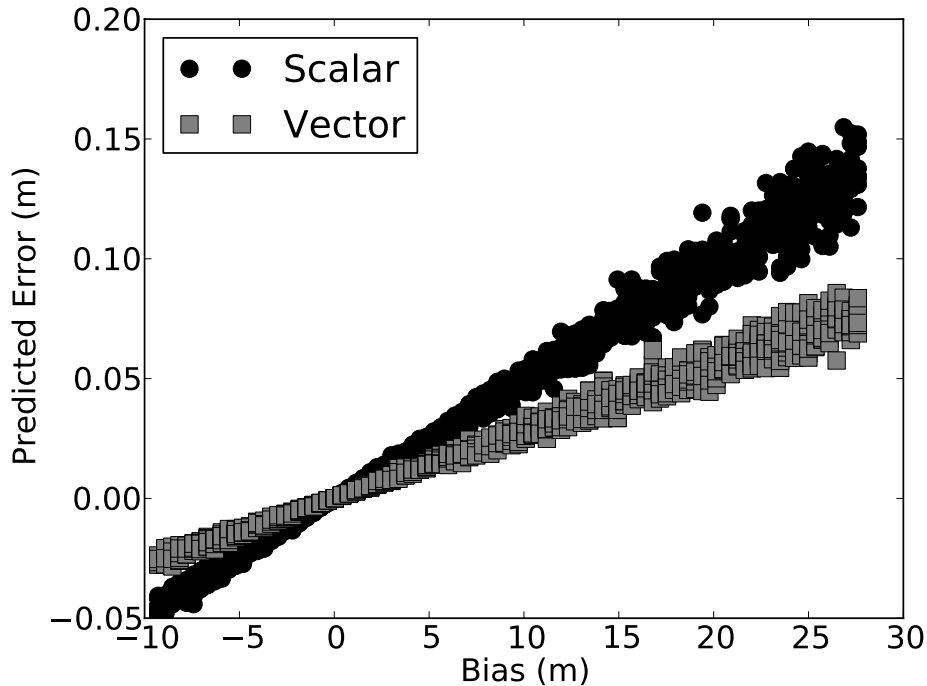


Figure 4.5: Effect of injected bias on single channel tracking error

The above comparison shows that in the range domain a vector receiver induces less error in a faulty channel than an equivalent scalar receiver. It is noted that the trade-off for this improvement is that in a vector receiver other channels will contain induced errors. This spreading of the induced error means that no single channel suffers for a multipath error, but all channels are affected. However, a channel's ability to track is greatly affected by how closely aligned the local replica is with the incoming signal. This is especially true in a traditional scalar receiver where the replica, rather than the range estimate, is used for positioning.

4.2 Multipath Simulation

To test the effectiveness of a vector algorithm in the presence of multipath, a signal simulator was developed. In order to speed development of the algorithm and increase control of the test, the simulator was designed to simulate at the correlator level. In this way each integrate and dump step of the receiver generates only one resulting simulated value. This contrasts with an intermediate

frequency simulator in which each sample is simulated at a much higher frequency. A breakdown of the simulator is shown in Figure 4.6.

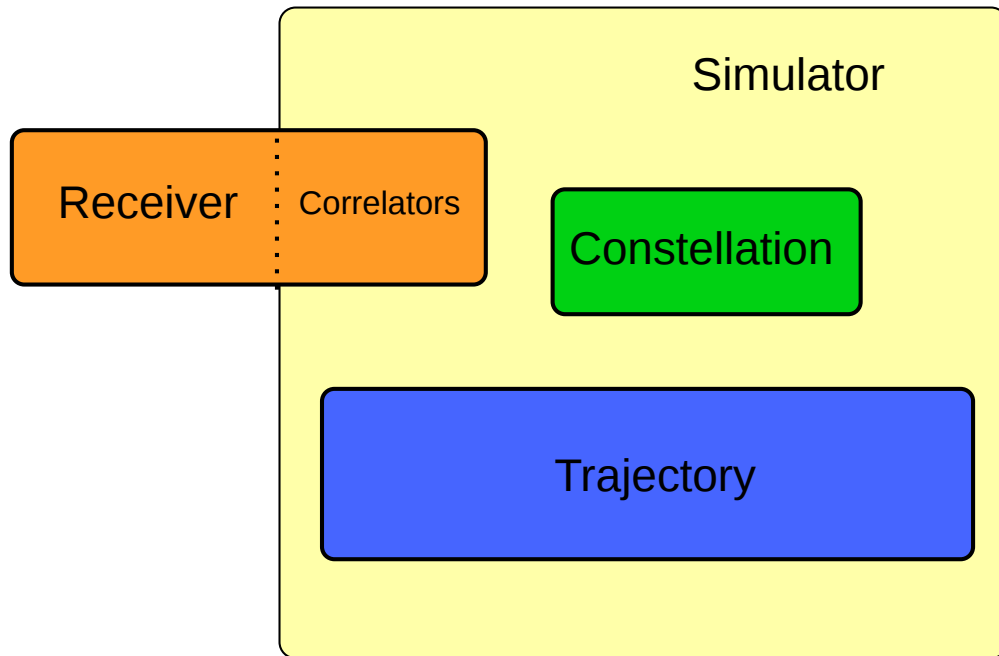


Figure 4.6: Simulator Diagram

Testing is made easy in this situation since no hardware is required between the simulator and the receiver. The connection from the receiver to the simulator is made by overloading the receiver's correlators with simulator-driven outputs. Therefore when a channel requests the correlation at a certain time, the receiver gets the correlation outputs from the simulator rather than performing a correlation with actual IF data. The simulator uses a model of the trajectory to determine the range and range rate errors. This trajectory includes motion along arbitrary paths as well as a clock model for the user. A constellation portion of the simulator determines what satellites are in view at the time. The simulator can also model the effects of a multipath source on the correlation outputs.

4.2.1 Simulation Validation

In order to validate the analysis presented in Section 4.1.1, measurement errors are simulated at the correlator level outputs. The results are shown in Figure 4.7 below. The horizontal axis is the range delay of the reflected multipath signal. The vertical axis is the measurement error

induced by this multipath signal. The error boundary lines up well with that shown in Figure 4.2. The additional noise around the bounds is due to the measurement noise in the simulation. The signal simulated here was at a moderate power of 38 dB-Hz relative to a direct signal at 50 dB-Hz (MDR of 0.063).

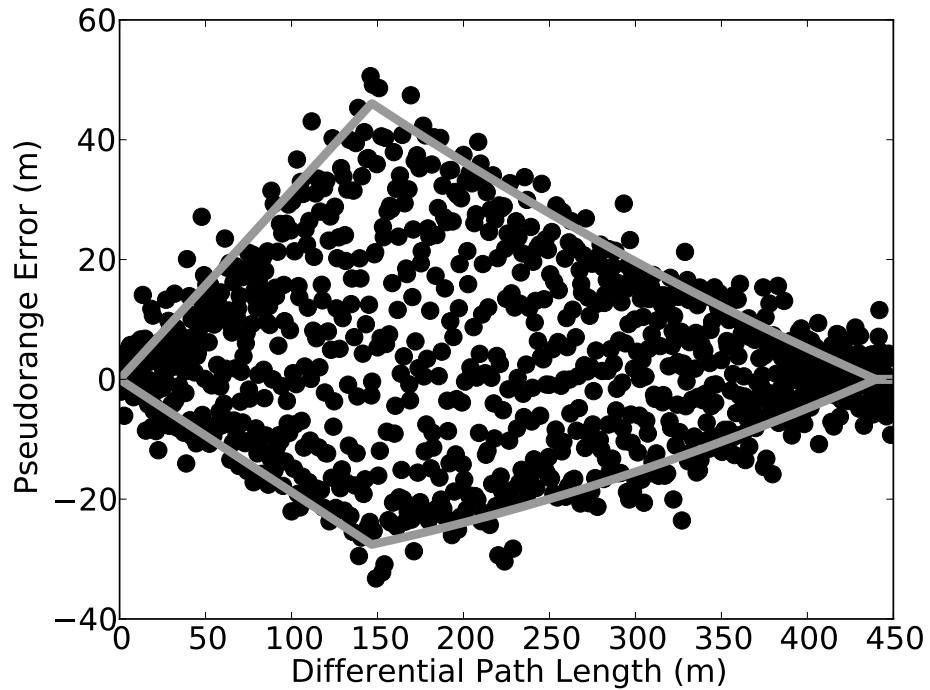


Figure 4.7: Simulated multipath induced bias versus relative delay, shown with predicted error bounds.

Figure 4.7 was generated using perfect knowledge of the direct signal carrier to noise ratio. Because the signal strength is used in generating the measurement, the error will be different in normal simulation. However, the error injected through the additional correlators matches that predicted by the multipath model.

4.2.2 Correlator Models

Since the key component of the simulator is the simulated correlators, an accurate model is needed. In this simulator, the individual integrated and dumped correlator values are simulated

with in-phase and quad-phase components as

$$I(k, \tau) = AR(\epsilon + \tau)D(k) \cos(\pi fT + \theta) + \eta_I(k) \quad (4.12a)$$

$$Q(k, \tau) = AR(\epsilon + \tau)D(k) \sin(\pi fT + \theta) + \eta_Q(k) \quad (4.12b)$$

$$A = \sqrt{\frac{2CT}{N_0} \frac{\sin(\pi fT)}{\pi fT}} \quad (4.12c)$$

where R represents the correlation function of the GPS signal, ϵ indicates the range error, τ is the intentional offset of the correlator (correlator spacing for early, prompt, and late correlators), $D(k)$ represents the k th sample of the simulated data message, f is the frequency error, T is the correlation time, and η is a zero mean unit variance Gaussian distributed noise.

4.2.3 Multipath Model

In order to model the multipath in the simulator, a single reflection model was chosen. A point of reflection is chosen to generate the multipath signal. This point can have its own dynamics in order to induce a different Doppler shift at the receiver. In that case the reflected signal acts as a true signal with a delay equal to the total reflection path. This includes the line of sight distance from the satellite to the reflection point plus the distance from the reflection point to the receiver. For consistent results using the carrier phase, the reflection path is used to determine the phase of the multipath signal. The frequency of the multipath signal is calculated as an additional Doppler shift from the received direct signal at the reflection point, f_M , to the received multipath frequency at the receiver, f_R . Therefore the received frequency of the multipath signal is

$$f_R = \left(1 - \frac{\dot{r}}{c}\right) f_M \quad (4.13)$$

where f_M is the apparent received frequency at the reflection point. This also corresponds to the transmitted frequency of the multipath signal at the reflection point. Also, f_R is the received frequency of the antenna from the reflection. It is different from the received frequency of the direct signal. This difference arises from the range rate between the receiver and reflection point. This is illustrated in Figure 4.8. The phase of the multipath signal is generated in a manner similar to the direct signal. The phase at the reflection point is found based on the range and the transmitted

frequency f_T . Therefore the phase of the multipath signal at the antenna is this phase plus the phase change caused by the additional travel along r at the frequency f_M .

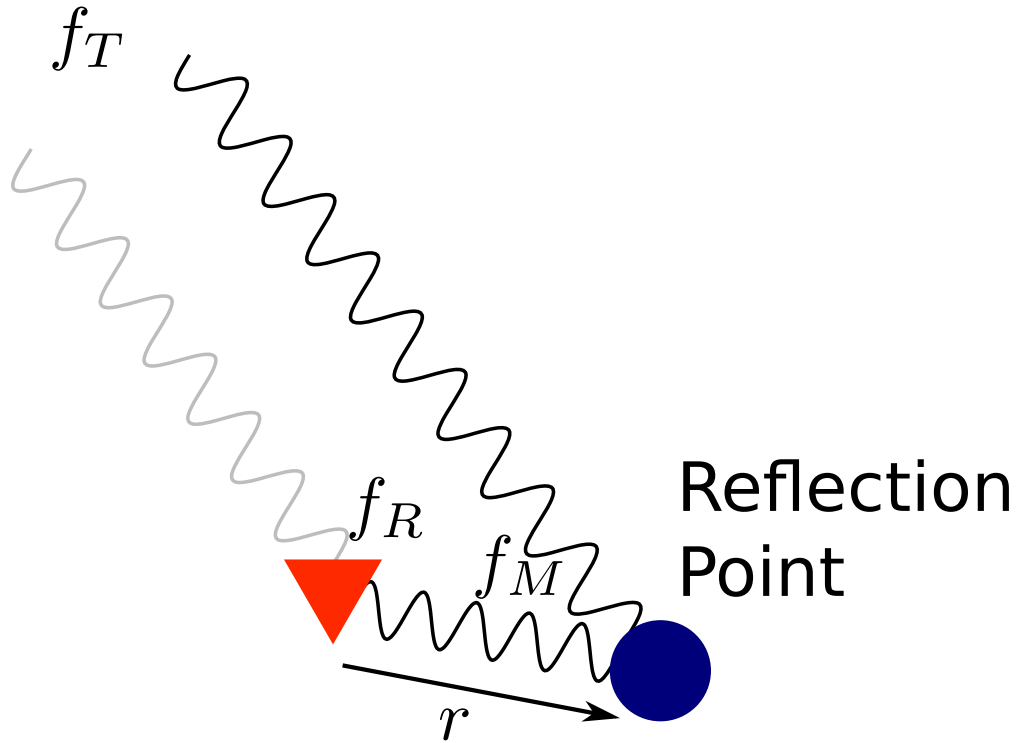


Figure 4.8: Diagram showing multipath reflection received by antenna

4.2.4 Simulated Performance

Under a static scenario, the performance for the scalar receiver and vector receiver are shown in the following figures. In both cases, the multipath error simulated included a phase error that was only a function of the additional range error induced by the multipath. This simulation method ignores those effects dependent on the electromagnetic properties of the reflective surface [6]. The scalar receiver operating under induced multipath resulted in a tracking error with mean $-0.0328m$ and variance $0.0517m^2$. Under the same inputs, the vector receiver resulted in a tracking error of mean $-0.0169m$ and variance $0.048m^2$. The actual tracking errors for both the scalar and vector receivers are shown in Figure 4.9. Notice that both receivers experience a bulge near half a chip of delay ($150m$). This is predicted in Section 4.1.1, particularly Figure 4.2. However, the magnitudes of Figures 4.2 and 4.9 are not the same since the multipath bounds are being filtered in with the solutions in the simulation. These are bounds for the measurement error, not the resulting range

error after the measurements are included in the solution. This deviation is caused by the increased distortion of the correlation peak when the multipath is interfering at the correlator spacing. In this case, that spacing is half a chip. The corresponding position error is shown in Figure 4.10.

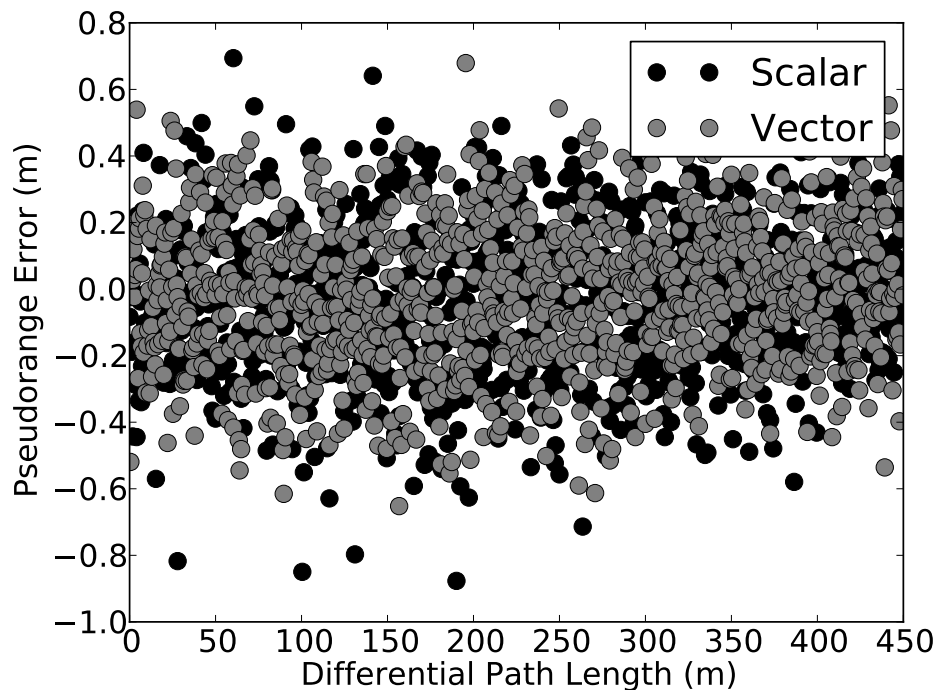


Figure 4.9: Scalar and vector tracking error versus multipath delay

Two tables of results (Table 4.1 for scalar and Table 4.2 for vector) from various scenarios is shown below. The simulated scenarios included variations in satellite geometry. Since a vector receiver uses an overdetermined system to aid tracking its performance will change depending on the geometry and number of satellites available. Therefore, the geometries were generated with 9, 8, 7, 6, and 5 satellites. Also multiple multipath to direct ratios were used from 0.001 to 1, ranging from negligible multipath up to a lossless reflection of the direct signal. Each of these signals is altered to induce a delay across the full range of the channel correlators. The accumulated statistics of each run are shown in Tables 4.1 and 4.2. Plots from each table entry are presented in Appendix D.

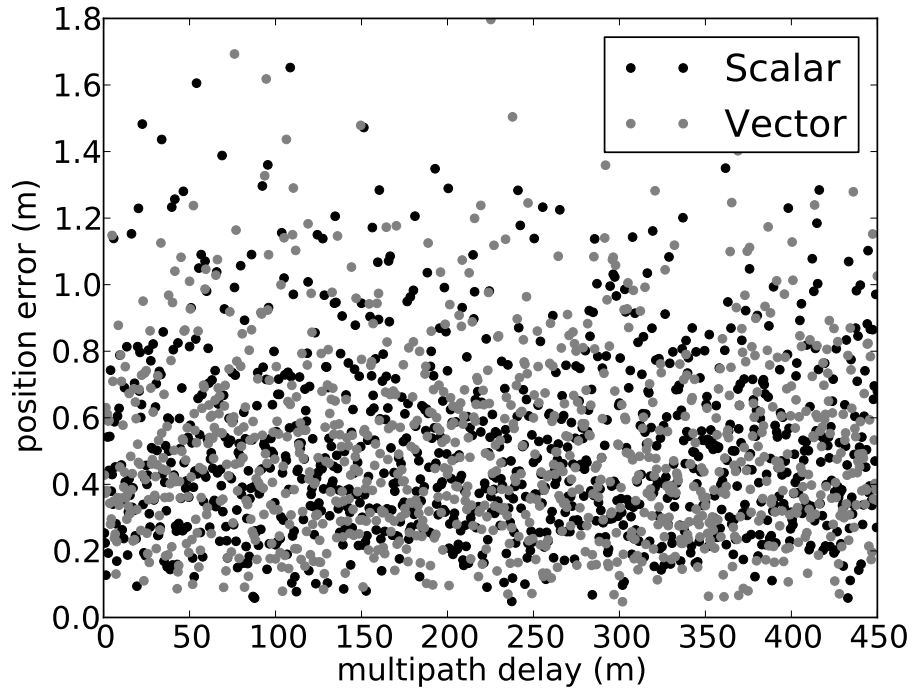


Figure 4.10: Scalar and vector position over simulated multipath range

Table 4.1: Mean and variance from simulated scenarios for scalar receiver

MDR	Number of Satellites				
	9	8	7	6	5
0.001	-0.01066, 0.04193	-0.01676, 0.04124	0.03115, 0.04352	-0.07767, 0.03764	-0.04064, 0.06264
0.003	0.02763, 0.05097	0.07987, 0.04519	0.00959, 0.03685	0.00511, 0.05170	-0.03604, 0.05372
0.01	0.01393, 0.05866	0.03983, 0.04935	-0.00179, 0.04317	-0.00592, 0.02304	-0.01978, 0.04178
0.031	-0.00755, 0.04611	0.03814, 0.04428	0.03659, 0.04182	-0.06869, 0.03568	-0.05120, 0.06457
0.1	0.03342, 0.06467	-0.04872, 0.05250	0.00131, 0.06556	0.02304, 0.04587	0.00014, 0.04630
0.316	-0.02014, 0.05885	0.07092, 0.04203	0.00336, 0.05666	-0.03864, 0.05326	-0.00622, 0.05337
1.0	0.00217, 0.03054	-0.05742, 0.04138	-0.04181, 0.06115	-0.03567, 0.04030	0.04398, 0.07092

Table 4.2: Mean and variance from simulated scenarios for vector receiver

MDR	Number of Satellites				
	9	8	7	6	5
0.001	-0.00761, 0.02176	-0.02774, 0.02313	-0.00068, 0.01661	-0.00814, 0.02919	-0.01772, 0.03725
0.003	-0.00260, 0.02214	0.01900, 0.02891	0.00918, 0.01731	-0.01188, 0.03233	0.01702, 0.02652
0.01	0.01471, 0.01759	-0.02016, 0.02713	-0.02427, 0.02074	-0.00729, 0.02968	-0.00494, 0.02845
0.031	0.00849, 0.02421	-0.00149, 0.01718	-0.02769, 0.01869	-0.00392, 0.02089	-0.01667, 0.03079
0.1	0.02441, 0.02681	0.02745, 0.01737	0.00792, 0.03189	0.04463, 0.02227	-0.05077, 0.03371
0.316	0.00812, 0.01835	-0.01158, 0.03393	0.01812, 0.01919	-0.03170, 0.05462	0.02318, 0.01814
1.0	0.00198, 0.01629	0.03524, 0.02463	-0.00893, 0.02548	-0.00007, 0.01983	-0.01082, 0.02359

4.3 Experimental Results

In order to validate the theoretical and experimental results, two sets of experimental tests were conducted. In each experiment, data was logged by a NordNav GPS front-end and post-processed by both the scalar and vector software receivers. A common truth was taken to be a NovAtel RTK solution using the same antenna as the NordNav. Both receivers were logged concurrently with synchronization taking place in post-process by time stamp matching. Since introduction of controlled multipath errors is difficult in experimental cases, the receivers were placed in areas where multipath is expected to occur.

4.3.1 Static Performance

The first experiment involved testing with a stationary vehicle. Two runs were taken with moderate proximity to a reflective wall (approx 15 m) and close proximity to the same reflective wall (approx 5 m). In both cases, the vehicle was located due North of an East-to-West wall. These two locations will produce differing multipath effects since the length of the delay is dependent on proximity to the reflecting object. The reflective wall was the side of a multi-story hotel in downtown Auburn, Alabama. Several other walls were nearby but the sky was mostly open except for the large wall to the south. Truth was taken from a NovAtel RTK solution. Although it is also adversely affected by multipath, it is taken as the best option for comparison. The horizontal positioning results are shown in Figures 4.12 - 4.14. In Figures 4.11 and 4.13 the results are presented in the UTM coordinate system. In both cases the receivers are initialized with a solution to the west of the convergence point but both come to be centered about the same location. This initial position is generated by a coarse software receiver that decodes ephemerides in order to initialize both receivers.

With moderate proximity to a reflective wall, there are two effects that can be expected. The first is a degraded geometry as satellites from a large portion of the sky are unavailable. This increases the available dilution of precision, also increasing the error. Such a case also limits the capabilities of the vector receiver since it makes use of an overdetermined system to get the most benefit in averaging the channels.

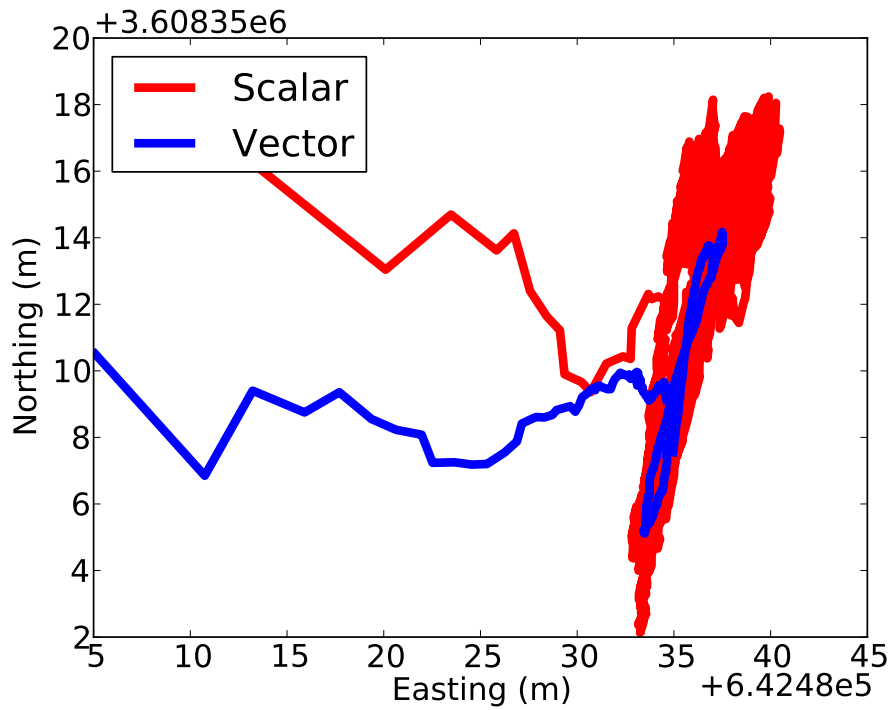


Figure 4.11: Scalar and vector receiver position solution in static scenario with moderate proximity to reflective wall

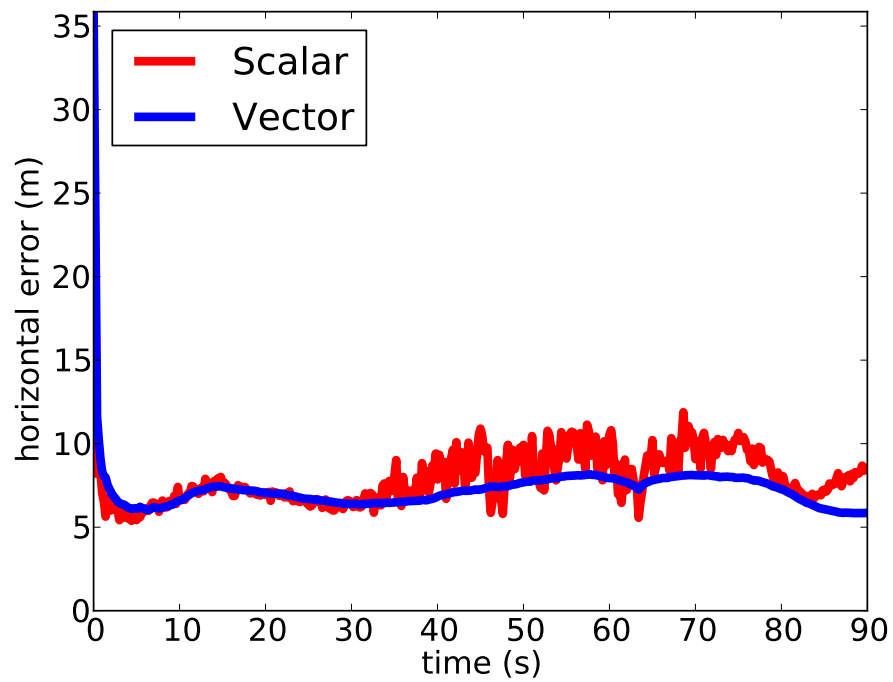


Figure 4.12: Scalar and vector receiver position errors in static scenario with moderate proximity to reflective wall

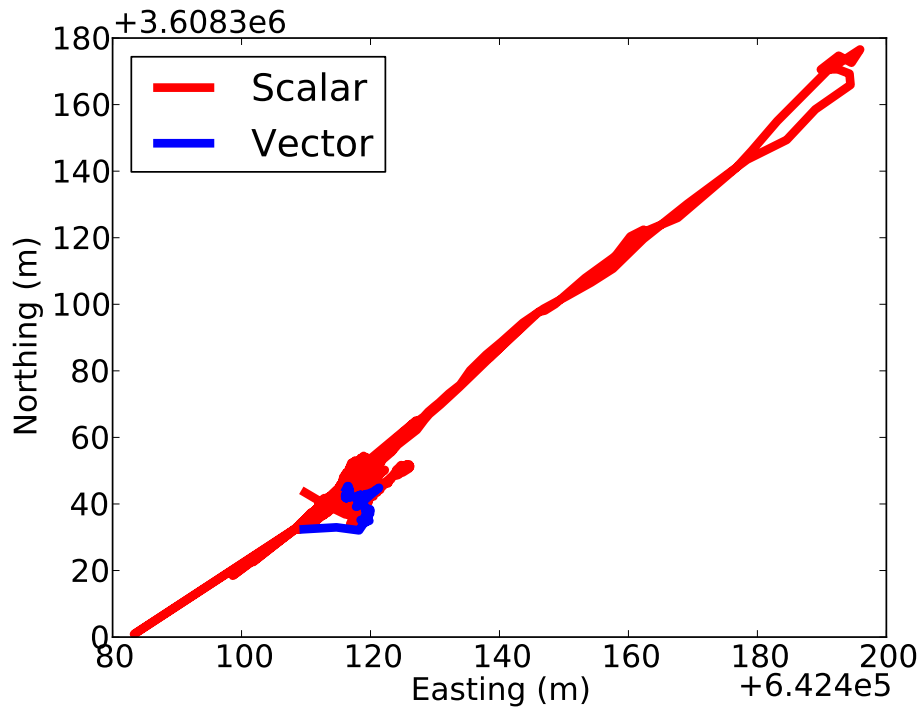


Figure 4.13: Scalar and vector receiver position solution in static scenario with close proximity to reflective wall

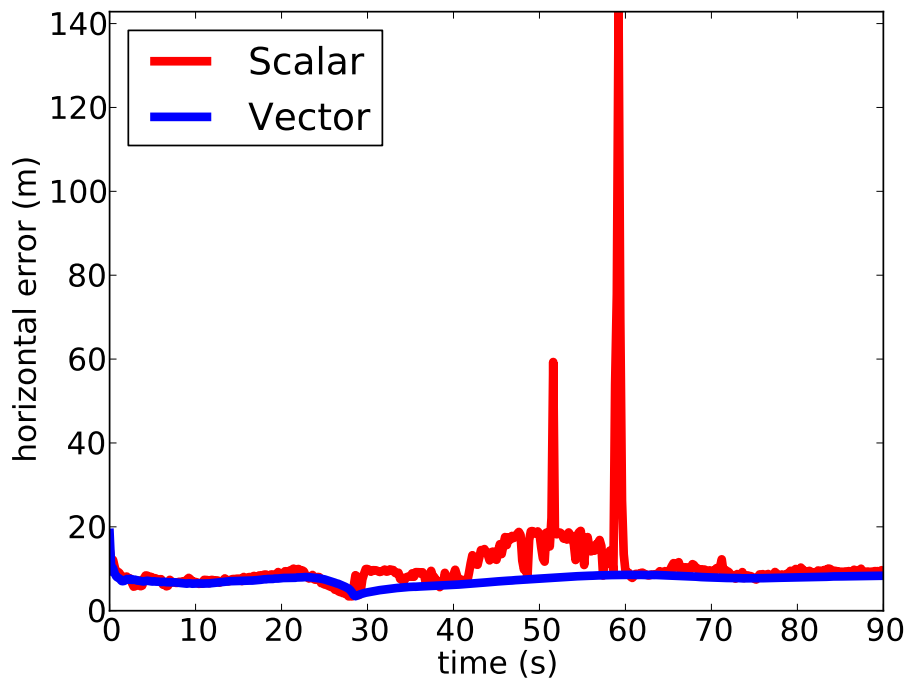


Figure 4.14: Scalar and vector receiver position errors in static scenario with close proximity to reflective wall

The second error mode is from the multipath, which is the desired effect to be seen. In both Figure 4.12 and Figure 4.14 there is a systematic error occurring due to the environment. This error manifests as a slowly varying position error over time. This is assumed to be partially attributable to multipath. In this case, vector tracking is better able to mitigate this error in positioning than the scalar receiver. This is seen in Figures 4.12 and 4.14 since the vector solution consistently shows a lower error. Both cases show superior positioning of the vector receiver. In Figures 4.11 and 4.13 the vector solution keeps a higher precision in these static cases.

In both static experiments a similar motion is observed between the receivers. This is due to the fact that they are using the same raw data file. During both data runs the scalar receiver introduces instances of solution jumps due to its handling of the measurement errors. The runs are short due to the large data size required when operating on raw intermediate frequency GPS data.

4.3.2 Dynamic Performance

A dynamic set of data was taken around multiple reflective walls and along a downtown road with multiple story buildings. One of the main effects of this maneuver was to induce a different received Doppler frequency on the multipath signal. Again the scalar and vector performances are compared as shown in Figure 4.15. From these experiments it is seen that during the run the vector receiver is better able to navigate around multipath sources. The spikes in the scalar error are strictly due to the handling of the environment induced errors. Since the vector receiver spreads out this error among its channels, its position error does not spike in a similar manner. Toward the end of the segment, one of the scalar channels begins to lose lock in the scalar receiver while the vector receiver could continually operate. This takes place at a long intersection with moderate foliage on most sides.

4.4 Summary

This chapter presents a novel comparison of scalar and vector receivers in multipath scenarios. Since the traditional multipath error model is based on scalar assumptions, a new measurement model is developed. This model is based on correlation of the line of sight signal with a delayed replica. The theoretical performance in the presence of this error predicts that the vector receiver

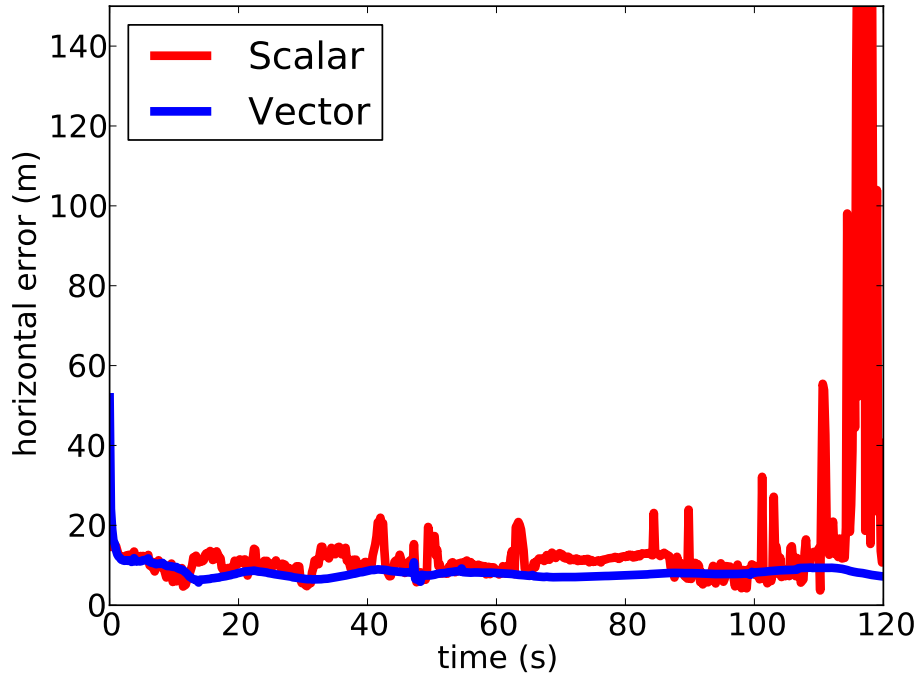


Figure 4.15: Scalar and vector receiver position errors in dynamic scenario.

is better able to handle the biased measurements. Simulated and experimental results also confirm that the vector receiver can operate more accurately in multipath environments. In the experimental results, vector tracking reduced the positioning error several meters with respect to the scalar receiver.

Chapter 5

Receiver Lock Detection

One of the major challenges to the widespread adoption of vector tracking is the lack of integrity assurance. Several methods exist to determine lock in scalar architectures. This process is more straightforward because each part is self-contained and can reasonably be treated with some linear assumptions. However, vector receivers blend errors from all channels through the position and velocity solution. Thus it is difficult to determine when a channel is not operating correctly.

This chapter first presents some background on lock detection. A demonstration of the robustness of the vector solution is then given. This process provides clear bounds on viable vector solutions. Finally a novel variance parameter is derived and demonstrated to serve the purpose of vector state detection.

5.1 Channel Lock Detection

The operation of a receiver channel can continue whether or not a received signal is present. This can be the case when a signal is blocked, shadowed, or goes below the horizon. As long as front-end data is provided to the channel, the correlations can be performed, discriminators can operate, and the loop filter can update. However, if there is no signal present these operations will only be driven by noise and will tend to walk randomly away from the range and velocity for that satellite. Lock detection allows a receiver to determine when this is taking place. In typical scalar tracking for a GPS receiver, lock detection is an important mechanism for operation. It is another of the baseband operations that allow a receiver to calculate and maintain positioning [45]. Without some indication that lock is being maintained, there will be no indication when the channel's measurements should not be used. Lock detection is important in a receiver context because it determines when acquisition should be performed again as in Section 2.4. A loss of lock is distinguished from blockage as discussed in [19]. In a temporary blockage situation there is an expectation that if the blockage is removed the loop will again provide useful tracking information.

A loss of lock is declared when the receiver no longer expects to provide useful information if blockage is removed. While blockage over a sufficient time (varies, but typically around ten seconds) triggers loss of lock, a blockage can occur without the need to trigger loss of lock. The important point of lock detection is that the receiver reacts when it will not be able to operate without reinitialization. This occurs when the local replica and incoming unblocked signal are sufficiently unmatched that the discriminators will no longer provide accurate error signals to the tracking loops. This region of lockable error is referred to as ‘pull-in range’ in Section 2.3.

Loss of channel lock is not directly applicable to a vector receiver due to the way the local replica is generated. Whereas scalar channels use their loop filter output to drive the local replica generation, vector channels use the estimated position and velocity solution to generate the local replica. Therefore, vector tracking is considered ‘permanent lock’ [39]. As long as position remains sufficiently accurate from the tracking of other signals, reacquisition is not needed. However, the receiver needs to use a detection metric to know when to ignore signals from a particular channel.

Several methods for detecting channel lock have been developed for a scalar receiver. These techniques typically use the correlator outputs to determine whether the channel is in lock. This is natural since the correlator outputs determine how closely the incoming and local replica signals match. Alternative methods to detect lock based on code and carrier tracking loop outputs, the shape of the autocorrelation function, or received signal strength have been developed [19]. However, the implementation of these methods are still based on detecting individual channel lock using correlations between the incoming signal and the local replica.

5.1.1 Lock Detection Based on Carrier to Noise Ratio

One metric that can be used in determining a loss of lock situation would be a comparison of the C/N_0 to some threshold. Since most receivers have difficulty operating with signals weaker than a certain C/N_0 , this can be helpful in maintaining confidence in the solution. However for vector tracking this threshold can be difficult to describe. It is typically reported from Monte Carlo type simulations [35]. For a scalar loop it is more straightforward since the tracking error is directly related to only one signal, without including the complexity of the satellite geometry and signal power distribution between channels.

5.2 Scalar Lock

When a channel is considered in lock, it is generating beneficial tracking information from the incoming signal. The fidelity and accuracy of the error generation mechanism determines how closely the local replica matches the incoming signal. When the tracking errors are driven to zero, the local replica is in correct alignment with the received signal. Any motion between the receiver and satellite will change this signal and it is the job of the channel to keep the error close to zero. There are many instances when this is not possible. During a typical period of operation a signal may be blocked, shadowed, reflected, or lost. In these cases the incoming signal is slightly or completely corrupted and the tracking loop will not be able to generate accurate tracking errors. Without accurate tracking errors, the loop may track random noise, possibly away from the true signal. Lock detection is designed so that a channel can determine when it thinks this is happening. With a scalar receiver, the channel can be reinitialized through the acquisition engine, just as it was when it first started tracking the signal. However, this is a time-consuming process and is avoided whenever possible. If differential carrier phase is being used, this would also require new estimation of the integer ambiguity.

In a scalar loop, a channel can use several metrics to detect whether or not lock is maintained. Since tracking requires both code and carrier loops to be in lock, detection typically occurs based on the carrier loop since it is more likely to lose lock [45]. A phase lock loop indicator can be designed based on the fact that when tracking is properly operating the I correlation is maximum and the Q correlation is minimum. Each is summed and filtered over an interval then compared. The loop can be designed with two decision thresholds: an optimistic and pessimistic decision. The optimistic decision confirms that lock is valid quickly and indicates loss of lock only after several failures to meet the threshold. The pessimistic decision confirms lock slowly only after the I is greater than Q for many steps, while it indicates loss of lock after only one failure.

Another scalar channel lock detector is the false frequency lock detector. This detector compares the cross and dot terms from a frequency lock loop. If there is an error detected, a 25 Hz correction can be triggered to correct the carrier frequency.

5.3 Vector Lock

The difficulty in making a loss of lock call for a vector receiver comes from the very advantages of the vector receiver. Since the vector receiver is predicting the code phase and carrier frequency based on known satellite position and velocity and estimated user position and velocity, the uncertainty is in the PVT of the user. This uncertainty permeates all channels at once. Therefore the ability of the vector receiver to continue tracking is determined by the correlator's ability to generate an error estimate between the local replica and the received signal. Since the performance of this tracking is situation dependent (being a function of received signal strength, PVT uncertainty, and satellite geometry), the lock detection problem is nontrivial. In this dissertation, proper tracking is defined as the ability of a receiver to gain useful information from correlator outputs.

In the basic vector formulation described in Chapter 3, all channels are used in the navigation solution calculation and therefore in the tracking variable estimation. This is true even when the signals are blocked. This is possible because these measurements are weighted based on the estimate of the signal C/N_0 , which drops when a signal is blocked. However, no declaration of loss of lock is made since the receiver will continue to predict the local replica without continuous correlation error information.

In the open literature, several sources describe methods used in simulation to determine when loss of lock of the solution occurs. A rule of thumb tracking threshold is described in [43] as

$$3\sigma_w + f_e \leq \frac{1}{4T} \quad (5.1)$$

where σ_w is the 1σ frequency jitter from thermal noise, f_e is the dynamic stress error in Hz, and T is the integration time. The relation in Equation (5.1) is used in [34] to compare tracking performance to a Monte-Carlo simulation of a vector receiver. In [34], two steps are taken to find the value for σ_w . First, the steady state sub-optimal Extended Kalman gain is found with the modeled process noise present. This assumes that the satellite geometry is fixed, measurements arrive concurrently, and the EKF is operating close to the true state (small state errors). Second, the EKF state covariance matrix is found using the steady state gain but with only clock error and measurement noise modeled. Thus, the covariance matrix is a function of the clock characteristics and the

measurement variance, which is a function of the estimated C/N_0 . This technique is described in more detail in [34]. The tracking threshold is compared with varying levels of f_e . This threshold is compared to a Monte-Carlo simulated set of receiver architectures. For the vector architectures, lock is declared lost when any of the pseudorange discriminator errors exceed a half chip. This corresponds to the one-sided range of the nonlinear discriminator function, which is dependent on the Gold code autocorrelation function described in Section 2.3.3. It is admitted in [34] that this comparison between the code range in the vector architecture and the frequency threshold in the rule of thumb is not a straight comparison. This is due to the fact that loss of lock is checked for two different domains, the range and range rate. However, the author states that the lock comparison is still valid because the vector receiver can tolerate occasional loss of frequency lock without diverging. It is also stated that when loss of lock does occur, a catastrophic failing begins which quickly causes the range to cross the half-chip threshold.

In [39], the operation of a standard DLL/FLL scalar receiver is compared against a VDFLL receiver in blockage situations. The demonstration includes a user-defined threshold based on estimated C/N_0 . In this demonstration, loss of lock is declared at 20 dB-Hz for a scalar channel and 25 dB-Hz for the vector channel. Even with this 5 dB-Hz threshold difference, the vector receiver is able to operate better in a blockage scenario.

In [1], results from code tracking are simulated and reported. In this case, the wideband noise floor is raised until the 1σ error of the code tracking loop reaches $1/6$ of a chip. Once this threshold has been reached, the loop is said to have lost lock. For the scalar receiver, this ends the simulation. However, for the vector receiver, the unlocked loop is excluded from the solution and a check of the other loops is done. If none of the other loops have reached the loss of lock threshold, the simulation continues as the VDLL is still tracking [1]. This continues until less than four loops are still locked, at which the simulation is terminated. This requirement of at least four satellites for vector operation is also mentioned in [30]. In the face of these differing criteria, a new and absolute criteria was developed for declaring vector receiver failure.

5.3.1 Vector Divergence

Since tracking lock is a method to determine when the receiver cannot continue operating without reinitialization, the bounds of this situation must be developed. This is accomplished by

determining the errors from which a vector receiver can be expected to recover. A static portion of raw intermediate frequency (IF) data was collected from a NordNav GPS front end. The receiver was connected to a NovAtel survey antenna on the roof of Auburn University's Shop 3, above one of the GPS and Vehicle Dynamics Laboratory office locations. This antenna location provides a good view of the sky with very few obstructions. The receiver was initially run with scalar tracking for satellite acquisition, data demodulation, and initial position calculation. After this information is gathered, the data set is run with an initialized vector receiver. For each satellite in view, a combination of position and velocity errors are injected along its line of sight. This situation is a worst-case scenario for tracking the signal from this satellite, as shown in Figure 5.1. The u terms represent the unit vectors to each satellite.

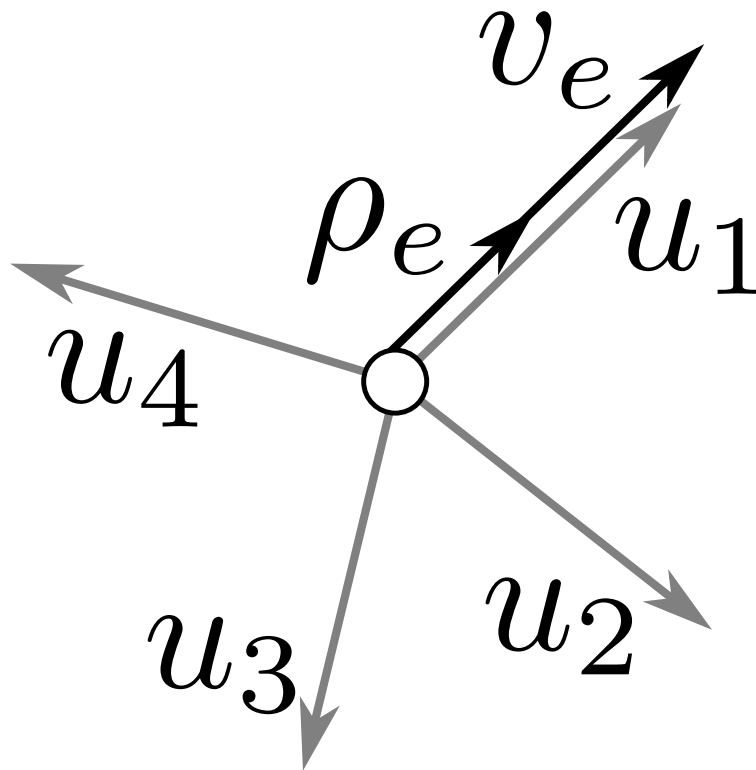


Figure 5.1: Error injection vector along line of sight to one satellite.

Due to the nature of the vector receiver, this error will also be viewed from the other channels, but with a lesser effect. This is because the line of sight error for all other channels is less than the total error magnitude because the error is only projected onto the channel line of sight. Since position and velocity information will be produced from each channel, four channels are sufficient

to keep the receiver in lock. Therefore, given the position and velocity error, these errors must be projected onto the line of sight for each tracked satellite. If the projected errors are inside the discriminator ranges for less than four satellites, failure has occurred. To demonstrate this criterion, an experiment was conducted.

5.3.2 Vector Divergence Experiment

This experiment is run to determine at what combination of position and velocity error the receiver will fail to converge. When such a failure occurs, this indicates a situation that a vector receiver cannot recover from. With these bounds, the point at which lock detection must occur is determined. After this point the receiver must reacquire. Since the vector receiver is allowed to converge for an initial period, a better idea of the divergence performance is acquired.

Convergence is declared when the position and velocity errors are less than half the injected error for position and velocity, respectively. This indicates that a vector receiver is capable of recovering from this error since the solution is driving the position and velocity error toward zero. The receiver is run for an additional twenty seconds after the errors have been injected to check for convergence. If convergence is detected, the run is complete and moves to the next error combination. If convergence is not detected, another ten seconds are allowed to elapse. If the convergence check continues to fail up to sixty seconds after the error injection, the receiver is declared to be diverged. Results from this divergence experiment are shown below in Figure 5.2. Results from this same constellation are shown from simulation in Figure 5.3. In both cases a band is shown that indicates the predicted failure condition. This is the combination of errors that would leave less than four channels within their discriminator ranges.

Because of the discriminator functions described in Section 2.3.3, there is only a certain region in which a code phase or carrier frequency error will generate useful errors. This is the region shown in Figures 5.2 and 5.3. Within this region, the receiver quickly recovers its solution and does not need to reinitialize. In this linear region, correlations can be combined by a discriminator to keep the vector receiver in tracking mode. Outside this region, a discriminator will not yield accurate measurements and can actually cause the receiver to diverge. However, since the vector receiver utilizes weighted information from multiple satellites, it is only when the error is large enough to push more than four channels into this divergent situation that overall divergence occurs.

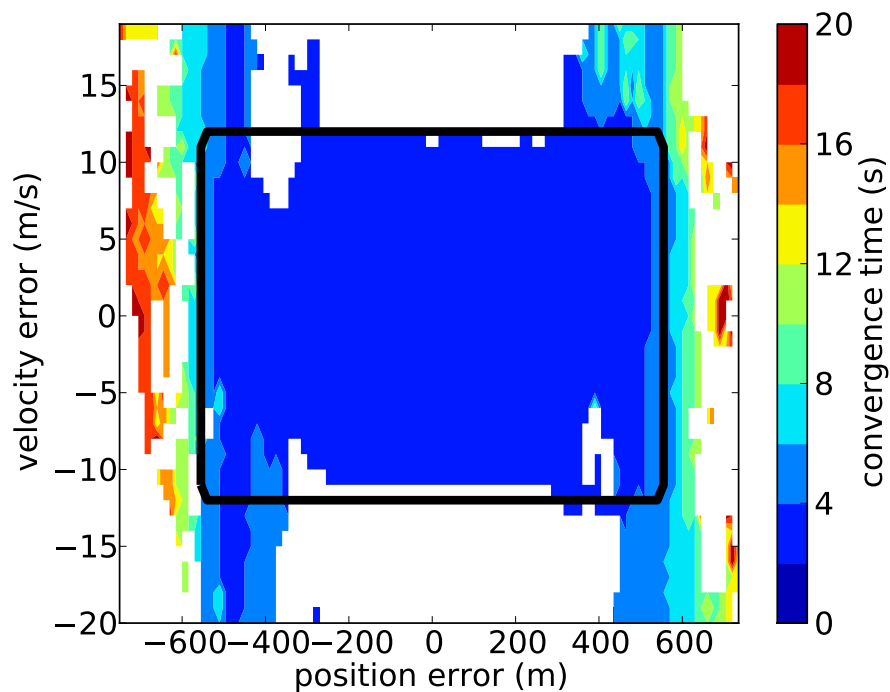


Figure 5.2: Divergence experimental results for injected line of sight position and velocity errors on satellite 14.

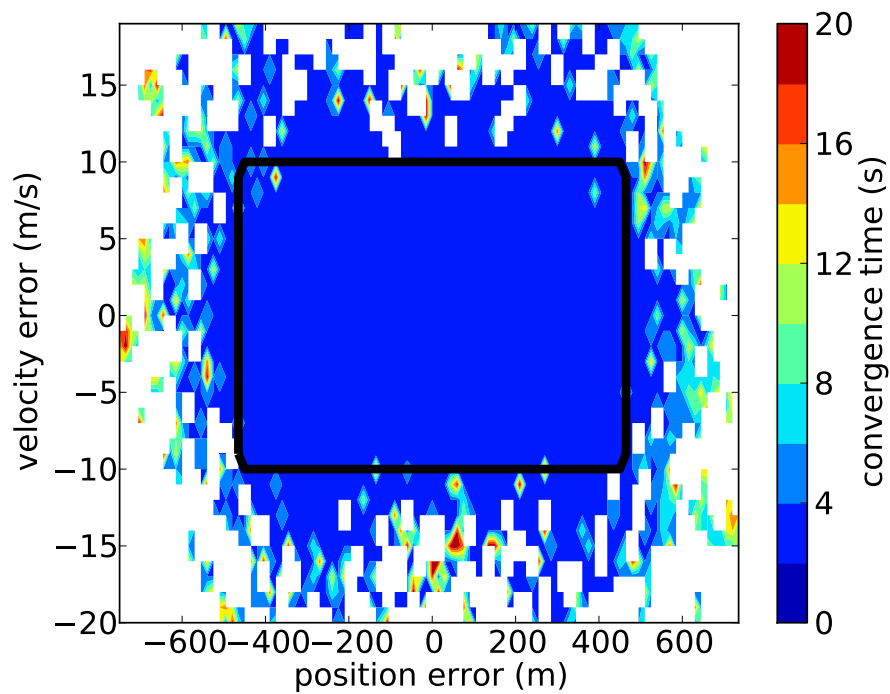


Figure 5.3: Divergence simulated results for injected line of sight position and velocity errors on satellite 14.

5.4 Maximum Range Variance

The navigation system's concept of uncertainty is tied up in its state covariance matrix, P . Whenever the state vector, \bar{x} , is updated, so is the covariance matrix. The proper updating and propagation of P is what determines the performance of a Kalman filter, particularly the vector tracking filter. This section will derive a new parameter, the range variance, that will be used to monitor the uncertainty of a vector receiver.

The state vector in Equation 3.1 is ordered as four pairs of position and velocity components. If it is multiplied by a row selector matrix

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

then

$$S\bar{x} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta cb \\ \delta \dot{x} \\ \delta \dot{y} \\ \delta \dot{z} \\ \delta \dot{cb} \end{bmatrix} = \begin{bmatrix} \bar{x}_r \\ \bar{x}_{rr} \end{bmatrix} \quad (5.3)$$

where the range related states (position and clock bias), \bar{x}_r , and range-rate related states (velocity and clock drift), \bar{x}_{rr} , are in separate sub-matrices. A similar operation can be performed on the state covariance matrix, P , to break it into corresponding sub-matrices. To do this the row selector

matrix, S , is applied to both row and column as

$$SPS^T = \begin{bmatrix} P_r & P_{r \times rr} \\ P_{r \times rr} & P_{rr} \end{bmatrix} \quad (5.4)$$

where the matrix is partitioned into sub-matrices related to the range state covariances, P_r , the range-rate state covariances, P_{rr} , and the cross-covariances, $P_{r \times rr}$.

As in Equation (3.6), the range errors are related to the state errors by the matrix of line of sight vectors

$$C_r = \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & a_{y,N} & a_{z,N} & 1 \end{bmatrix} \quad (5.5)$$

as

$$\bar{y}_r = C_r \bar{x}_r \quad (5.6)$$

Under the assumptions of the vector tracking Kalman filter these range states are noisy such that

$$\bar{y}_r = C_r (\bar{x}_r + \mu_r) \quad (5.7a)$$

where μ_r is the vector of noise on the state, which is ideally zero mean with covariance P_r . Of course the receiver does not have an accurate measure of the state noise statistics. The state covariance is the filter's best estimate of the distribution of the state noise. However, with a properly tuned system this is a serviceable estimate.

Under these noise assumptions and using Equation (5.7) then

$$\begin{aligned} \text{E} [\bar{y}_r] &= \text{E} [C_r \bar{x}_r + C_r \mu_r] \\ &= C_r \bar{x}_r + C_r \text{E} [\mu_r] \\ &= C_r \bar{x}_r \end{aligned} \quad (5.8)$$

and

$$\begin{aligned}
\mathbb{E} \left[(\bar{y}_r - C_r \bar{x}_r) (\bar{y}_r - C_r \bar{x}_r)^T \right] &= \mathbb{E} [C_r \mu_r \mu_r^T C_r^T] \\
&= C_r \mathbb{E} [\mu_r \mu_r^T] C_r^T \\
&= C_r P_r C_r^T
\end{aligned} \tag{5.9}$$

Under the vector tracking assumption of permanent lock, a single metric is needed to monitor the receiver's lock state. This is in opposition to a scalar receiver which monitors each channel for lock. As shown in Section 5.3 the point at which the vector receiver is out of lock and unable to recover is dependent on how the position and velocity estimates relate to the discriminator limits. At such high uncertainty the receiver would have little hope of recovering and reinitialization would be needed. The parameter proposed here is the maximum range variance given the state covariance matrix.

As shown in Equation (5.9), the covariance of the range errors is dependent on the line of sight vectors in C_r and the current state covariance P_r . This matrix gives the range variances that are induced by the position uncertainty. The quantity in Equation (5.9) translates the state covariance into the variances of the individual channels. However, this quantity is highly geometry dependent. The range uncertainty would need to be monitored for each channel, which is not what is needed in vector tracing.

With this newly proposed method, the covariance matrices can be manipulated to get a single monitoring range variance. This single variance is a good indicator for the entire receiver. This variance corresponds with the direction of maximum range uncertainty. Consider a partitioning of

the range related state covariance matrix P_r as

$$P_r = \begin{bmatrix} P_p & \rho_c \\ \rho_c^T & \sigma_{cb}^2 \end{bmatrix} \quad (5.10a)$$

$$P_p = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y & \rho_{xz}\sigma_x\sigma_z \\ \rho_{xy}\sigma_x\sigma_y & \sigma_y^2 & \rho_{yz}\sigma_y\sigma_z \\ \rho_{xz}\sigma_x\sigma_z & \rho_{yz}\sigma_y\sigma_z & \sigma_z^2 \end{bmatrix} \quad (5.10b)$$

$$\rho_c = \begin{bmatrix} \rho_{xc}\sigma_x\sigma_{cb} \\ \rho_{yc}\sigma_y\sigma_{cb} \\ \rho_{zc}\sigma_z\sigma_{cb} \end{bmatrix} \quad (5.10c)$$

where σ_i^2 represents the variance of state i , ρ_{ij} is the correlation coefficient between state i and j . Since the sub-matrix P_p is itself a covariance matrix, it can be decomposed into a matrix of its eigenvectors Q and the matrix of its eigenvalues Λ . The largest eigenvalue of P_p , λ_1 , corresponds to the highest variance of a single direction vector. The eigenvector that corresponds to λ_1 is the corresponding direction vector q_1 . Therefore the total variance of this direction can be found by augmenting the eigenvector with a fourth row with a value of 1 for the clock bias. Now

$$\begin{aligned} \sigma_r^2 &= \begin{bmatrix} q_1^T & 1 \end{bmatrix} \begin{bmatrix} P_p & \rho_c \\ \rho_c^T & \sigma_{cb}^2 \end{bmatrix} \begin{bmatrix} q_1 \\ 1 \end{bmatrix} \\ &= q_1^T P_p q_1 + 2\rho_c^T q_1 + \sigma_{cb}^2 \\ &= \lambda_1 + 2\rho_c^T q_1 + \sigma_{cb}^2 \end{aligned} \quad (5.11)$$

An example of the maximum variance calculation is shown below in Figure 5.4 The figure is shown using standard deviations to reduce the distortion of squaring a range. The unit vector and standard deviation, σ_r , found from Equation (5.11) are shown as the line and point, respectively. The surface represents the standard deviations from other unit vectors. Figure 5.4 demonstrates the ability to obtain the direction of maximum range error for a given covariance matrix of position and clock bias errors. This variance parameter will be used as an indicator of vector receiver integrity. The bulge to one end of the surface is a result of the application of the clock cross-covariances.

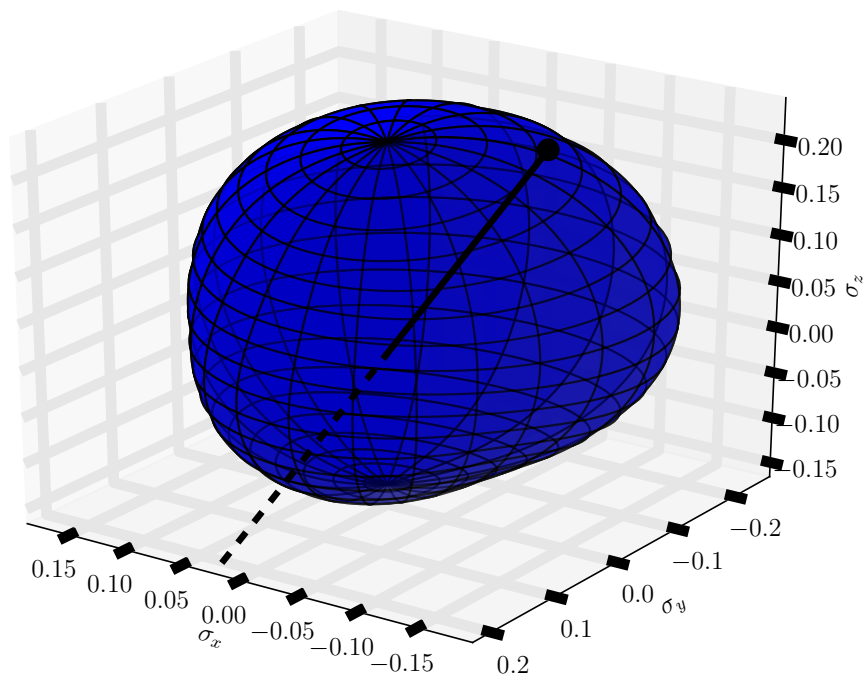


Figure 5.4: Found maximum standard deviation direction and value with surface of standard deviations for a surface of signal directions.

This maximum range variance parameter relates back to the vector receiver limits described in Section 5.3.1. When dealing with the limits of when the receiver will converge, discriminator ranges for the full available constellation are taken into consideration. These limits are geometry dependent and change as the satellite line of sight vectors change. The maximum range variance parameter relates to the discriminator ranges for a single line of sight. It relates to the bounds on the range variances for a single direction, regardless of the current geometry.

5.4.1 Range Variance Verification

To verify the use of the range variance parameter, a static simulation was performed. The range error results are shown in Figure 5.5. Also shown are the one and three sigma bounds of the range variance parameter over the simulation. As can be seen, this metric bounds the range errors well.

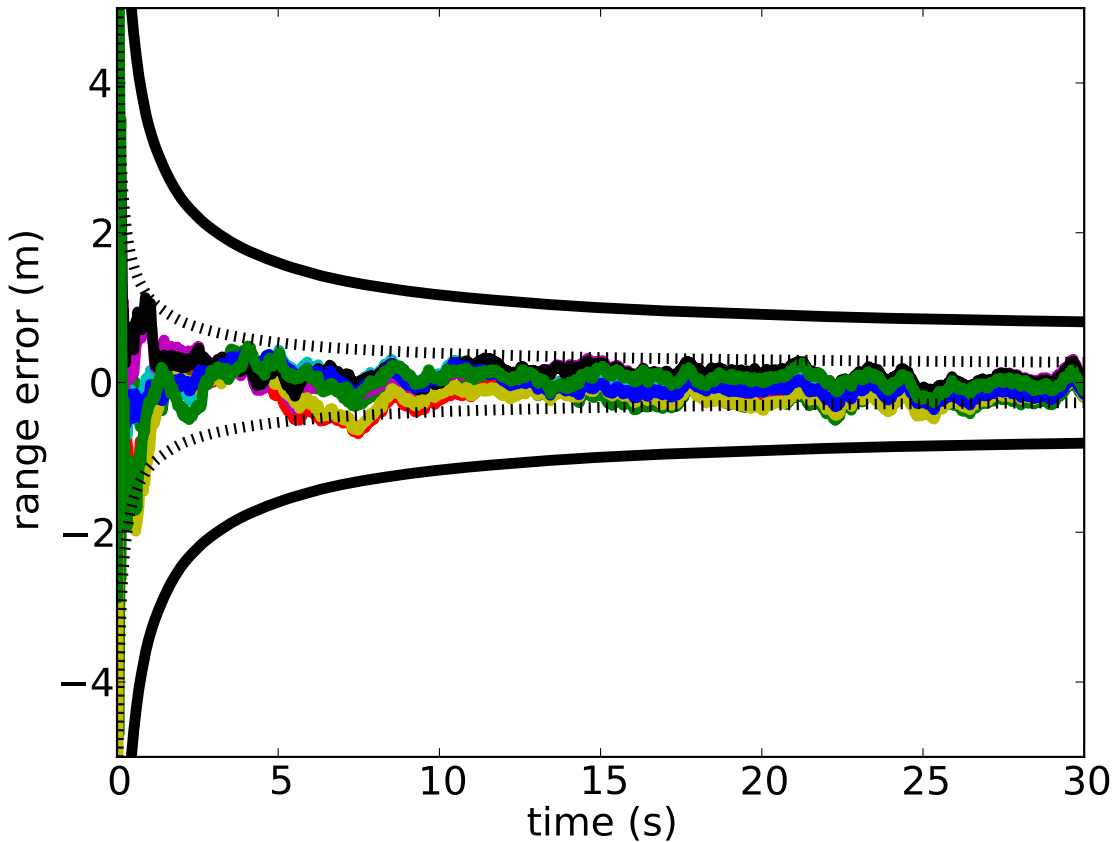


Figure 5.5: Range vector bounds on simulated range error for a static scenario.

5.5 Range Variance Results

The range variance parameter is useful in determining the current state of the vector solution. This is demonstrated with live sky data. This data set was logged on June 7, 2009 by Matthew Lashley and John Allen. It was taken through downtown Atlanta to test the vector receiver architecture in an extreme urban canyon. The raw data was generated by a NordNav GPS front-end. It was recorded using the NordNav's provided logging software.

The overview of the route is shown below in Figure 5.6. The variance parameter is shown as the color of the solution, ranging from magenta (low variance) to red (high variance). The range of colors was selected to accentuate the times the receiver detected degraded environment.

Two sections of severe blockage are shown in Figures 5.7 and 5.8. Figure 5.7 shows the increasing variance while traveling under two overpasses. The test vehicle entered the section from the South-East. The delay in the parameter increase occurs because of the long averaging time necessary for the amplitude and C/N_0 estimation. Figure 5.8 is a segment in a severe urban canyon. The prolonged poor environment leads to drift of the solution. However, the range variance parameter indicates this degraded situation.

The mechanism by which the maximum range variance can indicate the loss of signal is through the amplitude estimator described in Section 3.3.1. The amplitude estimate when going beneath the first overpass in Figure 5.7 is shown below in Figure 5.9 for satellite 31. This amplitude also drives the C/N_0 estimate as shown in Figure 5.10, which also drops beneath the overpass. Note that the C/N_0 is dropped to zero when the amplitude and noise lines meet in Figure 5.9. These decreasing estimates increase the measurement variance for this satellite as shown in Figure 5.11. This occurs for all signals, leading to the increased maximum range variance parameter as shown in Figure 5.7. In Figures 5.9 through 5.11 the approximate time of entering the overpass is shown with vertical lines. These times were taken from visually determining the longitude at which the path went beneath the overpass.

From these results it is shown that the vector receiver can monitor the received amplitudes of the signal components. Using these estimates and the measurement covariance calculations leads to less correcting of the state covariance. When this occurs, the maximum range covariance parameter

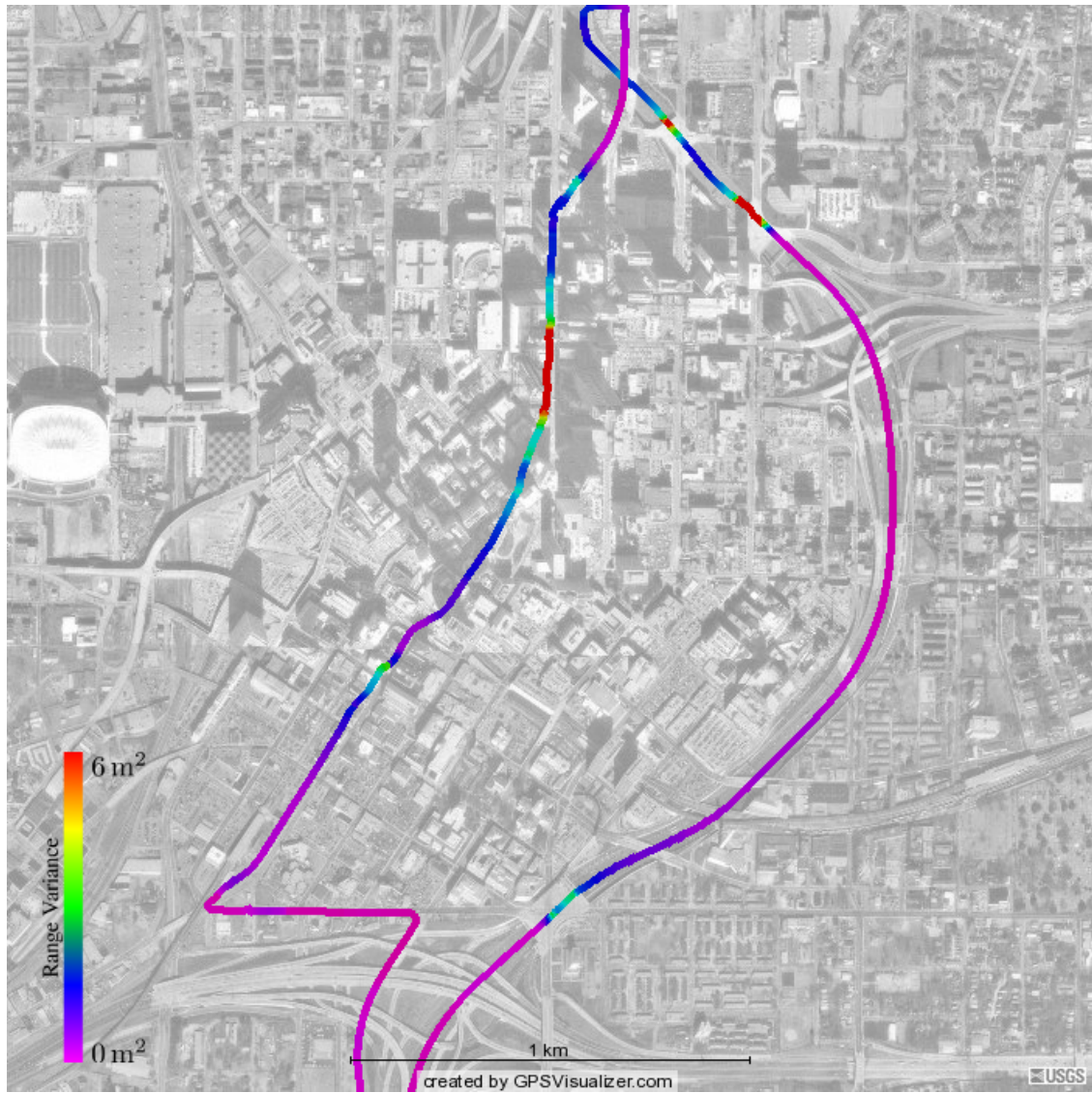


Figure 5.6: Range variance parameter along downtown Atlanta, GA route.

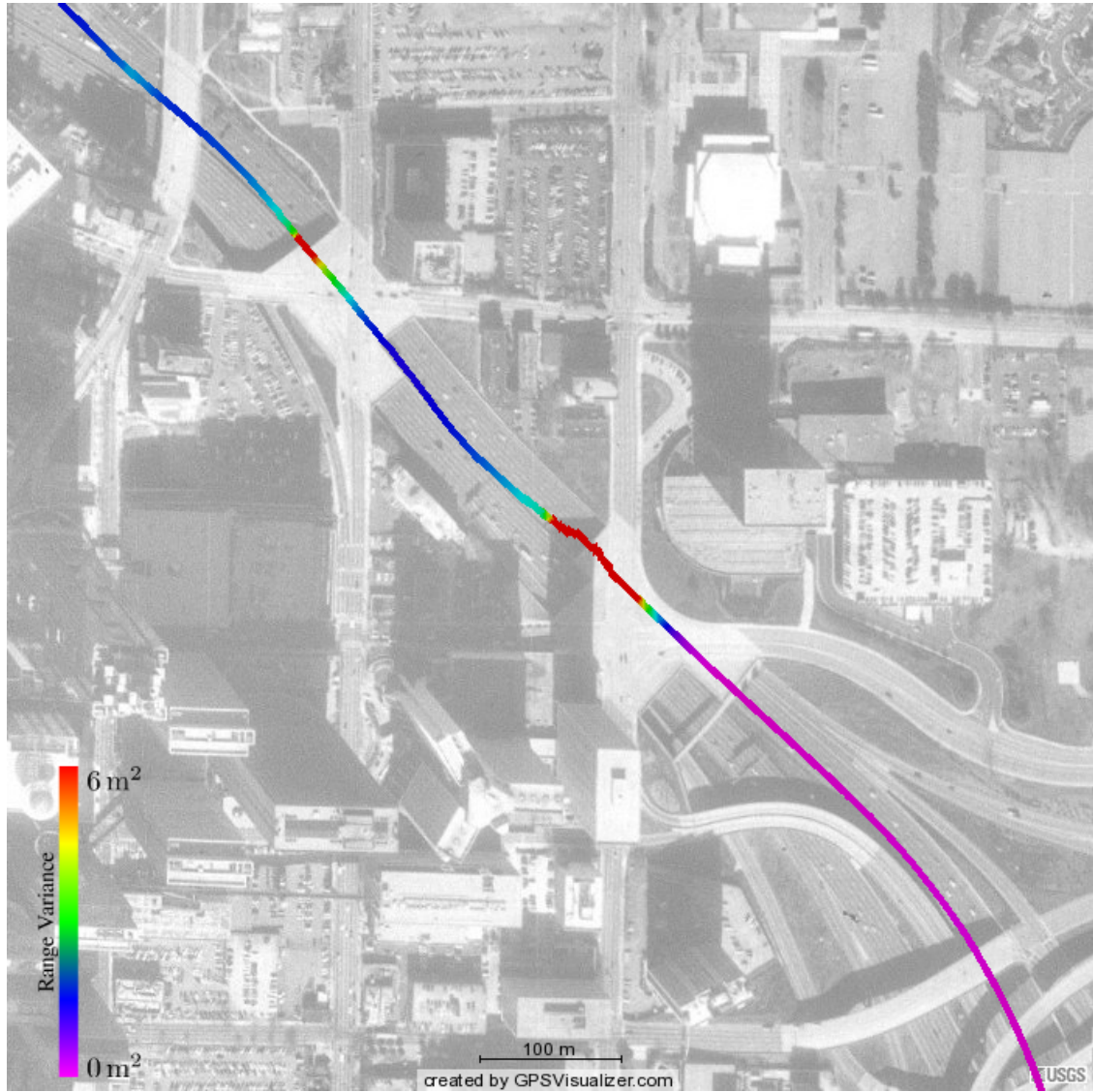


Figure 5.7: Range variance parameter beneath two overpasses in Atlanta, GA.

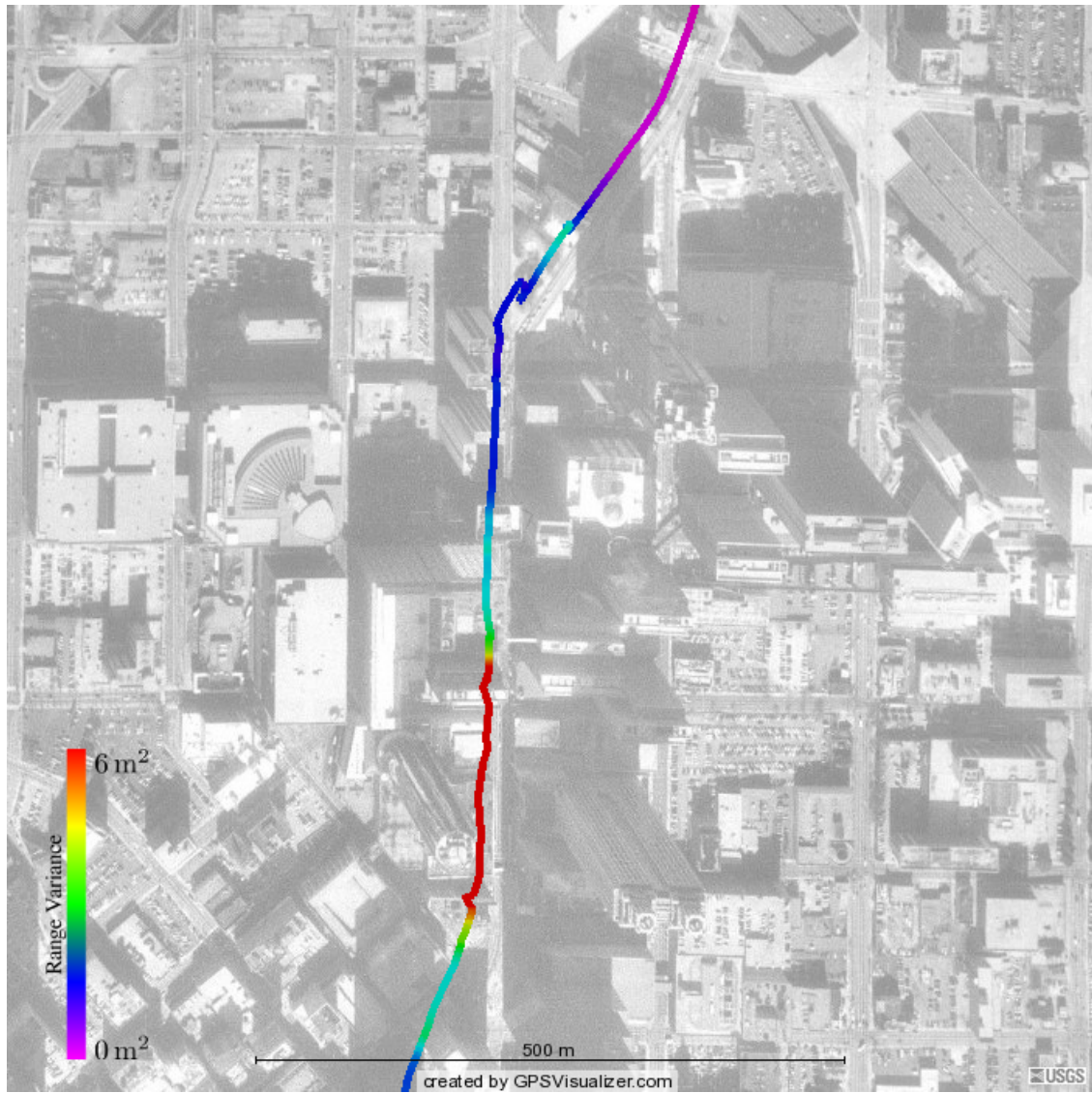


Figure 5.8: Range variance parameter in urban canyon in Atlanta, GA.

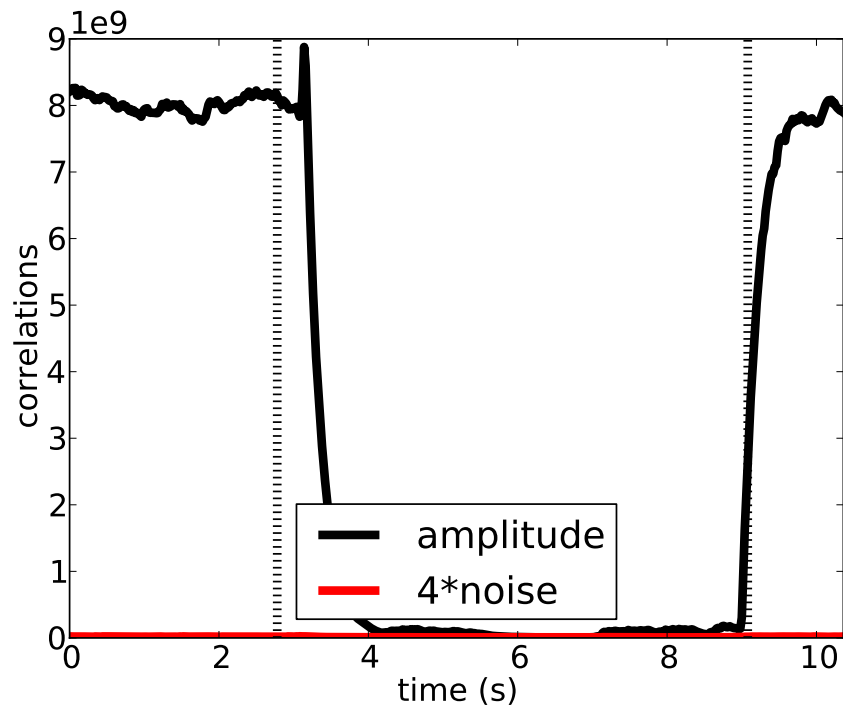


Figure 5.9: Amplitude estimate decay for satellite 31 when traveling beneath overpass.

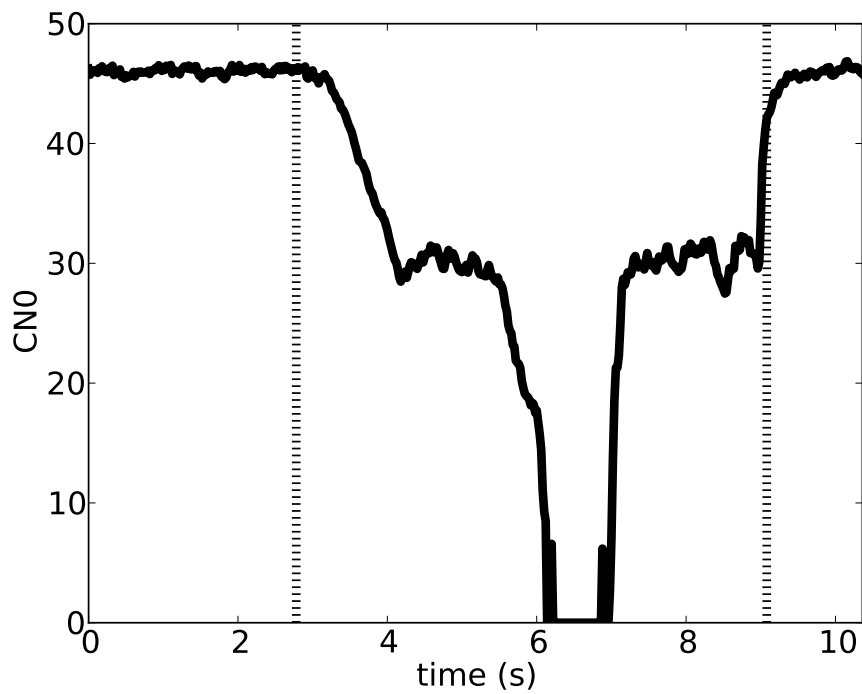


Figure 5.10: C/N_0 estimate decay for satellite 31 when traveling beneath overpass.

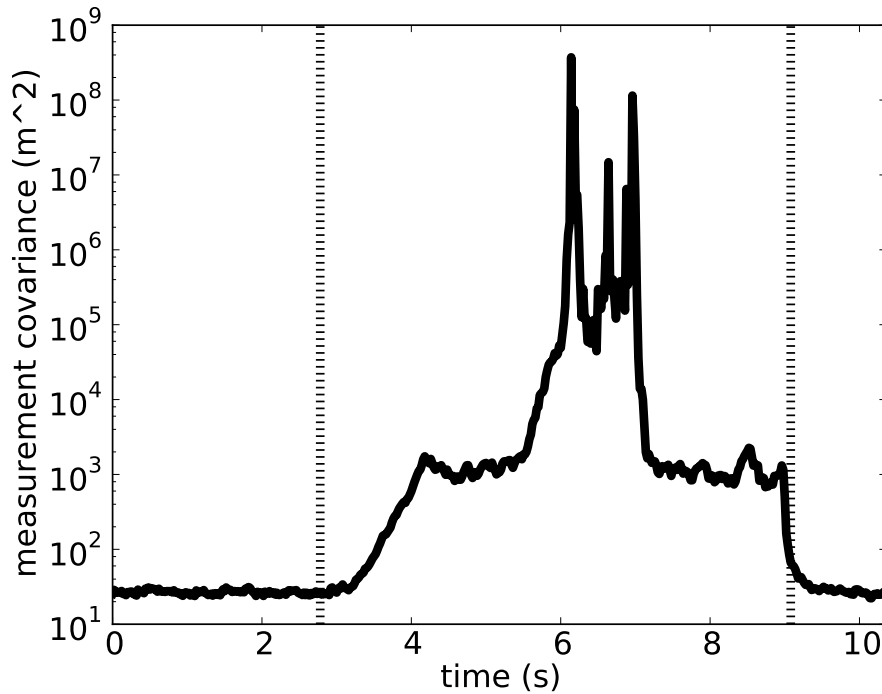


Figure 5.11: Measurement variance increase for satellite 31 when traveling beneath overpass.

can indicate this state by increasing until accurate measurements are again available. Using this mechanism the receiver can monitor when measurements are not providing accurate updates.

5.6 Summary

This chapter gave some theory and results for dealing with channel lock in a scalar or vector receiver. While scalar lock methods exist, they do not strictly apply to a vector architecture. Since a vector receiver is considered to be in permanent lock, other criteria were developed to increase solution integrity. First the robustness of the vector solution was demonstrated by injecting severe position and velocity errors. From this it was proven that a vector receiver can recover as long as the projected line of sight error remains within the discriminator bounds of four channels. Next a novel range variance parameter was introduced to serve as an indicator for vector precision. This parameter then was demonstrated in simulation and live sky experimentation.

Chapter 6

Vector Fault Detection and Exclusion

As a GPS receiver operates, there will be conditions in which the signals will not meet the assumptions made in the navigation algorithm. For instance, in a heavy foliage environment, multipath and attenuation errors can render the received signal biased from the direct line of sight range [15]. This bias contradicts the zero mean assumption of the pseudorange measurement. Therefore using this measurement can lead to unwanted, and sometimes unnecessary, error in the navigation solution. Detecting this fault is important because it can notify the GPS user that integrity has been degraded. Excluding the fault allows the receiver to remove the measurement from consideration and continue operating with measurements that better meet the filter assumptions. Details and demonstration of these methods are presented by the author in [15]. While they were originally presented in a closely coupled GPS/INS integration scheme, they provide a framework for extension into deeply integrated GPS/INS architectures, which will be described in Chapter 7.

In this section the normalized innovation method is applied to the centralized vector receiver. This approach allows the receiver to maintain solution integrity in the presence of signal errors. This new approach to a vector receiver will then be shown in live sky demonstrations.

6.1 Normalized Innovation

The state covariance represents the expected variances on the states. The measurement covariance represents the expected variances of the measurements due to signal attenuation and noise (which are functions of the C/N_0 and receiver parameters). A normalized innovation can be calculated as a test statistic for each measurement to determine whether or not the new information fits into the expected range of values. The calculation of each normalized innovation takes the form of

$$y_i = \frac{z_i}{\sqrt{C_{ii}}} \quad (6.1)$$

where $C = HPH^T + R$, which is calculated as part of the filter equations.

These normalized innovations should (under normal operating conditions) be normally distributed with zero mean and unit variance. Assuming the normalization makes the statistic unit variance, values that lie outside the threshold indicate that the innovations are non-zero mean and thus have errors that would bias the navigation solution. These measurements are to be excluded from the measurement update as shown in Figure 6.1.

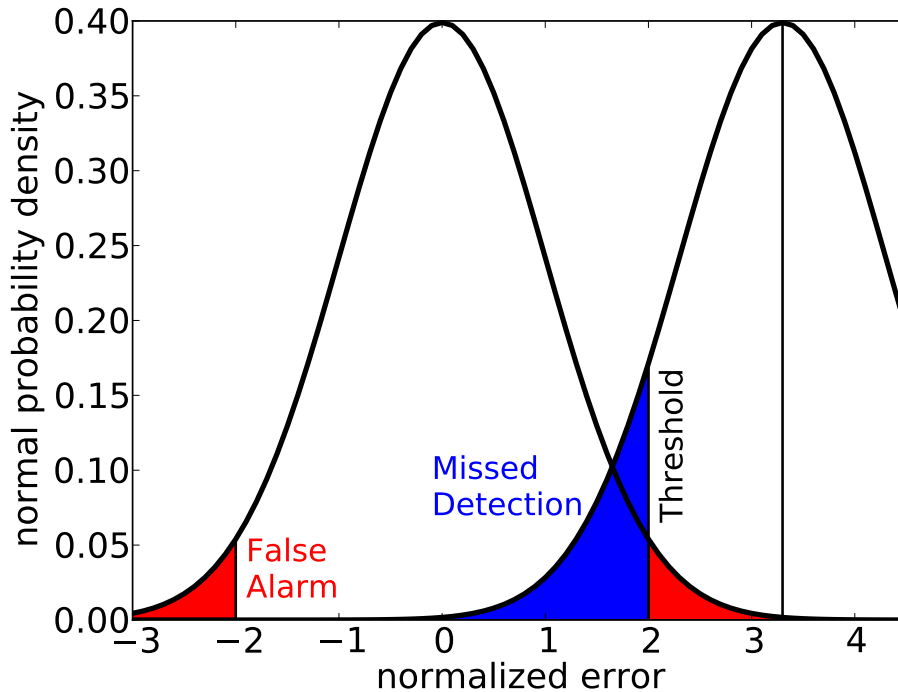


Figure 6.1: Normalized Innovation Density Function for Valid and Invalid Measurements

Selection of the threshold is determined by the probability of false alarm P_{FA} that is tolerable for the application. With a given P_{FA} , the threshold is selected so that the area under a zero mean unit variance Gaussian distribution from the threshold toward positive and negative infinity equals the P_{FA} . In the event that a pseudorange is rejected, the corresponding pseudorange rate measurement should also be rejected. However, a rejected rate measurement does not require the removal of the corresponding pseudorange [24]. This is seen from the pseudorange error measurement in Equation (3.14) and the pseudorange rate error measurement of Equation (3.16). The pseudorange rate is a function of the range error because the cross and dot products of Equation (3.16) both contain the range error term ϵ . However, the pseudorange error is not a function of the range rate

error. Therefore any detected range error leads to the rejection of both measurements while the range measurement is more robust in the presence of rate errors.

Using the measurements directly from the discriminators permits the vector receiver to catch errors before they corrupt the navigation solution. While most integrity monitoring techniques apply after the measurement update, the normalized innovation check detects before the measurements are applied. This allows the detection method to keep an accurate solution, avoiding the spreading of a measurement error into all channels. Such a scenario is illustrated in Figures 6.2 through 6.4. In Figure 6.2 a 10 m bias is injected into one satellite 10 seconds into the simulation. The range errors from the simulation are shown along with the 1 and 3 sigma bounds of the maximum range standard deviation described in Section 5.4. Here the bounds represent the expected range values given the current filter position uncertainty. In Figure 6.3 the same bias is injected but the filter is running a normalized innovation FDE. The effect of the range error is more delayed and less pronounced when using the FDE. Due to the probability of missed detection, the error can still enter the solution but its effects will be mitigated. The resulting position error is shown in Figure 6.4. The position error result after 10 seconds of being exposed to the biased measurement causes an error growth of more than half a meter in the receiver without FDE.

6.2 Theoretical Threshold Selection

When applying a normalized innovation detection method a threshold is selected based on the desired probability of false alarm, P_{FA} . When dealing with fault detection in this sense, a detection is made when an erroneous measurement is found. The null hypothesis of this situation is that the normalized innovations are zero mean unit variance Gaussian random variables. The alternative hypothesis is that the normalized innovations are not zero mean but may be unit variance Gaussian random variables. The probability of false alarm is the probability that a fault is detected when the null hypothesis stands. This probability is the likelihood that a correctly distributed measurement will be rejected. Since it is assumed that the normalized measurements are zero mean, unit variance Gaussian random variables, the error function, $\text{erf}(x)$, is used where

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (6.2)$$

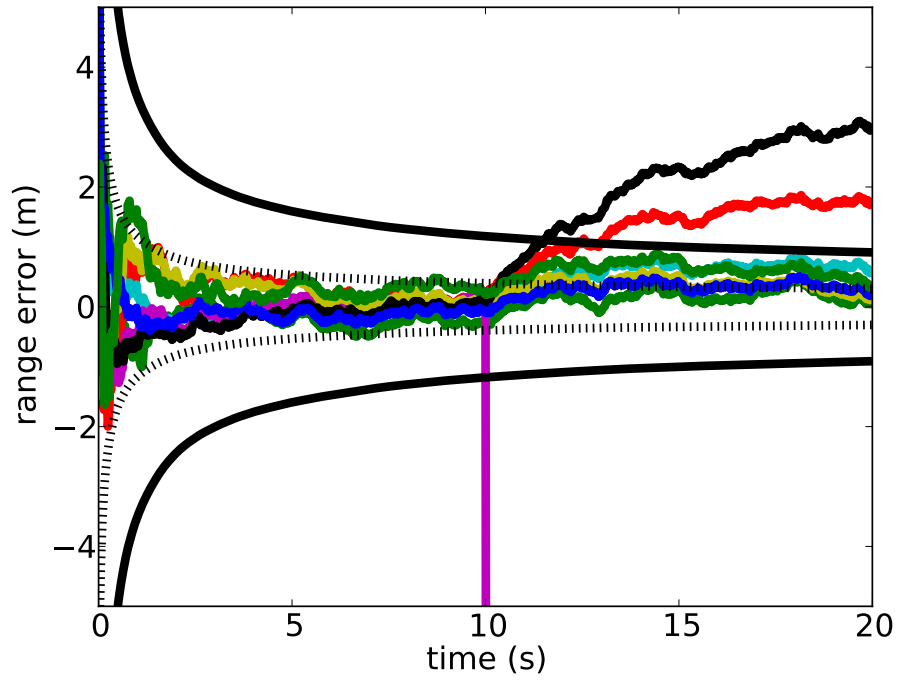


Figure 6.2: Range errors in simulation with 10 m bias and no FDE

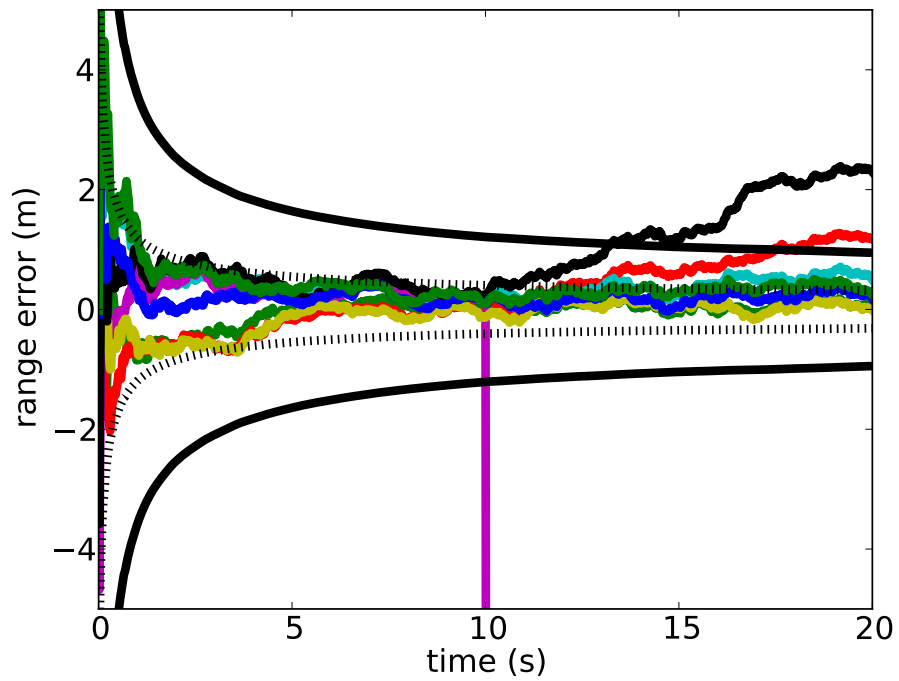


Figure 6.3: Range errors in simulation with 10 m bias and FDE

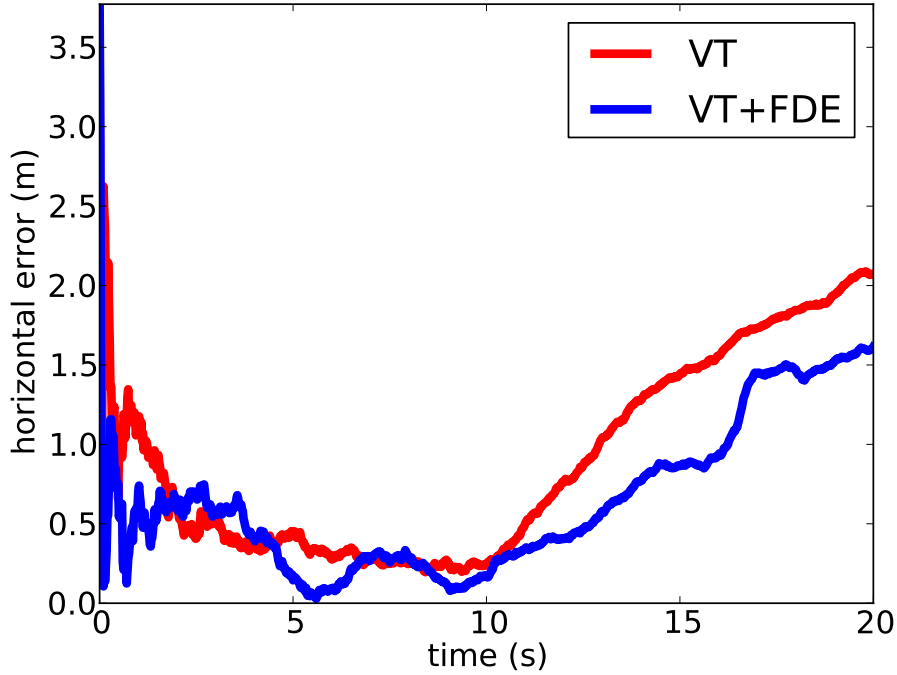


Figure 6.4: Position errors in 10 m bias simulation with and without FDE

and the complementary error function, erfc , is given as

$$\text{erfc}(x) = 1 - \text{erf} = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt \quad (6.3)$$

With these two definitions, it can be found that the probability of false alarm is based on the threshold choice, X , as

$$P_{FA} = \text{erfc}\left(\frac{X}{\sqrt{2}}\right) \quad (6.4)$$

There is no simple algebraic solution for this integral, so it is typically approximated or looked up from a table. For a probability of false alarm of $P_{FA} = 0.0025$, the resulting threshold is about $X = 3$, which is the value used in this work and in [13].

Another important statistic for the FDE method is the probability of missed detection, P_{MD} . This is based on the assumption that a faulty measurement (one the receiver should exclude) is a unit variance Gaussian random variable with non-zero mean. These biased measurements should be kept out of the position solution because they would bias the entire solution, including each tracking channel. Of course the actual P_{MD} is dependent on the value of the bias, B , but can be

found to be

$$P_{MD} = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{X - B}{\sqrt{2}} \right) \quad (6.5)$$

The greater the bias, B , the less likely a missed detection will occur. However, the larger the bias, the more important it is to keep the measurement out of the navigation solution. With the given threshold of $X = 3$ the probability of missed detection is shown below in Figure 6.5 for a range of pseudorange biases.

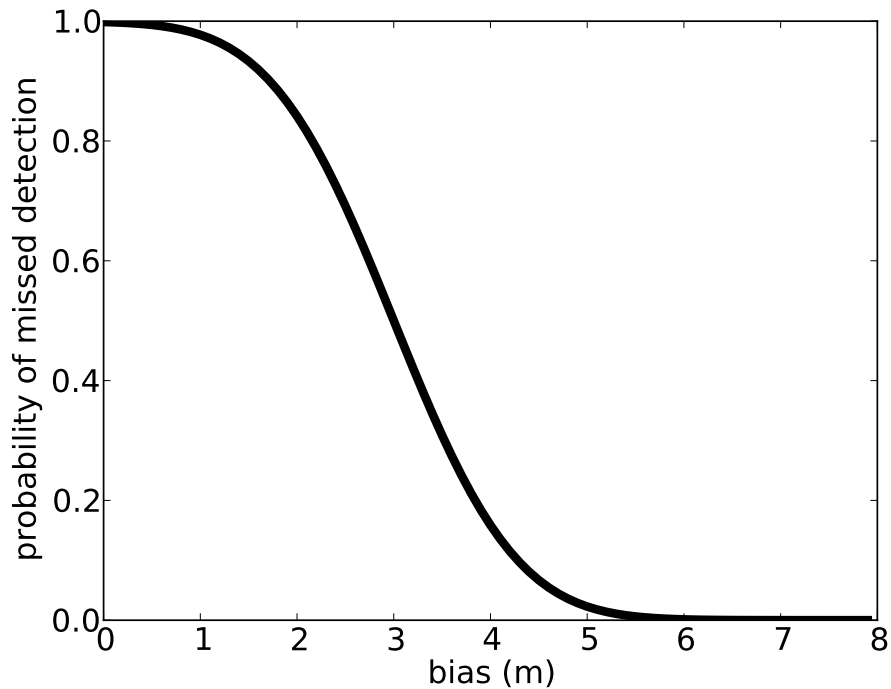


Figure 6.5: Theoretical probability of missed detection over a range of pseudorange biases given a threshold of 3.

6.3 Probability Validation

The normalized innovation test parameter is dependent on both the accuracy of the measurement and the accuracy of the covariance. Normalization of the measurement is assumed to strip the variance from the test statistic. However, the measurement error is made up of two parts: the noise in the measurement and the uncertainty in the state propagation. Both of these are accounted

for in the normalization factor C in Equation (6.1). The measurement noise is included from the measurement covariance, R , while the state uncertainty is included in the state covariance, P .

In steady state the state covariance is a balancing of the measurement covariance and the state propagation covariance. In vector tracking, however, the state propagation uncertainty is just a matrix of tuning parameters. It is used to model how the state propagation increases the state uncertainty. The difficulty in analytically solving for Q is handled by modeling its effect. In a vector system, it is populated by the receiver's maximum expected dynamics and clock quality. Recall Equation (3.5) where each position and velocity pair are affected by process noise of

$$Q = \begin{bmatrix} \sigma^2 \frac{T^3}{3} & \sigma^2 \frac{T^2}{2} \\ \sigma^2 \frac{T^2}{2} & \sigma^2 T \end{bmatrix} \quad (6.6)$$

where σ^2 is the velocity variance and T is the correlation time. At each state update Q increases the state covariance P . Changing σ^2 changes how much uncertainty is expected in the measurements when a new range and range rate are available. This is important because the normalization factor C is a function of the process uncertainty. Therefore it will affect the detection probabilities.

Figure 6.6 illustrates this in simulation. The horizontal axis is the velocity variance used by the receiver. The vertical axis is the probability of false alarm. From Section 6.2 the theoretical probability of false alarm for a threshold of 3 is around 0.0025. In this static scenario the probability of false alarm begins around 0.02 and increases to 0.1. Since the receiver is static, the actual velocity variance is zero. As Figure 6.6 shows, higher variances lead to worse performance.

The probability of missed detection can also be experimentally investigated, as shown in Figure 6.7. Now the horizontal axis is the pseudorange bias injected into the simulation. The satellite it is injected into is randomly selected at each Monte Carlo run. The vertical axis is the probability of missed detection. The same trend is seen here as in Figure 6.5 with the detection accuracy increasing with increasing bias. This is natural since there is less overlap of the distributions with higher biases.

These probabilities of false alarm and missed detection are important considerations in FDE threshold selection. Engineering judgment is required in the selection of the threshold and other

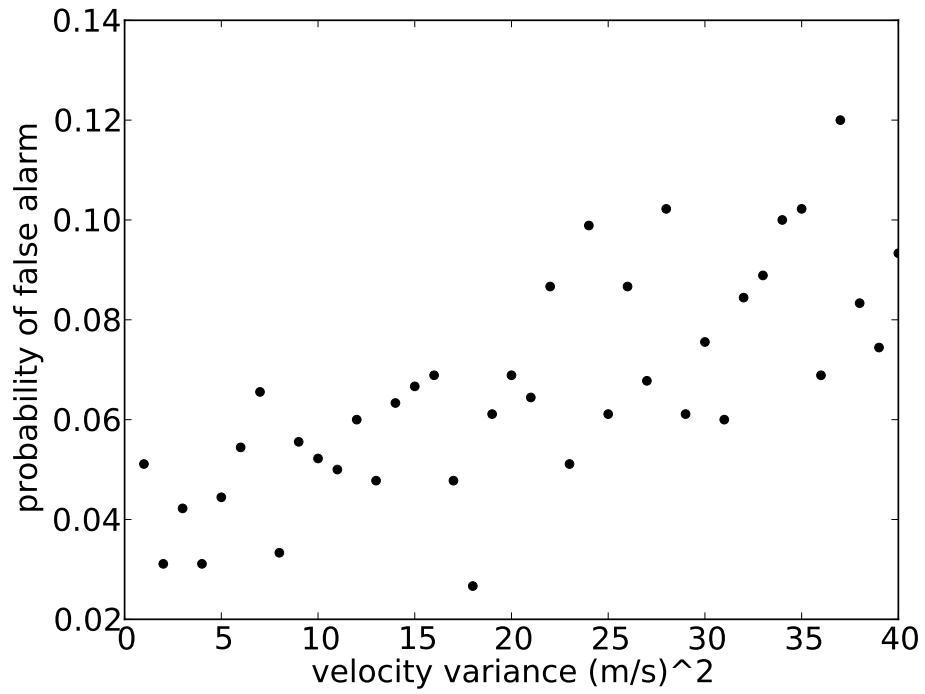


Figure 6.6: Simulated probability of false alarm versus velocity variance for a static receiver.

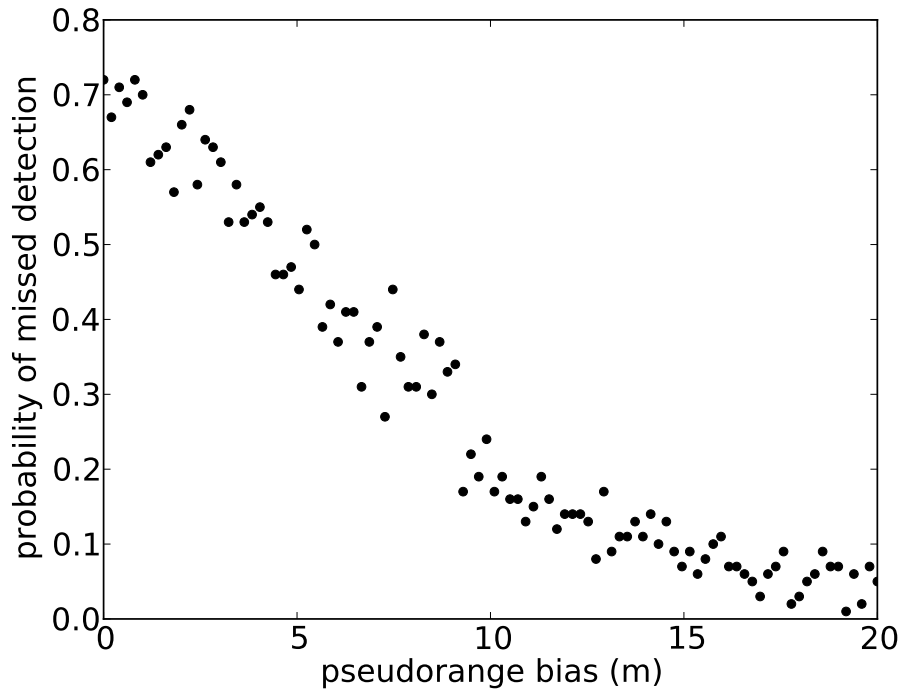


Figure 6.7: Simulated probability of missed detection versus pseudorange bias for a static receiver.

parameters, such as velocity variance. As with all things there are trade-offs involved in this FDE method. However, it plays an important part in assuring the navigation solution integrity.

6.4 Vector FDE Results

The results of FDE in a vector receiver are demonstrated in the sections that follow. In the first two, the data was taken on September 27, 2011 with the same equipment setup. These were done with the assistance of Scott Martin. In these scenarios a single NovAtel antenna was split and logged with a NordNav GPS front-end and a NovAtel Propak commercial receiver. The raw GPS results were logged to a laptop using NordNav's supplied logging tool. The NovAtel was logged on an in-car system developed by Auburn University's GPS and Vehicle Dynamics Laboratory (GAVLAB). All equipment used in the GAVLAB's G-35 test vehicle. The NovAtel was logged with time-tagging software along with the Crossbow 440 IMU. This synchronization was used in the DI system described in Chapter 7. The third set of results was taken in Atlanta, GA. Details of this data set were described in Section 5.5.

6.4.1 Open Sky in Auburn, AL

To demonstrate the use of FDE in a vector receiver, some live-sky data was taken along College Street in Auburn, AL. The overview of the route is shown in Figure 6.8. The background pictures for the results that follow were taken from openly available USGS aerial photograph databases.

As seen in Figure 6.8, there is little difference between the two receivers. This is due to the open nature of this portion of College Street. Along this stretch there is some light foliage and moderate commercial structures. However, most of the structures are several meters off the street. There is a portion of the run when the vehicle is stopped that the FDE keeps velocity errors from biasing the position. This is shown in Figure 6.9. The drift in position here is a little over one meter.

The portion shown in Figure 6.9 is while the test vehicle is stopped at a traffic light. The velocity magnitude drops to zero as shown in Figure 6.10. During this portion a series of faults is detected in satellite 28, whose range rate normalized innovations are shown in Figure 6.11. With

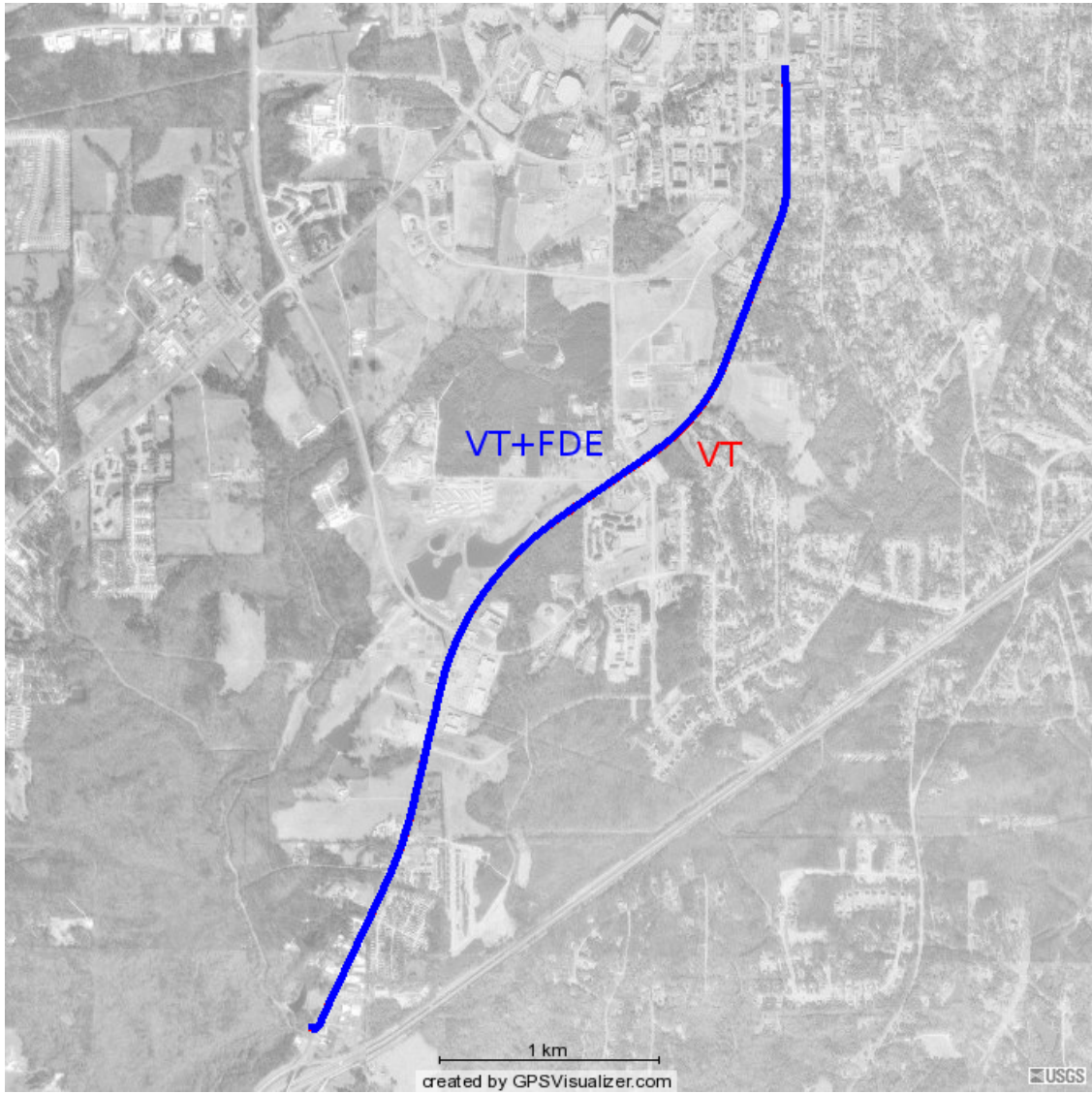


Figure 6.8: Vector receiver with and without FDE along College Street in Auburn, AL.

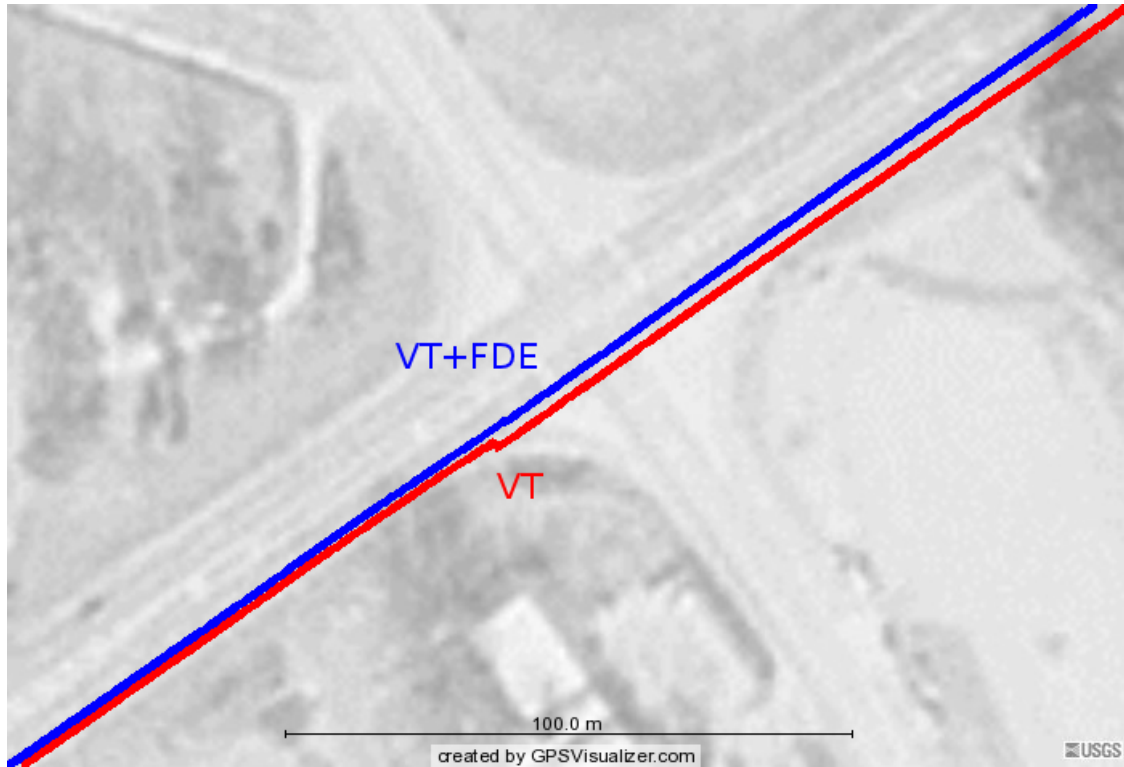


Figure 6.9: Portion of vector receiver with and without FDE along College Street.

these faults, the position vector drifted slightly more during the 12 second stop period, leading to the drift shown in Figure 6.9.

6.4.2 Moderately Blocked Sky in Auburn, AL

A second portion of data was collected primarily on College Street and Magnolia Avenue in Auburn, AL. This segment was chosen in proximity to more foliage and buildings. This was to determine the FDE method's effectiveness in these environments. The overview of the route is shown below in Figure 6.12. The path includes some heavy multipath areas in the South-East corner near The Hotel at Auburn. Having this segment at the initialization of the vector receivers affects their positioning performance severely.

Later in the data collection time an error is detected and rejected by the FDE as shown in Figure 6.13. This error is due to a range error as shown in Figure 6.14. This signal is from satellite 5, possibly induced from a multipath reflection. As described in Chapter 4, the multipath can

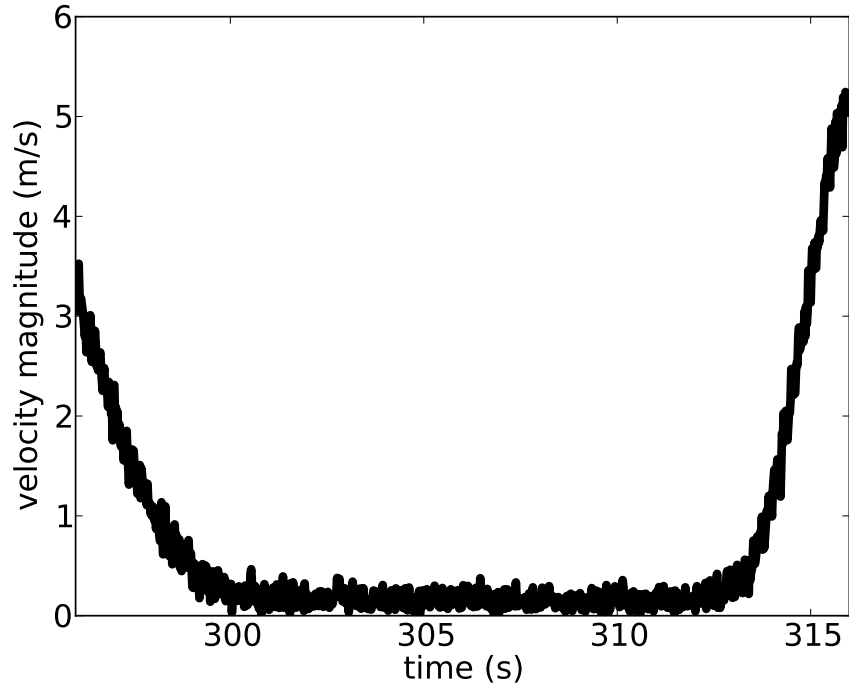


Figure 6.10: Velocity magnitude profile at static portion along College Street.

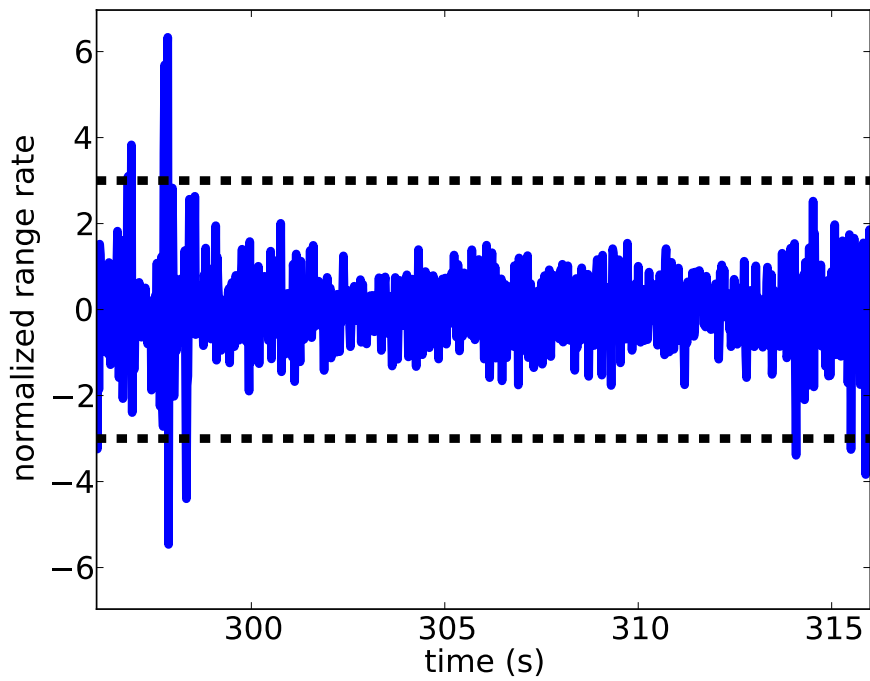


Figure 6.11: Range rate normalized innovations for satellite 28 at static portion along College Street.

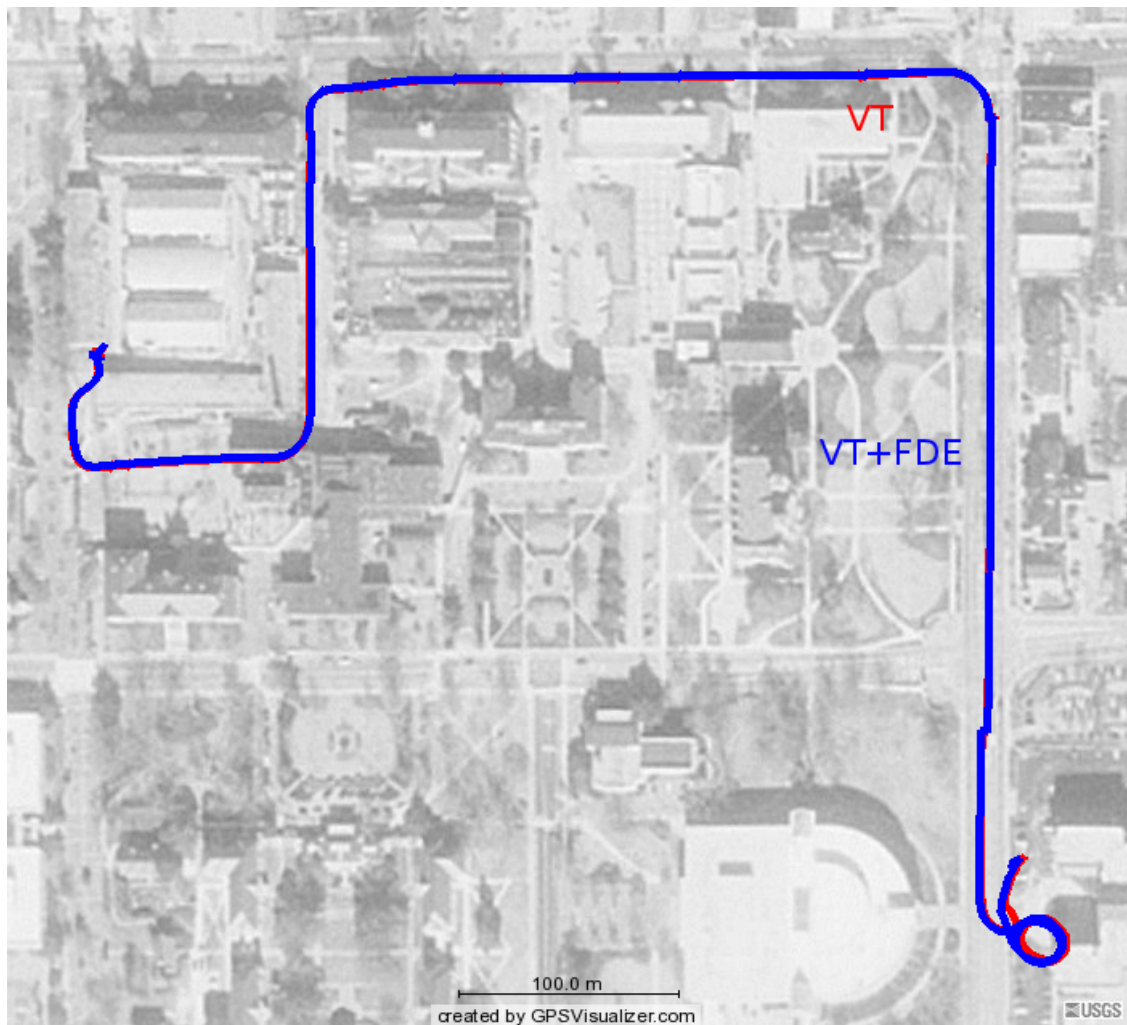


Figure 6.12: Vector receiver with and without FDE along College Street and Magnolia Avenue in Auburn, AL.

quickly change the direction of the delay based on the received phase. If the reflected signal is strong compared to the direct signal, this could have a large impact on the range.

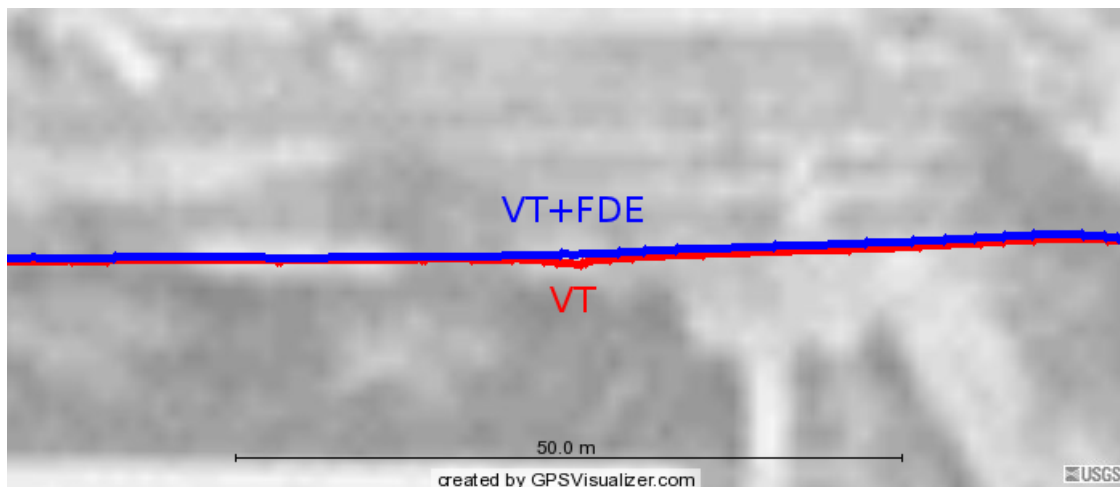


Figure 6.13: Portion of vector receiver with and without FDE along College Street and Magnolia Avenue in Auburn, AL.

6.4.3 Urban Canyon in Atlanta, GA

A third segment of data was used to validate the FDE method in a vector receiver. This data was provided from work in [33]. The overview of this path is shown in Figure 6.15. The over ten minute recording includes several roads in Atlanta, GA. The majority of the length is along Interstate 85. Its overpasses can severely degrade any GPS-based navigation solution. One particular junction underneath Baker Street and Piedmont Avenue is shown in Figure 6.16. While emerging from beneath these streets, the vector receiver sporadically begins to receive measurements as signals come into view. Without failure detection, the receiver makes corrections of more than 5 meters while the fault detection algorithm reduces the magnitudes of the jumps to around 2 meters.

While traveling under the overpass shown in Figure 6.16 there were several signal failures caught by the FDE. The normalized innovations for three of the more severe faults are shown in Figures 6.17 through 6.19. Figure 6.17 shows the range rate innovations for satellite 31. Figure 6.18 shows the range innovations for satellite 16. Figure 6.19 shows the range rate innovations for satellite 22. Each of these signals became visible and induced large measurement errors that were caught by the FDE. When signals are not available, the vector receiver state covariance increases, as

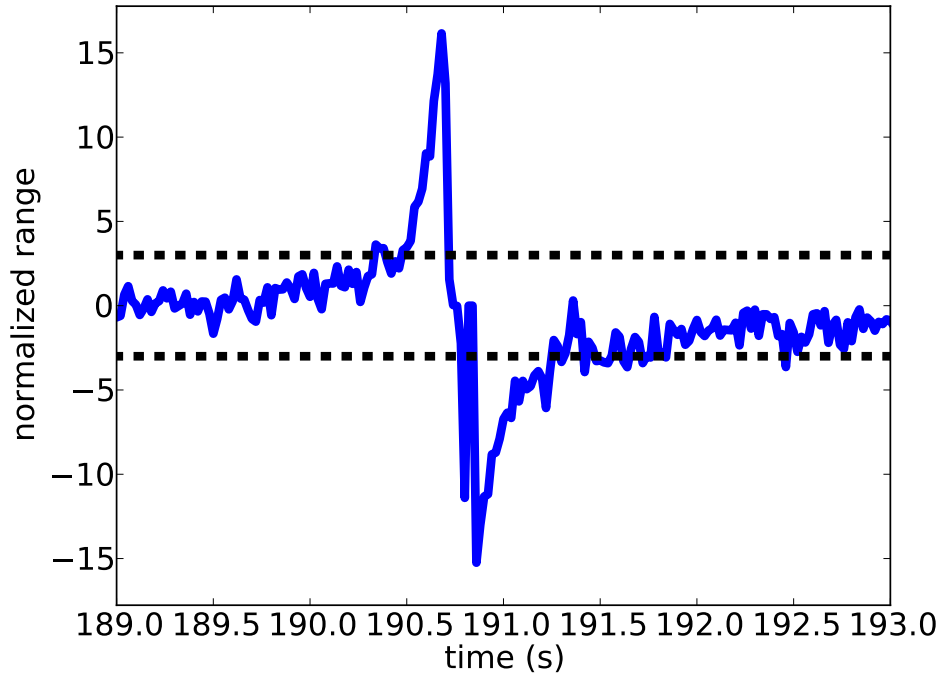


Figure 6.14: Range normalized innovations for satellite 5 at error along Magnolia Avenue.

shown in Section 5.5. This increases the range of permissible measurements such shown in Figures 6.17 through 6.19. However, at this point the covariance has not increased enough to accept these faulty measurements. Thus the receiver with FDE shows less variation after coming out from beneath the overpass.

Another improvement to notice is shown in Figures 6.17 and 6.19. There are range rate faults detected before the receiver goes completely under the overpass. Since there is complete blockage during this time, the errors propagate as initial conditions to the motion model. Beneath this overpass the vehicle is assumed to remain in a single lane. This is verified by the fact that both vector and vector with FDE receivers converge to the same lane after coming out of the overpass. Taking these convergence points as the beginning and end of a straight lane line, the positions in relation to this lane are shown in Figure 6.20. Here the FDE method has prevented errors in the solution just before signals are dropped. This leads to results that are closer to the assumed lane. This effect is similar to that shown in Section 6.4.1.

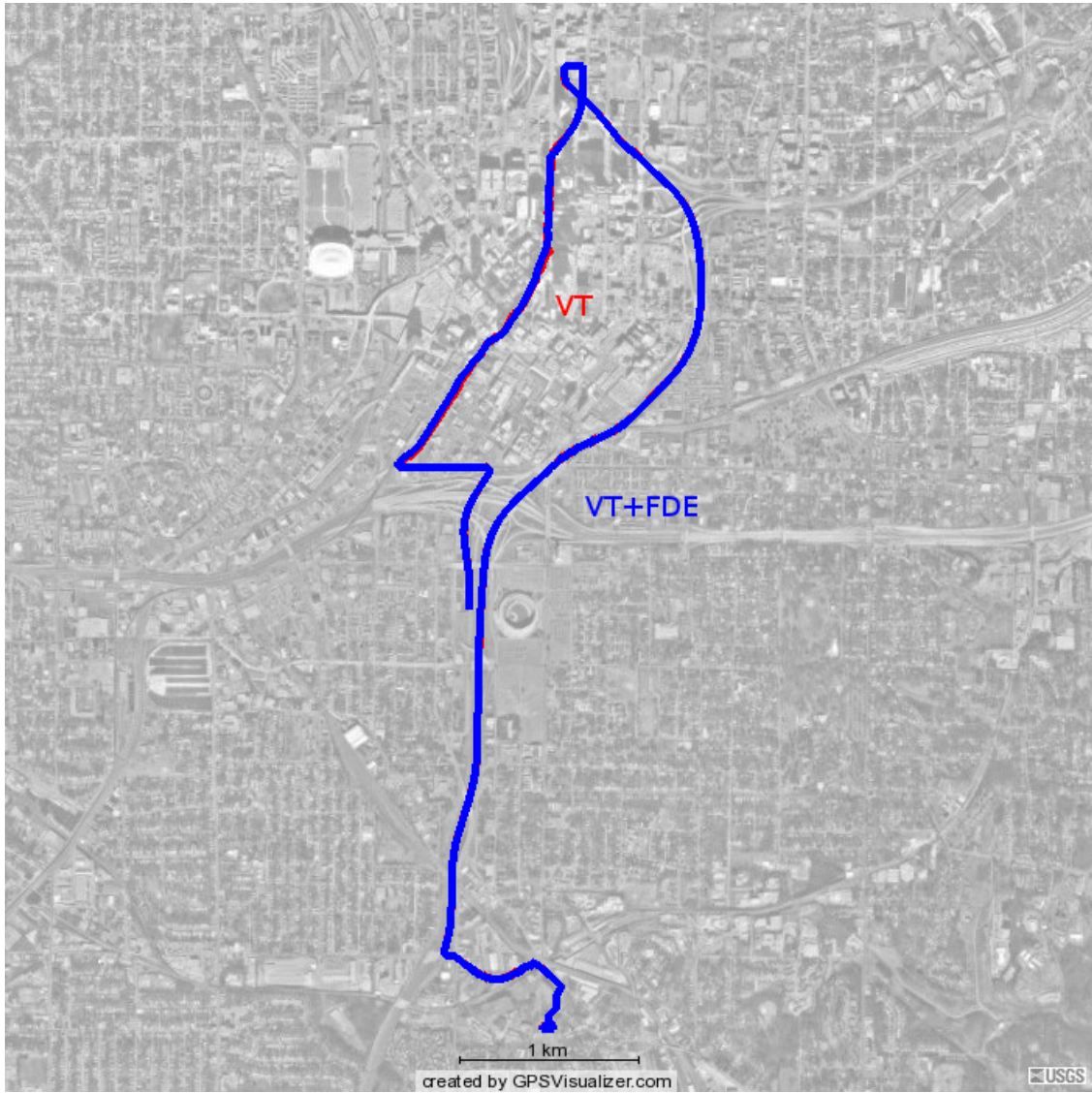


Figure 6.15: Vector receiver with and without FDE in Downtown Atlanta, GA.

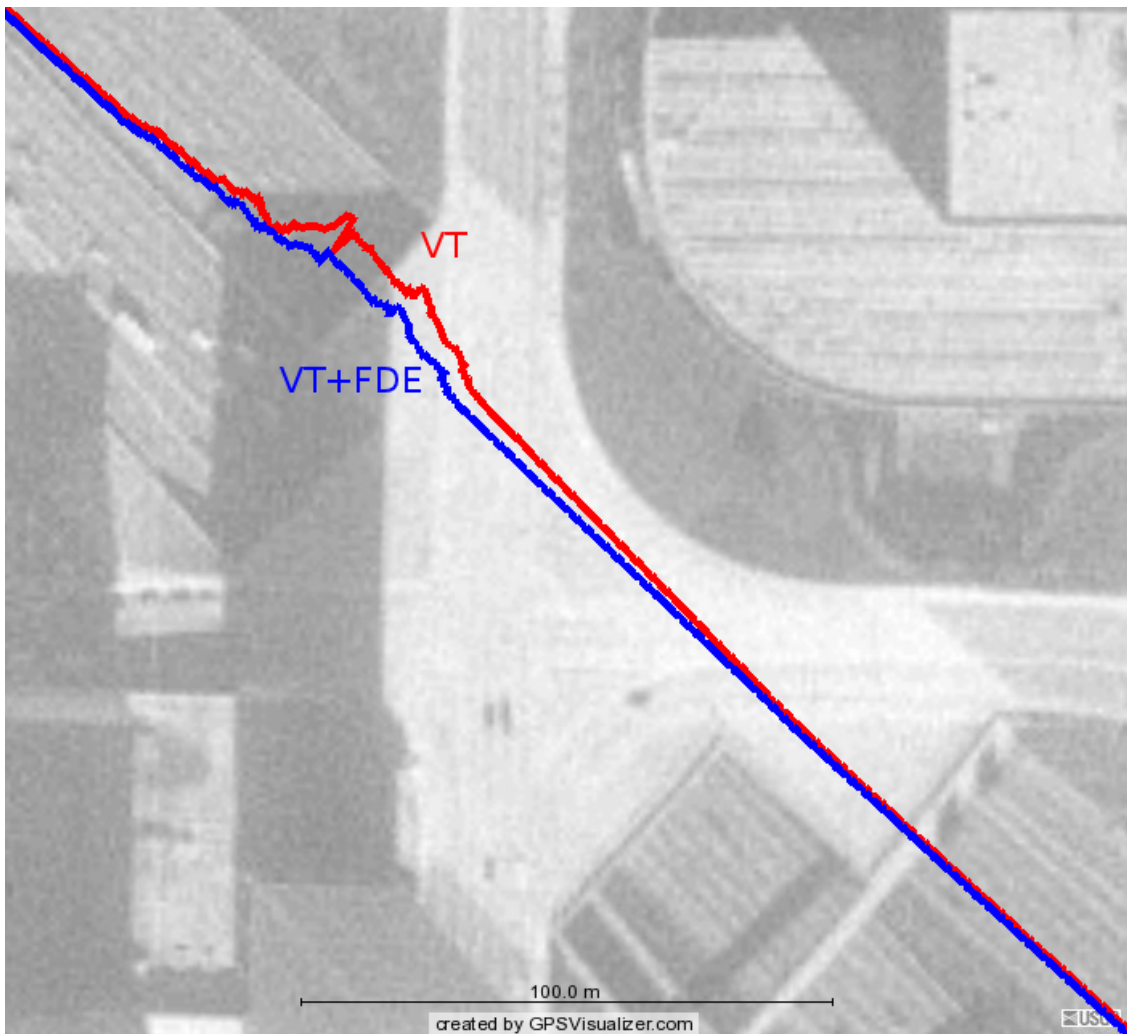


Figure 6.16: Portion of vector receiver with and without FDE beneath I-85 overpass.

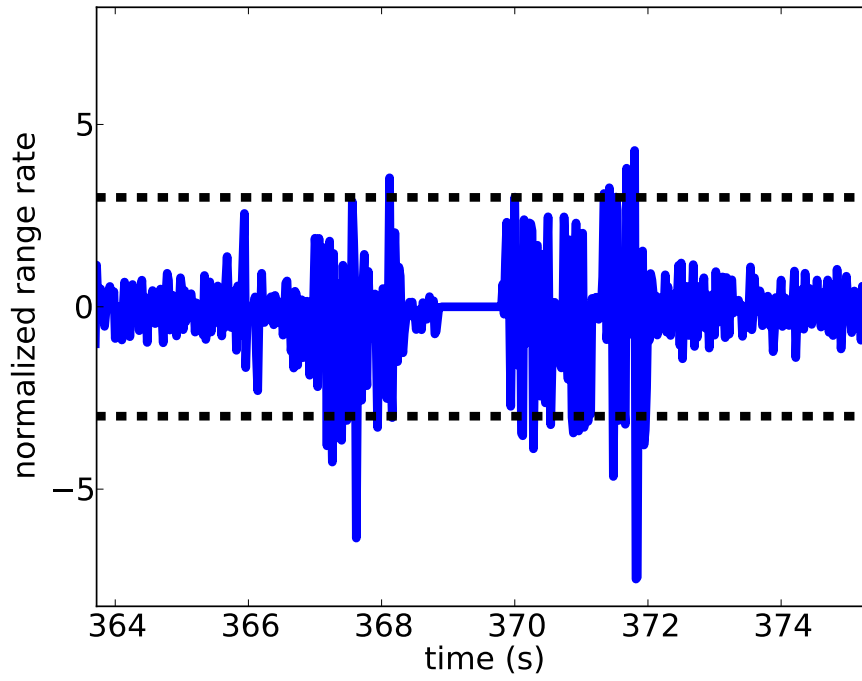


Figure 6.17: Range rate normalized innovations for satellite 31 beneath I-85 overpass.

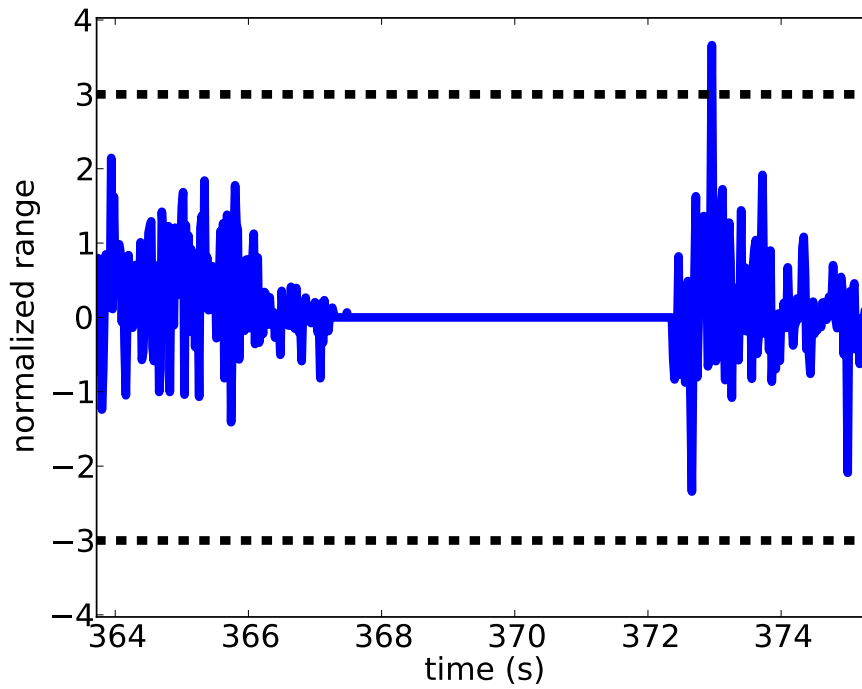


Figure 6.18: Range normalized innovations for satellite 16 beneath I-85 overpass.

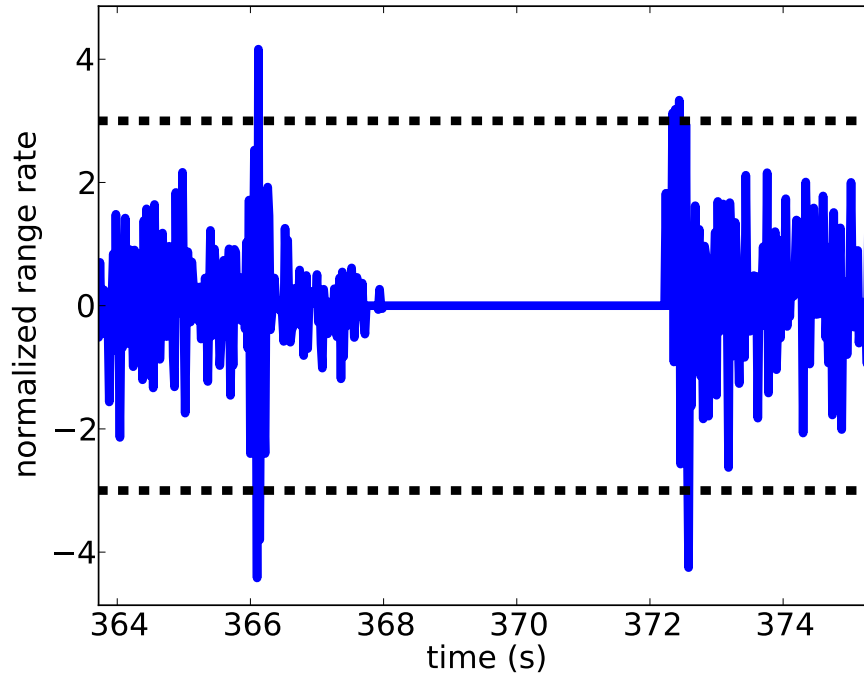


Figure 6.19: Range rate normalized innovations for satellite 22 beneath I-85 overpass.

6.5 Summary

In this chapter a novel fault detection and exclusion method in a centralized vector receiver is presented. The FDE algorithm's probabilities are examined from a theoretical and simulated basis. Finally the FDE method is demonstrated with live sky data. It results in reduced position drift of a meter and reduced position jumps of several meters.

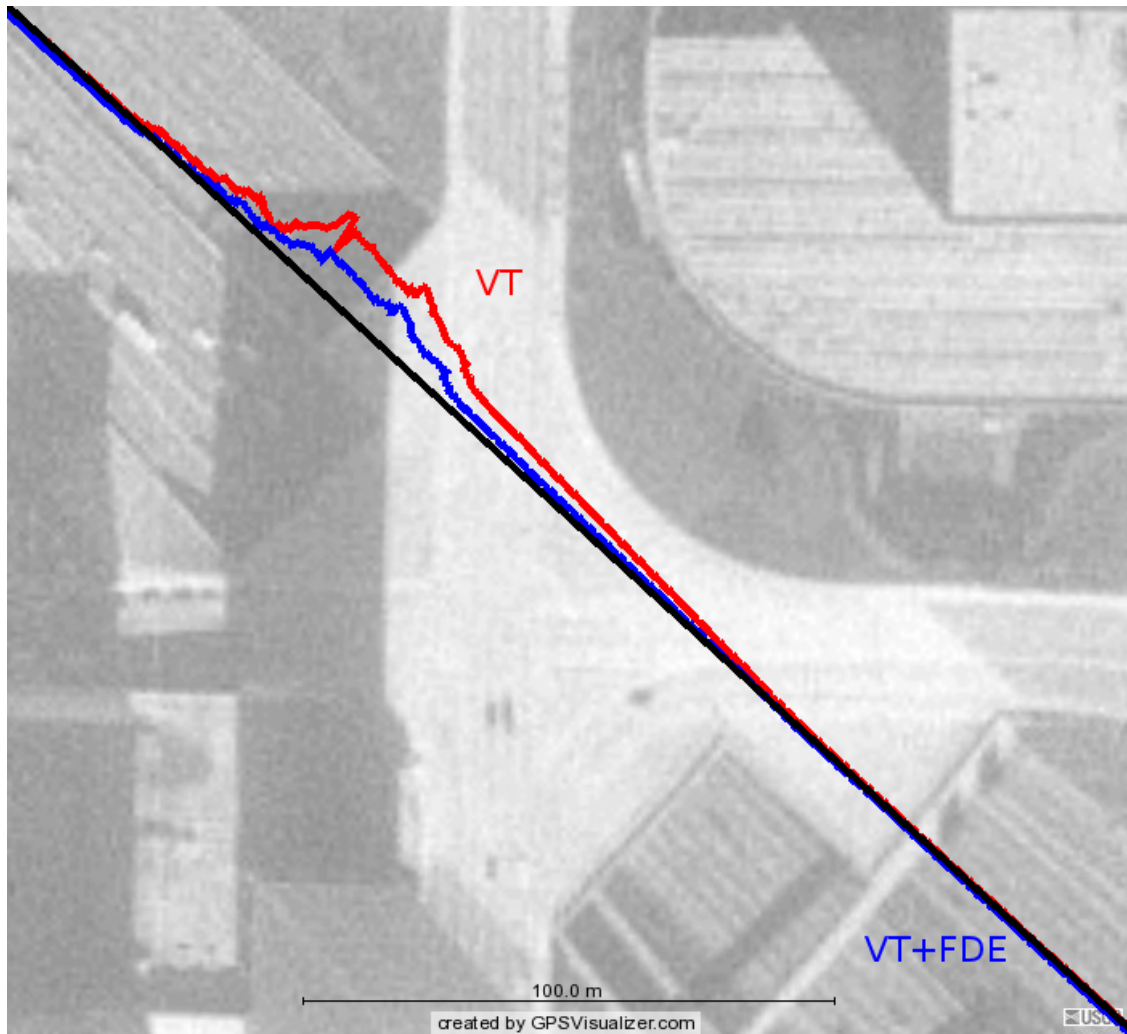


Figure 6.20: Portion of vector receiver with and without FDE beneath I-85 overpass with estimate of lane.

Chapter 7

Deeply Integrated GPS/INS

The final method for improving GPS operation in difficult environments is to stop relying on GPS only. Since such a ranging system is completely dependent on externally generated signals, there are many cases where it simply cannot operate alone. While signals are available, valuable range information is provided to the receiver through the tracking and measurement generation processes. However, any blocking or loss of these signals will severely degrade the use of the measurements. Short of not navigating blind in difficult environments, the system must be built to expect some loss of signal. When a vehicle is operating on roadways, these losses may occur only over short periods of time. In other cases, like mobile phone use, the losses could be over longer times while the receiver is indoors. Whatever the case, an alternative way to position would increase the availability of navigation information.

To overcome these difficulties, additional sensors are often coupled with GPS to provide navigation coverage in these situations. Several alternative sources of navigation have been proposed. In [46], GPS is combined with a map to complement GPS when blocked by buildings. In the case of a vehicle, being blocked by buildings naturally constrains the solution to be between the buildings. This makes a road map a good metric to consider. Radar systems have also been combined with GPS to provide ranging using both external and internal sources [20]. Camera systems are also integrated with GPS and other sensors in [12] to improve overall performance. Many other systems such as barometers and magnetometers are often coupled with GPS in a navigation system.

Of all the possible complements, perhaps the most frequently coupled system is an inertial navigation system (INS) [9], [21], [30], [31]. Even in most complete navigation solutions that include other sensors, an INS is usually included with GPS as in [12]. This comes from their complementary advantages and disadvantages. A GPS receiver has high stability in the position domain over the long term. An INS, however, has low variability in the short term but completely unbounded position error growth in the moderate to long term (depending on sensor quality). An

INS can have a very high update rate (> 100 Hz) that can maintain a solution with rapidly changing position while GPS receivers are typically limited to less than 50 Hz, and often as low as 1 Hz. Even down to their place in a filter formulation they are complimentary, with the INS capable of driving the state update while GPS performs the measurement update. With all these advantages, there have been many ways developed to combine these two navigation systems.

7.1 Published Architectures

Three common GPS/INS integration architectures are shown in Figure 7.1 [15]. These coupling methods all combine an INS with some independent GPS receiver. Even in the case of a full tightly coupled solution the aiding is not necessary for the navigation system to operate. However, none of these methods use a vector tracking receiver to improve the tracking based on positioning. Each tracking channel still tracks its assigned signal independent of other channels. The entire point of coupling with an INS is to improve positioning performance. With an INS, there are ways to make use of this position performance to also improve the tracking capabilities.

The coupling of a vector GPS receiver with an INS is called ultra tight coupling (UTC) or deeply integrated (DI) GPS/INS. In [30] a vector code tracking system is coupled with a PLL to generate velocity information. There the receiver is aided by an INS. In [25] a code tracking system is described using inertial aiding in a deeply integrated way. It focusses on the tracking performance in jamming from use of the inertial sensors and vector tracking of the GPS signals. The operation of the deeply integrated navigation system is attributed to gain adapting and covariance propagation being driven by measurements of the signal strength. It is also possible to couple an IMU to the carrier tracking, then couple the carrier to the code tracking, as in [17]. That method is called Advanced Tightly Coupled (ATC).

With increased computational capabilities, other methods are also being investigated. In [32] an architecture is described to open the tracking loops. Measurements are generated essentially in an acquisition-like fashion. Several code and frequency offsets are used to generate a field of measurements to which the correlation peak is mapped. This allows for accurate estimation of the true offset even if the prompt replica is slightly off. In traditional tracking loops, the accuracy of the measurement is only as good as the loop's current tracking error. While proper tracking loops

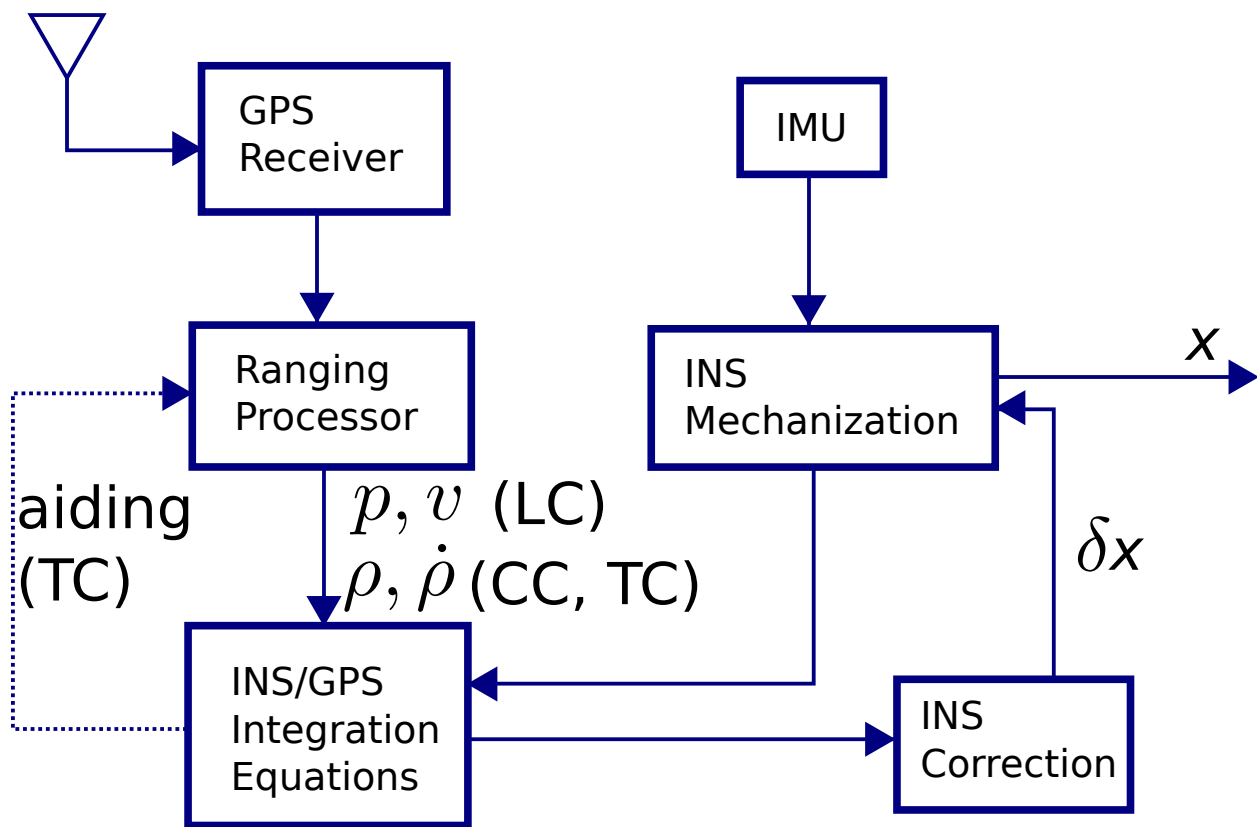


Figure 7.1: Diagram for loosely coupled (LC), closely coupled (CC), and tightly coupled (TC) GPS/INS architectures.

drive these errors to zero, this is an iterative process where erroneous measurements are generated along the way.

7.2 INS Mechanization Equations

Since the coupling algorithm corrects solutions generated by the inertial measurement unit (IMU), a navigation solution is given by these corrected measurements from the IMU mechanization equations. The IMU combined with its mechanization is called an inertial navigation system (INS). This system includes the raw IMU measurements and the technique used to convert these measures into navigation states. The IMU measurements of angular rates, $\omega_{ib}^b = \begin{bmatrix} g_x & g_y & g_z \end{bmatrix}^T$, and specific force, $f_{ib}^b = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T$, are used to drive these states as shown in Equations (7.2 - 7.5). To simplify notation, Ω_{ib}^b is the skew-symmetric form of the angular rate vector ω_{ib}^b such that

$$\Omega_{ib}^b = \begin{bmatrix} 0 & -g_z & g_y \\ g_z & 0 & -g_x \\ -g_y & g_x & 0 \end{bmatrix} \quad (7.1)$$

Taking Δt as the IMU measurement time step, the coordinate transformation matrix, C_b^e , from the body frame, b , to the ECEF frame, e , is propagated as in [24]

$$C_b^e = C_b^e \cdot \left(I_3 + \Omega_{ib}^b \Delta t \right) - (\Omega_{ie}^e \cdot C_b^e) \Delta t \quad (7.2)$$

The specific force vector is translated to the ECEF frame by

$$f_{ib}^e = C_b^e \cdot f_{ib}^b \quad (7.3)$$

With the acceleration due to gravity and earth rotation given as g_b^e , the velocity state is propagated as

$$v_{eb}^e = v_{eb}^e + (f_{ib}^e + g_b^e - 2\Omega_{ie}^e \cdot v_{eb}^e) \Delta t \quad (7.4)$$

This accounts for the specific force measured by the accelerometer, and the corrupting effects of the gravity vector and Coriolis acceleration.

The position vector can then be updated as

$$r_{eb}^e = r_{eb}^e + v_{eb}^e \Delta t \quad (7.5)$$

Simple Euler numerical integration schemes were chosen for ease of computation. Although the error due to this numerical approximation would tend to accumulate over long time periods, with GPS corrections this effect is reduced.

Equations (7.2 - 7.5) are processed whenever a new measurement is available from the IMU. As correction measurements (GPS observables) are available, the position (r_{eb}^e), velocity (v_{eb}^e), and coordinate transformation matrix (C_b^e) are updated to include these corrections. This correction process is detailed in subsequent sections. By periodically applying the corrections, the INS solution represents the system's best estimate of the antenna's navigation state at the current time. In the implementation described in Sections 7.3.1 and 7.3.2, this INS solution is used to linearize the state and measurement equations.

7.3 Deeply Integrated Filter

The deeply integrated GPS/INS filter operates in a similar fashion to the vector tracking filter described in Chapter 3. One of the main differences is that the filter state update is changed from a kinematic model to a measurement driven model. These measurements are from the inertial sensors that propagate in a nonlinear fashion as described in Section 7.2. The overall filter itself operates as shown in Figure 7.2.

In Figure 7.2, there are only a few modifications from the diagram in Figure 3.3 for the vector receiver. The major modification is that the state model internal to the navigation processor is replaced with the mechanized inertial navigation system (INS). Raw angular motion rates and specific force measurements are processed into a stand-alone navigation solution. Depending on the grade of the IMU, the resulting solution will drift from the true position and velocity. This is exacerbated by the fact that the errors on the IMU are dominated by a bias. Integration of these bias errors leads to gross velocity, position, and heading errors. In order to maintain the

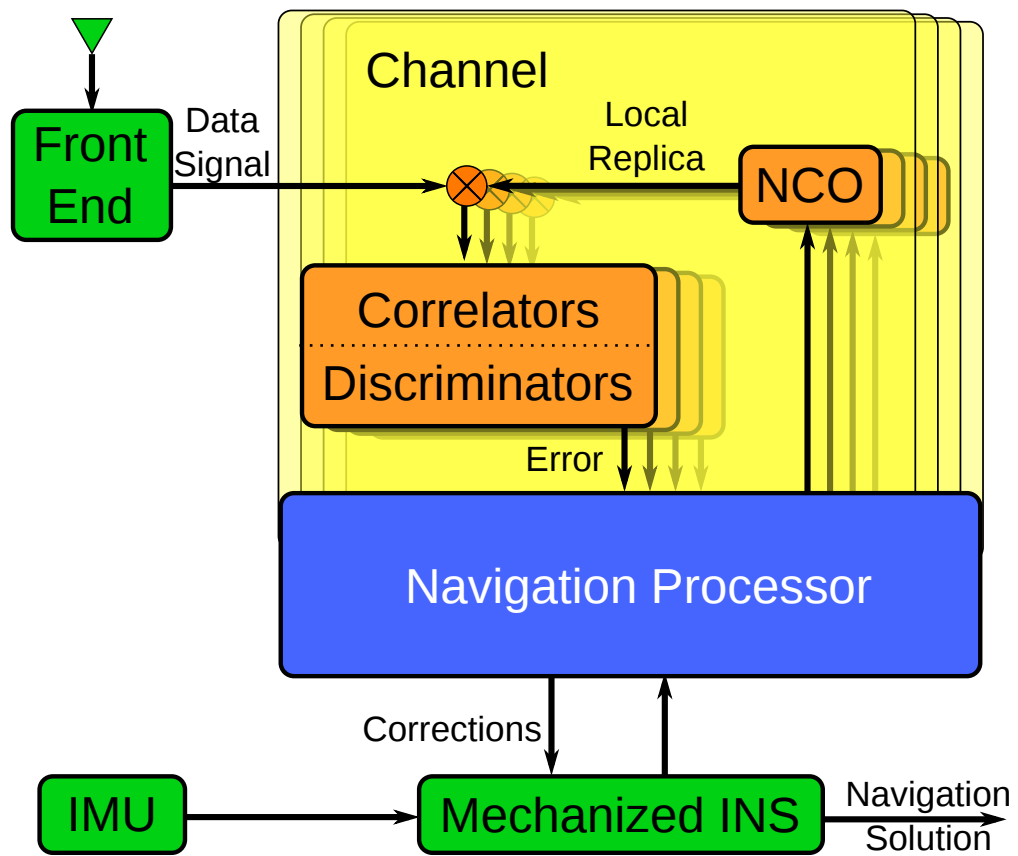


Figure 7.2: Deeply Integrated GPS/INS diagram.

solution integrity, corrections can be applied from the navigation processor, driven by the GPS measurements. In this case, the solution is maintained by the INS which is available at a higher rate than the PVT of the vector receiver.

7.3.1 State Dynamics

The GPS/INS integration scheme includes 17 states used to keep track of the vehicle's motion, shown in Equation (7.6).

$$x = \begin{bmatrix} \delta\psi_{eb}^e \\ \delta v_{eb}^e \\ \delta r_{eb}^e \\ b_a \\ b_g \\ cdt_u \\ c\dot{d}t_u \end{bmatrix} \quad (7.6)$$

These states are partitioned into six groupings of similar variables. The state vector includes three components of attitude error, ψ_{eb}^e , which are used to update the coordinate transformation matrix from the body to the ECEF frame, C_b^e ; three components of velocity error in the ECEF frame, v_{eb}^e ; and three components of position error in the ECEF frame, r_{eb}^e . The three accelerometer biases, b_a , and three gyro biases, b_g , are included and modeled as constants with process noise. The receiver clock bias, cdt_u , and receiver clock drift, $c\dot{d}t_u$, are included with the clock drift modeled as a constant.

Since the INS provides a set of nonlinear dynamic equations to propagate the state vector in Equation (7.6) and the GPS measurements are nonlinear functions of these states (to be described in Section 7.3.2), an Extended Kalman Filter (EKF) can be used in the integration scheme [24]. In order to use the coupling architecture in an EKF form, the equations given in Section 7.2 are linearized for propagation of the state errors (δx) and its error covariance matrix (P). The state vector includes errors of the attitude, velocity, and position states but keeps track of the actual

IMU bias and receiver clock states rather than their errors. The form of the state equation is

$$\dot{\delta x} = F\delta x + w_s \quad (7.7)$$

where

$$F = \begin{bmatrix} -\Omega_{ie}^e & O_3 & O_3 & O_3 & C_b^e & O_2 \\ F_{21} & -2\Omega_{ie}^e & F_{23} & C_b^e & O_3 & O_2 \\ O_3 & I_3 & O_3 & O_3 & O_3 & O_2 \\ O_3 & O_3 & O_3 & O_3 & O_3 & O_2 \\ O_3 & O_3 & O_3 & O_3 & O_3 & O_2 \\ O_2^T & O_2^T & O_2^T & O_2^T & O_2^T & F_{66} \end{bmatrix} \quad (7.8)$$

$$F_{21} = \begin{bmatrix} 0 & -f_z & f_y \\ f_z & 0 & -f_x \\ -f_y & f_x & 0 \end{bmatrix} \quad (7.9)$$

$$F_{23} = \frac{2g_0}{r_{en}^{e2}} \frac{r_{eb}^e r_{eb}^{eT}}{|r_{eb}^e|} \quad (7.10)$$

$$F_{66} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (7.11)$$

where I_m represents an identity matrix of size $m \times m$, O_3 represents a 3×3 null matrix, O_2 is a 3×2 null matrix, and w_s is the process noise included in the system derivation.

The state transition matrix at the current time step, k , can be approximated as

$$\Phi_k = I_{17} + F\Delta t \quad (7.12)$$

This allows for the propagation of the state correction as

$$\delta x_k^- = \Phi_k \delta x_{k-1}^+ \quad (7.13)$$

With the state transition matrix, the covariance matrix can be updated to

$$P_k^- = \Phi_k P_{k-1}^+ \Phi_k^T + Q \quad (7.14)$$

Here Q is the system noise covariance matrix which is assumed to be a diagonal matrix with entries given in Table 7.1. These values were tuned by trial and error to achieve the desired performance.

Table 7.1: System Noise Covariance Matrix Values

State	Value	Units
Attitude Errors	1e-6	(rad/s) ²
Velocity Errors	1e-5	(m/s) ²
Position Errors	1e-4	(m) ²
Accelerometer Bias Errors	1e-4	(m/s ²) ²
Gyro Bias Errors	1e-4	(rad/s) ²
Clock Bias Error	1e-4	(m) ²
Clock Drift Error	1e-5	(m/s) ²

7.3.2 State Measurement Relations

The Deeply Integrated GPS/INS system applies measurement updates similar to a vector receiver, described in Section 3.2. At the measurement update time, the best estimate of the receiver position, velocity, and clock states is used to determine the local replica as in vector tracking. Since the states in Equation (7.6) are altered from the vector receiver, the measurement update is also changed to

$$\bar{y} = C\bar{x} + \bar{v}$$

$$C = \begin{bmatrix} 0_3 & 0_3 & a_1 & 0_3 & 0_3 & 1 & 0 \\ 0_3 & a_1 & 0_3 & 0_3 & 0_3 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0_3 & 0_3 & a_N & 0_3 & 0_3 & 1 & 0 \\ 0_3 & a_N & 0_3 & 0_3 & 0_3 & 0 & 1 \end{bmatrix} \quad (7.15)$$

7.4 Time Synchronization

Proper alignment of the two sets of sensors (GPS receiver and INS) will affect performance. However, time synchronization is typically done at a hardware level with some amount of synchronization with the GPS provided Pulse Per Second (PPS). In a hardware receiver the PPS signal is generated to pulse on the whole second of GPS time (after positioning and clock bias has been estimated). For different cases, different time synchronization specifications must be met [21]. In [40], a metric is given for this accuracy. This metric is provided as a rule of thumb but is very useful in specifying system requirements. For proper use, the synchronization errors should be an order of magnitude below the measurement noise. That is,

$$v_{max}dt_{sync} = \frac{d\rho}{10} \quad (7.16)$$

Here v_{max} is the maximum expected velocity of the vehicle, dt_{sync} is the accuracy level of the time synchronization, and $d\rho$ is the measurement accuracy.

Therefore, when using pseudorange level measurements with accuracy on the order of 5 m and maximum velocity of 1 m/s, the time synchronization would need to be 0.5 seconds. Alternatively, when using carrier phase measurements with accuracy of 0.02 m then the time synchronization would need to be 0.002 seconds. This time requirement is much more stringent than the pseudorange level requirements. In this work only the pseudoranges are used therefore the alignment was able to be done post-process by aligning software receiver derived paths with concurrently logged NovAtel and Crossbow IMU paths.

7.5 Fault Detection and Exclusion

In order to increase the robustness of the navigation solution, fault detection and exclusion is included in the algorithm. Very little has been reported in the field of integrity monitoring for DI GPS/INS systems. In [3], a RAIM-like algorithm is described for vector tracking. Since it applies only to measurements, there is no improvement from adding an INS as far as the algorithm goes. The method used in [3] comes from the standard RAIM algorithm and pulls the test statistic from before the measurements have been applied to the solution. This is done by forming a test statistic

from the correlator outputs rather than the residuals after the measurement update. If successful, the solution would not be corrupted by the error and can continue operation. This is similar to the point at which normalized innovation detection occurs [14].

A normalized innovation detection method is applied to phase estimators in [17]. However, this detection is only applied to the sub-filters of a federated architecture from L1 and L2 measurements. Another approach to monitoring a vector receiver is described in [36]. However, that work deals with context detection to detect what environment the receiver is operating in. Receiver operation is altered after the context has been detected.

The method used here is equivalent to that presented in Chapter 5. This is possible because the DI algorithm uses the same measurements as the vector receiver. However, there is a difference in that FDE in a DI navigation system has a better model of the motion and thus can more accurately filter faulty measurements. In the vector instance, the motion was set as a constant velocity model whereas the INS gives a higher fidelity view of the state update. Of course the INS is corrupted by noise and bias errors of its own but the short term accuracy of the solution aids the navigation system.

7.6 DI Results

In order to demonstrate a working DI solution, raw GPS IF data was recorded alongside a MEMS IMU, the Crossbow 440. The synchronization of the IMU was accomplished through concurrently logging RTK GPS from a NovAtel receiver in an asynchronous way. With the absolute time provided by the RTK, the IMU could be aligned in post-process. This data was taken as part of the experiments run in Chapter 4. As part of this experiment, portions of the data run are intentionally close to large sources of multipath and signal blockage. While the low grade of the IMU did not generate a significant improvement in performance, the inclusion of the FDE algorithm further improved the results. The positioning results are shown below in Figure 7.3.

Similar to Section 6.4.2, which used the same data, there is a fault just after turning onto Magnolia Avenue. The position during this time is shown in Figure 7.4. Again there is a range error induced as shown in Figure 7.5. The effect of this error is slightly higher with the DI solution



Figure 7.3: DI and DI with FDE positioning solution along semi-urban environment.

without FDE as compared to the vector tracking solution without FDE. Here the lateral change in position is over one meter.

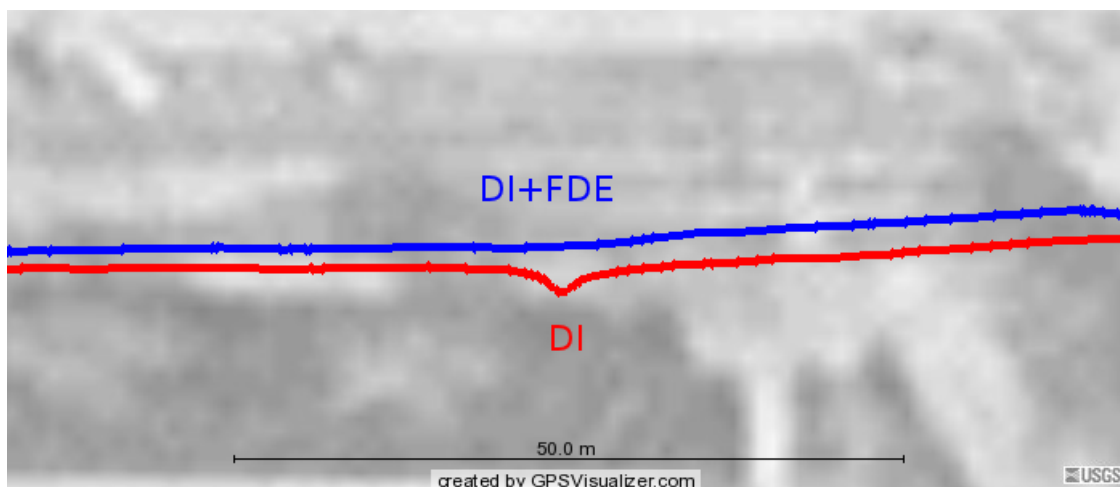


Figure 7.4: DI and DI with FDE positioning solution along Magnolia Avenue.

These results show the benefit of including this FDE method into the DI solution in the same way as in the vector receiver solution. Operating without the FDE leads to erroneous shifts in the position that may undermine the receiver’s ability to track. While the solution is typically robust in the face of errors by virtue of being a vector solution, inclusion of the FDE method is here shown to further improve results.

7.6.1 Vector Tracking or Deeply Integrated

Since both the vector receiver and the deeply integrated GPS/INS architectures were demonstrated with the same data, there are other comparisons to be made. Below in Figure 7.6, the maximum range variance parameter is shown for both receivers. As this figure indicates, the simple constant velocity model of the vector receiver (blue) has a lower maximum range variance than the DI solution (red). This is partly due to the automotive grade of the IMU being used. Although it provides more information about the receiver motion, it also brings bias errors and noise that must be accounted for. Therefore the resulting variances are actually higher.

In the case of signal outage, the DI solution will also increase the position variance of the solution more quickly. This is shown in Figure 7.7 where the vector receiver and deeply integrated solution are initialized to the same equivalent covariance. The lines shown are the square root of

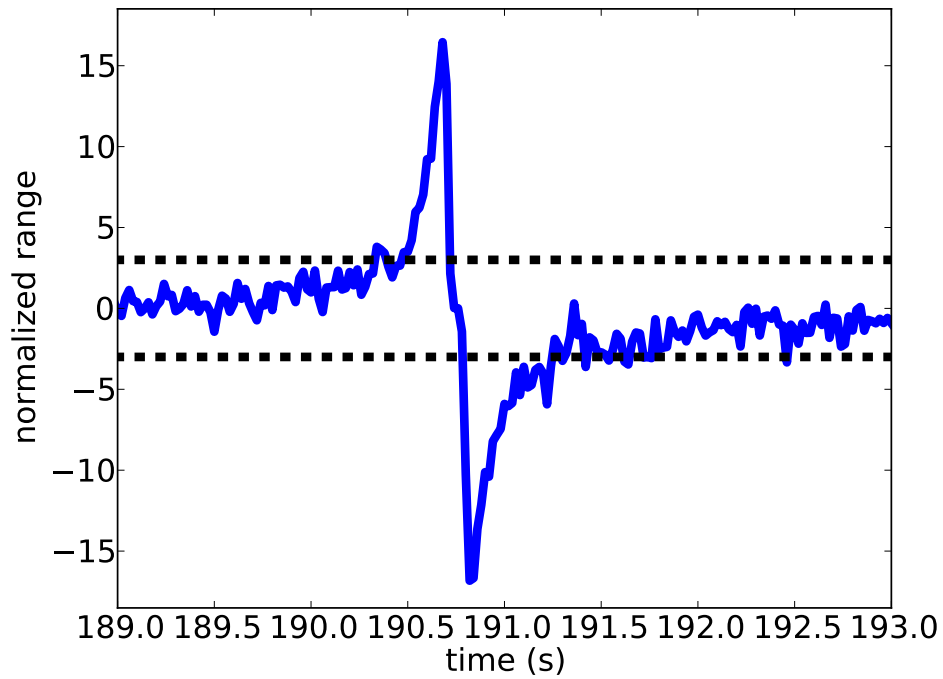


Figure 7.5: Range normalized innovations for satellite 5 at DI error along Magnolia Avenue.

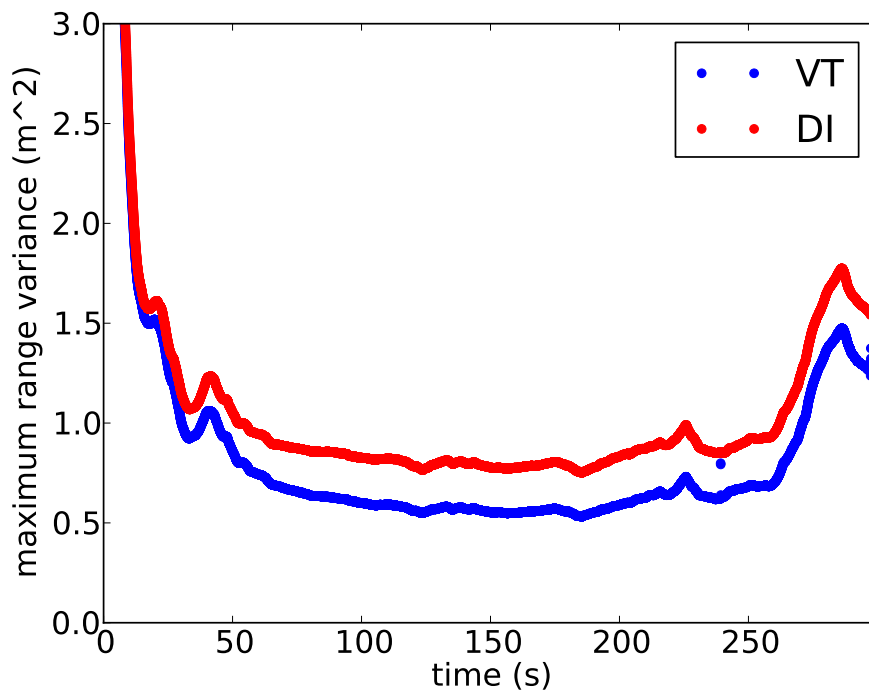


Figure 7.6: Maximum range variance for vector receiver and deeply integrated GPS/INS.

the ECEF x-axis variance for the state covariance matrix, P . As Figure 7.7 shows, the constant velocity model of the vector receiver does not increase the state uncertainty as significantly as the DI solution.

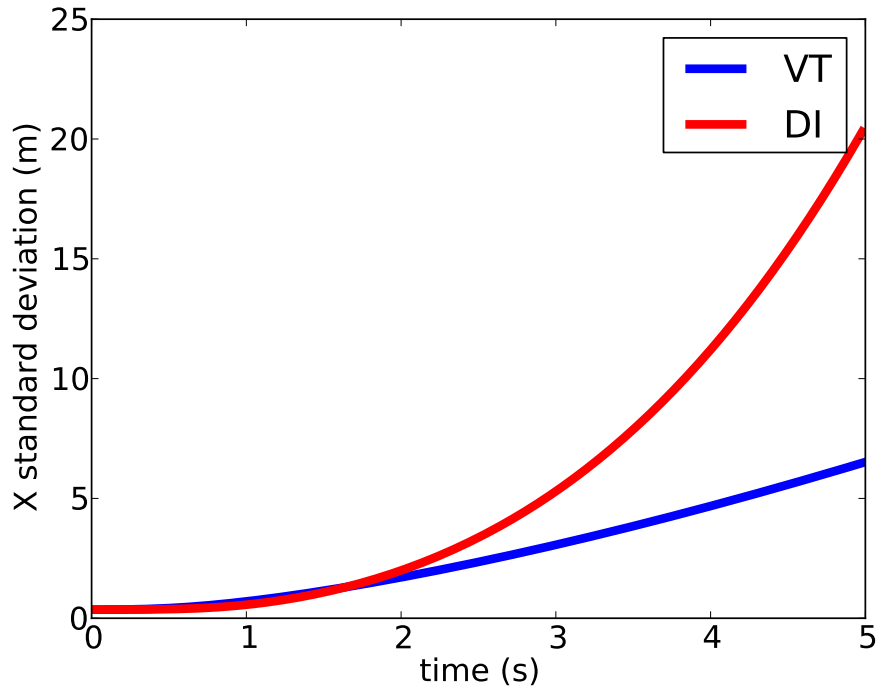


Figure 7.7: Dead reckoning for vector receiver and deeply integrated GPS/INS in a complete signal outage.

Figures 7.6 and 7.7 show that care should be taken in selecting the motion model to be used, either constant velocity or INS based. Accurately modeling the motion of the desired platform is of paramount importance. Sometimes sufficient simplifications can yield lower variances than the inclusion of more sensors.

7.7 Summary

This chapter presented mathematical models and methods to perform deeply integrated GPS/INS integration. This is currently the most closely integrated way to combine inertial measurements into a GPS receiver. This method was demonstrated using a post-processed software GPS receiver and automotive grade IMU. In addition to the integration techniques, fault detection and exclusion

is demonstrated. This method makes use of the improved motion modeling provided by the INS. These techniques were compared in typically difficult GPS environments.

Chapter 8

Conclusion

This dissertation presented advanced approaches to operating a GPS receiver in difficult environments. Its contributions include an improved multipath model for dealing with vector receivers in the presence of reflected signals. Additionally a novel method was developed to achieve solution robustness in the centralized vector receiver and deeply integrated GPS/INS systems.

The vector receiver was shown to be particularly suited for cases when there are several signals that are strong. Certain multipath situations, especially single channel multipath, are an example of this. This dissertation presented a thorough comparison of vector and scalar tracking in a multipath case. The form of the induced errors also did not match the traditional multipath error model so a new model was developed. The traditional approach was based on a steady state error when the early and late correlations are in balance. A vector receiver, however, has other inputs deciding this steady state situation. The new model looked instead at an instantaneous error. This allows performance of the vector and scalar receivers to properly be compared. In this comparison, the vector receiver was able to reduce the range error. It was also demonstrated to reduce position error in multipath environments with live-sky data.

The concept of signal lock was also discussed as it applies to a vector receiver. It was shown that typical scalar lock detection does not apply in a vector context. This was demonstrated with simulated and experimental scenarios in which a vector receiver recovered from hundreds of meters of position error. A new range variance factor was derived to monitor the maximum error on a channel from the receiver position uncertainty. This factor removes the geometry considerations required in making position to range variance predictions for limit testing. To meet the need for robust positioning, a fault detection and exclusion method was adapted for use in a vector receiver. The performance of the FDE algorithm was simulated and its effectiveness in real data was shown. Use of the FDE method proved capable of removing position drift by a meter.

Finally an inertial navigation system was integrated with a GPS vector receiver with fault detection and exclusion capabilities. This natural extension to the GPS-only mode was demonstrated with recorded data. The fault detection method was able to reduce the error of the DI system by several meters when difficult environments caused errors to arise in the measurements.

All of these methods, advanced tracking, fault detection and exclusion, and inertial integration combine to give a robust receiver for operation in difficult environments. As with most problems, this layered solution offers the best results. It is also the most flexible since techniques can be applied as they are available. In all, the Global Positioning System is an amazing resource for the navigation community. This research improves performance in environments that the GPS receiver has typically fared poorly.

As with all potential improvements, there are costs to be considered. In any range-based navigation system, the maximum range variance is a useful parameter to calculate. Its calculation requirements are relatively low and the parameter provides range domain information from position domain covariances. However, the architectural changes required to implement a vector receiver are significant. The main benefit of a vector receiver with FDE is its robustness in the presence of sporadic signals. When a receiver is expected to operate in heavy foliage, urban canyons, or in the presence of noisy signals, a vector receiver with FDE should be considered. The robustness of this method will directly improve performance in these difficult scenarios. When even more robustness is needed, a high quality IMU can be combined for a DI navigation system. However, care should be taken when including an IMU as the introduction of other sensor errors is another trade-off factor in the system design.

8.1 Future Work

There are several promising areas for extension of this work in all three areas of performance improvement. One new advanced tracking architecture is the open loop GPS receiver [32]. While this method is very computationally intensive, it opens up many options in performance by giving the receiver a much more open view of the signal environment. Additional signals, such as multipath, can be seen using this method [41]. Vector tracking has also had limited success when applied to high accuracy carrier phase measurements. While it leverages redundancy well in tracking, it

is still difficult to maintain the navigation solution to centimeter level accuracy. However, this challenge provides investigative opportunities for researchers.

When dealing with the error propagation of a vector receiver, the clock drifts have not been considered. With the increasing viability of chip-scale atomic clocks, study can be done to compare vector convergence bounds among different quality clocks. This comparison can also be made with differing discriminator equations to more generically define the vector convergence bounds.

Widespread use of the vector architecture is dependent on integrity monitoring. This provides a great opportunity for future research into method comparison and development. Defining a lock threshold for a vector receiver has begun in this dissertation but a single threshold check has not been developed. Such a threshold would be useful in determining when to go through reinitialization. Similarly the field of fault detection in a vector receiver is just beginning to grow. Fault detection is the key to more widespread use of vector tracking because one of the limiting factors in a system's use is the statistical confidence in the solution robustness. This thesis, as well as [3] provides the insertion point for further integrity improvements.

Finally the process of fusing sensors with GPS tends to begin with the IMU but extends to other sensor sets. The IMU provides a convenient package that blends well with a GPS receiver, but it too is limited in capabilities. As vector tracking technology develops, other sensors will be more easily integrated with the advanced architecture. Although these sensors may be limited in their applicability, the better performing vector techniques can find their way into more specialized scenarios. One example may be combining map matching with other sensors and a vector tracking GPS receiver. Also, a method should be developed to quantitatively relate IMU quality to DI with FDE performance.

Bibliography

- [1] Don Benson, *Interference benefits of a vector delay lock loop (vdll) gps receiver*, Proceedings of the ION AM Conference (Cambridge, MA), The Institute of Navigation, January 2007.
- [2] S. Bhattacharyya and D. Gebre-Egziabher, *Development and validation of a parametric model for vector tracking loops*, Proceedings of the ION GNSS Conference (Savannah, GA), The Institute of Navigation, September 2009.
- [3] ———, *Integrity analysis of vector tracking architecture*, Proceedings of the ION GNSS Conference (Portland, OR), The Institute of Navigation, September 2010.
- [4] Kai Borre, Dennis M. Akos, Nicolaj Bertelsen, Peter Rinder, and Soren Holdt Jensen, *A software-designed gps and galileo receiver: A single-frequency approach*, Applied and Numerical Harmonic Analysis, Birkhauser, Boston, MA, 2007.
- [5] M. S. Braasch, *Performance comparison of multipath mitigating receiver architectures*, Proceedings of the IEEE Aerospace Conference (Big Sky, MT), IEEE, March 2001.
- [6] Michael S. Braasch, *Multipath effects*, Global Positioning System: Theory and Application, Volume 1 (Bradford W. Parkinson and James J. Spilker Jr., eds.), Progress in Astronautics and Aeronautics, vol. 163, American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1996.
- [7] Robert Grover Brown and Patrick Y.C. Hwang, *Introduction to random signals and applied kalman filtering*, third ed., Wiley, New York, 1997.
- [8] Alan Cameron, *The system: Lightsquared interference with gps*, GPS World, July 2011.
- [9] H. Carvalho, P. Del Moral, A. Monin, and G. Salut, *Optimal nonlinear filtering in gps/ins integration*, IEEE Transactions on Aerospace and Electronic Systems **33** (1997), no. 3.
- [10] ———, *Car jammers: Interference analysis*, GPS World **22** (2011), no. 10.
- [11] Tsung-Yu Chiou, Demoz Gebre-Egziabher, Todd Walter, and Per Enge, *Model analysis on the performance for an inertial aided fll-assisted-pll carrier-tracking loop in the presence of ionospheric scintillation*, Proceedings of the ION NTM Conference (San Diego, CA), The Institute of Navigation, January 2007.
- [12] J.M. Clanton, D.M. Bevly, and A.S. Hodel, *A low-cost solution for an integrated multisensor lane departure warning system*, IEEE Transactions on Intelligent Transportation Systems **10** (2009), no. 1, 47–59.
- [13] Benjamin Clark, *Gps/ins operation in shadowed environments*, Master's thesis, Auburn University, 2006.

- [14] Benjamin Clark and David Bevely, *Gps/ins integration with fault detection and exclusion in shadowed environments*, Proceedings of the ION/IEEE PLANS Conference (San Diego, CA), IEEE, May 2008.
- [15] ———, *Fde implementations for a low-cost gps/ins module*, Proceedings of the ION GNSS (Savannah, GA), Institute of Navigation, September 2009.
- [16] Arinc Research Corporation, *Navstar interface control document*, <http://www.navcen.uscg.gov/gps/geninfo/IS-GPS-200D.pdf>, April 2000.
- [17] Robert N Crane, *A simplified method for deep coupling of gps and inertial data*, Proceedings of the NTM Conference (San Diego, CA), The Institute of Navigation, January 2007.
- [18] A.J. Van Dierendonck, Chris Hegarty, Walt Scales, and Swen Ericson, *Signal specification for the future gps civil signal at l5*, Proceedings of the IAIN World Congress and the 56th Annual Meeting of The Institute of Navigation (San Diego, CA), June 2000, pp. 232–241.
- [19] Jose Diez, Joao S. Silva, and Antonio Fernandez, *Optimised lock detection strategy using boc correlator outputs*, Proceedings of the ION GNSS Conference (2007), 782–788.
- [20] L. Ge, H.C. Chang, V. Janssen, and C. Rizox, *Integration of gps, radar interferometry and gis for ground deformation monitoring*, Proc. 2003 Int. Symp. on GPS/GNSS, Tokyo, Japan, November 2003, pp. 465–472.
- [21] Demoz Gebre-Egziabher, Mark Petovello, and David Bevely, *Integration of gnss and ins: Part 1*, GNSS Applications and Methods (Scot Gleason and Demoz Gebre-Egziabher, eds.), Artech House, Norwood, MA, 2009, pp. 149–176.
- [22] Scott Gleason and Demoz Gebre-Egziabher, *Gnss navigation: Estimating position, velocity, and time*, GNSS Applications and Methods (Scot Gleason and Demoz Gebre-Egziabher, eds.), Artech House, Norwood, MA, 2009, pp. 55–86.
- [23] Paul D. Groves, *Gps signal-to-noise measurement in weak signal and high-interference environments*, Navigation **52** (2005), no. 2, 83–94.
- [24] ———, *Principles of gnss, inertial, and multisensor integrated navigation systems*, Artech House, 2008.
- [25] Donald Gustafson, John Dowdle, and Karl Flueckiger, *A high anti-jam gps-based navigator*, Proceedings of the ION NTM (Anaheim, CA), The Institute of Navigation, January 2000, pp. 495–503.
- [26] Thomas Hobiger, Tadahiro Gotoh, Jun Amagai, Yasuhiro Koyama, and Tetsuro Kondo, *A gpu based real-time gps software receiver*, GPS Solutions **14** (2010), no. 2.
- [27] Todd E. Humphreys, Brent M. Ledvina, Mark L. Psiaki, Brady W. O’Hanlon, and Paul M. Kintner Jr., *Assessing the spoofing threat: Development of a portable gps civilian spoofer*, Proceedings of the ION GNSS Conference (Savannah, GA), Institute of Navigation, September 2008.
- [28] G.-I. Jee, H.S. Kim, Y.J. Lee, and C.G. Park, *A gps c/a code tracking loop based on extended kalman filter with multipath mitigation*, Proceedings of the ION GNSS Conference (Portland, OR), The Institute of Navigation, September 2002.

- [29] J. J. Spilker Jr., *Fundamentals of signal tracking theory*, Global Positioning System: Theory and Application, Volume 1 (Bradford W. Parkinson and James J. Spilker Jr., eds.), Progress in Astronautics and Aeronautics, vol. 163, American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1996.
- [30] Stefan Kiesel, Moritz M. Held, and Gert F. Trommer, *Realization of a deeply coupled gps/ins navigation system based on ins-vdll integration*, Proceedings of the ION NTM Conference (San Diego, CA), The Institute of Navigation, January 2007.
- [31] Stefan Kiesel, Markus Langer, and Gert F. Trommer, *Real time implementation of a non-coherent deeply coupled gps/ins system*, Proceedings of the ION ITM Conference (San Diego, CA), The Institute of Navigation, January 2010.
- [32] Norman F. Krasner, *Method for open loop tracking gps signals*, 10 2003.
- [33] Matthew Lashley, *Modeling and performance analysis of gps vector tracking algorithms*, Ph.D. thesis, Auburn University, 2009.
- [34] Matthew Lashley, David Bevly, and John Hung, *A valid comparison of vector and scalar tracking loops*, Proceedings of the IEEE/ION Position, Location, and Navigation Symposium (Indian Wells, CA), IEEE/The Institute of Navigation, May 2010.
- [35] Matthew Lashley, David M. Bevly, and John Y. Hung, *Analysis of deeply integrated and tightly coupled architectures*, Proceedings of the Position Location and Navigation Symposium (PLANS) (Indian Wells, CA), IEEE/ION, May 2010, pp. 382–396.
- [36] Tao Lin, C. O’Driscoll, and G. Lachapelle, *Channel context detection and signal quality monitoring for vector-based tracking loops*, Proceedings of the ION GNSS Conference (Portland, OR), The Institute of Navigation, September 2010, pp. 1875–1888.
- [37] Pratap Misra and Per Enge, *Global positioning system: Signals, measurements, and performance*, second ed., Ganga-Jamuna, Lincoln, Massachusetts, 2006.
- [38] Dignus-Jan Moelker, *Multiple antennas for advanced gnss multipath mitigation and multipath direction finding*, Proceedings of ION GPS-97, September 1997.
- [39] Thomas Pany, Ronald Kaniuth, and Bernd Eissfeller, *Deep integration of navigation solution and signal processing*, Proceedings of the ION GNSS Conference (Long Beach, CA), The Institute of Navigation, September 2005.
- [40] Andrey Soloviev and Demoz Gebre-Egziabher, *Challenges in gnss/inertial integration*, Webinar presented by Inside GNSS, February 2012.
- [41] Andrey Soloviev and Frank van Graas, *Utilizing multipath reflections in deeply integrated gps/ins architecture for navigation in urban environments*, Proceedings of the IEEE/ION Position, Location, and Navigation Symposium (Monterey, CA), IEEE/The Institute of Navigation, May 2008.
- [42] Frank van Diggelen, *Agps: Assisted gps, gnss, and sbas*, Artech House, 2009.
- [43] Phillip W. Ward, *Performance comparisons between fll, pll, and a novel fll-assisted-pll carrier tracking loop under rf interference conditions*, Proceedings of the ION GPS Conference (Nashville, TN), The Institute of Navigation, September 1998.

- [44] Phillip W. Ward, John W. Betz, and Christopher J. Hegarty, *Interference, multipath, and scintillation*, Understanding GPS: Principles and Applications (Elliott D. Kaplan and Christopher J. Hegarty, eds.), Artech House, Inc., Norwood, MA, second ed., 2006.
- [45] ———, *Satellite signal acquisition, tracking, and data demodulation*, Understanding GPS: Principles and Applications (Elliott D. Kaplan and Christopher J. Hegarty, eds.), Artech House, Inc., Norwood, MA, second ed., 2006.
- [46] L. R. Weill, *A high performance code and carrier tracking architecture for ground-based mobile gnss receivers*, Proceedings of the ION GNSS Conference (2010).
- [47] Global Positioning System Wing, *Navstar gps space segment/navigation user interfaces*, June 2010.
- [48] Nesreen I. Ziedan, *Gnss receivers for weak signals*, GNSS Technology and Applications, Artech House, Norwood, MA, 2006.

Nomenclature

$\dot{c}dt_u$	clock drift
η_i	measurement noise for satellite i
\hat{x}	estimate of x
Ω_e	Earth's rotation rate
ϕ_i	received carrier phase for satellite i
ψ_{eb}^e	attitude error
ρ	pseudorange
erfc	complementary error function
erf	error function
σ_w	thermal noise 1σ frequency jitter
τ_i	transit time for signal i
a_i	unit vector from satellite i to receiver
b	receiver clock bias
b_a	accelerometer biases
b_g	gyroscope biases
c	speed of light
C/N_0	carrier to noise ratio
C_b^e	body to ECEF coordinate transformation matrix

C_i	C/A code for satellite i
cdt_u	clock bias
D_i	data message for satellite i
f_e	dynamic stress error in Hz
f_{Di}	Doppler frequency shift for satellite i
f_{IF}	receiver intermediate frequency
f_{L1}	GPS L1 frequency, 1575.42 MHz
IE	in-phase early correlator
IL	in-phase late correlator
IP	in-phase prompt correlator
P_i	P(Y) code for satellite i
P_{Ci}	power of the C/A signal for satellite i
P_{Pi}	power of the P(Y) signal for satellite i
QE	quadra-phase early correlator
QL	quadra-phase late correlator
QP	quadra-phase prompt correlator
r_{eb}^e	position error
s_i	satellite i
T	correlation time
t_t	transmit time
t_{ri}	received time for signal i

u receiver position

v_{eb}^e velocity error

C/A coarse acquisition code

NCO numerically controlled oscillator

P(Y) precise encrypted code

Appendix A

Correlator

In order to generate the range and range rate errors, a receiver must correlate the incoming and local replica signals. Both the local replica and received signals are at the front-end sampling frequency, which is in the tens of megahertz. No other part of the software receiver must be performed at the sampling frequency. This means that correlation is an extreme bottleneck for any receiver. In a commercial receiver this is typically done in a hardware routine. In a software receiver trade-offs must be taken into account. These trade-offs can broadly be taken as computation time and measurement accuracy. The main driver of accuracy and time is the generation of the local code replica. Three methods of performing correlation were considered and will be detailed in this chapter.

Part of the difficulty in generating a code replica is due to the code frequency being different from the nominal transmitted frequency. The Doppler effect greatly changes the carrier frequency since its wavelength is small but there is little change on the code which has a larger chip length. During a single code period there is little difference among the possible code frequency shifts. This is not true when integrating over the full data bit, or 20 code periods.

The other issue that brings inaccuracy is a quantization effect when the local replica code is generated to begin exactly at a sample. Since these samples are not perfectly aligned with the beginning of a received code period, inaccuracy is introduced. With a sampling frequency of 16.3676 Hz (as used in this work), this translates to a pseudorange resolution of about 18 m.

When both of these error sources are present, the effect can be extreme pseudorange errors. The effect will also appear as a sawtooth pseudorange error. This occurs because small pseudorange corrections do not change the replica initial sample or the code frequency. When the replica sample slips to the next nearest sample, an 18 m pseudorange jump occurs, leading to the sawtooth appearance. This is illustrated in Figure A.1.

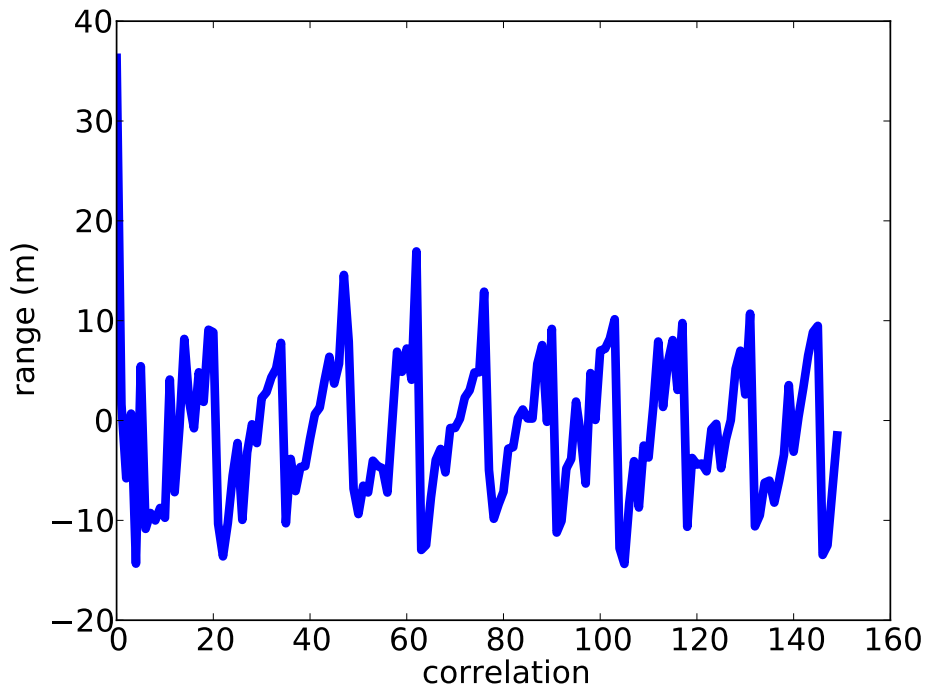


Figure A.1: Sawtooth pseudorange error with correlation errors

A.1 Fast Correlation

In order to speed the process of each correlation step, the code replica can be generated once at initialization. Here each correlation step is simply an aligning of the initial replica and received signal indices and performing the multiplication and summation. This results in the minimal amount of work to generate the local replica every time a correlation occurs. It is relatively easy to perform an integrate and dump operation on two previously generated sequences in memory. Correlating in this way leads to the fastest method. This method is described in Figure A.2.

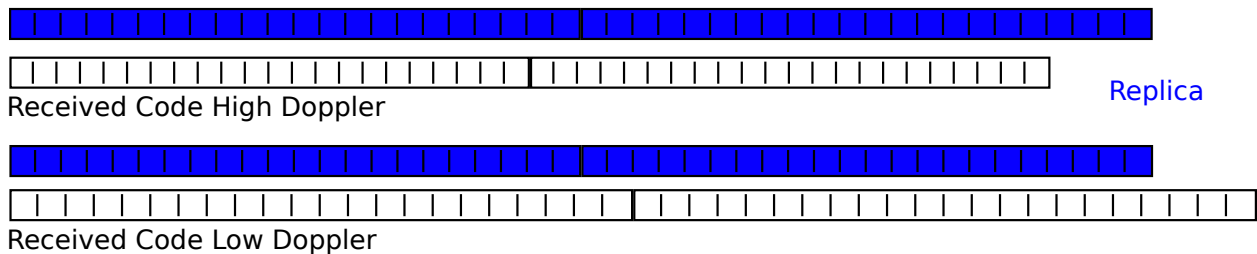


Figure A.2: Alignment of replica code and received code for fast correlation

However, there is a loss of accuracy in generating the local code replica at initialization. Since the Doppler effect also causes a change on the C/A code, the received signal will not perfectly match the previously generated replica. As the correlation time expands, this misalignment adds up so that there is a systemic error in the measurement generation. The results of this effect are shown in Section A.4.

A.2 Fine Correlation

On the opposite end of the spectrum lies a correlation in which each epoch is generated as needed. In this case there is little that can be generated at the initialization. Here the local replica is upsampled when the correlation is needed. This replica takes into account the predicted code frequency and the fractional code period beginning between absolute samples. If the C/A code, C , is the array of chip values of length 1023 then the local replica, L , is generated as

$$L[i] = C \left[\text{floor} \left((\theta + i) \frac{f_C}{f_s} \right) \right] \quad (\text{A.1})$$

Where θ is the fractional portion of the time between the signal samples that the code period is estimated to begin, f_C is the code frequency (accounting for Doppler shift as in Section 2.11, f_s is the sampling frequency, and $[]$ represents indexing. Note that using the $\text{floor}()$ function, this is essentially a zero order hold along a chip value. The resulting replica code contains the best estimate for each chip at each of the sample locations. This method is described in Figure A.3.

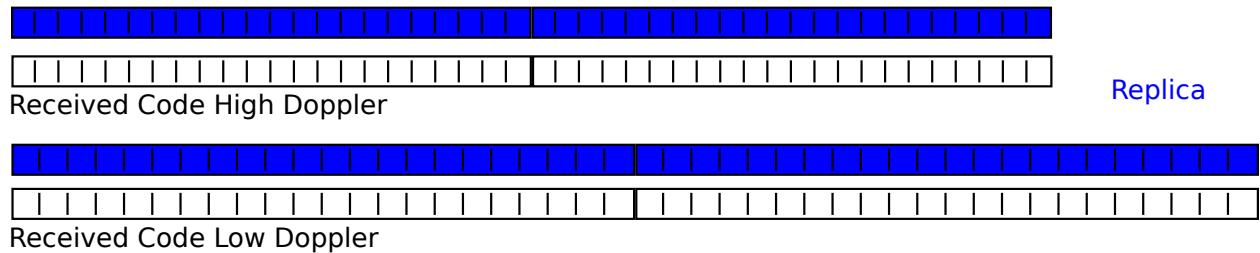


Figure A.3: Alignment of replica code and received code for fine correlation

However, this approach is heavily computationally intensive. With an example sampling frequency of 16.3676 MHz (the sampling frequency of the NordNav), and correlating for 20 ms, this means generating 327352 samples for each correlation. Each channel has six correlators for tracking

plus nine more for noise computation. Therefore in a receiver tracking eight satellites the system must generate effectively a length 39282240 array. At this point generation is pressing the efficiency of the implementation language. Since this software receiver was written in Python for flexibility, this correlation step was pre-compiled as a C module to be called from the receiver.

A.3 Efficient Correlation

A good trade-off between efficiency and accuracy can be found by considering how misalignment affects a series of code periods. When using an initialized replica code at the transmitted code frequency, only one parameter is needed. That parameter is what index in the received code that corresponds to a code period start. With a sampling frequency of 16.3676 MHz, each code chip is just under 16 samples. Even significant changes to the code frequency due to Doppler shift cause very little contraction or expansion of the code due to its long wavelength. Considering a single code replica at the high and low ends of the Doppler range for a single code period there is a difference of only 0.00489%. Therefore for a correlation time of only 1 ms there is negligible accuracy difference in the three methods. This is due to the coarse resolution of the code chips.

Within a code period, which contains 1023 chips, there are only a few samples at the edges of the transitions that may be different. However, when multiple periods are used in a multiple code period correlation time, the period start slips more and more relative to the actual start of the code period. The fast correlation method ignores this slipping and tries to align the entire correlation time based on the phase of the first code. This leads to errors accumulating as more periods are summed. The fine correlation method handles this situation by determining where each sample should lie. However, this is done at great computational cost. A compromise is made by aligning each of the code periods independently. This is similar to applying the fast correlation method sequentially for each code period within the correlation time. This method is described in Figure A.4.

A.4 Correlation Results

For each of the correlation methods and for different signal scenarios, the following results are shown. Figures A.5 - A.7 show all three methods over a range of delays with their corresponding

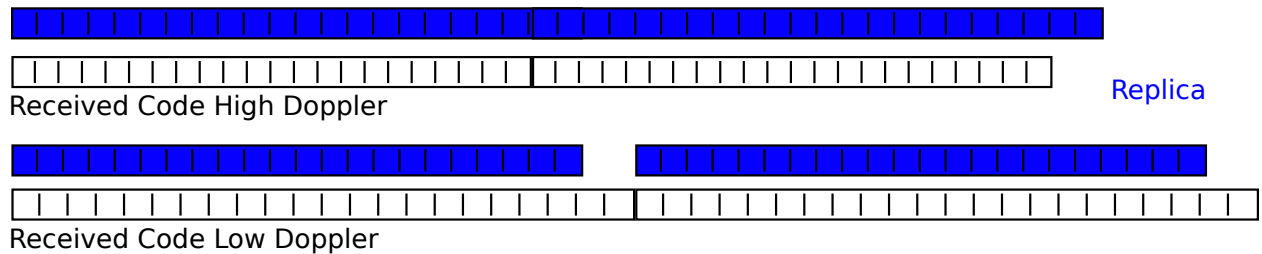


Figure A.4: Alignment of replica code and received code for efficient correlation

correlations for correlations of length 2, 10, and 20 ms. This is equivalent to sampling the correlation function at many points, similar to using early, prompt, and late correlators.

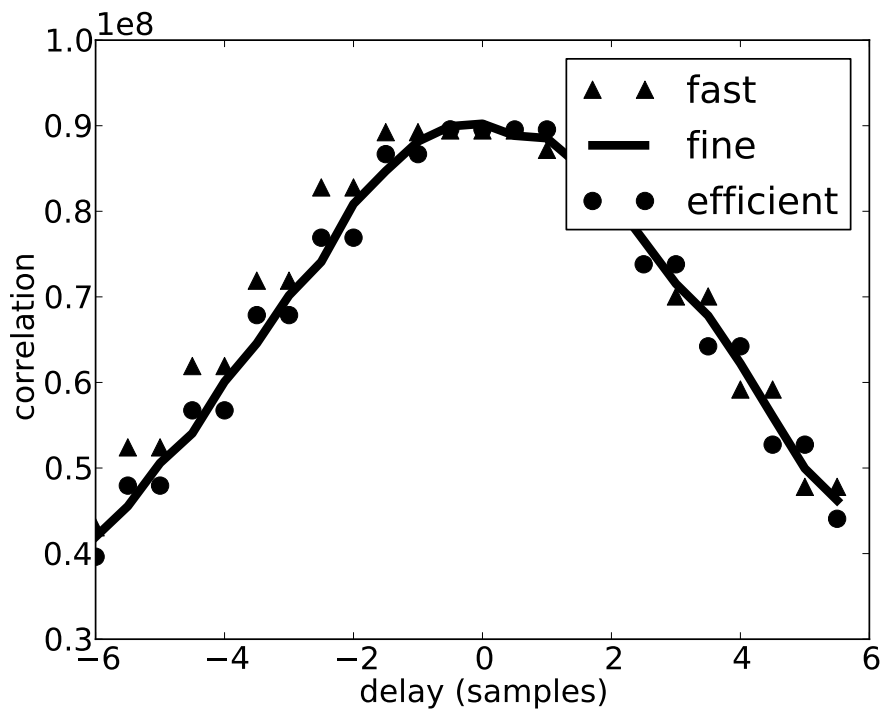


Figure A.5: Correlation methods for 2 ms correlation

Notice how the fast correlation performs well at short correlation times. However, at longer integration times the measurement accuracy is greatly reduced. There is a significant difference between the generated codes at any Doppler shift and at 2965.233 Hz 11.30% of the fast code samples are incorrect while 0.38% of the efficient code samples are incorrect. This is a significant improvement in accuracy with only a moderate amount of extra processing. The resulting time per correlation call with a 20 ms correlation time results in the following table.

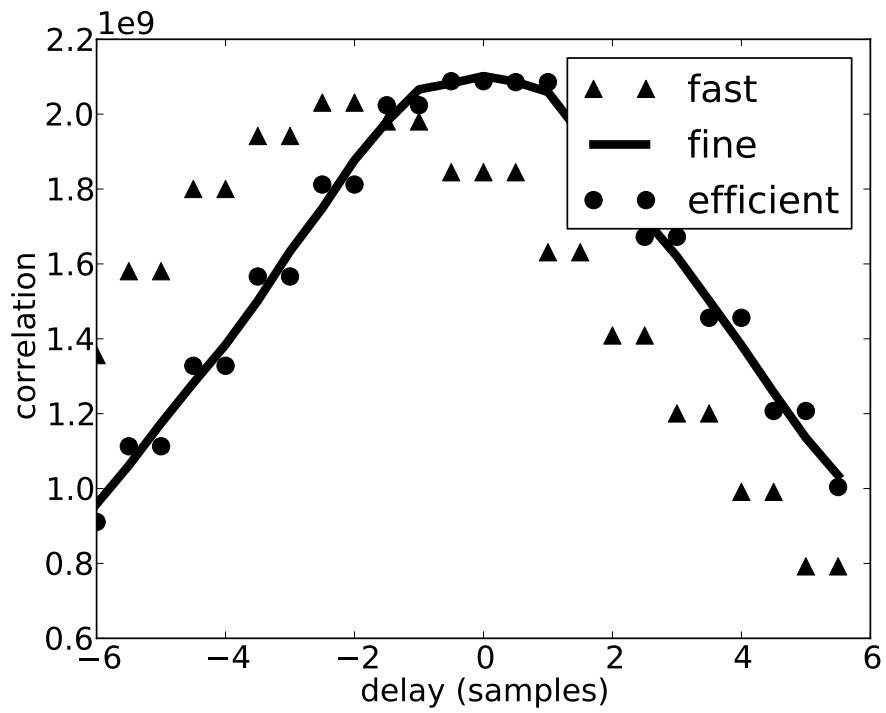


Figure A.6: Correlation methods for 10 ms correlation

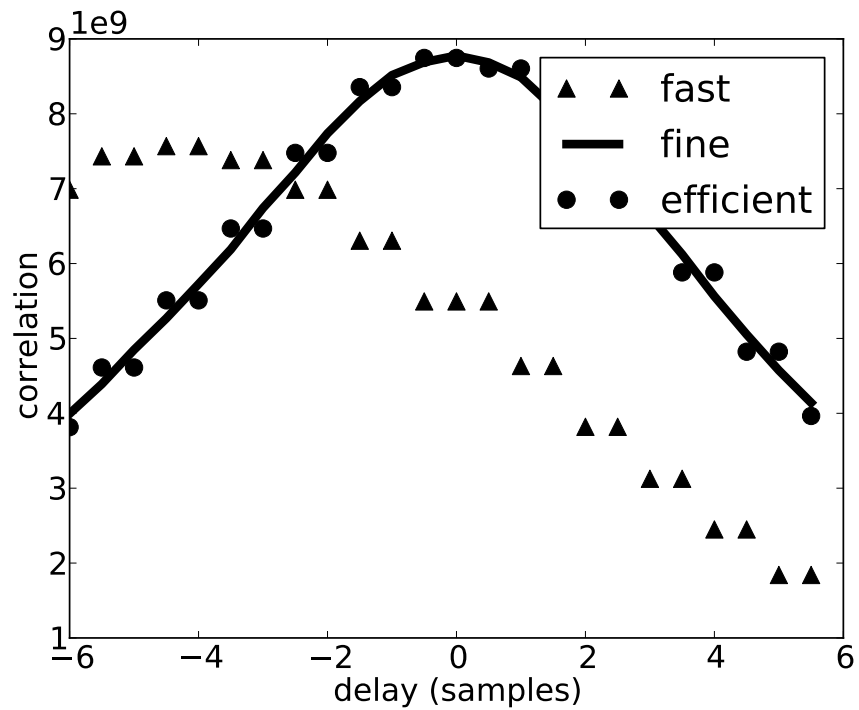


Figure A.7: Correlation methods for 20 ms correlation

Table A.1: Seconds per correlation call for three correlation methods

Fast	Fine	Efficient
0.123	1.619	0.128

Appendix B

Simulator

Since GPS difficult environments are difficult to precisely control, simulation can be used as a receiver design tool. When dealing with a post-processed data file in a GPS software receiver, samples are read as needed from a static file as they would appear in time to a receiver. Designing this way makes a receiver ‘real-time capable’ since no major modifications are necessary to move to a live data link.

B.1 RF Level Simulator

Commercial GPS simulators fall into this category. In an RF simulator, the resulting output involves upconversion and generation of an analog signal that is then fed into hardware GPS receivers. This technique provides some significant capabilities for testing receivers. However, for software receivers such as those demonstrated in this thesis, this adds another layer of complexity. Going all the way to an analog RF signal requires the use of a GPS hardware front-end. It also requires significant hardware cost to generate this signal fast enough. For testing a software receiver, using this type of simulation would only be helpful in a pre-production type of project.

B.2 IF Level Simulator

Typical simulations for a software receiver employ this same method by predicting what the data file will look like. Given the receiver’s f_{IF} and f_s , along with user and satellite dynamics, the expected received signals can be mathematically predicted. These signals would look like those in Equation (2.16). By properly altering the signals with the correct delays and frequency shifts, a nominal data file can be produced. Using this single data file, a receiver can run through the simulated scenario as many times as needed. Combining this type of simulator with the software receiver described in Chapter 2 will then perform as in Figure B.1.

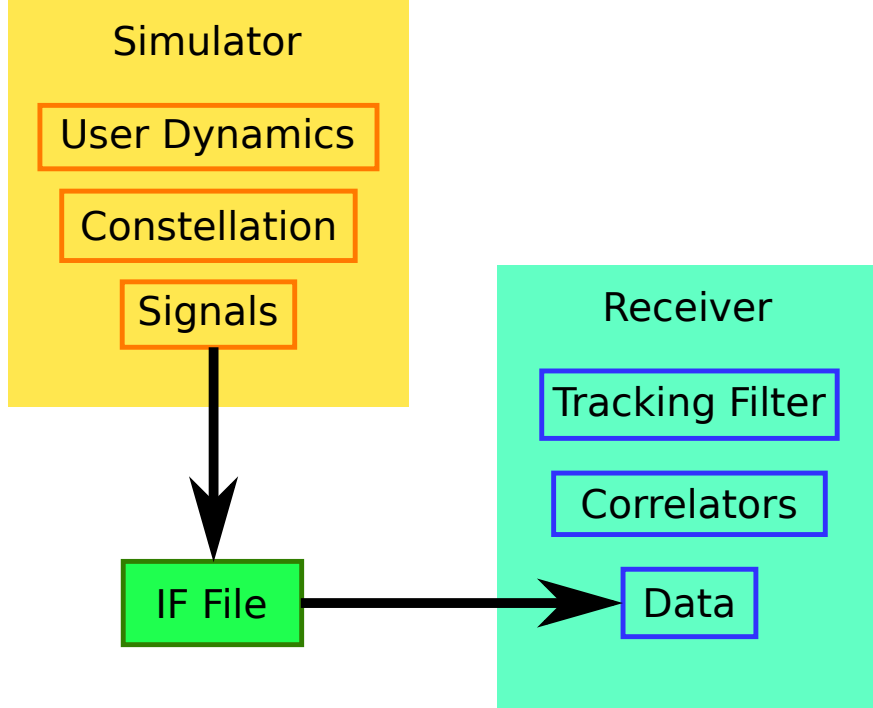


Figure B.1: Simulator generating IF data file for use in a software receiver

B.3 Correlation Level Simulator

From a simulation efficiency standpoint, this method retains the most time consuming portion of the software receiver operation: the correlation. Correlator implementations are described in more detail in Appendix A. The correlation step is taken to effectively reduce the data from the sampling rate, f_s , to a much lower correlation rate, usually 50 to 1000 Hz. Doing so averages out a significant amount of the received noise while also reducing the rate required to process the tracking loops (either scalar or vector). From a simulation perspective, the correlator outputs are highly predictable since they relate only to signal amplitude, delays, frequency shifts, and noise. These correlator outputs have been modeled as

$$I = AdR(\epsilon + \tau) \text{sinc}(2\pi\phi T) \cos(\theta) + \nu_I \quad (\text{B.1a})$$

$$Q = AdR(\epsilon + \tau) \text{sinc}(2\pi\phi T) \sin(\theta) + \nu_Q \quad (\text{B.1b})$$

$$A = \sqrt{2aC/N_0T} \quad (\text{B.1c})$$

Where A is the amplitude, d is the data message bit (either 1 or -1), R is the C/A code autocorrelation function, ϵ is the signal delay, τ is the intentional correlator delay (for early, prompt, or late correlators), ϕ is the frequency error, θ is the phase error over the correlation period, and ν is the unit variance, zero mean noise.

$$R(\epsilon) = \begin{cases} 1 - \frac{|\epsilon|}{b} & -b \leq \epsilon \leq b \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.2})$$

Using these correlation models, the generation of the intermediate frequency samples is no longer necessary. In fact, a software receiver's correlation function can be overloaded to provide the model outputs for the requested correlation. This case is shown in Figure B.2.

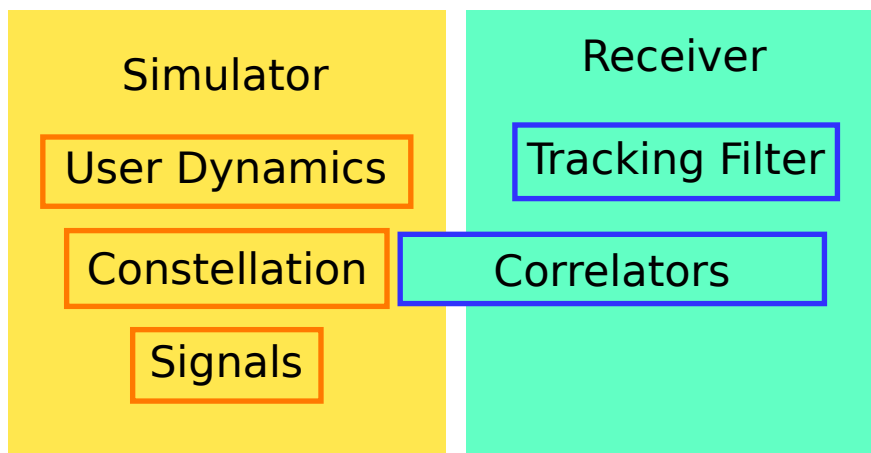


Figure B.2: Simulator generating correlator outputs for use in a software receiver

The main complication if this method comes from the fact that the delays and frequency shifts cannot be predicted beforehand like the IF samples can. Therefore a simulator running in this way must be run in parallel with the software receiver. As the receiver requires correlator outputs, the simulator generates these outputs as if the receiver had a bank of correlators pulling data from a file and combining it with a local replica. The simulator is able to accurately predict what a correlator would output based on the requested code phase, carrier frequency, and carrier phase. However, this prediction must be made at runtime rather than before in a file. In this way comparisons between methods using the same random values becomes more complicated. However, this trade-off is rewarded with a much faster implementation of the overall simulated environment.

This implementation allows for easy and fast comparison using the same receiver architecture with real IF data and simulated correlations. The receiver generates its predicted code phase, carrier frequency, and carrier phase in exactly the same way as before. The simulator keeps takes these values and calculates what the range delay and frequency shift should be. These values are then used in Equation (B.1). Additional delays can be included such as atmospheric delay. User clock bias is included in the delay calculation to give consistent results when dealing with a time offset from GPS time.

Appendix C

Vector Tracking in a Scalar Formulation

As an extension of the equivalent scalar formulation described in Chapter 3, it appears that the scalar formulation can be operated in a vector manner. In fact, for covariance comparison the scalar formulation was used in [34] to draw conclusions between scalar tracking and vector tracking. However, the formulation described in Section 3.2.2 suffers from poor numerical properties. While the algorithms theoretically yield the same results, they do not operate equivalently when implemented.

C.1 Condition Number

A metric used to describe how well conditioned a matrix is for solving is the condition number. If a matrix A relates vector \bar{x} to vector \bar{y} as

$$A\bar{x} = \bar{y} \tag{C.1}$$

Then a change in \bar{x} will yield a change in \bar{y} that is determined by the system A . That is,

$$A(\bar{x} + \bar{q}) = \bar{y} + \bar{p} \tag{C.2}$$

The condition number k gives a bound on the magnitude of these changes as

$$\frac{|\bar{p}|}{|\bar{x}|} \leq k \frac{|\bar{q}|}{|\bar{y}|} \tag{C.3}$$

Where $|\bar{x}|$ represents the norm of \bar{x} .

From an algorithmic standpoint, the condition number is a measure of how much precision is lost when solving the system in Equation (C.1). That is, how many digits of information are changed in the solution verses digits of information changed in the input. It represents a magnifying

effect from the input to the output. With a high condition number, small changes in the input result in large changes in the solution. Therefore extremely high precision is needed to get moderate precision for the system. The larger k gets, the worse this effect becomes.

C.2 Experimental Results

To illustrate the numerical properties of vector tracking in a scalar formulation, both vector receivers are shown operating in a fairly difficult environment. This environment includes large blockages of view by operating close to multi-story buildings as well as some foliage along the road. The position is shown in Figure C.1. It can be seen that the performance is greatly degraded when the vector receiver attempts to operate in the scalar formulation. Figure C.2 shows the high condition number of the run. This condition number is taken of the filter state covariance matrix, which shows how the states are related to each other.

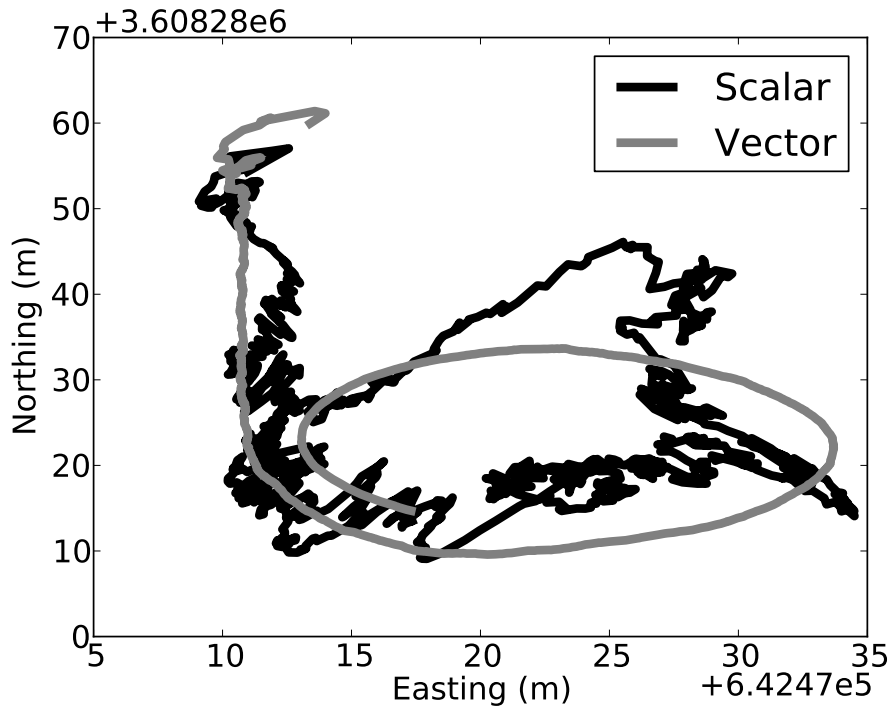


Figure C.1: Vector receiver operation in difficult environments

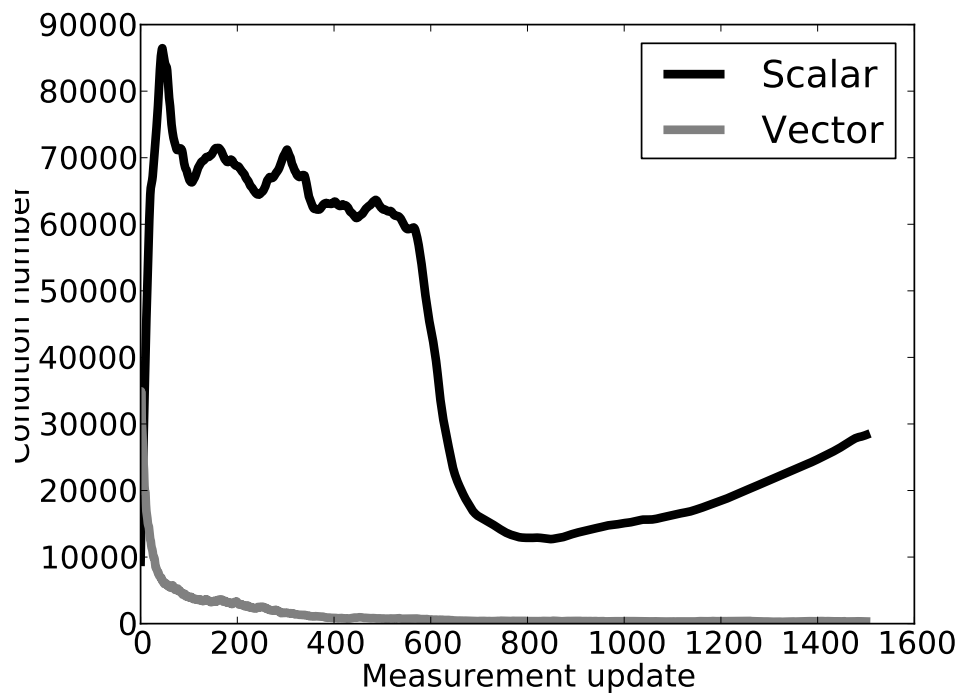


Figure C.2: Condition number for vector receiver operation in difficult environments

Appendix D
Multipath Results

The following figures provide a graphical documentation for the results described in Chapter 4. These results were generated using a parallel simulation setup implemented on the high performance computing cluster at Auburn University. Each dot represents an independent run of the simulator and receiver.

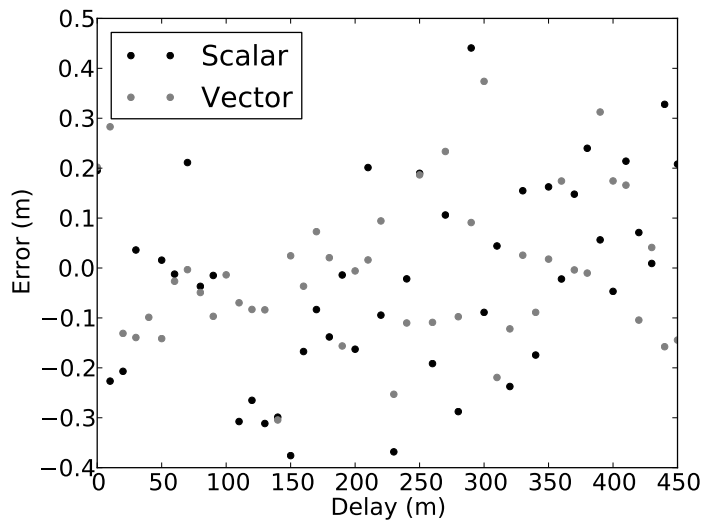


Figure D.1: Tracking error for 9 satellites and multipath-to-direct ratio 0.001.

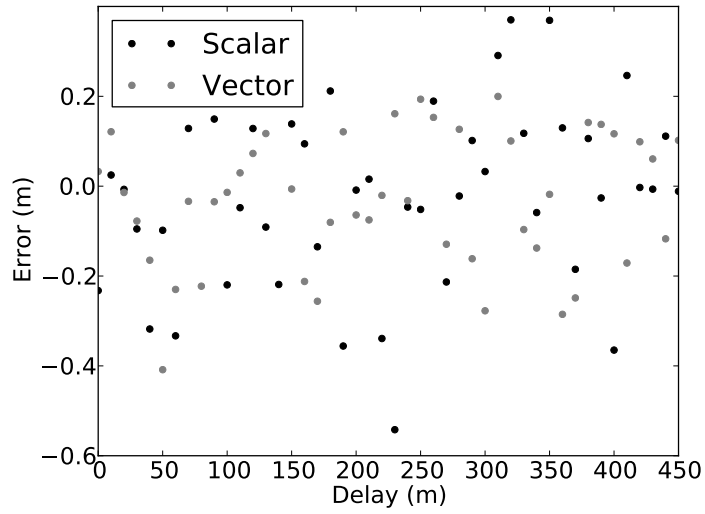


Figure D.2: Tracking error for 8 satellites and multipath-to-direct ratio 0.001.

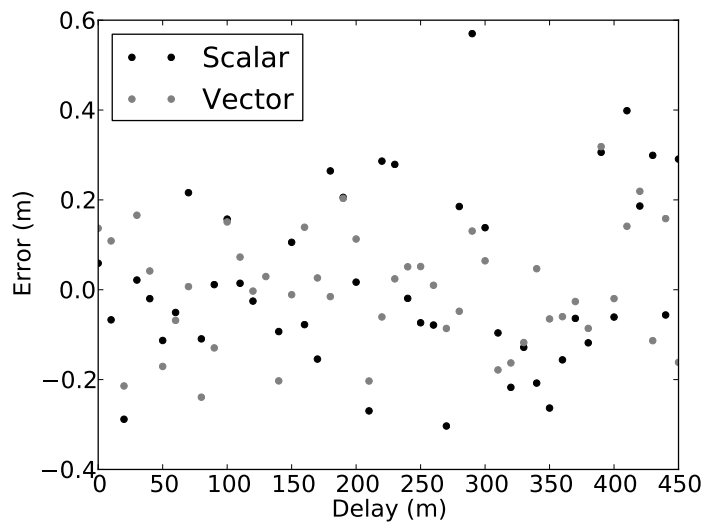


Figure D.3: Tracking error for 7 satellites and multipath-to-direct ratio 0.001.

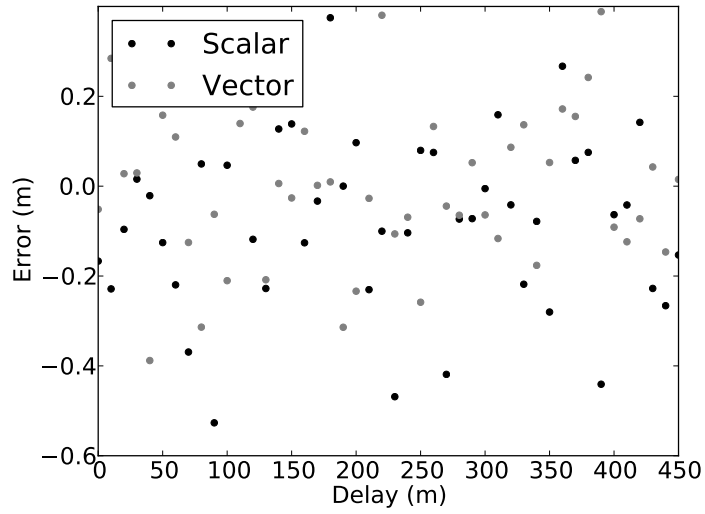


Figure D.4: Tracking error for 6 satellites and multipath-to-direct ratio 0.001.

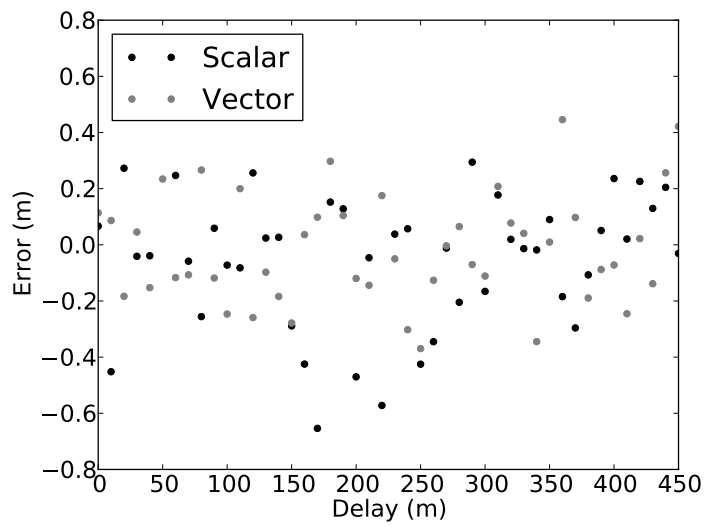


Figure D.5: Tracking error for 5 satellites and multipath-to-direct ratio 0.001.

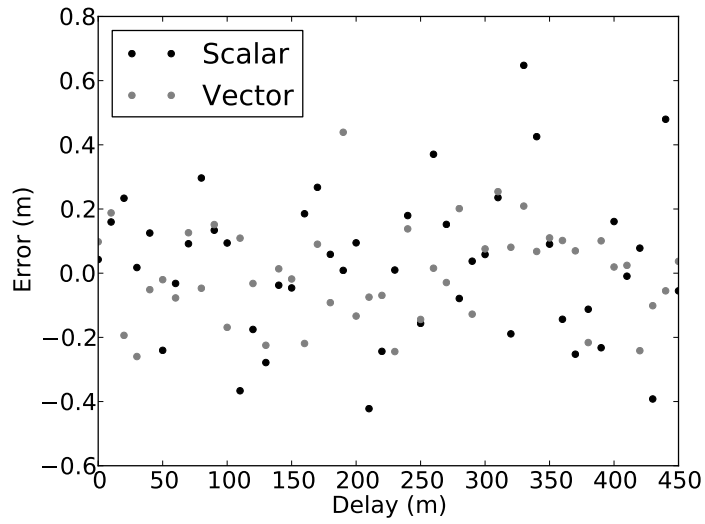


Figure D.6: Tracking error for 9 satellites and multipath-to-direct ratio 0.003.

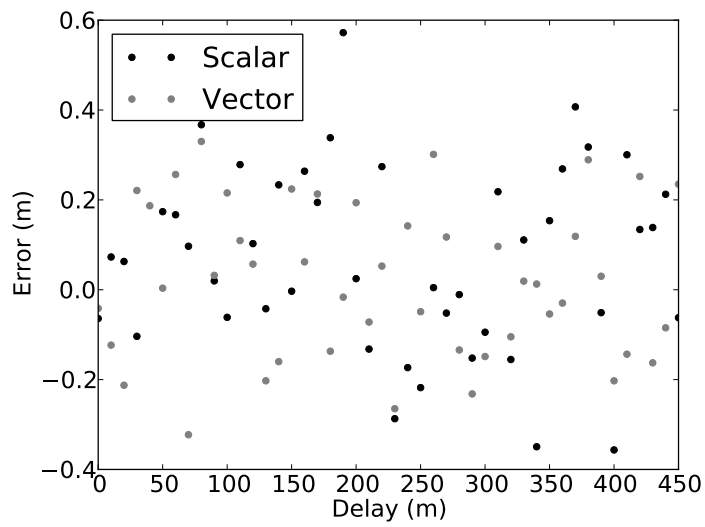


Figure D.7: Tracking error for 8 satellites and multipath-to-direct ratio 0.003.

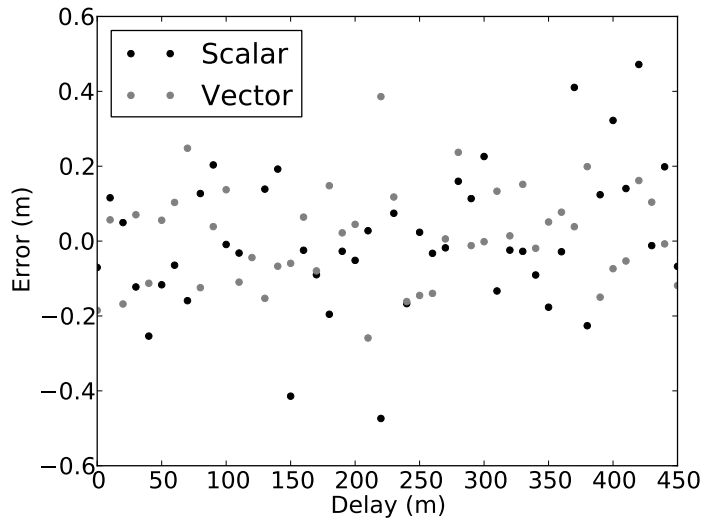


Figure D.8: Tracking error for 7 satellites and multipath-to-direct ratio 0.003.

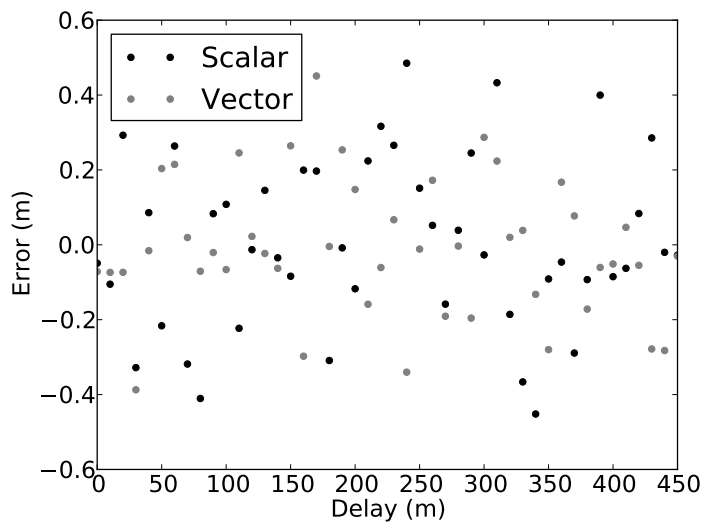


Figure D.9: Tracking error for 6 satellites and multipath-to-direct ratio 0.003.

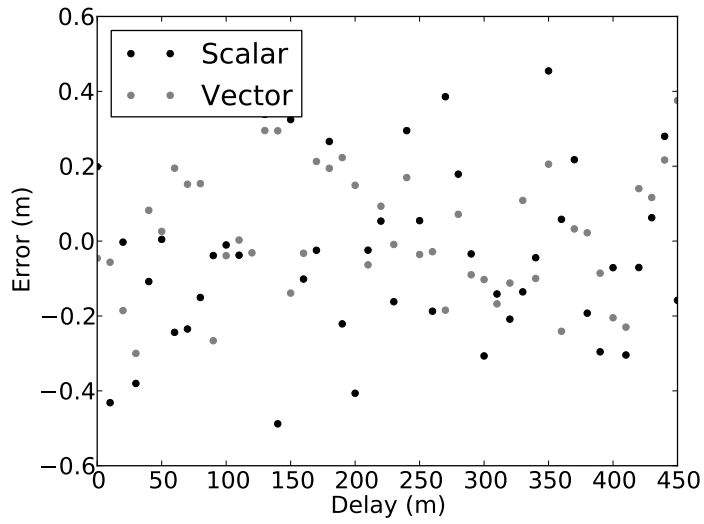


Figure D.10: Tracking error for 5 satellites and multipath-to-direct ratio 0.003.

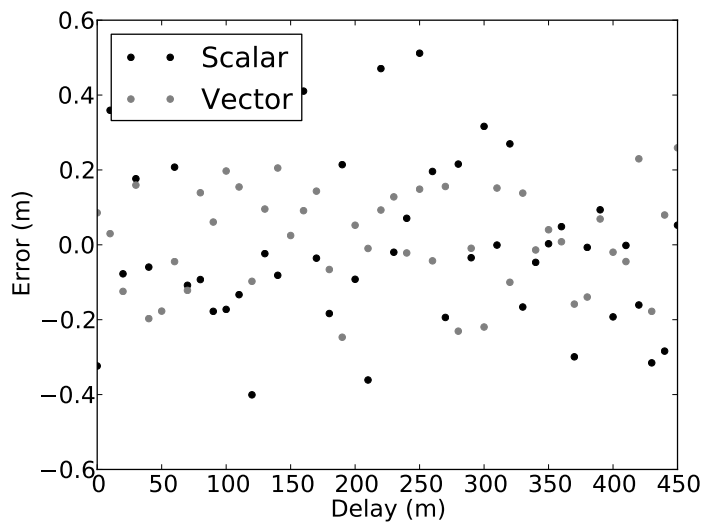


Figure D.11: Tracking error for 9 satellites and multipath-to-direct ratio 0.01.

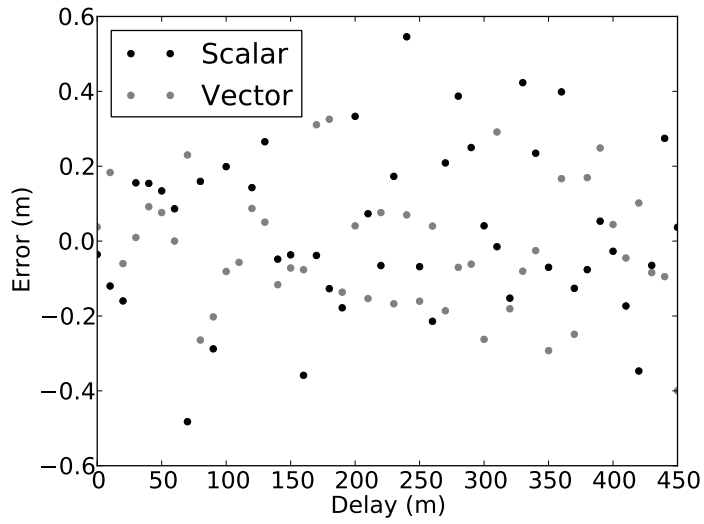


Figure D.12: Tracking error for 8 satellites and multipath-to-direct ratio 0.01.

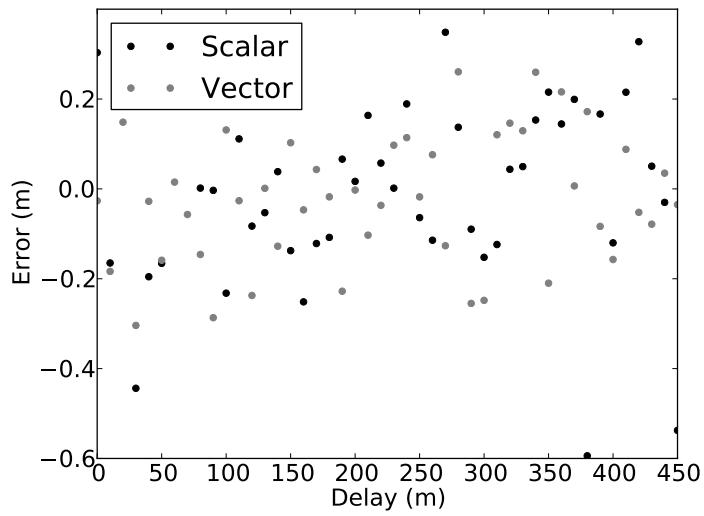


Figure D.13: Tracking error for 7 satellites and multipath-to-direct ratio 0.01.

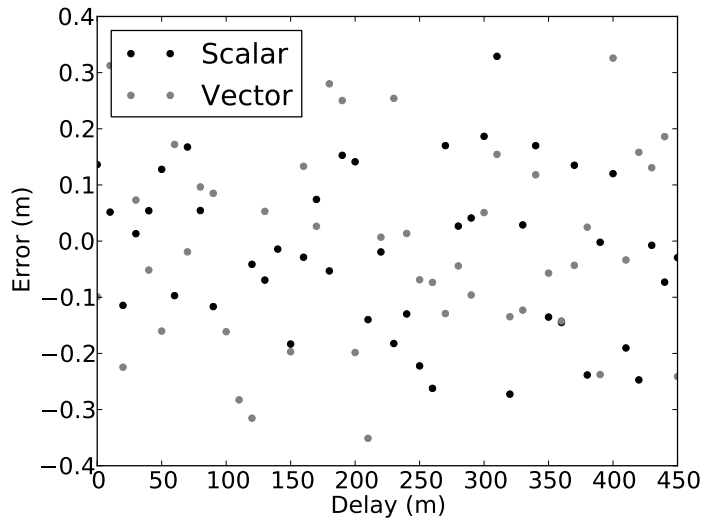


Figure D.14: Tracking error for 6 satellites and multipath-to-direct ratio 0.01.

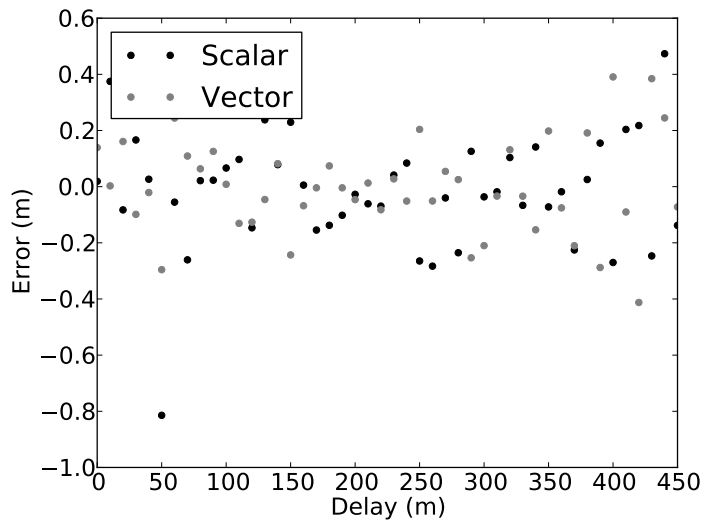


Figure D.15: Tracking error for 5 satellites and multipath-to-direct ratio 0.01.

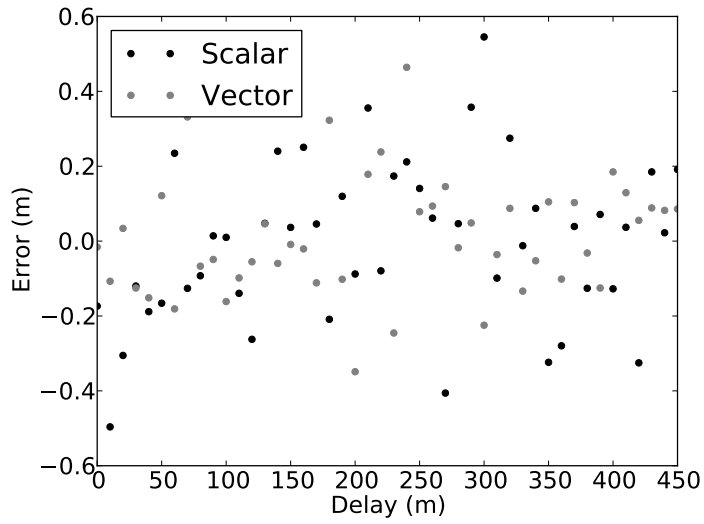


Figure D.16: Tracking error for 9 satellites and multipath-to-direct ratio 0.031.

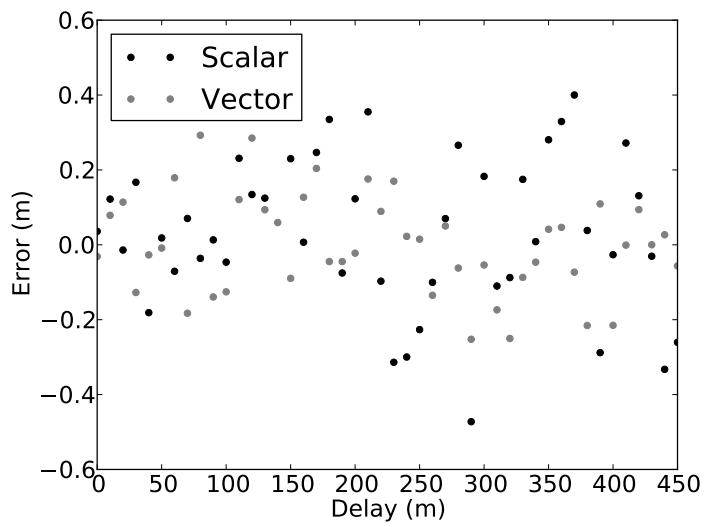


Figure D.17: Tracking error for 8 satellites and multipath-to-direct ratio 0.031.

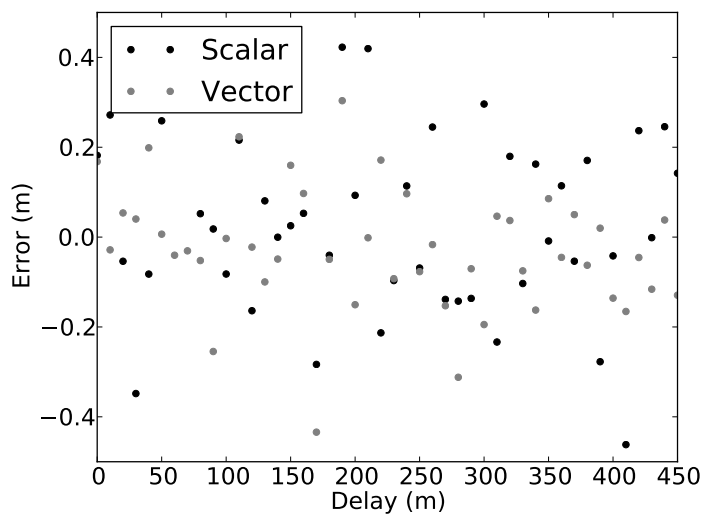


Figure D.18: Tracking error for 7 satellites and multipath-to-direct ratio 0.031.

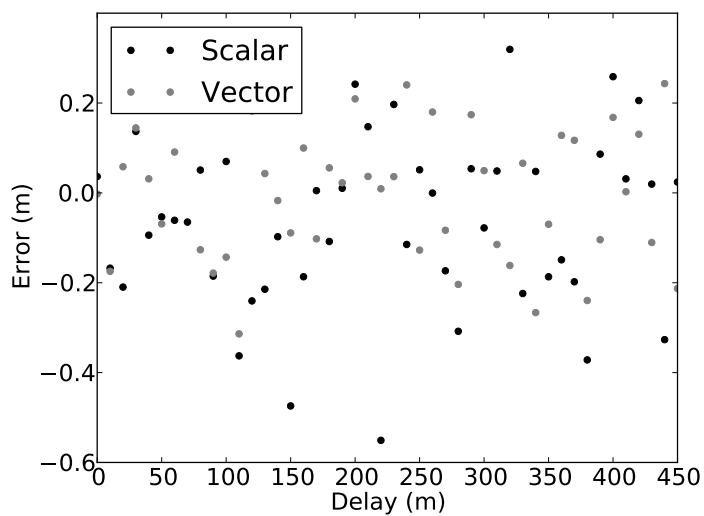


Figure D.19: Tracking error for 6 satellites and multipath-to-direct ratio 0.031.

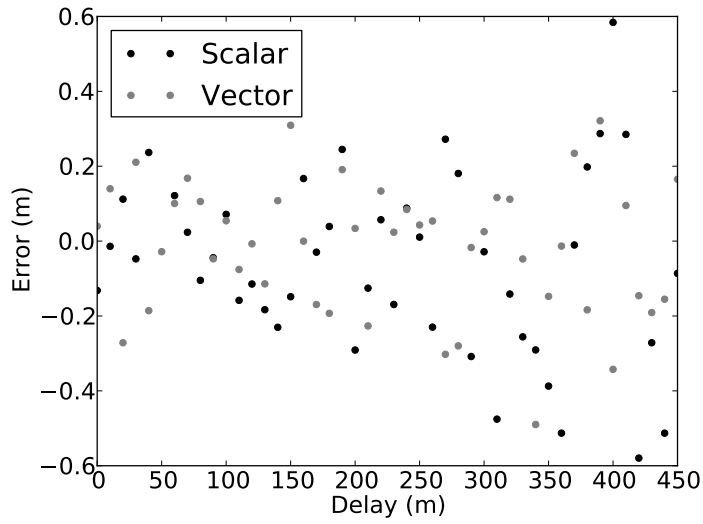


Figure D.20: Tracking error for 5 satellites and multipath-to-direct ratio 0.031.

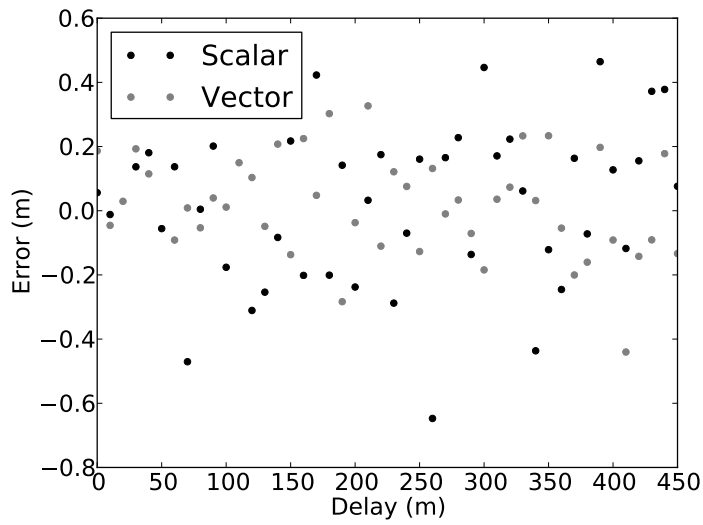


Figure D.21: Tracking error for 9 satellites and multipath-to-direct ratio 0.1.

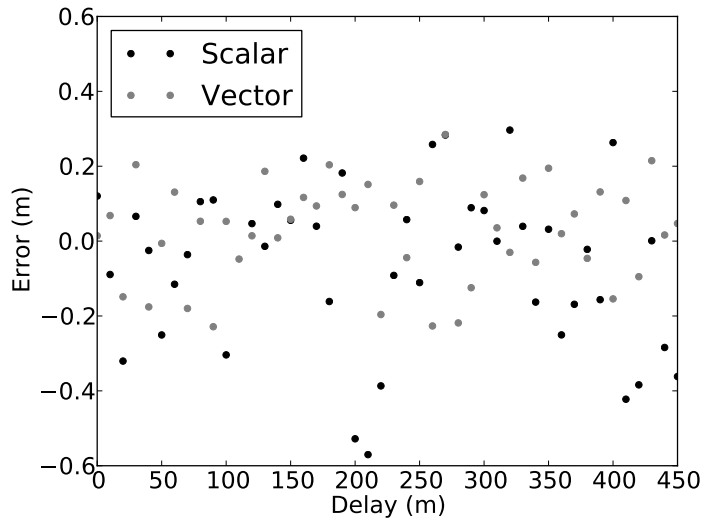


Figure D.22: Tracking error for 8 satellites and multipath-to-direct ratio 0.1.

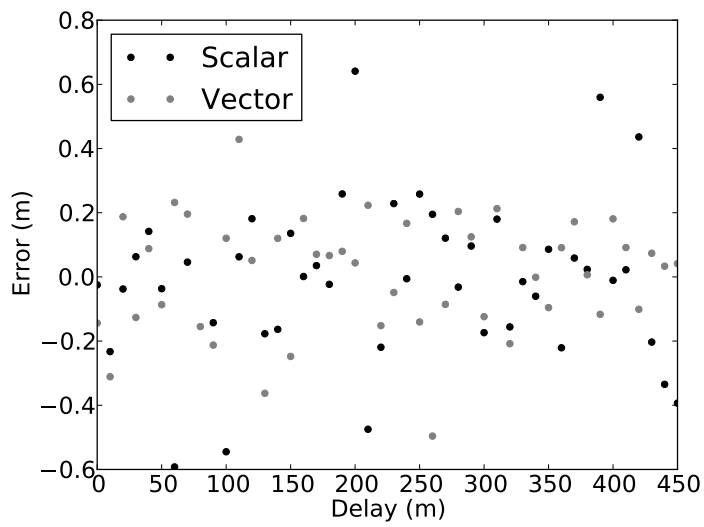


Figure D.23: Tracking error for 7 satellites and multipath-to-direct ratio 0.1.

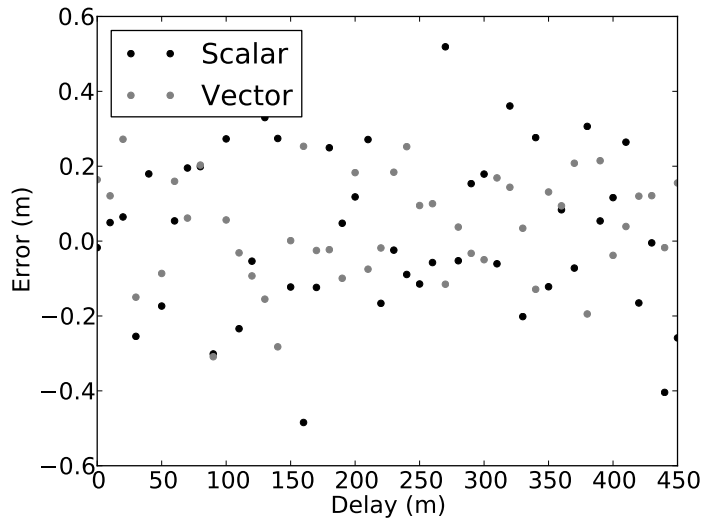


Figure D.24: Tracking error for 6 satellites and multipath-to-direct ratio 0.1.

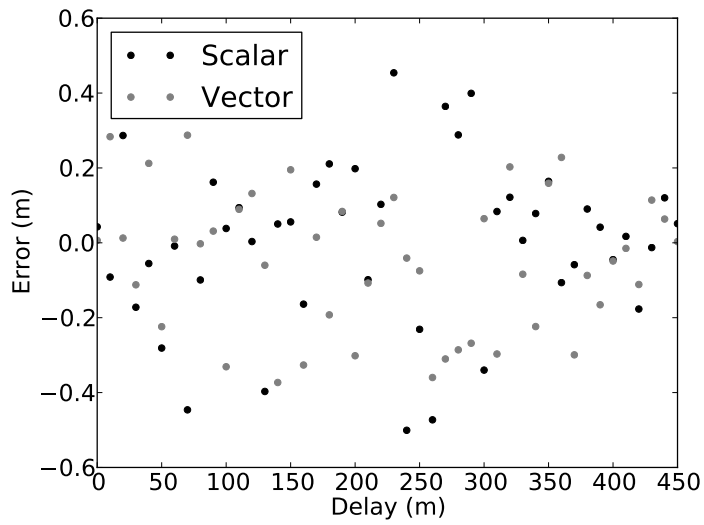


Figure D.25: Tracking error for 5 satellites and multipath-to-direct ratio 0.1.

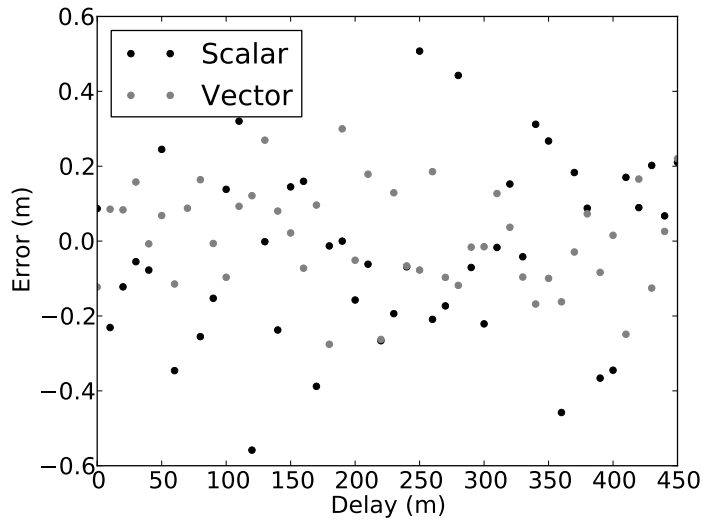


Figure D.26: Tracking error for 9 satellites and multipath-to-direct ratio 0.316.

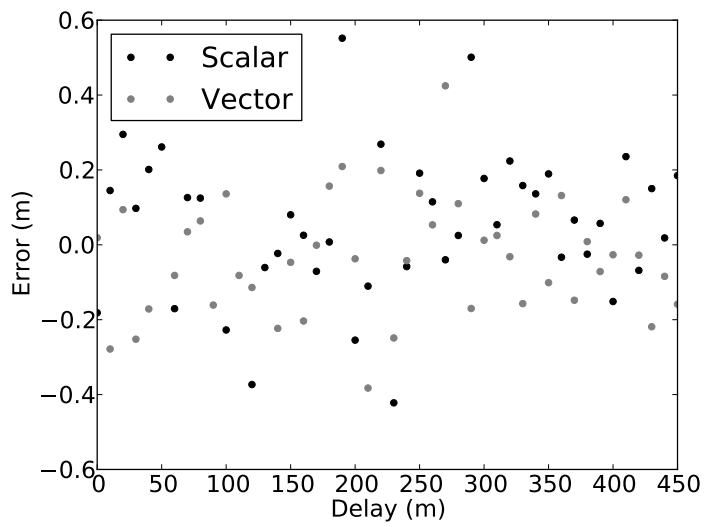


Figure D.27: Tracking error for 8 satellites and multipath-to-direct ratio 0.316.

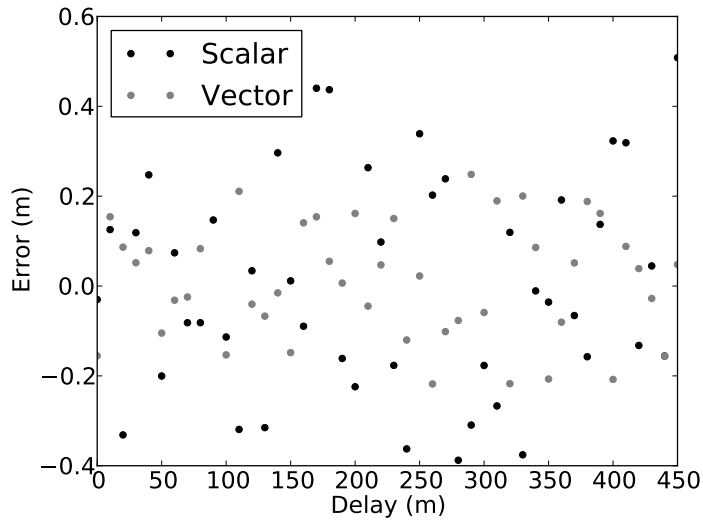


Figure D.28: Tracking error for 7 satellites and multipath-to-direct ratio 0.316.

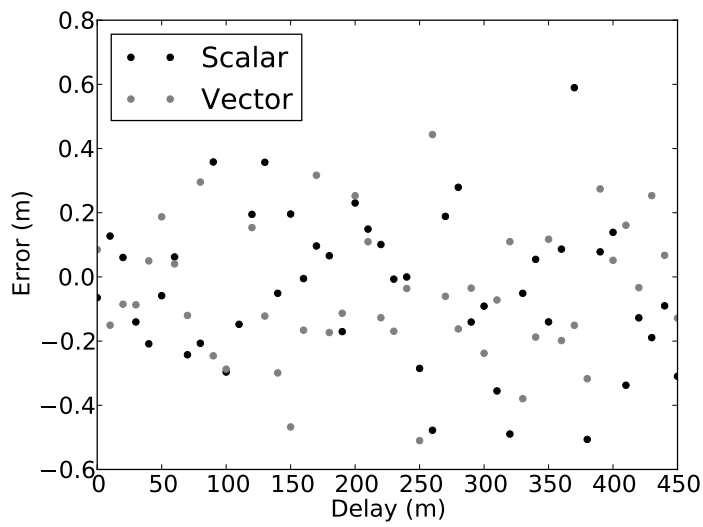


Figure D.29: Tracking error for 6 satellites and multipath-to-direct ratio 0.316.

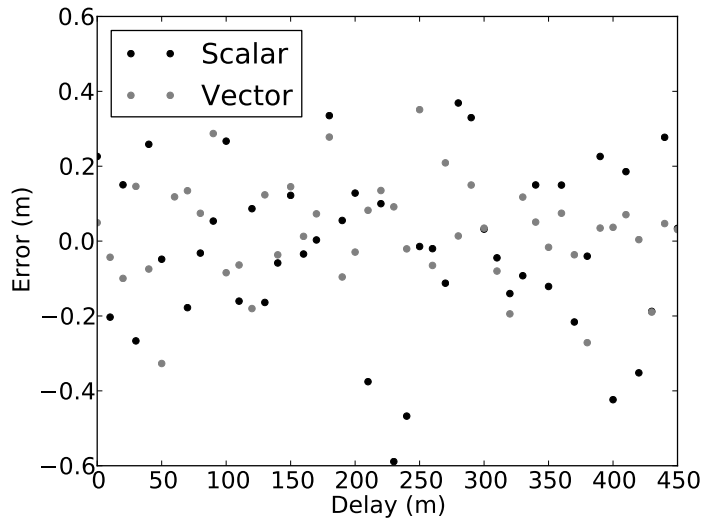


Figure D.30: Tracking error for 5 satellites and multipath-to-direct ratio 0.316.

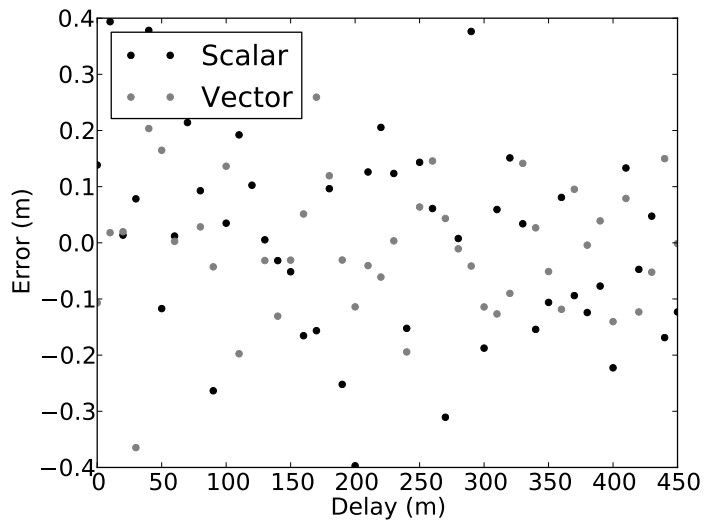


Figure D.31: Tracking error for 9 satellites and multipath-to-direct ratio 1.0.

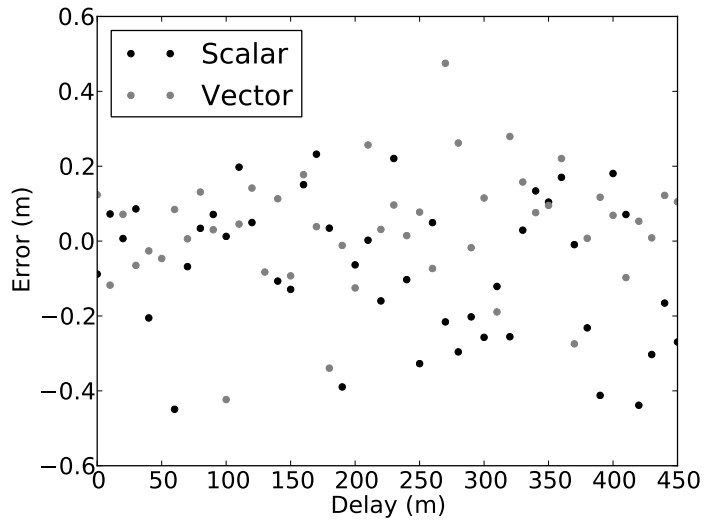


Figure D.32: Tracking error for 8 satellites and multipath-to-direct ratio 1.0.

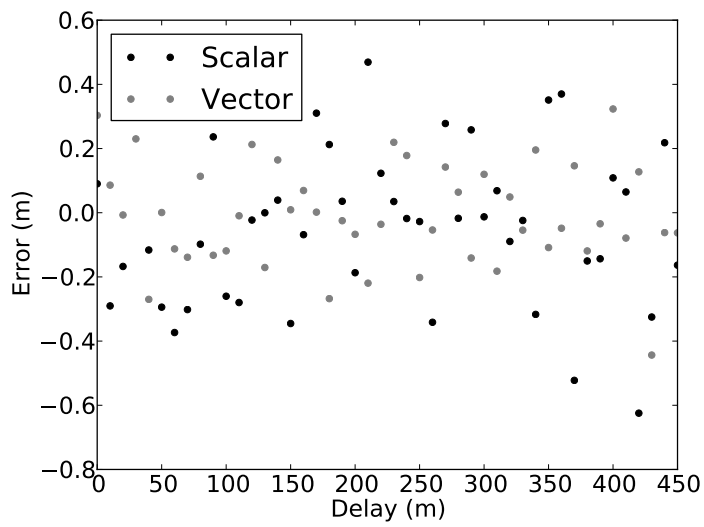


Figure D.33: Tracking error for 7 satellites and multipath-to-direct ratio 1.0.

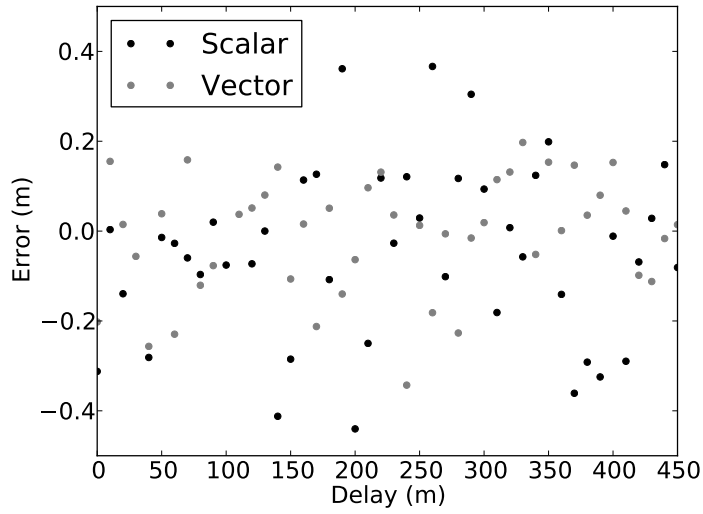


Figure D.34: Tracking error for 6 satellites and multipath-to-direct ratio 1.0.

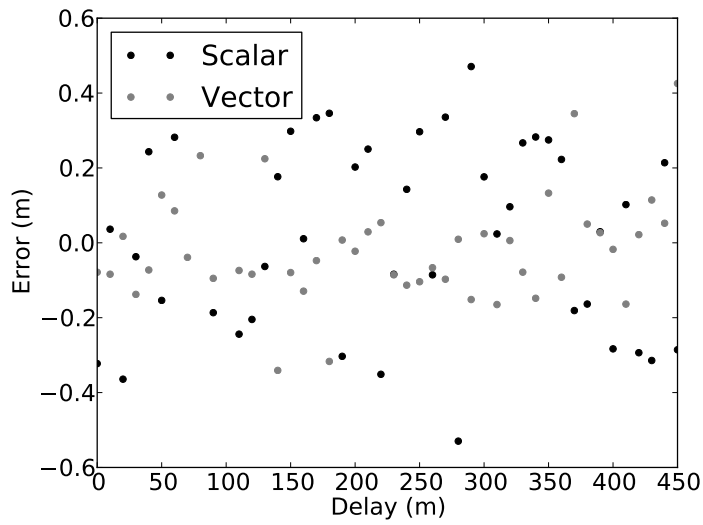


Figure D.35: Tracking error for 5 satellites and multipath-to-direct ratio 1.0.