

KALMAN FILTER BASED TRACKING ALGORITHMS FOR SOFTWARE GPS RECEIVERS

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Matthew Lashley

Certificate of Approval:

David M. Bevly, Co-Chair
Assistant Professor
Mechanical Engineering

John Y. Hung, Co-Chair
Professor
Electrical and Computer Engineering

Thomas S. Denney
Professor
Electrical and Computer Engineering

Joe F. Pittman
Interim Dean
Graduate School

KALMAN FILTER BASED TRACKING ALGORITHMS FOR SOFTWARE GPS RECEIVERS

Matthew Lashley

A Thesis
Submitted to
the Graduate Faculty of
Auburn University
in Partial Fulfillment of the
Requirements for the
Degree of
Master of Science

Auburn, Alabama
December 15, 2006

KALMAN FILTER BASED TRACKING ALGORITHMS FOR SOFTWARE GPS RECEIVERS

Matthew Lashley

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Matthew Vernon Lashley was born in Roanoke, Alabama on November 12, 1981. He is the second child of Vernon and Vicky Lashley, and has an older brother, William. Matthew grew up in the small community of Malone, near the town of Wadley, Alabama. He attended Wadley High School and graduated in 2000. After graduating from high school, Matthew attended Southern Union State Community College in Wadley. He transferred to Auburn University in the summer of 2002 and initially was pursuing a physics undergraduate degree. In the Fall of 2002, he transferred to the electrical engineering program. Matthew earned his bachelor's degree in electrical engineering in May 2004. He then worked for Phase IV Systems in Huntsville, Alabama before entering graduate school at Auburn University in the Fall of 2004. Matthew started graduate school in the electrical engineering department studying control systems under Dr. John Hung. He worked as a graduate teaching assistant for two semesters at Auburn before returning to work Phase IV systems in the summer of 2005. In the Fall of 2005 Matthew accepted a position in the GPS and Vehicle Dynamics Lab (GAVLAB) as a graduate research assistant, working for Dr. David Bevly. Matthew worked two semesters at the GAVLAB researching Deeply Integrated (DI) GPS algorithms. In the summer of 2006, Matthew worked for the company NTA in Huntsville, Alabama. There, he helped in the testing for the Deeply Integrated GPS Navigation Unit (DIGNU) and continued his research into DI GPS algorithms. In the Fall of 2006 Matthew returned to the GAVLAB and finished his thesis. He plans on continuing his research at Auburn University and pursuing his doctorate.

THESIS ABSTRACT

KALMAN FILTER BASED TRACKING ALGORITHMS FOR SOFTWARE GPS RECEIVERS

Matthew Lashley

Master of Science, December 15, 2006
(B.E.E., Auburn University, 2004)

115 Typed Pages

Directed by John Hung and David Bevly

In this thesis several new Kalman filter based tracking algorithms for GPS software receivers are presented. Traditional receivers use Costas loops and Delay Lock Loops (DLL) to track the carrier and Pseudo-Random Noise (PRN) signals broadcast by the GPS satellites, respectively. The tasks of tracking the the carrier and PRN signals are done separately. The Kalman filter based algorithms introduced in this thesis provide an alternative to the Costas loop and DLL. The task of tracking the PRN sequences is handled by a single Extended Kalman Filter (EKF). The EKF is used to estimate the user's position in the Earth-Centered Earth-Fixed (ECEF) coordinate frame. Using the EKF's estimates, the code phases of the PRN sequences being received from the different satellites are predicted. Estimates of the code phase error between the predicted and received codes are generated using discriminator functions. The estimates of the code phase errors are used to update the EKF's estimates of the user's navigation states. To provide proof of concept, data was collected using a Spirent GPS simulator. The recorded data was used to show that the new Kalman filter based algorithms outperform traditional tracking methods.

ACKNOWLEDGMENTS

I would like to thank my family and friends for their support during my time in graduate school. I would not have been able to complete my Master's degree without their help. I would also like to thank Dr. John Hung for advising me during my graduate studies. Dr. Hung has always been willing to take time to share his wisdom and sagacity with me.

Gratitude is also in order for Dr. David Bevly and the members of the GAVLAB. Dr. Bevly has supported me for the majority of my time in graduate school. He has also always been willing to make time to discuss matters with me. The other members of the GAVLAB have always been willing to help and share their knowledge with me.

I would like to acknowledge the U.S. Army Aviation and Missile Research, Development, and Engineering Center (AMRDEC) for their financial patronage and technical support. Specifically, I would like to thank Mr. Brian Baeder at AMRDEC. Additionally, I owe NTA a debt of gratitude for their technical and financial support. Namely, I would like to thank Mr. Jeff Rhea and Dr. Shannon Fields at NTA for their help. I also would like to acknowledge Phase IV Systems for their contributions.

Style manual or journal used Journal of Approximation Theory (together with the style known as “aums”). Bibliography follows van Leunen’s *A Handbook for Scholars*.

Computer software used The document preparation package T_EX (specifically L^AT_EX) together with the departmental style-file aums.sty.

TABLE OF CONTENTS

LIST OF FIGURES	x
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Background and Literature Survey	2
1.2.1 Loose Coupling	4
1.2.2 Tight Coupling	4
1.2.3 Deeply Integrated or Ultra-tight Coupling	5
1.2.4 Raytheon Method	7
1.2.5 Anthony Abbott Method	8
1.2.6 Draper Method	10
1.3 Contributions and Outline	12
2 GPS SIGNAL STRUCTURE	14
2.1 Overview of GPS Signals	14
2.1.1 GPS Carrier Signal	15
2.1.2 Gold Code	16
2.1.3 Data Message	20
2.2 Conclusion	20
3 TRADITIONAL SOFTWARE GPS RECEIVER	23
3.1 Overview	23
3.2 Receiver Front-end	25
3.3 Acquisition	26
3.4 Tracking Loops	29
3.4.1 The Phase-Locked Loop	29
3.4.2 Costas Loop	38
3.4.3 Delay Lock Loop	41
3.4.4 Combined Costas and Delay Lock Loops	44
3.5 Position Determination	48
3.5.1 Least Squares Solution	49
3.5.2 EKF Solution	51
3.6 Conclusion	56
4 DEVELOPMENT OF KALMAN FILTER BASED TRACKING ALGORITHMS	57
4.1 Introduction	57
4.2 Tracking Loops as Kalman Filters	57
4.3 Vector Delay Lock Loop	63

4.4	Conclusion	69
5	VALIDATION AND PERFORMANCE OF KALMAN FILTER BASED ALGORITHMS	71
5.1	Simulation Setup	71
5.2	Code and Carrier Tracking Results	74
5.3	Positioning Results	87
5.4	Clock Results	91
6	CONCLUSION	93
6.1	Concluding Remarks	93
6.2	Future Work	94
	BIBLIOGRAPHY	95
	APPENDIX - KALMAN FILTER	99

LIST OF FIGURES

1.1	Loosely Coupled Architecture [Hamm, 2005]	5
1.2	Tightly Coupled Architecture [Hamm, 2005]	6
1.3	Ultra-Tightly Coupled Architecture [Hamm, 2005]	7
1.4	Raytheon Method [Horslund and Hooker, 1999]	8
1.5	Anthony Abbott Method [Abbott and Lillo, 2003]	9
1.6	Draper Method [Gustafson et al., 2001]	11
2.1	Autocorrelation Function of Gold Codes	18
2.2	Ideal Autocorrelation Function of Gold Codes	19
2.3	Cross-Correlation of Different Gold Codes	19
2.4	Arrangement of the GPS Navigation Message [Hamm, 2005]	21
3.1	NordNav Receiver Front-end [Normack et al., 2002]	24
3.2	NordNav GUI [Normack et al., 2002]	24
3.3	General GPS Receiver Block Diagram	25
3.4	Acquisition	28
3.5	Basic Phase Locked Loop	30
3.6	Analog Phase Locked Loop	31
3.7	Phase Locked Loop Acquiring A Signal Inside The Lock-In Range	32
3.8	Phase Locked Loop Acquiring A Signal Inside The Pull-In Range	33
3.9	Linearized Phase Locked Loop	34
3.10	Costas Loop	39

3.11	Digital Costas Loop	40
3.12	Delay Lock Loop	42
3.13	Autocorrelation Function of Gold Code	43
3.14	Complete Loop Architecture	46
4.1	Output of Code Discriminator	61
4.2	Output of Arctangent Phase Discriminator	62
4.3	Output of Frequency Discriminator	62
4.4	Vector Delay Lock Loop Flow Chart	64
4.5	General Vector Delay Lock Loop Architecture	65
4.6	Arrival Time of Satellite Data Bits	66
4.7	Determination of Transmission Index	67
5.1	In-Phase Prompt Output for SV 16 by VDLL Algorithm	75
5.2	In-Phase Prompt Output for SV 16 by VDLL Algorithm	75
5.3	Carrier Tracking for SV 16	77
5.4	PRN Code Phase Error Measurements for SV 16 by the Traditional Delay Lock Loop (top) and the Vector Delay Lock Loop (bottom)	77
5.5	In-Phase Prompt Output for SV 16 by Costas Loop	78
5.6	In-Phase Prompt Output for SV 5 by VDLL Algorithm	79
5.7	In-Phase Prompt Output for SV 5 by VDLL Algorithm	79
5.8	Carrier Tracking for SV 5	80
5.9	PRN Code Phase Error Measurements for SV 5 by the Traditional Delay Lock Loop (top) and the Vector Delay Lock Loop (bottom)	81
5.10	In-Phase Prompt Output for SV 5 by Costas Loop	81
5.11	In-Phase Prompt Output for SV 3 by VDLL Algorithm	82

5.12	Carrier Tracking for SV 3	83
5.13	PRN Code Phase Error Measurements for SV 3 by the Traditional Delay Lock Loop (top) and the Vector Delay Lock Loop (bottom)	83
5.14	In-Phase Prompt Output for SV 12 by VDLL Algorithm	84
5.15	In-Phase Prompt Output for SV 12 by VDLL Algorithm	85
5.16	Carrier Tracking for SV 12	85
5.17	PRN Code Phase Error Measurements for SV 12 by the Traditional Delay Lock Loop (top) and the Vector Delay Lock Loop (bottom)	86
5.18	In-Phase Prompt Output for SV 12 by Costas Loop	86
5.19	Error in X-ECEF Coordinate Estimate	88
5.20	Error in Y-ECEF Coordinate Estimate	88
5.21	Error in Z-ECEF Coordinate Estimate	89
5.22	User Longitude Based on EKF Estimates	90
5.23	User Latitude Based on EKF Estimates	90
5.24	User Altitude Based on EKF Estimates	91
5.25	EKF Estimate of Clock Bias	92
5.26	EKF Estimate of Clock Drift	92
6.1	Kalman Filter Algorithm [Welch and Bishop, 2006]	101
6.2	Extended Kalman Filter Algorithm [Welch and Bishop, 2006]	103

CHAPTER 1

INTRODUCTION

1.1 Motivation

Navigation is fundamental task of living creatures. Animals must travel long distances and return to their origin in order to survive. Many species of birds migrate hundreds of miles annually, driven by instinct and using only their senses. Salmon traverse long distances in the open ocean to return to the stream of their origin to spawn. Human beings must likewise navigate to survive, yet they cannot rely on their instincts, but on their ingenuity. People have been applying their rational minds to the task of navigating for at least six thousand years. Ancient mariners learned to use measurements of the stars and constellations to navigate the seas. In modern times, humans have replaced the natural stars with an artificial constellation of satellites for navigation. People can now determine their position anywhere on the Earth using radio ranging to the constellation of artificial satellites, called NAVSTARs [Parkinson, 1996].

The NAVSTAR Global Positioning System (GPS) was developed in the 1970's by the Department of Defense (DoD). The purpose of the GPS system is to provide the U.S. military accurate measurements of position, velocity, and time (PVT). The DoD desired position estimates with a root-mean-square (rms) error of ten meters, velocity estimates with an rms error of .1 m/s, and time estimates with an rms error of 100 ns [Misra and Enge, 2001]. The system was also equipped with a less accurate, civilian component that non-military user's can use. Today the military uses the GPS system extensively to navigate and guide munitions. Civilian usage of the NAVSTAR GPS

system has become ubiquitous. Many automobiles come equipped with GPS receivers and hand held receivers have become commonplace among joggers and hikers.

This research was performed under the patronage of the U.S. Army Aviation and Missile Research, Development, and Engineering Center (AMRDEC) on Redstone Arsenal in Huntsville, Alabama. The motivation of this research is the need to make the GPS system more robust. Since the military relies heavily on GPS to guide munitions, its integrity is of the utmost importance. In addition to some inherent flaws, the GPS signal is the target of enemy jamming and spoofing in hostile environments. The point of this research is to provide the U.S. military with a more reliable form of navigation and guidance. To that end, advanced algorithms that combine GPS with inertial systems are investigated. These hybrid systems provide more robust navigation solutions.

1.2 Background and Literature Survey

The GPS signal has three basic shortcomings. The first is the power level of the received signal. For a civilian user, the power of the received is approximately 10^{-16} watts [Misra and Enge, 2001]. This corresponds to a carrier power-to-noise density ratio (C/N_0) of approximately 45 dB-Hz [Spilker, Jr., 1996b]. The low power level of the GPS signal makes it very susceptible to jamming. Only about 3.2 picowatts of incident RF interference is required to disable a civilian GPS receiver [Ward, 1996a]. The second shortcoming is the need to maintain line of sight between the user's antenna and the satellites, since blockage of the satellite signal disables the GPS receiver. During periods of blockage, it has been shown that the tracking loops inside the receiver are in a state of random walk [Dierendonck, 1996]. After periods of signal unavailability, the tracking loops must be reinitialized to re-lock to the satellite signal. The third shortcoming

of GPS is its low measurement rate. A GPS receiver produces estimates of position, velocity, and time (PVT) at approximately 1 Hz. For high dynamic vehicles, position updates are required at much higher rates.

Additional sensors are often fused with GPS to produce a more robust navigation system. The most common sensors used are inertial measurement units (IMUs). Other sensors include dopplerometers, altimeters, speedometers, and odometers [Bullock et al., 2006]. Inertial sensors provide acceleration measurements and angular rate measurements in the reference frame that they are mounted. Integrating the IMU measurements produces estimates of the user's change in velocity and change in position. The strength of an IMU is that they can provide updates at several hundred Hertz and are immune to jamming. However, IMUs have inherent flaws of their own. They suffer from biases and drifts that accumulate over time. Additionally, IMUs contain scale factors that must be estimated in order to determine the true accelerations and angular rates experienced [Flenniken, IV, 2005].

The strengths and weaknesses of IMUs and GPS complement each other. The measurements provided by an IMU serve to update the user's PVT between GPS updates. During times of satellite occlusion, the IMU can be used to dead reckon and can help in re-initializing the tracking loops when the satellites reappear. On the other hand, the GPS measurements can be used to correct the drift in the dead reckoned IMU position estimates. The bias's and scale factors in the IMU's can also be estimated at the GPS updates [Godha, 2004].

The Kalman filter is the default method of integrating GPS and IMU measurements. Researchers have shown that combining the two systems through a Kalman filter provides overall improved performance [Levy, 1997]. However, there is always the question

of how exactly to fuse the two systems. Currently, the various architectures used to meld GPS and IMU systems fall into three categories [Kreye et al., 2004]. The three classes of coupling, in order of complexity, are loose coupling, tight coupling, and ultra-tight coupling, which is also referred to as deeply integrated (DI).

1.2.1 Loose Coupling

Most integrated navigation systems are loosely coupled. The loosely coupled architecture is the simplest coupling scheme and provides significantly better performance than stand alone GPS or INS systems. Figure 1.1 shows a block diagram of a loosely coupled system which uses the GPS and INS to separately produce estimates of the platform's navigation states. The GPS and INS estimates are then combined by a Kalman filter which produces improved state estimates and estimates the error states of the INS system [Kreye et al., 2004]. An advantage of the loose coupling method is that the GPS receiver itself does not need to be modeled extensively. However, the GPS receiver contains internal filters and the noise in the provided GPS measurement is not white. The colored noise means the Kalman filter is no longer an optimal observer and requires additional tuning. In addition, a loosely coupled system only receives GPS measurements if there is enough information to complete a full, GPS alone, position solution [Gautier, 2003].

1.2.2 Tight Coupling

Tightly coupled systems combine the inertial sensors and GPS at a lower level. Figure 1.2 shows a diagram of a generic tightly coupled system. The navigation filter

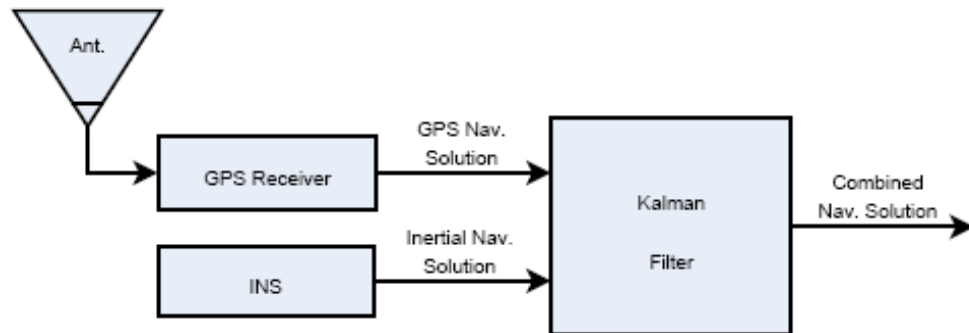


Figure 1.1: Loosely Coupled Architecture [Hamm, 2005]

receives the raw pseudo-range and pseudo-range rate measurements from the GPS receiver. This allows the filter to still estimate the error states in the INS when there is not a complete GPS solution available. Many tightly coupled systems also use the acceleration information provided by the INS to predict the Doppler shift caused by the user's dynamics. This improves the performance of the GPS receiver's tracking loops and their ability to maintain lock in highly dynamic environments [Kreye et al., 2004]. The bandwidths of the tracking loops can also be narrowed because of the reduced need to track user dynamics. The main drawback of the tightly coupled system is its complexity [Gautier, 2003].

1.2.3 Deeply Integrated or Ultra-tight Coupling

Ultra-tight coupling or deep integration is the most complicated and involved form of INS/GPS fusion. Figure 1.3 shows a general block diagram of a DI/ultra-tight system. In a DI navigation system, the INS and the GPS receiver are tied together at the most basic level. The raw IMU measurements are fed directly into the navigation Kalman

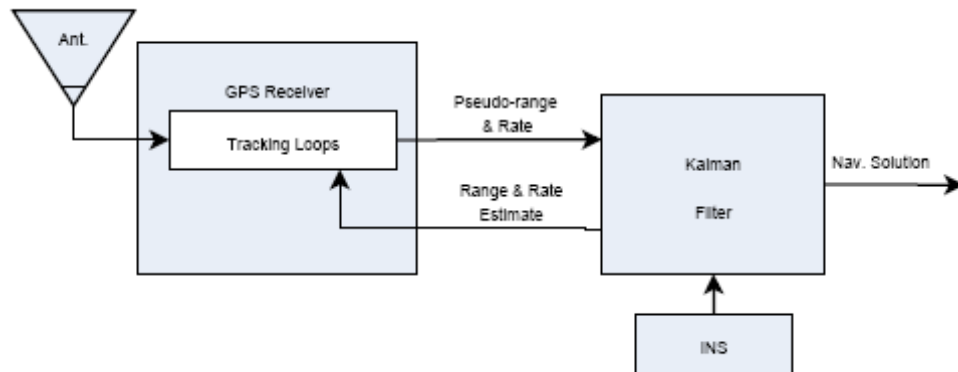


Figure 1.2: Tightly Coupled Architecture [Hamm, 2005]

filter. The filter then predicts the pseudo-range and pseudo-range rates. The filter’s predictions are then used to control the inputs to the integrate and dump functions inside the GPS receiver. The signals produced by the integrate and dump functions are then processed by the filter to estimate the errors in the IMU related states. In a complete DI system, the task of tracking the GPS signals is accomplished by the navigation filter itself [Kreye et al., 2004].

The original concept of ultra-tight coupling/DI is based around vector tracking [Spilker, Jr., 1996a]. In a vector tracking loop, the GPS satellite signals are not processed individually in their own channels. Instead, they are processed collectively by a single algorithm. The pseudo-range and pseudo-range rate observed by the user on the GPS signals are defined by the user’s position and velocity and the satellite geometry. The satellite geometry can be predicted by using the broadcast satellite ephemeris. The user’s position and velocity can then be determined from the received signals. Conversely, the GPS signals received from the satellites can be predicted based on the user’s position

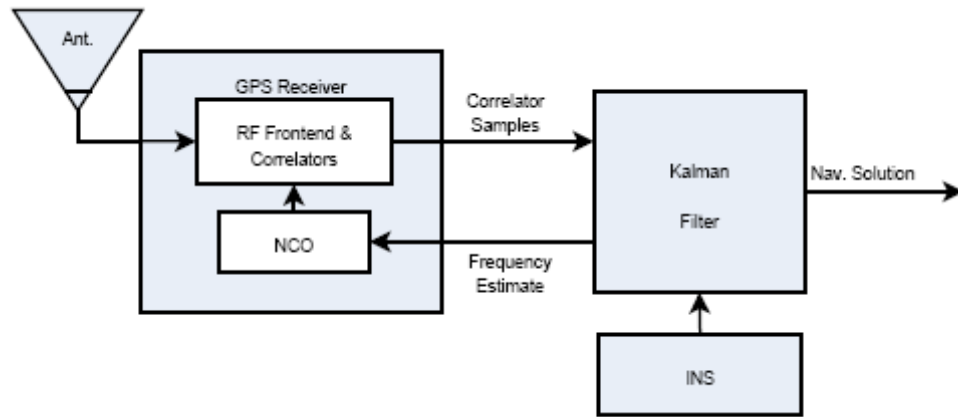


Figure 1.3: Ultra-Tightly Coupled Architecture [Hamm, 2005]

and velocity. Predicting the GPS signals based on the user’s navigation states and then updating the user navigation states based on the residuals formed by the correlators is the basis of DI [Pany et al., 2005, Petovello and Lachapelle, 2006].

1.2.4 Raytheon Method

The Raytheon company obtained a patent for their method of performing ultra-tight coupling in 1998 [Horslund and Hooker, 1999]. Figure 1.4 shows the block diagram of their method included in the patent. The Raytheon method uses a NAV function to estimate the user’s navigation states in the earth centered, earth fixed (ECEF) coordinate system. The NAV function is driven by the output of a GPS residual Kalman filter and IMU’s. The states include user position, velocity, IMU alignment states, and initial attitude. The estimated state vector of the NAV function is used by a line of sight (LOS) geometry function to predict the pseudo-range and pseudo-range rate for each individual satellite being tracked. The LOS function then sends the predictions to a

signal processing chip (SPC) that performs the integrate and dump functions for each channel (satellite). The outputs of the integrate and dump functions are then used to form pseudo-range and pseudo-range rate residuals. These residuals are then processed by a Kalman filter. The Kalman filter estimates corrections to the navigation states from the GPS residuals. These corrections are in turn used by the NAV function to close the loop.

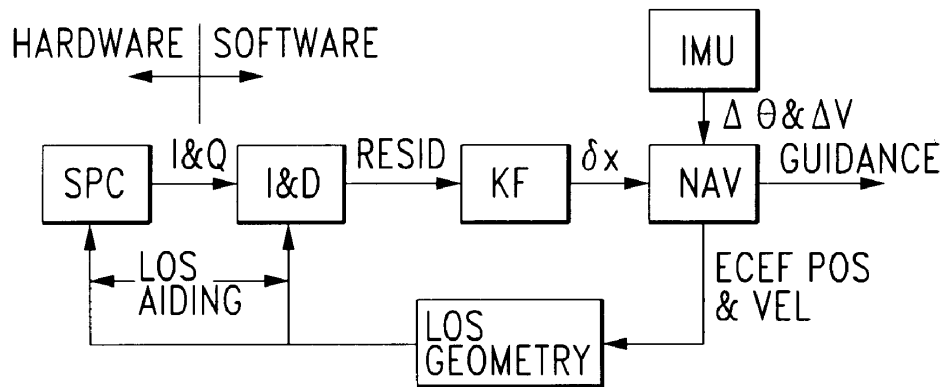


Figure 1.4: Raytheon Method [Horslund and Hooker, 1999]

1.2.5 Anthony Abbott Method

Anthony Abbott developed a different ultra-tight method and filed for a patent in 1999 [Abbott and Lillo, 2003]. Figure 1.5 shows a block diagram of his method. The Abbott method operates on similar principles as the Raytheon method, the main difference is that Abbott uses a federated Kalman filtering scheme [Carlson, 1988]. A large integration Kalman filter updates the error state vector every second. The previous epochs error state vector and the IMU samples are used between epochs of the integration filter

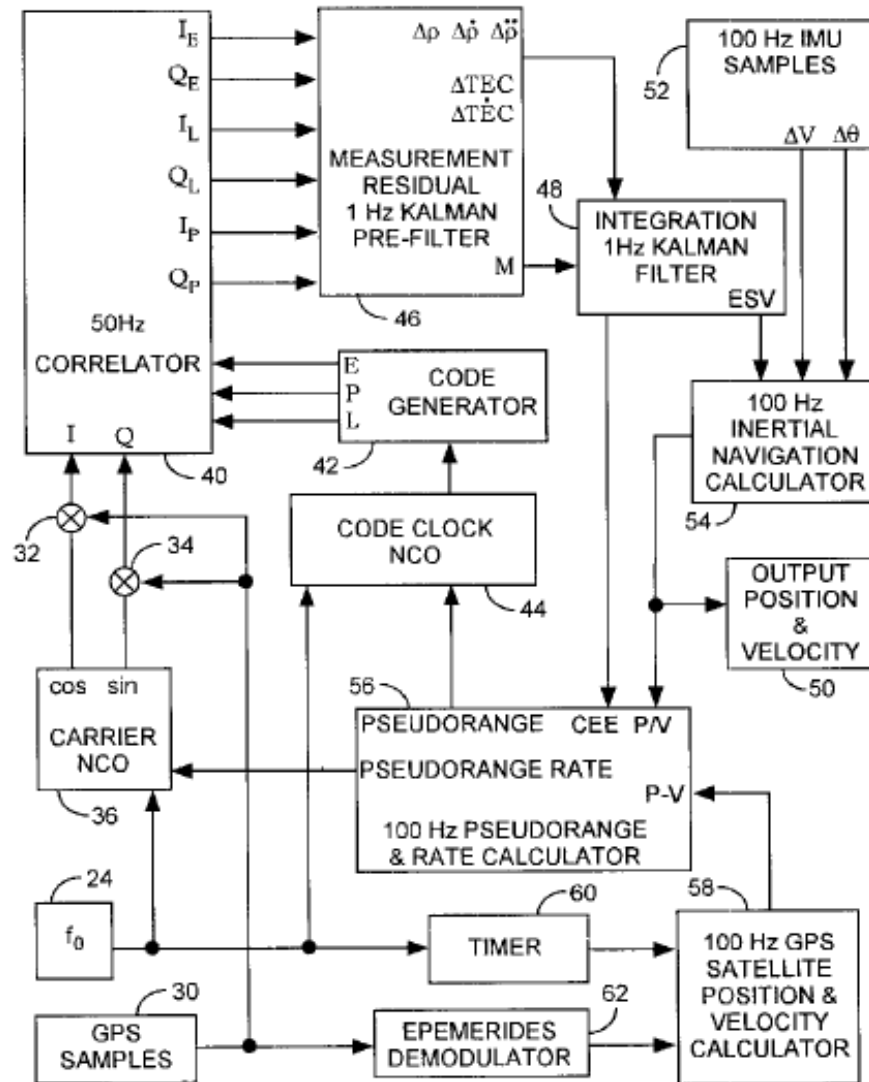


Figure 1.5: Anthony Abbott Method [Abbott and Lillo, 2003]

by the inertial navigation calculator. The inertial navigation calculator provides estimates of the user's navigation states. These estimates are used by the pseudo-range and pseudo-range rate calculator to predict the inputs to the 50 Hz correlator bank. The output of each bank of correlators is processed by a Kalman prefilter which exists for each satellite being tracked. The prefilters process the outputs of their respective correlators between updates of the integration filter. At measurement epochs of the integration filter, each prefilter provides the integration filter with measurements of pseudo-range, pseudo-range rate, and pseudo-range acceleration. The prefilters also provide an empirically determined 3x3 measurement covariance matrix. The integration filter processes the measurements of the smaller prefilters and updates the error state vector. The update error state vector is then used until the next measurement epoch of the integration filter.

1.2.6 Draper Method

The Charles Stark Draper Laboratory patented their own DI algorithm in 2001 [Gustafson et al., 2001]. Figure 1.6 shows a block diagram of the Draper method, which is similar in basic concept to the Raytheon and Abbott methods. The inputs to the correlation and integration block are derived from the user's navigation state. The outputs of the correlation and integration block are then used to update the navigation state vector. The major distinction in the Draper method is the filtering algorithm used to process the GPS and IMU measurements. The navigation filter process used by Draper "are significant departures from traditional Kalman and extended Kalman filter algorithms generally used for GPS-based navigation." In addition to their unorthodox filtering algorithm's, Draper Labs makes use of an extended bank of range

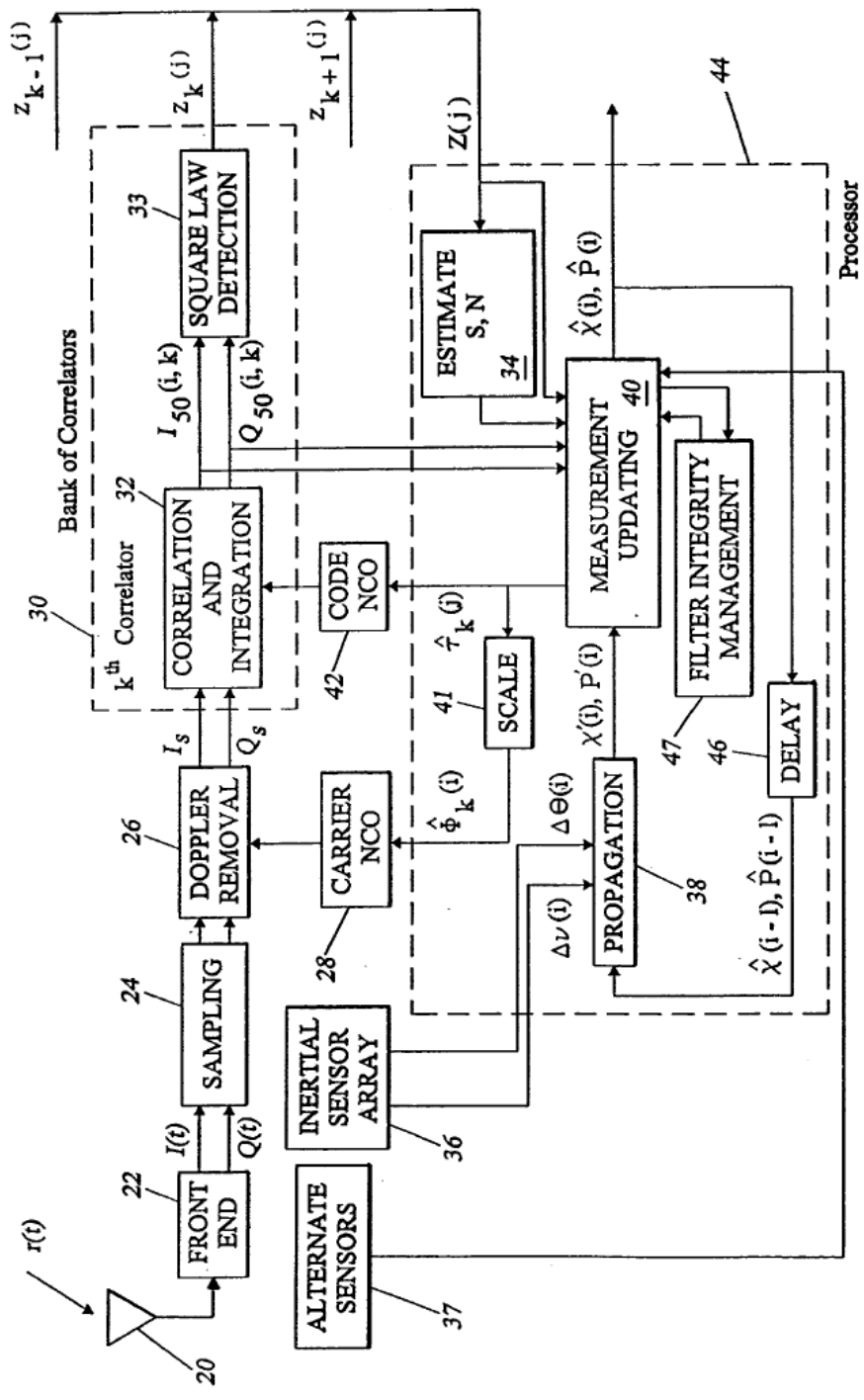


Figure 1.6: Draper Method [Gustafson et al., 2001]

correlators [Gustafson et al., 2000]. An analysis of the Draper labs method is given in [Gustafson and Dowdle, 2003].

1.3 Contributions and Outline

This thesis describes unique Kalman filter based tracking algorithms and provides details of their implementation in a software GPS receiver. The software receiver uses samples of the GPS signal to compute its location. The tasks of signal acquisition, signal tracking, navigation message decoding, and position determination are all performed in software using the sampled data. The Kalman filter based tracking algorithm's are used to replace the tracking loops found in traditional receivers. The algorithms offer improved tracking performance by using time-varying gains. The Kalman filter based algorithm's are verified using data collected from a high fidelity Spirent GPS constellation simulator. The ability of the algorithms to rapidly reacquire the satellite signals after a blockage is demonstrated using the GPS simulator. The performance of the algorithms during the blockage is also demonstrated.

This research is important because the majority of research and analysis of the different ultra-tight methods has been performed by the inventors themselves [Hamm, 2005]. Currently, there is a lack of independent validation of the various ultra-tight coupling architectures [Rounds, 2004]. The exact mathematical details of how these algorithms are implemented also remain largely unpublished [Soloviev et al., 2004, Kreye et al., 2004, Hamm, 2005]. The goal of the research program supported by AMRDEC is the eventual construction of a DI GPS navigation system. Vector (Kalman filter based) tracking methods are an intermediate step in this process and are examined in this thesis.

Chapter 2 of this thesis describes the signals used by GPS. The signals broadcast by the satellites are composed of three components. These components are a sinusoidal carrier, a pseudo-random noise (PRN) sequence, and a binary data message. The signal structure and properties of the individual components are discussed and how they apply to position determination. Chapter 3 details the operations of a software receiver and the tasks of signal acquisition and signal tracking are explained. The concept of tracking loops is introduced and explained. The least squares and extended Kalman filter methods of position determination are also described in Chapter 3. Chapter 4 explains two methods of replacing tracking loops with Kalman filters. The first method combines the tracking of the carrier and PRN sequence into a single Kalman filter for each channel. The second method combines the tracking of all the PRN sequences into a single Kalman filter. This method is commonly referred to as a Vector Delay Lock Loop. Chapter 5 describes the test setup used to validate the effectiveness and advantages of the tracking algorithm's. The experimental results that validate the utility of the algorithms are presented and explained. Chapter 6 summarizes the contributions of the thesis and recommends future work in this research area.

CHAPTER 2
GPS SIGNAL STRUCTURE

2.1 Overview of GPS Signals

This chapter provides details on the GPS signal structure. The GPS system is a code division, multiple access (CDMA) digital communication link. The satellite signals consist of a sinusoidal carrier, a digital navigation message, and a unique wide bandwidth pseudo random noise (PRN) sequence. The navigation message and PRN sequence (code) are encoded in the phase of the carrier signal by using binary phase shift keying (BPSK) modulation techniques [Sklar, 2001]. The system's multiple satellites all operate at the same frequency while not interfering with each other. Such operation is possible through the use of the unique PRN sequences. Additionally, the PRN codes are used to measure the range between each satellite and the user. The data message broadcast by each satellite provides the sufficient information to calculate the position of the satellite. Using range measurements to four or more satellites and using the data message to determine the satellite's positions, the user can solve for his position on the Earth.

The satellite signals currently consist of civilian and military components. The civilian component of the signal is available to all users and it can be used for position determination. The military portion of the satellite signal is only available to secure users and provides enhanced positioning capabilities. For the work described in this thesis, only the civilian component of the GPS signal is used.

The user must obtain the data message to determine the satellite positions. The user must also measure the phase of each of the PRN sequences being received from

the satellites. Measuring the PRN phase of each received code amounts to measuring the relative range to each satellite. It is the job of the receiver to accomplish these tasks, which is done by exploiting the properties of the carrier and PRN signals. The autocorrelation properties of the PRN sequences are used to determine the phases of the received PRN codes.

2.1.1 GPS Carrier Signal

The GPS satellites broadcast at two different center frequencies. The link 1 (L_1) frequency is approximately 1575.42 MHz and the link 2 (L_2) frequency is roughly 1227.6 MHz. The exact frequencies of L_1 and L_2 are shown in Equation (2.1).

$$L_1 = 154 \cdot 10.23 \text{ MHz} \quad (2.1)$$

$$L_2 = 120 \cdot 10.23 \text{ MHz}$$

The L_1 and L_2 frequencies are both multiples of a 10.23 MHz master clock. The L_1 and L_2 signals are partially offset from their nominal values when transmitted to help correct for relativistic effects. Additionally, dual frequency users can compensate for ionospheric effects [Spilker, Jr., 1996b].

The L_1 signal, broadcast by the i th satellite, is composed of an in-phase and quadrature component, as shown below in Equation (2.2).

$$S_{L1_i} = \sqrt{2P_c}G_i(t)D_i(t)\cos(\omega_1t + \phi) + \sqrt{2P_{p,L1}}P_i(t)D_i(t)\sin(\omega_1t + \phi) \quad (2.2)$$

Both components consist of a PRN sequence (also called Gold code), the navigation data message, and their sinusoidal carriers. The in-phase component is modulated by Coarse/Acquisition (C/A) Gold code $G_i(t)$ and the data message $D_i(t)$ of the satellite. The quadrature component of the L_1 signal is modulated by Precision (P) Gold code $P_i(t)$ and the same data message as the in-phase component. The C/A code, P code, and data message are all specific to each satellite. The power P_c of the in-phase component is 3 dB stronger than the power of the quadrature component, P_p [Spilker, Jr., 1996b].

The L_2 signal is modulated by the same data message as the L_1 signal, as shown in Equation (2.3).

$$S_{L2_i} = \sqrt{2P_{p,L2}} P_i(t) D_i(t) \cos(\omega_2 t + \phi) \quad (2.3)$$

The L_2 signal is modulated by either P or C/A code. Currently, the L_2 channel is modulated with P-code. Since access to the P-code is proprietary, it was not available during the writing of this thesis. Consequently, P-code and the L_2 signal will not be discussed further.

2.1.2 Gold Code

The PRN sequences, $G_i(t)$ and $P_i(t)$, used by the satellites are called Gold codes. The Gold codes have a very high bandwidth and are used to spread the spectrum of the data message over a much wider bandwidth. In the receiver, the spreading effect of the PRN sequences is removed by using locally generated replicas of the broadcast Gold codes. Each satellite transmits its own unique Gold code and the user receives multiple satellite signals at nearly the same frequency. The user can track the individual codes received because the Gold codes are nearly orthogonal to each other. This means

that if the receiver is using one Gold code to track a specific satellite in one channel of the receiver, the signals received from the other satellites are effectively nulled in that channel. This aspect of Gold codes makes the CDMA properties of the GPS system possible [Misra and Enge, 2001].

Gold codes are formed by modulo-2 addition of two maximal length sequences. Maximal length sequences are formed by shift register generators (SRG) and repeat themselves every $2^N - 1$ bits, where N is the number of stages of the SRG. The C/A Gold codes used by the satellites are all formed by adding the same two maximal length sequences, labeled as $G1$ and $G2$. However, the Gold codes broadcast by each satellite are different. The different Gold codes are formed by shifting the $G1$ and $G2$ sequences with respect to each other in time before adding them. Each Gold code has a specific time shift associated with it. All that is needed to produce a specific satellite's PRN sequence is the delay time defined for that satellite and the $G1$ and $G2$ sequences [Spilker, Jr., 1996b].

The C/A Gold codes are transmitted at a chipping rate of 1.023 Mbps. The individual symbols of the codes are referred to as chips, as opposed to the bits of the navigation message. A code $g(t)$ is originally generated as a binary sequence and is used to modulate the phase of the carrier signal, as shown below in Equation (2.4).

$$S(t) = \cos(\omega_1 t + \pi g(t)) \quad (2.4)$$

Since the phase of the signal is being shifted by 180° , the Gold code can be equivalently expressed as shown in Equation (2.5).

$$S(t) = G(t) \cos(\omega_1 t) \quad (2.5)$$

In Equation (2.5), the original binary sequence $g(t)$ has been mapped to a sequence of ± 1 's, $G(t)$. The form shown in Equation (2.5) is the typical way the modulation of the Gold code is represented. Gold codes have a very narrow time autocorrelation function,

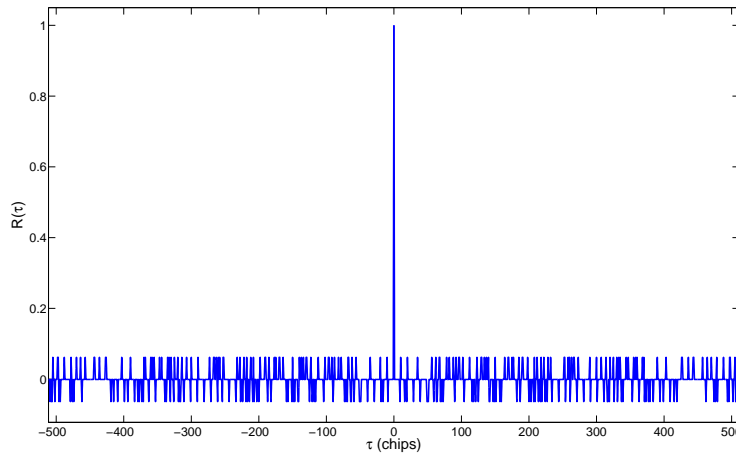


Figure 2.1: Autocorrelation Function of Gold Codes

as can be seen in Figures 2.1 and 2.2. The autocorrelation function is linear for $\tau < 1$ chip of misalignment. Beyond $\tau = 1$ chip, the autocorrelation function is approximately zero. The autocorrelation properties of Gold code are used to track the satellite signal and make pseudorange measurements. The cross-correlation properties of Gold code allow the satellites to all transmit on the same frequency. The different Gold codes are nearly orthogonal to each other, as can be seen in Figure 2.3. When a receiver collects RF data, multiple satellite signals are recorded in the same frequency band. The receiver “picks out” a specific satellite by multiplying the data by an in-phase replica of the satellite’s Gold code. This multiplication effectively cancels out all the other satellite signals while removing the PRN modulation from the signal received from the selected satellite.

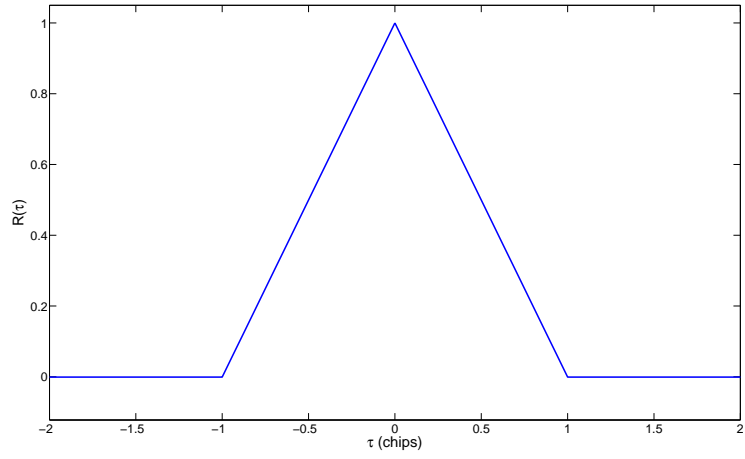


Figure 2.2: Ideal Autocorrelation Function of Gold Codes

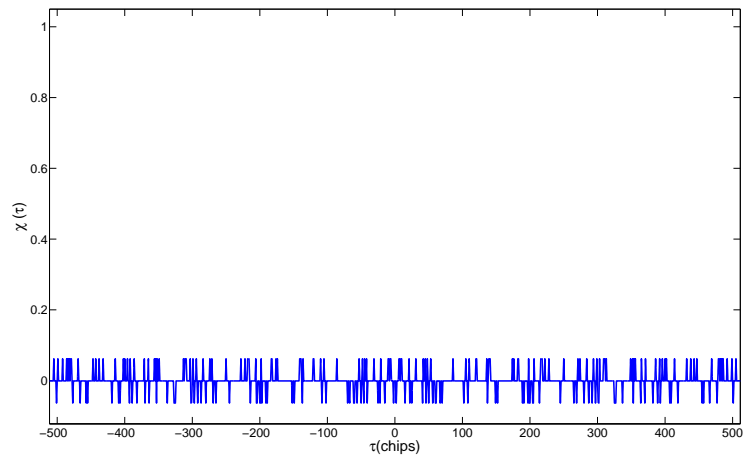


Figure 2.3: Cross-Correlation of Different Gold Codes

2.1.3 Data Message

The data message transmitted by each satellite contains information about the satellite's orbit, its health, GPS system time, and almanac data for the other satellites in the constellation. The data message contains the necessary information for users to synchronize their clock to system time and determine their position.

The data message is broadcast at 50 bits per second (bps). The message is arranged into thirty bit words. Ten words constitute a subframe and five subframes make a frame. Figure 2.4 shows how the navigation message is arranged. In each frame, the first three subframes contain clock and ephemeris information for the transmitting satellite. Subframes four and five alternately provide support information, such as almanac data and ionospheric correction data [US Dept. of Defense, 2000]. The satellite ephemeris data is information about the satellite's orbit. Using the ephemeris data, the satellite's position is calculated. The almanac data is approximate ephemeris data for the other satellites in the constellation. It is not as accurate as the ephemeris data in subframes 2 and 3, but is valid for a longer period of time [Tsui, 2000].

2.2 Conclusion

The ability of users to calculate their position, velocity, and time is made possible by the GPS signals. The satellites all broadcast unique PRN sequences. These PRN sequences allow the satellites to all broadcast on the same frequency and not interfere with each other. The PRN sequences are also used to make range measurements to the satellites. Each satellite also broadcasts a navigation message. This navigation message enables the user to precisely calculate the position of the emitting satellite. In addition,

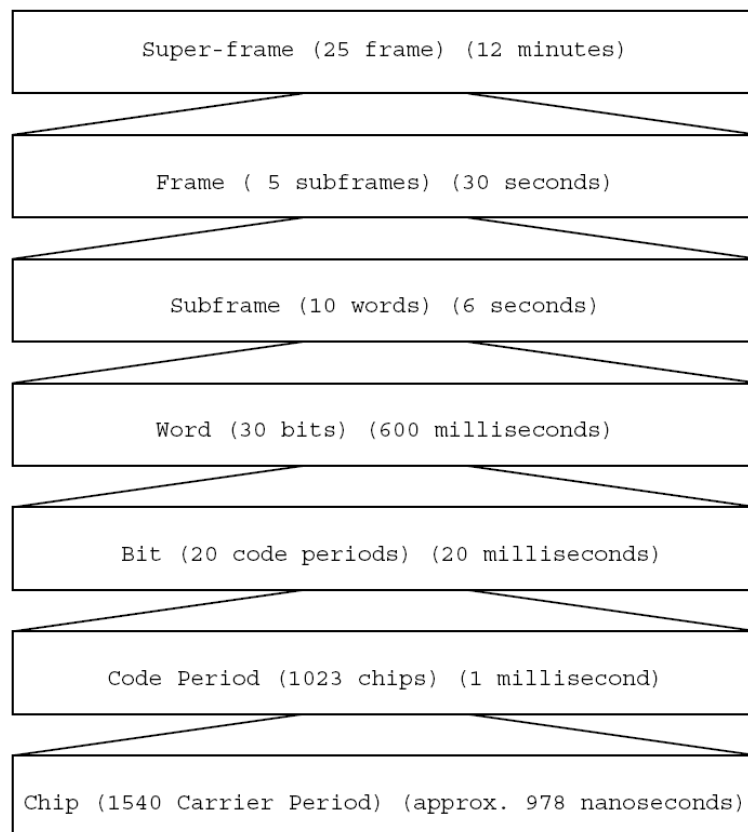


Figure 2.4: Arrangement of the GPS Navigation Message [Hamm, 2005]

the navigation message from one satellite can be used to calculate the approximate positions of the other NAVSTAR satellites. Users of the GPS system must process the received signals in order to extract the PRN code phases, navigation data, and carrier frequency. The next chapter will explain the details of how a receiver processes the received signal.

CHAPTER 3

TRADITIONAL SOFTWARE GPS RECEIVER

3.1 Overview

This chapter provides an explanation of a software receiver. The first task of any receiver is to determine which satellites are in view. After determining that a given satellite signal is available, the receiver attempts to track the carrier and PRN components of the signal. A traditional receiver uses a Costas loop to track the carrier and a Delay Lock Loop to track the PRN sequence. Using the output of the tracking loops, the navigation message for each satellite is decoded. The navigation message provides the user with enough information to calculate the positions of the satellites. Finally, the user can calculate his position using the pseudo-range measurements from the tracking loops.

The work presented in this thesis was done entirely in software. All the tasks of the receiver, including signal acquisition, satellite tracking, navigation message demodulation, and user position determination, were done using code in MATLAB[®] and C. The raw data was provided by a NordNav receiver, consisting of hardware and software components [Normack et al., 2002]. The NordNav hardware is a USB powered RF front-end, Fig. 3.1. The RF front-end performs the electronic operations to downconvert the L_1 signal and digitize it. The digitized signal is then saved for post processing. Using NordNav software, the data can be converted into a signed character representation, which can then be imported into MATLAB[®]. The data can also be processed using the NordNav software receiver graphical user interface (GUI), shown in Figure 3.2.



Figure 3.1: NordNav Receiver Front-end [Normack et al., 2002]

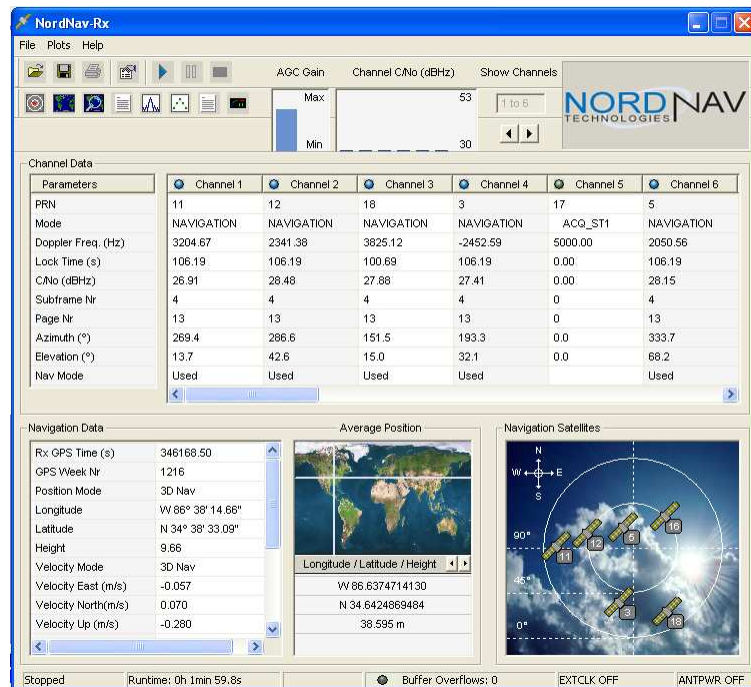


Figure 3.2: NordNav GUI [Normack et al., 2002]

3.2 Receiver Front-end

Before the L_1 signal is sampled, the signal is passed through several stages of amplification, down conversion, and filtering. Figure 3.3 shows a block diagram of the NordNav GPS receiver [Normack et al., 2002]. The L_1 signal is first received by an antenna and is then passed through a preamplifier. The preamplifier usually consists of filtering, burnout protection, and a low noise amplifier (LNA) [Dierendonck, 1996]. Next, the signal is mixed down to an intermediate frequency (IF). The GPS L_1 signal is broadcast at a frequency of 1.57542 GHz. The NordNav receiver mixes the L_1 signal down to an IF frequency of 4.1304 MHz. The output of the reference oscillator is used by the frequency synthesizer to generate the necessary local oscillators (LOs) for the downconverter. Finally, the downconverted L_1 signal is sampled by the analog to digital converter (ADC). The NordNav front end samples at a frequency of 16.3676 MHz and is capable of taking one, two, and four bit samples. The digitized IF data is then processed by the software receiver.

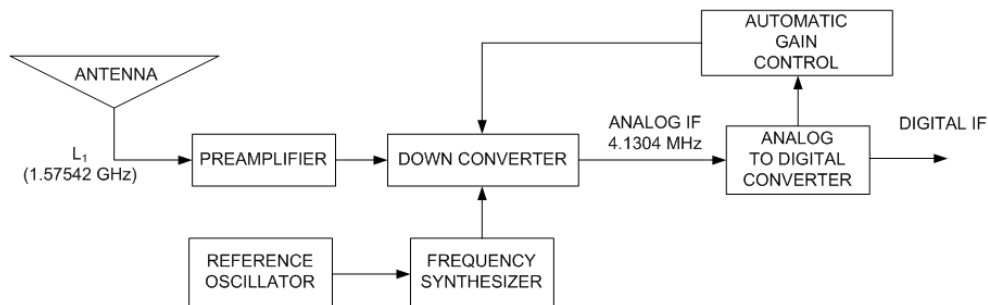


Figure 3.3: General GPS Receiver Block Diagram

3.3 Acquisition

Before the receiver can begin to track satellites, it must determine which satellites are in view. In addition, the receiver must determine the Doppler shift caused by the relative motion of the satellite vehicles and the code phase of the satellite's PRN sequence. The process of determining these variables is called acquisition. The Doppler shift estimate must be accurate to within a few tens of Hertz for the tracking loops to pull-in and lock to the carrier. The phase of the PRN sequence must be accurate to within one half chip because of the Gold code's autocorrelation function [Tsui, 2000]. The number of satellites that must be searched for can be reduced greatly by using *a priori* information, such as almanac data or a rough estimate of the receiver's location.

Many different algorithms exist to search over the possible Doppler shifts and code phases. However, the algorithms all function in basically the same manner. First, data is collected and a specific satellite is selected to be searched for. Between 1 and 20 milliseconds of data is usually used because the C/A code has a period of 1 millisecond and the likelihood of a data bit transition occurring increases after 20 milliseconds. A range of possible Doppler shifts is then defined. The algorithm searches over the range of Doppler shifts by starting at the lowest frequency and then incrementing through the range in frequency steps. For a stationary observer, the range of the Doppler shift is from -5 KHz to +5 KHz [Tsui, 2000]. A common step size for the acquisition algorithm is around 500 Hz. Each frequency estimate is used to shift the received signal to baseband. Assuming the frequency estimate is sufficiently close to the actual Doppler frequency, the received signal will be shifted to baseband and only the satellite's Gold code will remain. A replica version of the satellite's Gold code is then convolved with the received

PRN sequence. When the received and locally generated codes align to within half a chip, a peak will appear in the convolution function output. If the peak exceeds some threshold value, the acquisition program will generally consider the current frequency estimate and index of the convolution function to be the correct Doppler shift and code phase, respectively. The cross-correlation properties of the Gold codes prevents the PRN sequence from one satellite being mistaken for another.

The acquisition algorithm used in this research utilizes the circular convolution property of the Discrete Fourier Transform (DFT). The Doppler estimate is first used to shift the received data to baseband. The locally generated Gold code is then convolved with the received Gold code using the DFT. This approach allows the significant body of work devoted to making the DFT more numerically efficient to be utilized. Specifically, the Fast Fourier Transform algorithm can be applied. As the lengths of data sequences increase, it becomes increasingly faster to use the FFT instead of time domain convolution [Porat, 1997].

The cross correlation of two discrete sequences is defined in Equation (3.1).

$$c(n) = \sum_{m=0}^{N-1} x(m)y(n+m) \quad (3.1)$$

The definition of the DFT is given in Equation (3.2).

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad (3.2)$$

The cross correlation $c(n)$ of the two sequences $x(m)$ and $y(m)$ can be performed in the frequency domain using the DFT, as shown in Equation (3.3).

$$C(k) = X^*(k)Y(k) \quad (3.3)$$

This frequency domain approach to cross correlation can be used to simultaneously search over all possible PRN code phases [Bertelsen et al., 2004]. For the NordNav receiver, there are approximately sixteen samples per chip of the Gold code. The DFT based method therefore returns a code phase estimate that is precise to a sixteenth of a chip.

Figure 3.4 shows the results of the search algorithm graphically. The results of the search algorithm are plotted for the Doppler shift estimates used and the possible code phases. The coordinates of the peak in the center of the graph represent the correct Doppler shift and code phase of the received signal.

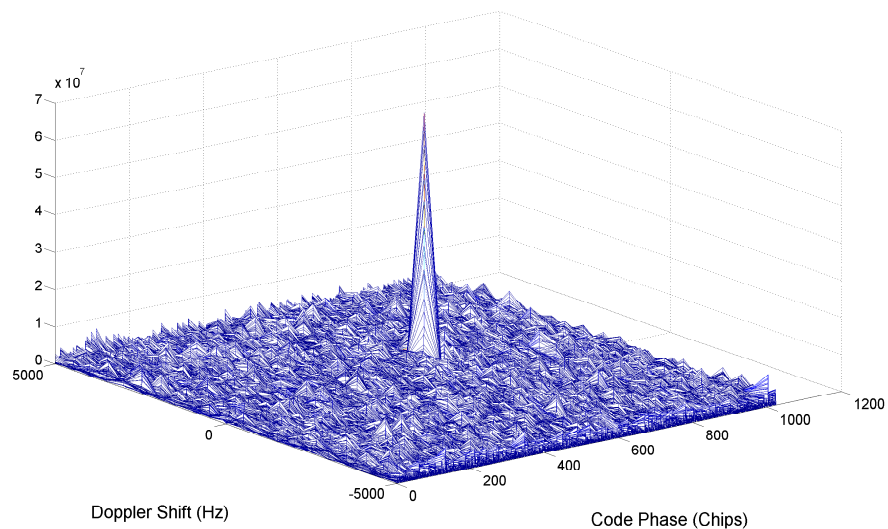


Figure 3.4: Acquisition

3.4 Tracking Loops

Tracking loops play a central role in the operation of a GPS receiver. Tracking loops allow the ephemeris information (sent by the satellites) to be decoded and used to determine their position. They are also used to make the pseudorange measurements that allow the user's position to be calculated. Tracking loops operate by generating replicas of the Gold code and carrier signal broadcast by each satellite. The tracking loops attempt to produce a replica of identical phase and frequency as what is received. The modulation of the carrier and Gold code is removed by using these locally generated replicas, leaving the data message behind. Pseudorange measurements are made by using the locally generated Gold code.

3.4.1 The Phase-Locked Loop

The concept of the phase-locked loop (PLL) underlies tracking loop operation. Figure 3.5 shows a basic PLL block diagram. All PLL's consist of three fundamental components. First, the synchronized oscillator is used to generate a replica of the received signal. The synchronized oscillator is generally a voltage controlled oscillator (VCO) in an analog PLL, meaning that its frequency is controlled by an input voltage. For digital designs, the synchronized oscillator is a numerically controlled oscillator (NCO), because its output is determined by a digital number. Second, a phase detector produces a signal that is proportional to the phase error between the output of the synchronized oscillator and the received signal. A simple realization of a phase detector is a multiplier. All phase detectors have an inherent nonlinearity in their output. Finally, a loop filter operates on the output of the phase detector to produce a control signal for the synchronized

oscillator. Most of the tracking characteristics of the PLL are controlled by the design of the loop filter. The PLL is a simple feedback loop which attempts to track a reference input by driving an error signal to zero. In the case of the PLL, the error signal is the phase difference between the local replica and the received signal.

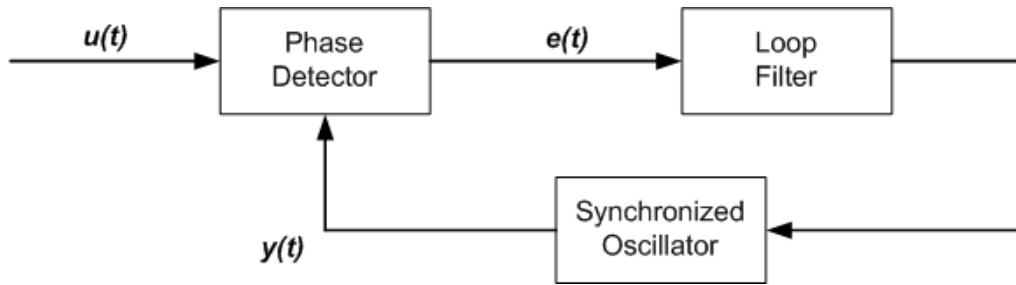


Figure 3.5: Basic Phase Locked Loop

When the output of a PLL's oscillator and the incoming signal are matched in phase and frequency, the PLL is considered to be locked. The way a phase-locked loop locks onto a reference signal can be explained by considering the analog PLL in figure 3.6. A sinusoidal reference signal $u(t)$ having frequency ω_r is applied as the input to the PLL, as shown in Equation 3.4.

$$u(t) = \sin(\omega_r t) \quad (3.4)$$

The output $y(t)$ of the reference oscillator is a cosine function of a different frequency ω_{vco} , shown in Equation 3.5.

$$y(t) = 2 \cos(\omega_{vco} t) \quad (3.5)$$

The phase detector in this case is a multiplier (mixer). The mixer output $e(t)$ can be expressed as Equation (3.6) by using trigonometric identities.

$$e(t) = 2 \sin(\omega_r t) \cos(\omega_{vco} t) = \sin[(\omega_r + \omega_{vco})t] + \sin[(\omega_r - \omega_{vco})t] \quad (3.6)$$

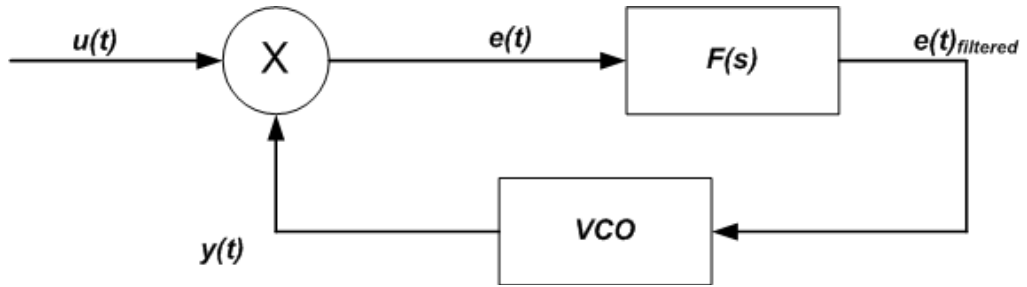


Figure 3.6: Analog Phase Locked Loop

The higher frequency term in Equation (3.6) is eliminated by the loop filter, leaving behind a sinusoid whose frequency is dependent upon the frequency error between the received and generated signals, shown in Equation (3.7).

$$e(t)_{filtered} = \sin[(\omega_r - \omega_{vco})t] \quad (3.7)$$

Assuming that ω_r is greater than ω_{vco} , the filtered error signal is initially positive. The oscillator is therefore driven to increase its frequency, bringing it closer to the frequency of $u(t)$. If the frequency of the VCO is driven to the frequency of the input before the phase of the filtered error signal exceeds 180 degrees, the initial frequency error is within the lock-in range of the PLL. If the initial frequency error is within the lock-in range the

frequency error will decrease rapidly toward zero. Figure 3.7 shows the time response of a PLL VCO when an input is applied that is within its lock-in range [Egan, 1998].

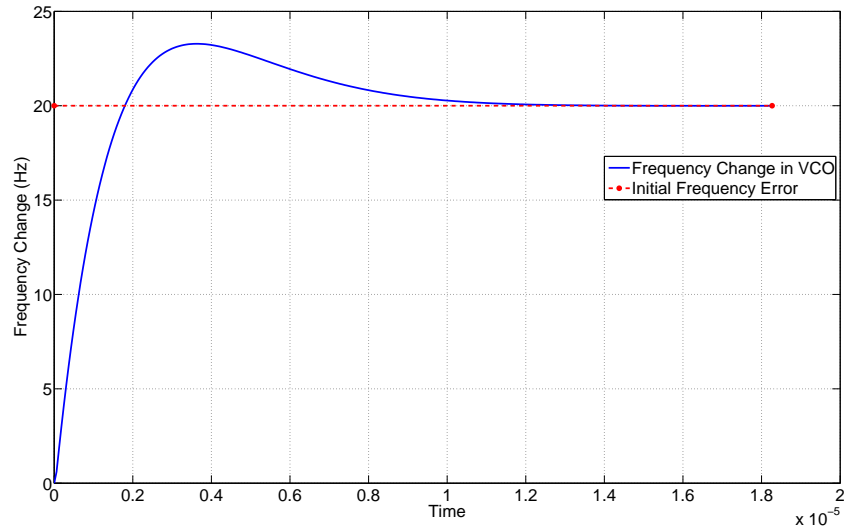


Figure 3.7: Phase Locked Loop Acquiring A Signal Inside The Lock-In Range

If the frequency error is outside of the lock-in range, the phase of the error signal will exceed 180 degrees. In this case the error signal will become negative and drive the oscillator away from the frequency of the input. The sign of the error signal will eventually change, however, and drive the VCO output in the correct direction. This process of the oscillator being driven intermittently in the right and wrong directions has the net effect of synchronizing the oscillator with the reference, because the frequency of the error signal changes as the oscillator approaches the correct frequency. When the error signal is driving the oscillator in the correct direction, its frequency decreases. When the error signal begins pushing the oscillator in the wrong direction, its frequency increases. The net effect of this aharmonic phenomena is that the oscillator is driven

in the right direction longer than it is driven in the wrong direction. Initial frequency errors that produce this type of behavior before the PLL locks are said to be within the pull-in range of the PLL [Best, 1999]. Figure 3.8 shows the time response of a PLL VCO when an input is applied that is within its pull-in range.

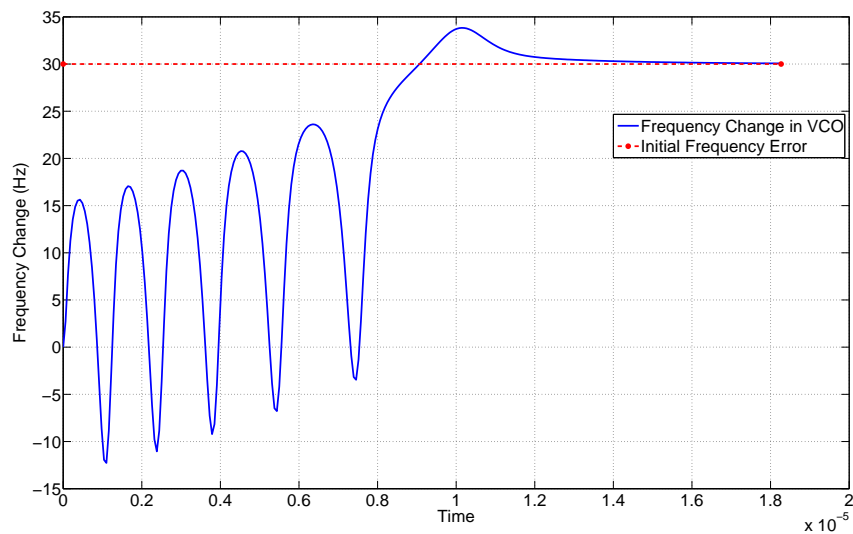


Figure 3.8: Phase Locked Loop Acquiring A Signal Inside The Pull-In Range

The phase-locked loop is inherently a nonlinear device. Analysis of the PLL can be simplified however, by linearizing it about an operating point. Classical control techniques can then be used to analyze the system. Assuming the PLL is initially locked, the loop is linearized so the response to changes in the phase of the input can then be analyzed. The small angle approximation given in Equation (3.8) is used to linearize the phase detector.

$$\sin(\theta) \approx \theta \tag{3.8}$$

The block diagram in Figure 3.9 is produced by expressing the PLL in terms of the phases of the signals. The new variables Θ_1 and Θ_2 represent the phase of the input and the VCO signals in the Laplace domain, respectively. The VCO is modeled as an integrator because inputs to the VCO cause a change in frequency. The error between Θ_1 and Θ_2 is assumed to be sufficiently small so that the phase detector operates linearly. The gain K_d represents the gain of the phase detector (discriminator). The synchronized oscillator is modeled as an integrator with a gain of K_o [Best, 1999].

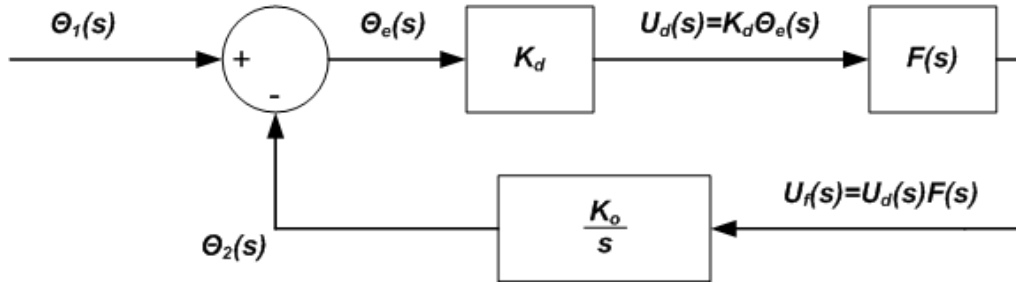


Figure 3.9: Linearized Phase Locked Loop

Using the linearized model in Figure 3.9, the phase-transfer function $H(s)$ is expressed in Equation (3.9).

$$H(s) = \frac{\Theta_2(s)}{\Theta_1(s)} \quad H(s) = \frac{K_o K_d F(s)}{s + K_o K_d F(s)} \quad (3.9)$$

The transfer function between the input and the error signal $\Theta_e(s)$ is given by Equation (3.10).

$$\begin{aligned} H_e(s) &= \frac{\Theta_e(s)}{\Theta_1(s)} \\ H_e(s) &= \frac{s}{s + K_o K_d F(s)} \end{aligned} \quad (3.10)$$

The loop filter $F(s)$ largely determines the response of the PLL. Specifically, the number of poles in the filter at $s = 0$ determines if the loop can track a phase step, a frequency step, or a frequency ramp. The loop filter can be expressed as Equation (3.11), where $P(s)$ and $Q(s)$ are both polynomials in s and the number of poles at the origin in $F(s)$ is represented as N [Best, 1999].

$$F(s) = \frac{P(s)}{Q(s)s^N} \quad (3.11)$$

The order of a PLL is a common parameter used to describe its tracking abilities. The order of a PLL is equal to the number of poles at $s = 0$ in its loop filter plus one and is the same as the system type. It should be noted that the terminology used in PLL analysis is slightly different than that used in describing classical control system type.

The ability of the PLL to track different inputs with zero steady state error is dependent on the order of the loop filter. For example, consider a unit step in phase, shown in Equation (3.12).

$$\Theta_1(s) = \frac{1}{s} \quad (3.12)$$

This type of input occurs when the distance between the receiver and transmitter changes abruptly. The final value theorem of the Laplace transform can be used to determine the steady-state error, displayed in Equation (3.13).

$$\begin{aligned} \Theta_e(s) &= H_e(s)\Theta_1(s) \\ \theta_e(\infty) &= \lim_{s \rightarrow \infty} s\Theta_e(s) \\ \theta_e(\infty) &= \lim_{s \rightarrow \infty} \frac{s^2 s^N Q(s)}{s(s \cdot s^N Q(s) + K_o K_d P(s))} \end{aligned} \quad (3.13)$$

The steady state error approaches 0 for any number of poles at the origin in $F(s)$, including $N = 0$. Therefore, a PLL of any order can track a step change in phase with no steady state error.

The response of different order PLL's to various inputs can be determined by applying the final value theorem of the Laplace transform. The ability of different order loops to track various inputs with zero steady state error is summarized in Table 3.1. As stated above, a phase step can be tracked by all order loop filters. A frequency step can only be tracked by second and higher order loops. A frequency ramp can only be tracked by third and higher order PLL's.

Table 3.1: Steady State Error Of Different Order Loops To Different Inputs

Loop Order	Phase Step	Frequency Step	Frequency Ramp
1	0	Finite	∞
2	0	0	Finite
3	0	0	0

The response of the PLL to the inputs displayed in Table 3.1 is very important. The inputs considered represent the effects that changes in position, velocity, and acceleration have on the received signal. These changes in position, velocity, and acceleration are caused by the user dynamics and the motion of the GPS satellites. A change in position appears as a change in the phase of the received signals. A change in velocity causes the relative Doppler shift between the transmitter and receiver to change. The change in frequency of the received signal translates into a phase ramp. If the receiver accelerates relative to the transmitter, a ramp in frequency is observed on the received signal. A ramp in frequency translates into a parabolic function of phase.

Note, as shown in Table 3.1, only third and higher order loops can track out a frequency ramp. However, third and higher order loops are not unconditionally stable [Best, 1999]. In practice, most PLL's are first and second order loops, because they are unconditionally stable.

It is apparent that one of the major drawbacks of tracking loops is their limited ability to track the dynamics of the relative motion between the receiver and satellites. This can cause the tracking loops to become unlocked during high dynamic maneuvers. In order to improve the response of the tracking loops to platform dynamics, the bandwidth of the loop filter must be increased. However, this also increases the amount of noise passed by the filter.

A second shortcoming of classical tracking loops is that they employ fixed bandwidths and gains, which operate on the received data regardless of high or low SNR values. The signal to noise ratio (SNR) can change significantly when tracking satellites. Under these conditions, the loop operates off noise rather than ignoring the measurements. For example, when a satellite is blocked, the tracking loops are in a state of random walk [Dierendonck, 1996]. Depending upon the duration of the signal loss and SNR, the loop may need to be re-initialized to reacquire the signal. If the satellite outage is for a prolonged time, the acquisition process may have to be repeated.

Two tracking loops exist in traditional GPS receivers. One tracking loop is required to maintain phase-lock with the carrier and another is needed to maintain phase-lock with the Gold code. The loop that recovers the carrier is typically a Costas loop. The Gold code is tracked by a separate tracking loop called a Delay Locked Loop (DLL). Each loop is fed only the portion of the signal that it tracks. The output of the DLL is used to remove the Gold code modulation on the received signal and the Costas loop

then operates on the carrier modulated by the data bits only. In a similar manner, the synchronized replica of the carrier produced by the Costas loop is used to remove the carrier modulation from the signal that the DLL operates on.

3.4.2 Costas Loop

As mentioned before, the GPS carrier is modulated by the data message using binary phase shift keying (BPSK) modulation. This means that during every bit transition, the phase of the carrier changes by 180° . If a basic PLL were used to track the signal, it would see a 180° phase error every time a data bit transition occurred. Maintaining phase lock with such a signal would therefore be impossible using a regular PLL. The carrier tracking loop used in a GPS receiver must be insensitive to the 180° phase shifts. A Costas loop is the most common method used to track the BPSK waveforms in GPS.

Figure 3.10 shows the basic architecture of an analog Costas loop. The received carrier plus data modulation $D(t)$ is feed into the loop. The input is then multiplied by the loop's in-phase estimate of the carrier in the upper, in-phase (I) arm of the Costas loop. The input is multiplied by an orthogonal (90° out of phase) version of the carrier in the lower, quadrature (Q) arm. The multiplication of the sinusoids in the I and Q branches produces the sum and difference terms, shown previously in Equation 3.6. The loop filters, LPF_1 and LPF_2 , eliminate the high frequency terms. If the loop is phase-locked, the difference between θ_1 and θ_2 is zero, leaving the data message $D(t)$ in the in-phase channel to be decoded. The filters are generally matched to prevent prolonged settling times. The signals are then processed by the discriminator [Feigin, 2002].

The discriminator function uses the I and Q signals to generate an error signal that is immune to the 180° phase shifts caused by the navigation message. A simple

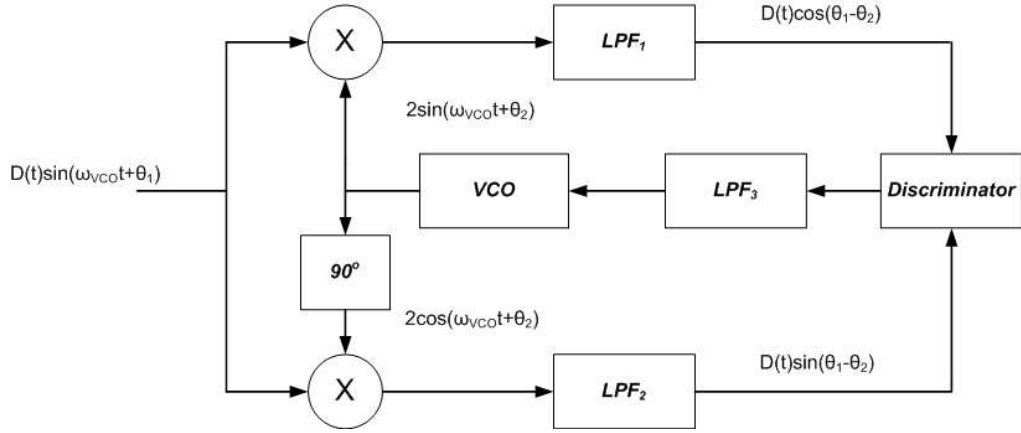


Figure 3.10: Costas Loop

discriminator function is a multiplier. Multiplying the I and Q signals together squares the data message, as shown in Figure (3.14).

$$\begin{aligned}
 D(t) \cos(\theta_{err}) D(t) \sin(\theta_{err}) &= \frac{D(t)^2}{2} [\sin(0) + \sin(2\theta_{err})] & (3.14) \\
 \frac{D(t)^2}{2} [\sin(0) + \sin(2\theta_{err})] &= \frac{\sin(2\theta_{err})}{2}
 \end{aligned}$$

Because the data message is a series of ± 1 's, squaring the signal leaves a series of 1's and therefore removes its effects from the carrier. The maximum likelihood estimator for the phase error is a four-quadrant arctangent discriminator for high and low SNR's [Ward, 1996b], presented in Equation (3.15).

$$\theta_{err} = \arctan \left(\frac{\sin(\theta_{err})}{\cos(\theta_{err})} \right) \quad (3.15)$$

The third low pass filter, LPF_3 , reduces noise in the system and eliminates harmonics created by the previous multiplications [Feigin, 2002].

In a software GPS receiver, the signals used in the Costas loop are not continuous, but sampled. The digital Costas loop is modified to accommodate discrete time signals, as shown in Figure 3.11. The low pass filters in the arms of the analog Costas loop are replaced by ‘integrate and dump’ functions. The products in the I and Q channels are summed over an integration interval and then the accumulated values are operated on. The integration periods must be coordinated to avoid integrating over a data bit transition. The integrate and dump functions filter out the harmonics caused by the multiplications and also provide an integration gain. A discriminator that is immune to the data message produces an error signal proportional to the phase error between the incoming signal and the in-phase replica. The low pass filter following the discriminator functions like the loop filter in a standard PLL and it defines most of the response characteristics of the Costas loop. The NCO is fed the output of the low pass filter to close the feedback loop.

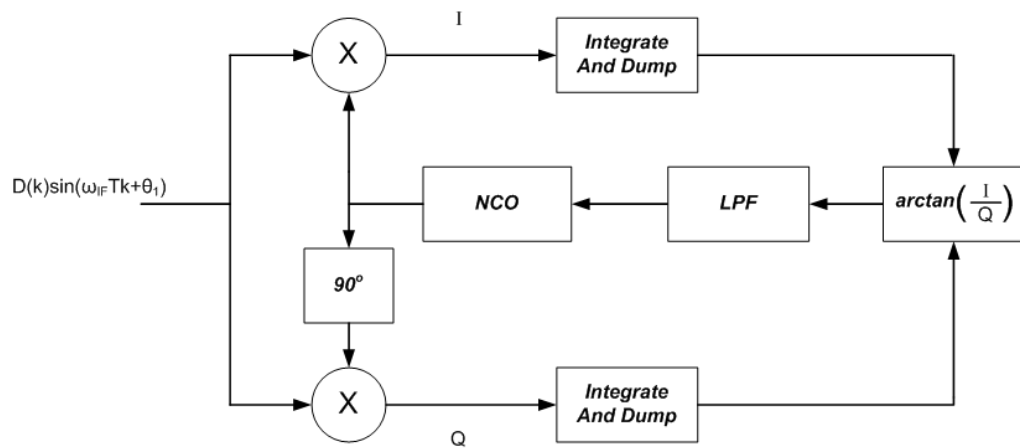


Figure 3.11: Digital Costas Loop

3.4.3 Delay Lock Loop

The Gold code that modulates the carrier is removed by multiplying the received signal by an in-phase replica of itself. Since the Gold code is a series of ± 1 's, multiplying it by itself results in a sequence of 1's. The caveat is that an in-phase version of the received Gold code must be maintained. A delay lock loop (DLL) is a control loop that maintains phase-lock between the received and replica codes, using the same basic principles as a PLL. An error signal proportional to the phase difference between the two signals is used to adjust the frequency of the replica after being low pass filtered.

The main difference between a PLL and a DLL lies in the generation of the error signal. Whereas the PLL uses a single replica signal, a DLL operates by generating three local replicas of the received code. These signals are referred to as the early, prompt, and late Gold codes. The prompt code is the DLL's in-phase estimate of the received Gold code. The early replica is advanced τ seconds from the prompt code. The late code is delayed τ seconds from the prompt code. The most commonly used value of τ corresponds to half the time duration of one chip of the Gold code in seconds. The received Gold code is multiplied by the local replicas and the three resulting signals are summed over an integration period. The three accumulated values are then processed by a discriminator function to produce an error signal [Spilker, Jr., 1996a].

The three accumulated values can be modeled mathematically as shown below in Equation (3.16) [Sayre, 2003].

$$\begin{aligned} \text{Early} &= A \sum_{n=1}^N G_i(n) \cdot G_{i,r}(n + \epsilon + \tau) \\ \text{Prompt} &= A \sum_{n=1}^N G_i(n) \cdot G_{i,r}(n + \epsilon) \end{aligned} \tag{3.16}$$

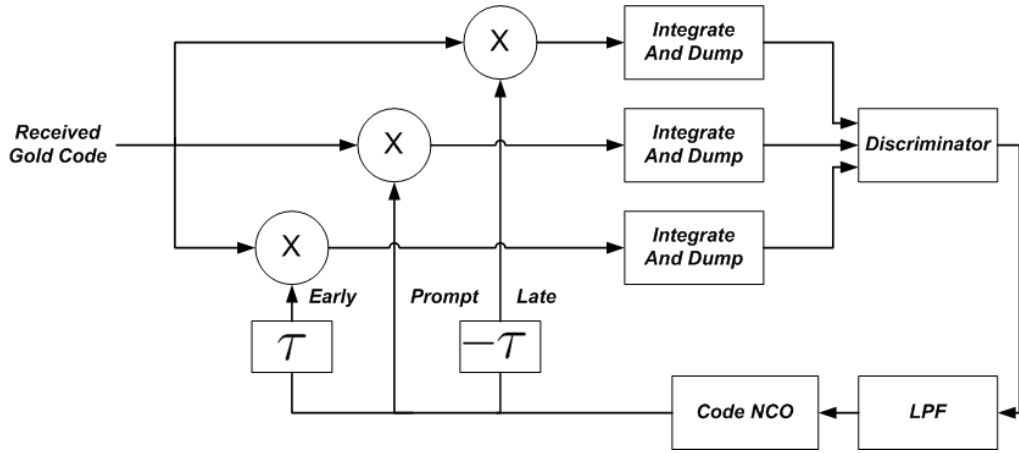


Figure 3.12: Delay Lock Loop

$$Late = A \sum_{n=1}^N G_i(n) \cdot G_{i,r}(n + \epsilon - \tau)$$

The sampled Gold code received from the i th satellite, $G_i(n)$, is mixed with the local replicas of itself. The prompt replica code, $G_{i,r}$, is offset from the phase of the received code by ϵ . The summations are effectively performing a cross-correlation between the two signals. Since the two signals are the same code, the output of each integrate and dump block can be expressed in terms of the autocorrelation function of the Gold code, given in Equation (3.17).

$$\begin{aligned}
 Early &= K \cdot R(\epsilon + \tau) \\
 Prompt &= K \cdot R(\epsilon) \\
 Late &= K \cdot R(\epsilon - \tau)
 \end{aligned} \tag{3.17}$$

The gain K is included to account for the amplitude of the received signal, sign of the data message bit, and the integration gain of the summation.

Figure 3.13 shows the values produced in each branch of the DLL superimposed over the autocorrelation triangle of the Gold code. When the misalignment ϵ is zero, the values produced in the Early and Late channels will match. The actual autocorrelation function observed in a receiver tends to be rounded at the center of the autocorrelation triangle because of the bandpass filtering in the RF front-end.

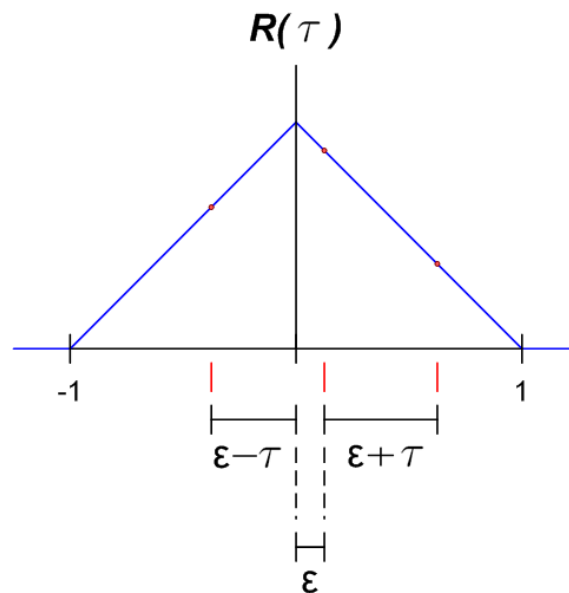


Figure 3.13: Autocorrelation Function of Gold Code

The discriminator uses the accumulated values in the early, prompt, and late channels to produce an error signal. Multiple discriminator algorithms exist to process the accumulated values. The functions can be divided into two categories. One category produces an error signal independent of the amplitude of the received signal, while the

error signal produced by the second category is dependent on the received signals amplitude [Ward, 1996b]. The error signal produced by the discriminator is passed through the loop filter. The output of the loop filter is then used to adjust the frequency of the code NCO to close the feedback loop.

3.4.4 Combined Costas and Delay Lock Loops

The previous discussions of Costas loops and Delay Locked Loops assumed that the Costas loop only operated on the carrier portion of the received signal and that the DLL only operated on the received Gold code. In reality, the received signal cannot be split perfectly into its code and carrier components. Instead, the local replicas maintained by the Costas loop and DLL are used to remove the carrier and Gold code modulation, respectively. The received signal is multiplied by the prompt code replica produced by the DLL in an attempt to remove the Gold code modulation. If the two codes are aligned, multiplying the two is equivalent to squaring the Gold code. Since the code is a series of ± 1 's, squaring the code thus removes the code. The product of this multiplication is then used by the Costas loop to track the carrier. Similarly, the received signal is multiplied by the in-phase and quadrature replicas of the Costas loop in order to remove the carrier modulation. The DLL then uses the resulting signals to track the Gold code.

The Costas and Delay Locked Loop operate on the output of a bank of correlators. The architecture of the correlators, Costas, and Delay Locked loop is shown in Figure 3.14. The outputs labeled as IP (in-phase prompt) and QP (quadrature prompt) are the products of multiplying the received signal by the prompt Gold code. These two signals are then used by the Costas loop, as discussed above. The other four signals are used by the DLL to control the replica Gold code. The expected value of the outputs

are shown in Equation (3.18), where ϵ is the phase error between the Gold codes, θ_{err} is the phase error between the carriers, and f_{err} is the frequency error between the carriers [Sayre, 2003].

$$\begin{aligned}
IE &= \frac{A}{\sqrt{2}} R(\epsilon + \tau) \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \cos(\pi f_{err} T + \theta_{err}) \\
IP &= \frac{A}{\sqrt{2}} R(\epsilon) \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \cos(\pi f_{err} T + \theta_{err}) \\
IL &= \frac{A}{\sqrt{2}} R(\epsilon - \tau) \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \cos(\pi f_{err} T + \theta_{err}) \\
QE &= \frac{A}{\sqrt{2}} R(\epsilon + \tau) \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \sin(\pi f_{err} T + \theta_{err}) \\
QP &= \frac{A}{\sqrt{2}} R(\epsilon) \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \sin(\pi f_{err} T + \theta_{err}) \\
QL &= \frac{A}{\sqrt{2}} R(\epsilon - \tau) \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \sin(\pi f_{err} T + \theta_{err})
\end{aligned} \tag{3.18}$$

The accumulated signals are used to produce error signals for the Costas and Delay Locked Loop. The error signals in each loop are generated by using nonlinear discriminator functions. The discriminator functions operate on the accumulated values to produce linear estimates of the carrier phase error θ_{err} and code delay error ϵ . The in-phase prompt (IP) and quadrature prompt (QP) values are used to produce an estimate of the carrier phase error θ_{err} by using a two quadrant arctangent discriminator, shown below in Equation (3.19).

$$\begin{aligned}
\theta &= \arctan\left(\frac{\cos(\theta)}{\sin(\theta)}\right) \\
\theta_{err} &\approx \arctan\left(\frac{\frac{A}{\sqrt{2}} R(\epsilon) \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \cos(\pi f_{err} T + \theta_{err})}{\frac{A}{\sqrt{2}} R(\epsilon) \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \sin(\pi f_{err} T + \theta_{err})}\right) \\
\theta_{err} &\approx \arctan\left(\frac{\cos(\pi f_{err} T + \theta_{err})}{\sin(\pi f_{err} T + \theta_{err})}\right)
\end{aligned} \tag{3.19}$$

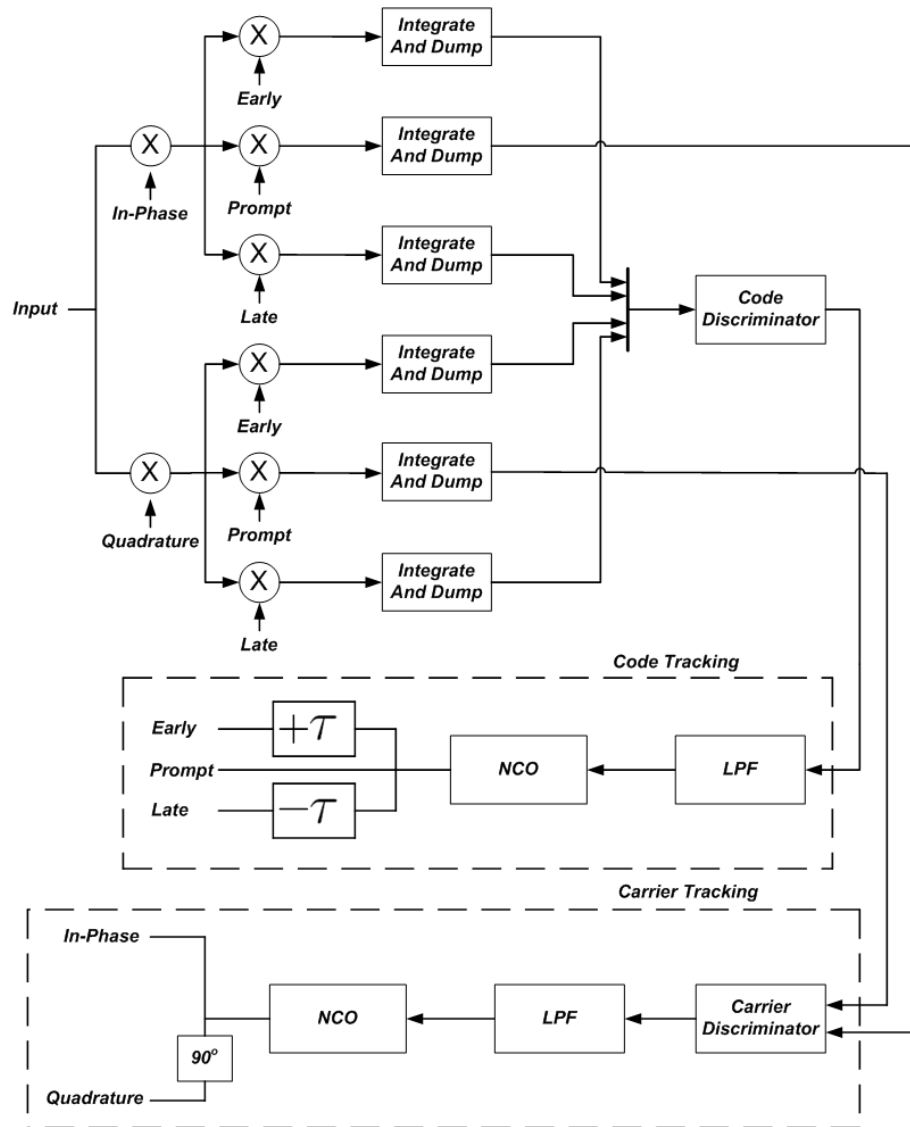


Figure 3.14: Complete Loop Architecture

The two quadrant arctangent discriminator is used for two primary reasons. The first reason is that it is immune to the 180° phase shifts caused by the data message. The second reason is that it performs well at varying signal to noise ratios. However, as a trade off for its performance, the arctangent discriminator has a high computational burden [Ward, 1996b].

The in-phase early (IE), in-phase late (IL), quadrature early (QE) and quadrature late (QL) signals are used by the code discriminator to produce an estimate of the code misalignment error ϵ , presented below in Equation (3.20).

$$\epsilon = \frac{\sum \sqrt{IE^2 + QE^2} - \sum \sqrt{IL^2 + QL^2}}{\sum \sqrt{IE^2 + QE^2} + \sum \sqrt{IL^2 + QL^2}} \quad (3.20)$$

The code discriminator is a normalized early-minus-late envelope detector. This type of discriminator performs well at varying C/No levels and its output is independent of the signal's amplitude [Ward, 1996b].

The outputs of the Costas and Code NCO's are used as feedback to the correlators to close the loop. If the initial code phase and carrier frequency are within the pull-in range of the two combined loops, the code and carrier NCO's will quickly become phase-locked with the received signal. Many factors can impede the lock-in process, such as low C/No levels, incorrect initialization, and improper choice of loop filters. Once the loops are locked, receiver dynamics, declining C/No levels, and jamming sources can cause the loops to unlock. When the tracking loops unlock, they are in a state of random walk and often times the acquisition process must be repeated to reacquire the signal.

3.5 Position Determination

The data message sent from each satellite is recovered by the tracking loops and is used to calculate the position of the satellites. In addition, the Delay Lock Loop's produce pseudorange measurements to each satellite. Once the positions of at least four satellite's are known and pseudorange measurements are produced, the receiver can determine its position. Each individual pseudorange measurement can be expressed as shown below in Equation (3.21).

$$\begin{aligned}\rho_i &= \|s_i - \mathbf{u}\| + ct_u \\ \rho_i &= \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + ct_u\end{aligned}\quad (3.21)$$

The above model ignores the errors caused by the ionosphere, troposphere, and satellite clock offset. The pseudorange ρ_i to the i th satellite is a function of the user's coordinates (x_u, y_u, z_u) and clock bias t_u .

Obtaining the user's position and clock bias requires solving the second order, non-linear equations shown in Equation (3.22).

$$\begin{aligned}\rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + ct_u \\ \rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + ct_u \\ &\vdots = \vdots \\ \rho_i &= \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + ct_u\end{aligned}\quad (3.22)$$

These equations can be solved by using analytical solutions [Kleusberg, 1999], iterative numerical techniques, or the extended Kalman filter [Kaplan et al., 1996].

3.5.1 Least Squares Solution

A popular way to solve for the user's position and clock bias is to use an iterative least squares approach. The least squares method involves first choosing a nominal estimation of the user's position and clock bias. The pseudorange formulas are then linearized about the nominal values. Solving the resulting linear system of equations produces corrections in the user's position and clock bias. These corrections are added to the initial estimates. The corrected estimates are then used to linearize the pseudorange equations in the same manner as the initial estimates. Corrections are again solved for to produce a further refined estimate of the user's position and clock bias. This iterative process continues until the estimated values approach constant values. Usually, the magnitude of the correction terms is computed and if they are below a threshold the process is terminated [Tsui, 2000].

The pseudorange measured to the i th satellite is expressed as shown in Equation (3.23).

$$\begin{aligned}\rho_i &= \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + ct_u \\ \rho_i &= f(x_u, y_u, z_u, t_u) \\ f(x_u, y_u, z_u, t_u) &= f(\hat{x}_u + \delta x_u, \hat{y}_u + \delta y_u, \hat{z}_u + \delta z_u, \hat{t}_u + \delta t_u)\end{aligned}\tag{3.23}$$

The pseudorange to a particular satellite is a nonlinear function of the user's position and clock bias. This function, of the user's position and clock bias, can be reformulated

as a function of a nominal trajectory $(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)$ plus error terms $(\delta x_u, \delta y_u, \delta z_u, \delta t_u)$ about the nominal trajectory [Kaplan et al., 1996].

The pseudoranges can then be approximated by a first order Taylor series about the nominal trajectory. Truncating the Taylor series at the first order produces a linear system of equations, shown below in Equation (3.24).

$$\rho_i \approx \hat{\rho}_i - \frac{x_i - \hat{x}_u}{\hat{r}_i} \delta x_u - \frac{y_i - \hat{y}_u}{\hat{r}_i} \delta y_u - \frac{z_i - \hat{z}_u}{\hat{r}_i} \delta z_u + c \delta t_u \quad (3.24)$$

where

$$\begin{aligned} \hat{\rho}_i &= \sqrt{(\hat{x}_i - x_u)^2 + (\hat{y}_i - y_u)^2 + (\hat{z}_i - z_u)^2} + c \hat{t}_u \\ \hat{r}_i &= \hat{\rho}_i - c \hat{t}_u \end{aligned}$$

The linearized pseudorange equations for the different satellites can be rearranged into a concise matrix formulation, presented below in Equation (3.25).

$$\begin{aligned} \rho_i - \hat{\rho}_i &\approx \frac{x_i - \hat{x}_u}{\hat{r}_i} \delta x_u + \frac{y_i - \hat{y}_u}{\hat{r}_i} \delta y_u + \frac{z_i - \hat{z}_u}{\hat{r}_i} \delta z_u - c \delta t_u \\ \delta \rho_i &\approx a_{x,i} \delta x_u + a_{y,i} \delta y_u + a_{z,i} \delta z_u - c \delta t_u \end{aligned}$$

$$\begin{bmatrix} \delta \rho_1 \\ \vdots \\ \delta \rho_i \end{bmatrix} \approx \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,i} & a_{y,i} & a_{z,i} & 1 \end{bmatrix} \begin{bmatrix} \delta x_u \\ \delta y_u \\ \delta z_u \\ \delta t_u \end{bmatrix} \quad (3.25)$$

$$\Delta \rho = \mathbf{H} \Delta x$$

Using the matrix interpretation of the linearized pseudorange measurements, the vector of correction terms Δx can be solved. In the case of four pseudorange measurements, this is accomplished by inverting the \mathbf{H} matrix. When more than four pseudorange measurements are used, the system of equations is solved using the Moore-Penrose pseudoinverse, as shown below in Equation (3.26) [Kaplan et al., 1996].

$$\begin{aligned}\Delta x &= \mathbf{H}^{-1}\Delta\rho, \text{ when } n = 4 \\ \Delta x &= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\Delta\rho, \text{ when } n > 4\end{aligned}\tag{3.26}$$

The vector of corrections is then added to the initial nominal trajectory to produce a second, refined estimate of the user's position and clock bias. These new estimates can then be used to produce more accurate estimates of position by repeating the process. The least squares approach is commonly used and can also produce an estimate of the accuracy of the position solution [Axelrad and Brown, 1996].

3.5.2 EKF Solution

The extended Kalman filter (EKF) is an alternate method to solve for the user's position and clock bias. The recursive nature of the EKF naturally lends itself to this task. The EKF formulation incorporates knowledge of the previous measurements into the current estimate of the user states. In contrast, the least squares solution is a point solution. The EKF method also allows the user's clock and platform dynamics to be modeled.

The EKF architecture generally used to calculate user position and clock bias differs from that used in other applications. The difference is that the EKF used in GPS tracks

the residuals of the states, not the states themselves. The states of the filter are errors in the user's coordinates and clock bias and estimates of the user's states are maintained outside of the filter. At each iteration of the filter, the estimated states are used to linearize the pseudorange equations. The state vector of the EKF is set to zero initially. The EKF then estimates the errors in the states using the pseudorange measurements and the estimated errors are then added to the states outside the filter. The state vector of EKF is then reset to zeros [Bullock et al., 2006].

The states of the EKF and the state transition matrix are shown below in Equation (3.27) for a stationary user.

$$\begin{bmatrix} \delta x_{k+1} \\ \delta y_{k+1} \\ \delta z_{k+1} \\ \delta t_{k+1} \\ \delta \dot{t}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta x_k \\ \delta y_k \\ \delta z_k \\ \delta t_k \\ \delta \dot{t}_k \end{bmatrix} \quad (3.27)$$

The states correspond to errors in the user's position, clock bias, and clock drift. Outside of the filter, estimates of the user's position, clock bias, and clock drift are maintained. The state error vector is estimated at each measurement epoch. The estimated error state vector is then added to the estimates of the user's states outside the filter and the error state vector is then reset to zero's at the beginning of the next epoch.

At each measurement epoch, the estimates of the user states $(\hat{x}_k^-, \hat{y}_k^-, \hat{z}_k^-, \hat{t}_k^-, \hat{t}_k^-)$ are used to linearize the observation matrix, shown in Equation (3.28).

$$H = \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 & 0 \\ a_{x,2} & a_{y,2} & a_{z,2} & 1 & 0 \\ \vdots & & & & \\ a_{x,i} & a_{y,i} & a_{z,i} & 1 & 0 \end{bmatrix} \quad (3.28)$$

In Equation (3.28), $a_{x,i}$, $a_{y,i}$, and $a_{z,i}$ are the components of a unit vector pointing from the user's estimated position to the i th satellite.

The measurements used by the filter are the range residuals, as shown below in Equation (3.29).

$$z_k = \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 & 0 \\ a_{x,2} & a_{y,2} & a_{z,2} & 1 & 0 \\ \vdots & & & & \\ a_{x,i} & a_{y,i} & a_{z,i} & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{\rho}_1 - \rho_1 \\ \hat{\rho}_2 - \rho_2 \\ \vdots \\ \hat{\rho}_i - \rho_i \end{bmatrix} \quad (3.29)$$

The range residuals are the difference between the predicted pseudoranges, based on the estimates of the user position $(\hat{x}_k^-, \hat{y}_k^-, \hat{z}_k^-)$ and clock bias \hat{t}_k^- , and the observed pseudoranges, as shown below in Equation (3.30).

$$\hat{\rho}_i = \sqrt{(\hat{x}_i - \hat{x}_k^-)^2 + (\hat{y}_i - \hat{y}_k^-)^2 + (\hat{z}_i - \hat{z}_k^-)^2} + c\hat{t}_k^- \quad (3.30)$$

Since the error state vector is set to all zero's at the beginning of each epoch, the measurement update is the single multiplication, given in Equation (3.31).

$$\begin{bmatrix} \delta x_k^+ \\ \delta y_k^+ \\ \delta z_k^+ \\ \delta t_k^+ \\ \delta \hat{t}_k^+ \end{bmatrix} = K_k \cdot z_k \quad (3.31)$$

The estimated error state vector is then added to the estimates of the user's states outside the filter, shown in Equation (3.32).

$$\begin{bmatrix} \hat{x}_k^+ \\ \hat{y}_k^+ \\ \hat{z}_k^+ \\ \hat{t}_k^+ \\ \hat{t}_k^- \end{bmatrix} = \begin{bmatrix} \hat{x}_k^- + \delta x_k^+ \\ \hat{y}_k^- + \delta y_k^+ \\ \hat{z}_k^- + \delta z_k^+ \\ \hat{t}_k^- + \delta t_k^+ \\ \hat{t}_k^- + \delta \hat{t}_k^+ \end{bmatrix} \quad (3.32)$$

The updated user state vector is then projected ahead for the next measurement epoch and the error state vector is reset to all zero's.

The measurement noise covariance matrix is a diagonal matrix with zeros above and below the principle diagonal [Bullock et al., 2006]. The entries on the diagonal depend on the variance of the individual pseudorange measurements, as shown in Equation

(3.33) below.

$$R = \begin{bmatrix} \sigma_{\rho_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{\rho_2}^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{\rho_i}^2 \end{bmatrix} \quad (3.33)$$

The process noise covariance is similarly a diagonal matrix with zeros off the principle diagonal. The terms in the process noise covariance matrix are used to account for unmodeled velocity variance, unmodeled clock variance, as well as to prevent the filter from ‘falling asleep’ [Bullock et al., 2006]. Although these parameters are often hand tuned to yield satisfactory tracking, a reasonable starting value for the terms related to the position states is approximately one hundredth of the variance of the pseudorange measurements [Negast, 2006]. The nominal variance of the pseudorange measurements is approximately 100 meters. For the clock terms, the values for a commercial-grade C/A code receiver integrated with tactical-grade sensors were used in this thesis as a starting point, as shown below in Figure (3.34) [Bullock et al., 2006].

$$Q \approx \begin{bmatrix} (\frac{\sigma_p}{10})^2 & 0 & 0 & 0 & 0 \\ 0 & (\frac{\sigma_p}{10})^2 & 0 & 0 & 0 \\ 0 & 0 & (\frac{\sigma_p}{10})^2 & 0 & 0 \\ 0 & 0 & 0 & 10^{-3} & 0 \\ 0 & 0 & 0 & 0 & 10^{-4} \end{bmatrix} \quad (3.34)$$

3.6 Conclusion

The general operations that a software GPS receiver performs were outlined in this chapter. Upon start up, a receiver first ascertains which satellites are in view. The receiver then determines the Doppler shift and PRN code phase associated with each satellite. Using this data, the receiver initializes the tracking loops. A Costas loop is used to track the carrier and a DLL is used to track the PRN code. The tracking loops provide the user with the navigation data message, used to calculate the satellite positions, and pseudorange measurements. It was then shown how a user can solve for his position using an iterative least squares algorithm or EKF.

The vulnerable parts of the receiver are the tracking loops, especially the carrier tracking loops. Tracking loops cannot operate during periods of signal loss and often lose lock during high dynamic maneuvers. Additionally, they are susceptible to intentional and unintentional jamming. The next chapters will explore vector tracking methods aimed at making the GPS receiver more robust.

CHAPTER 4

DEVELOPMENT OF KALMAN FILTER BASED TRACKING ALGORITHMS

4.1 Introduction

In this chapter, Kalman filter based tracking algorithms are introduced. The operation of the algorithms is explained and their potential advantages are elucidated. Later, in Chapter 5, the operation of the Kalman filter based algorithms is compared to the operation of traditional methods.

4.2 Tracking Loops as Kalman Filters

The tracking loops described in Chapter 3.4 operate well in environments with low user dynamics and high C/No levels. However, tracking loops have inherent flaws. They use fixed bandwidth filters that do not adapt to varying C/No levels and user dynamics. The measurements made by the discriminator are all weighted equally. Measurements made during periods of high C/No are weighted equally with those made during times of low C/No. Tracking loops also have a limited ability track changes in the user's dynamics. The order of the loop filter dictates the dynamics the loop can track with no steady state error. When designing the loop filter, the designer is faced with a trade off. On the one hand, increasing the bandwidth of the filter allows the loop to better track the user dynamics. However, wider bandwidths make the loop more susceptible to noise and jamming [Jwo, 2001].

The Kalman filter has the capacity to overcome the drawbacks of the traditional tracking loop. The Kalman filter is, in essence, a filter with time varying gains. The

gains vary with changing measurement noise statistics and process noise statistics. The measurement noise statistics change with C/No levels and jamming. The process noise statistics change with user dynamics. Provided with the process and measurement noise covariance matrices, the Kalman filter can optimally separate signal from noise. The tasks of tracking the carrier and code can be combined in one Kalman filter, which replaces the two tracking loops in each channel with a single Kalman filter per satellite [Grewal and Andrews, 1993].

The states of the Kalman filter and the state transition matrix are shown below in Equation (4.1).

$$\begin{bmatrix} \theta_{c,k+1} \\ f_{c,k+1} \\ \dot{f}_{c,k+1} \\ \theta_{G,k+1} \\ f_{G,k+1} \\ C_{G,k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0 \\ 0 & K_1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{c,k} \\ f_{c,k} \\ \dot{f}_{c,k} \\ \theta_{G,k} \\ f_{G,k} \\ C_{G,k} \end{bmatrix} \quad (4.1)$$

The first state of the filter $\theta_{c,k+1}$ corresponds to the phase of the IF carrier. The derivative of the phase of the carrier is the IF frequency, which is the second state, $f_{c,k+1}$. The frequency of the carrier is modeled as varying linearly with time. The derivative of the carrier's frequency is modeled as a slowly varying bias in the third state, $\dot{f}_{c,k+1}$. The value of the third state can be determined by using the satellite ephemeris or by curve fitting. The fourth state is the phase of the local Gold code, $\theta_{G,k+1}$. The frequency that the receiver increments through the local Gold code is the fifth state, $f_{G,k+1}$. The frequency of the code is a combination of a nominal frequency $C_{G,k+1}$ plus a term caused

by the Doppler shift on the carrier. The term caused by the Doppler shift is modeled as the carrier frequency multiplied by a constant, K_1 . This is the equivalent of using the Doppler frequency from a Costas loop to aid the code tracking loop.

The Kalman filter used to track the carrier and code is similar to the filter used to track the user's position. The measurements provided to the filter are not measurements of the states directly, but rather the errors between certain states and those of the received signal. The measurements are therefore residuals. The filter has access to three residuals, which are measurements of the error in the phase and frequency of the carrier, and the error in the phase of the local Gold code. At each measurement epoch, the residuals and Kalman gain are used to compute corrections to the states. The states after the measurement update are then the propagated states plus the vector of corrections.

The signals used to generate the measurements for the filter are summarized in Equation (4.2) [Sayre, 2003].

$$\begin{aligned}
IE_k &= \sqrt{\frac{2ST}{N_o}} R(\epsilon + \tau) D_k \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \cos(\pi f_{err} T + \theta_{err}) + \eta_{IE,k} \\
IP &= \sqrt{\frac{2ST}{N_o}} R(\epsilon) D_k \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \cos(\pi f_{err} T + \theta_{err}) + \eta_{IP,k} \\
IL &= \sqrt{\frac{2ST}{N_o}} R(\epsilon - \tau) D_k \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \cos(\pi f_{err} T + \theta_{err}) + \eta_{IL,k} \\
QE &= \sqrt{\frac{2ST}{N_o}} R(\epsilon + \tau) D_k \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \sin(\pi f_{err} T + \theta_{err}) + \eta_{QE,k} \\
QP &= \sqrt{\frac{2ST}{N_o}} R(\epsilon) D_k \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \sin(\pi f_{err} T + \theta_{err}) + \eta_{QP,k} \\
QL &= \sqrt{\frac{2ST}{N_o}} R(\epsilon - \tau) D_k \frac{\sin(\pi f_{err} T)}{\pi f_{err} T} \sin(\pi f_{err} T + \theta_{err}) + \eta_{QL,k} \\
\eta &= N(0, 1)
\end{aligned} \tag{4.2}$$

In Equation (4.2), η represents the noise samples in the output of each integrate and dump block. The noise samples are assumed to be Gaussian, white, and independent of the noise samples in the other integrate and dump functions [Cahn et al., 1977]. The error in the alignment of the prompt gold code replica and the received code is represented by ϵ . The frequency error and phase error between the received and replicated carriers are denoted by f_{err} and θ_{err} , respectively.

Measurements of the three errors in the local signals are generated by using nonlinear discriminator functions. A normalized early-minus-late discriminator is used to produce an estimate of the error in alignment between the received and replica Gold codes, as shown in Equation (4.3).

$$\epsilon \approx \frac{\sum \sqrt{IE^2 + QE^2} - \sum \sqrt{IL^2 + QL^2}}{\sum \sqrt{IE^2 + QE^2} + \sum \sqrt{IL^2 + QL^2}} \quad (4.3)$$

A two quadrant arctangent discriminator is used to measure the phase error θ_{err} , as shown below in Equation (4.4).

$$\theta_{err} \approx \arctan \left(\frac{QP}{IP} \right) \quad (4.4)$$

A frequency discriminator generates a measurement of the frequency error f_{err} by using data collected over two adjacent integrate and dump intervals, as shown below in Equation (4.5).

$$\begin{aligned} dot &= IP_{t_1} \cdot IP_{t_2} + QP_{t_1} \cdot QP_{t_2} \\ cross &= IP_{t_1} \cdot QP_{t_2} - IP_{t_2} \cdot QP_{t_1} \end{aligned}$$

$$f_{err} \approx \frac{\arctan(\frac{cross}{dot})}{360(t_2 - t_1)} \quad (4.5)$$

These intervals cannot straddle a data bit transition. The first interval is denoted by t_1 and the second interval is denoted by t_2 . The in-phase and quadrature signals are used to form dot and cross products. These products are then used to estimate the frequency error [Ward, 1996b].

All of the discriminators contain inherit nonlinearities. The output of the normalized early-minus-late discriminator saturates at a misalignment ϵ corresponding to half a chip of the Gold code. Figure 4.1 shows output of the code discriminator versus actual code misalignment.

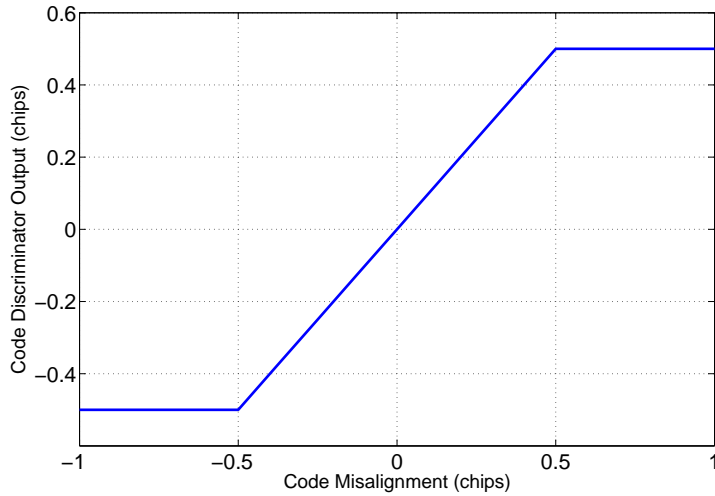


Figure 4.1: Output of Code Discriminator

The two quadrant arctangent discriminator only produces unambiguous estimates of the phase error (θ_{err}) for values of the error between $\pm 90^\circ$. Figure 4.2 displays the output of the phase discriminator for different phase errors. The frequency discriminator

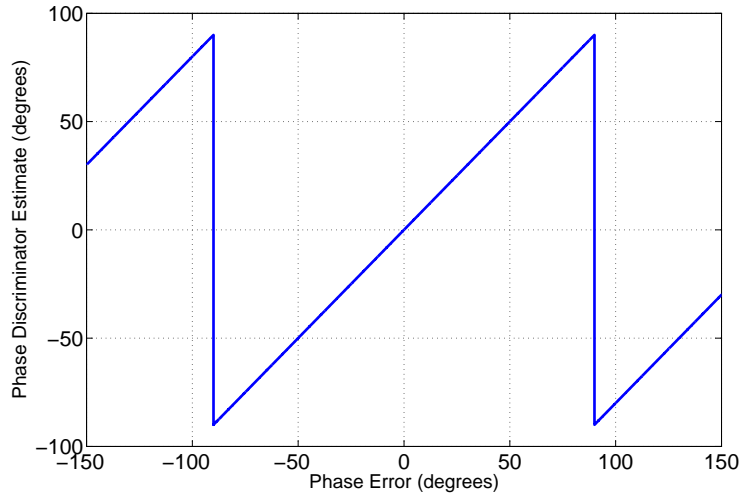


Figure 4.2: Output of Arctangent Phase Discriminator

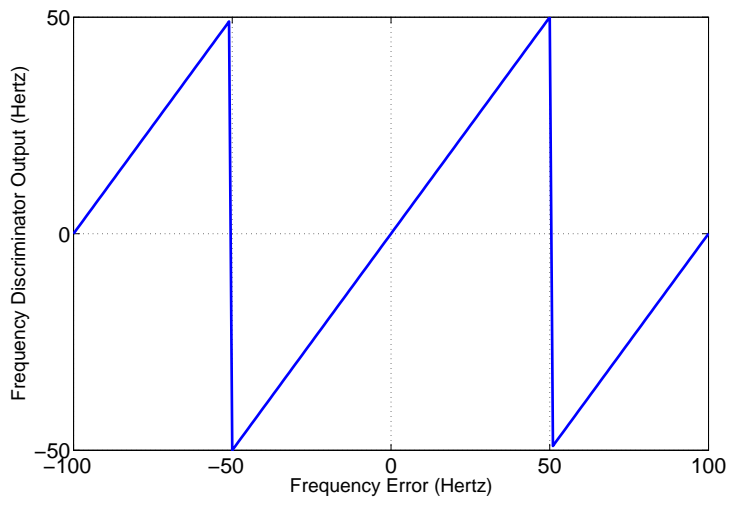


Figure 4.3: Output of Frequency Discriminator

produces the same type of wrap-around error as the phase discriminator. The magnitude of the frequency error at which the wrap-around occurs is dependent on the integration period of t_1 and t_2 . A ten millisecond integration period yields the discriminator output shown in Figure 4.3. Decreasing the integration time increases the unambiguous range of the frequency discriminator, but it also increases the amount of noise in the measurement [Ward, 1996b].

The measurement noise covariance matrix contains the statistics of the measurement noise, shown below in Equation (4.6).

$$\begin{bmatrix} E[\theta_{err}^2] & E[\theta_{err} \cdot f_{err}] & E[\theta_{err} \cdot \epsilon] \\ E[\theta_{err} \cdot f_{err}] & E[f_{err}^2] & E[\epsilon \cdot f_{err}] \\ E[\theta_{err} \cdot \epsilon] & E[\epsilon \cdot f_{err}] & E[\epsilon^2] \end{bmatrix} \quad (4.6)$$

The off-diagonal terms of the measurement noise covariance matrix are assumed to be zero. The diagonal terms of the matrix are computed empirically and can be implemented as a look-up table for varying C/No levels. The process noise covariance is used to accommodate unmodeled errors and is hand-tuned to yield satisfactory tracking.

4.3 Vector Delay Lock Loop

The previous method of tracking replaces the Costas and delay lock loops with a single Kalman filter with the different channels still operating independently of each other. In the Vector Delay Lock Loop (VDLL) approach, the tracking of the different PRN sequences is done collectively. The code phases of the received PRN sequences can be predicted using the satellite ephemeris data, the user's position, and the user's

clock bias. The VDLL operates by using an Extended Kalman filter to estimate the user's position, clock bias, and clock drift, which allows the code phases of the PRN sequences can then be predicted. The NCO's that produce the replica Gold code's for each satellite are initialized using the predicted PRN code phases. At the end of each integrate and dump period, the output of the correlators is used to generate an estimate of the phase error between the received and replica codes. The states of the Extended Kalman filter are then updated using the residuals produced by using the correlator outputs. Figures 4.4 and 4.5 show a flow chart of the VDLL and it's general architecture, respectively.

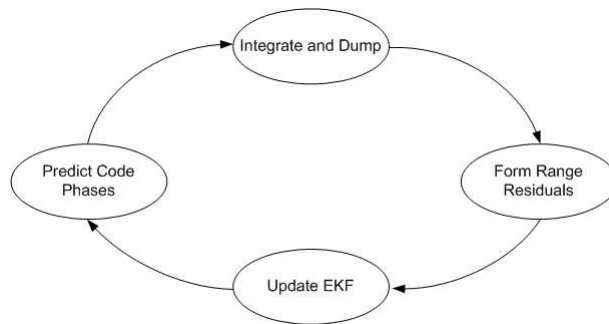


Figure 4.4: Vector Delay Lock Loop Flow Chart

The vector delay lock loop has several advantages over traditional delay lock loops. The first is the ability of the VDLL to weight measurements. Measurements made during periods of high C/No levels contain less noise and can be weighted more heavily. Conversely, measurements made during times of low C/No levels, or satellite blockage, can be ignored. Another advantage of the VDLL is that the tracking of weak signals can be facilitated by using information from stronger signals. This is because the PRN tracking is based on the user's position, not a single filter operating on a weak signal.

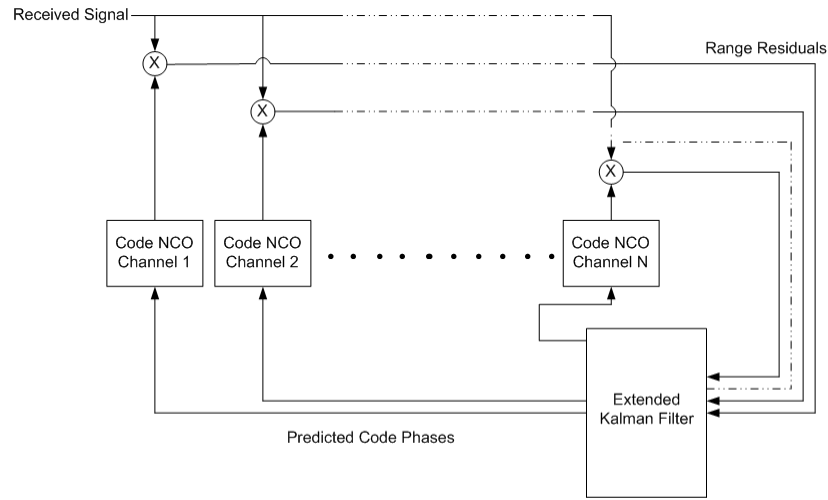


Figure 4.5: General Vector Delay Lock Loop Architecture

The user's position can be determined by using information from the stronger signals present. The weak signals can then be accurately predicted based on the states of the EKF [Spilker, Jr., 1996a].

The VDLL designed in this thesis is distinct from that described in [Spilker, Jr., 1996a]. The VDLL developed in this research is based on discriminator functions and operates by predicting the arrival time of the navigation data bits. This is equivalent to predicting the code phases of the received Gold codes. Figure 4.6 shows the navigation messages being received from four different satellites. The satellites are arranged from top to bottom in order of their distance from the user. With the exception of the individual satellite clock bias', the navigation messages are transmitted simultaneously by the different satellites. The effects of the satellite clock errors can be virtually removed by using the satellite clock correction polynomial and its coefficients, broadcast by the satellites. It will be assumed that the satellite clock errors have been corrected. Initially, only the

relative distances of the satellites can be observed. The user can only determine which satellite is closest, and how much further away the other satellites are. The difference in the arrival time of the bits from the first and second closest satellites is labeled as $\Delta t_{1,2}$. The relative arrival times between the other satellites are similarly numbered. Since sampled data is used, the estimated arrival times of the different navigation bits are indexes that point to the estimated first sample of the received bits.

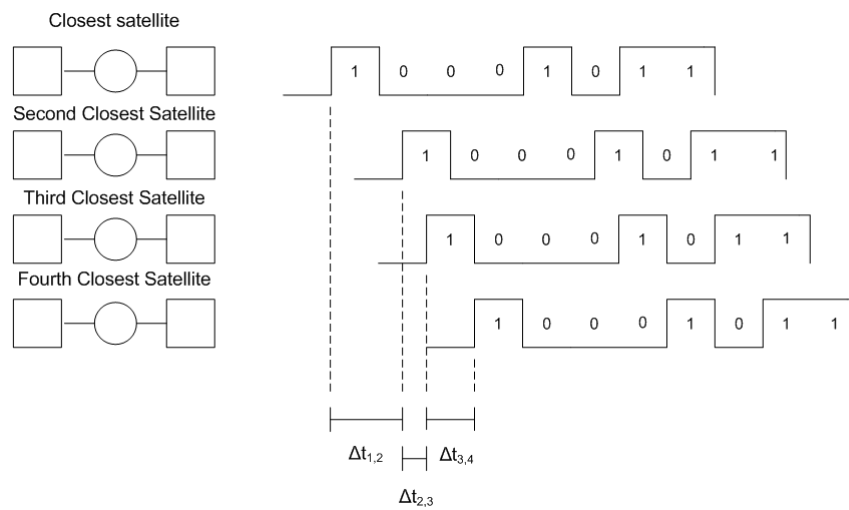


Figure 4.6: Arrival Time of Satellite Data Bits

An iterative least squares technique is initially used to determine the user's position and clock bias. These results are used to initialize the EKF. Equation (4.7) shows how

the arrival times are converted into pseudorange measurements.

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} = \begin{bmatrix} 0 \\ c\Delta t_{1,2} \\ c(\Delta t_{1,2} + \Delta t_{2,3}) \\ c(\Delta t_{1,2} + \Delta t_{2,3} + \Delta t_{3,4}) \end{bmatrix} \quad (4.7)$$

Using this method, the clock bias that is calculated is the distance from the user to the closest satellite divided by the speed of light. Using the clock bias, the sample index that corresponds to when the satellites broadcast their signals can be determined. Figure 4.7 shows this operation graphically.

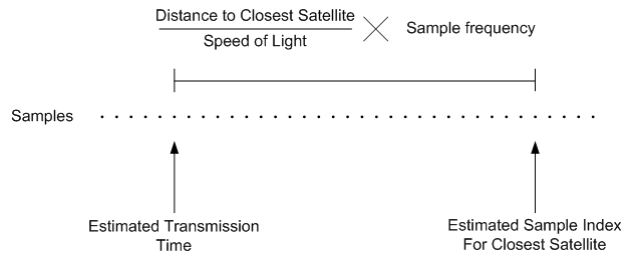


Figure 4.7: Determination of Transmission Index

The user can propagate the index corresponding to the transmission time of the satellites forward to the next bit transmission. This is accomplished by adding the number of samples corresponding to 20 milliseconds to the original transmission index. The user can then predict the arrival indexes of the data bits for the different satellites. The prediction is based on the satellite's position, determined from the broadcast ephemeris, and the user's estimated position. The user's estimated position is provided by propagating the states of the Extended Kalman filter forward 20 milliseconds. The predicted

arrival indexes are used as the starting points for the integrate and dump functions. Note that the PRN code phases are set to zero at the start of the integrate and dump intervals. This is because the data bit transitions are synchronous with the start of the C/A code. The satellite signals are then integrated during the period of the data bits. At the end of each integration period, the accumulated signals are used to generate estimates of the error between the predicted arrival indexes and the actual arrival indexes.

The errors between the predicted and actual arrival indexes are used as range residuals to update the Extended Kalman filter. Equation (4.8) shows the states and discrete state transition matrix used in a generic navigation EKF.

$$\begin{bmatrix} \delta x_{k+1} \\ \delta y_{k+1} \\ \delta z_{k+1} \\ \delta \dot{x}_{k+1} \\ \delta \dot{y}_{k+1} \\ \delta \dot{z}_{k+1} \\ \delta t_{k+1} \\ \delta \dot{t}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta x_k \\ \delta y_k \\ \delta z_k \\ \delta \dot{x}_k \\ \delta \dot{y}_k \\ \delta \dot{z}_k \\ \delta t_k \\ \delta \dot{t}_k \end{bmatrix} \quad (4.8)$$

The filter tracks in errors, as described in Chapter 3. The only difference between this filter and the one explained in Chapter 3 is the inclusion of velocity states. The filter is nonlinear because of the relationship between the states of the filter and the measurements. The observation matrix is linearized about the estimated state vector,

as shown in Equation (4.9).

$$H = \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 0 & 0 & 0 & 1 & 0 \\ a_{x,2} & a_{y,2} & a_{z,2} & 0 & 0 & 0 & 1 & 0 \\ \vdots & & & & & & & \\ a_{x,i} & a_{y,i} & a_{z,i} & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.9)$$

The Kalman gain is computed using the linearized measurement matrix. The process and measurement noise covariance matrices are determined in the same manner as described in Chapter 3.

The different carrier signals are still tracked independently in each channel. The Kalman filter architecture described in Section 1 of this chapter is used in each channel for carrier tracking. The states of the filter that correspond to the code are omitted.

4.4 Conclusion

This chapter has provided algorithms which can be used to replace tracking loops with Kalman filters. The vector tracking methods explained in this chapter are distinct from those described in published literature. The algorithms described in Section 1 replace the individual code and carrier tracking loops with a single Kalman filter. The Kalman filter has the ability to weight the measurements provided by the discriminators according to their variances. The Vector Delay Lock Loop presented in Section 4.3 goes further and performs the tasks of tracking the PRN code phases and estimating the user's position together. The VDLL has the potential to perform during intermittent

periods of satellite blockage and to improve the tracking of weak signals. This is accomplished by predicting the code phases of the PRN sequences based on the estimated user's position and clock bias. The estimated user's position and clock bias are provided by an Extended Kalman filter. Residuals are produced using the accumulated signals from integrate and dump functions, which are used as measurements for updating the EKF. The VDLL outlined in this chapter is different from the methods presented in the referenced literature. The VDLL used in this thesis is based on normalized discriminator functions, which eliminate the need to estimate the received signals' amplitude, and operates by predicting the arrival times of the navigation message data bits. The next chapter will present experimental results from a GPS constellation simulator that demonstrate the ability of the VDLL to operate during periods of satellite outage.

CHAPTER 5

VALIDATION AND PERFORMANCE OF KALMAN FILTER BASED ALGORITHMS

In this chapter, the VDLL algorithms are shown to function by using data collected from a Spirent GPS constellation simulator. The VDLL is shown to operate at nominal C/No levels. The ability of the VDLL to operate during periods of satellite blockage is also investigated. For comparison, the results of using traditional tracking loops are presented. The results presented in this chapter offer proof of concept for the VDLL and its ability to operate during satellite blockages.

In Chapter 4, tracking algorithms based on the Kalman filter were introduced. The Vector Delay Lock Loop is the most sophisticated of the algorithms in Chapter 4. In a VDLL, the tracking of the PRN code phases in the different channels is done by a single Extended Kalman filter. The EKF maintains estimates of the user's position and velocity in the Earth-Centered Earth-Fixed coordinate frame. The estimates of the EKF are used to predict the received PRN code phases and the residuals produced by the integrate and dump functions are used as measurements by the EKF. The residuals serve to correct errors in the states of the EKF.

5.1 Simulation Setup

The Spirent GPS simulator generates the received GPS signals based on user profiles. The profile used in this chapter is a stationary user located in Huntsville, AL. The simulator produces the RF signals that a user would receive based on the profile and

was used as the input to the NordNav RF front-end. The NordNav front-end samples the simulator signals and stores the samples for post processing.

An IMU input was added for the simulation. It was assumed that the IMU provides acceleration data in the ECEF coordinate frame at 50 Hz, synchronous with updates of the EKF. The IMU measurements are modeled as inputs to the velocity states of the EKF. The simulated IMU provides measurements of the same quality as a Crossbow IMU400CD [Crossbow Technology, 2006].

For the test setup, the satellite signals are received at a C/No ratio of 40 dB/Hz. Signals for satellites 16, 5, 3, and 12 were generated. All the satellite signals are present for the first 20 seconds of the simulation at 40 dB/Hz. At approximately 25 seconds into the simulation, the signals for satellites 16, 5, and 12 are turned off. This simulates a blockage of the signals from the respective satellites. The signals remain blocked for approximately 15 seconds. After wards, the signals are returned to their previous C/No levels.

The process noise covariance matrix used for the carrier tracking Kalman filters is shown below in Equation (5.1).

$$Q = \begin{bmatrix} 0.0100 & 0 & 0 \\ 0 & 0.2000 & 0 \\ 0 & 0 & 0.003 \end{bmatrix} \quad (5.1)$$

The carrier tracking Kalman filter uses units of radians, radians per second, and radians per second squared for the phase, frequency, and frequency derivative states, respectively. The noise covariance matrix for the carrier tracking filter is shown below in Equation

(5.2).

$$R = \begin{bmatrix} 0.0042 & 0 \\ 0 & 35.1508 \end{bmatrix} \quad (5.2)$$

The measurements used by the filter are in units of radians and radians per second.

The process noise covariance matrix used for the Extended Kalman filter is shown below in Equation (5.3).

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0139 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0139 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0139 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

The measurement noise covariance matrix for the EKF is shown below in Equation (5.4).

$$R = \begin{bmatrix} 75.6142 & 0 & 0 & 0 \\ 0 & 75.6142 & 0 & 0 \\ 0 & 0 & 75.6142 & 0 \\ 0 & 0 & 0 & 75.6142 \end{bmatrix} \quad (5.4)$$

The measurements used by the EKF are in units of meters

5.2 Code and Carrier Tracking Results

Figure 5.1 shows the output of the in-phase prompt correlator for satellite 16 using the VDLL algorithm. When the carrier and PRN code are being tracked, the in-phase prompt value is the sign of the navigation data bit multiplied by an integration gain and the amplitude of the received signal. The carrier and code are being tracked for the first 26 seconds. At approximately 27 seconds into the simulation, the satellite signal is turned off. At this point, the value of the in-phase prompt correlator drops below a predefined threshold value of 8000. Once the in-phase prompt value decreases below the threshold, the diagonal elements of the measurement noise covariance matrix for the carrier tracking Kalman filter are inflated. The carrier tracking Kalman filter then begins to ignore the measurements produced by the discriminator functions. Similarly, the measurement noise covariance matrix for the navigation EKF is adjusted. The diagonal element representing the variance of the PRN code phase measurement is inflated. A value of 10^{36} is used to inflate the noise covariance matrices and make the Kalman filters effectively ignore measurements. At approximately 39.3 seconds, the satellite signal is turned back on. In figure 5.1, the in-phase prompt signal is seen as increasing above the threshold at this time. The estimates of the carrier frequency and code phase are accurate enough that the signal exceeds the threshold. The estimates maintained by the Kalman filters allows the signal to be reacquired nearly instantaneously. Only after the signal exceeds the threshold are the noise covariance matrices reset to their pre-blockage values. After the signal exceeds the threshold the Kalman filter begins to weight measurements again.

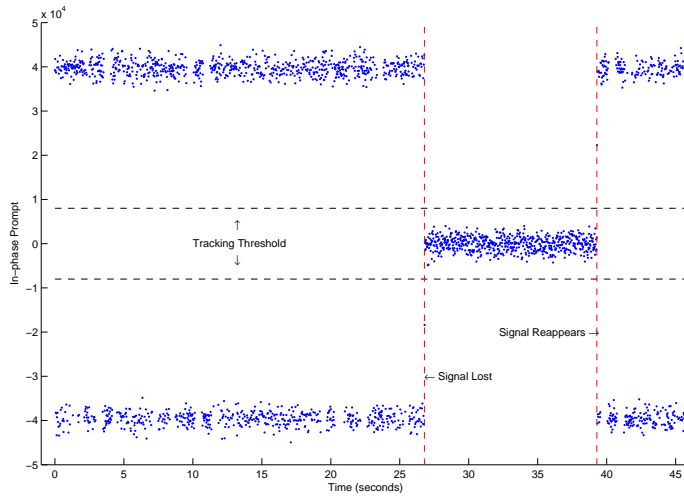


Figure 5.1: In-Phase Prompt Output for SV 16 by VDLL Algorithm

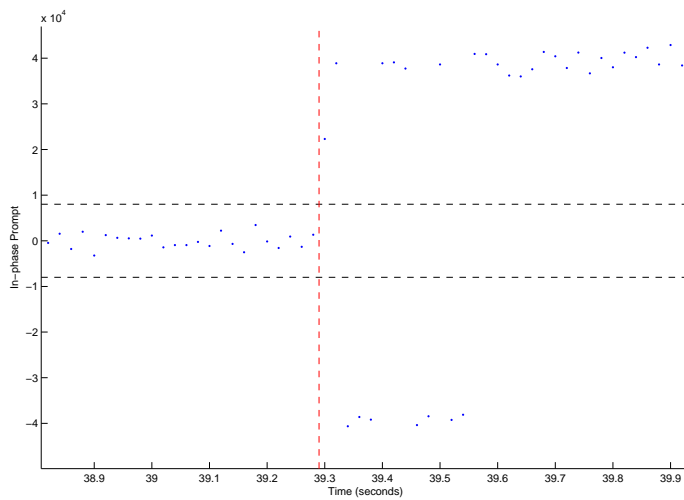


Figure 5.2: In-Phase Prompt Output for SV 16 by VDLL Algorithm

Figure 5.3 shows the Doppler frequency estimates produced by the Kalman filter based algorithm and a traditional tracking loop. The traditional loop uses an identical integration time of 20 milliseconds and has a noise bandwidth of 10 Hz. The value of the third state of the carrier tracking Kalman filter, corresponding to the rate of change in the carrier frequency, is estimated by the filter from the data prior to the outage. During the satellite outage, the traditional loop is in a state of random walk. The pull-in range of the traditional loop is approximately 5 Hz. The frequency error of the traditional loop exceeds its pull-in range before the satellite reappears. The Kalman filter based carrier tracking algorithm ignores measurements during the satellite blockage and propagates forward its estimates of the carrier frequency. The rate of change in the Doppler frequency during the outage is determined by the carrier tracking filter's estimation of the rate of change using the data prior to the outage. When the satellite signal returns, the Kalman filter's estimate of the Doppler frequency is within a few Hertz.

Figure 5.4 shows the output of the code phase discriminator for the VDLL algorithm and a traditional DLL. The DLL uses an integration time of 20 milliseconds and a noise bandwidth of 1 Hertz. The variance of the discriminator is seen to increase dramatically during the satellite outage. When the signal returns, the estimate of the PRN code phase maintained by the VDLL is close to the actual PRN code phase. The code discriminator only produces accurate estimates of the phase error when the codes are aligned to within approximately half a chip, or about 146 meters. Outside that range, the accumulated signals used by the discriminators are dominated by noise. The traditional DLL's estimate of the PRN code phase is not within the linear region of the code discriminator. The traditional methods therefore fail to reacquire the satellite signal.

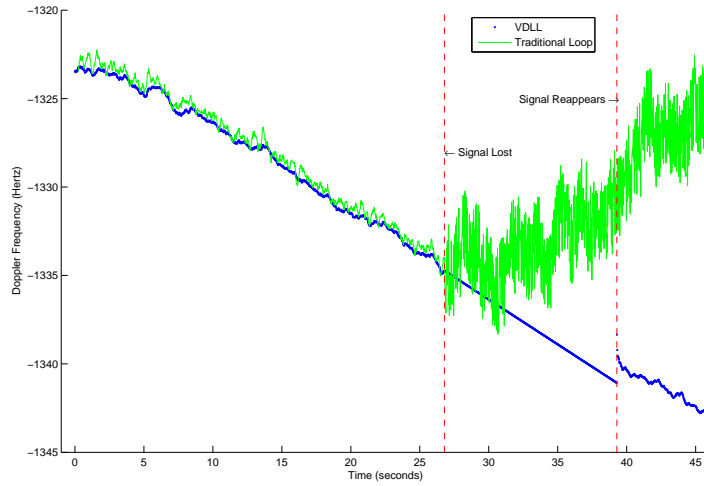


Figure 5.3: Carrier Tracking for SV 16

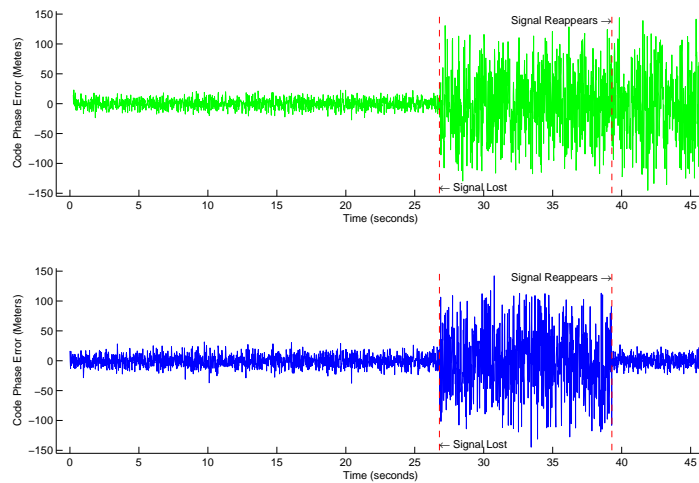


Figure 5.4: PRN Code Phase Error Measurements for SV 16 by the Traditional Delay Lock Loop (top) and the Vector Delay Lock Loop (bottom)

Figure 5.5 shows the output of the in-phase prompt correlator for the traditional Costas loop. The VDLL's code phase estimate is accurate enough that the code discriminator produces accurate estimates as soon as the signal returns. The variance of the output of the code discriminator returns to its pre-blockage value for the VDLL.

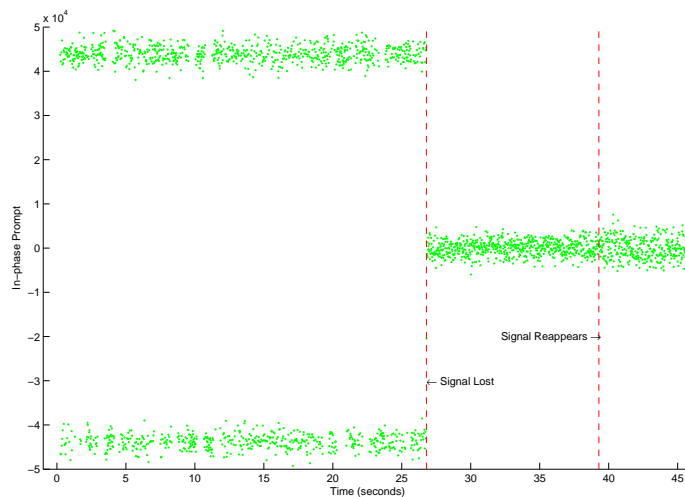


Figure 5.5: In-Phase Prompt Output for SV 16 by Costas Loop

Figure 5.6 shows the output of the in-phase prompt correlator for satellite 5. At approximately 27 seconds into the simulation, the signal from satellite five is lost. The value produced by the in-phase prompt correlator drops below the tracking threshold and the noise covariance matrices are augmented as discussed previously. The signal for satellite 5 returns at about 36.8 seconds. Figure 5.7 shows the signal returning and exceeding the tracking threshold. Again, the estimates of the carrier frequency and code phase were maintained to a sufficient accuracy so as to allow the near instantaneous reacquisition of the signals.

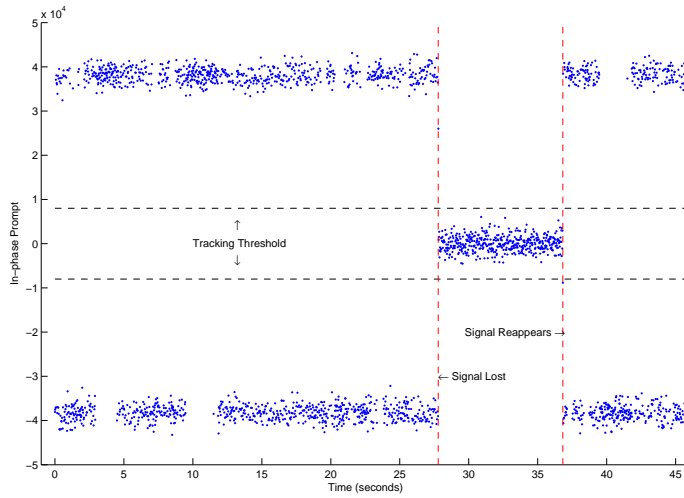


Figure 5.6: In-Phase Prompt Output for SV 5 by VDLL Algorithm

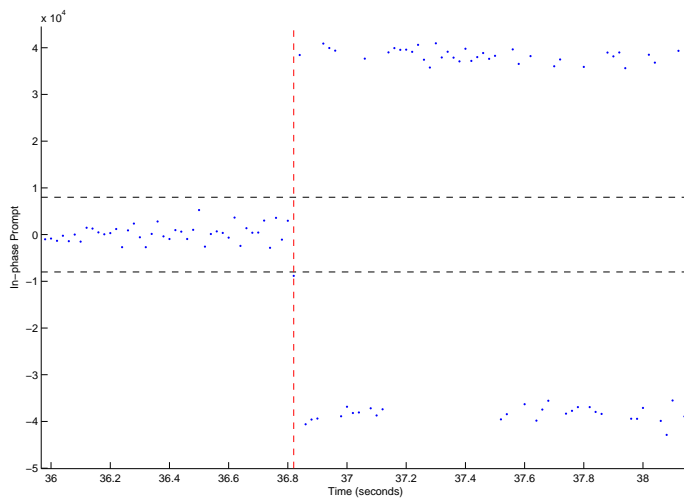


Figure 5.7: In-Phase Prompt Output for SV 5 by VDLL Algorithm

Figure 5.8 shows the Doppler estimates maintained by the traditional Costas loop and the Kalman filter based algorithms. The Kalman filter based method begins ignoring measurements when the in-phase prompt value drops below the threshold. The Costas loop is in a state of random walk during the outage and never recovers. The estimate provided by the Kalman filter method is close enough to the correct value so that when the signal reappears, it exceeds the tracking threshold. Once the threshold is exceeded, the noise covariance matrices are reset to their prior values. The Kalman filter then begins to weight the phase and frequency measurements and track the carrier again.

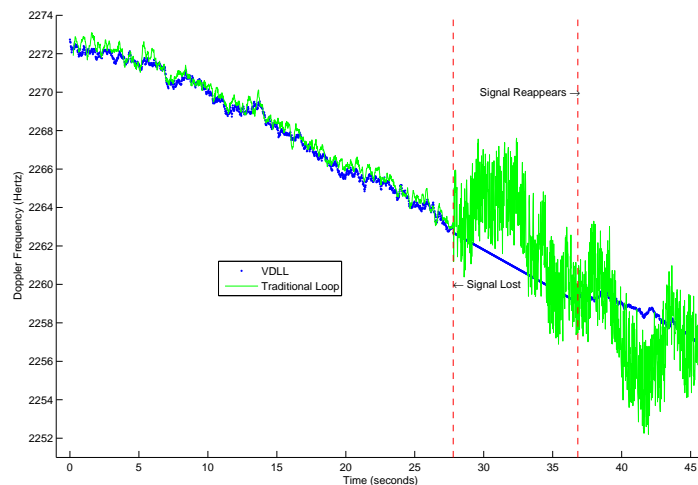


Figure 5.8: Carrier Tracking for SV 5

Figure 5.9 shows the output of the code phase discriminators for the traditional DLL and the VDLL. The code phase estimate of the DLL again diverges during the outage. When the signal returns, the code phase estimate of the DLL is not within half a chip of the true code phase. However, when the signal returns for the case of the VDLL, the

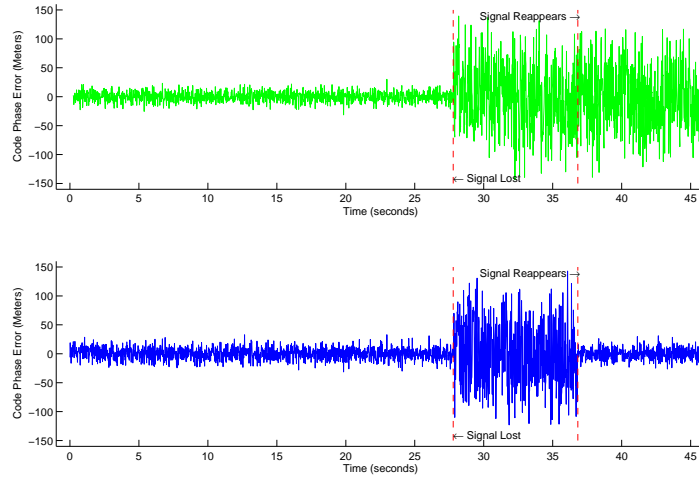


Figure 5.9: PRN Code Phase Error Measurements for SV 5 by the Traditional Delay Lock Loop (top) and the Vector Delay Lock Loop (bottom)

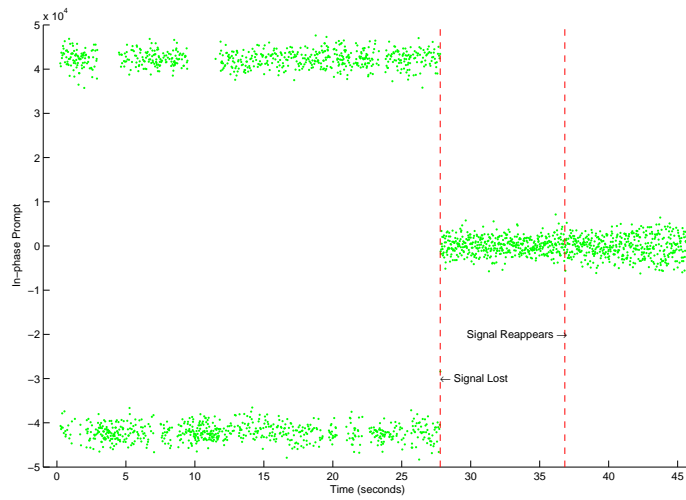


Figure 5.10: In-Phase Prompt Output for SV 5 by Costas Loop

code phase estimate is within the linear region of the discriminator. Figure 5.10 shows the in-phase prompt output for the traditional Costas loop. The traditional loops fail to reacquire the signal after the outage again.

Figure 5.11 shows the output of the in-phase prompt correlator for satellite 3. Satellite 3 remained in view during the entire simulation. The Kalman filter algorithms successfully tracked the signal for the duration of simulation. Figure 5.12 shows the Doppler estimates produced by the traditional and Kalman filter based methods. Both methods maintain lock with the received satellite signal. Figure 5.13 displays the code tracking performance of the two different approaches. Again, both work successfully for the simulation.

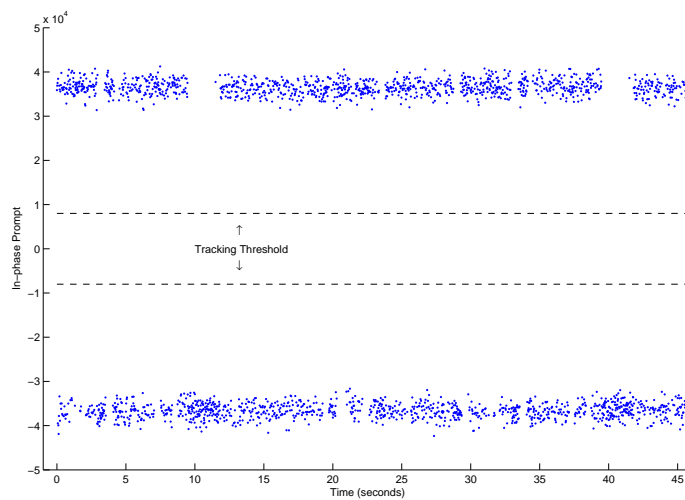


Figure 5.11: In-Phase Prompt Output for SV 3 by VDLL Algorithm

Figures 5.14 and 5.15 show the in-phase prompt correlator output for satellite 12. The signal is blocked at approximately 26 seconds into the simulation. As before, when

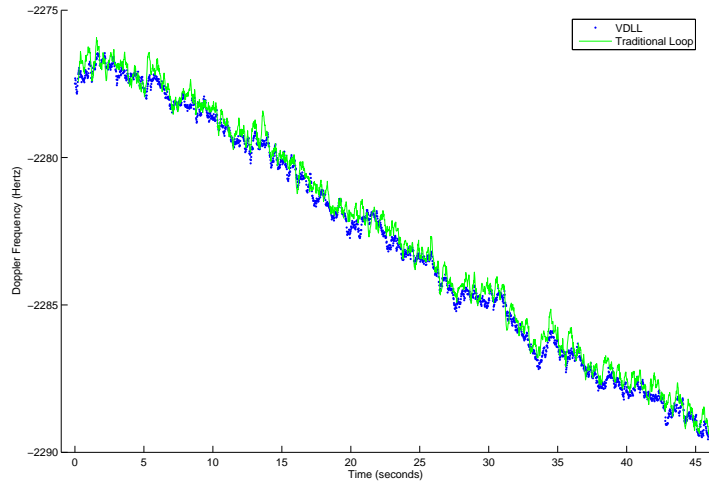


Figure 5.12: Carrier Tracking for SV 3

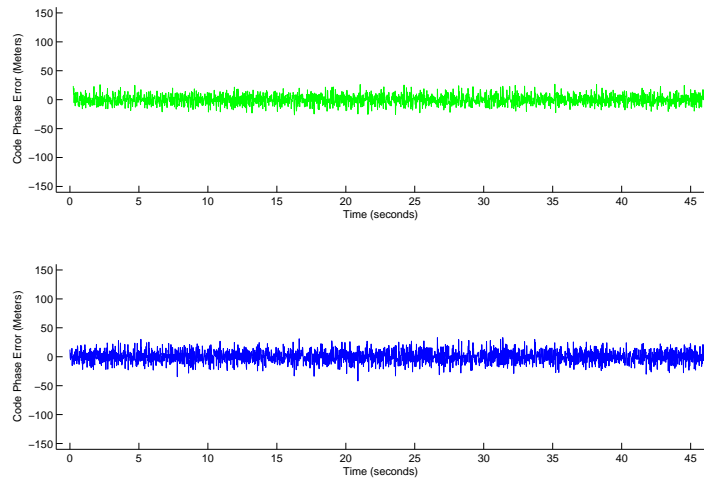


Figure 5.13: PRN Code Phase Error Measurements for SV 3 by the Traditional Delay Lock Loop (top) and the Vector Delay Lock Loop (bottom)

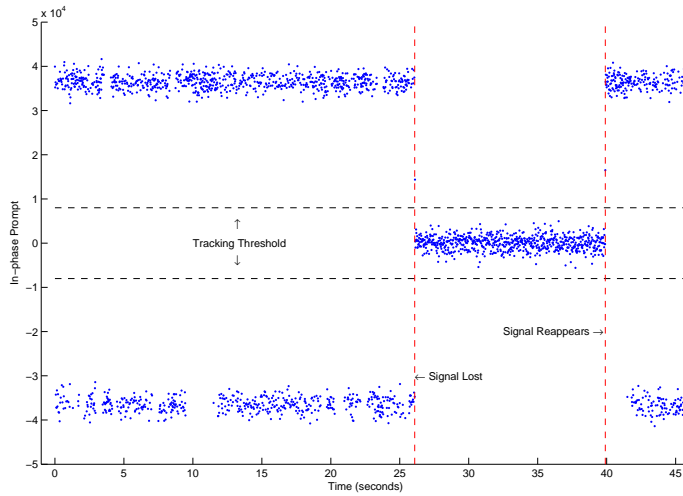


Figure 5.14: In-Phase Prompt Output for SV 12 by VDLL Algorithm

the satellite reappears, the signal exceeds the tracking threshold. The estimates of the PRN code phase and Doppler frequency are close enough to their correct values to make this possible. As in the case of satellite 16 and 5, the VDLL is able to reacquire the signal nearly instantaneously.

Figure 5.16 shows the Doppler estimates maintained by the traditional loop and the Kalman filter based algorithms. The Kalman filter based method begins ignoring measurements when the satellite signal disappears. The estimates provided by the Kalman filter method are again close enough to the correct value to allow fast reacquisition. Figure 5.18 shows the traditional loops failing to re-acquire the signals again.

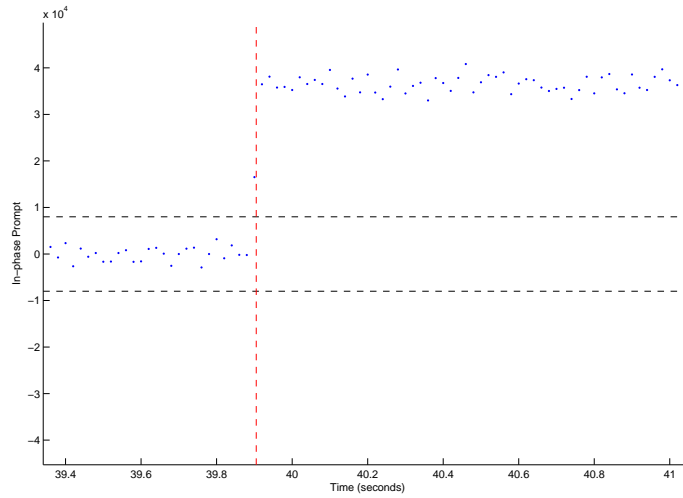


Figure 5.15: In-Phase Prompt Output for SV 12 by VDLL Algorithm

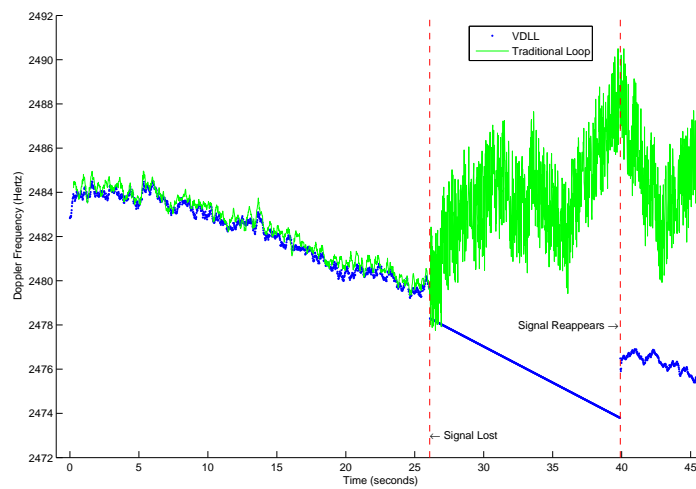


Figure 5.16: Carrier Tracking for SV 12

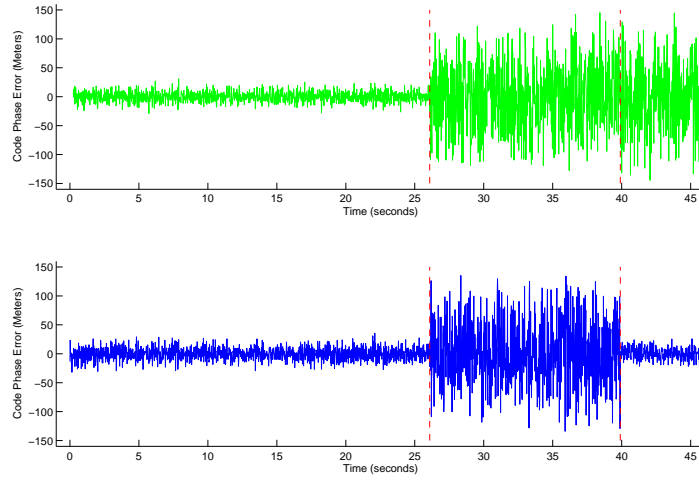


Figure 5.17: PRN Code Phase Error Measurements for SV 12 by the Traditional Delay Lock Loop (top) and the Vector Delay Lock Loop (bottom)

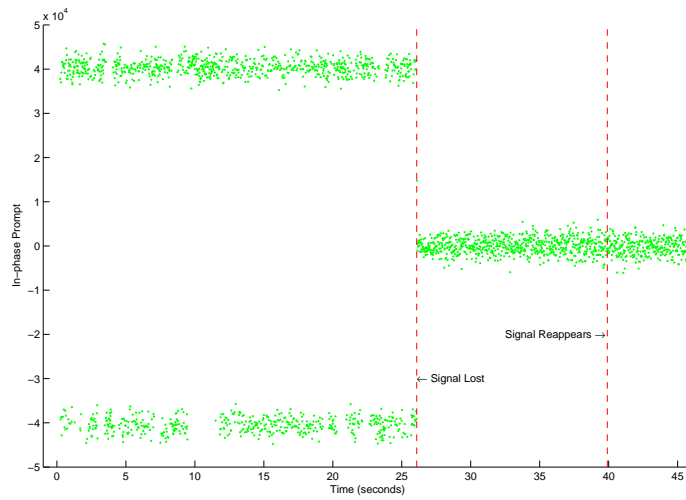


Figure 5.18: In-Phase Prompt Output for SV 12 by Costas Loop

5.3 Positioning Results

As mentioned previously, the profile used in the simulation was a stationary user located at 34.6426° North latitude, 86.6378° West longitude, and an altitude of 30 meters. An iterative least squares solution is used to solve for the user's position and initialize the EKF. The iterative least squares solution yields a position estimate of 34.6427° North latitude, 86.6380° West longitude, and an altitude of -56.0102 meters. The EKF is initialized using this noisy estimate of the user's position.

Figures 5.19, 5.20, and 5.21 show the errors in the Extended Kalman filter's estimates of the user's x, y, and z-coordinates in the ECEF reference frame, respectively. The EKF's estimates of the user's coordinates rapidly begin converging toward the correct values. At approximately 26 seconds into the simulation, three of the four satellite signals are blocked. The diagonal elements of the measurement noise covariance matrix of the EKF corresponding to the three satellites are inflated. The EKF begins ignoring measurements during the period of satellite blockage consequently. This explains the flat line in the EKF's estimates of the user's ECEF coordinates during the satellite outage. When the satellites return, the elements of the measurement covariance matrix are reset to the appropriate values. The EKF then resumes estimating the user's position based on the PRN code phase residuals.

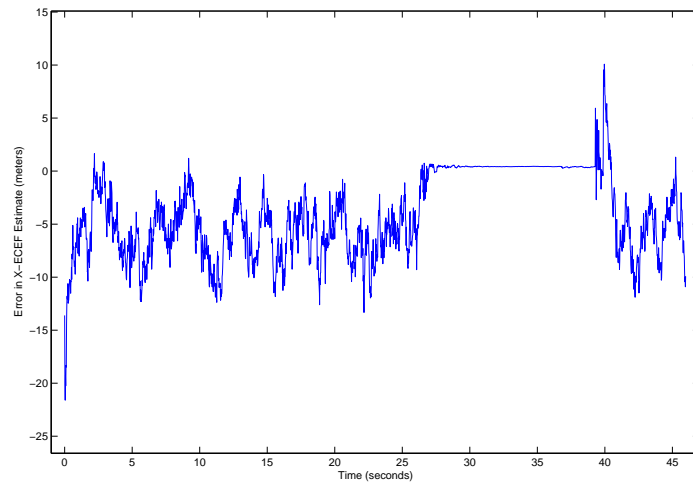


Figure 5.19: Error in X-ECEF Coordinate Estimate

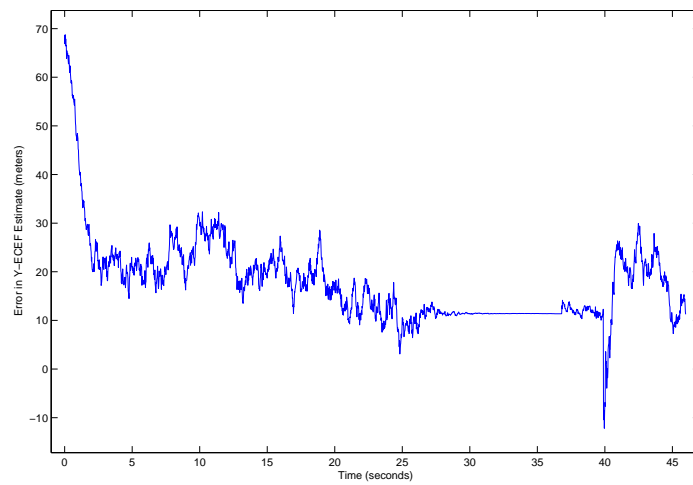


Figure 5.20: Error in Y-ECEF Coordinate Estimate

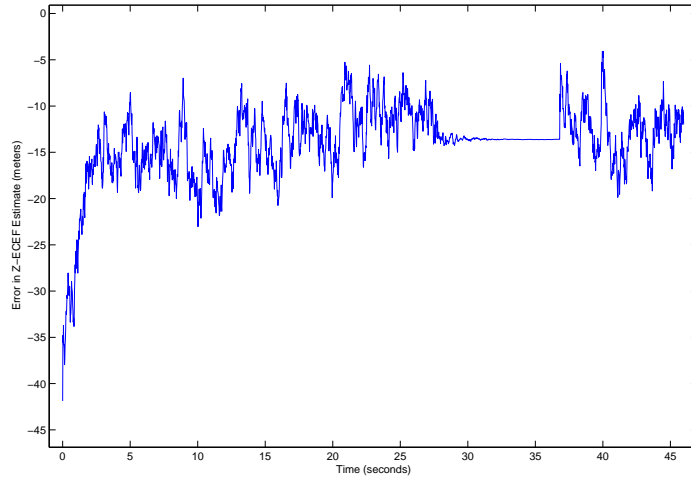


Figure 5.21: Error in Z-ECEF Coordinate Estimate

Figures 5.22, 5.23, and 5.24 display the user's position in longitude, latitude and altitude format. The user's position in this format remains by and large static during the satellite outage as well. The user's altitude can be seen approaching the correct value after being mis-initialized. When the satellite's reappear, the user's position is again being estimated based on the code discriminator outputs.

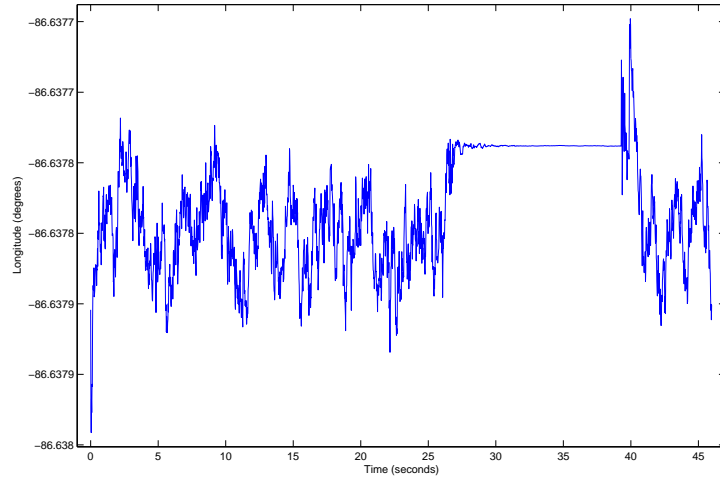


Figure 5.22: User Longitude Based on EKF Estimates

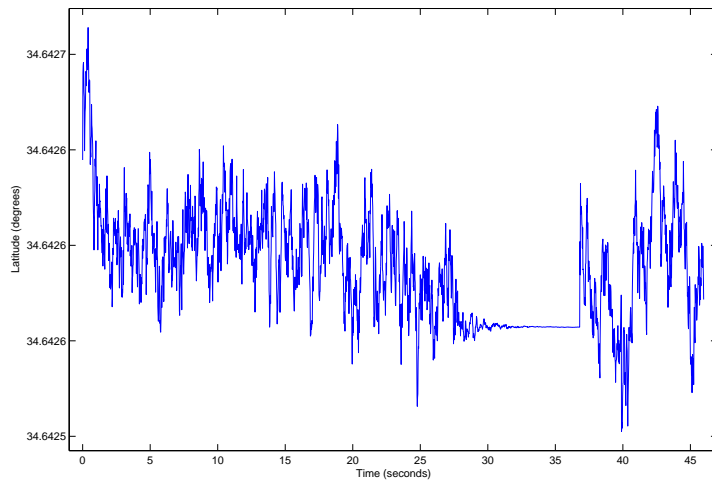


Figure 5.23: User Latitude Based on EKF Estimates

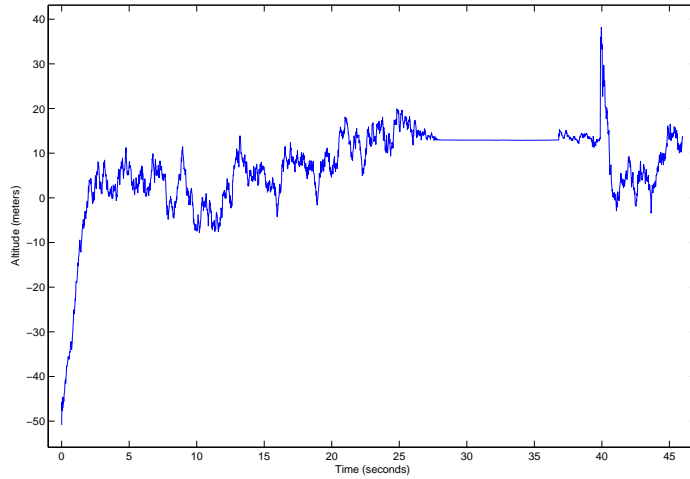


Figure 5.24: User Altitude Based on EKF Estimates

5.4 Clock Results

Figures 5.25 and 5.26 show the EKF's estimates of the user's clock bias and clock drift, respectively. The estimates of the user's clock bias and drift do not fall asleep during the satellite outage. This is due to the linearization of the observation matrix discussed in Chapter 3. The Kalman filter attempts to estimate the clock drift and clock bias based on the single code phase residual. The presence of the IMU decreases the process noise for the position and velocity states greatly. The Kalman filter therefore does not place the code phase residual in the position states. By comparison, the process noise for the clock drift and bias is much greater. The Kalman filter therefore continues to place the code phase residual in the clock bias and drift states.

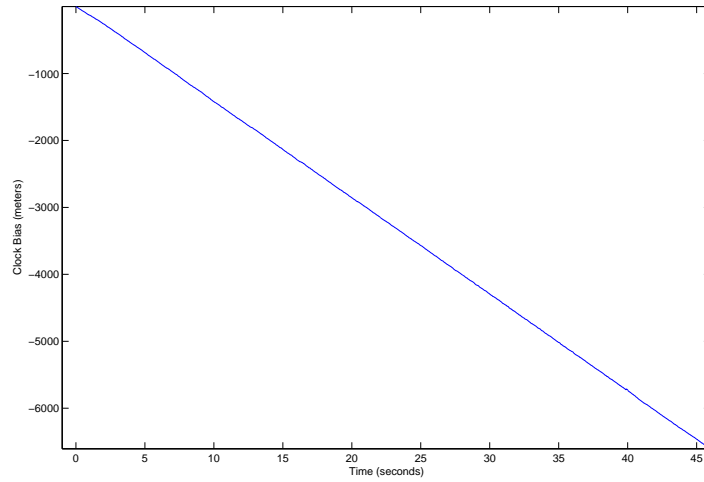


Figure 5.25: EKF Estimate of Clock Bias

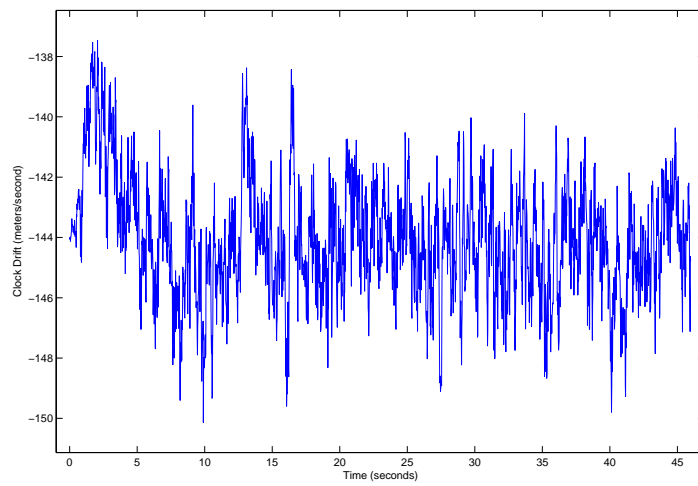


Figure 5.26: EKF Estimate of Clock Drift

6.1 Concluding Remarks

In this thesis, the traditional tracking loops used in GPS receivers were replaced with Kalman filters. The traditional loops and their shortcomings were described in Chapter 3. In Chapter 4, the author presented several new Kalman filter based tracking algorithms. The vector delay lock loop was introduced and its potential advantages were explained. The VDLL developed by the author is based on non-linear discriminator functions and does not need to estimate the amplitudes of the received signals. In Chapter 5, data was collected using a GPS constellation simulator in order to validate the operation of the new vector delay lock loop algorithm. Chapter 5 details the operation of the VDLL during normal signal conditions and during a temporary satellite blockage. The performance of the VDLL in these circumstances was also compared to the operation of traditional tracking loops. The results offer proof of concept for the author's VDLL algorithm and demonstrate its ability to rapidly reacquire signals after a temporary satellite blockage. The traditional methods failed to reacquire the blocked signals when they reappeared.

The VDLL algorithm developed in this thesis was shown to be able to reacquire lost signals nearly instantaneously. In addition, the VDLL, when combined with an IMU, was able to operate during periods of satellite blockage. The new VDLL method has the ability to estimate the user's position when a full complement of satellite's is not available. The VDLL method also has the potential to operate at significantly lower carrier to noise power levels than traditional methods.

6.2 Future Work

There is a great potential for future work based on the vector delay lock loop algorithms developed in this thesis. The first is to extend the vector tracking concept to integrated vector frequency tracking. In a vector frequency locked loop, the tracking of the satellite carrier signals is performed, along with the PRN codes, by an Extended Kalman filter. The second area of future work is to further integrate the tracking of the GPS signals with inertial measurement units. This would allow the EKF to better operate in scenarios where the user is moving. The third area that needs to be further explored is the determination of the process and measurement noise covariance matrices. Analytical determination of the process noise covariance matrix would include extensive clock modeling and modeling of other error sources. Predicting the measurement noise covariance matrix would involve estimating the current C/N_0 levels and placing the appropriate noise statistics in the noise covariance matrix. The fourth area of future work is filter integrity. It is desirable to have an algorithm that gracefully degrades as C/N_0 levels decrease or jamming levels increase.

BIBLIOGRAPHY

- [Abbott and Lillo, 2003] Abbott, A. S. and Lillo, W. E. (2003). Global Positioning Systems and Inertial Measuring Unit Ultratight Coupling Method. U. S. Patent 6,516,021, The Aerospace Corporation, Segundo, CA.
- [Axelrad and Brown, 1996] Axelrad, P. and Brown, R. G. (1996). GPS Navigation Algorithms. In Parkinson, B. W., editor, *Global Positioning System: Theory and Applications, Volume 1*, volume 163 of *Progress in Astronautics and Aeronautics*, chapter 9. American Institute of Aeronautics and Astronautics, Washington, DC.
- [Bertelsen et al., 2004] Bertelsen, N., Borre, K., and Rinder, P. (2004). The GPS Code Software Receiver at Aalborg University. Technical report, Danish GPS Center, Aalborg University.
- [Best, 1999] Best, R. E. (1999). *Phase-Locked Loops*. McGraw-Hill.
- [Bullock et al., 2006] Bullock, J. B., Foss, M., Geier, G. J., and King, M. (2006). Integration of GPS With Other Sensors and Network Assistance. In Kaplan, E. D., editor, *Understanding GPS: Principles and Applications*, Mobile Communication Series, chapter 9, pages 459–558. Artech House Publishers, 2 edition.
- [Cahn et al., 1977] Cahn, C. R., Leimer, D. K., Marsh, C. L., Huntowski, F. J., and Larue, G. D. (1977). Software Implementation of a PN Spread Spectrum Receiver to Accommodate Dynamics. *IEEE Transactions on Communications*, 25:832–839.
- [Carlson, 1988] Carlson, N. A. (1988). Federated filter for fault-tolerant integrated navigation systems. In *IEEE Position Location and Navigation systems*, pages 110–119.
- [Crossbow Technology, 2006] Crossbow Technology (2006). IMU400CD Data Sheet.
- [Dierendonck, 1996] Dierendonck, A. J. V. (1996). GPS receivers. In Parkinson, B. W., editor, *Global Positioning System: Theory and Applications, Volume 1*, volume 163 of *Progress in Astronautics and Aeronautics*, chapter 8. American Institute of Aeronautics and Astronautics, Washington, DC.
- [Egan, 1998] Egan, W. F. (1998). *Phase-Lock Basics*. John Wiley & Sons.
- [Feigin, 2002] Feigin, J. (2002). Practical Costas Loop Design. Technical report, Analog Devices.
- [Flenniken, IV, 2005] Flenniken, IV, W. S. (2005). Modeling Inertial Measurement Units and Analyzing the Effect of Their Errors in Navigation Applications. Master’s thesis, Auburn University.

- [Gautier, 2003] Gautier, J. D. (2003). *GPS/INS Generalized Evaluation Tool (GIGET) For The Design and Testing of Integrated Navigation Systems*. PhD thesis, STANFORD UNIVERSITY.
- [Gelb, 1974] Gelb, A., editor (1974). *Applied Optimal Estimation*. The M. I. T. Press.
- [Godha, 2004] Godha, S. (2004). Strategies for GPS/INS Integration. Presentation.
- [Grewal and Andrews, 1993] Grewal, M. S. and Andrews, A. P. (1993). *Kalman Filtering Theory and Practice*. Prentice Hall.
- [Gustafson and Dowdle, 2003] Gustafson, D. and Dowdle, J. (2003). Deeply Integrated Code Tracking: Comparative Performance Analysis. In *Proceedings of Institute of Navigation GPS/GNSS Conference*, pages 2553–2561, Portland, OR. Institute of Navigation.
- [Gustafson et al., 2000] Gustafson, D., Dowdle, J., and Flueckiger, K. (2000). A High Anti-Jam GPS-Based Navigator. In *Institute of Navigation, Annual Technical Meeting*.
- [Gustafson et al., 2001] Gustafson, D. E., Dowdle, J. R., and John M. Elwell, J. (2001). Deeply-Integrated Adaptive GPS-Based Navigator with Extended-Range Code Tracking. U. S. Patent 6,331,835, The Charles Stark Draper Laboratory, Inc., Cambridge, MA.
- [Hamm, 2005] Hamm, C. R. (2005). Analysis of Simulated Performance of Integrated Vector Tracking and Navigation Loops For GPS. Master’s thesis, Auburn University.
- [Horslund and Hooker, 1999] Horslund, J. M. and Hooker, J. R. (1999). Increase Jamming Immunity by Optimizing Processing Gain for GPS/INS Systems. U. S. Patent 5,983,160, Raytheon Company, Lexington, MA.
- [Jazwinski, 1970] Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press.
- [Jwo, 2001] Jwo, D. J. (2001). Optimization and Sensitivity Analysis of GPS Receiver Tracking Loops in Dynamic Environments. *IEE Proceedings of Radar, Sonar Navigation*, 148:241–250.
- [Kaplan et al., 1996] Kaplan, E. D., Leva, J. L., and Pavloff, M. S. (1996). Fundamentals of Satellite Navigation. In Kaplan, E. D., editor, *Understanding GPS: Principles and Applications*, Mobile Communication Series, chapter 2, pages 15–58. Artech House Publishers.
- [Kleusberg, 1999] Kleusberg, A. (1999). Analytical GPS Navigation Solution. Technical report, University of Stuttgart.

- [Kreye et al., 2004] Kreye, C., Eissfeller, B., and Ameres, G. (2004). Architectures of GNSS/INS Integrations: Theoretical Approach and Practical Tests. In *Symposium on Gyro Technology*, pages 14.0–14.16.
- [Levy, 1997] Levy, L. J. (1997). The Kalman filter: Navigations Integration Workhorse. *GPS World*, 8:22.
- [Misra and Enge, 2001] Misra, P. and Enge, P. (2001). *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press.
- [Negast, 2006] Negast, B. (2006). Personal communication.
- [Normack et al., 2002] Normack, P. L., Borjesson, M., and Stahlberg, C. (2002). NordNav Tehnologies: Fast, Flexible, Easy to Adapt. Technical report, NordNav Tehnologies.
- [Pany et al., 2005] Pany, T., Kaniuth, R., and Eissfeller, B. (2005). Deep Integration of Navigation Solution and Signal Processing. In *Proceedings of Institute of Navigation GPS/GNSS Conference*, Long Beach, CA. Institute of Navigation.
- [Parkinson, 1996] Parkinson, B. W. (1996). Introduction and Heritage of NAVSTAR, the Global Positioning System. In Parkinson, B. W., editor, *Global Positioning System: Theory and Applications, Volume 1*, volume 163 of *Progress in Astronautics and Aeronautics*, chapter 1. American Institute of Aeronautics and Astronautics, Washington, DC.
- [Petovello and Lachapelle, 2006] Petovello, M. and Lachapelle, G. (2006). Comparison of Vector-Based Software Receiver Implementations With Application to Ultra-Tight GPS/INS Integration. In *Proceedings of Institute of Navigation GPS/GNSS Conference*, Fort Worth, TX. Institute of Navigation.
- [Porat, 1997] Porat, B. (1997). *A Course in Digital Signal Processing*. John Wiley & Sons.
- [Rounds, 2004] Rounds, S. (2004). Jamming Protection of GPS Receivers: Part I: Receiver Enhancements. *GPS World*.
- [Sayre, 2003] Sayre, M. M. (2003). Development of a Block Processing Carrier to Noise Ratio Estimator for the Global Positioning System. Master’s thesis, Ohio University.
- [Sklar, 2001] Sklar, B. (2001). *Digital Communications: Fundamentals and Applications*. Prentice Hall PTR, 2nd edition.
- [Soloviev et al., 2004] Soloviev, A., van Graas, F., and Gunawardena, S. (2004). Implementation of Deeply Integrated GPS/Low-Cost IMU for Reacquisition and Tracking of Low CNR GPS Signals. In *Proceedings of the Institute of Navigation National Technical Meeting*, pages 923–935, San Diego, CA. Institute of Navigation.

- [Spilker, Jr., 1996a] Spilker, Jr., J. J. (1996a). Fundamentals of Signal Tracking Theory. In Parkinson, B. W., editor, *Global Positioning System: Theory and Applications, Volume 1*, volume 163 of *Progress in Astronautics and Aeronautics*, chapter 4. American Institute of Aeronautics and Astronautics, Washington, DC.
- [Spilker, Jr., 1996b] Spilker, Jr., J. J. (1996b). GPS Signal Structure and Theoretical Performance. In Parkinson, B. W., editor, *Global Positioning System: Theory and Applications, Volume 1*, volume 163 of *Progress in Astronautics and Aeronautics*, chapter 3. American Institute of Aeronautics and Astronautics, Washington, DC.
- [Tsui, 2000] Tsui, J. B. Y. (2000). *Fundamentals of Global Positioning System Receivers, A Software Approach*. John Wiley & Sons.
- [US Dept. of Defense, 2000] US Dept. of Defense (2000). NAVSTAR GPS Space Segment/Navigation User Interfaces. Interface Control Document No. ICD-GPS-200C, Department of Defense.
- [Ward, 1996a] Ward, P. (1996a). Effects of RF Interference on GPS Satellite Signal Receiver Tracking. In Kaplan, E. D., editor, *Understanding GPS: Principles and Applications*, Mobile Communication Series, chapter 6, pages 209 – 236. Artech House Publishers.
- [Ward, 1996b] Ward, P. (1996b). Satellite Signal Acquisition and Tracking. In *Understanding GPS: Principles and Applications*, Mobile Communication Series, chapter 5, pages 119 – 208. Artech House Publishers.
- [Welch and Bishop, 2006] Welch, G. and Bishop, G. (2006). An Introduction to the Kalman Filter. Technical Report 95-041, University of North Carolina at Chapel Hill.

APPENDIX - KALMAN FILTER

The Kalman Filter is an optimal state estimator for linear dynamic systems with disturbances modeled by Gaussian random processes. The filter's performance is optimal with respect to a quadratic cost function. In general, the Kalman filter estimates the states of a system in which Gaussian noise both drives the system and corrupts measurements made of the system's states. The Kalman filter is an optimal observer in the sense that it produces *unbiased* and *minimum variance* estimates of the states of the system. The term *unbiased* means that the expected value of the error between the filter's estimate and the true state of the system is zero. The Kalman filter's estimates of the states of the system are *minimum variance* because the expected value of the squared error between the real and estimated states is minimized. The Kalman filter algorithm exists for both discrete and continuous time models (the continuous time version is generally referred to as the Kalman-Bucy filter). Only the discrete time Kalman filter was employed, a discussion of the continuous case will therefore be neglected [Grewal and Andrews, 1993] [Gelb, 1974].

Consider the system described by (6.1). The states of the system at time k are produced by a linear combination of the states at time $k - 1$ plus noise w_{k-1} . The noise w_k is assumed to be Gaussian with zero mean and covariance Q_k .

$$\begin{aligned}x_k &= A_k x_{k-1} + w_{k-1} \\w_k &\sim \mathcal{N}(0, Q_k)\end{aligned}\tag{6.1}$$

Noisy measurements are made of the system at each time step k , (6.2). The measurements are a linear combination of the current states of the system plus noise v_k . The noise v_k is Gaussian with covariance matrix R_k .

$$z_k = H_k x_k + v_k \quad (6.2)$$

$$v_k \sim \mathcal{N}(0, R_k)$$

The process noise w_k and the measurement noise v_k are assumed to be uncorrelated for all past and future values, (6.3). The Kalman filter produces estimates \hat{x}_k at each measurement epoch of the state vector x_k that minimize the expected value of a weighted mean-squared cost function, (6.4). The weighting matrix M can be any symmetric nonnegative definite matrix.

$$E[w_m v_n^T] = 0 \text{ for all } m \neq n \quad (6.3)$$

$$J = E[x_k - \hat{x}_k]^T M [x_k - \hat{x}_k] \quad (6.4)$$

The Kalman filter is generally arranged into a distinctive predictor-correction algorithm. The filter is initialized with estimates of the mean and covariance of the state vector, \hat{x}_0 and P_0 respectively. The Kalman filter then propagates the states of the system and the error covariance matrix ahead to the next sampling time. Using the propagated error covariance matrix, the so called *Kalman gain* matrix K_k is computed. The propagated states are the Kalman filter's prediction of the state vector at the next epoch. After the measurement takes place, the differences between the measured states and the predicted states are calculated. These errors are referred to as the *residuals* of

the filter. The vector of residuals is multiplied by the Kalman gain matrix and added to the filter's predicted state matrix. This affine operation produces a corrected estimate of the state vector at the sampling epoch. The error covariance matrix is then updated to reflected the covariance of the corrected state vector estimate. The updated error covariance matrix and estimated state vector are then used as the initial conditions were. This cycle of prediction and correction is shown graphically in figure 6.1 .

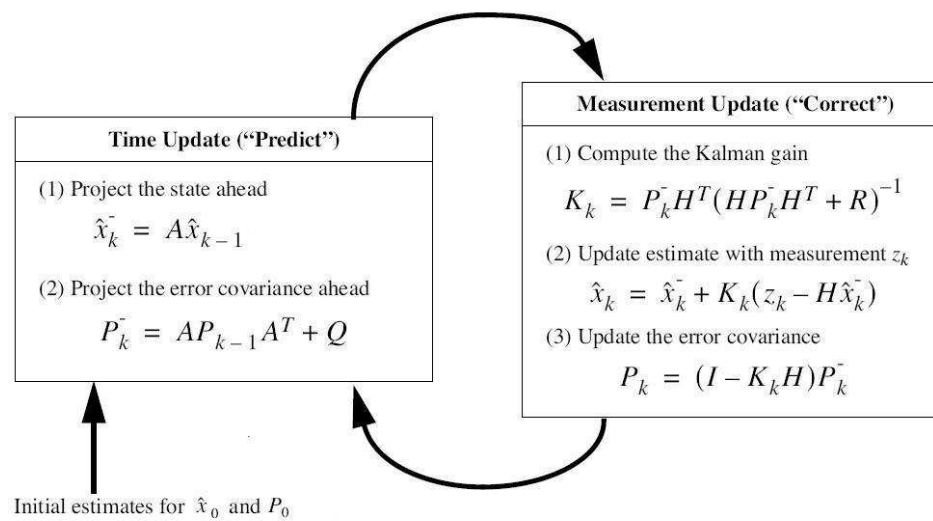


Figure 6.1: Kalman Filter Algorithm [Welch and Bishop, 2006]

The Kalman filter is an optimal estimator for linear systems only. However, the vast majority of actual systems are nonlinear. The Extended Kalman Filter (EKF) is an *ad hoc* application of the Kalman filter for nonlinear systems. The EKF is not an optimal estimator in the least squares sense, but approximates the operation of an optimal filter. The EKF functions by linearizing the system around the present mean and covariance. Consider the nonlinear system described by (6.5). The state vector at the next time step

is a nonlinear function of the previous states and sample index $k - 1$, plus process noise w_{k-1} . The measurements of the system are a nonlinear function of the current states of the system and sample index k , plus measurement noise v_k .

$$x_k = f(x_{k-1}, k - 1) + w_{k-1} \quad (6.5)$$

$$z_k = h(x_k, k) + v_k$$

The EKF operates by linearizing the system around the current best estimates of the state vector. The nonlinear function $f(x_{k-1}, k - 1)$ is approximated by the matrix A_k between measurement epochs, (6.6). The nonlinear measurement function $h(x_k, k)$ is approximated by the matrix H_k , (6.7). The matrices A_k and H_k must be reevaluated at each epoch. Using the two linear approximations A_k and H_k , the Kalman filter algorithm can be used for the nonlinear system. Figure 6.2 shows the Extended Kalman filter update equations graphically.

$$A_k \approx \left. \frac{\partial f(x_{k-1}, k - 1)}{\partial x_k} \right|_{x=\hat{x}_{k-1}^-} \quad (6.6)$$

$$H_k \approx \left. \frac{\partial h(x_k, k)}{\partial x} \right|_{x=\hat{x}_{k-1}^-} \quad (6.7)$$

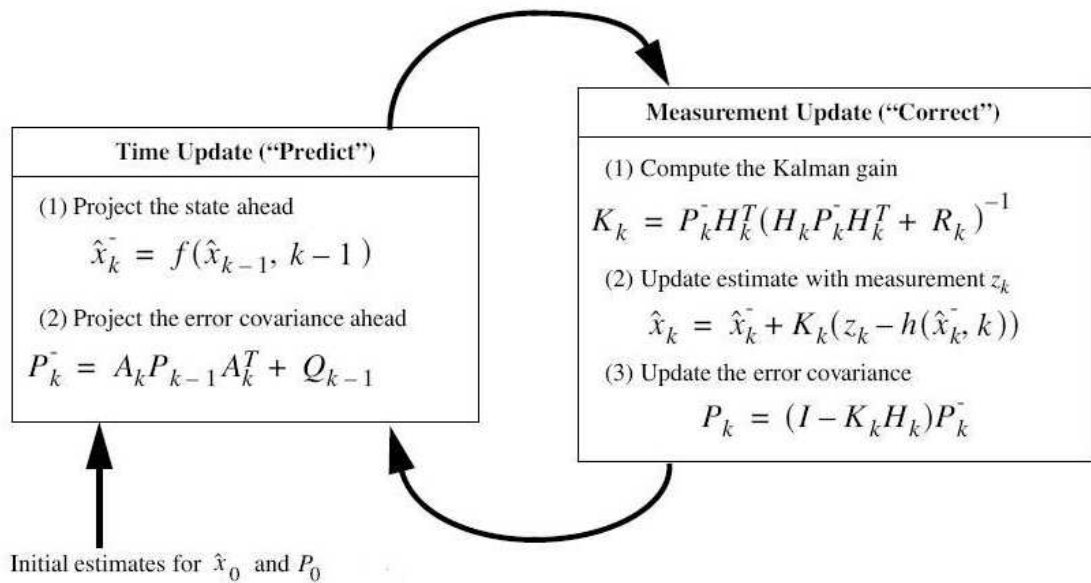


Figure 6.2: Extended Kalman Filter Algorithm [Welch and Bishop, 2006]

The EKF's method of linearization requires the nonlinear functions $f(x_{k-1}, k-1)$ and $h(x_k, k)$ both be twice continuously differentiable. If the errors between the estimated state vector and the true state vector remain small, the linearization assumption is accurate. Higher order approximations have been derived but they typically involve significantly greater complexity while not markedly outperforming the EKF [Jazwinski, 1970] [Grewal and Andrews, 1993] [Gelb, 1974].