



**THE DIFFERENTIAL VECTOR PHASE-LOCKED LOOP FOR GLOBAL
NAVIGATION SATELLITE SYSTEM SIGNAL TRACKING**

DISSERTATION

James J. Brewer, GS-14, DAF

AFIT-ENG-DS-14-J-2

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-DS-14-J-2

THE DIFFERENTIAL VECTOR PHASE-LOCKED LOOP FOR GLOBAL
NAVIGATION SATELLITE SYSTEM SIGNAL TRACKING

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Electrical Engineering

James J. Brewer, BSEE, MSEE

GS-14, DAF

June 2014

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Abstract

A novel differential vector phase-locked loop (DVPLL) is derived that takes GNSS code-phase and carrier-phase measurements from a base station and uses them to maintain an integer ambiguity resolved quality solution directly in the vector tracking loop of a rover receiver. The only state variables estimated and used to create the replica code and carrier signals from the base station measurements are three position and two clock states for a static test. Closing the individual loops solely through the navigation filter makes this a pure vector method. For short baselines, where differential atmospheric errors are small, the DVPLL can be used on single-frequency data. An L1-only live-sky static test was performed using the method resulting in a 3D accuracy of 5.3 mm for an 18.5 m baseline.

An acquisition algorithm is also developed to initialize the DVPLL. The algorithm performs a search in the space-time domain vice the measurement domain. An upper bound on the failure rate of the algorithm can be set by the user. The algorithm was tested on 24-h dual- and single-frequency CORS data sets with close to a 100% success rate and on a 15-min data set of single-frequency IF samples with a 100% success rate.

Acknowledgements

My first and greatest thanks go to God and His Son Jesus Christ for giving me everything I am and have. Nothing would have been possible without Him.

Doc Raq for doing way more than I could ever mention here as my adviser. Your genius for explaining very complex ideas in simple ways makes you one of the best teachers I've ever had. It still amazes me you had time for me as busy as you keep yourself. Dr. Mark Oxley and Dr. Marshall Haker for your sage guidance as members of my committee. Dr. Ken Fisher for making Kalman Filters so much fun to learn and the crew who helped me hypothetically defeat a certain tyrannical country.

Dr. Eric Vinande and Mark Smearcheck for setting up the tests and collecting many GBs of data. Tina Cole for making sure the proper forms are filled out correctly, getting books from the library mailed out to me, and keeping me on Doc Raq's busy schedule. Eric Lagier for being *the plant* to bounce my ideas off or just unwind over a frosty mug at the end of the day. ATIS, Spot, Smiley, Shock, between our weekly runs to Faux Mo's and witty banter in the outhouse you guys kept it real.

Shock, I really appreciate you, your wife and daughter making me a part of your family while I was away from mine. My mom who's been bragging about me since I was little inadvertently setting high bars for me along the way and my dad whose quiet example of hard work and unwavering integrity has greatly influenced my life.

I will always remember the trip home and eating cheese curds at Curly's in Lambeau Field with my lovely daughter as one of the highlights of my life. And especially the love of my life, my beautiful wife, your undying support and ability to hold down the home front, especially during our eight months apart, continues to amaze me. God Bless.

James J. Brewer

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
Table of Contents	vi
List of Figures	ix
List of Tables	xii
I. Introduction and Dissertation Overview	1
II. Background	3
2.1 Signal Description	3
2.1.1 Time Definitions	3
2.1.2 Signal-in-Space Description	5
2.1.3 Down Conversion and Sampling	5
2.2 Common Tracking Functions	7
2.2.1 Doppler Removal	8
2.2.2 Correlators	9
2.2.3 Accumulators	9
2.2.4 Discriminators	11
2.2.5 Loop Filters	14
2.2.6 Doppler Removal Through Discriminators as a Difference Operator	15
2.3 Tracking Loops	15
2.3.1 Scalar Tracking Loops	15
2.3.2 Vector Tracking Loops	17
2.3.2.1 Past Research Summary	18
2.3.2.2 Vector Delay Locked Loop/Vector Frequency Locked Loop	19
2.3.2.3 Past VDLL/VFLL Research	22
2.3.2.4 Vector Phase Locked Loops	27
III. Differential Vector Phase Locked Loop Tracking	33
3.1 Base Station Measurements	34
3.2 Rover pre-Processing	34
3.3 Translating Base Station Replica Signals to Create Rover Replica Signals .	35

	Page
3.3.1 Receiver Indicated Time	35
3.3.2 Creating the Replica Phase	37
3.3.3 Code and Data Bits	39
3.4 Kalman Filter to Update Rover Position, Velocity, and Time Offset Estimates	40
3.5 Results	50
3.5.1 Test 1	50
3.5.2 Test 2	56
3.5.3 Test 3	59
3.6 Conclusion	65
 IV. Differential Vector Phase Locked Loop Acquisition	 66
4.1 Past Research	66
4.1.1 Integer Ambiguity Resolution Methods	66
4.1.2 Ambiguity Function Method (AFM)	68
4.2 DVPLL Acquisition Algorithm	72
4.2.1 Stage 1: Widelane or L ₁ -Only Solutions	75
4.2.2 Stage 2: Individual L ₁ and L ₂ Measurement Solutions Around Widelane Solutions	78
4.2.3 Stage 3: Validation: Propagate Solutions in DVPLL and Eliminate Candidates Until One Left	80
4.2.3.1 Failure Rate	81
4.3 Results	82
4.3.1 24-h Test Using CORS Data	82
4.3.1.1 L ₁ /L ₂ Case	85
4.3.1.2 L ₁ -Only Case	89
4.3.2 15-min Test Using TRIGR as Rover and NovAtel as Base Station .	93
4.3.3 Conclusion	94
 V. Conclusions and Recommendations	 97
5.1 Research Contributions	97
5.1.1 DVPLL Tracking	97
5.1.2 DVPLL Acquisition	97
5.2 Future Work	98
5.2.1 Longer Baselines	98
5.2.2 High Accuracy in High Dynamics	98
5.2.3 Deep Integration with an IMU	99
5.2.4 Smoother	99
5.2.5 Other GNSS Signals	99
5.2.6 Real-Time DVPLL	99
5.2.7 Acquisition Tradeoffs	100

	Page
Appendix: Linear Combinations and the Effects of Removing Integer Offset	101
Bibliography	105

List of Figures

Figure	Page
2.1 Typical GPS Receiver Front End	6
2.2 Scalar Tracking Loop	7
2.3 Block Diagram of Doppler Removal Function	8
2.4 Block Diagram of Correlation Function	10
2.5 Graph of CA-code Autocorrelation Function	12
2.6 Typical Feedback Loop Using a Loop Filter	14
2.7 Tracking Loop Ontology	15
2.8 Example of Geometric Correlation Between Satellites	18
2.9 Vector Delay and Vector Frequency Locked Loop	22
2.10 Differential Vector Delay and Vector Frequency Locked Loop Following Chan and Petovello [11]	23
2.11 Vector Phase Locked Loop with Co-op Tracking	29
2.12 Vector Phase Locked Loop with Estimation of Atmospheric Errors Using Loop Filters	30
2.13 Vector Phase Locked Loop with Estimation of Atmospheric Errors Using a Kalman Filter	30
2.14 Vector Phase Locked Loop with Joint Code and Carrier Tracking	31
2.15 Vector Phase Locked Loop with Position Domain Joint Tracking	32
3.1 Differential Vector Phase Locked Loop Flow Chart	33
3.2 Extended Kalman Filter	40
3.3 Antenna Locations, North Marker is AFIT, South is AFRL, Map Data: Google, DigitalGlobe, State of Ohio, U.S. Geological Survey, USDA Farm Agency	51
3.4 AFIT Antenna Looking North-East	52

Figure	Page
3.5 AFRL Antenna Looking East	53
3.6 Sky Plot of Satellite Locations	53
3.7 Plot of CN0 Values for Each Receiver	54
3.8 Phase Discriminators - PRN 19 not used in solution	55
3.9 Position Errors and Epsilon	56
3.10 Looking North Toward CRPA and AFIT Ashtech Antenna	57
3.11 DVPLL and Single-Difference Horizontal Position Errors for Northeast CRPA Antenna from 4:54 PM Data Collect	58
3.12 DVPLL and Single-Difference Vertical Position Errors for Northeast CRPA Antenna from 4:54 PM Data Collect	59
3.13 Ambiguity-Resolved Single-Difference Phase Corrected for Surveyed Baseline Offset for Northeast CRPA Antenna from 4:54 PM Data Collect	60
3.14 Sky Plot for 4:54 PM Data Collect	60
3.15 Configuration for Test 3	61
3.16 Chokering Antennas on ANT Center Roof Looking North, <i>Blue</i> Antenna in Background, <i>Yellow</i> Antenna in Foreground	61
3.17 ANT Center Rooftop Antennas Single-Difference Phase Error rms vs Eleva- tion, 24-h period, 12-13 Dec 13	62
3.18 Sky Plot for 13 Dec 13, 15-min Data Collect	63
3.19 DVPLL 3D Position Errors, 13 Dec 13	64
3.20 TRIGR-NovaTel Relative Clock Frequency Offsets, 13 Dec 13	65
4.1 Probability Tree for DVPLL Acquisition	81
4.2 Single-Difference Phase Errors vs Elevation, 24-h period, 3 Sep 13	83
4.3 Single-Difference Phase Error rms vs Elevation, 24-h period, 3 Sep 13	83
4.4 Differential Vector Phase Locked Loop Flow Chart Using CORS Data	84

Figure	Page
4.5 3D Position Error of Final Solution L_1/L_2 , 24-h period, 3 Sep 13	86
4.6 Mean Time to Fix L_1/L_2 , 24-h period, 3 Sep 13	86
4.7 Minimum 3D Error of Candidates and #Satellites Tracked L_1/L_2 No Stage 3, 24-h period, 3 Sep 13	87
4.8 Percentage of Single-Epoch Solutions vs Satellites Tracked L_1/L_2 No Stage 3, 24-h period, 3 Sep 13	88
4.9 Average Number of Solutions at End of Stage 2 vs Satellites Tracked L_1/L_2 No Stage 3, 24-h period, 3 Sep 13	88
4.10 Distance from True Position to Second Best Solution for non-Single Epoch Trials L_1/L_2 , 24-h period, 3 Sep 13	90
4.11 Average Number of Solutions at End of Stage 1 vs Satellites Tracked L_1 -Only, 24-h period, 3 Sep 13	91
4.12 Mean Time to Fix L_1 -Only, 24-h period, 3 Sep 13	91
4.13 3D Position Error of Final Solution L_1 -Only, 24-h period, 3 Sep 13	92
4.14 Candidate 3D Position Errors vs Weighted Sum of Squares of Residuals Over Threshold, L_1 -Only, Single Epoch, 3 Sep 13	93
4.15 Sample 3D Position Error of Candidate Solutions vs Acquisition Time, 13 Dec 13	95
4.16 Sample Weighted Sum of Squares of Residuals Divided by Threshold vs Acquisition Time, 13 Dec 13	95

List of Tables

Table	Page
2.1 Past Research Summary	20
2.2 Definitions for Table 2.1	21
3.1 Antenna Reference Point IGS-08 Survey Coordinates for ANT Center Rooftop Antennas from OPUS 12-13 Dec 13	62
4.1 L_1/L_2 Single Epoch Results, 24-h Period, 3 Sep 13	89
4.2 L_1 -Only Acquisition Results, 24-h Period, 3 Sep 13	92
4.3 L_1 -only DVPLL Acquisition Success Rate, 15-min Period, 13 Dec 13	94
A.1 Parameter Values for Various Linear Combinations	104

THE DIFFERENTIAL VECTOR PHASE-LOCKED LOOP FOR GLOBAL NAVIGATION SATELLITE SYSTEM SIGNAL TRACKING

I. Introduction and Dissertation Overview

The geometric correlation of Global Positioning System (GPS) satellite vehicle (SV) signals was first exploited by Sennott and Senffner in 1991 as a way of overcoming cycle slips [62]. This was followed by Spilker's development of the vector delay locked loop (VDLL) in 1996 [55]. Vector tracking has seen a flurry of activity since these two pioneering papers were written. Taking advantage of the spatial correlation between satellites has opened vast frontiers of research. To date, vector-tracking research has focused on obtaining real-time solutions without the benefit of precise base station measurements. To make vector phase tracking possible, slow-varying phase errors on each channel must be accounted for in some way. This dissertation presents a novel method designed for a test and evaluation environment where sampled intermediate-frequency (IF) GNSS data can be post-processed. Under these conditions, base station measurements can be used in a differential vector phase locked loop (DVPLL) algorithm to obtain a position solution directly in the vector tracking loop of a rover receiver that has an accuracy comparable to an integer-resolved carrier-phase differential GPS solution. The phase errors are common to both receivers and, for the most part, cancel.

This dissertation derives a novel vector tracking technique that translates the code and carrier-phase measurements obtained from a receiver at a surveyed location (base station) to a Kalman-filter predicted location for a receiver at a different location (rover). The translated code and carrier-phase measurements are used to generate local replicas of the predicted signals for each channel of the rover. These replica signals are correlated with the

incoming signal obtained at the rover to generate errors. The errors are used by a Kalman filter to update its navigation state and clock offset states, completing the loop.

The DVPLL is a pure vector technique in that no local channel information is saved to be used in the next iteration. This is in contrast to other techniques, which use a combination of local-channel and vector information in a given loop.

As will be shown in Chapter 2, the DVPLL is unique in the literature. Several papers introduce vector carrier-phase tracking techniques. These techniques are based on the work of Zhodzishsky et al. [76]. However, none of these papers use differential carrier-phase measurements directly in the tracking loop. Only Chan and Petovello use differential corrections directly in the tracking loops [11]. These corrections are limited to code-phase and carrier-frequency measurements vice carrier-phase measurements. Carrier-phase measurements must be used to obtain an ambiguity-resolved differential carrier-phase quality solution. Only the DVPLL does this directly in the tracking loops.

The rest of the dissertation is organized as follows: Chapter 2 provides basic information on the GPS signal structure, the scalar tracking loop, and various vector tracking techniques. A literature review is detailed for each vector tracking technique. Chapter 3 introduces the DVPLL, derives the equations governing the translation of base station code and carrier-phase measurements to the rover location, and details the Kalman filter employed in this work. The chapter ends with results obtained using the technique. Chapter 4 provides a literature review of integer ambiguity resolution techniques and the ambiguity function method, introduces a signal acquisition method for the DVPLL, and ends with results. Chapter 5 concludes the dissertation and provides recommendations for future work.

II. Background

This chapter details the current state-of-art for vector tracking loops. The chapter starts by explaining the signal-in-space, down conversion, and digital sampling normally performed by a receiver front end. Next, a number of functions common to current scalar tracking loops are presented. The chapter continues by reviewing the traditional scalar tracking loop and various vector tracking loop mechanizations. The chapter contains a literature review integrated into each vector tracking section.

The equations presented in this chapter are derived using the global positioning system (GPS) Link 1 (L1) Coarse/acquisition-code (C/A-code) but are easily extended to any other GNSS signal.

2.1 Signal Description

2.1.1 Time Definitions.

In the following derivation, signals are defined either by the global navigation satellite system (GNSS) system time when the signal was transmitted or the system time when the signal arrived at the receiver's antenna phase center. This derivation uses notation similar to Kaplan and Hegarty [35]. The system time of transmission, also known as the true time, is denoted t_s . The system time of arrival is denoted t_u . The relationship between these two, for the i th satellite, is given by

$$t_u^i = t_s^i + \Delta t^i \quad (2.1)$$

where

t_u^i = system time i th satellite vehicle (SV) signal arrived at receiver (s)

t_s^i = system time signal transmitted by i th SV (s)

Δt^i = time for signal to transit from i th SV to receiver (s)

The transit time is given by

$$\Delta t^i = \frac{r^i}{c} + \tau_{prop}^i \quad (2.2)$$

where

r^i = range to i th SV accounting for earth rotation and propagation time (m)

c = speed of light (m/s)

τ_{prop}^i = propagation delay due to troposphere and ionosphere (s)

The propagation delay is different for the code and carrier since the ionosphere is a dispersive media. The code is delayed due to the ionosphere, while the phase is advanced, resulting in different values for τ_{prop}^i between code and carrier [35]. These values will be defined as τ_{code}^i for the code and τ_{carr}^i for the carrier.

Substituting (2.2) in (2.1), using the appropriate definitions for τ_{prop}^i , and rearranging terms yields the system time when the code and carrier were transmitted by the i th satellite as

$$t_{code}^i \triangleq t_{s,code}^i = t_u^i - \frac{r^i}{c} - \tau_{code}^i \quad (2.3)$$

$$t_{carr}^i \triangleq t_{s,carr}^i = t_u^i - \frac{r^i}{c} - \tau_{carr}^i \quad (2.4)$$

where

t_{code}^i = system time when code was transmitted by the i th SV (s)

t_{carr}^i = system time when carrier was transmitted by the i th SV (s)

In the subsequent sections t_u^i is redefined simply as t^i .

2.1.2 *Signal-in-Space Description.*

The signal emitted from a GNSS satellite consists of a carrier signal modulated by a known code signal and unknown data signal. For the GPS L1 CA-code signal the carrier frequency is 1575.42 MHz, the code is a 1023 length binary phase shift key (BPSK) gold code output at 1.023 MHz for a code length of 1 ms. The data are modulated on the signal at a rate of 50 bits per second. Using orthogonal codes in a code division multiple access (CDMA) scheme, all satellites share the same frequency.

The signal-in-space from the i th SV arriving at the receiver's antenna phase center is described by [35]

$$s_R^i(t^i) = A_R^i(t^i)C(t_{code}^i)D(t_{code}^i) \cos(\phi_T^i(t_{carr}^i)) \quad (2.5)$$

where

s_R^i = i th SV signal at receiver's antenna phase center (V)

A_R^i = i th SV received signal amplitude (V)

C = spreading code (unitless) at time of transmission

D = data (unitless) at time of transmission

ϕ_T^i = signal phase at time of transmission (rad)

2.1.3 *Down Conversion and Sampling.*

The signal, s_R^i , is then moved to an intermediate frequency through one or more stages of amplification, filtering, and mixing as shown in Figure 2.1.

The overall effect can be modeled as a single mixing stage. The mixing process results in terms that are the sum and difference of the incoming signal's carrier phase and the mixing signal's phase. The sum term is filtered out, leaving the difference term as follows [35]

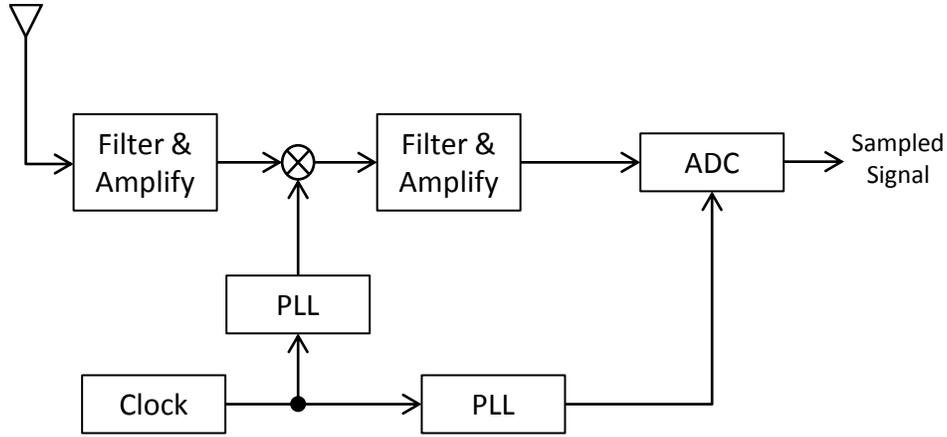


Figure 2.1: Typical GPS Receiver Front End

$$s^i(t^i) = A^i(t^i)C(t_{code}^i)D(t_{code}^i) \cos(\phi_T^i(t_{carr}^i) - \phi_m(t^i)) \quad (2.6)$$

$$= A^i(t^i)C(t_{code}^i)D(t_{code}^i) \cos(\phi^i(t_{carr}^i)) \quad (2.7)$$

where

s^i = i th satellite signal mixed to intermediate frequency (V)

A^i = intermediate frequency signal amplitude (V)

ϕ_m = mixer phase as a function of time (rad)

ϕ^i = intermediate frequency phase as a function of time (rad)

The intermediate frequency signal is then sampled and either processed immediately or recorded and processed at a later time. The mixer and sampler are typically driven by the same clock with the result that they both drift the same way. If the mixer and sampler are not driven by the same clock, the carrier and code will drift in different ways, making the carrier-aided code loop more difficult to implement [44].

2.2 Common Tracking Functions

This section is intended to be an introduction to the notation used in this dissertation rather than an in-depth treatise of the common functions. The word *function* in this section refers to a basic receiver operation that can be implemented either in hardware or software. Excellent texts on the traditional scalar tracking loop and the functions involved can be found in Parkinson and Spilker [55], Misra and Enge [49], or Kaplan and Hegarty [35].

Figure 2.2 is a flowchart of a typical scalar tracking loop implementation. The functions shown along the left hand side of each channel *page* are explained in the Sections 2.2.1 to 2.2.5.

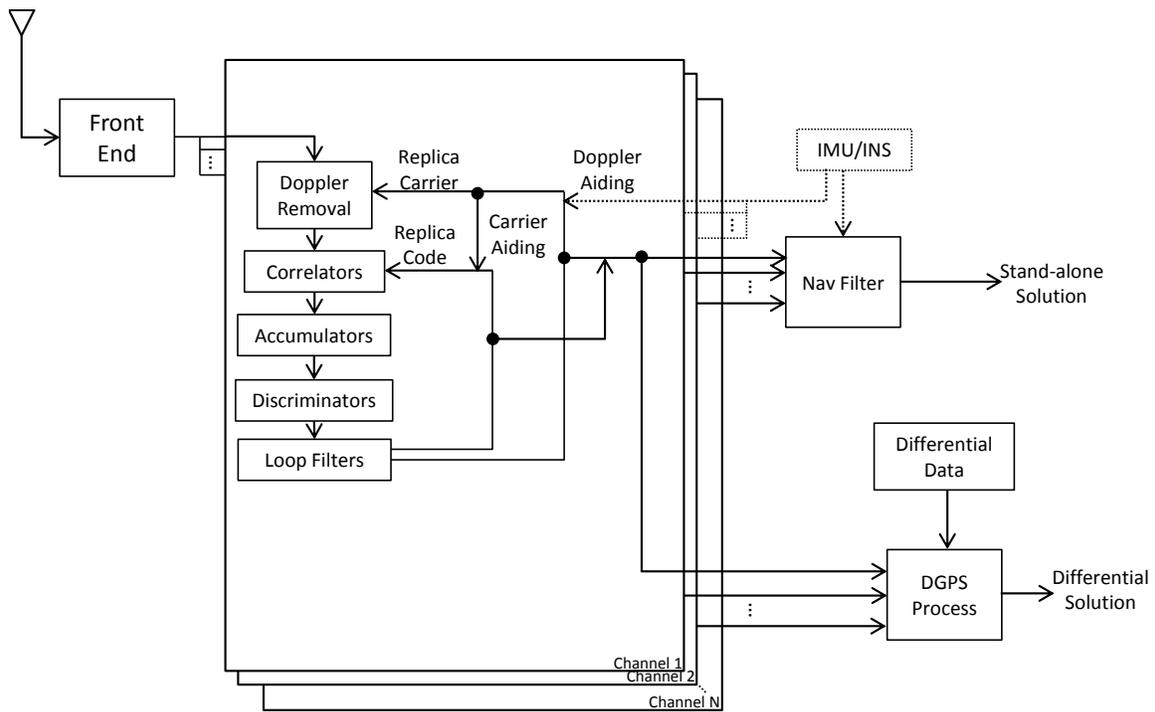


Figure 2.2: Scalar Tracking Loop

2.2.1 Doppler Removal.

The Doppler removal function is shown in Figure 2.3. Doppler removal is performed by multiplying the sampled signal by the cosine and sine of a replica carrier. This multiplication results in cosine and sine terms with arguments that are the sum and difference of the sampled signal's phase and the replica carrier. The sum terms will be filtered out in later stages, leaving the difference terms. The resulting two difference signals are known as the in-phase (cosine) and quadrature (sine) signals. Ignoring the sum terms, the resulting signals are modeled as

$$I^i(t^i) = 0.5A^i(t^i)C(t_{code}^i)D(t_{code}^i) \cos(\phi^i(t_{carr}^i) - \hat{\phi}^i(t^i)) \quad (2.8)$$

$$= 0.5A^i(t^i)C(t_{code}^i)D(t_{code}^i) \cos(\delta\phi^i(t^i)) \quad (2.9)$$

$$Q^i(t^i) = 0.5A^i(t^i)C(t_{code}^i)D(t_{code}^i) \sin(\delta\phi^i(t^i)) \quad (2.10)$$

where

I^i = in-phase signal (V)

$\hat{\phi}^i$ = replica carrier (rad)

$\delta\phi^i$ = phase difference (rad)

Q^i = quadrature signal (V)

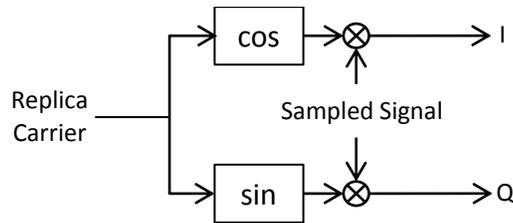


Figure 2.3: Block Diagram of Doppler Removal Function

2.2.2 Correlators.

The I and Q signals are then multiplied by early, prompt and late code replicas, as shown in Figure 2.4. The early and late code replicas are typically spaced a spreading code chip apart from each other (advanced and delayed by half a chip each). This results in 6 signals with equations given by

$$IP1^i(t^i) = 0.5A^i(t^i)C(t_{code}^i)C(\hat{t}_{code}^i)D(t_{code}^i) \cos(\delta\phi^i(t^i)) \quad (2.11)$$

$$IE1^i(t^i) = 0.5A^i(t^i)C(t_{code}^i)C(\hat{t}_{code}^i - 0.5d)D(t_{code}^i) \cos(\delta\phi^i(t^i)) \quad (2.12)$$

$$IL1^i(t^i) = 0.5A^i(t^i)C(t_{code}^i)C(\hat{t}_{code}^i + 0.5d)D(t_{code}^i) \cos(\delta\phi^i(t^i)) \quad (2.13)$$

$$QP1^i(t^i) = 0.5A^i(t^i)C(t_{code}^i)C(\hat{t}_{code}^i)D(t_{code}^i) \sin(\delta\phi^i(t^i)) \quad (2.14)$$

$$QE1^i(t^i) = 0.5A^i(t^i)C(t_{code}^i)C(\hat{t}_{code}^i - 0.5d)D(t_{code}^i) \sin(\delta\phi^i(t^i)) \quad (2.15)$$

$$QL1^i(t^i) = 0.5A^i(t^i)C(t_{code}^i)C(\hat{t}_{code}^i + 0.5d)D(t_{code}^i) \sin(\delta\phi^i(t^i)) \quad (2.16)$$

where

P, E, L = signal correlated with prompt, early or late code

\hat{t}_{code}^i = replica code time (s)

d = time difference between early and late replica code (s)

2.2.3 Accumulators.

The outputs of the correlators are then filtered using accumulators which simply add the values over a given interval. The following derivation will focus on the in-phase prompt signal and apply the final result to the other five signals. The summation of the in-phase

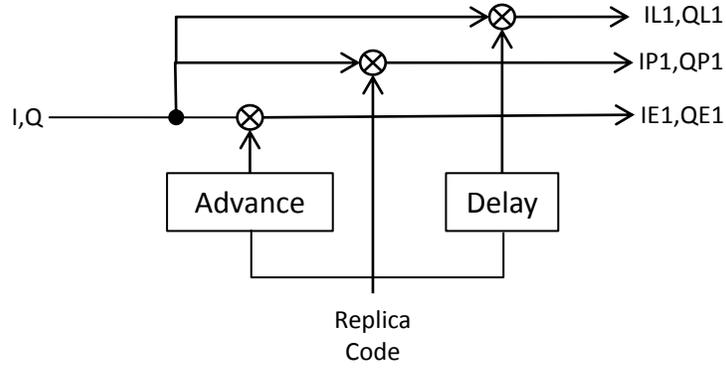


Figure 2.4: Block Diagram of Correlation Function

signal correlated with the prompt code is given by

$$IP^i(j) = \sum_{k=0}^{\frac{\Delta T(j)}{T_s}} IP1^i(t_0(j) + kT_s) \quad (2.17)$$

$$= \sum_{k=0}^{\frac{\Delta T(j)}{T_s}} 0.5A^i(t_0(j) + kT_s)C(t_0(j) + kT_s)C(\hat{t}_{code}^i)D(t_0(j) + kT_s) \cos(\delta\phi^i(t_0(j) + kT_s)) \quad (2.18)$$

where

$IP^i(j)$ = j th accumulated in-phase signal correlated with prompt code (V)

$t_0(j)$ = time at beginning of the j th summation interval (s)

$\Delta T(j)$ = accumulation period of the j th interval (s)

j = interval index

T_s = sampling interval (s)

Note the switch between a time-based output and an index-based output. Next, the assumption is made that the phase error, $\delta\phi$, and the difference between the true code time and replica code time, denoted by τ , are approximately constant across the accumulation

period. Furthermore, the data bit is assumed constant across accumulation period yielding

$$IP^i(j) \approx M^i(j)R(\tau^i(j))D(j) \cos(\delta\phi^i(j)) \quad (2.19)$$

$$IE^i(j) \approx M^i(j)R(\tau^i(j) - 0.5d)D(j) \cos(\delta\phi^i(j)) \quad (2.20)$$

$$IL^i(j) \approx M^i(j)R(\tau^i(j) + 0.5d)D(j) \cos(\delta\phi^i(j)) \quad (2.21)$$

$$QP^i(j) \approx M^i(j)R(\tau^i(j))D(j) \sin(\delta\phi^i(j)) \quad (2.22)$$

$$QE^i(j) \approx M^i(j)R(\tau^i(j) - 0.5d)D(j) \sin(\delta\phi^i(j)) \quad (2.23)$$

$$QL^i(j) \approx M^i(j)R(\tau^i(j) + 0.5d)D(j) \sin(\delta\phi^i(j)) \quad (2.24)$$

where

M^i = magnitude of the i th signal (V)

R = code autocorrelation function (unitless)

The idealized code autocorrelation function, for most GNSS signals, is a triangular-shaped function as depicted in Figure 2.5 for the GPS C/A-code. The equation for the autocorrelation function, $R(x)$, is given by

$$R(x) = \begin{cases} 1 - |x| & \text{if } |x| \leq 1 \\ -\frac{1}{1023} & \text{if } |x| > 1 \end{cases} \quad (2.25)$$

2.2.4 Discriminators.

The previous functions are common to most tracking algorithms (a noted exception are tracking loops based on maximum likelihood techniques [12]). The implementation may vary from that presented here but each of the previous functions can be found in some form. In contrast, most, but not all, mechanizations use discriminators to estimate the tracking errors τ and $\delta\phi$. Many different discriminators are used in the literature. Four popular examples are explained here and the reader is left to research others if interested.

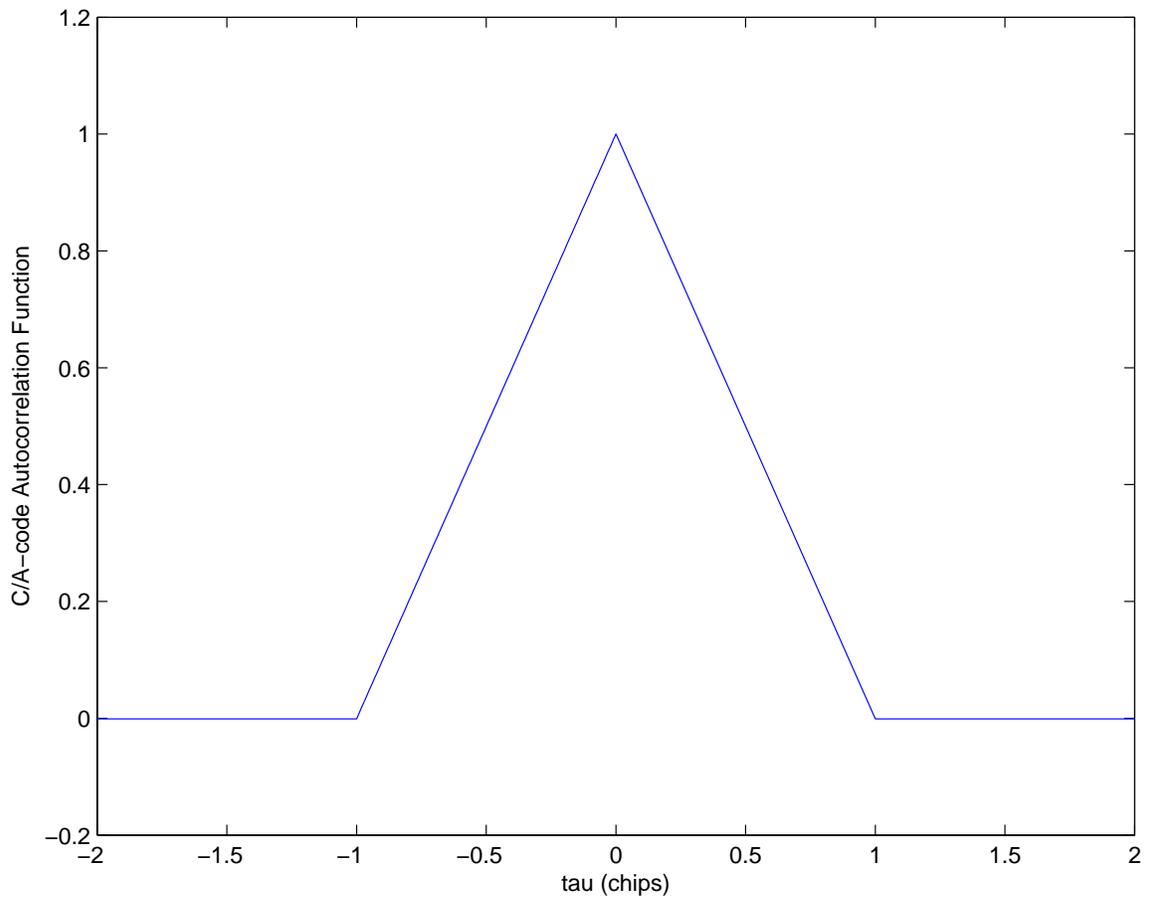


Figure 2.5: Graph of CA-code Autocorrelation Function

The first discriminator, the normalized non-coherent early-minus-late envelope, estimates the code tracking error, τ . The equation for this discriminator is given by [35]

$$\hat{\tau}^i = \frac{1}{2} \frac{\sqrt{IE^{i^2} + QE^{i^2}} - \sqrt{IL^{i^2} + QL^{i^2}}}{\sqrt{IE^{i^2} + QE^{i^2}} + \sqrt{IL^{i^2} + QL^{i^2}}} \quad (2.26)$$

where

$$\hat{\tau}^i = \text{estimate of code tracking error (chips)}$$

The next discriminator is the Costas Loop phase discriminator. This discriminator is insensitive to the 180 degree phase shifts introduced by the unknown 50 Hz data bits. The discriminator is given by [35]

$$\hat{\delta\phi}^i = \text{atan}(QP^i/IP^i) \quad (2.27)$$

where

$$\hat{\delta\phi}^i = \text{phase error estimate (rad)}$$

This discriminator is used if the data bit is unknown. For a known data bit the four quadrant inverse tangent can be used as follows [35]

$$\hat{\delta\phi}^i = \text{atan2}(QP^i, IP^i) \quad (2.28)$$

where

$$\text{atan2} = \text{four quadrant arc tangent}$$

Each of these discriminators are nonlinear and, in general, result in a non-Gaussian estimate [35].

The last discriminator explained here is the frequency discriminator. This discriminator estimates the frequency tracking error vice the phase error. A common version is the maximum likelihood estimator given by [35]

$$\hat{f}^i(j) = \frac{\text{atan2}(\text{cross}, \text{dot})}{\Delta T(j)} \quad (2.29)$$

where

\hat{f}^i = frequency error estimate (rad/s)

$$cross = IP^i(j-1)IP^i(j) + QP^i(j-1)QP^i(j)$$

$$dot = IP^i(j-1)QP^i(j) - IP^i(j)QP^i(j-1)$$

2.2.5 Loop Filters.

The objective of a tracking loop is to create a replica signal that matches the input signal. To this end, the difference between the original signal and replica signal is fed back as an error steering the replica signal in the correct direction. However, each discriminator listed in the previous section is a noisy estimate of the error signal. Loop filters are used to reduce this noise and create a better estimate of the replica signal. Figure 2.6 shows a typical feedback loop using a loop filter.

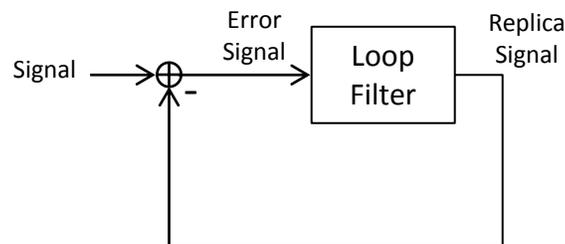


Figure 2.6: Typical Feedback Loop Using a Loop Filter

A very basic loop filter is a simple integrator. The discriminator output is scaled and integrated to give the replica signal. The magnitude of the scaling sets the filter's bandwidth. This filter is called a first-order loop filter and is typically used for the code tracking loop. There are, of course, second- and third-order loop filters each having two and three integrators, respectively. The second-order loop filter is typically used with frequency tracking and the third-order loop filter is typically used for phase tracking [35].

There is a trade-off between reducing noise (reducing bandwidth) and being able to track dynamics (increasing bandwidth). This trade-off becomes a parameter in receiver design.

2.2.6 Doppler Removal Through Discriminators as a Difference Operator.

The output of the *Discriminators* is the average difference of the incoming carrier and code and the replicas.

2.3 Tracking Loops

Figure 2.7 shows the ontology of the current state of tracking loop technology and provides an overview of this section.

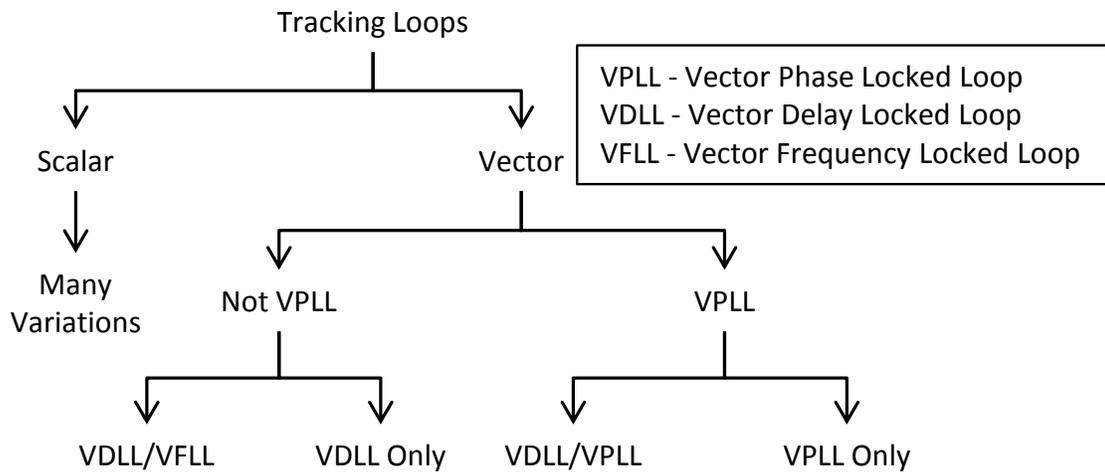


Figure 2.7: Tracking Loop Ontology

2.3.1 Scalar Tracking Loops.

As previously mentioned, the prominent method of tracking GPS satellites is known as the scalar tracking loop. Figure 2.2 depicts this method. In the figure, *pages* exist for every

channel. Anything on a *page* is associated with a single tracking channel and anything off the *pages* needs information from, or feeds information to, all the channels. With this in mind, the figure clearly shows that each satellite is tracked independently of the other satellites. In fact, for recorded intermediate frequency data, a popular method of processing is to track each satellite one at a time through the whole data set. The replica code, replica carrier, and navigation data bit estimates are then brought together to estimate the position and time.

Starting down the left side of Figure 2.2 and using the nomenclature introduced in the previous section, the replica carrier and sampled data are multiplied in the Doppler removal block to create I and Q values. These values are then correlated with early, prompt, and late versions of the replica code and the resulting values accumulated. Discriminators transform the accumulated values into carrier and code error estimates that are input to loop filters to create new estimates of the replicas, thus completing the tracking loop.

Carrier-Aided Code Loop. Once carrier lock is achieved, many implementations use the highly precise carrier measurements to aid the code loop. This is represented by the line labeled *Carrier Aiding* in Figure 2.2. This aiding greatly lowers the dynamics on the code loop, allowing for a commiserate reduction in the bandwidth of the loop filters and, hence, a reduction in the noise present in the code measurements.

Doppler Aiding. Mechanizations that include an inertial navigation system (INS) often aid the carrier loop using the velocities derived from an inertial measurement unit (IMU). In Figure 2.2, the INS, IMU and Doppler aiding have dashed lines indicating these features are not always present.

Differential Processing. Many times the replica carrier and replica code are used, along with replica carrier and replica code measurements from a receiver at a known location, to perform differential processing. This processing removes many of the errors common to both receivers and provides a highly precise solution especially if a kinematic

carrier phase solution can be realized. Differential processing is shown in the lower right of Figure 2.2.

Phase vs Frequency Locked Loop. There are two ways to close the carrier loop. The first is to use a phase discriminator and drive the difference between the phases of the incoming and replica signals to zero, thus matching the phases of the two signals. This approach is the most precise method of tracking the carrier and is known as a phase-locked loop (PLL). However, the PLL is also susceptible to losing lock due to noise and dynamics. In high-dynamic or high-noise environments an alternative method of tracking is to use a frequency-locked loop (FLL). The FLL uses the frequency discriminator of (2.29). The frequency of the replica carrier signal is driven to match that of the incoming signal, leaving the phase difference to wander. During acquisition, a receiver normally starts out using the FLL method early and then switches to a PLL implementation once the Doppler frequency is tracked [35].

2.3.2 Vector Tracking Loops.

In contrast to scalar tracking loops, vector tracking loops are characterized by their exploitation of the geometric correlations between satellite tracking channels. As seen in Figure 2.8, if two satellites are close together in angle in the sky and the receiver moves towards one, it will also move towards the other. This is the correlation that is leveraged in a vector tracking loop.

The geometric correlation is leveraged by projecting from the N -dimensional individual channel domain to the 4-dimensional time-space domain. During the projection process individual channel information is lost. In a pure vector implementation, no individual channel information is retained to use in creation of the replica signals. All vector phase locked loop implementations, uncovered in the literature, retain some

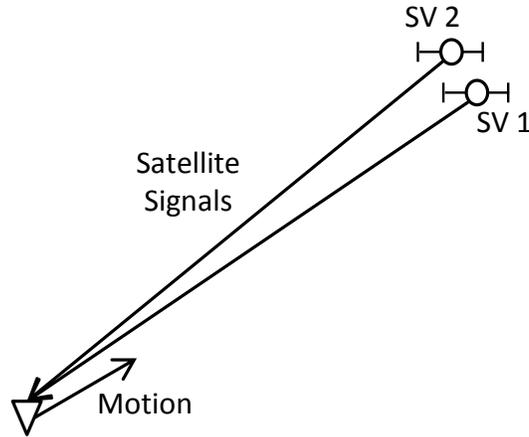


Figure 2.8: Example of Geometric Correlation Between Satellites

individual channel information, however, to account for slowly changing biases (errors) in the channel.

Vector tracking loops are named by how the geometric correlation is leveraged. As the name implies, the vector delay locked loop (VDLL) matches the replica code of all channels simultaneously by exploiting their geometric correlation. In a similar manner, the vector frequency locked loop (VFLL) matches the replica carrier frequency, and the vector phase locked loop (VPLL) matches the replica carrier phase.

2.3.2.1 *Past Research Summary.*

Table 2.1 tabulates past research dealing with vector tracking found by the author. Table 2.2 defines the acronyms used in Table 2.1. The first column of Table 2.1 is the author(s) and a reference to the work, the second is the year(s) the work was accomplished, the third is the primary organization responsible for the work, the fourth is the name given the method by the authors of the paper. The fifth, sixth, and seventh columns show whether the method is scalar or vector for the particular loop, delay (DLL), frequency (FLL), or phase (PLL). The eighth column is the position domain filter (PDF) type specifying loop filter (LF), Kalman filter (KF), maximum likelihood (ML), or least squares (LS). The

ninth column shows the measurement type used in the filter, either I s and Q s (IQ), or discriminator values (D). The tenth column tells whether the method integrates an IMU. The final column designates whether the method uses differential corrections directly in the vector loop and the type of corrections, code (C), frequency (F), or phase (P).

As shown in Table 2.1, there has been a lot of research into VDLL/VFLL tracking loops and very little into the pure VPLL. It also shows that only two methods use differential corrections directly in the tracking loop. Of those two, only the method proposed in this dissertation uses differential phase measurements directly in the tracking loops.

The reader is encouraged to refer back to Table 2.1 while proceeding through the rest of this chapter.

2.3.2.2 Vector Delay Locked Loop/Vector Frequency Locked Loop.

Sennott and Senffner [62] were the first to exploit the geometric correlation of GPS signals to prevent cycle slips in the carrier signal. Spilker [55] closed the code loop through the navigation filter, giving the first introduction to the VDLL. Figure 2.9 shows a flow chart of a VDLL/VFLL system. The left side of each tracking channel is the same as the scalar case; however, the similarity ends there. Each channel's estimate of the code time offset and frequency tracking error are fed as measurements into the navigation filter. The navigation filter uses these measurements to update the navigation states (position, velocity, etc.), receiver clock bias and drift, and any other estimated errors. The velocity is then projected back to the line of sight of each satellite, the satellite's velocity and receiver clock drift are added, and the result is then used to create a replica frequency. The replica frequency is integrated to create a replica phase and the phase loop is closed. For the code loop, the estimated position and satellite's position are subtracted to create a range. The estimated receiver clock offset and drift are then added to create a pseudorange. The replica code is then derived from this pseudorange estimate and the code loop is closed. To summarize, both the code and frequency loops are closed through the navigation filter.

Table 2.1: Past Research Summary

Author(s) [Reference(s)]	Year(s)	Organization(s)	Name Given	DLL	FLL	PLL	PDF	Meas	IMU	Diff
Sennott and Senffner [62]	91	Bradley U	Coupled	?	?	?	LF	IQ		
Spilker [55]	96	Stanford Telecom	VDLL	V		S	KF	D		
Zhodzishsky et al. [76]	98	Javad	Co-op Track	S		V,S	LF	D		
Gustafson, Dowdle et al. [26–28]	00/01/03	Draper	Deeply Integrated	V	V		KF	IQ	Y	
Abbott and Lillo [1]	03	Aerospace Corp	UltraTight	V			KF	IQ	Y	
Soloviev et al. [63–66]	04/08/10/11	Ohio U	Deeply Integrated	V		V	KF	D	Y	
Pany... Eissfeller et al. [52, 53]	05/06	U FAF Munich	Deeply Integrated	V	V		KF	D		
Petovello, Lachapelle [57]	06	U of Calgary	Ultra-Tight	V	V	S	KF	D		
Closas et al. [12, 13]	07/09	U of Catalonia	Max Likelihood		V		ML			
Groves et al. [24]	07	QinetiQ	Deeply Integrated	V	V		KF	D	Y	
Petovello... Lachapelle et al. [58, 59]	07/08	U of Calgary	Ultra-Tight	V	V	S	KF	D	Y	
Kiesel et al. [36]	08	U of Karlsruhe	VFLL-PLL	V	V	S	LS	D		
Lashley, Bevil, Hung [39]	08	Nav Tech/Auburn	Deeply Integrated	V	V		KF	D	Y	
Giger, Henkel, Günther [23]	09	U of Munich	MC-MS-VPLL	S		V	KF	D		
Henkel, Giger, Günther et al. [31, 32]	09	Stanford/U of Munich	MC-MS-VPLL	S		V	LF	D		
Lashley, Bevil, Hung [40]	09	Auburn	VDFLL	V	V		KF	D		
Pany... Eissfeller et al. [50, 54]	09/10	Various	Ultra-Tight/DINGPOS	V	V		KF	D	Y	
Edwards... Bevil et al. [17]	10	Auburn	Deeply Integrated	V	V		KF	D	Y	
Giger, Henkel, Günther [19, 20, 22]	10/11	U of Munich	Position Domain Track	V	V		KF	D		
Hui et al. [33, 34]	10	Hefei U	VDFLL	V	V		KF	D		
Nunes [51]	10	Ins of Telecom, Lisbon	Low-Complex	V			ML			
Weill et al. [43, 72]	10/11	Various	Max Like VTL	V	V		ML			
ZhenZhen et al. [75]	10	U of Def Tech, China	VDLL	V		S	KF	D		
Chan, Petovello [10, 11]	11/13	U of Calgary	Ultra-Tight	V	V	S	KF	D	Y	C/F
Liu et al. [44–46]	11/13	Tsinghua U, China	Direct Position	V	?		KF	D		
Wang et al. [71]	11	Beijing Jiaotong U	Deeply Integrated	V	V		KF	D	Y	
Zhao and Akos [74]	11	U of Colorado	VLL	V	V		KF	D		
Langer, Kiesel et al. [38]	12	U of Karlsruhe	Deeply Coupled	V	V		KF	D	Y	
Peng, Morton et al. [56]	12	Miami U	VTL	V,S	V,S		KF	D		
Brewer proposed		AFTT	DVPLL	V		V	KF	D		C/P

Note: See Table 2.2 for Definitions

Table 2.2: Definitions for Table 2.1

Acronym	Definition	Meaning (if applicable)
DLL	Delay Locked Loop	
FLL	Frequency Locked Loop	
PLL	Phase Locked Loop	
PDF	Position Domain Filter	Type of filtering performed on position domain estimates
Meas	Measurement Type	
IMU	Inertial Measurement Unit Used?	
Diff	Differential Type	
VDLL	Vector Delay Locked Loop	
VFLL	Vector Frequency Locked Loop	
PLL	Phase Locked Loop	
VPLL	Vector PLL	
MC-MS-VPLL	Multi-Constellation Multi-Satellite VPLL	
DINGPOS	Acronym not defined by authors	
VTL	Vector Tracking Loop	
DVPLL	Differential Vector Phase Locked Loop	
V	Vector	Parameter tracked by projecting from channel to position domain, filtering/estimating, and transforming back.
S	Scalar	Parameter tracked on a channel-by-channel basis.
V,S	V and S	A combination of Vector and Scalar such as High Frequency Vector and Low Frequency Scalar
LF	Loop Filter	
KF	Kalman Filter	
ML	Maximum Likelihood	
LS	Least Squares	
IQ	In-phase and Quadrature Accumulations	
D	Discriminator	
Y	Yes	
C	Code	
F	Frequency	
P	Phase	

The IMU, if present, is typically used to create more accurate position and velocity reference trajectories in the next iteration by propagating forward the best estimate of the navigation state at that point. This creates better estimates of the replica code and carrier.

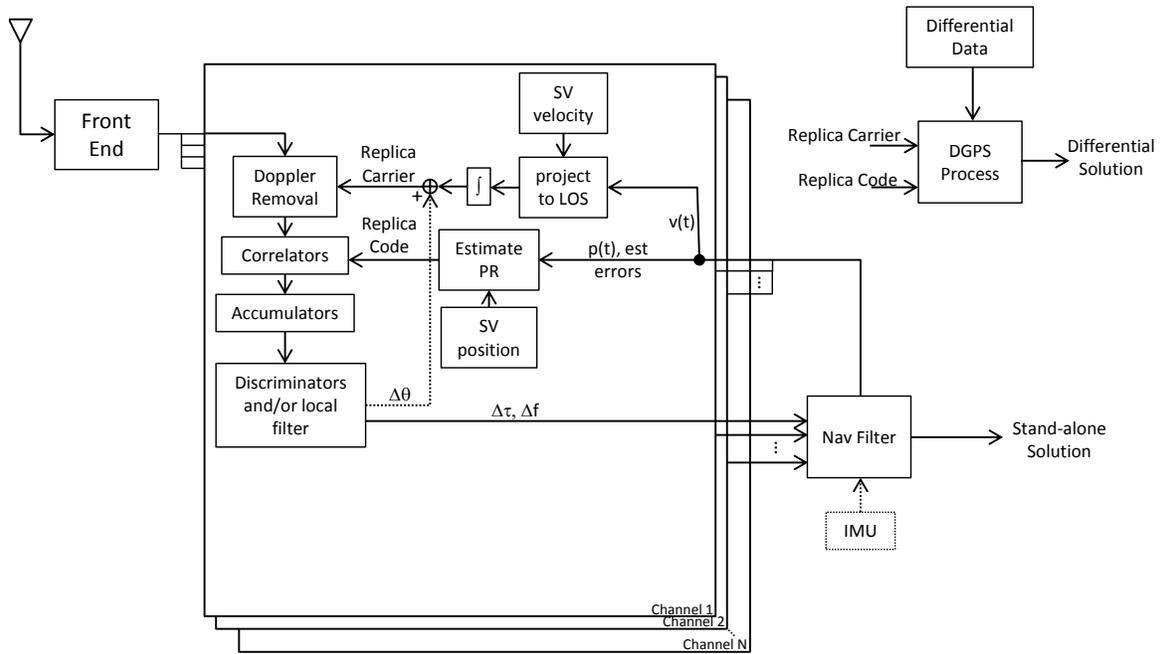


Figure 2.9: Vector Delay and Vector Frequency Locked Loop

2.3.2.3 Past VDLL/VFLL Research.

Gustafson et al. developed a VDLL solution coupled with an IMU through a centralized Kalman filter. The authors showed the joining of the two yielded a 15 dB performance advantage over conventional approaches [26–28].

Abbott and Lillo introduced a VDLL coupled with an IMU using a federated approach [1]. Higher rate pre-filters estimated range and range-rate information which is passed to a lower rate navigation filter.

Pany et al. developed a VDLL/VFLL with code and frequency discriminators and used it to better estimate signal power in bad signal conditions [52, 53]. The same group also developed a VDLL/VFLL coupled with an IMU and other sensors and demonstrated the ability to track a signal down to 1.5 dB-Hz carrier-to-noise-density ratio (C/N0) using long coherent integration times [50, 54].

Petovello and Lachapelle compared different local filter implementations for use in a cascaded vector-tracking approach [57]. The vector filter in question is a VDLL/VFLL mechanization with local phase tracking. Petovello et al. also added an IMU to a VDLL/VFLL mechanization with local phase tracking to show it gave a 7dB margin in maintaining kinematic quality carrier phase track over standard tracking loops [58]. Petovello then went on to extend the coherent integration time from 20 ms to 80 ms but demonstrated only improvement in tracking margin [59]. Chan and Petovello augmented a VDLL/VFLL with differential code and frequency measurements from a base station receiver as well as range measurements from an ultra-wideband device with minor improvements in tracking performance [10, 11]. A flowchart of this differential code method is shown in Figure 2.10. This is the only method found by the author to incorporate differential GPS measurements directly in the tracking loops.

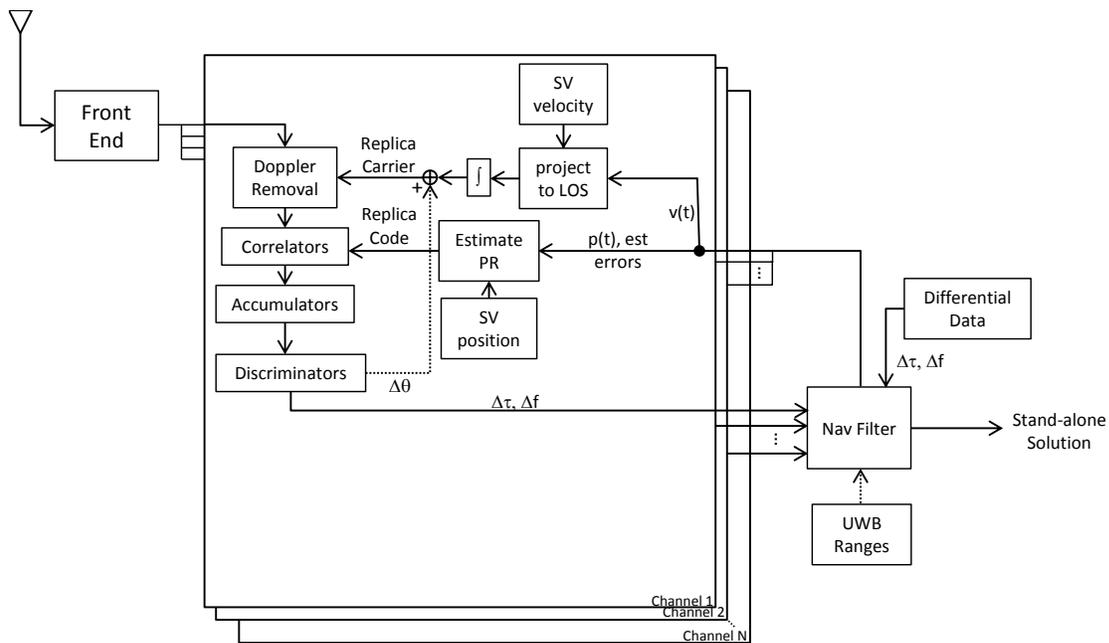


Figure 2.10: Differential Vector Delay and Vector Frequency Locked Loop Following Chan and Petovello [11]

DiEsposti [16] and Axelrad et al. [2, 3] presented methods of acquisition that take advantage of geometric correlations between satellites. These authors did not present a tracking method but are mentioned here for completeness. DiEsposti did mention that incorporating phase measurements from a receiver at a known location would help shorten the acquisition time.

Closas et al. used maximum likelihood techniques to directly determine the position and velocity from the input data [12]. They showed the maximum likelihood technique helps better mitigate multipath, compared to other vector techniques, in individual channels since correlation across channels are leveraged even within the integration period. The maximum likelihood method also removes the intermediate step of estimating the code delay and frequency offset parameters. The same authors then went on to develop a Cramér-Rao lower bound for scalar loops and the direct positioning technique, and showed the two are equivalent for clean signals of equivalent C/N_0 on all satellites [13]. In the case of lower C/N_0 on a subset of satellites or in the presence of multipath, however, the direct positioning technique had a lower Cramér-Rao lower bound.

Groves et al. developed a version of the VDLL/VFLL aided by an IMU and showed it to be an optimum integration architecture in applications where phase precision is unimportant [24].

Kiesel et al. designed a VDLL that had a local PLL with associated loop filter that was aided by a VFLL [36]. This method allows the receiver to operate using scalar tracking of phase under normal conditions but switch to vector frequency tracking for those channels with low C/N_0 .

Lashley et al. developed a VDLL/VFLL coupled with an IMU and studied the effects of noise, dynamics, and IMU quality on navigation performance compared with the performance without an IMU [39]. The version with a tactical-grade IMU tracked through an 8-g turn at 16 dB-Hz C/N_0 , while the version without an IMU could only track

through the turn down to 19 dB-Hz. Neither could track down to 14 dB-Hz. The same authors also detailed the same VDLL/VFLL only without an IMU in [40]. In [41] the same authors compared different architectures of a VDLL/VFLL coupled with an IMU. They compared the effects of using a federated versus centralized Kalman filter and the effects of scalar versus vector tracking loops. They showed that the centralized and federated filters are similar performance and that scalar loops performed poorer than both Kalman filter types, but the performance difference shrinks as the C/N0 declined. The C/N0 for all satellites was reduced at the same time as would happen in an interference environment vice selected satellites as would happen in a blockage situation. In [42] the same authors developed a method of comparing vector to scalar loops where everything is equal except the exploitation of geometric correlation in the vector algorithm. The paper asserts a 6 dB improvement in tracking threshold for vector tracking with an 11 satellite constellation.

Won et al. compared different approaches to Kalman filter design for vector tracking loops and concluded that different Kalman filter methods all performed similarly to each other and were better than the scalar method [73]. The authors also concluded that the Kalman filter methods provide 2-3 dB in tracking improvement, with the rest of the improvement coming from vector loop closure.

Edwards et al. described a VDLL/VFLL design and then implemented this design in hardware [17]. The hardware is not capable of working in real time, but the authors were confident the ability to do so would materialize in the near future.

Hui and Jingshu performed analyses of vector tracking that assumes a VDLL/VFLL architecture for weak signals and high dynamics [33, 34]. They showed that the vector method tracks well when a subset of satellites is weaker than the other satellites being tracked. The authors also showed the vector method works well under high-dynamic conditions.

Nunes et al. developed a VDLL implementation for low-dynamic applications where they constrained the position to be within the current cell and nearest neighbor cells [51]. A search is made through these cells to choose the one with the highest tracking power. This is equivalent to a discrete maximum likelihood method. The authors concluded this is a low-cost design for indoor applications.

Weill applied maximum likelihood techniques to code phase and frequency tracking in a VDLL/VFLL and listed the advantages for weak signal tracking [72]. Weill named this technique a maximum likelihood vector tracking loop. Lin et al. implemented these techniques in a real software receiver and performed a field test. The tests showed the navigation domain technique outperforms a centralized vector-based tracking loop for shorter integration periods [43]. However, the performance gains lessens as the integration period is lengthened.

Zhenzhen et al. developed a VDLL with a centralized Kalman filter and showed how a blocked satellite can be immediately tracked, once it became visible again, using this implementation [75]. The authors went on to demonstrate that a scalar tracking loop cannot immediately regain lock under the same conditions.

Liu et al. introduced a technique that jointly estimated the replica signals by taking advantage of the fact that the code and carrier rates of each channel are related, even across the integration period [44]. They used a maximum likelihood technique to estimate synchronization parameters in a joint vector discriminator and then fed these parameters to a Kalman filter. They showed that this helps reduce errors due to cross-correlation between strong and weak satellites and improves the ultimate position solution. The author of this dissertation could not determine if true phase lock occurred using this method. Liu et al. went on to propose a direct position tracking loop and explained it using the code tracking portion [46]. The method uses *early-late* I and Q values in the x , y , z and time directions. The sum of squares of the I and Q values were then summed for all satellites and the results

used as a geometric discriminator in the normal way. The method is shown to improve tracking of weak signals under dynamics [45].

Wang et al. described a VDLL/VFLL coupled with an IMU and showed how it worked well in a train navigation system [71]. A centralized Kalman filter uses loop-filtered discriminator outputs from the individual channels as measurements.

Zhao and Akos implemented a VDLL/VFLL and gave some advice on tuning [74]. They performed two simulator tests and a drive test to show the benefits of a VDLL/VFLL over a conventional scalar loop.

Langer, Kiesel et al. developed a VDLL/VFLL with an IMU for pedestrian navigation indoors [38]. The authors showed it was possible to track signals below 20 dB-Hz C/N0.

Peng et al. also implemented a VDLL/VFLL [56]. The authors showed the performance benefits under ionospheric scintillation or long periods of signal outage.

Summary. The VDLL/VFLL encompasses the bulk of vector tracking research performed to date. The research shows that vector tracking provides improvements over scalar tracking, especially in conditions where a subset of SVs is lower in power than the others, there is blockage of a subset of SVs, the C/N0 values are low, and/or the receiver is under high dynamics. The research proposed in this dissertation endeavors to bring these benefits and others to the VPLL.

2.3.2.4 Vector Phase Locked Loops.

Ignoring receiver clock bias, the errors that affect a GPS receiver are typically on the order of a few meters or so. This error is much less than the code length (300m for CA-code) and is slowly changing, making vector tracking a realistic solution for the code loop and for the carrier loop if tracking frequency. However, a few meters is much larger than the wavelength (19 cm for L_1), making vector phase tracking a challenging proposition. A few adventurous souls have attempted it and this section details their efforts. Since there are so few examples in the literature, each will be explained in some detail.

Low Bandwidth Local/High Bandwidth Vector Phase Locked Loop. The first example of a VPLL is Co-op tracking as proposed by Zhodzishsky et al. [76]. Figure 2.11 shows a flowchart of this method. The left side is the same as with other methods of tracking. The code loop is handled locally with a very low bandwidth loop filter and is aided by the carrier loop. The replica code is used in a navigation processor to obtain a position solution. The position solution and the satellite positions are used to create the projection matrix, \mathbf{H} . This is the matrix consisting of normalized pointing vectors to each satellite in the first 3 columns and ones in the last column. Least squares is used to estimate position and receiver clock offset errors, in cycles, from the phase error estimates out of the channel discriminators. A high bandwidth loop filter creates velocity and clock drift replicas. These replicas are projected to line of sight for each channel. Each channel also feeds the error estimates out of the phase discriminators through a low bandwidth loop filter to create a slowly changing estimate of the frequency offset. This estimate tracks the slowly changing biases due to atmospheric errors, and other effects. The slowly changing channel estimate and the fast changing vector estimate of frequency are then combined and integrated to close the phase loop. To summarize, the carrier loop is broken into two parts: (1) a local low bandwidth loop to track slowly changing per channel errors, and (2) a vector high bandwidth loop to track navigation and clock dynamics. This is not a pure vector method due to the need to follow the slowly changing phase errors.

Atmospheric Error Estimation. Henkel et al. introduce a form of tracking similar to Co-op tracking [32]. Figure 2.12 shows a flowchart of this method. As seen in the figure, instead of using local low bandwidth feedback to follow the slowly changing phase errors, the authors project the phase errors, using weighted least squares, into position, receiver clock, ionospheric and zenith tropospheric errors. These errors are then run through loop filters and transformed back to the individual channel phase domain, and the loop is closed. An initial estimate of the phase errors is made prior to vector lock and

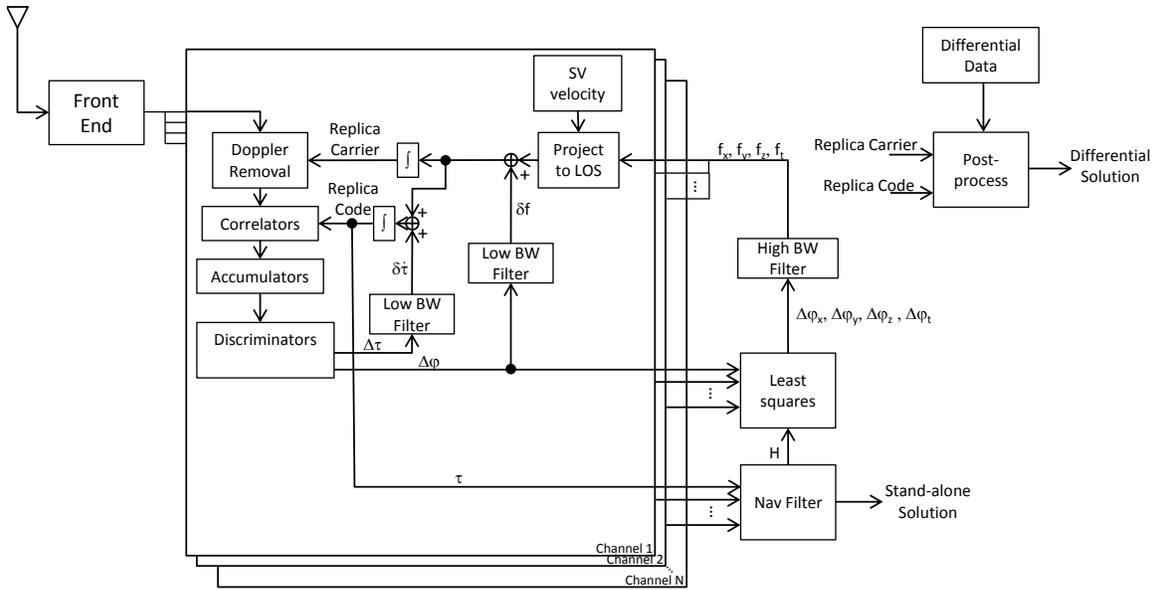


Figure 2.11: Vector Phase Locked Loop with Co-op Tracking

periodic reinitialization is required to maintain vector lock, keeping this from being a pure vector method. The authors show how this method reduces tracking errors on channels with deep amplitude fades due to ionospheric activity. Note how the position is maintained in a separate navigation filter.

Henkel et al. add receiver autonomous integrity monitoring (RAIM) to this method and show how to correct for the PLL bias due to ionospheric dispersion across the very wideband (51MHz) Galileo E5 signal [31]. Giger et al. replace the weighted least squares/loop filter mechanization with a Kalman filter as shown in Figure 2.13 to jointly estimate optimal replica phase estimates [23].

Giger et al. then include the code delay tracking error estimates in the Kalman filter, creating a vector code loop as shown in Figure 2.14. They refer to this as joint carrier and code tracking [22]. Giger et al. go on to estimate the position in the tracking loop Kalman filter vice in a separate navigation filter as shown in Figure 2.15. They refer to this

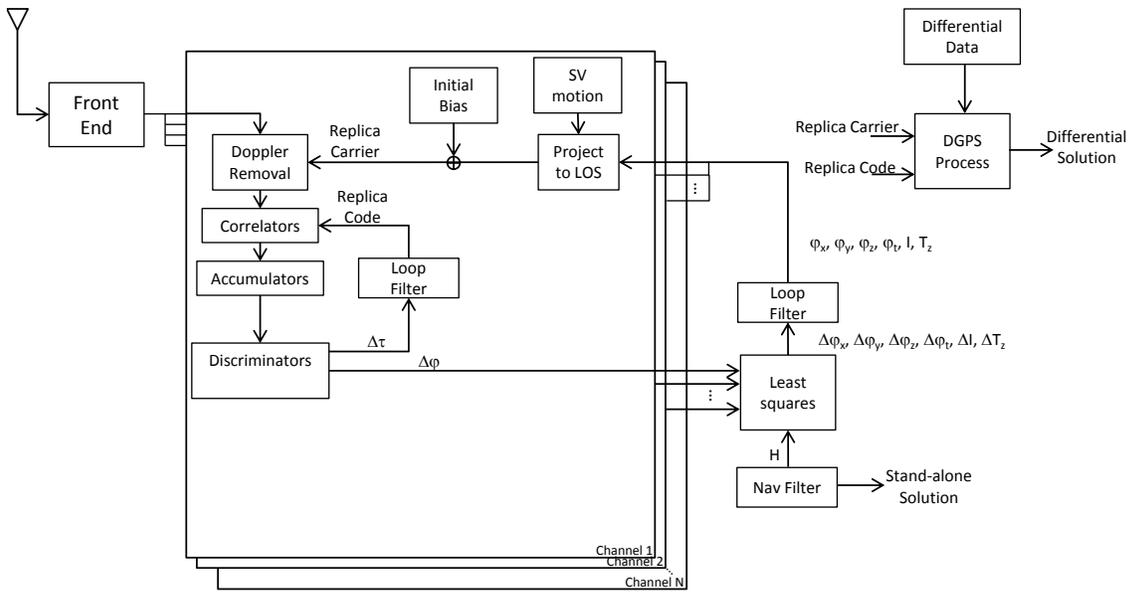


Figure 2.12: Vector Phase Locked Loop with Estimation of Atmospheric Errors Using Loop Filters

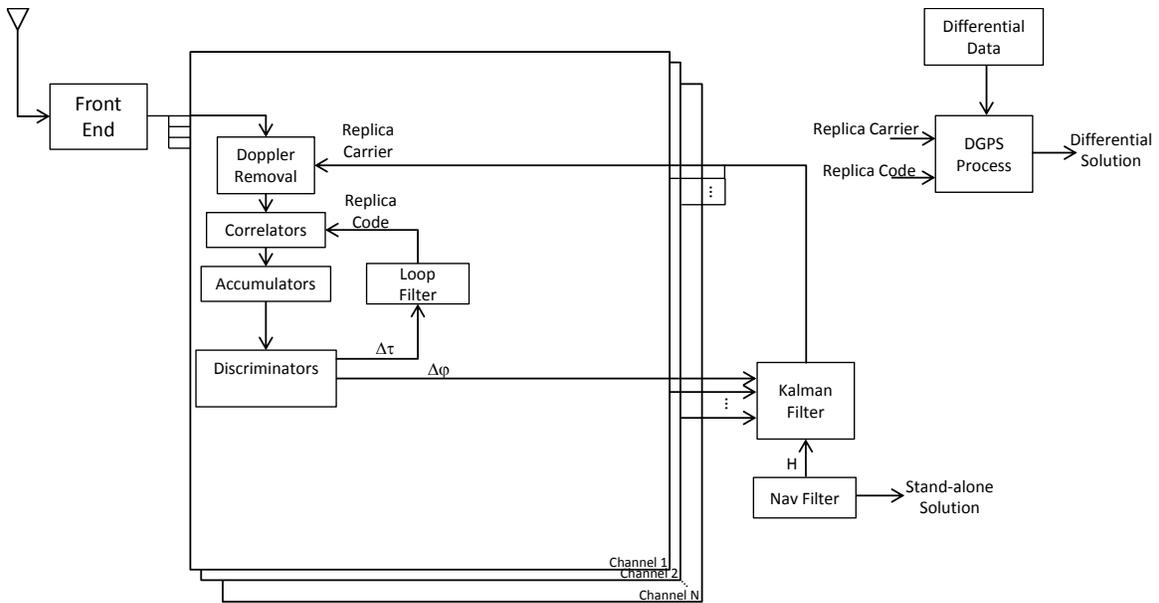


Figure 2.13: Vector Phase Locked Loop with Estimation of Atmospheric Errors Using a Kalman Filter

as position domain joint tracking [19]. Giger and C. Günther then developed a multiple antenna version incorporating an estimate of the platform attitude [20]. The authors show the benefits and drawbacks of this scheme compared to a digital beamformer. Giger and C. Günther also showed the robustness of their method in a multipath environment [21]. The Kalman filter methods have several states to estimate each SV's phase keeping these from being pure vector methods. The Kalman filter estimates at least 2 states per SV for one frequency and then estimates an ionospheric correction to account for all the other frequencies of the same SV.

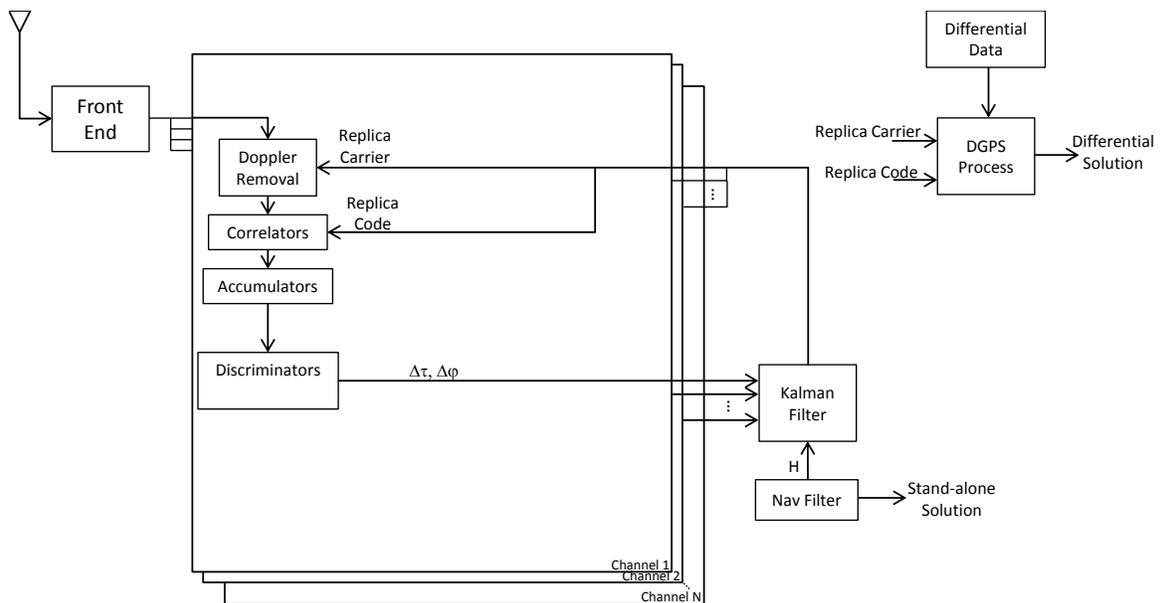


Figure 2.14: Vector Phase Locked Loop with Joint Code and Carrier Tracking

Soloviev et al. have a mechanization where the phase discriminators are used to update the accumulated Doppler and estimate the beginning phase for the next interval [65]. The difference between the change in accumulated Doppler over the accumulation interval for each satellite and those predicted by the navigation filter are used as measurements to

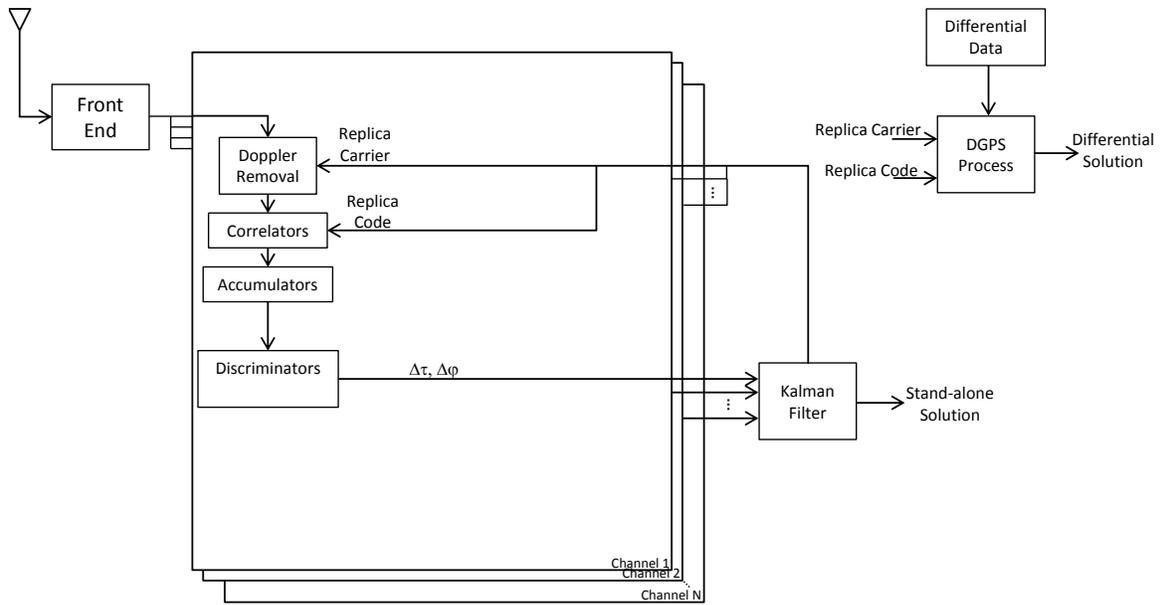


Figure 2.15: Vector Phase Locked Loop with Position Domain Joint Tracking

update the navigation states. The integration period is extended in this scheme using an efficient method of determining the navigation bits across bit transitions. The same authors then used this mechanization to perform flight tests [66]. The results showed very good carrier phase track and relative positioning accuracy at 15 dB-Hz C/N0. Soloviev and Dickman added a multipath monitoring system into the architecture and were able to track carrier phase to the same 15 dB-Hz level indoors [63, 64]. This method uses two states per channel to follow slowly changing errors in each channel [65].

Summary. Current VPLL methods are not purely vector and do not incorporate differential corrections directly in their tracking loops. The state-of-the-art in this line of research requires many states (at least 2 per SV tracked) in the Kalman filter to estimate each SV's phase. The DVPLL, derived in Chapter 3, requires only navigation and clock offset states for short baselines.

III. Differential Vector Phase Locked Loop Tracking

This chapter details the DVPLL that is being proposed as the foundation for the research described in this dissertation. The DVPLL process, depicted in Figure 3.1, brings code and carrier measurements from a stationary receiver at a surveyed location directly into the tracking loops of a rover receiver. The stationary receiver will be referred to as the base station for the rest of this dissertation. The base station code phase and carrier phase measurements are used to create replica code and carrier at the rover receiver, given a navigation state.

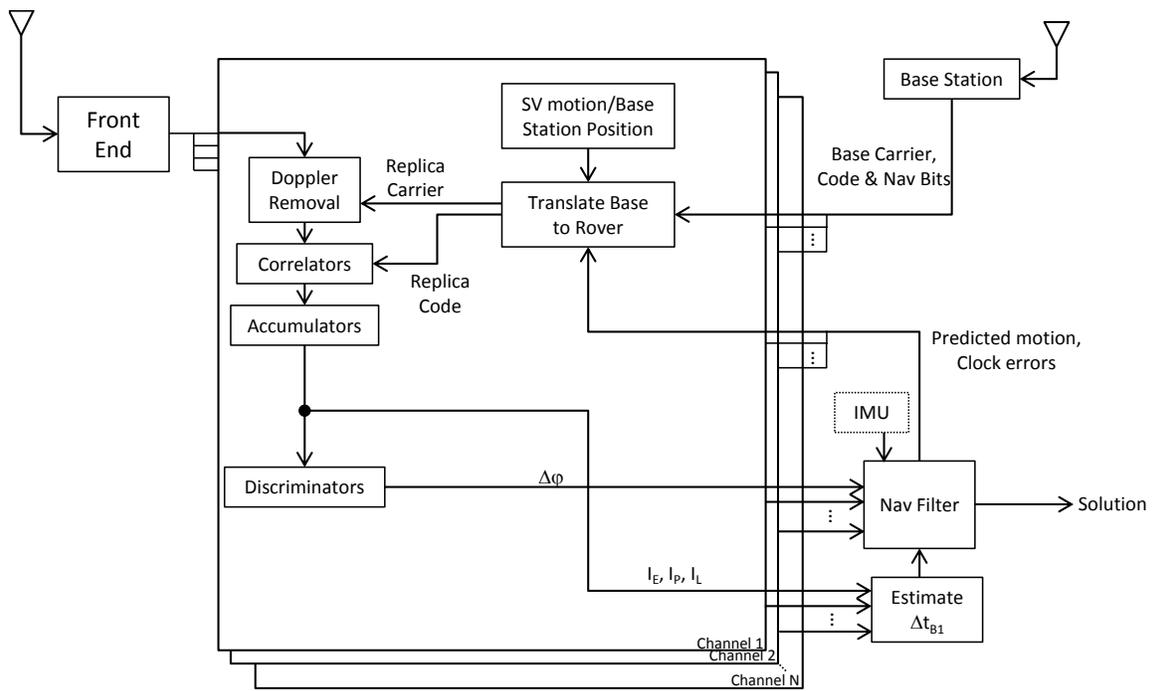


Figure 3.1: Differential Vector Phase Locked Loop Flow Chart

This chapter is organized in the following manner. First the base station measurement requirements are detailed. This is followed by a section describing the rover pre-processing

to get the tracking loop started. The method of translating the base station measurements to create the replica signals is then detailed, followed by a derivation of the Kalman filter used to estimate the navigation state of the rover. Finally, results from static data taken at the Air Force Institute of Technology (AFIT) and the Air Force Research Laboratory (AFRL) Sensors Directorate are provided.

The processing introduced in this chapter assumes a base station receiver connected to a surveyed antenna and a rover receiver front end connected to a receiver aboard a mobile platform. The data from each system are recorded in such a way that there is continuous recording from the base station any time the rover is recording.

3.1 Base Station Measurements

The base station measurements required by the DVPLL are simply code and carrier measurements (tagged with receiver time) and the raw navigation data bits (tagged with the satellite clock time). For this research two types of data were used as input. The first type was TRIGR data processed using a software GPS receiver developed by Dr. John Raquet, Dr. Marshall Haker, and Mr. Ben Downing at AFIT, following notes for Dr. Raquet's Advanced GPS Receiver Design class. The second type was RINEX data converted from NovaTel binary files. The carrier phase measurements in the RINEX files has an opposite sign to those in the SW receiver which had to be accounted for.

3.2 Rover pre-Processing

This chapter explains the method for maintaining differential vector phase lock in the current integration period assuming differential vector phase lock in the previous integration period. To that end, the rover receiver position is initialized with the surveyed coordinates and allowed to wander from there. To initialize the clock offset states the TRIGR rover data are pre-processed using the same GPS software receiver mentioned in the previous section in order to get the correct alignment between the two data sets. Only a

short time segment is needed from an SV common to the base station. Knowing the location of the rover antenna makes it possible to find the relative clock offset as the difference of the two receiver's indicated times for a common code phase. The relative clock drift can be found as the difference of the Doppler frequency estimates (corrected for range-rate differences) divided by the nominal satellite frequency. In general the rover receiver's antenna is not surveyed at the start of a test. An acquisition process for the general case is more fully explored in the next chapter.

3.3 Translating Base Station Replica Signals to Create Rover Replica Signals

In order to use the code and carrier measurements recorded at a base station in the tracking loops of a rover, the measurements must be corrected for the base station's clock offset, drift, and range difference from the rover. This section derives the method for accomplishing this task.

3.3.1 Receiver Indicated Time.

The base station and rover receiver indicated elapsed times are given by

$$t_B = \frac{k_B}{f_s} \quad (3.1)$$

$$t_R = \frac{k_R}{f_s} \quad (3.2)$$

where

t_B = base station indicated elapsed time since sample 0 (s)

t_R = rover indicated elapsed time since sample 0 (s)

k_B = base station sample number (samples)

k_R = rover sample number (samples)

f_s = nominal sampling frequency (Hz)

The relationship between the times indicated by the base station, rover, and system time is given by the equations

$$t_B = \int_{t_{0B}}^t 1 + \epsilon_B(\xi) d\xi \quad (3.3)$$

$$= \underbrace{\int_{t_{0B}}^{t_1} 1 + \epsilon_B(\xi) d\xi}_{t_{B1}} + \int_{t_1}^t 1 + \epsilon_B(\xi) d\xi \quad (3.4)$$

where

t_{0B} = system time of base station sample 0 (s)

t_1 = system time at beginning of period of interest (e.g. integration period) (s)

t = system time (s)

ϵ_B = time varying base station offset from nominal frequency (unitless)

t_{B1} = base station indicated time at beginning of integration period (s).

For a certain time interval ϵ_B is approximately constant, so

$$t_B \approx t_{B1} + \int_{t_1}^t 1 + \epsilon_B d\xi \quad (3.5)$$

$$\approx t_{B1} + (t - t_1)(1 + \epsilon_B) \quad (3.6)$$

Solving for t yields

$$t \approx \frac{t_B - t_{B1}}{1 + \epsilon_B} + t_1 \quad (3.7)$$

Similarly the rover indicated time is derived as

$$t_R \approx t_{R1} + (t - t_1)(1 + \epsilon_R) \quad (3.8)$$

and

$$t \approx \frac{t_R - t_{R1}}{1 + \epsilon_R} + t_1 \quad (3.9)$$

3.3.2 Creating the Replica Phase.

The measured phase for satellite i at the base station is given by

$$\phi_B^i(t_B) = \phi_T^i(t - \Delta t_B^i) - \phi_{mB}(t_B) - f_b t_B \quad (3.10)$$

where

ϕ_{mB} = phase of base station mixer (cycles)

f_b = nominal baseband frequency (Hz)

Since the mixer and sampler are in phase lock the mixer phase can be expanded as

$$\phi_{mB}(t) = \phi_{0mB} + \int_{t_{0B}}^t f_{mB}(\xi) d\xi \quad (3.11)$$

$$= \phi_{0mB} + \int_{t_{0B}}^t f_m(1 + \epsilon_B(\xi)) d\xi \quad (3.12)$$

$$= \phi_{0mB} + f_m \int_{t_{0B}}^t 1 + \epsilon_B(\xi) d\xi \quad (3.13)$$

$$= \phi_{0mb} + f_m t_B \quad (3.14)$$

where

ϕ_{0mB} = mixer phase at sample zero (cycles)

f_m = nominal mixing frequency (Hz)

Substituting (3.14) into (3.10) and using the fact that $f_m + f_b = f_{sat}$ where f_{sat} is the nominal satellite frequency, yields

$$\phi_B^i(t_B) = \phi_T^i(t - \Delta t_B^i) - f_{sat} t_B - \phi_{0mB} \quad (3.15)$$

Further substituting (3.7) into (3.15) gives

$$\phi_B^i(t_B) = \phi_T^i\left(\frac{t_B - t_{B1}}{1 + \epsilon_B}\right) + t_1 - \Delta t_B^i - f_{sat} t_B - \phi_{0mB} \quad (3.16)$$

Similarly, the phase of the rover, without subtracting the baseband phase, yields the rover replica phase given by

$$\phi_R^i(t_R) = \phi_T^i\left(\frac{t_R - t_{R1}}{1 + \epsilon_R} + t_1 - \Delta t_R^i\right) - f_m t_R - \phi_{0mR} \quad (3.17)$$

The t_B , denoted t'_B , is found such that the operands of ϕ_T^i in (3.16) and (3.17) are equal or

$$\frac{t'_B - t_{B1}}{1 + \epsilon_B} + t_1 - \Delta t_B^i = \frac{t_R - t_{R1}}{1 + \epsilon_R} t_1 + t_1 - \Delta t_R^i \quad (3.18)$$

Solving for t'_B yields

$$t'_B = t_{B1} + \frac{1 + \epsilon_B}{1 + \epsilon_R}(t_R - t_{R1}) + (\Delta t_B^i - \Delta t_R^i)(1 + \epsilon_B) \quad (3.19)$$

Using the definition of Δt^i in (2.2) gives

$$\Delta t_B^i - \Delta t_R^i = \frac{r_B^i - r_R^i}{c} + \tau_{propB}^i - \tau_{propR}^i \quad (3.20)$$

and

$$t'_B = t_{B1} + \frac{1 + \epsilon_B}{1 + \epsilon_R}(t_R - t_{R1}) + \left(\frac{r_B^i - r_R^i}{c} + \tau_{propB}^i - \tau_{propR}^i\right)(1 + \epsilon_B) \quad (3.21)$$

$$= t_{B1} + (1 + \epsilon_2)(t_R - t_{R1}) + \left(\frac{r_B^i - r_R^i}{c} + \tau_{propB}^i - \tau_{propR}^i\right)(1 + \epsilon_B) \quad (3.22)$$

Substituting t'_B from (3.19) for t_B in (3.16) yields

$$\phi_B^i(t'_B) = \phi_T^i\left(\frac{t_R - t_{R1}}{1 + \epsilon_R} + t_1 - \Delta t_R^i\right) - f_{sat} t'_B - \phi_{0mB} \quad (3.23)$$

or

$$\phi_T^i\left(\frac{t_R - t_{R1}}{1 + \epsilon_R} + t_1 - \Delta t_R^i\right) = \phi_B^i(t'_B) + f_{sat} t'_B + \phi_{0mB} \quad (3.24)$$

and using (3.17) yields

$$\phi_R^i(t_R) = \phi_B^i(t'_B) + f_{sat} t'_B - f_m t_R - \phi_{0mR} + \phi_{0mB} \quad (3.25)$$

The phase measurements estimated from the base station data can be translated to the rover using (3.22) and (3.25). ϵ_B can be ignored in (3.22) for small baselines or accurate

receiver clocks. If not, it can be estimated as well. The difference of the atmospheric errors in (3.22) are approximately zero for small baselines and similar altitudes. However, the differential tropospheric errors need to be compensated if the altitudes are different. Also, keep in mind that $\tau_{prop} = \tau_{carr}$ in this case.

Equations (3.22) and (3.25) capture the essence of this approach by enabling calculation of the phase of the rover using only the base station receiver measurements combined with knowledge of the relative position, time and frequency offsets. Equation (3.22) finds the base station time when the corresponding i th satellite signal is sampled by the base station. Equation (3.25) takes the base station's phase measurement at this time, mixes it back up to estimate the signal at the antenna, and then mixes it back down using the rover's mixer. The first term on the right side of (3.25) contains all the errors common to both receivers while the final two terms are the difference in phases between the two mixers at the start of sampling. This difference is the same constant for all satellites and becomes an error in the time offset estimate.

3.3.3 Code and Data Bits.

The replica code of the rover is generated from the translated replica code of the base station. The translation is performed similarly to the translation of the phase. The code time of the base station is given by

$$tc_B^i(t_B) = tc^i(t - \Delta t_B^i) \quad (3.26)$$

$$= tc^i\left(\frac{t_B - t_{B1}}{1 + \epsilon_B} + t_1 - \Delta t_B^i\right) \quad (3.27)$$

where

tc_B^i = base station code time (s) at time t

tc^i = code time of satellite i at time of transmission (s)

Similarly,

$$tc_R^i(t_R) = tc^i\left(\frac{t_R - t_{R1}}{1 + \epsilon_R} + t_1 - \Delta t_R^i\right) \quad (3.28)$$

Using the t'_B in (3.19) gives

$$tc_R^i(t_R) = tc_B^i(t'_B). \quad (3.29)$$

Keep in mind that $\tau_{prop} = \tau_{code}$ in this case. The data bits from the base station corresponding to the same code time are used to create the rover's data bits.

3.4 Kalman Filter to Update Rover Position, Velocity, and Time Offset Estimates

The data collected are from two static receivers so a simplified Kalman filter is developed using a stationary model to prove out the concept. For an excellent text on the derivation of the Kalman filter, consult Maybeck's definitive work [48]. The following derivation uses an extended Kalman filter going through the normal propagate and update cycles as shown in Figure 3.2.

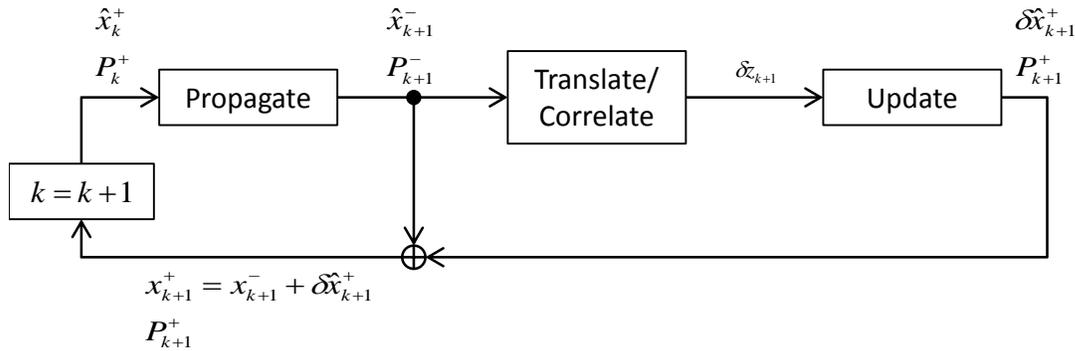


Figure 3.2: Extended Kalman Filter

The discretized propagate and update equations are given by

Propagate [48]:

$$\mathbf{x}_{k+1}^- = \Phi_k \mathbf{x}_k^+ + \mathbf{B} \mathbf{u}_k \quad (3.30)$$

$$\mathbf{P}_{k+1}^- = \Phi_k \mathbf{P}_k^+ \Phi_k^T + \mathbf{Q}_k \quad (3.31)$$

Update [48]:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1} \quad (3.32)$$

$$\delta \mathbf{x}_{k+1}^+ = \mathbf{K}_{k+1} \delta \mathbf{z}_{k+1} \quad (3.33)$$

$$\mathbf{x}_{k+1}^+ = \mathbf{x}_{k+1}^- + \delta \mathbf{x}_{k+1}^+ \quad (3.34)$$

$$\mathbf{P}_{k+1}^+ = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^- \quad (3.35)$$

where

\mathbf{x} = state vector

Φ = state transition matrix

$\mathbf{B} \mathbf{u}$ = input vector

\mathbf{P} = state covariance matrix

\mathbf{Q} = system noise matrix

\mathbf{K} = Kalman gain matrix

\mathbf{H} = measurement matrix

\mathbf{R} = measurement noise covariance matrix

\mathbf{z} = measurement vector

Each of these matrices will be developed in the paragraphs below.

Propagate. Derivation of the propagation equations starts with the continuous time state model given by [48]

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{w} \quad (3.36)$$

where

\mathbf{F} = state transition matrix

\mathbf{B} = input matrix

\mathbf{u} = input vector

\mathbf{G} = noise transformation matrix

\mathbf{w} = system noise vector

The state vector consists of the parameters to be estimated, namely

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ t_{B1} \\ \epsilon_2 \end{bmatrix} \quad (3.37)$$

where

x = rover ECEF x position (m)

y = rover ECEF y position (m)

z = rover ECEF z position (m)

t_{B1} = as defined in (3.4) (s)

ϵ_2 = as defined in (3.22) (unitless)

Since the rover is modeled as a stationary receiver, the first derivative of position is zero, so

$$\dot{x} = w_x \quad (3.38)$$

$$\dot{y} = w_y \quad (3.39)$$

$$\dot{z} = w_z \quad (3.40)$$

where

w_x = Gaussian white noise with strength q_x

w_y = Gaussian white noise with strength q_y

w_z = Gaussian white noise with strength q_z

The noise is added to the model to mimic Brownian motion or a slowly wandering position. In the stationary case, it would normally be a very small value.

The Kalman filter models the rover receiver so all derivatives are taken with respect to t_{R1} . The derivative of t_{B1} with respect to t_{R1} is found by starting with (3.4)

$$\frac{dt_{B1}}{dt_1} = \frac{d}{dt_1} \left(\int_{t_{0B}}^{t_1} 1 + \epsilon_B(\xi) d\xi \right) \quad (3.41)$$

$$= 1 + \epsilon_B(\xi)|_{\xi=t_1} \quad (3.42)$$

$$= 1 + \epsilon_B(t_1) \quad (3.43)$$

Similarly

$$\frac{dt_{R1}}{dt_1} = 1 + \epsilon_R(t_1) \quad (3.44)$$

and

$$\frac{dt_{B1}}{dt_{R1}} = \frac{1 + \epsilon_B(t_1)}{1 + \epsilon_R(t_1)} = 1 + \epsilon_2(t_1) \quad (3.45)$$

finally

$$\dot{t}_{B1} = 1 + \epsilon_2 + w_t \quad (3.46)$$

where

$$w_t = \text{Gaussian white noise with strength } q_t$$

The last variable in the state matrix is modeled as Brownian motion

$$\dot{\epsilon}_2 = w_\epsilon \quad (3.47)$$

where

$$w_\epsilon = \text{Gaussian white noise with strength } q_\epsilon$$

Pulling all the equations together yields the continuous time state space model matrices

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.48)$$

$$\mathbf{B} = \mathbf{G} = \mathbf{I} \quad (3.49)$$

$$\mathbf{u} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \quad (3.50)$$

$$\mathbf{w} = \begin{bmatrix} w_x & w_y & w_z & w_t & w_\epsilon \end{bmatrix}^T \quad (3.51)$$

The state transition matrix, Φ_k , is defined as [48]

$$\Phi_k = \Phi(t_{k+1}, t_k) \quad (3.52)$$

For a linear time-invariant system this is simply [48]

$$\mathbf{\Phi}_k = e^{\mathbf{F}\Delta T} = \mathbf{I} + \mathbf{F}\Delta T + \frac{\mathbf{F}^2\Delta T^2}{2} + \dots \quad (3.53)$$

where

$$\Delta T = t_{k+1} - t_k$$

In this case $\mathbf{F}^2 = \mathbf{0}$ so

$$\mathbf{\Phi}_k = \mathbf{I} + \mathbf{F}\Delta T \quad (3.54)$$

The matrix $\mathbf{B}\mathbf{u}_k$ is found by [48]

$$\mathbf{B}\mathbf{u}_k = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1} - \tau)\mathbf{B}\mathbf{u}(\tau)d\tau \quad (3.55)$$

and since \mathbf{B} is the identity matrix, and \mathbf{u} is a constant input vector then

$$\mathbf{B}\mathbf{u}_k = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1} - \tau)d\tau\mathbf{u} \quad (3.56)$$

$$= \int_{t_k}^{t_{k+1}} \mathbf{I} + \mathbf{F}(t_{k+1} - \tau)d\tau\mathbf{u} \quad (3.57)$$

$$= \left(\mathbf{I}\Delta T + \mathbf{F} \int_{t_k}^{t_{k+1}} (t_{k+1} - \tau)d\tau \right) \mathbf{u} \quad (3.58)$$

$$= \left(\mathbf{I}\Delta T + \mathbf{F} \left(t_{k+1}\tau - \frac{\tau^2}{2} \right) \Big|_{t_k}^{t_{k+1}} \right) \mathbf{u} \quad (3.59)$$

$$= \left(\mathbf{I}\Delta T + \mathbf{F} \frac{\Delta T^2}{2} \right) \mathbf{u} \quad (3.60)$$

Substituting in the definitions of each matrix yields

$$\mathbf{B}\mathbf{u}_k = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta T \\ 0 \end{bmatrix} \quad (3.61)$$

To find \mathbf{Q}_k in (3.31) we follow Van Loan [47]

$$\mathbf{A} = \begin{bmatrix} -\mathbf{F} & \mathbf{G}\mathbf{Q}\mathbf{G}^T \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \Delta T \quad (3.62)$$

$$\mathbf{B} = e^{\mathbf{A}} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} \quad (3.63)$$

$$\mathbf{Q}_k = \mathbf{B}_{22}^T \mathbf{B}_{12} \quad (3.64)$$

Substituting the known values of the matrices, using the fact that

$$\mathbf{Q} = \begin{bmatrix} q_x & 0 & 0 & 0 & 0 \\ 0 & q_y & 0 & 0 & 0 \\ 0 & 0 & q_z & 0 & 0 \\ 0 & 0 & 0 & q_t & 0 \\ 0 & 0 & 0 & 0 & q_\epsilon \end{bmatrix} \quad (3.65)$$

and reducing yields

$$\mathbf{A}^2 = \begin{bmatrix} \mathbf{0} & -\mathbf{F}\mathbf{Q} + \mathbf{Q}\mathbf{F}^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \Delta T^2 \quad (3.66)$$

$$\mathbf{A}^3 = \begin{bmatrix} \mathbf{0} & \mathbf{F}\mathbf{Q}\mathbf{F}^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \Delta T^3 \quad (3.67)$$

$$\mathbf{A}^n = \mathbf{0}, \text{ for } n \geq 4 \quad (3.68)$$

So

$$\mathbf{B} = \mathbf{I} + \mathbf{A} + \frac{\mathbf{A}^2}{2} + \frac{\mathbf{A}^3}{6} \quad (3.69)$$

$$\begin{aligned} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} + \begin{bmatrix} -\mathbf{F} & \mathbf{Q} \\ \mathbf{0} & \mathbf{F}^T \end{bmatrix} \Delta T + \begin{bmatrix} \mathbf{0} & -\mathbf{F}\mathbf{Q} + \mathbf{Q}\mathbf{F}^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \frac{\Delta T^2}{2} \\ &+ \begin{bmatrix} \mathbf{0} & \mathbf{F}\mathbf{Q}\mathbf{F}^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \frac{\Delta T^3}{6} \end{aligned} \quad (3.70)$$

and

$$\mathbf{Q}_k = (\mathbf{I} + \mathbf{F}\Delta T)(\mathbf{Q}\Delta T - \mathbf{FQ}\frac{\Delta T^2}{2} + \mathbf{QF}^T\frac{\Delta T^2}{2} + \mathbf{FQF}^T\frac{\Delta T^3}{6}) \quad (3.71)$$

$$= \mathbf{Q}\Delta T + \mathbf{FQ}\frac{\Delta T^2}{2} + \mathbf{QF}^T\frac{\Delta T^2}{2} + \mathbf{FQF}^T\frac{2\Delta T^3}{3} \quad (3.72)$$

hence,

$$\mathbf{Q}_k = \begin{bmatrix} q_x\Delta T & 0 & 0 & 0 & 0 \\ 0 & q_y\Delta T & 0 & 0 & 0 \\ 0 & 0 & q_z\Delta T & 0 & 0 \\ 0 & 0 & 0 & q_t\Delta T + q_\epsilon\frac{2\Delta T^3}{3} & q_\epsilon\frac{\Delta T^2}{2} \\ 0 & 0 & 0 & q_\epsilon\frac{\Delta T^2}{2} & q_\epsilon\Delta T \end{bmatrix} \quad (3.73)$$

Update. The measurement equation is given by [48]

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \quad (3.74)$$

where

$$\mathbf{v} = \text{measurement noise } E[\mathbf{v}^T \mathbf{v}] = \mathbf{R}$$

However, the measurements going into the Kalman filter are the differences between the replica signal and the incoming signal. Therefore the measurement equation can be linearized around the propagated state vector as [48]

$$\delta \mathbf{z} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^-} \delta \mathbf{x} + \mathbf{v} \quad (3.75)$$

The derivation of the Kalman filter assumes the measurement noise is zero-mean white Gaussian. However, the nonlinear discriminator maps the zero-mean white Gaussian I and Q measurements into phase measurements where these assumptions are not necessarily met. This, along with the nonlinear nature of the measurement equations, makes the EKF suboptimal.

The matrix $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ consists of N rows, where N is the number of satellites tracked, with each row as

$$\mathbf{H}(i, \dots) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(i, \dots) = \left[\frac{\partial \phi^i}{\partial x} \quad \frac{\partial \phi^i}{\partial y} \quad \frac{\partial \phi^i}{\partial z} \quad \frac{\partial \phi^i}{\partial t_{B1}} \quad \frac{\partial \phi^i}{\partial \epsilon_2} \right] \quad (3.76)$$

The partial derivatives are derived from (3.22) and (3.25) as follows. From (3.25)

$$\frac{\partial \phi_R^i(t_R)}{\partial x} = \frac{\partial \phi_B^i(t'_B)}{\partial x} + \frac{\partial f_{sat} t'_B}{\partial x} \quad (3.77)$$

$$= \frac{\partial \phi_B^i(t'_B)}{\partial t'_B} \frac{\partial t'_B}{\partial x} + f_{sat} \frac{\partial t'_B}{\partial x} \quad (3.78)$$

$$= \left(\frac{\partial \phi_B^i(t'_B)}{\partial t'_B} + f_{sat} \right) \frac{\partial t'_B}{\partial x} \quad (3.79)$$

$$= (\text{Doppler}_B^i + f_{sat}) \frac{\partial t'_B}{\partial x} \quad (3.80)$$

Using (3.22)

$$\frac{\partial t'_B}{\partial x} = -\frac{1}{c} \frac{\partial R_R^i}{\partial x} \quad (3.81)$$

and

$$R_R^i{}^2 = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 \quad (3.82)$$

so

$$2R_R^i \frac{\partial R_R^i}{\partial x} = 2(x - x_i) \quad (3.83)$$

$$\frac{\partial R_R^i}{\partial x} = \frac{(x - x_i)}{R_R^i} \quad (3.84)$$

$$= e_x \quad (3.85)$$

where

$$e_x = x \text{ component of pointing vector from SV to rover} \quad (3.86)$$

Substituting yields

$$\frac{\partial \phi_R^i(t_R)}{\partial x} = -\frac{\text{Doppler}_B^i + f_{sat}}{c} e_x \quad (3.87)$$

similarly

$$\frac{\partial \phi_R^i(t_R)}{\partial y} = -\frac{Doppler_B^i + f_{sat}}{c} e_y \quad (3.88)$$

$$\frac{\partial \phi_R^i(t_R)}{\partial z} = -\frac{Doppler_B^i + f_{sat}}{c} e_z \quad (3.89)$$

From (3.25) the next derivative is

$$\frac{\partial \phi_R^i(t_R)}{\partial t_{B1}} = (Doppler_B^i + f_{sat}) \frac{\partial t'_B}{\partial t_{B1}} \quad (3.90)$$

from (3.22)

$$\frac{\partial t'_B}{\partial t_{B1}} = 1 \quad (3.91)$$

so

$$\frac{\partial \phi_R^i(t_R)}{\partial t_{B1}} = (Doppler_B^i + f_{sat}) \quad (3.92)$$

From (3.25) the last derivative is

$$\frac{\partial \phi_R^i(t_R)}{\partial \epsilon_2} = (Doppler_B^i + f_{sat}) \frac{\partial t'_B}{\partial \epsilon_2} \quad (3.93)$$

from (3.22)

$$\frac{\partial t'_B}{\partial \epsilon_2} = (t_R - t_{R1}) \quad (3.94)$$

so

$$\frac{\partial \phi_R^i(t_R)}{\partial \epsilon_2} = (Doppler_B^i + f_{sat})(t_R - t_{R1}) \quad (3.95)$$

Since the measurement is taken at the end of the cycle $t_R = t_{R1} + \Delta T$ so

$$\frac{\partial \phi_R^i(t_R)}{\partial \epsilon_2} = (Doppler_B^i + f_{sat})\Delta T \quad (3.96)$$

The maximum Doppler frequency for a stationary receiver is 7000 Hz. If the $Doppler_B^i$ term is ignored in $Doppler_B^i + f_{sat}$ the error in estimating the state variable offsets would be less than

$$error < \frac{7000}{f_{sat}} = \frac{7000}{1575 \times 10^6} < 5 \text{ ppm} \quad (3.97)$$

Ignoring this term gives an i th row of \mathbf{H} as

$$\mathbf{H}(i, \dots) = f_{sat} \begin{bmatrix} -\frac{e_x}{c} & -\frac{e_y}{c} & -\frac{e_z}{c} & 1 & \Delta T \end{bmatrix} \quad (3.98)$$

The phase discriminator is a four-quadrant arctangent since the navigation data bits are used. This discriminator limits the measurements to a half cycle on either side of zero. As in the scalar case with navigation data bits known, the total phase error from all sources (dynamics, noise, atmosphere mismodeling, etc) must be kept less than this for all SVs in order to maintain vector phase lock [35]. This is easily done with proper design. Since the raw discriminator outputs are used in the navigation filter, there are no integers to solve for and monitor. A ‘cycle slip’ has no meaning in the algorithm. However, due to the interpolations in translating the base station measurements to the rover, the base station measurements must not have any cycle slips across the integration period. This is not a challenge for modern survey-grade receivers.

3.5 Results

3.5.1 Test 1.

For the data shown in this section, two transform-domain instrumentation GPS receiver (TRIGR) [25] front ends were used. One, considered the base station, was connected to a surveyed antenna at AFIT and the other, the rover, was connected to a surveyed antenna at AFRL. The data were recorded at 56.32 MHz using 8-bit analog-to-digital converters. The mixer was at 1505.42 MHz bringing the signal down to 70 MHz and subsampling brought the signal down a further 56.32 MHz for a final baseband frequency, f_b , of 13.68 MHz. The effective mixing frequency is then 1561.74 MHz and this is the value used for f_m .

The relative antenna locations are shown on the map in Figure 3.3. Figure 3.4 shows a close-up image of the AFIT antenna and Figure 3.5 shows a close-up image of the AFRL

Antenna. The two antennas are 570 m apart mostly in a North-South direction and the AFRL antenna is approximately 12 m higher than the AFIT antenna.

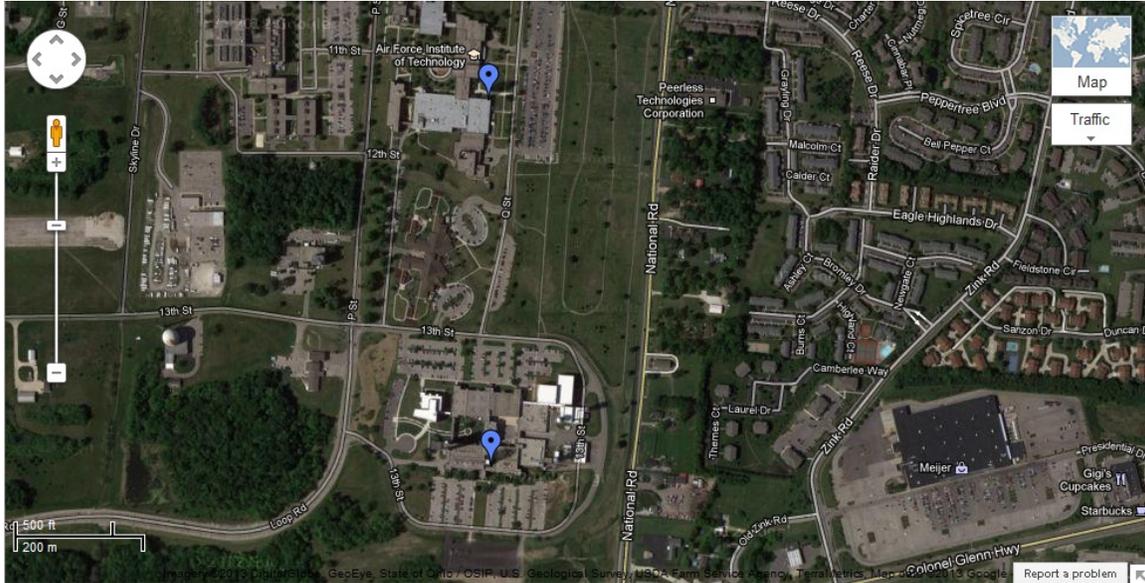


Figure 3.3: Antenna Locations, North Marker is AFIT, South is AFRL, Map Data: Google, DigitalGlobe, State of Ohio, U.S. Geological Survey, USDA Farm Agency

Approximately 3 minutes of data were simultaneously collected from both locations at 0830 EST 14 Nov 11. The GPS L1 data were stripped out of the files and processed using a scalar software receiver. Figure 3.6 shows the locations of the satellites at that time.

Figure 3.7 shows the satellite C/N0 values for each receiver. The satellites tracked by the base station were PRNs 1, 7, 8, 11, 17, 19, 26, and 28 and those tracked by the rover were 1, 7, 8, 11, 17, 19, and 28. The common set between the two was 1, 7, 8, 11, 17, 19, and 28, so these were chosen for further processing by the method outlined in the previous



Figure 3.4: AFIT Antenna Looking North-East



Figure 3.5: AFRL Antenna Looking East

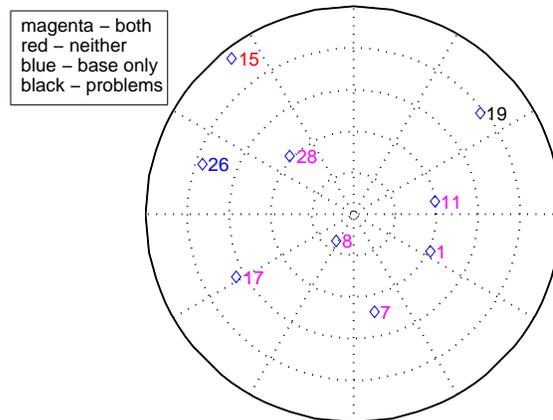


Figure 3.6: Sky Plot of Satellite Locations

sections. Note how the CN0 values for the rover in Figure 3.7 are almost all the same except for PRN 19. This behavior is unusual and the reason for it is currently unknown.

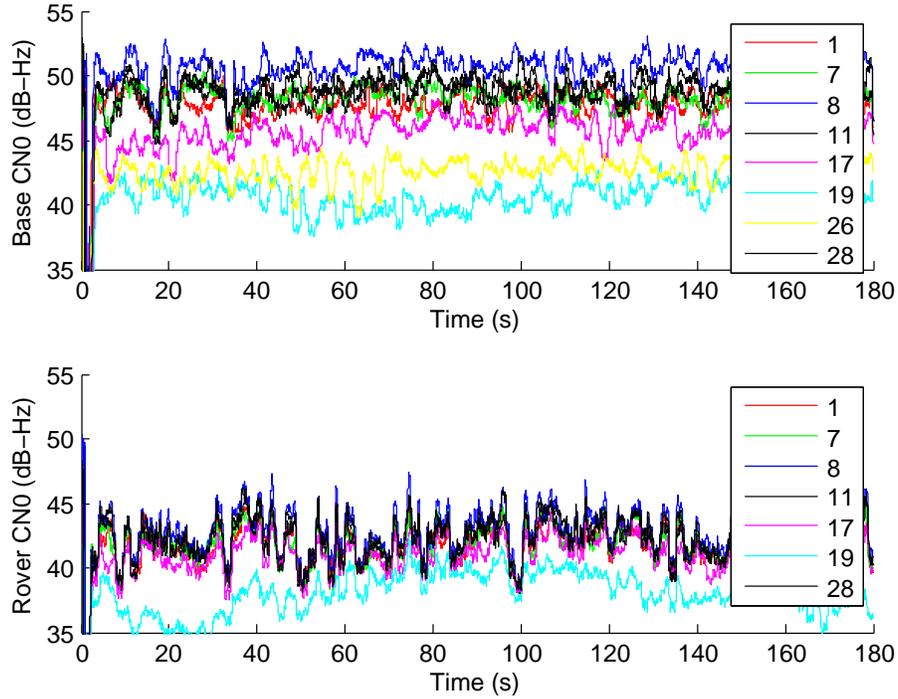


Figure 3.7: Plot of CN0 Values for Each Receiver

The values for the Q matrix are chosen such that

$$q_x = q_y = q_z = 0.05^2 \text{ m}^2/\text{s} \quad (3.99)$$

$$q_t = 8 \times 10^{-20} f_{sat}^2 \text{ cycles}^2/\text{s} \quad (3.100)$$

$$q_\epsilon = 2\pi^2(4 \times 10^{-23})f_{sat}^2 \text{ cycles/s}^2/\text{s} \quad (3.101)$$

The last two values are derived from Brown and Hwang [4] and are converted to cycles to avoid numerical problems during the update cycle. The \mathbf{R} matrix was set so $\mathbf{R}(i, i) = 0.01^2 \text{ cycles}^2$.

The filter would not converge and after further investigation the filter was found to converge if PRN 19 was removed. Figure 3.8 shows the phase discriminators when processing was completed without using PRN 19 in the solution. Notice how the PRN 19 phase discriminator behaves differently from the other satellites. The reason for this difference is unknown and the antennas had clear lines of sight to the satellite. PRN 19 was lower in the sky and its C/N_0 performance seemed different than the other satellites especially at the rover. It is suspected that multipath may be the culprit.

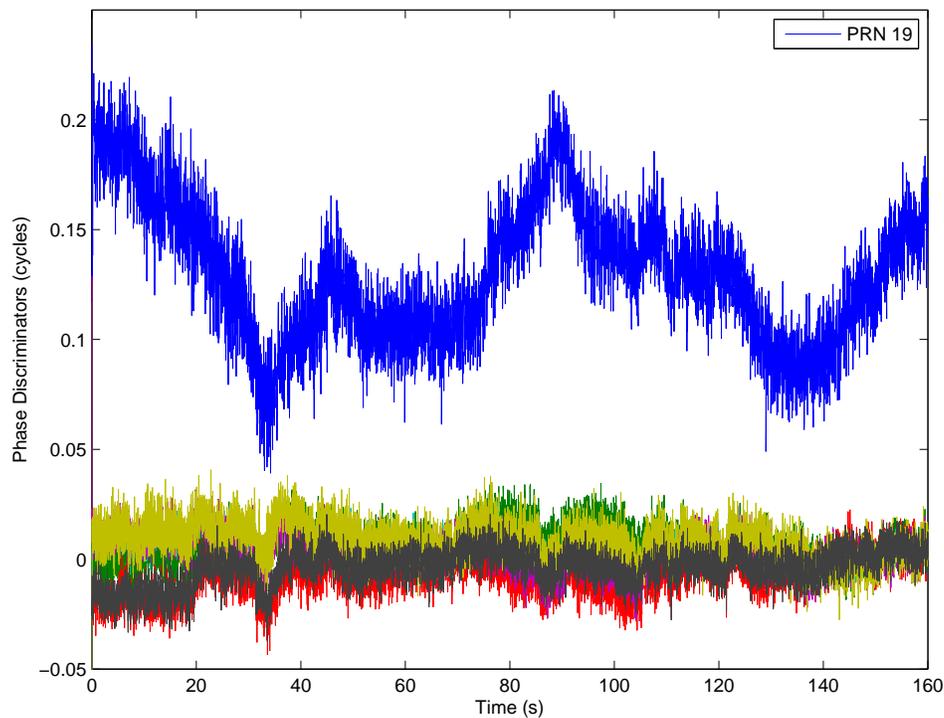


Figure 3.8: Phase Discriminators - PRN 19 not used in solution

Figure 3.9 shows the filter state estimates. The position states were differenced from the surveyed location to show the error. The filter converged to a position about 9 cm away

from the rover's surveyed location. The quality of the survey coordinates on each end of the baseline are suspect and may be the cause of the difference.

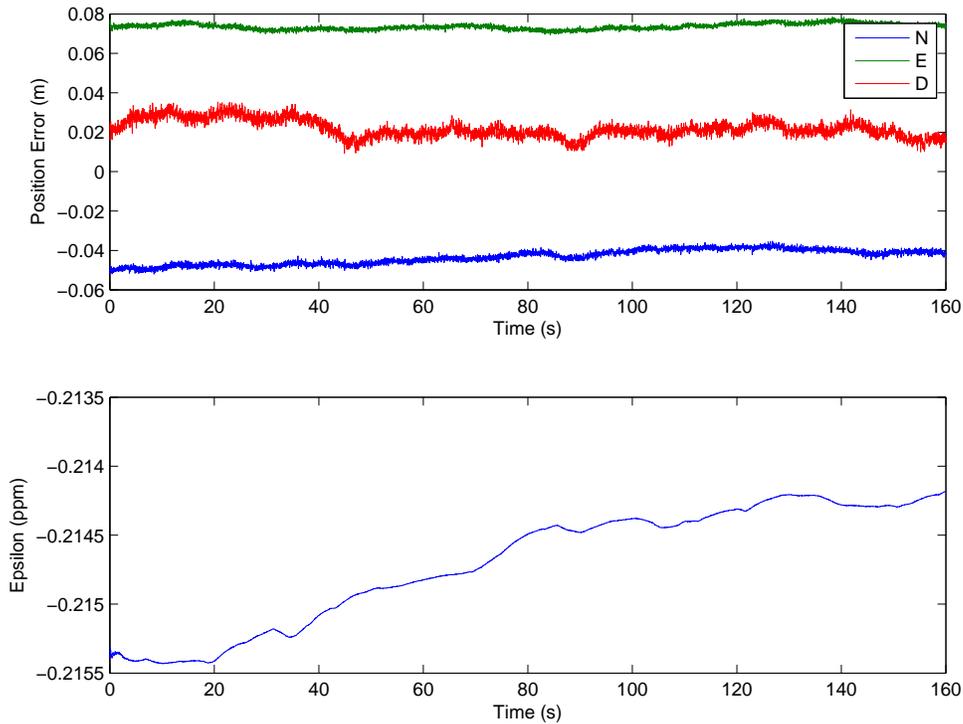


Figure 3.9: Position Errors and Epsilon

To summarize, the filter did converge to a position and the phase discriminators were very stable at this position. However, this position was offset slightly from the surveyed location of the antenna. The results demonstrate that this method is feasible, so further experiments were executed.

3.5.2 Test 2.

This test took advantage of a data collect for a different project. A specialized TRIGR front end was developed for controlled reception pattern antenna (CRPA) research at AFIT.

The front end has four separate L1 channels triggered off the same clock. Three of the channels were connected to three antennas on an Antcom 7-element antenna. The three antennas formed an equilateral triangle with the elements spaced 19.51 cm apart. The antenna was mounted to a ground plane placed on the roof of AFIT's ANT center and oriented such that it was level. One side of the triangle pointed North and one vertex pointed South. The fourth channel was connected to an Ashtech antenna located approximately 7 meters from the CRPA. The center element of the CRPA was connected to a Septentrio receiver for a simultaneous collection. Figure 3.10 shows a picture of the antennas on the roof of the ANT center.



Figure 3.10: Looking North Toward CRPA and AFIT Ashtech Antenna

Approximately 15 minutes of data were collected on 14 Jun 13 during two separate sessions. The first session began at approximately 1:25 PM EDT and the other at approximately 4:54 PM EDT. The data were processed using a scalar software receiver to obtain code and phase measurements. Using the Ashtech antenna as a base station, the phase measurements were processed using ambiguity-resolved single-difference techniques [49]. The bias between the channels due to different path lengths was estimated as well. The data were also processed using the DVPLL and the results were compared. Figures 3.11 and 3.12 show the horizontal and vertical position errors obtained from both methods of processing. The single-difference estimates demonstrate similarity with the DVPLL estimates as easily observed in Figures 3.11 and 3.12. The results are similar for the other antennas.

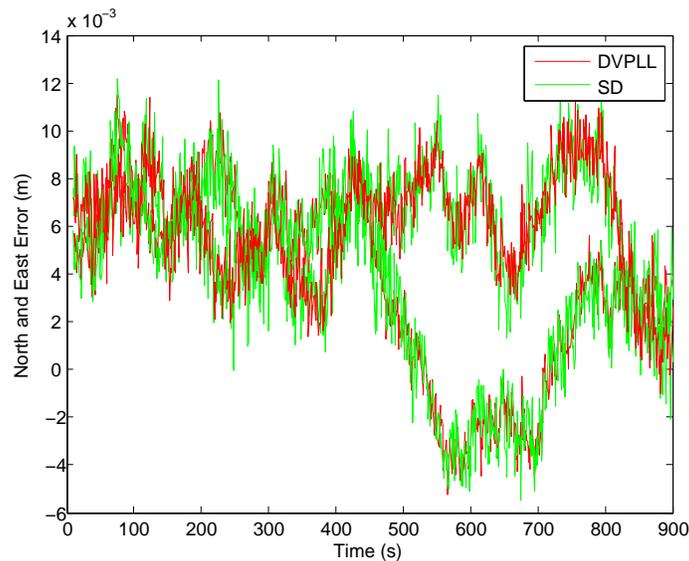


Figure 3.11: DVPLL and Single-Difference Horizontal Position Errors for Northeast CRPA Antenna from 4:54 PM Data Collect

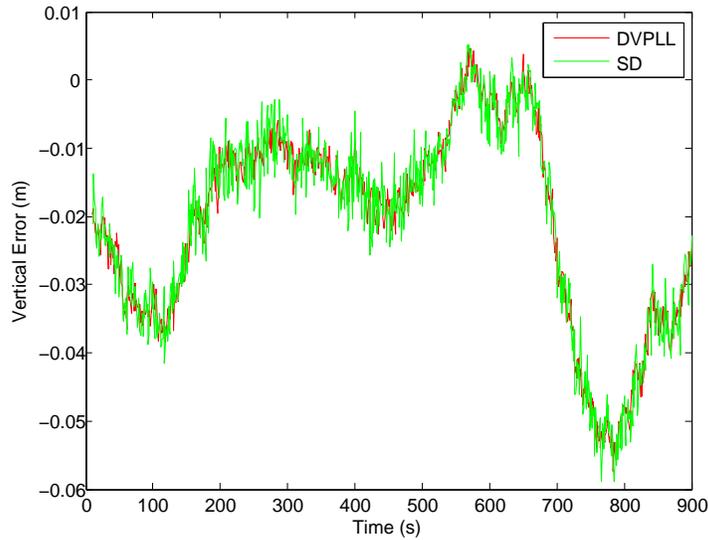


Figure 3.12: DVPLL and Single-Difference Vertical Position Errors for Northeast CRPA Antenna from 4:54 PM Data Collect

Figure 3.13 shows the ambiguity-resolved single-difference phase corrected for the surveyed baseline. The figure clearly demonstrates phase variations on the order of 0.15 cycles. As seen in Figure 3.14, even high satellites (PRNs 7 and 8) had large phase variations. Multipath is normally higher on the horizon than in the zenith, so these high values of phase variation may be due to either actual antenna phase offsets or mutual coupling [35, 37].

3.5.3 Test 3.

A final test was designed to demonstrate the accuracy of the DVPLL. This test was configured as shown in Figure 3.15. Two choking antennas were placed on the roof of the ANT center and the output of each was split and input into a Novatel survey-grade receiver and into one channel of the TRIGR front end used in Test 2. Figure 3.16 shows the antenna

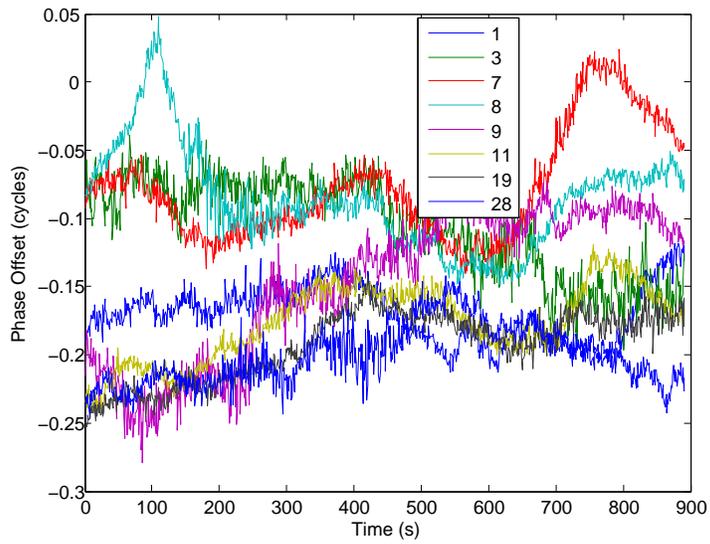


Figure 3.13: Ambiguity-Resolved Single-Difference Phase Corrected for Surveyed Baseline Offset for Northeast CRPA Antenna from 4:54 PM Data Collect

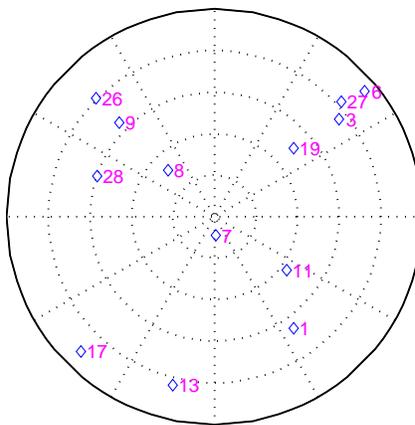


Figure 3.14: Sky Plot for 4:54 PM Data Collect

placement on the roof of the ANT center. Data were recorded from the NovaTel receivers for 24 hours, on 12-13 Dec 13, and sent to NGA's Online Positioning User Service (OPUS) to obtain survey coordinates. Table 3.1 shows the survey coordinates obtained from OPUS.

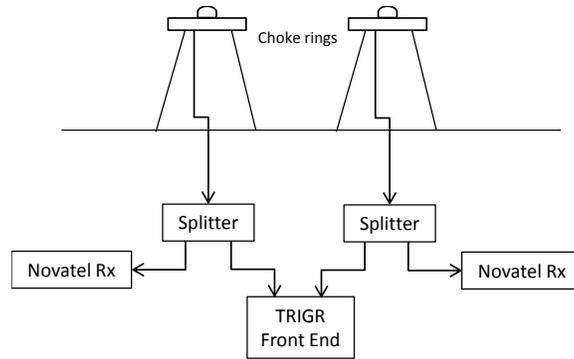


Figure 3.15: Configuration for Test 3



Figure 3.16: Choking Antennas on ANT Center Roof Looking North, *Blue* Antenna in Background, *Yellow* Antenna in Foreground

Table 3.1: Antenna Reference Point IGS-08 Survey Coordinates for ANT Center Rooftop Antennas from OPUS 12-13 Dec 13

Antenna	x (m)	y (m)	z (m)
Blue	506063.533	-4882262.798	4059609.662
Yellow	506056.683	-4882274.265	4059596.818

The 24-h NovaTel carrier-phase observations were reprocessed using the survey coordinates to obtain single-difference phase residuals. The rms of the phase residuals vs SV elevation angle is plotted in Figure 3.17 along with a parametrically derived function to approximate the L_1 curve. The parametrically derived function is used in subsequent processing to create the measurement covariance matrix, R .

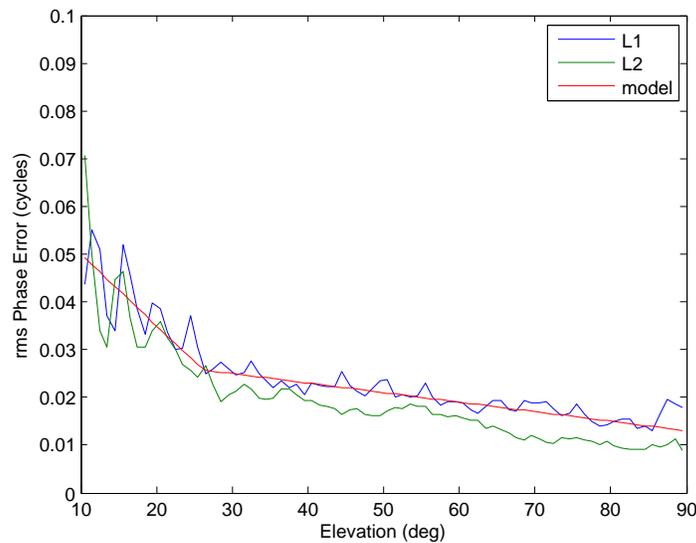


Figure 3.17: ANT Center Rooftop Antennas Single-Difference Phase Error rms vs Elevation, 24-h period, 12-13 Dec 13

Both the TRIGR front end and NovaTel receivers were used to collect 15 minutes of 4 Hz data on 13 Dec 13. The NovaTel receivers were not configured up to record the raw navigation data bits so the TRIGR data were processed using a conventional SW receiver to obtain the bits. The TRIGR data were then processed using the DVPLL.

Two methods were used to process the data in the DVPLL, the zero-baseline method and the short-baseline method. Zero-baseline results were obtained by using each Novatel as a base station and the TRIGR channel corresponding to the same antenna as the rover. In this mode, the signal is identical from SV to splitter, so all differential errors (except those due to receiver noise and receiver clocks) would be exactly zero. Short-baseline (18.5 m) results were obtained by using each Novatel as a base station and the TRIGR corresponding to the other antenna as the rover. Figure 3.18 shows a sky plot of the constellation at the time of the test. Eight to nine satellites were above the 10 degree mask angle set for the test.

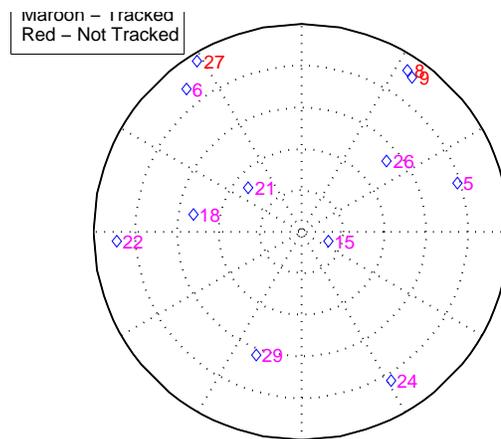


Figure 3.18: Sky Plot for 13 Dec 13, 15-min Data Collect

Figure 3.19 shows a sample plot of the DVPLL 3D error versus time for each mode. The overall 3D error is 1.3 mm for the zero-baseline test and 5.3 mm for the short-baseline

test, which are comparable to an ambiguity-resolved differential carrier-phase solution [35]. This shows that the DVPLL clearly maintained vector phase lock on the correct solution. Figure 3.20 plots a sample of the relative clock frequency offset terms, ϵ_2 , versus time for each mode. Figure 3.20 plots both processing modes using the TRIGR channel attached to the blue antenna as the rover. Thus the rover clock drift will be the same in each case and any difference in ϵ_2 , between processing modes, is solely due to the difference in clocks in the two NovAtel receivers used as base stations. Note that ϵ_2 is approximately the same between processing modes. This is due to the NovAtel receiver's correction of measurement data for clock variations. The -1.8×10^{-7} s/s drift is comparable to the value expected of the OCXO used in the TRIGR front end [25].

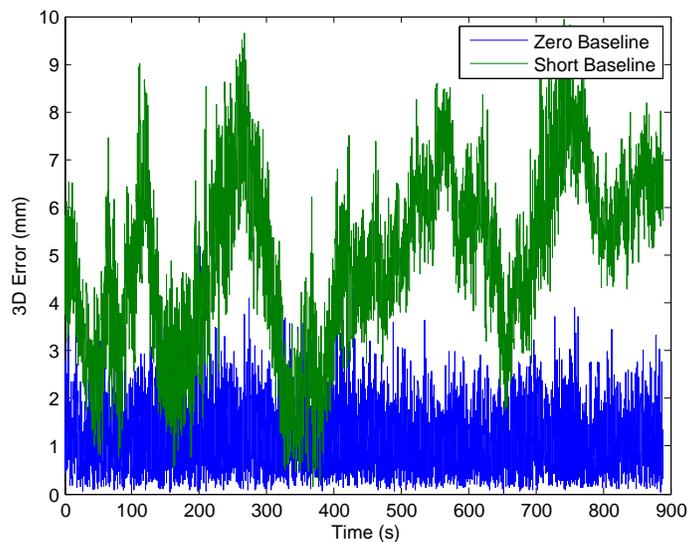


Figure 3.19: DVPLL 3D Position Errors, 13 Dec 13

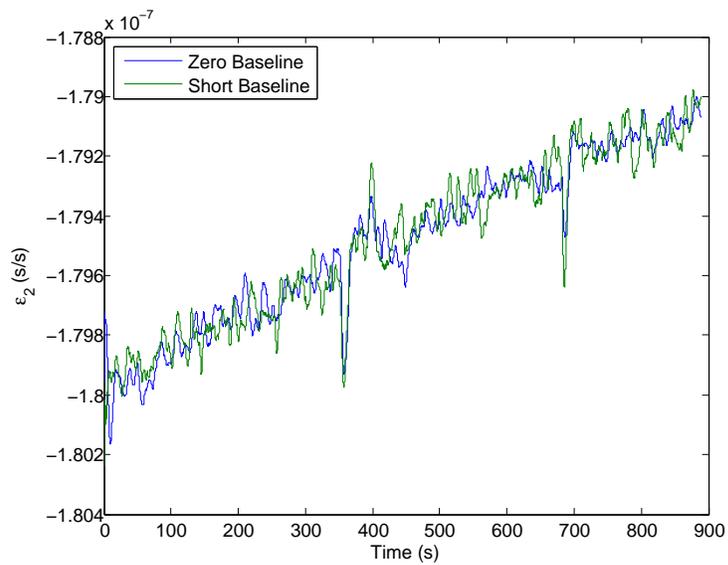


Figure 3.20: TRIGR-NovaTel Relative Clock Frequency Offsets, 13 Dec 13

3.6 Conclusion

In this chapter, the DVPLL tracking method is derived and test results are presented. The DVPLL's accuracy was around 5 mm for a short-baseline static test. These results show the DVPLL tracking algorithm is operating correctly and gives an ambiguity-resolved quality solution directly in the tracking loop.

IV. Differential Vector Phase Locked Loop Acquisition

The previous chapter details the DVPLL tracking algorithm for maintaining vector phase lock in the current integration period assuming vector phase lock in the previous integration period. This chapter deals with the acquisition algorithm used to obtain initial vector phase lock. In contrast to other methods, the DVPLL does not solve for, or maintain, ambiguities (integer or float). Similar to Cellmer's MAFA method, the *integerness* of the ambiguities is naturally maintained within the algorithm itself [9]. Acquisition deals with estimating an initial state vector that is close enough to truth that the algorithm will converge on the correct state vector when tracking begins. This chapter begins with a review of past acquisition algorithm research, continues with the development of the DVPLL acquisition algorithm, and ends with results.

4.1 Past Research

This section contains a review of past research associated with carrier-phase quality acquisition algorithms. The algorithms referenced in this section all start by narrowing the search space in some way. Most do this by utilizing a carrier-smoothed code approach to obtain a position estimate to within approximately 2 m of the true position [35]. Within this search space there are usually many possible solutions, and each algorithm has its own method of discarding the incorrect solutions until the *optimal* solution is the only one that remains. The section begins with a cursory review of integer ambiguity resolution methods and ends with a review of the ambiguity function method, which is the method most similar to the approach being proposed in this dissertation.

4.1.1 Integer Ambiguity Resolution Methods.

Integer ambiguity resolution is by far the most popular method of obtaining a carrier-phase quality solution. Kaplan and Hegarty ignore errors and reduce the double-difference

carrier-phase equation down to [35]

$$\mathbf{DD}_{cp} = \mathbf{H}\mathbf{b} + \mathbf{N}\lambda \quad (4.1)$$

where

\mathbf{DD}_{cp} = (n-1)x1 column of carrier-phase double differences (m)

\mathbf{H} = (n-1)x3 matrix of differenced unit vectors for the two SVs (unitless)

\mathbf{b} = 3x1 column of baseline coordinates (m)

\mathbf{N} = (n-1)x1 column of integer ambiguities (unitless)

λ = carrier wavelength (m)

n = number of signals being tracked

A high-accuracy solution to \mathbf{b} requires resolution of the integer ambiguities, as well as, monitoring to ensure no cycle slips occur. The baseline coordinates can then be estimated using least squares on the overdetermined system of equations epoch-by-epoch. Many methods have been developed to solve for the integer ambiguities such as the fast ambiguity resolution approach of Frei and Beutler [18] and the least squares ambiguity search technique of Hatch [30]. More recently, a commonly-used method is the ambiguity decorrelation method of Teunissen [67].

These integer ambiguity resolution methods work in the measurement domain by first solving for the integer ambiguities in the set of real numbers, fixing those ambiguities as integers, and then finally solving for the baseline coordinates. Continuous monitoring is required to ensure cycle slips do not occur. A characteristic of these methods is that the number of possibilities to search increases geometrically with the number of SVs tracked. Some methods limit the search space by picking a subset of the tracked satellites to solve for the ambiguities and using the remainder as a check [30].

Teunissen's method efficiently finds the set of ambiguities that minimizes the integer least-squares function. This set may not necessarily be the true set of ambiguities [67]. A validation process is required to ensure the ambiguities are correct.

Three methods of validating the ambiguities are the F-ratio test of Counselman and Abbot [14], the W-ratio test of Wang [70], and the fixed failure rate ratio test of Verhagen and Teunissen [69]. These methods monitor the second *best* set of ambiguities and if the variance of the ambiguity residuals for the *best* set becomes better than the second *best* set by a set margin, the *best* set of ambiguities are considered the true set. If not, a float solution is maintained and new sets of ambiguities are obtained and tested in the next integration period. As time progresses, more measurements are obtained and geometry changes, so the correct set becomes more obvious.

These integer ambiguity resolution methods find, fix, and validate the ambiguities. However, the DVPLL does not maintain an integrated Doppler measurement and cannot use these techniques. Furthermore, these techniques generally operate on double-difference measurements to remove the relative receiver time error. The DVPLL measures and accounts for the time error in the algorithm.

One way to use integer ambiguity resolution techniques for DVPLL acquisition would be to operate a standard SW receiver on the data, find and fix the ambiguities, solve the baseline coordinates, and use those baseline coordinates along with, a small search in the time domain, to initialize the DVPLL. However, this approach does not take advantage of the vector nature of the DVPLL.

4.1.2 Ambiguity Function Method (AFM).

This method of measuring static baselines was pioneered by Counselman and Gourevitch in the early 1980s [15]. Results using the method were reported by Remondi [61]. The method works in the position domain and leverages the fact that if the correct baseline is used, the phase of the measured cross complex power will match the predicted

phase of the cross complex power for each satellite. The match of each SV is offset by the relative phase offset of the receiver clocks at each end of the baseline. This offset is removed by a magnitude operator in the final step of the ambiguity function. The two clocks must match within a quarter of a wavelength across the integration period. If double-difference (DD) measurements are used, the magnitude operator is no longer needed and the quarter-cycle requirement is removed. In equation form, the ambiguity function is given as

$$f(\hat{\mathbf{b}}) = \left| \sum_{i=1}^{nepochs} \sum_{k=1}^n e^{j(\phi_r^k - \phi_b^k - \frac{\rho^k(\hat{\mathbf{b}})}{\lambda})} \right| \quad (4.2)$$

where

f = ambiguity function

$nepochs$ = number of time epochs

$\hat{\mathbf{b}}$ = trial baseline

ϕ_r^k, ϕ_b^k = rover and base station phase measurements

ρ^k = difference between satellite to receiver distances

The function will be maximized if the trial baseline equals the true baseline. The method grids the search region and finds the maximum within that region. The volume around the maximum is then further gridded if a finer resolution of the baseline is needed.

Han and Rizos improved the reliability and computational efficiency of the AFM in dual-frequency receivers by taking advantage of various linear combinations and optimizing the step size of the search [29]. They concluded that 0.1λ is an optimal step size to use. They recommend a search region of $\pm 2\lambda$ for 6 SVs and $\pm 1.2\lambda$ for 5 SVs to reduce the number of maxima within the region. If 6 SVs are tracked, their search algorithm consists of using the $\phi_{-3,4}$ linear combination in a first stage, searching in 16 cm steps in a region $\pm 1.6\text{m}$ around the initial point. The linear combinations are formed from the L_1 and L_2 phase measurements as $\phi_{j,k} = j\phi_{L1} + k\phi_{L2}$. For more information on

linear combinations, see Appendix A. Once a maximum is found, the second stage consists of using the $\phi_{1,-1}$ linear combination and searching in 4 cm steps in a region ± 3 standard deviations around this maximum. The observations from both frequencies are then used separately in the final two stages, searching in 1 cm and then 0.2 cm steps in regions ± 3 standard deviations around the maximum found in the previous step.

Cellmer developed a modified ambiguity function approach (MAFA) and wrote several papers concerning its implementation and use [5–9]. Cellmer’s method operates on dual-frequency DD measurements and starts from an initial estimate of the rover’s position using DD-code measurements. The DD-phase observables are calculated and a misclosures vector is formed as [5]

$$\Delta = \text{round}\left(\phi_{DD} - \frac{\rho}{\lambda}\right) - \left(\phi_{DD} - \frac{\rho}{\lambda}\right) \quad (4.3)$$

where

Δ = misclosure vector (cycles)

ϕ_{DD} = DD-phase observables (cycles)

ρ = DD-geometric ranges using initial guess (m)

For short baselines, if the initial guess is correct, the difference $\phi_{DD} - \frac{\rho}{\lambda}$ is an integer and Δ is zero. Otherwise, Δ is a measure of the projected position error along the DD of the SV pointing vectors. Weighted least squares on the misclosures vector is used to adjust the initial guess resulting in a solution closer to the true position. The procedure is used in a cascaded approach which starts with the 1.6281 m wavelength $\phi_{-3,4}$ linear combination in the first step, then uses the 0.8619 m widelane linear combination, $\phi_{1,-1}$, and ends with the L_1 -only solution, $\phi_{1,0}$. Cellmer later added decorrelation of the ambiguities to improve efficiency, although he never defines or shows the efficiency improvement [6].

Due to the ambiguous nature of the phase measurements, there is only a small region around the true position where the method will converge. Cellmer develops the

conditions for convergence and a method of graphically representing the conditions [7]. The conditions reduce down to having the absolute value of the dot product of each double-difference pointing vector multiplied by the position error be less than half the wavelength at each step of the cascaded approach. The MAFA method has an 81% success rate for an hour-long session over a 2.5 km baseline and 5-SV constellation [6].

To increase the success rate of the approach, Cellmer went on to initiate a ± 1 wavelength search in each SV direction [8]. Cellmer also abandoned using the $\phi_{-3,4}$ [8] combination. This approach results in a 90.8% success rate for a 1.5 km baseline, 87.5% for 9.5 km and 79.1% for 29.1 km. Searching in this manner extends the convergence region to three times the size in each direction (or 27 times the volume). He gives a numerical example where convergence is achieved even though the a-priori position is more than 5 m offset from the true position. However, the absolute value of the dot product of the double-difference pointing vectors and the a-priori position error is at most 1.03 wavelength cycles, easily within the 1.5 cycle convergence region.

It is reasonable for MAFA to be effective for dual-frequency receivers with a good estimate of the a-priori position. However, the method will have difficulty for single frequency measurements (0.29 m convergence region along each double-difference pointing vector) and/or a poor a-priori position estimate. Also, the method is developed for double-difference measurements whereas, in the current configuration, the DVPLL uses single-difference measurements to solve for the clock offset. MAFA could be rederived for single-difference measurements by adding a clock term to the model. However, this would further shrink the already small convergence region since any clock error would add to each SV's phase offset. A similar method to the MAFA method that is designed to operate under the conditions required of the DVPLL is developed in the next section.

4.2 DVPLL Acquisition Algorithm

A 4-quadrant arc-tangent is used for the phase discriminator in the DVPLL and these values are not corrected for rollover in any way. In other words, the measurements are not integrated Doppler. The phase offset of each SV due to error in the initial state vector can be modeled using

$$\phi_V^i = \frac{\mathbf{e}^i \bullet (\mathbf{x}_t - \mathbf{x}_0)}{\lambda} + \delta t_t - \delta t_0 + w^i \quad (4.4)$$

where

ϕ_V^i = phase offset of i th SV (cycles)

\mathbf{x}_0 = initial estimate of position (m)

\mathbf{x}_t = true position (m)

δt_0 = initial estimate of time offset (cycles)

δt_t = true time offset (cycles)

w^i = error from all other sources (cycles)

However, the measurements, ϕ_0^i , obtained using the initial state vector are simply the non-integer portion of ϕ_V^i . These can be modeled as

$$\phi_0^i = \phi_V^i - \text{round}(\phi_V^i) \quad (4.5)$$

To ensure convergence to the correct state vector, the initial state vector must be close enough that the phase discriminators remain in the non-ambiguous ± 0.5 cycle region around the true state vector. In other words, the measurements are to remain in the region where $\text{round}(\phi_V^i)$ is zero for all i . If other errors are greatly reduced in the single-difference, the total phase error is composed of position error, relative time error, receiver noise and multipath. The addition of all these errors must be within ± 0.5 cycles for every phase measurement for convergence to occur. The following algorithm grids the search region in

such a way that at least one grid point is near enough to the true state vector for convergence to happen. This results in many candidate points which are then culled and validated.

Assume a search grid is established with a spacing in each spatial direction given by δ_g cycles containing the true position within its boundary. The maximum distance between the true position and the nearest grid point is $\frac{\sqrt{3}}{2}\delta_g$. This is the distance between the center of a cube, δ_g on a side, and one of the corners. The value, $\frac{\sqrt{3}}{2}\delta_g$, is also the maximum possible phase error due to the spatial grid, as would happen if an SV were aligned with a grid corner and the true position were in the center.

Furthermore, assume a search is performed in the time dimension using grid points δ_t cycles apart. This would cause the error due to time to be a maximum of $\frac{\delta_t}{2}$.

Ignoring receiver noise and multipath, the total phase error is the addition of the spatial error and the time error. Using the maximum values and combining this with the fact that the magnitude of the total phase error must be kept less than a half cycle leads to

$$\begin{aligned}\frac{\sqrt{3}}{2}\delta_g + \frac{\delta_t}{2} &< \frac{1}{2} \\ \delta_g &< \frac{\sqrt{3}}{3}(1 - \delta_t)\end{aligned}\quad (4.6)$$

The number of grid points searched over a region Δx , Δy , Δz m on a side is given by

$$N_{grid} = \left(\frac{\Delta x}{\lambda\delta_g}\right)\left(\frac{\Delta y}{\lambda\delta_g}\right)\left(\frac{\Delta z}{\lambda\delta_g}\right)\left(\frac{1}{\delta_t}\right) \quad (4.7)$$

$$= \frac{3\sqrt{3}\Delta x\Delta y\Delta z}{\lambda^3(1 - \delta_t)^3\delta_t} \quad (4.8)$$

The minimum of N_{grid} is at $\delta_t = \frac{1}{4}$. Any time grid that spans a full cycle will suffice. Examples are $\{-\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{2}\}$ or $\{-\frac{3}{8}, -\frac{1}{8}, \frac{1}{8}, \frac{3}{8}\}$. Solving for the value of δ_g yields $\frac{\sqrt{3}}{4}$ cycles.

For two frequency data (L_1 and L_2), used separately, a time bias is estimated for each frequency. The spatial search grid will be set up on the L_1 frequency since it has the smaller

wavelength. If the spatial grid is set at $\frac{\sqrt{3}}{4} L_1$ cycles then the maximum distance to a corner of a cube will be $\frac{3}{8}\lambda_1$. This is equivalent to $\frac{3}{8}\frac{\lambda_1}{\lambda_2} L_2$ cycles. To keep the total L_2 grid search error less than a half cycle

$$\frac{3}{8}\frac{\lambda_1}{\lambda_2} + \frac{\delta_{t,L_2}}{2} \leq \frac{1}{2} \quad (4.9)$$

$$\delta_{t,L_2} \leq 1 - \frac{3}{4}\frac{\lambda_1}{\lambda_2} \quad (4.10)$$

$$\delta_{t,L_2} \leq 1 - \frac{3}{4}\frac{60}{77} \quad (4.11)$$

$$\delta_{t,L_2} \leq 0.4156 \quad (4.12)$$

The largest evenly spaced interval to meet this criterion would be $\frac{1}{3}$ cycles for an L_2 time grid of $\{-\frac{1}{3}, 0, \frac{1}{3}\}$. This is the grid that must be established in the L_2 time domain for each grid point in the L_1 time domain, yielding a final result of 12 time grid points for each spatial grid point when treating the L_1 and L_2 measurements separately.

This form of gridding is conservative and guarantees a grid point will be within the convergence region of the true state vector under the worst case condition if no noise is present. Under the worst case condition that the true position is in the center of a grid cube, and the time offset is off by exactly $\frac{1}{8}$ cycle, the probability of an erroneous measurement would be $\frac{1}{2}$ for each nearby grid point. If error pushes the phase measurement outside the ± 0.5 cycle range at one corner of the grid cube, however, it will pull it within the range for the opposite grid point. Under other conditions, several grid points will be within the convergence region of the true state vector.

If two-frequency data are available, the number of points searched can be lowered by creating the appropriate linear combination. For example, the wavelength of the widelane linear combination $\phi_{1,-1}$ is 0.86 m, reducing the number of points searched by a factor of about 93. For dual-frequency measurements, the following algorithm starts by finding solutions using the widelane linear combination in stage 1 and proceeding to find solutions using the L_1 and L_2 measurements separately in stage 2. For single-frequency receivers,

only the first stage of the algorithm is used. The algorithm is not limited to these two choices. Other combinations, for dual- or tri-frequency measurements, could easily be used, such as widelane ($\phi_{1,-1}$) in stage 1 to ionospheric reduced ($\phi_{4,-3}$) in stage 2. The steps would need to be modified appropriately for wavelength and number of measurements. The acquisition algorithm is as follows.

4.2.1 Stage 1: Widelane or L1-Only Solutions.

- *Obtain Initial State Vector Estimate.* First determine an a priori position and its associated covariance matrix along with an estimate of the relative clock time and frequency offset. This step is performed using standard differential-code techniques [35].
- *Get Phase Measurements.* The initial state vector estimate is used in the DVPLL to obtain L₁ and, if appropriate, L₂ single-difference phase measurements.
- *Grid Search Space.* The initial state vector and covariance matrix are used to construct a prediction-interval position ellipse which is gridded in the spatial ($\frac{\sqrt{3}}{4}$ cycle spacing) and time dimensions ($\frac{1}{4}$ cycle spacing) using the appropriate wavelength (widelane or L1). The prediction-interval position ellipsoid is obtained using the 3x3 portion of the covariance matrix that pertains to the position states \mathbf{P}_p as

$$(\mathbf{x}_{grid} - \hat{\mathbf{x}}_p)^T \mathbf{P}_p^{-1} (\mathbf{x}_{grid} - \hat{\mathbf{x}}_p) \leq \chi_{p|3}^2 \quad (4.13)$$

where

\mathbf{x}_{grid} = xyz candidate grid point

$\hat{\mathbf{x}}_p$ = initial position estimate

\mathbf{P}_p = initial position covariance matrix

$\chi_{p|3}^2$ = value to ensure probability of keeping correct solution is p

The value $\chi_{p|\nu}^2$ is found by

$$\int_0^{\chi_{p|\nu}^2} \chi_{\nu}^2(x) dx = p \quad (4.14)$$

where

$\chi_{\nu}^2(x)$ = chi-squared probability density function

ν = the number of degrees of freedom

- *Estimate Phase Measurement at Each Grid Point.* The single-difference phase measurements are translated to each grid point using

$$\phi_{GPj}^i = \phi_0^i - \frac{\mathbf{e}^i \cdot \Delta \mathbf{x}_{GPj}}{\lambda} + \delta t_{GPj} \quad (4.15)$$

where

ϕ_{GPj}^i = phase measurement for SV i at grid point j (cycles)

ϕ_0^i = phase measurement for SV i at initial position (cycles)

\mathbf{e}^i = pointing vector from SV i to initial position (unitless)

$\Delta \mathbf{x}_{GPj}$ = vector from initial position to grid point j (m)

λ = wavelength (m)

δt_{GPj} = time offset for grid point j (cycles)

Note that δt_{GPj} cycles through $\{-\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{2}\}$ for each spatial grid point. The phase measurements are then moved to the range $-0.5 < \phi_{GPj}^i < 0.5$ by

$$\phi_{GPj}^i = \phi_{GPj}^i - \text{round}(\phi_{GPj}^i) \quad (4.16)$$

where

round = function that returns nearest integer

This creates phase measurements as if the tracking had been accomplished at each grid point.

- *Estimate Position and Time Adjustment for Each Grid Point.* The weighted least-squares estimate of the position and time adjustment is obtained as

$$\hat{\delta \mathbf{x}}_{GPj} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \boldsymbol{\phi}_{GPj} \quad (4.17)$$

where

$\hat{\delta \mathbf{x}}_{GPj}$ = position and time adjustment for grid point j (m)

\mathbf{H} = design matrix

\mathbf{R} = measurement covariance matrix

$\boldsymbol{\phi}_{GPj}$ = column vector of grid point j phase measurements (cycles)

The design matrix consists of rows with \mathbf{e}^i as the first 3 columns and a 1 in final column all divided by λ . The covariance matrix \mathbf{R} is a matrix with the single-difference variances of the measurements along the diagonal and zeros elsewhere.

- *Calculate Weighted Sum of Squares of the Residuals.* The least squares residuals, \mathbf{res}_j are estimated as

$$\mathbf{res}_j = \boldsymbol{\phi}_{GPj} - \mathbf{H} \hat{\delta \mathbf{x}}_{GPj} \quad (4.18)$$

The weighted sum of squares of the residuals is then calculated

$$\Sigma_{resj}^2 = \mathbf{res}_j^T \mathbf{R}^{-1} \mathbf{res}_j \quad (4.19)$$

- *Keep Solutions with Σ_{resj}^2 Less than Threshold.* The sum of squares of the residuals are then compared to the threshold value of

$$\Sigma_{th}^2 = \chi_{p|n-4}^2 \quad (4.20)$$

where

$$\Sigma_{th}^2 = \text{threshold value (unitless)}$$

If Σ_{resj}^2 is less than the threshold, then if there is a stage 2, the first three elements of

$$\Delta \mathbf{x}_{SP} = \begin{bmatrix} \Delta \mathbf{x}_{GPj} \\ \delta t_{GPj} \end{bmatrix} + \hat{\delta} \mathbf{x}_{GPj} \quad (4.21)$$

are passed to stage 2; otherwise, if there is no stage 2, then all four elements are passed to stage 3 as a candidate solution.

- *Estimate Covariance Matrix for Solutions.* The covariance matrix for the stage 1 solutions is given by

$$\mathbf{P} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \quad (4.22)$$

4.2.2 Stage 2: Individual L_1 and L_2 Measurement Solutions Around Widelane Solutions.

The following steps are performed for each widelane solution found in stage 1.

- *Calculate Phase Measurements at Solution Point.* The phase measurements estimated at the initial point are translated to the solution point

$$\phi_{SP}^i = \phi_0^i - \frac{\mathbf{e}^i \bullet \Delta \mathbf{x}_{SP}}{\lambda} \quad (4.23)$$

where

ϕ_{SP}^i = phase measurements for SV i at the solution point (cycles)

$\Delta \mathbf{x}_{SP}$ = vector from initial point to solution point (m)

- *Grid Search Space.* The covariance matrix, estimated at the end of stage 1, is used to construct a search region which is gridded as before except using the shorter L_1 wavelength.
- *Estimate Phase Measurements at Each Grid Point.* The single difference phase measurements are translated to each grid point using

$$\phi_{GPj}^i = \phi_{SP}^i - \frac{\mathbf{e}^i \bullet \Delta \mathbf{x}_{GPj}}{\lambda} + \delta t_{GPj} \quad (4.24)$$

where

ϕ_{GPj}^i = phase measurement for SV i at grid point j (cycles)

$\Delta \mathbf{x}_{GPj}$ = vector from solution point to grid point j (m)

λ = L_1 or L_2 wavelength (m)

δt_{GPj} = time offset for grid point j (cycles)

Note that δt_{GPj} cycles through $\{-\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{2}\}$ for L_1 and $\{-\frac{1}{3}, 0, \frac{1}{3}\}$ for L_2 yielding a total of 12 time grid points for each spatial grid point. The phase measurements are then adjusted to the range $-0.5 < \phi_{GPj}^i < 0.5$ as in (4.16).

- *Estimate Position and Time Adjustment for Each Grid Point.* The position and time adjustment is estimated as in stage 1 except the design matrix has $2n$ rows and 5 columns. The first n rows are constructed with $\frac{e^i}{\lambda_{L_1}}$ as the first 3 columns, a $\frac{1}{\lambda_{L_1}}$ in the 4th column, and a zero in the final column. The last n rows are constructed with $\frac{e^i}{\lambda_{L_2}}$ as the first 3 columns, a zero in the 4th column, and a $\frac{1}{\lambda_{L_2}}$ in the final column. This design matrix estimates separate time bias terms for each frequency. The covariance matrix \mathbf{R} is also augmented to include the L_2 variances.
- *Calculate sum of squares of Residuals.* The weighted sum of squares of the residuals is estimated as in (4.18).
- *Keep Solutions with Sum of Squares Less than Threshold.* This step is the same as in stage 1 except the threshold value is calculated as $\Sigma_{th} = \chi_{p|2n-5}^2$. If the sum of squares of the residuals Σ_{resj}^2 is less than the threshold then the solution

$$\Delta \mathbf{x}_{sol} = \begin{bmatrix} \Delta \mathbf{x}_{SP} \\ \delta t_{GPj} \end{bmatrix} + \hat{\delta \mathbf{x}}_{GPj} \quad (4.25)$$

is saved in the collection of stage 2 solutions and passed to stage 3 as a possible solution.

- *Estimate Covariance Matrix for Solutions.* The covariance matrix for the stage 2 solutions is estimated as in (4.22).

4.2.3 Stage 3: Validation: Propagate Solutions in DVPLL and Eliminate Candidates Until One Left.

Due to the density of the search, many of the time grid points will converge to the same solution. The candidate solutions are searched. Any that are within a tenth of a wavelength of each other are replaced by the solution with the lowest sum of squares of the residuals.

Each unique candidate solution is propagated epoch-by-epoch in the DVPLL and allowed to stabilize. Following the stabilization period the weighted sum of squares of the phase residuals of the solutions is checked against a 97% threshold each second. If the value is below the threshold a *success* is registered for that solution. Every minute the number of successes is checked against a binomial distribution at the 99.99% level (for a 97% success rate and 60 trials this would be 52 successes). Solutions that meet the binomial threshold are kept and all others are eliminated. The last solution left is deemed the correct solution.

In summary candidates are found in the first epoch and, if necessary, incorrect candidates are eliminated until the final solution remains. Tracking of all remaining candidates is maintained throughout stage 3 so there is no need to go back and track past samples again. The final candidate's full accuracy solution is already recorded. When propagating and updating the candidates in the Kalman filter, the covariance matrix and hence the Kalman gain are the same for each candidate. The phase measurements only need to be obtained for one candidate and translated to the positions of the other candidates. For this algorithm, the remaining candidate with the lowest \sum_{resj}^2 is chosen as the candidate to use to obtain the phase measurements.

4.2.3.1 Failure Rate.

A successful DVPLL acquisition depends on the correct state vector being among the initial candidates and keeping this state vector until all erroneous candidates are eliminated. If all candidates are eliminated, the process restarts with a fresh set of candidates. If the single candidate left following the last binary check is an erroneous one, however, the acquisition is a failure. The situation is illustrated in Figure 4.1.

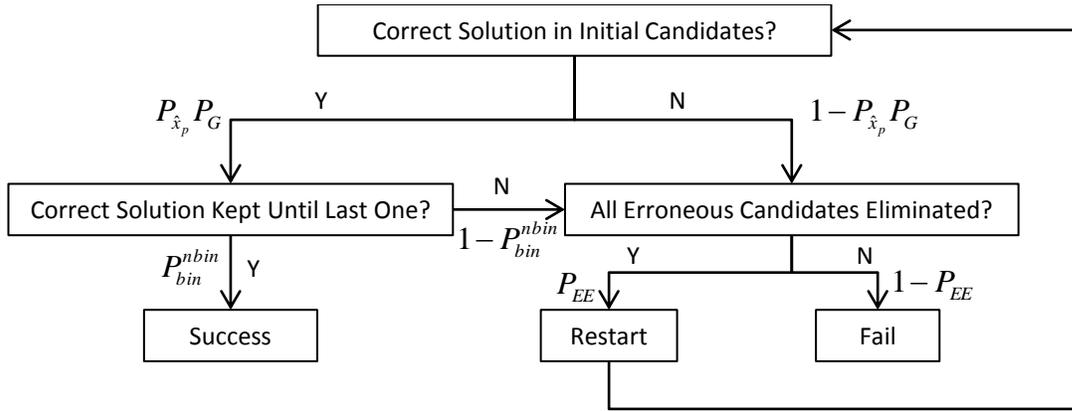


Figure 4.1: Probability Tree for DVPLL Acquisition

The probability of eliminating all erroneous candidates, P_{EE} , is very difficult to calculate and depends heavily on a number of factors. These factors include the initial number of erroneous candidates, the geometry of the SVs compared to the error vector, how quickly the geometry changes, etc. However, it is apparent from Figure 4.1 that the probability of a successful acquisition is bounded by

$$P_S \geq P_{\hat{x}_p} P_G P_{bin}^{nb} \quad (4.26)$$

where

$P_{\hat{x}_p}$ = probability initial solution within ellipsoid given by (4.13)

P_G = probability a grid point is within convergence region of true point

P_{bin} = probability correct candidate passes single binary check

$nbin$ = number of binary checks until erroneous candidates eliminated

The probabilities $P_{\hat{x}_p}$ and P_{bin} are set by the user and the search region is gridded in such a way that the probability of convergence is very close to one if the phase errors are reasonable. The tradeoffs associated with how often to perform the binary checks and where to set threshold values has not been performed. For example, performing the binary checks more often will allow earlier detection of the last erroneous candidate but will increase $nbin$ and, consequently, decrease the probability of success.

4.3 Results

4.3.1 24-h Test Using CORS Data.

A full GPS day's worth of data were downloaded from two CORS sites, PRDU and INWL, for 3 Sep 13. The baseline distance between the two sites is 3.6 km. The single-difference phase errors were calculated using the surveyed locations of each site and are plotted in Figure 4.2 versus elevation. No correction was made for antenna phase variations or phase center offsets. The phase errors were then binned into 1 degree elevation increments and rms values estimated. The rms values for L_1 and L_2 are shown in Figure 4.3 versus elevation along with an empirically derived formula for each. These formulas were used to create the \mathbf{R} matrices. There were a number of measurements with half-cycle errors below 12 degrees so the elevation mask was set at this level.

When using CORS data for both the base station and rover, in-phase and quadrature single-difference measurements were replicated as if raw sampled data were used in the

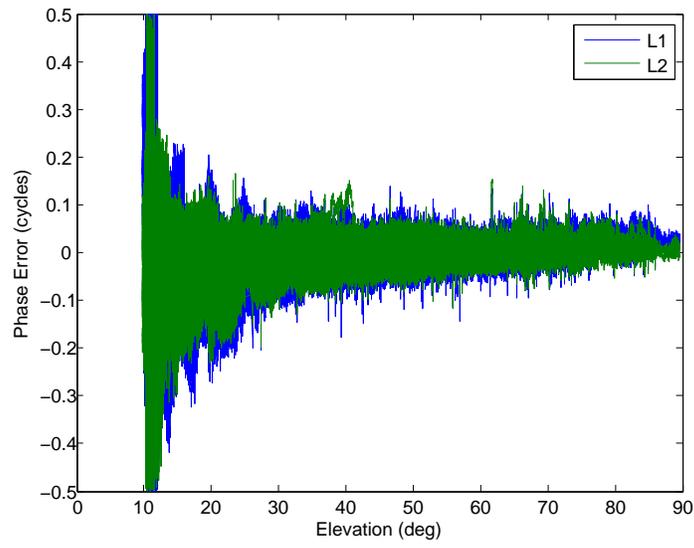


Figure 4.2: Single-Difference Phase Errors vs Elevation, 24-h period, 3 Sep 13

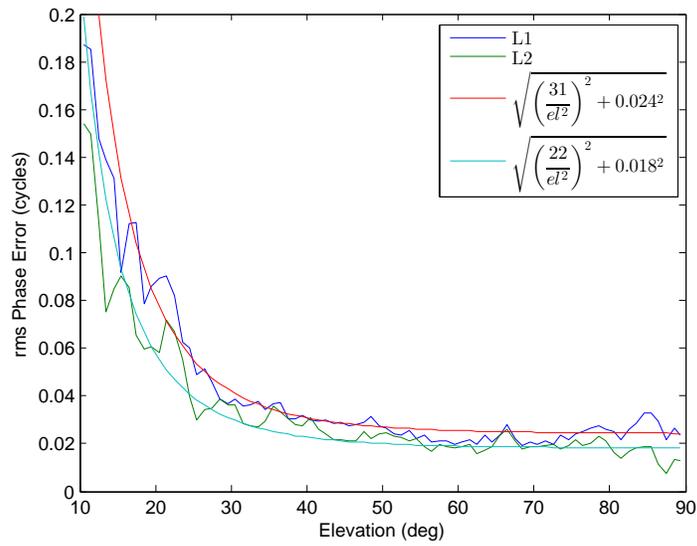


Figure 4.3: Single-Difference Phase Error rms vs Elevation, 24-h period, 3 Sep 13

DVPLL correlators. As shown in Figure 4.4, this was done by first subtracting the translated base station phase measurements from the rover phase measurements. The cosines of the differences were then used as the in-phase measurements and the sines of the differences were used as the quadrature measurements. Code measurements were not used.

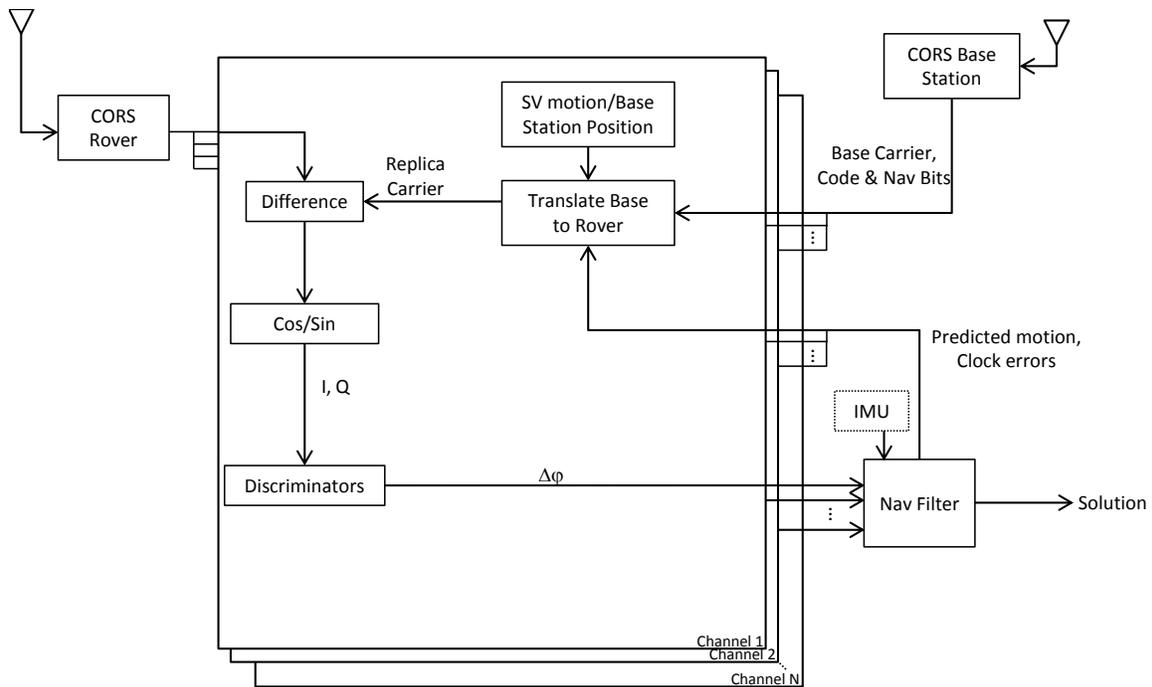


Figure 4.4: Differential Vector Phase Locked Loop Flow Chart Using CORS Data

The efficacy of the acquisition algorithm was then tested. Starting at the beginning of the file, an initial position error was simulated by adding a ± 1.5 m uniformly distributed random offset to each axis of the rover's surveyed location. Instead of using a search region based on a prediction-interval ellipsoid, a ± 2 m search region in each axis was set to ensure the true solution was contained in the search region. The rest of the algorithm was then executed until a single solution remained. If the solution was within 8 cm of the surveyed position the trial was considered a success. If not, the trial was a failure. The algorithm was

reinitialized and the time point following the end of the previous trial was used to restart the algorithm. This continued until the end of the file was reached. The success rate was estimated as the number of successes divided by the total number of trials.

The algorithm was executed twice, once using both the L_1 and L_2 frequencies and a second time for L_1 -only processing.

4.3.1.1 L_1/L_2 Case.

For this case, the threshold value $\chi_{0.97|n-4}^2$ was used in stage 1 and $\chi_{0.97|2n-5}^2$ was used in stage 2. During stage 3, the threshold was set at $\chi_{0.97|2n-6}^2$ and the binomial threshold for 60 trials was set at 52.

Figure 4.5 shows the 3D error of the final solution versus time. If all solutions were eliminated in stage 3, a value of -1 was inserted. There were 54115 solutions, 2 of which had all solutions eliminated in stage 3. Elimination of all solutions is not a failure since the algorithm is just restarted to find a fresh set of candidates. Only one solution's 3D error was greater than 8 cm, giving a success rate of almost 100%. For the erroneous solution, the correct solution was eliminated prior to the start of stage 3. Most acquisitions resulted in a single solution out of stage 2. These *single-epoch* solutions did not need a stage 3. The average time to fix the correct solution (TTF) is plotted in Figure 4.6 versus number of satellites tracked. The average time is only a few seconds except for the 5 SV case, suggesting a finer check interval could have been used. Determining the optimal check interval is a subject of further investigation. All cases, with 10 SVs or more, had only single-epoch solutions so the average TTF is zero for those cases.

There were 53674 single-epoch solutions (only a single solution out of stage 2) out of 54115 or 99.2%. This number is an optimistic estimate since, if there was more than one solution, stage 3 used at least 60 seconds of data to eliminate solutions. Hence, other nearby epochs which may have poor conditions for obtaining a single-epoch solution were also removed. With this in mind, stages 1 and 2 of the algorithm were reaccomplished

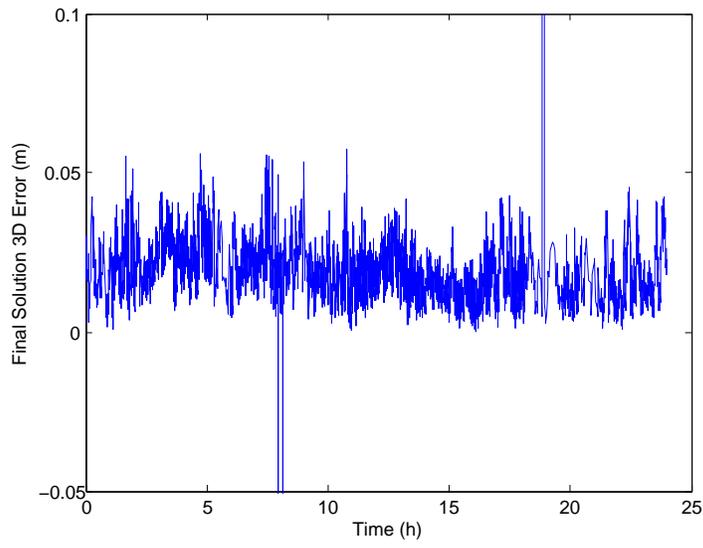


Figure 4.5: 3D Position Error of Final Solution L_1/L_2 , 24-h period, 3 Sep 13

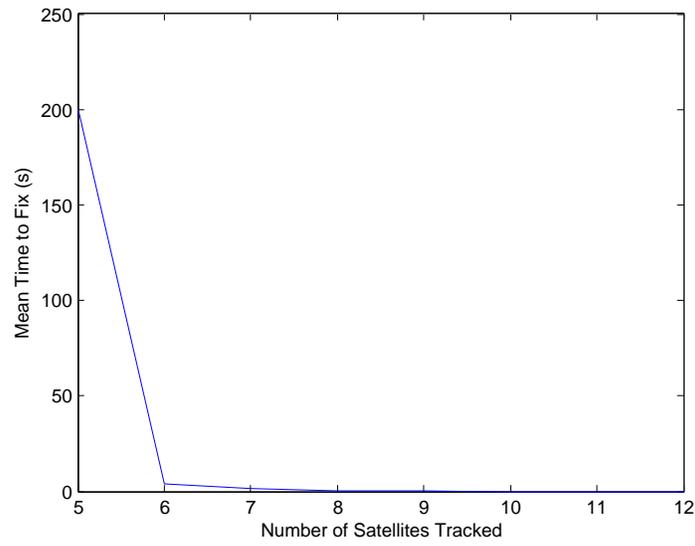


Figure 4.6: Mean Time to Fix L_1/L_2 , 24-h period, 3 Sep 13

every epoch, skipping stage 3, and the resulting number of candidates recorded. Figure 4.7 shows the 3D error of the candidate closest to the true solution (minimum 3D error) and number of satellites tracked versus time for this run. There were 89 epochs with a minimum 3D error greater than 8 cm, giving a 0.1% chance of not having the true solution in the initial set of candidates. Figure 4.8 shows the single-epoch success rate versus the number of satellites tracked. As expected, the single-epoch success rate increases as the number of satellites tracked increases. The average number of solutions versus the number of satellites is shown in Figure 4.9. The numbers are also listed in Table 4.1.

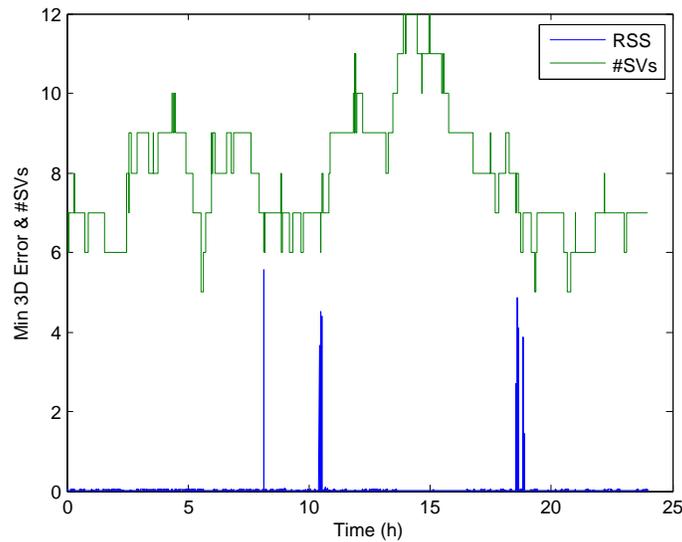


Figure 4.7: Minimum 3D Error of Candidates and #Satellites Tracked L_1/L_2 No Stage 3, 24-h period, 3 Sep 13

There are a couple of ways to increase the single-epoch success rate. First, increasing the accuracy of the a-priori position will reduce the search region and, therefore, the chance

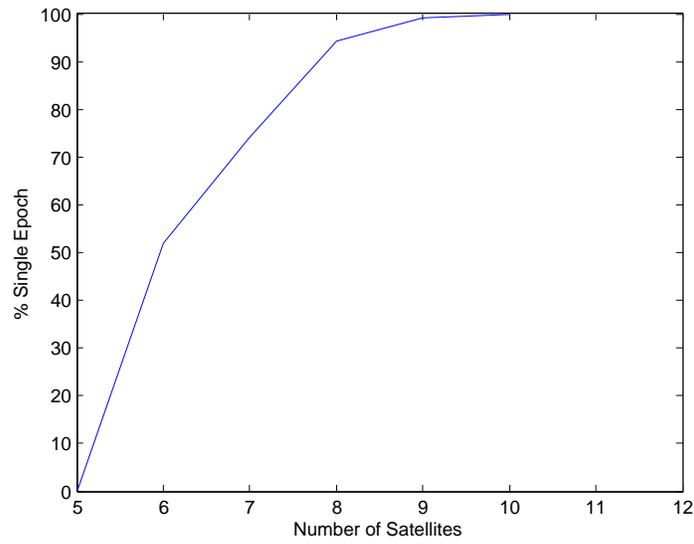


Figure 4.8: Percentage of Single-Epoch Solutions vs Satellites Tracked L_1/L_2 No Stage 3, 24-h period, 3 Sep 13

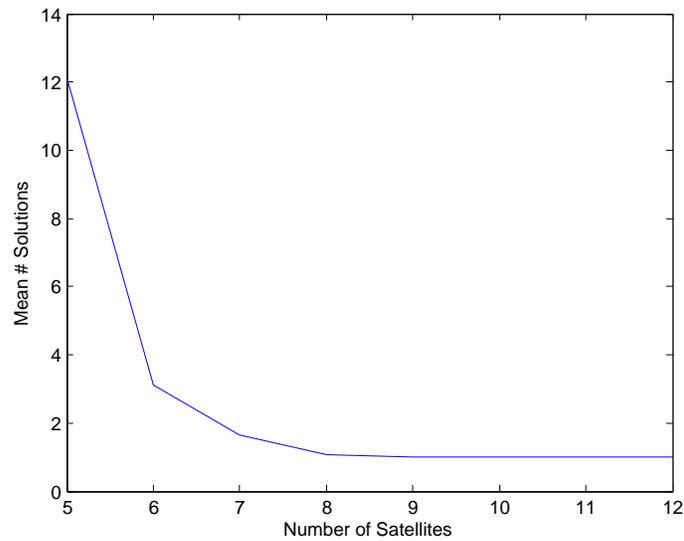


Figure 4.9: Average Number of Solutions at End of Stage 2 vs Satellites Tracked L_1/L_2 No Stage 3, 24-h period, 3 Sep 13

Table 4.1: L₁/L₂ Single Epoch Results, 24-h Period, 3 Sep 13

n	#Single Epoch	Total	%	Mean # Sol'ns
5	1	1022	0.1	12.1
6	5633	10838	52.0	3.1
7	21721	29366	74.0	1.7
8	12467	13237	94.2	1.1
9	21357	21551	99.1	1.0
10	3435	3435	100.0	1.0
11	4599	4599	100.0	1.0
12	2175	2175	100.0	1.0
All	71388	86233	82.8	1.6

an erroneous solution will meet the threshold requirement. Figure 4.10 shows a cumulative distribution of the distance between the second closest solution and the true solution, for those trials containing more than one solution. This is the result of a ± 1.5 m uniformly distributed initial error and a ± 2 m search region. Grid values as far as 6.1 m from the true solution are expected. It is easy to see, if the initial error and the grid search region are reduced, many more of the acquisitions would become single epoch. Second, decreasing the thresholds would decrease the number of erroneous solutions included in the list of candidates at the expense of an increased risk of also eliminating the correct solution.

4.3.1.2 L₁-Only Case.

For this case the threshold value $\chi_{0.997|n-4}^2$ was used in stage 1 and there was no stage 2. The Stage 3 threshold was set at $\chi_{0.97|n-5}^2$, and the binomial threshold was set as in the dual-frequency case. There are $n - 5$ degrees of freedom during stage 3, so at least six

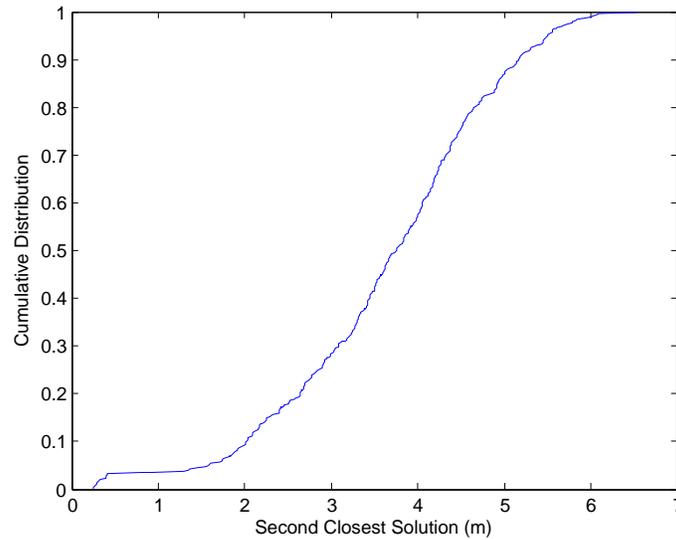


Figure 4.10: Distance from True Position to Second Best Solution for non-Single Epoch Trials L₁/L₂, 24-h period, 3 Sep 13

satellites are needed to form a threshold. The candidates were simply propagated during the epochs when fewer than six satellites were tracked, leading to longer acquisition times for these trials.

As expected, the L₁-Only case gave many more solutions out of stage 1 and took longer to discern the correct solution among these. These results are shown in Figures 4.11 and 4.12 and Table 4.2. There is a sharp reduction in time to fix and number of initial solutions as the number of satellites tracked increases to 9. Figure 4.13 shows the rss error of the final solution versus time. There were 303 acquisitions, and 100% successfully obtained the correct solution.

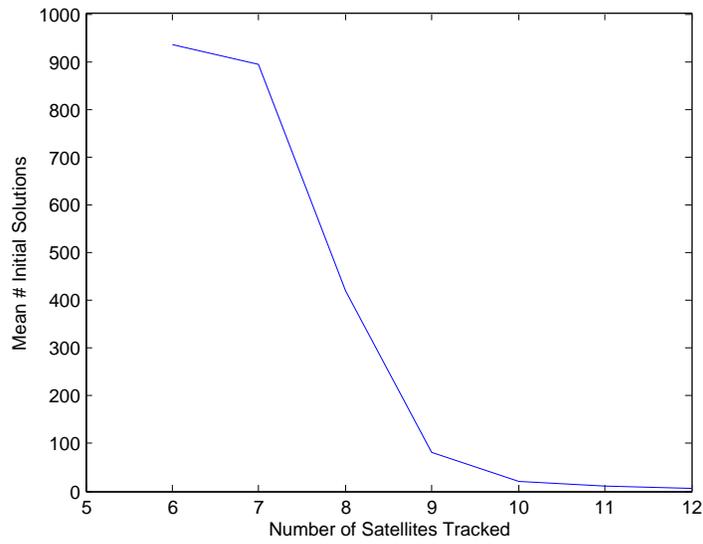


Figure 4.11: Average Number of Solutions at End of Stage 1 vs Satellites Tracked L_1 -Only, 24-h period, 3 Sep 13

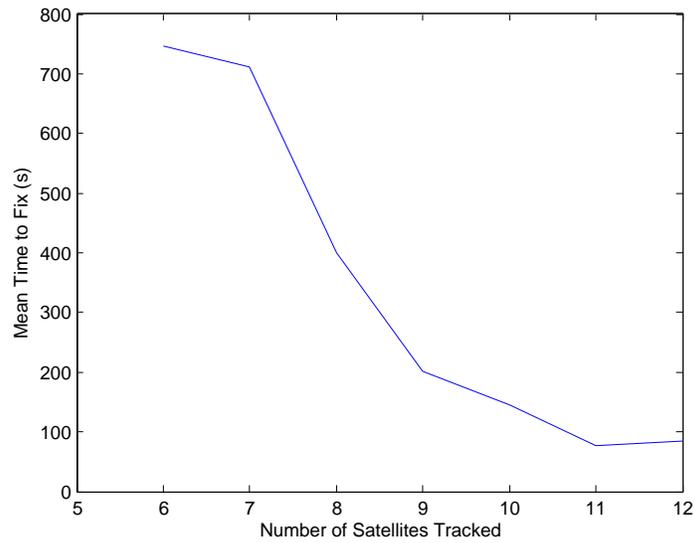


Figure 4.12: Mean Time to Fix L_1 -Only, 24-h period, 3 Sep 13

Table 4.2: L_1 -Only Acquisition Results, 24-h Period, 3 Sep 13

n	Mean Time to Fix (s)	Mean # Initial Solutions
6	747	937
7	711	895
8	399	421
9	202	81
10	145	20
11	78	10
12	85	6

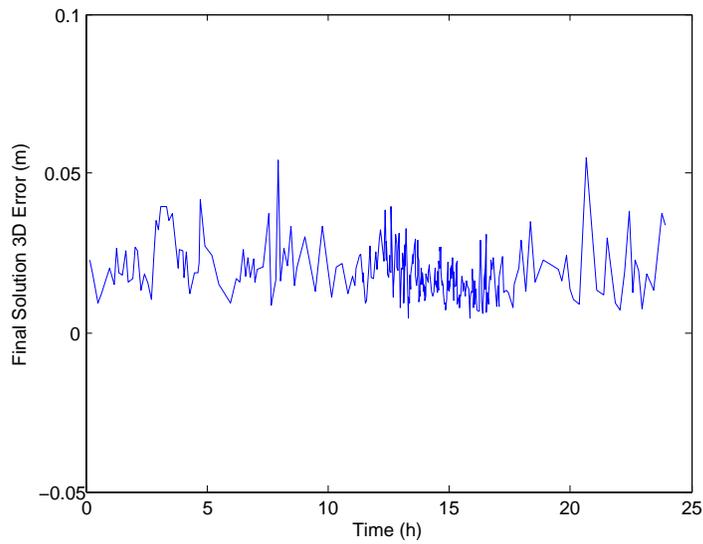


Figure 4.13: 3D Position Error of Final Solution L_1 -Only, 24-h period, 3 Sep 13

Figure 4.14 shows the candidate 3D position error versus the weighted sum of squares of residuals over the threshold for a single epoch. The figure is zoomed in to spread out the candidates with small weighted residuals. The correct solution is the one with the smallest 3D position error. The figure shows that there are several solutions with smaller weighted residuals than the correct solution. This was true in general for the L_1 -Only case, making ratio tests a poor validation choice for this method.

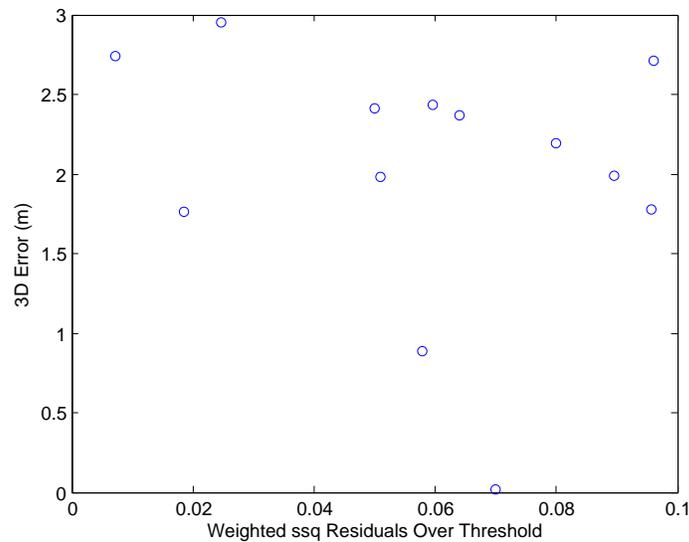


Figure 4.14: Candidate 3D Position Errors vs Weighted Sum of Squares of Residuals Over Threshold, L_1 -Only, Single Epoch, 3 Sep 13

4.3.2 15-min Test Using TRIGR as Rover and NovAtel as Base Station.

This test used the same data and \mathbf{R} matrix as reported in Section 3.5.3. The binomial distribution threshold was set at 6 out of 10 vice 52 out of 60, as in the test with CORS data, giving a threshold check every 10 seconds. The TRIGR data were L_1 -only, so only Stages 1 and 3 were accomplished. There were 8 SVs for most of the first acquisition and 9

SVs during the remaining acquisitions. The acquisitions converged on the correct solution 100% of the time with results detailed in Table 4.3.

Table 4.3: L₁-only DVPLL Acquisition Success Rate, 15-min Period, 13 Dec 13

Rover	Zero Baseline	18.5 m Baseline
Blue	21/21	18/18
Yellow	21/21	16/16

Figure 4.15 shows a sample of one short-baseline acquisition. Stage 1 resulted in 12 possible solutions. After 10 seconds 8 solutions were eliminated. The remaining incorrect solutions were eliminated at 30, 40 and 70 seconds into the acquisition. This left the final correct solution to continue tracking. There were many fewer initial candidates in these data than the CORS data due to the single-difference phase errors being much smaller. This is seen by comparing Figures 3.17 and 4.3. Figure 4.16 shows the weighted sum squared residuals divided by the threshold each second during this same acquisition. During each 10 second interval, the number of values greater than one were counted and if the total was 6 or more the solution was retained. It can be seen that the two best candidates are indistinguishable at the start of the acquisition. However, near the 30 second point the second best candidate started to diverge and the best candidate is clearly seen.

4.3.3 Conclusion.

A method is presented for initializing the DVPLL. The method includes an algorithm for generating candidates (estimation) as well as discerning the correct solution among them (validation). The method is shown to work for single- and dual-frequency measurements using CORS data. The test results also shows the importance of using dual-

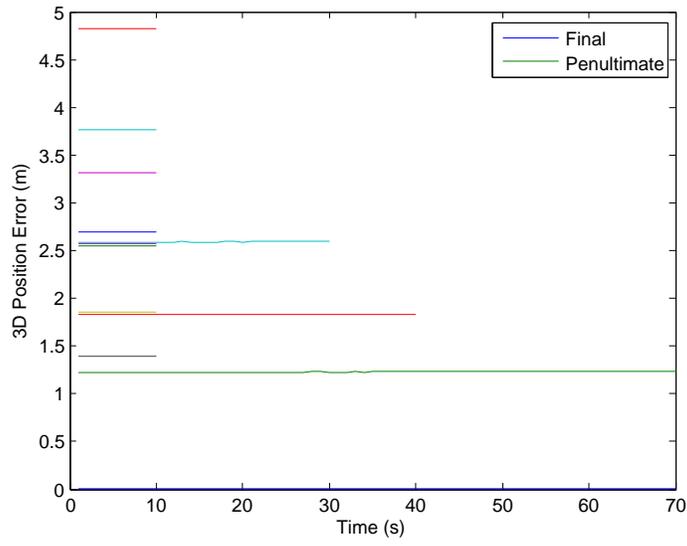


Figure 4.15: Sample 3D Position Error of Candidate Solutions vs Acquisition Time, 13 Dec 13

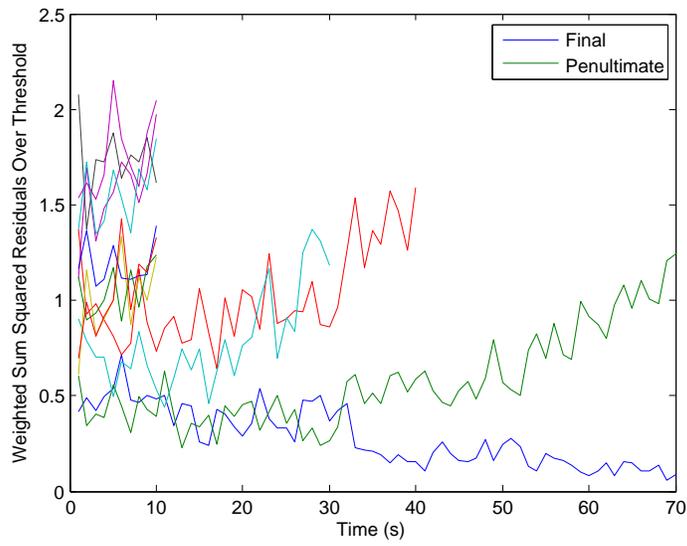


Figure 4.16: Sample Weighted Sum of Squares of Residuals Divided by Threshold vs Acquisition Time, 13 Dec 13

frequency data during the acquisition process. There is a very high probability of getting the correct answer in a single-epoch and, if not, the true solution can be found very quickly among the candidates.

Unlike the integer ambiguity search techniques, the number of grid points does not increase geometrically with the number of signals. In fact, it may decrease as the increased number of signals will make the a-priori position more accurate and, hence, reduce the search region. Also, unlike the MAFA, the convergence region is not limited to a small volume around the true position. The search region of the DVPLL can be made as large as necessary to ensure the true position is contained. As with other methods, the validation time of the DVPLL acquisition method also decreases with an increased number of satellites. This is important given the number of GNSS signals there are to exploit. The DVPLL also maintains full-accuracy for the correct solution throughout the validation phase, obviating the need to go back and track the sampled data again for full accuracy.

V. Conclusions and Recommendations

This document shows an integer ambiguity-resolved quality solution can be obtained and maintained directly in the vector phase tracking loop of a rover receiver. This chapter outlines the research contributions made by this dissertation and recommends areas for future research.

5.1 Research Contributions

5.1.1 DVPLL Tracking.

This research contributes a method of vector tracking that uses phase and code measurements from a base station receiver and applies them in the creation of reference signals in a rover receiver. The phase and code measurements in the base station contain errors common to both receivers that end up canceling those in the rover, leaving much smaller differential errors. For shorter baselines, these residual errors are small enough that they can be ignored. It appears that no other method uses base station phase measurements directly in the vector tracking loop of a rover receiver, therefore this method is novel. In this case, a vector phase locked loop can be created using only navigation and clock states without the need to model individual channel information. Other implementations must use numerous states to account for slowly changing phase errors on each channel. The DVPLL demonstrates an integer ambiguity resolved quality solution directly in the tracking loops. No other approach claims this level of accuracy directly in the vector tracking loops of a rover receiver.

5.1.2 DVPLL Acquisition.

The DVPLL tracking method assumes the state vector is within the pull-in region of the algorithm for the current integration period. This dissertation further contributes an acquisition method that ensures these conditions are met for the initial integration

period. The technique extends Cellmer's MAFA method [5] to single-frequency and single-difference measurements and greatly increases the convergence probability by optimally gridding the search region. The algorithm only requires an initial state vector and covariance matrix for the rover and phase and code measurements from a base station as inputs. This work also contributes a validation method where the user can set an upper bound on the failure rate by defining various parameters. In contrast to other approaches that only monitor the *best* and *second-best* candidates, this algorithm monitors all candidates, rejecting those that do not meet certain criteria until only the final solution is left.

5.2 Future Work

5.2.1 Longer Baselines.

For longer baselines the differential errors are no longer small enough to ignore. Two methods can be used to overcome this limitation. The first method models and removes the differential errors. The second method would be to create a network of base stations and optimally combine their measurements as in the work of Raquet [60]. Raquet's NetAdjust method is based on double-difference measurements which removes any biases associated with base station clock offsets. It is recommended that Raquet's method be revisited with the single-difference measurements required by the DVPLL in mind. The best method depends on the application and a study of the tradeoffs is recommended.

5.2.2 High Accuracy in High Dynamics.

The DVPLL is demonstrated to work very well in a static short baseline environment. Vector tracking has been shown to work better than scalar tracking in dynamic situations since the SVs aid each other. However, this benefit has not been demonstrated for the DVPLL. A study of the DVPLL's behavior under various dynamic conditions is recommended, paying attention to the effects of antenna attitude on satellite shading. For real flight tests an airworthy GNSS front end is needed. It is recommended that a GNSS

front end be found or developed that is capable of operating in a T-38 test environment. Special care must be given to the acceleration and vibration sensitivity of the reference oscillator.

5.2.3 Deep Integration with an IMU.

Deeply integrating the DVPLL with an IMU would make for a powerful combination especially under conditions where periodic outages are expected such as during a barrel roll or GPS jamming. It is recommended the DVPLL be deeply integrated with an IMU.

5.2.4 Smoother.

Tracking signals in recorded IF data allows the DVPLL to take advantage of future as well as past measurements. It is recommended that a smoother be implemented in the DVPLL.

5.2.5 Other GNSS Signals.

The current embodiment of the DVPLL tracks the GPS L1 CA-Code only. The method has been shown to work on dual-frequency CORS data using measurements from scalar tracking loops. Increasing the number of signals tracked by the DVPLL increases the accuracy, availability, robustness, etc. The DVPLL can be expanded to track all GNSS signals available if the front end is capable of recording them. Finding or developing a GNSS front end capable of recording data for all available GNSS signals is recommended. It is further recommended that the DVPLL be expanded to vector track these signals and that a base station be found that can provide carrier-phase and code-phase measurements, as well as the navigation data bits associated with each available GNSS signal.

5.2.6 Real-Time DVPLL.

Two obstacles need to be overcome to make a real-time DVPLL. First the current version assumes the data bits are known. This limitation can be overcome by using a dataless channel such as the L₅ pilot channel or by deriving the DVPLL to work with unknown data. The integrations would then have to be within data bit boundaries or loss

of signal power would occur. A two-quadrant arctangent would also have to be used, limiting the convergence region to ± 0.25 wavelengths. The second obstacle is the latency introduced by the transmission of the base station data and the associated extrapolation needed.

5.2.7 Acquisition Tradeoffs.

Further study of the acquisition threshold settings and binary check frequency under various conditions to find the optimal settings for a given situation is recommended.

Appendix: Linear Combinations and the Effects of Removing Integer Offset

Identities. The *round* operator just returns the nearest integer. So the following holds if N is an integer.

$$\text{round}(x + N) = \text{round}(x) + N \quad (\text{A.1})$$

For the following, all subscripted N s are integers. If $-0.5 < y < 0.5$, k is a rational number such that $k = \frac{N_n}{N_d}$ and x is any real then

$$f_k(x, y) = y + k \text{round}(x) - \text{round}(y + k \text{round}(x)) \quad (\text{A.2})$$

$$= y + kN_x - \text{round}(y + kN_x) \quad (\text{A.3})$$

$$= y + \frac{N_n N_x}{N_d} - \text{round}\left(y + \frac{N_n N_x}{N_d}\right) \quad (\text{A.4})$$

$$= y + \frac{N_j N_d + N_r}{N_d} - \text{round}\left(y + \frac{N_j N_d + N_r}{N_d}\right), \quad N_r = 0, 1, \dots, N_d - 1 \quad (\text{A.5})$$

$$= y + N_j + \frac{N_r}{N_d} - \text{round}\left(y + N_j + \frac{N_r}{N_d}\right) \quad (\text{A.6})$$

$$= y + \frac{N_r}{N_d} - \text{round}\left(y + \frac{N_r}{N_d}\right) \quad (\text{A.7})$$

$$= \begin{cases} y + \frac{N_r}{N_d} & y + \frac{N_r}{N_d} < 0.5 \\ y + \frac{N_r}{N_d} - 1 & 0.5 < y + \frac{N_r}{N_d} < 1.5 \end{cases} \quad (\text{A.8})$$

Which is to say, y will be offset by a multiple of $\frac{1}{N_d}$, either positive or negative, within the range ± 1 .

Effects. Ignoring multipath errors and receiver noise and using $\lambda_1 = \frac{77}{60} \lambda_2$, the phase equations are

$$\phi_1 = \frac{\Delta\rho + \Delta T}{\lambda_1} - \frac{\Delta I}{c f_1} + \Delta N_1 \quad (\text{A.9})$$

$$\phi_2 = \frac{\Delta\rho + \Delta T}{\lambda_2} - \frac{77\Delta I}{60c f_1} + \Delta N_2 \quad (\text{A.10})$$

Measurements are given by

$$\hat{\phi}_1 = \phi_1 - \text{round}(\phi_1) \quad (\text{A.11})$$

$$= \frac{\Delta\rho + \Delta T}{\lambda_1} - \frac{\Delta I}{cf_1} - \text{round}\left(\frac{\Delta\rho + \Delta T}{\lambda_1} - \frac{\Delta I}{cf_1}\right) \quad (\text{A.12})$$

$$= \frac{\Delta\rho + \Delta T}{\lambda_1} - \frac{\Delta I}{cf_1} - \hat{N}_1 \quad (\text{A.13})$$

$$\hat{\phi}_2 = \frac{\Delta\rho + \Delta T}{\lambda_2} - \frac{77\Delta I}{60cf_1} - \hat{N}_2 \quad (\text{A.14})$$

Linear combinations are formed in order to increase wavelength, reduce residual atmospheric error, and/or reduce noise [68]. A linear combination is obtained as

$$\phi_{jk} = j\hat{\phi}_1 + k\hat{\phi}_2 \quad (\text{A.15})$$

$$= \frac{\Delta\rho}{\lambda_{jk}} + \frac{\Delta T}{\lambda_1} \frac{77j + 60k}{77} - \frac{\Delta I}{cf_1} \frac{60j + 77k}{60} - j\hat{N}_1 - k\hat{N}_2 \quad (\text{A.16})$$

$$= \frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1} - j\hat{N}_1 - k\hat{N}_2 \quad (\text{A.17})$$

$$\lambda_{jk} = \frac{77}{77j + 60k} \lambda_1 \quad (\text{A.18})$$

and the linear combination measurement is

$$\hat{\phi}_{jk} = \phi_{jk} - \text{round}(\phi_{jk}) \quad (\text{A.19})$$

If j and k are integers then

$$\hat{\phi}_{jk} = \frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1} - \text{round}\left(\frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1}\right) \quad (\text{A.20})$$

The final term must be zero for each SV for convergence to occur. If j is an integer and k is a rational number then

$$\hat{\phi}_{jk} = \frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1} - k\hat{N}_2 - \text{round}\left(\frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1} - k\hat{N}_2\right) \quad (\text{A.21})$$

$$= \frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1} - \frac{N_n \hat{N}_2}{N_d} - \text{round}\left(\frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1} - \frac{N_n \hat{N}_2}{N_d}\right) \quad (\text{A.22})$$

$$= \frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1} + \frac{\hat{N}_r}{N_d} - \text{round}\left(\frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{cf_1} + \frac{\hat{N}_r}{N_d}\right) \quad (\text{A.23})$$

Using the same convergence criterion as for the integer-only case, (A.8) gives

$$\hat{\phi}_{jk} = \begin{cases} y + \frac{\hat{N}_r}{N_d} & y + \frac{\hat{N}_r}{N_d} < 0.5 \\ y + \frac{\hat{N}_r}{N_d} - 1 & 0.5 < y + \frac{\hat{N}_r}{N_d} < 1.5 \end{cases} \quad (\text{A.24})$$

$$y = \frac{\Delta\rho}{\lambda_{jk}} + m_T \frac{\Delta T}{\lambda_1} - m_I \frac{\Delta I}{c f_1} \quad (\text{A.25})$$

In other words, each measurement will be offset by a multiple of $\frac{1}{N_d}$. The multiple will be different for each SV depending on the value of \hat{N}_r . This spacing of levels will require another search for each SV to find the proper numerator to use. The distance between each level in meters is $\frac{\lambda_{jk}}{N_d}$. This is the same as the wavelength obtained if j and k were each multiplied by N_d and hence are integers. For example, using $j = 1$, $k = -\frac{3}{4}$ gives a wavelength of 0.458 m and level spacing of 0.114 m. Using $j = 4$, $k = -3$ gives a wavelength of 0.114 m and no levels within a cycle. In conclusion, it is better to using integer multipliers for the linear combinations to remove the step of searching for the correct level for each SV.

Linear Combinations. A noise analysis wasn't performed in the previous section. However, it is easy to show that if each individual phase measurement has a variance of σ_1 cycles, then the linear combination noise variance would be $\sigma_{jk} = \sigma_1 \sqrt{j^2 + k^2}$. Table A.1 shows values of various factors for different integer linear combinations.

The ionospheric-free combination completely removes the ionospheric error. However, the wavelength is extremely small with large multipliers on the tropospheric error and noise. The ionospheric-reduced combination almost removes all the ionospheric error with a more moderate increase to tropospheric error and noise. A good approach, under high differential ionospheric errors, would be to use the widelane combination to find initial candidates and then find ionospheric reduced candidates near those initial candidates.

Table A.1: Parameter Values for Various Linear Combinations

Name	j, k	λ_{jk} (m)	m_T	m_I	m_n
Widelane	1, -1	0.862	0.22	-0.28	1.41
Wider Lane	-3, 4	1.628	0.12	2.13	5.00
Iono Free	77, -60	0.006	30.25	0.00	97.6
Iono Reduced	4, -3	0.114	1.66	0.15	5.00
L ₁	1, 0	0.190	1.00	1.00	1.00
L ₂	0, 1	0.244	0.78	1.28	1.00

Bibliography

- [1] Abbott, A. and W. Lillo. “Global Positioning Systems and Inertial Measuring Unit Ultratight Coupling Method”. U.S. Patent 6516021, 2003.
- [2] Axelrad, P., J. Donna, and M. Mitchell. “Enhancing GNSS Acquisition by Combining Signals from Multiple Channels and Satellites”. *Proceedings of the ION-ITM-10*, 2617–2628. Institute of Navigation, Savannah, GA, Sep 2009.
- [3] Bradley, B., P. Axelrad, J. Donna, and S. Mohiuddin. “Performance Analysis of Collective Detection of Weak GPS Signals”. *Proceedings of the ION-ITM-10*, 3041–3053. Institute of Navigation, Portland, OR, Sep 2010.
- [4] Brown, R. and P. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*, volume 3. John Wiley and Sons, New York, NY, 1997. ISBN 0-471-12838-2.
- [5] Cellmer, S. “Proposal for New Strategy in Precise Positioning”. *Reports on Geodesy*, 87(2):77–85, 2009.
- [6] Cellmer, S. “The Real-Time Precise Positioning Using MAFA Method”. *Proceedings of 8th International Conference on Environmental Engineering*, 1310–1314. Vilnius Gediminas Technical University, Vilnius, Lithuania, May 2011.
- [7] Cellmer, S. “A Graphic Representation of the Necessary Condition for the MAFA Method”. *IEEE Transaction on Geoscience and Remote Sensing*, 50(2):482–488, 2012.
- [8] Cellmer, S. “Search Procedure for Improving Modified Ambiguity Function Approach”. *Survey Review*, 45(332):380–385, 2013.
- [9] Cellmer, S., P. Wielgosz, and Z. Rzepecka. “Modified Ambiguity Function Approach for GPS Carrier Phase Positioning”. *Journal of Geodesy*, 84(4):267–275, 2010.
- [10] Chan, B. *DGPS and UWB Aided Vector-Based GNSS Receiver for Weak Signal Environments*. Master’s thesis, University of Calgary, 2013.
- [11] Chan, B. and M. Petovello. “Collaborative Vector Tracking of GNSS Signals with Ultra-Wideband Augmentation in Degraded Signal Environments”. *Proceedings of ION ITM-11*. Institute of Navigation, San Diego, CA, Jan 2011.
- [12] Closas, P., C. Fernández-Prades, and J. Fernández-Rubio. “Maximum Likelihood Estimation of Position in GNSS”. *IEEE Signal Processing Letters*, 14(5):359–362, 2007.

- [13] Closas, P., C. Fernández-Prades, and J. Fernández-Rubio. “Cramer-Rao Bound Analysis of Positioning Approaches in GNSS Receivers”. *IEEE Transactions on Signal Processing*, 57(10):3775–3786, 2009.
- [14] Counselman, C. III and R. Abbot. “Method of Resolving Radio Phase Ambiguity in Satellite Orbit Determination”. *Journal of Geophysical Research*, 94(B6):7058–7064, 1989.
- [15] Counselman, C. III and S. Gourevitch. “Miniature Interferometer Terminals for Earth Surveying: Ambiguity and Multipath with Global Positioning System”. *IEEE Transactions on Geoscience and Remote Sensing*, GE-19(4):244–252, 1981.
- [16] DiEsposti, R. “GPS PRN Code Signal Processing and Receiver Design for Simultaneous All-in-View Coherent Signal Acquisition and Navigation Solution Determination”. *Proceedings of ION NTM-07*. Institute of Navigation, San Diego, CA, Jan 2007.
- [17] Edwards, L., B. Clark, and D. Bevly. “Implementation Details of a Deeply Integrated GPS/INS Software Receiver”. *Proceedings of PLANS-10*, 1137–1146. IEEE/ION, Palm Springs, CA, May 2010.
- [18] Frei, E. and G. Beutler. “Rapid Static Positioning Based on the Fast Ambiguity Resolution Approach (FARA): Theory and First Results”. *Manuscript Geodaetica*, 15:325–356, 1990.
- [19] Giger, K. and C. Günther. “Position Domain Joint Tracking”. *Proceedings of the 5th European Workshop on GNSS Signals and Signal Processing*. IEEE, Dec 2010.
- [20] Giger, K. and C. Günther. “Multiantenna Multisatellite Code and Carrier Tracking”. *Proceedings of European Navigation Conference 2011*. UK Royal Institute of Navigation, London, England, Nov 2011.
- [21] Giger, K. and C. Günther. “Multisatellite Tracking GNSS Receivers in Multipath”. *Proceedings of ION GNSS 2011*, 1140–1151. Institute of Navigation, Portland, OR, Sep 2011.
- [22] Giger, K. and P. Henkel. “Joint Satellite Code and Carrier Tracking”. *Proceedings of the ION 2010 National Technical Meeting*, 636–645. Institute of Navigation, San Diego, CA, Jan 2010.
- [23] Giger, K., P. Henkel, and C. Günther. “Multi-Frequency, Multi-Satellite Carrier Tracking”. *Proceedings of the 4th European Workshop on GNSS Signals and Signal Processing*. Oberpfaffenhofen, Germany, 2009.
- [24] Groves, P., C. Mather, and A Macaulay. “Demonstration of Non-coherent Deep INS/GPS Integration for Optimised Signal-to-noise Performance”. *Proceedings of ION ITM-07*, 2627–2638. Institute of Navigation, Fort Worth, TX, Sep 2007.

- [25] Gunawardena, S. *Development of a Transform-Domain Instrumentation Global Positioning System Receiver for Signal Quality and Anomalous Event Monitoring*. Ph.D. thesis, Ohio University, 2007.
- [26] Gustafson, D. and J. Dowdle. “Deeply Integrated Code Tracking: Comparative Performance Analysis”. *Proceedings of ION GNSS-03*, 2553–2561. IEEE, Portland, OR, Sep 2003.
- [27] Gustafson, D., J. Dowdle, and J. Elwell. “Deeply-Integrated Adaptive INS/GPS Navigator with Extended-Range Code Tracking”. U.S Patent 6630904, 2001.
- [28] Gustafson, D., J. Dowdle, and K. Flueckiger. “A Deeply Integrated Adaptive GPS-Based Navigator with Extended Range Code Tracking”. *Proceedings of PLANS-00*, 118–124. IEEE, San Diego, CA, Mar 2000.
- [29] Han, S. and C. Rizos. “Improving the Computational Efficiency of the Ambiguity Function Method”. *Journal of Geodesy*, 70:330–341, 1996.
- [30] Hatch, R. “Instantaneous Ambiguity Resolution”. *Proceedings of the International Symposium 107 on Kinematic Systems in Geodesy, Surveying and Remote Sensing*, 299–308. Springer Verlag, New York, Sep 1991.
- [31] Henkel, P., G. Gao, T. Walter, and C. Günther. “Robust Multi-Carrier, Multi-Satellite Vector Phase Locked Loop with Wideband Ionospheric Correction and Integrated Weighted RAIM”. *Proceedings of the European Navigation Conference*. Naples, Italy, 2009.
- [32] Henkel, P., K. Giger, and C. Günther. “Multi-Frequency, Multi-Satellite Vector Phase-Locked Loop for Robust Carrier Tracking”. *IEEE Journal of Selected Topics in Signal Processing*, 3(4):674–681, 2009.
- [33] Hui, L. and Y. JingShu. “Analysis and Simulation of Vector Tracking Algorithms for Weak GPS Signals”. *Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation, and Robotics*, 215–218. IEEE, 2010.
- [34] Hui, L. and Y. JingShu. “Frequency Tracking for High Dynamic GPS Signals Based on Vector Frequency Lock Loop”. *Proceedings of the WASE International Conference on Information Engineering*, 98–101. IEEE, 2010.
- [35] Kaplan, E. and C. Hegarty. *Understanding GPS: Principles and Applications*. Artech House, Inc., Norwood, MA, 2nd edition, 2006. ISBN 978-1-58053-894-7.
- [36] Kiesel, S., C. Ascher, D. Gramm, and G. Trommer. “GNSS Receiver with Vector Based FLL-Assisted PLL Carrier Tracking Loop”. *Proceedings of ION-ITM-08*, 197–203. Institute of Navigation, Savannah, GA, Sep 2008.

- [37] Kim, U., D. De Lorenzo, D. Akos, P. Enge, and J. Orr. “Precise Phase Calibration of a Controlled Reception Pattern GPS Antenna for JPALS”. *Proceedings of PLANS-04*, 478–485. Institute of Navigation, Monterey, CA, Apr 2004.
- [38] Langer, M., S. Kiesel, C. Ascher, and G. Trommer. “Deeply Coupled GPS/INS Integration in Pedestrian Navigation Systems in Weak Signal Conditions”. *2012 International Conference on Indoor Positioning and Indoor Navigation*, 1–7. IEEE, Sydney, Australia, Nov 2012.
- [39] Lashley, M., D. Bevly, and J. Hung. “Impact of Carrier to Noise Power Density, Platform Dynamics, and IMU Quality on Deeply Integrated Navigation”. *Proceedings of PLANS-08*, 9–16. IEEE/ION, Monterey, CA, May 2008.
- [40] Lashley, M., D. Bevly, and J. Hung. “Performance Analysis of Vector Tracking Algorithms for Weak GPS Signals”. *IEEE Journal of Selected Topics in Signal Processing*, 3(4):661–673, 2009.
- [41] Lashley, M., D. Bevly, and J. Hung. “Analysis of Deeply Integrated and Tightly Coupled Architectures”. *Proceedings of PLANS-10*, 382–396. IEEE/ION, Palm Springs, CA, May 2010.
- [42] Lashley, M., D. Bevly, and J. Hung. “A valid comparison of Vector and Scalar Tracking Loops”. *Proceedings of PLANS-10*, 464–474. IEEE/ION, Palm Springs, CA, May 2010.
- [43] Lin, T., J. Curran, C. O’Driscoll, and G. Lachapelle. “Implementation of a Navigation Domain GNSS Signal Tracking Loop”. *Proceedings of ION GNSS 2011*, 3644–3651. Institute of Navigation, Portland, OR, Sep 2011.
- [44] Liu, J., X. Cui, Q. Chen, and M. Lu. “Joint Vector Tracking Loop in a GNSS Receiver”. *Proceedings of ION ITM-11*, 1025–1032. Institute of Navigation, San Diego, CA, Jan 2011.
- [45] Liu, J., X. Cui, M. Lu, and Z. Feng. “Vector Tracking Loops in GNSS Receivers for Dynamic Weak Signals”. *Journal of Systems Engineering and Electronics*, 24(3):349–364, 2013.
- [46] Liu, J., H. Yin, X. Cui, M. Lu, and Z. Feng. “A Direct Position Tracking Loop”. *Proceedings of GNSS 2011*, 3634–3643. Institute of Navigation, Portland, OR, Sep 2011.
- [47] Loan, C. Van. “Computing Integrals Involving the Matrix Exponential”. *IEEE Transactions on Automatic Control*, 23(3):395–404, 1978.
- [48] Maybeck, P. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, Inc, San Diego, CA, 1979. ISBN 0-12-480701-1.

- [49] Misra, P. and P. Enge. *Global Positioning System Signals, Measurements, and Performance*. Ganga-Jamuna Press, Lincoln, MA, 2001. ISBN 0-9709544-0-9.
- [50] Niedermeier, H., B. Eissfeller, J. Winkel, T. Pany, B. Riedl, T. Wörz, R. Schweikert, S. Lagrasta, G. Lopéz-Risue no, and D. Jiminez-Banos. “DINGPOS: High Sensitivity GNSS Platform for Deep Indoor Scenarios”. *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Zürich, Switzerland, Sep 2010.
- [51] Nunes, F., J. Marçal, and F. Sousa. “Low-Complexity VDLL Receiver for Multi-GNSS Constellations”. *Proceedings of the 5th European Workshop on GNSS Signals and Signal Processing*. IEEE, Dec 2010.
- [52] Pany, T. and B. Eissfeller. “Use of Vector Delay Lock Loop Receiver for GNSS Signal Power Analysis in Bad Signal Conditions”. *PLANS-06*, 893–903. IEEE/ION, San Diego, CA, Apr 2006.
- [53] Pany, T., R. Kanuith, and B. Eissfeller. “Deep Integration of Navigation Solution and Signal Processing”. *Proceedings of ION ITM-05*, 1095–1102. Institute of Navigation, Long Beach, CA, Sep 2005.
- [54] Pany, T., J. Winkel, B. Riedl, M. Restle, T. Wörz, R. Schweikert, H. Niedermeier, G. Ameres, B. Eissfeller, S. Lagrasta, and G. Lopéz-Risue no. “Performance of a Partially Coherent Ultra-Tightly Coupled GNSS/INS Pedestrian Navigation System Enabling Coherent Integration Times of Several Seconds to Track GNSS Signals Down to 1.5 dB-Hz”. *Proceedings of the ION-ITM-10*, 919–934. Institute of Navigation, Savannah, GA, Sep 2009.
- [55] Parkinson, B. and J. Spilker. *Global Positioning System: Theory and Applications*, volume 1. AIAA, Washington, DC, 1996. ISBN 1-56347-106-X.
- [56] Peng, S., J. Morton, and R. Di. “A Multiple-Frequency GPS Software Receiver Design Based on a Vector Tracking Loop”. *Proceedings of PLANS-10*, 495–505. IEEE/ION, Myrtle Beach, SC, Apr 2012.
- [57] Petovello, M. and G. Lachapelle. “Comparison of Vector-Based Software Receiver Implementations with Application to Ultra-Tight GPS/INS Integration”. *Proceedings of ION GNSS 06*. Institute of Navigation, Ft. Worth, TX, Sep 2006.
- [58] Petovello, M., C. O’Driscoll, and G. Lachapelle. “Ultra-Tight GPS/INS for Carrier Phase Positioning in Weak-Signal Environments”. *Proceedings of NATO RTO SET-104 Symposium on Military Capabilities Enabled by Advances in Navigation Sensors*. Antalya, Turkey, 2007.
- [59] Petovello, M., C. O’Driscoll, and G. Lachapelle. “Weak Signal Carrier Tracking Using Extended Coherent Integration with an Ultra-Tight GNSS/IMU Receiver”. *Proceedings of the European Navigation Conference*. Toulouse, France, Apr 2008.

- [60] Raquet, J. *Development of a Method for Kinematic GPS Carrier-Phase Ambiguity Resolution Using Multiple Reference Receivers*. Ph.D. thesis, University of Calgary, 1998.
- [61] Remondi, B. “Psuedo-Kinematic GPS Results Using the Ambiguity Function Method”. *NOAA Technical Memorandum*, NOS-NGS(52):1–23, 1990.
- [62] Sennott, J. and D. Senffner. “The Use of Satellite Constellation Geometry and A-Priori Motion Constraints for Prevention of Cycle Slips in a GPS Signal Processor”. *Proceedings of ION GPS-91*, 77–85. Institute of Navigation, Nashville, TN, Sep 1991.
- [63] Soloviev, A. and J. Dickman. “Deeply Integrated GPS for Indoor Navigation”. *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Zürich, Switzerland, Sep 2010.
- [64] Soloviev, A. and J. Dickman. “Extending GPS Carrier Phase Availability Indoors with a Deeply Integrated Receiver Architecture”. *IEEE Wireless Communications*, 18(1):36–44, 2011.
- [65] Soloviev, A., F. van Graas, and S. Gunawardena. “Implementation of Deeply Integrated GPS/Low-Cost IMU for Reacquisition and Tracking of Low CNR GPS Signal”. *Proceedings of ION NTM-04*, 923–935. Institute of Navigation, San Diego, CA, Jan 2004.
- [66] Soloviev, A., F. van Graas, and S. Gunawardena. “Deeply Integrated GPS/Low-Cost IMU for Low CNR Signal Processing: Concept Description and In-Flight Demonstration”. *Navigation: The Journal of the Institute of Navigation*, 55(1):1–13, 2008.
- [67] Teunissen, P. “A New Method for Fast Carrier Phase Ambiguity Estimation”. *Proceedings of IEEE PLANS 1994*, 562–573. IEEE, Las Vegas, NV, Apr 1991.
- [68] Urquhart, L. *An Analysis of Multi-Frequency Carrier Phase Linear Combinations for GNSS*. Technical Report 263, University of New Brunswick, 2009.
- [69] Verhagen, S. and P. Teunissen. “The Ratio Test for Future GNSS Ambiguity Resolution”. *GPS Solutions*, 17:535–548, 2013.
- [70] Wang, J., M. Stewart, and M. Tsakiri. “Ambiguity Resolution on-the-Fly”. *Journal of Geodesy*, 72(11):644–653, 1998.
- [71] Wang, J., X. Yan, B. Cai, and W. Shangguan. “Research on Deeply Integrated GPS/INS for Autonomous Train Positioning”. *Proceedings of 4th International Symposium on Microwave, Antenna, Propagation, and EMC Technologies for Wireless Communications*, 687–690. IEEE, Nov 2011.

- [72] Weill, L. “A High Performance Code and Carrier Tracking Architecture for Ground-Based Mobile GNSS Receivers”. *Proceedings of the ION-ITM-10*, 3054–3068. Institute of Navigation, Portland, OR, Sep 2010.
- [73] Won, J., D. Dötterböck, and B. Eissfeller. “Performance Comparison of Different Forms of Kalman Filter Approach for a Vector-Based GNSS Signal Tracking Loop”. *Proceedings of ION-ITM-09*, 3037–3048. Institute of Navigation, Savannah, GA, Sep 2009.
- [74] Zhao, S. and D. Akos. “An Open Source GPS/GNSS Vector Tracking Loop - Implementation, Filter Tuning, and Results”. *Proceedings of ION ITM-11*, 1293–1305. Institute of Navigation, San Diego, CA, Jan 2011.
- [75] Zhenzhen, Z., C. Zhu, T. Guangfu, L. Shufeng, and H. Fukan. “EKF Based Vector Delay Lock Loop Algorithm for GPS Signal Tracking”. *Proceedings of the International Conference on Computer Design and Applications*, 352–356. IEEE, 2010.
- [76] Zhodzishsky, M., S. Yudanov, V. Veitel, and J. Ashjaee. “Co-op Tracking for Carrier Phase”. *Proceedings of ION GPS-98*. Institute of Navigation, Nashville, TN, Sep 1998.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) 19-06-2014		2. REPORT TYPE Doctoral Dissertation		3. DATES COVERED (From — To) Oct 2011–Jun 2014			
4. TITLE AND SUBTITLE The Differential Vector Phase-Locked Loop for Global Navigation Satellite System Signal Tracking			5a. CONTRACT NUMBER				
			5b. GRANT NUMBER				
			5c. PROGRAM ELEMENT NUMBER				
			6. AUTHOR(S) Brewer, James J., GS-14, DAF			5d. PROJECT NUMBER	
						5e. TASK NUMBER	
						5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENG-DS-14-J-2			
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally Left Blank				10. SPONSOR/MONITOR'S ACRONYM(S)			
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)			
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED							
13. SUPPLEMENTARY NOTES This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.							
14. ABSTRACT A novel differential vector phase-locked loop (DVPLL) is derived that takes GNSS code-phase and carrier-phase measurements from a base station and uses them to maintain an integer ambiguity resolved quality solution directly in the vector tracking loop of a rover receiver. The only state variables estimated and used to create the replica code and carrier signals from the base station measurements are three position and two clock states for a static test. Closing the individual loops solely through the navigation filter makes this a pure vector method. For short baselines, where differential atmospheric errors are small, the DVPLL can be used on single-frequency data. An L1-only live-sky static test was performed using the method resulting in a 3D accuracy of 5.3 mm for an 18.5 m baseline. An acquisition algorithm is also developed to initialize the DVPLL. The algorithm performs a search in the space-time domain vice the measurement domain. An upper bound on the failure rate of the algorithm can be set by the user. The algorithm was tested on 24-h dual- and single-frequency CORS data sets with close to a 100% success rate and on a 15-min data set of single-frequency IF samples with a 100% success rate.							
15. SUBJECT TERMS vector phase locked loop (VPLL), differential, GPS, GNSS, vector tracking							
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON		
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. John F. Raquet (ENG)		
U	U	U	UU	125	19b. TELEPHONE NUMBER (include area code) (937) 255-3636 x4580 John.Raquet@afit.edu		