Performance and Integrity Analysis of the Vector Tracking Architecture of GNSS Receivers

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Susmita Bhattacharyya

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Adviser: Demoz Gebre-Egziabher

April 2012

Acknowledgments

First of all, I would like to thank my thesis advisor Prof. Demoz Gebre-Egziabher for his invaluable help and guidance throughout my PhD. He kept me focused and yet gave me wide latitude to explore new avenues which not only nourished my intellect, but provided me with the opportunity to savor every phase of my research. His insightful comments on a diverse range of topics - both technical and non-technical - have always enriched me and helped me broaden my perspectives. His help and initiatives to collect data for the last part of this thesis are greatly appreciated. I also owe him a great debt of gratitude for his careful reading of my thesis and for his thorough comments and suggestions.

I would like to acknowledge the comments and suggestions of my defense committee members Prof. Yiyuan Zhao, Prof. Peter Seiler and Prof. Max Donath. Special thanks are due to Prof. Seiler for his meticulous reading of my thesis. I also owe great thanks to Prof. Gary Balas whose constructive criticism during my Preliminary Oral Examination and group seminars has contributed greatly to improving the quality of my research.

I would like to thank all my colleagues, past and present, (Chen-Chi Chu, Zhefeng Li, F. Adhika Pradipta Lie, Hamid Mokhtarzadeh, Zhiqiang Xing and Guijin Zheng) for all formal and informal discussions I had with them. Staff of the Department of Aerospace Engineering and Mechanics and MEnet deserve special thanks for all the administrative and technical assistance I have received from them.

Special thanks are due to Honeywell Research Labs in Golden Valley, Minnesota for letting us use their Spirent simulator and receiver front end without which this research would not have been complete. I thank National Aeronautics and Space Administration and Honeywell International Inc. for funding part of this research. This work was also partially supported by the U.S. Department of Homeland Security under Award Number 2008-ST-061-BS0002. However, the views and conclusions contained in this thesis are of mine and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

I owe thanks to my friends Nupur Hajela, Mukesh Nathan, Narina Jung and Maureen Whitford for their help and support which made my stay in the US enjoyable. My involvement with the local chapter of the Art of Living Foundation - a spiritual organization based in India - toward the end of my PhD has proved beneficial in many respects. I owe thanks to all of my Art of Living friends for their generous support and help. I would like to acknowledge the Film Society of Minneapolis-St. Paul / Minnesota Film Arts for giving me the unique opportunity to glimpse the beautiful world of World Cinema that nicely complemented my work.

I am grateful to my undergraduate mentor Prof. Tapan Ghoshal for his guidance on numerous occasions and for kindling my interest in higher studies. I am also greatly indebted to my Masters thesis advisor Prof. Pravas Mahapatra for introducing me to the field of Navigation. I owe a great debt of gratitude to my spiritual teacher Sri Sri Ravi Shankar whose simple yet profound talks have greatly enriched me and helped me weather difficult times.

Last but not least, I would like to thank my elder sister Sukanya Mukherjee. Without her encouragement I could not have come this far. Finally, I owe a great debt of gratitude to my parents Sripati and Sujita Bhattacharyya for their constant moral support and encouragement from afar without which this thesis would not have been possible.

Abstract

Frequent loss or attenuation of signals in urban areas and integrity (or reliability of system performance) are two principal challenges facing the Global Navigation Satellite Systems or GNSS today. They are of critical importance especially to safety or liability-critical applications where system malfunction can cause safety problems or has legal/economic consequences. To deal with the problem of integrity, algorithms called integrity monitors have been developed and fielded. These monitors are designed to raise an alarm when situations resulting in misleading information are identified. However, they do not enhance the ability of a GNSS receiver to track weak signals. Among several approaches proposed to deal with the problem of frequent signal outage, an advanced GNSS receiver architecture called vector tracking loops has attracted much attention in recent years.

While there is an extensive body of knowledge that documents vector tracking's superiority to deal with weak signals, prior work on vector loop integrity monitoring is scant. Systematic designs of a vector loop-integrity monitoring scheme can find use in above-mentioned applications that are inherently vulnerable to frequent signal loss or attenuation. Developing such a system, however, warrants a thorough understanding of the workings of the vector architecture as the open literature provides very few preliminary studies in this regard.

To this end, the first aspect of this research thoroughly explains the internal operations of the vector architecture. It recasts the existing complex vector architecture equations into parametric models that are mathematically tractable. An in-depth theoretical analysis of these models reveals that inter-satellite aiding is the key to vector tracking's superiority.

The second aspect of this research performs integrity studies of the vector loops. Simulation results from the previous analysis show that inter-satellite aiding allows easy propagation of errors (and faults) among satellite loops in vector tracking mode. Hence, the basic single fault requirement of the traditional Receiver Autonomous Integrity Monitoring or RAIM is violated with the pseudorange measurements of the vector architecture. This work develops a vector loop RAIM scheme that addresses above limitation. The designed vector loop RAIM algorithm is validated via a high fidelity simulation of an aircraft making an instrument approach.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Although originally envisioned to serve the US military, the Global Positioning System or GPS today is used in a broad spectrum of non-military applications ranging from civil aviation [1], highway traffic management [2], [3] to guiding visually impaired individuals [4]. Over the years, it has evolved into an indispensable infrastructure of society as everyday activities, such as banking operations [5], power grid control [5] and precision agriculture [6], are increasingly dependent on its positioning, navigation and timing (PNT) services. The reliance on GPS for navigating from one location to another has grown so much in recent years that it is now used as a standard feature in many consumer products such as mobile phones, netbooks and tablets, etc. [7]. According to Berg Insight, a Swedish research company that tracks pivotal developments in next generation technologies, global shipments of GPS-enabled mobile phones increased almost 97 percent in 2010 to 295 million units [8]. With a compound annual growth rate of 28.8 percent, shipments will reach 940 million units in 2015, it forecasted. Furthermore, as the number of GPS applications grows, so are increasingly complex user needs which, in turn, open up new frontiers of GPS usage that could not have been imagined earlier. Remote sensing of the environment using GPS reflections is worthy of mention in this regard [9].

The trend of using GPS everywhere, including applications where system malfunction or disruption can cause safety problems or has legal/economic consequences, naturally brings forth the need to improve its robustness against numerous sources of signal degradations. Some of these include atmospheric disturbances; signal blockage by tall buildings or trees; and intentional or unintentional interference. For example, a GPS-based driver-assist system that alerts the driver when another vehicle is approaching around the next corner might lead to catastrophe if it malfunctions when needed most. Or, large position errors without warnings may result in the guidance system telling visually impaired persons that they are safe and on the sidewalk when they are actually a few meters in the crosswalk. Consequently, developing innovative electronics and receiver architectures to overcome the vulnerability of GPS (and other future Global Navigation Satellite Systems or GNSS) signals has long been the subject of much research. Vector tracking (or vector tracking loops) is one such advanced architecture which has attracted much attention in recent years because of its promise to address the issue of robustness. Analyzing some of the key features of this architecture is the focus of the work reported in this thesis.

It is well documented in the literature that vector tracking loops can track weak, attenuated GPS signals, bridge through momentary signal outage and withstand high user dynamics [10]. The ability of vector tracking to operate with low powered signals and high dynamics is attributed to the fact that, unlike scalar tracking, it processes all satellite signals in unison [11]. The conventional scalar architecture performs the functions of satellite signal tracking and user position, velocity and time (PVT) estimation independently, resulting in a two-stage, cascaded structure of parallel tracking loops - one for each satellite channel - and a navigation filter. The vector tracking architecture combines these two functions into one by processing all satellite signals together inside a central navigation filter and then closing the signal tracking loops with the filter outputs. Thus, it eliminates *independent* tracking loops used in the scalar architecture.

Even though there is a large body of work documenting vector tracking's superiority to deal with weak signals or high dynamics, prior work on vector loop-based integrity monitoring is scant [12]. Integrity refers to the ability of a system to issue timely warnings to the user when the system operation is not considered safe. Systematic designs and performance analysis of vector loop integrity monitoring schemes deserve extensive studies. This is because the vector architecture, when equipped with integrity monitoring, can be expected to have an edge in afore-mentioned applications. These applications are inherently susceptible to signal attenuation as well as hazardous and misleading information. However, owing to the fundamental difference in the way scalar and vector tracking loops operate, the principles of scalar loop-integrity monitoring may not be directly applicable to the vector architecture. This warrants a thorough understanding of vector loop internal operations as well as reasons for its superiority before designing its integrity monitoring algorithm. The literature (in particular, open literature) provides very few preliminary studies in this regard [10], [11]. Such insight can also lead to the development of architectures that are less complex, but offer almost all benefits of vector loops. In response to this challenge, the research in this thesis provides a different, more intuitive perspective on the internal workings of the vector architecture. This analysis, in parallel with explaining vector loops' superiority, points to an important pitfall of such complex architecture which naturally leads to extensive integrity studies of vector tracking. Thus, the second part of this thesis provides an in-depth integrity analysis of vector tracking. The next few sections of this chapter describe in detail the significance and motivation behind the research in this thesis.

1.2 Ubiquity of GNSS

Over the past decade, the advent of several global and regional satellite constellations has ushered in a new era of multiple satellite systems collectively known as GNSS. They comprise four global systems (the United State's GPS, the Russian Federation's GLONASS, the European Union's Galileo and China's Compass) and two regional systems (Japan's QZSS and India's IRNSS). Of all, GPS is the only component of GNSS systems that has full operational capability at the time of writing this thesis. Consequently, GPS and GNSS will be used synonymously in this thesis.

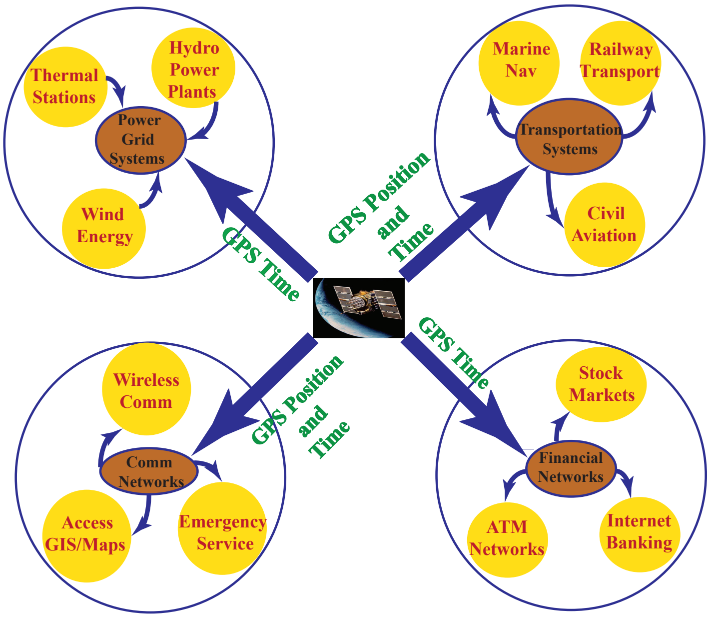


Figure 1.1: GPS Applications

GPS today has expanded so widely that its civilian applications now outnumber military ones, thus enabling it to become an integral part of the daily lives of many. Figure 1.1 illustrates four key facets of GPS applications which range from supporting transportation and communications systems, operating financial networks to synchronizing power grid time. Some other important uses of GPS include mass market applications (e.g., driver-assist systems, personal navigation devices), surveying and geodesy [13], scientific applications such as remote sensing [9] and atmospheric occultation measurements [14]. In addition, a new set of applications have emerged recently which involve the use of GPS signals for authentication purposes [15]. An example in this vein is the seamless tracking of shipments from source to destination, thereby enhancing the efficiency of customs operations at land and maritime ports of entry. It is now estimated that there are close to one billion GPS receivers worldwide [16]. The GPS market worldwide (including core GPS market and GPS-enabled market) is projected to reach about $216 billion in 2012 [17]. Thus, GPS now is so entrenched into day-to-day activities that any disruption or malfunction of its service can incur safety or legal/economic consequences. As a result, considerable effors are underway in both private and public sectors to detect, identify and mitigate GPS (and future GNSS) vulnerability to ensure robustness and integrity of the service. In what follows, potential sources of GNSS vulnerability and proposed ways to deal with them are discussed briefly.

1.3 GNSS Signal Vulnerability

The various sources of GNSS signal vulnerability can be categorized into three groups: Signal loss or attenuation due to physical obstructions, radio frequency (RF) interference and atmospheric disturbances. Each of these sources and their effects on GNSS signals are discussed in brief below.

1.3.1 Signal Loss or Attenuation in Urban Areas

With tremendous growth of GNSS as a PNT system in numerous civilian applications, frequent loss or attenuation of signals in urban environments has become a principal challenge facing the system today. This is because, while using GNSS for PNT operations in urban environments, direct line-of-sight (LOS) signals can be easily blocked, attenuated or reflected by a freeway overpass, dense foliage or high-rise buildings lined along narrow streets. A typical example of such a challenging urban canyon signal en-

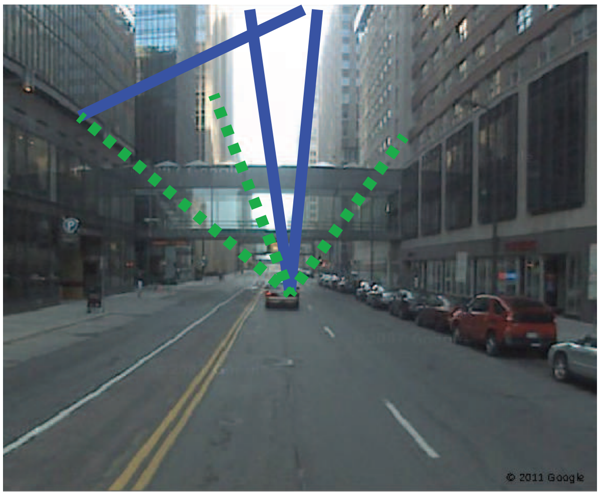


Figure 1.2: Satellite Signals in an Urban Canyon in Downtown Minneapolis; Solid Lines: Direct LOS Signals; Dashed Lines: Reflected Signals

vironment is illustrated in Figure 1.2. Analysis of field test data in [10] demonstrates that carrier power to noise power density (C/N_0) ratios of some of the visible satellites in such signal fading environments can drop below 5 dB-Hz. In this context, it should be noted that typical values of open-sky/good C/N_0 lie in the range of 35 to 55 dB-Hz [18]. The problem of signal blockage or attenuation is even more pronounced in indoor navigation where signals need to penetrate building materials to reach the receiver. According to a study conducted by the National Institute of Standards and Technology, attenuation of GPS signals caused by building walls and floors can be as high as 29 dB [19].

1.3.2 RF Interference

Alongside ever-growing number of GPS applications, increasing incidents of intentional and unintentional RF interference are being reported that have rendered GPS inoperable for critical infrastructure operations [16]. The problem is compounded by the availablity of GPS jammers which, while illegal, can be purchased easily and the difficulty associated with locating and mitigating the interference they cause. GNSS signals are inherently vulnerable to in-band RF interference as the satellite signal power reaching the Earth's surface is very weak (on the order of 10^{-16} Watts). It is apparent from such weak signals which remain buried in noise prior to code correlation that any terrestrial RF phenomenon can easily overwhelm or interfere with GNSS service.

Different sources of RF interference can be broadly classified into wideband and narrowband interference depending on their bandwidths compared to the GNSS spread spectrum bandwidth which is 2.046 MHz for the GPS L1 C/A-code. Wideband interference can be Gaussian (intentional jammers), phase/frequency modulated (television transmitter harmonics), pulsed (radar transmitters) or spread-spectrum (intentional spread spectrum jammers or near-field of pseudolites) in nature [20]. On the other hand, narrowband interference includes narrowband phase/frequency modulation (AM station harmonics), narrowband-swept continuous wave (CW) (FM station harmonics) and narrowband CW (intentional CW jammers) [20]. RF interference effectively reduces the C/N_0 of the GNSS signals, thereby causing the receiver to lose track or preventing it from acquiring satellite signals. It should be noted that a little RF interference energy from a far away RF source can jeopardize the operation of a GNSS receiver. For example, a 10 mw jammer can prevent receivers as far as 10 km away from acquiring the C/A-code. It can cause them to lose lock at about 1 km [21].

While intensive RF interference can deny GNSS service, a recently emerging threat called spoofing is even more dangerous. This is because a spoofer, by simulating fake signals, tricks the receiver into believing that it is tracking real GNSS signals [21]. However, spoofing is not as straightforward as jamming a receiver.

1.3.3 Atmospheric Disturbances

While the above are examples of man-made interference, atmospheric disturbances can be considered unintentional interference caused by Nature. On the way from satellites to the receiver, GNSS signals pass through the Earth's atmosphere of which a portion is the ionosphere. The ionosphere acts as a dispersive medium to GNSS signals, delaying the modulating code and advancing the carrier wave. Under quiet ionospheric conditions, code delay and carrier phase advance would not change significantly over time, allowing the receiver to track any slowly varying change without difficulty. On the other hand, ionospheric storms, steep electron density gradients and irregularities, depending on their magnitude and intesity, can cause receivers to lose lock over a wide region and, thus, render GNSS service unavailable [22]. Such widespread outages are mainly triggered by solar flares and geomagnetic storms during high solar activity.

Besides severe space weather causing widespread blackouts, GNSS receivers operating in the equatorial region can experience rapid signal amplitude and phase variations any day after sunset. Such variations are caused by the diffraction and refraction of the radio waves as they propagate through irregular electron density patches in the ionosphere formed over the equatorial region after sunset. This phenomenon is called equatorial scintillation. It can also be viewed as a space-based multipath which can significantly affect the receiver tracking performance by causing simultaneous deep power fades (> 20 dB) and rapid phase variations [23]. While equatorial scintillation is associated with electron density anomalies and can happen any day, it shows seasonal variations and is intensified during solar maxima. Scintillation also occurs in auroral and polar cap regions, being most intense during solar maximum periods. However, equatorial scintillation is of more importance as it affects a latitude band that covers about 50% of the Earth's surface (15 to 20 degrees on either side of the geomagnetic equator), compared to 7% for polar and auroral scintillation [24].

1.4 Tracking Threshold of a Stand-Alone GNSS Receiver

As discussed in the previous section, various sources of GNSS signal degradations affect receiver tracking performance by lowering the C/N_0 ratios. This naturally raises questions about the tracking threshold of a stand-alone GNSS receiver or the lowest

Figure 1.3: Portions of the Trip Testing Signal Outage in a Typical Urban Environment (Minneapolis/St. Paul, MN); Placemarks Show the Outbound Leg of the Test Route, with Thick Solid Lines Indicating Data Loss

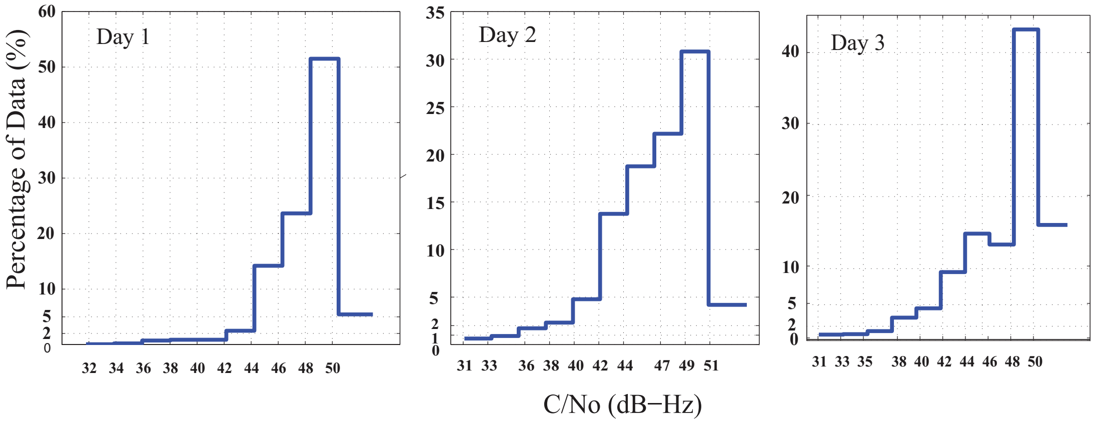


Figure 1.4: Histogram Plots of C/N$_0$ on Three Different Days. It Should be Noted that the Vertical Axis Represents the Percentage of the Data Output by the Receiver when the Signal was Tracked

C/N$_0$ that can be tracked by a receiver. In this context, it is important to note that tracking thresholds of a receiver depend on both receiver dynamics and noise levels and, therefore, vary from one scenario to another. In this section, an approximate tracking threshold of a stand-alone receiver is determined from the results of a drive test which consisted of traveling in an urban environment while recording GPS signal strength. This threshold is representative of that of a stand-alone conventional receiver operating in a typical urban environment. The drive test was performed with a conventional NovAtel receiver installed in a city bus. The trip started from the Minneapolis campus of the University of Minnesota (near the placemark on the left corner of Figure 1.3). The bus traveled to St. Paul, Minnesota and returned to Minneapolis. Portions of the test route are selected from Google Earth and shown in Figure 1.3, with breaks in position fixes indicated by thick solid lines. The data was collected on three different days. Histogram plots of the C/N$_0$ of all tracked satellites as recorded by the receiver on the three days are depicted in Figure 1.4. It is important to note that the vertical axis of the figure represents the percentage of the time when C/N$_0$ was output by the receiver (i.e., the percentage of the time when the signal was tracked). It is evident from the figure that less than 5% of the time the recorded C/N$_0$ was below 40 dB-Hz and no C/N$_0$ was recorded below 31 dB-Hz, indicating that the receiver lost lock on all satellites below 31 dB-Hz. Thus, it can be concluded that the tracking threshold of the NovAtel receiver for this drive test was around 30 dB-Hz. Note that the receiver tracked both carrier phase and carrier frequency. With only carrier frequency tracking the threshold can be expected to be approximately 5 dB-Hz higher i.e., 25 dB-Hz.

Needless to say, this threshold is inadequate for dealing with the GNSS vulnerability discussed previously and naturally calls for enhancement of receiver performance.

1.5 Methods for Improving Robustness

Several methods have been proposed to address the vulnerability of GNSS signals. These include advanced antenna technology such as adaptive antenna array [21]; backing up GNSS systems with inertial sensors [25], optical navigation sensors [26] or other RF signals of opportunity such as cellular [27] and WiFi [28] signals; multi-sensor integration [29], [30], [31]; and other advanced signal processing techniques such as vector tracking architecture [11] which is the subject of the research in this thesis. A brief overview of these methods is given below.

1.5.1 Advanced Antenna Technology

An adaptive antenna array consists of many antenna elements arranged into an array. Since a signal reaches the elements at slightly different times, the element signals sum up constructively or destructively depending on the signal phase difference at the time of arrival (or signal direction). Each element can also be assigned a weight to control its signal phase relative to that of a primary element. Thus, by suitably choosing the weight of each element, the antenna can steer nulls in the direction of a jammer (null steering) or steer maximum gains toward satellites (beamforming). This technique proves effective to mitigate jamming/interference as satellite and interference signals usually come from different directions [21].

1.5.2 Inertial Aiding

Fusion of GNSS and inertial navigation system (INS) is a well-researched topic with much heritage [25]. INS and GNSS, being complementary in nature, can be integrated into one system that retains the benefits of both by correcting/aiding one using the other. The integrated system uses GNSS signals to calibrate INS bias and drift while the INS coasts through GNSS outages for time intervals in the range of few seconds to few minutes depending on calibration and INS qualities. Different GNSS/INS integration architectures have been proposed. They are called loose, tight and ultra-tight (deep) integrations based on the type of data fusion of GNSS and INS. From loose to ultra-tight, better calibration and coasting performance is achieved at the price of independence and computational cost. The INS solution also helps in faster re-acquisition of GNSS signals when satellite position and velocities are known and receiver clock is calibrated. This is because user position and velocity information obtained from inertial sensors reduces the number of code phase and Doppler frequency combinations to be searched for signal re-acquisition. Reference [32] demonstrates INS-aided re-acquisition at C/N$_0$ down to about 10 dB-Hz. Integrated GNSS/INS systems typically find use in moving platforms to bridge signal outages and to sustain high user dynamics.

1.5.3 RF Aiding

Backing up GNSS with other RF systems is a relatively new concept. For example, [27] demonstrates a hybrid positioning method that blends the GPS solution with cellular signal-based position estimates to leverage the complementary benefits of both systems. The hybrid positioning scheme is shown to have approximately (30 - 40)% improvement in 1-σ accuracy and more than 50% improvement in 95% accuracy over either system alone in dense urban environments. In [28] WiFi data is integrated with low cost inertial sensors and a pedestrian dead reckoning system to bridge GPS outages. The integrated system is shown to have (5 - 15) meter indoor positioning accuracy. Reference [31] demonstrates an emergency responder locator which fuses an ultra-wideband (UWB) ranging radio and GPS with low cost dead reckoning systems and a human motion model to achieve a 95% accuracy of 2 m.

1.5.4 Vision Aiding

Vision-based sensors have also attracted signficant attention in recent years for navigation in GNSS-challenged environments. To obtain robust performance, vision data is usually blended with INS measurements. The INS improves feature tracking performance by robust feature matching. The vision data is, on the other hand, used to estimate INS errors. Reference [26], for example, presents a tightly coupled GPS, INS and vision system for precision geolocation. In [29] low cost inertial measurements and image data, when integrated with a barometric altimeter, a magnetic compass and a digital terrain elevation data (DTED) database, are shown to achieve performance equivalent to a navigation grade INS.

1.5.5 Vector Tracking

Over the past decade, vector tracking loops have been the subject much research because they can track weak, attenuated GPS signals, bridge through momentary signal outage and withstand high user dynamics [10]. For example, Figure 1.5 shows the results of a simulation study where real GPS RF signals are post-processed using vector loops operating in a weak signal environment. Different C/N$_0$ signals for this purpose are generated by injecting artificial noise into live GPS data collected from a static RF signal logger. It is evident from the figure that vector loops can track signals at C/N$_0$

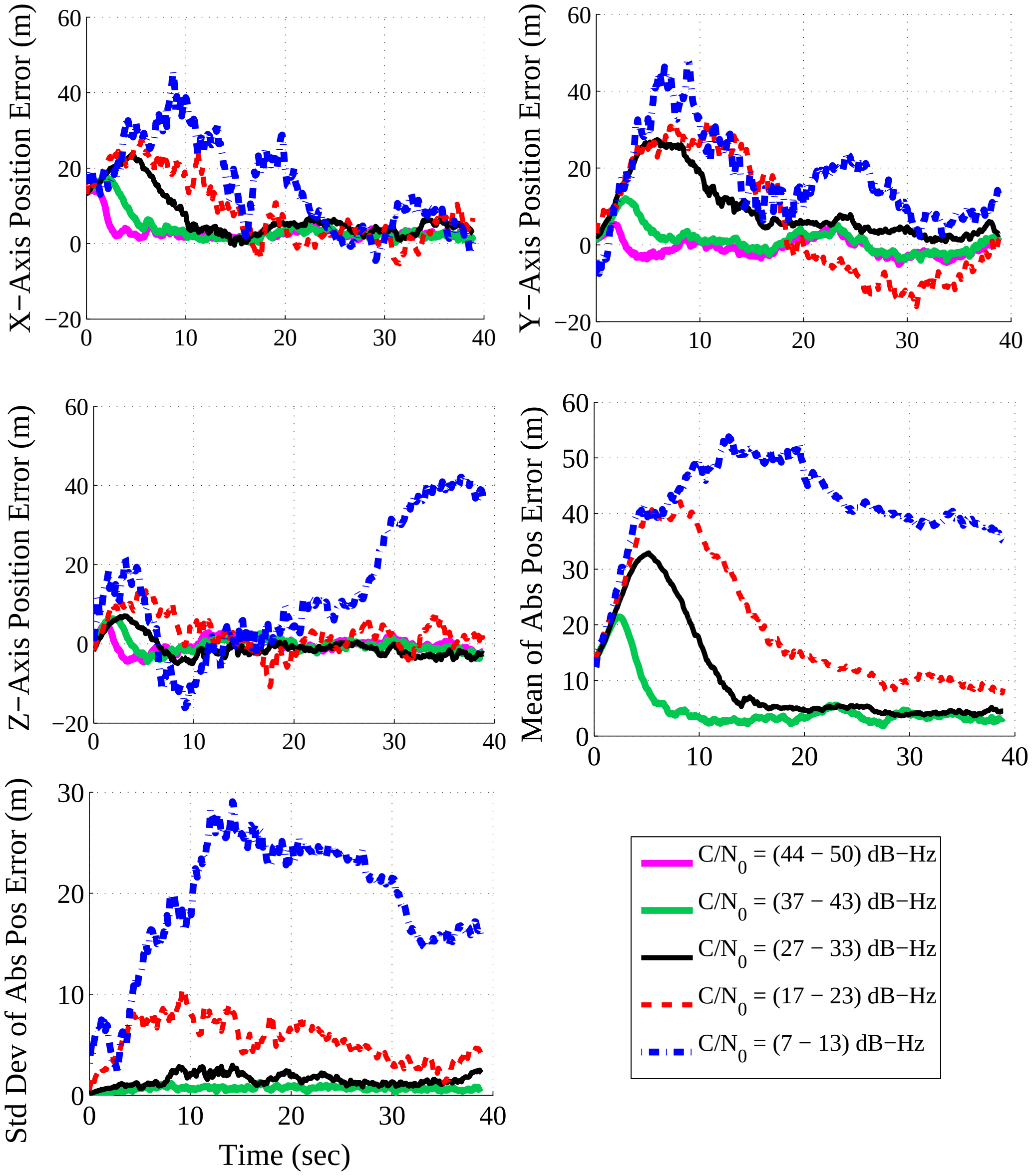


Figure 1.5: Vector Loop Performance with Static GPS Measurements

as low as 7 dB-Hz, although the navigation accuracy degrades with weak signals. It should be noted that this simulation assumes an extreme situation where all satellite signals are attenuated together. This does not happen unless the receiver passes underneath an overpass; through a tunnel; operates indoors; or is being jammed. Therefore, good C/N$_0$ satellites in most cases can be expected to improve the tracking performance of low C/N$_0$ satellites which may result in comparatively low position error. Except for the recorded data with C/N$_0$ in the range of 44 to 50 dB-Hz, the vector loop error statistics (i.e., means and standard deviations of the root sum square of x, y and z axis position errors) are shown in the last two subplots of the figure. The error statistics are calculated from ten Monte Carlo runs whose noises are varied by changing the initial state of the random number generator. The performance of a conventional

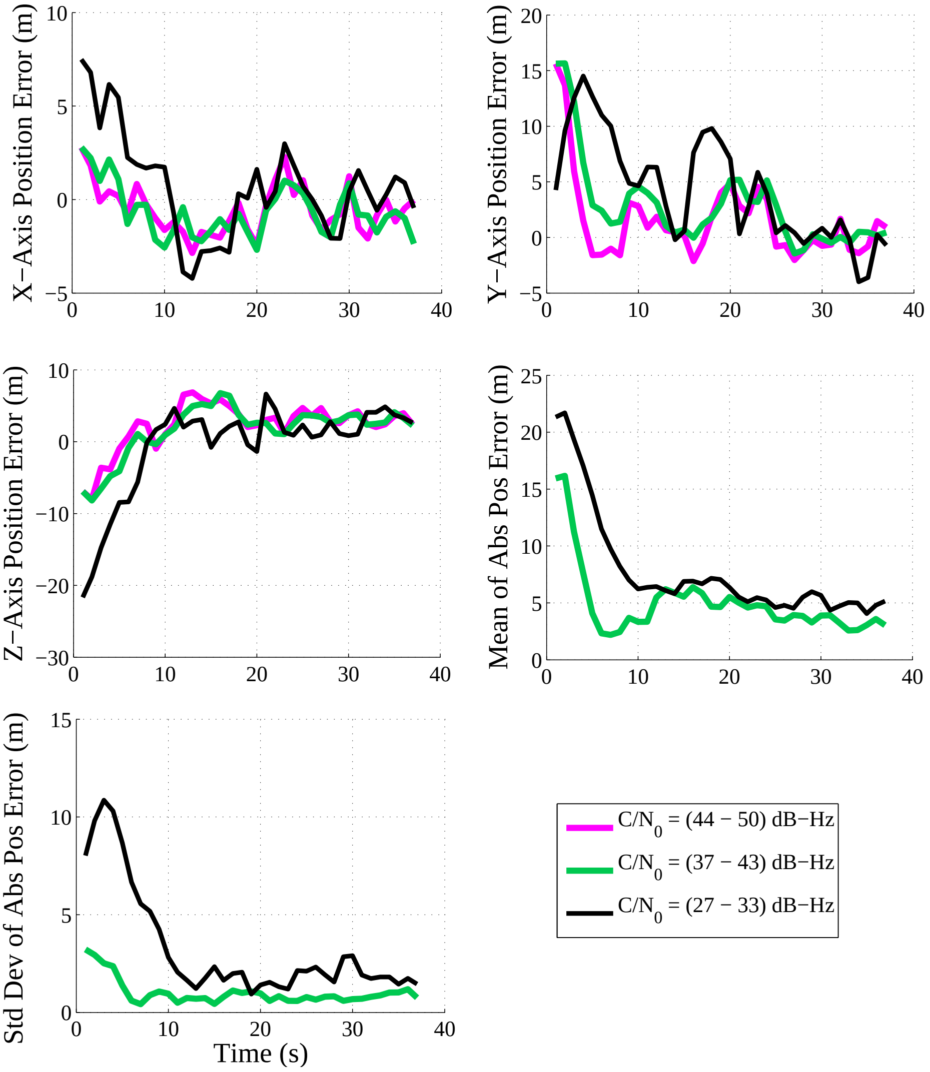


Figure 1.6: Scalar Loop Performance with Static GPS Measurements

scalar loop receiver with the same data sets is illustrated in Figure 1.6. It should be noted that the scalar loop results are not shown for the lowest two C/N$_0$ ranges as almost all satellites lost lock for these data sets, resulting in position error of several hundred meters. This clearly demonstrates the performance benefits of the vector loops with respect to low powered signal tracking. Thus, this simulation reinforces the fact documented in the literature that vector tracking architecture is a promising candidate for dealing with GNSS signal vulnerability.

1.6 Previous Research

The concept of vector tracking was first proposed in [33] in the form of an integrated GPS/INS system where GPS P-code tracking and navigation processing are performed as a single integrated function. In [33] such an integrated design is considered to be superior to a conventional cascaded approach. This is because the integrated design combines the cascaded formulations of tracking loops and navigation filter into a single optimal filter. Thus, the integrated approach exploits correlations among individual tracking errors and eliminates any instability issues that arise from cascaded designs, as concluded by the study in [33]. In [34] a detailed formulation of a coupled architecture is presented to perform combined carrier phase tracking. Simulation results in [34] illustrates that the coupled architecture reduces carrier phase tracking errors during periods of signal fades/blockage caused by aircraft maneuvers. The benefits of coupled tracking are attributed to “inter-satellite path correlations” at the receiver and shown to be dependent on overdetermined geometry.

Although the coupled architectures mentioned above include vector tracking at the core, the term “vector tracking” was coined first in [11]. In this reference, the structure of a combined code tracking methodology is first explicitely described in the form of Vector Delay Lock Loops (VDLL). The work in [11] also highlights the potential advantages offered by VDLL and presents a brief, intuitive explanation of its superiority over scalar loops. The explanation is as follows. Because of joint tracking of all satellite signals, the effective C/N$_0$ along each of the x, y and z axes of user position increases with more satellites. This allows individual satellite signal powers to degrade more while maintaining the same position estimation accuracy. A variant implementation of vector tracking is provided in [35]. In this formulation individual narrow bandwidth tracking loops which are used to track slowly changing LOS errors are aided by a wide bandwidth common loop. The common loop is identical to vector tracking and used to track receiver dynamics and clock drift. This scheme, termed co-op tracking, is shown to enable individual phase locked loops (PLL) to coast through a momentary signal outage without cycle slips and to restore tracking quickly as signal reappears. It is also stated in [35] that in co-op tracking strong signal satellites aid weak signal ones, but no analysis is presented to support it. In [12] the co-op tracking technique is modified by using individual PLLs only for the initialization of vector phase tracking denoted in the study as Vector Phase Locked Loops (VPLL). The scalar PLLs are switched off thereafter. In addition, the research in [12] uses an orthogonal projection method to eliminate clock and atmospheric errors from the carrier tracking error. This is done to increase robustness and to better optimize loop filters. The work in [12] is extended further to perform a brief preliminary study of VPLL-based integrity monitoring to detect multipath on low elevation satellites.

The ability of vector loops to track low powered signals has been documented in [36], [37] and [38], to name a few. In [36] vector loops are used to estimate C/N_0 as low as 10 dB-Hz whereas [37] reports a vector loop sensitivity improvement of about 7 dB over scalar tracking methods. Reference [38] shows a (3 - 6) dB-Hz improvement in interference performance of a VDLL receiver over a conventional receiver. In [10] the explanation provided in [11] is extended further to show the maximum and minimum gains in C/N_0 that can be obtained from vector tracking with various satellite geometries. It is important to note that [10] implements vector delay and frequency locked loops. The work in [10] also uses two implementations of vector loops (position state and pseudorange state formulations) to indirectly show that information exchange among channels is the key to vector tracking's performance benefits. The ability of vector tracking loops to track weak, attenuated GPS signals; to rapidly reacquire blocked signals; and to sustain high maneuvers are also demonstrated. Further, based on the rule-of-thumb tracking thresholds developed for conventional receivers, loss-of-lock metrics for vector loops are developed and validated with simulated data. In addition, the work in [10] compares the performance of a centralized vector architecture and a federated vector architecture with coupled channels between measurement updates to that of a federated vector architecture with decoupled channels between measurement updates. Comparison results show (4 - 5) dB-Hz performance gains of the former two only at high operating C/N_0 ($\sim$ 45 dB-Hz).

Vector tracking loops, when integrated with inertial sensors, result in the so-called GNSS/INS deep integration/ultra-tightly coupled (DI/UTC) architecture whose merits over other integration architectures are well documented in the literature [39] -

[43]. For example, [42] demonstrates continuous carrier phase tracking of GPS signals at 15 dB-Hz when a 90 degree flight turn is executed. The receiver is aided by a micro electromechanical system (MEMS) inertial measurement unit (IMU) using a DI architecture and a coherent integration time of 0.4 sec is used during tracking. A bit estimation algorithm based on maximum energy bit combination is also developed for data bit wipe-off. The carrier phase-based relative position error is illustrated to be within 0.1 m. Reference [43] features a prototype of UTC GNSS/INS pedestrian navigation system (PNS) called DINGPOS. It is capable of fusing GPS L1/L5 and Galileo E1/E5 signals with an IMU including a magnetometer and a barometer, WiFi power readings and Zigbee-based radio navigation systems in DI mode. Field test results using GPS C/A-code signals deeply integrated with IMU and magnetometer data are shown to continue tracking even when seven out of nine satellite signals are below 5 dB-Hz. However, the positioning accuracy degrades significantly at such low C/N_0. Deep integration algorithms have been developed by Aerospace Corporation [44], Honeywell/Rockwell Collins [45], Raytheon [46], L3-Communications [47], Center for Remote Sensing [48], Ohio University [49], the Charles Stark Draper Laboratory [50], University of Calgary [51], QinetiQ [41] and others.

1.7 Motivation

Even though, as noted above, the benefits of vector tracking are well documented in the open literature, comprehensive studies to illuminate its internal workings (i.e., reasons behind its superiority) are lacking. References [33], [34] and [35] provide brief discussions in this regard, but lack any theoretical analysis or simulations to prove their hypotheses. In [11] a brief, intuitive analysis is provided. However, it is not rigorous and its position domain explanation does not easily extend to the tracking loop levels in which performance benefits are usually quantified. Simulation results in [10] show that combined signal processing in vector loops leverages the redundancy of an overdetermined system to attain higher tracking thresholds. In addition, with the help of two different formulations of the vector architecture it reaches the following conclusion. The coupling of channels is the key to vector tracking's superiority. The research also states that coupled channels allow exchange of information among channels through the off-diagonal elements of the Kalman gain matrix. However, it does not delve deeply to thoroughly explain how such information sharing originates from the coupling of channels and from the redundancy of user-satellite geometry and how it can be beneficial. The primary motivation behind seeking such insight is that it may help design an integrity monitoring algorithm specific to the vector architecture. Indeed, it will be explained below that vector loop integrity monitoring requires a different approach than that of the scalar loops owing to the fundamental difference in their tracking methodologies. It is also well known that the benefits of vector tracking are obtained at the cost of complexity. Another motivation for developing such insight is, therefore, as follows: It can help design architectures that retain important benefits of vector tracking and yet are less complex than it. The above deficiency in understanding is addressed in the first part of this thesis from an entirely different, thorough and more intuitive perspective.

As mentioned previously, issues with integrity monitoring of the vector architecture have yet to be explored in great depth. This is important as the vector architecture, by dint of its higher tracking threshold, may provide better system availability in applications (e.g., road user charging schemes [56]) where frequent signal outages compromise the availability of scalar loop-based integrity monitoring. The extensive body of knowledge on receiver autonomous integrity monitoring (RAIM) for scalar architectures [52] - [55], however, is not directly applicable to the vector architecture. This is because, as the first part of this research will show, inter-satellite aiding or internal aiding in vector tracking allows easy propagation of errors (and similarly faults) among loops. Consequently, the pseudorange measurements output from a vector tracking receiver are all affected by a failure in one channel and, thus, violate the basic single fault assumption of the traditional or scalar loop-based RAIM algorithms.

Prior work on vector loop RAIM is scant and [12] and [57] are some of the few works in the open literature dealing with this topic. Although both references use RAIM inputs different from the pseudorange measurements, they have other limitations. An examination of the RAIM scheme presented in [57] reveals the following shortcomings: First, the test statistic (maximum magnitude of the residual inphase and quadrature correlator outputs among all channels) is assumed to be Gaussian distributed when, in fact, it is Rayleigh distributed. Second, in order to compute the vertical protection level (VPL), an approximately linear relationship between the test statistic and the vertical position error is considered based on only simulation results. No theoretical explanation is provided to support such an approximation. In [12], on the other hand, phase discriminator outputs are used to perform RAIM with VPLL. However, the study does not address an important design challenge associated with large noises at the discriminator level. Moreover, its protection level calculation method based on the standard least squares estimation of receiver position does not account for the effects of matrix filters used in vector tracking. It does not also apply to the Kalman filter implementations of vector architecture. Further, the work lacks a thorough performance comparison of scalar and vector loop RAIMs. Consequently, the second part of this thesis is devoted to integrity studies of vector tracking.

1.8 Contributions

In response to the above challenges, this thesis makes the following contributions to the GNSS literature on vector tracking:

1. It provides a thorough explanation for the performance benefits of the vector architecture over its scalar counterparts. In this regard, it recasts the existing complex vector architecture equations into a discrete parametric model that is easy to analyze. Then, it performs an in-depth theoretical analysis of this model by applying the concepts of system theory. This analysis illuminates the internal operations of vector tracking.

2. It provides preliminary guidelines under which scalar and vector architectures can perform comparably. It does this by using the discrete parametric model which facilitates a novel way to derive transfer function models of vector loops under certain assumptions. These transfer function representations are particularly useful for theoretically and numerically computing vector loop noise bandwidths which set the benchmark for their scalar counterparts.

3. It performs in-depth integrity studies of vector tracking. It develops a way to apply ideas from scalar loop RAIM to vector loops. The modified scheme also addresses the limitations of [12] and [57]. With simulation studies and proper design of pertinent scalar and vector loop parameters, it then assesses the integrity monitoring performance of vector loops with respect to scalar loops. Following this, it validates the vector loop RAIM algorithm with GPS measurements generated from a Spirent simulator.

1.9 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 provides a background on the basics of GPS/GNSS signal tracking. These include GPS signal structure, receiver front end processing, signal acquisition, signal tracking and navigation algorithms. Under signal tracking and navigation, details of scalar tracking architecture are discussed. Then, Chapter 3 presents the basics of an advanced tracking loop architecture known as vector tracking. Since vector tracking is the topic of the research in this thesis, this chapter develops the groundwork for analyses that follow in subsequent chapters. In Chapter 4 the vector loop equations are recast into a discrete parametric model from which transfer functions and noise bandwidths of vector loops are derived. A theoretical analysis of the discrete model is also provided in this chapter to explain the benefits of vector tracking. Simulation studies to support the theoretical analysis of the discrete parametric model are then presented. Chapter 5 first provides an overview of the scalar loop RAIM algorithm implemented in this thesis. Then, vector loop RAIM algorithms for coherent and non-coherent architectures are presented. Next, the vector loop RAIM algorithm for the non-coherent architecture is validated with simulation studies. Chapter 6 discusses the validation of the vector loop RAIM algorithm developed in Chapter 5 with GPS measurements. This includes a high fidelity simulation of an aircraft making an instrument approach. Halfway through the approach, the receiver is subject to sub-meter level pseudorange drifts in one satellite channel and elevated noise floor in all channels due to intentional interference. Finally, Chapter 7 summarizes the research carried out in this thesis and provides recommendations for future work.

Chapter 2

Basics of GPS Signal Tracking

2.1 Introduction

The medium Earth orbits of the GPS satellites vary in altitudes from 20193 km at zenith to 25237 km on the horizon when viewed by a receiver on the Earth's surface [58]. While medium Earth orbits were chosen to optimize satellite visibility all over the world, GPS signals traveling from these orbits arrive at the Earth's surface with an extremely low power of 10^{-16} to 10^{-17} Watts. Such weak signals remain buried in noise during the analog signal processing of the receiver front end. Though weaker than noise, the signals are designed to possess a remarkable correlation property by virtue of their signal structure. This property, when exploited by the multi-channel digital signal processor of the receiver, helps acquire multiple satellite signals simultaneously from below the noise floor. It is based on the fact that satellite signals are modulated with *near* orthogonal pseudorandom codes with high auto-correlation peak and low cross-correlation peaks which distinguish one code from another. Code modulation spreads the signal sprectrum, resulting in the name spread spectrum signaling. It also allows all satellites to share the same frequency band to transmit data for position fixing; provides ranging measurements to satellites; and offers some protection against signal reflections. Thus, spread spectrum signaling allows the GPS satellites to offer PNT services worldwide, although the satellite signals arrive at the antenna as mere electromagnetic "whispers" hidden in noise.

This chapter provides an overview of conventional GPS receiver architectures. The following are discussed briefly: GPS signal structure, analog signal processing of receiver front end, digital signal processing of baseband software, signal acquisition, signal tracking and navigation algorithms. This discussion serves as a basis for the mathematical development in subsequent chapters.

2.2 GPS Signal Structure

As illustrated in Figure 2.1, the GPS signals comprise three elements: Carrier, code and navigation data. Each carrier component is usually bi-phase shift key (BPSK) modulated with a bit train. Each bit train is obtained by the modulo-2 sum of a code and navigation data [59]. While the code is used to distinguish one satellite from another, the navigation data carries information about satellite orbits, clocks, satellite status information, etc.. The three signal elements are clocked by the same on-board frequency standard. In the time domain, signals broadcast from a GPS satellite are represented as

$$s(t) = \sum_{j=1}^{n} A_j D(t) x_j(t) \cos[2\pi f_{L_j} t + \phi_{o_j}] \tag{2.1}$$

where the subscript j denotes a carrier signal; A_j is the signal amplitude; D is the navigation data bit; x_j is the code modulating the j^{th} carrier frequency; n is number of carrier components; f_{L_j} is the j^{th} carrier frequency; ϕ_{o_j} is the initial phase of the j^{th} carrier component at time t. The three signal elements are briefly discussed below.

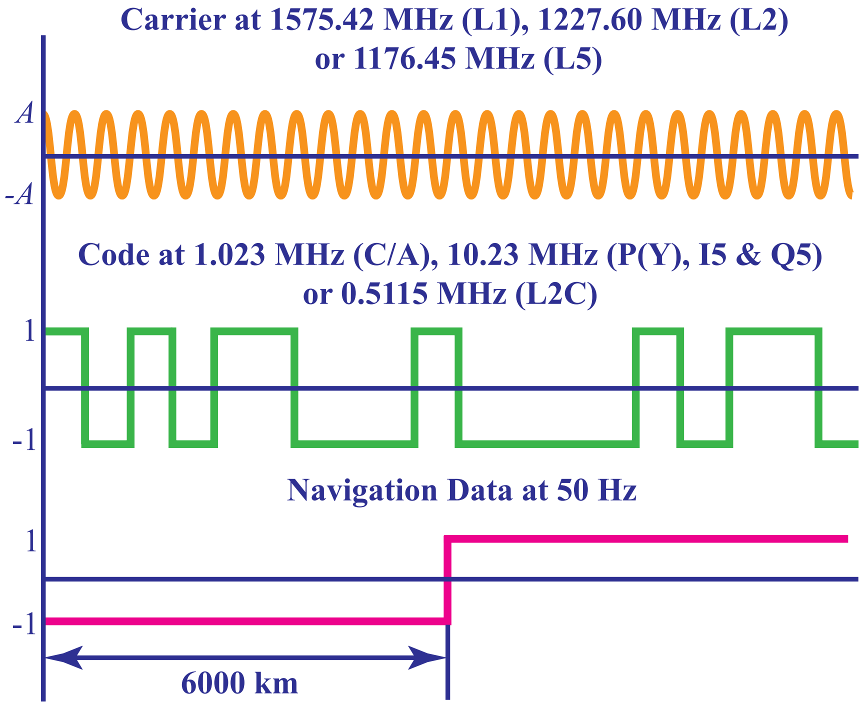


Figure 2.1: GPS Signal Structure

2.2.1 Carrier

GPS satellites transmit carrier signals at three different frequencies in the L-band: L1 (1575.42 MHz), L2 (1227.60 MHz) and L5 (1176.45 MHz). Until 2010, the satellites, which include Block I, II, IIA, IIR and IIR-M, broadcast signals at two frequencies: L1 and L2. Block IIF satellites, launched first in May 2010, introduced a new civil signal at L5. The new signal at L5 was added as part of GPS modernization efforts and will enable civil aviation users to use dual-frequency (L1 and L5) GPS measurements. The impetus for transmitting signals at more than one frequency is primarily two-fold. First, dual-frequency signals allow one to calculate and remove the full ionospheric delays from the GPS measurements. Since the ionosphere is the largest source of error after selective availability was switched off in 2000, a significant improvement of the positioning accuracy can, thus, be achieved [60]. Second, the time required for carrier-phase integer ambiguity resolution and verification is reduced [61].

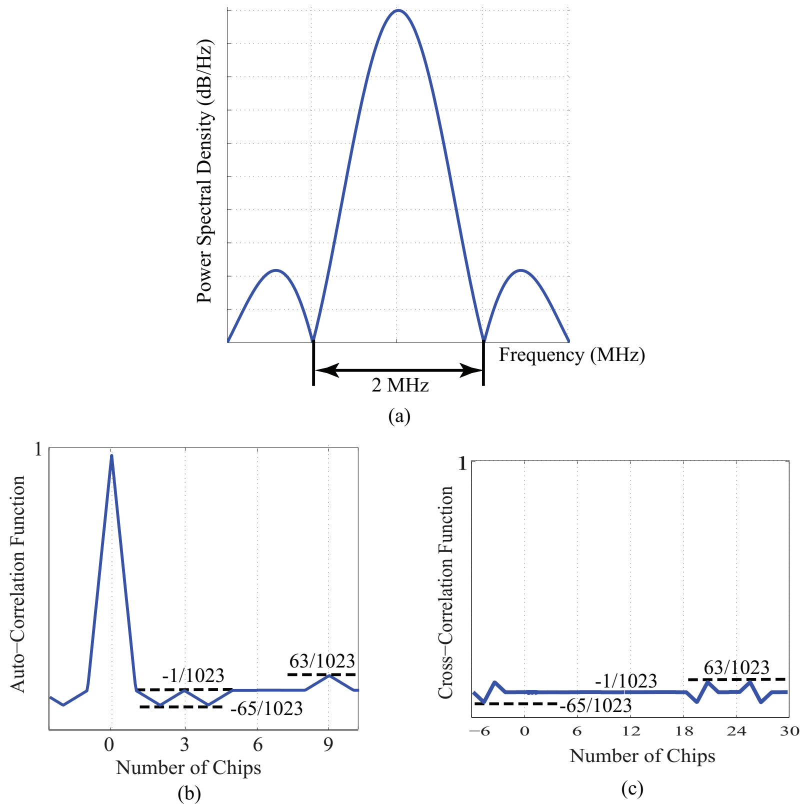
2.2.2 Code

In broad terms, a GPS ranging code consists of a train of rectangular pulse waveforms which is pseudorandom in nature, meaning it appears to be random over a short duration of time, but actually has a deterministic pattern that repeats after a certain time. Such psedorandom noise (PRN) codes are generated by suitably combining the outputs of two or more linear shift registers and have a remarkable correlation property. The property is as follows. A code will result in a sharp correlation peak only when it is correlated with a replica code aligned within one chip with the original code. A chip of a code denotes an elemental rectangular pulse in the pseudorandom sequence. Thus, by transmitting a unique ranging code, a GPS satellite can be distinctly identified from the rest of the satellites. The word “ranging” implies that the code is used to derive range measurements from the satellite to the user.

Three types of ranging codes are transmitted by all blocks of GPS satellites. They are the precision (P) code, providing precision ranging to the US military and US Department of Defense (DOD)-authorized users; the Y-code which is the encrypted P-code (denoted as P(Y)-code) when the anti-spoofing (A-S) mode of operation is in place; and the coarse/acquisition (C/A) code which is used for acquisition of the P(Y)-code and provides coarse ranging for civil applications. In addition, a new military code (M-code) at L1 and L2 is added to IIR-M and subsequent blocks. Two additional ranging codes are transmitted from Block IIR-M onwards: The L2 civil moderate (L2

CM) code and the L2 civil long (L2 CL) code. For IIF and subsequent blocks two new ranging codes are transmitted at L5. They are called the in-phase (I5) code and the quadraphase (Q5) code. Further, a new code will be broadcast at L1 from Block III onwards for civil users.

From Block I through Block IIF satellites, the L1 link is BPSK modulated with two bit trains which are in phase quadrature with each other. The two bit trains are the modulo-2 sum of the C/A-code and navigation data and the modulo-2 sum of the P(Y)-code and navigation data. For Block II/IIA and IIR the L2 link is BPSK modulated by the modulo-2 sum of either the C/A-code and navigation data or the P(Y)-code and navigation data; it is selected by ground command. A second modulation mode can also be selected by ground command which uses the P(Y)-code without navigation data. Since Block IIR-M satellites, the L2 is modulated by the modulo-2 sum of the P(Y)-code with or without navigation data. Besides, one of the following modulation modes is selectable by ground command: The modulo-2 sum of the C/A-code with or without navigation data, the L2 CM-code with data time-multiplexed with the L2 CL-code with no data at a 1023 KHz rate. In addition, both L1 and L2 are modulated by the M-code on IIR-M and subsequent blocks. The modulation method is binary



(a)

(b)

(c)

Figure 2.2: GPS C/A-Code Power Spectral Density and Correlation Functions

offset carrier (BOC) where a 10.23 MHz sine-phased subcarrier modulates the carrier along with the M-code and navigation data.

The P-code is of seven days in length at a chipping rate of 10.23 MHz. The C/A-code is much shorter, repeating every 1 ms with a chipping rate of 1.023 MHz. The time period of a code is called code epoch. The power spectral density (PSD) of a C/A-code chip is illustrated in Figure 2.2 (a). It is a sinc function with a null-to-null bandwidth of 2.046 MHz which spans the main lobe of its PSD spectrum. Though not shown in the figure, the PSD of the P-code is also a sinc function, but has 10 times as large bandwidth. It is instructive to note that when a code modulates a sinusoidal carrier, the code spectrum is shifted up and down to ±carrier frequency. Thus, the code spreads the impulse spectrum of the carrier signal. The auto-correlation function of a C/A code and its cross-correlation function with another C/A code are shown in Figure 2.2 (b) and (c), respectively. The auto-correlation function takes on four values (-65/1023, 1/1023, 63/1023 and 1) whereas the cross-correlation function has three values (-65/1023, -1/1023 and 63/1023) when the delay is an integer number of chips. The distinct main peak of the auto-correlation function allows GPS receivers to acquire signals by aligning a receiver-generated replica code with the incoming code. The narrower and steeper the main peak is, the better is the alignment of the replica code with the received code and, hence, the code ranging precision. For example, GPS receivers can measure C/A-code phase with a ranging precision of about 0.5 m and it is much better with the P-code [58]. The sharp main peak also helps distinguish the correlation (or shadow) peaks of the reflected signals that are delayed from the direct signals by more than one chip. However, for short delays shadow peaks mix with the main peak and distort it. Details of generating GPS codes can be found in [59].

The L2 CM-code is 20 ms in length while the L2 CL-code is 1.5 sec long, both having a chipping rate of 511.5 KHz. Both I5-code and Q5-code are transmitted at 10.23 MHz and 1 ms long. They are modulated with a 10-bit and a 20-bit Neuman-Hofman codes clocked at 1 KHz, respectively. It should be noted that new codes with higher chipping rates and longer periods are added for higher accuracy measurements and better correlation properties, respectively, to facilitate navigation in weak signal environments such as dense foliage, office buildings, etc. [60]. In addition, next generation GNSS signals will be transmitted at higher power levels to penetrate areas where current signals cannot be used reliably. The power levels of current C/A-code and P-code signals are -130 dBm (L1), -136 dBm (L2), -133 dBm (L1) and -136 dBm (L2), respectively.

2.2.3 Navigation Data

Similar to the PRN codes, the navigation data is made up of a sequence of bi-phase rectangular pulses known as bits. Each navigation bit is 20 ms long. A full navigation data set lasts 12.5 minutes. It is structured as follows. Thrity data bits form a word. Ten words make a subframe and five subframes make a page. Twenty five pages, also called a superframe, result in a complete navigation data set. The first two words of each subframe are the telemetry (TLM) word and hand over word (HOW), respectively. The TLM word has an eight-bit preamble which is used to detect the beginning of a subframe. The HOW contains the GPS time of week (TOW) calculated from Saturday midnight/Sunday morning. Subframe 1 contains information about the week number, user range accuracy (URA), satellite health, issue of data, clock (IODC), estimated group delay differential Tgd and clock correction parameters. Subframes 2 and 3 have the ephemeris parameters. They also include a parameter called the issue of data, ephemeris (IODE). When the IODE does not match the last eight bits of the IODC, it indicates that a data set cutover has occurred and new data must be used. While subframes 1 through 3 are repeated in all 25 pages of a superframe, subframes 4 and 5 have 25 versions, with each version transmitted in one page. Subframe 4 of page 18 includes ionospheric correction parameters and universal coordinated time (UTC). Subframe 4 of pages 2 through 5 and 7 through 10 has almanac data for satellites 25 through 32, respectively. The remaining versions of subframe 4 are either spares or reserved. Subframe 5 contains almanac data of satellites 1 through 24 (pages 1 - 24) and health information of all satellites (page 25). Reference [59] describes the details of navigation data formats.

2.3 Receiver Front End Processing

As alluded to earlier, satellite signals arrive at the receiver antenna with extremely low power and, therefore, need to be conditioned by an analog front end prior to digital baseband processing. The objectives of receiver front end processing are four-fold. First, the signal plus noise power is amplified by approximately 100 dB to raise it to a level within the working range of the analog to digital converter (ADC). Second, the signal bandwidth is gradually narrowed to filter out out-of-band noise and interference which would otherwise saturate the front end; reduces the ADC dynamic range; and-degrade overall performance. Third, the L band carrier frequency is down converted to an intermediate frequency (IF) which can be easily handled during front end process-

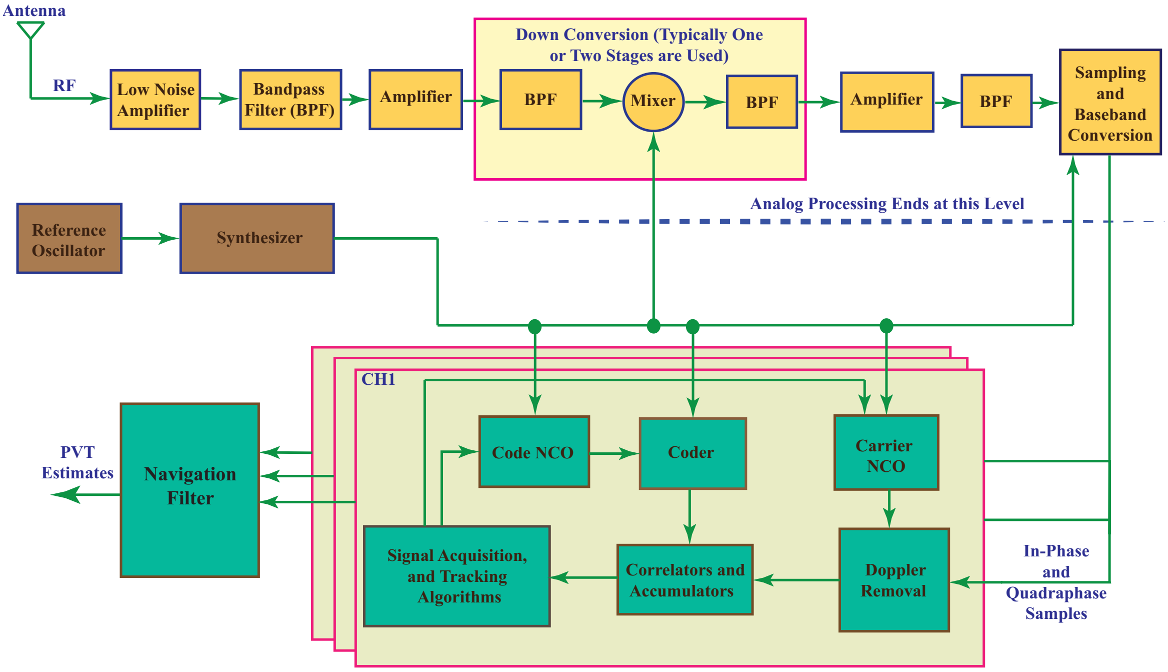


Figure 2.3: High Level Schematic Diagram of a Conventional GPS Receiver

ing. Fourth, the signal along with noise is converted from analog to digital form for subsequent digital processing. A high level functional block diagram of a GPS receiver is depicted in Figure 2.3. The functional blocks that perform front end processing are briefly described next.

2.3.1 Receiver Antenna

A GPS antenna must be able to capture satellite signals coming from all directions above the user's horizon. For this purpose, it is designed to be omdirectional in nature, with its gain being constant across azimuth. The antenna gain, however, varies as a function of elevation angle to provide some protection against RF interference and multipath which represents indirect LOS signals reaching the antenna after reflecting off nearby structures. This means that a signal with a given elevation angle will be received by the antenna with the same gain regardless of its azimuth. A typical antenna has a gain of approximately +4.5 dBic at zenith which gradually decreases by 8 to 10 dB or more at an elevation angle of 5^o [58]. dBic refers to the directive gain of an antenna in decibels with respect to that of an isotropic antenna with the same polarization characteristic. Sophisticated antennas are also available for applications which require additional resistance against interference and multipath.

2.3.2 Low Noise Amplifier and Noise Figure

An active antenna is integrated with a low noise amplifier (LNA) housed in the same unit. A passive antenna, on the other hand, does not include an LNA which is externally connected to the antenna unit using a coaxial cable. An LNA is designed for a high gain ($\sim$ 30 dB) and low internal noise. Being the first component of the front end with a high gain, it determines the effective noise figure of the entire front end, as shown by Friis' formula [58]. Noise figure is defined as the ratio of input C/N_0 to output C/N_0. It can be shown to be related to the ratio of the effective noise temperature to the input noise temperature. While specifying the noise figure of a radio device, the input noise temperature is assumed by manufacturers as room temperature or 290 K. Noise figure, therefore, gives a measure of the effective noise temperature of a subsystem which is equal to the internal noise temperature divided by the subsytem gain. Thus, it represents the degradation of C/N_0 at the output. If the gain of a subsystem is high i.e., it is an amplifier, the degradation is low. If the subsystem is a passive element such as a cable or a filter, the gain is less than one. For these components, output C/N_0 is degraded more due to internal loss. For example, the noise figure of an LNA is (2 - 3) dB and that of a cable that follows the LNA can be as high as 10 dB [58]. However, the noise contribution of any subsystem following the LNA is insignificant. This is because its effective noise temperature is scaled down by the LNA and all other preceding component gains when calculating the effective noise figure of the entire front end. Thus, the LNA not only amplifies weak signals, but also improves the front end noise figure.

2.3.3 Bandpass Filters

The front end includes a number of bandpass filters to gradually narrow down the signal bandwidth to eliminate out-of-band noise; to filter out image frequencies that are aliased down to the IF or baseband during mixing or sampling; and to reject spurious frequencies generated by the amplifiers and mixers. They also generate insertion losses, thereby contributing to the noise figure of the front end, although their noise contributions are minimal. The final bandpass filter of a C/A-code receiver has a typical one-sided bandwidth in the range of 1.25 MHz to 3 MHz. It should be noted that filtering also results in loss of signal power during correlation by truncating the sidelobes of the incoming code spectrum. However, with wideband noise the loss is within 0.5 dB if the final one-sided bandwidth is larger than the code chipping rate [63].

2.3.4 Amplifiers

As both signal and noise are much weaker than the level that can activate the ADC, the front end amplifies the received signal by approximately 100 dB. This is accomplished by judiciously distributing the amplifiers at the various stages of the front end processing. An amplifier is generally followed by a bandpass filter which removes any undesired frequencies generated by the amplifier.

2.3.5 Frequency Down Conversion

During down conversion, the L band carrier frequency is down converted to a suitable IF that can be easily handled by subsequent stages of the front end. The IF is chosen according to the receiver frequency plan. Down conversion is performed in one or two stages by mixing (or multiplying) the incoming signal with a local oscillator-generated reference signal, as described below.

The received satellite signal is expressed as

$$r(t) = \sum_{i=1}^{m} A_i D_i(t - t_{\tau_i}) C_i(t - t_{\tau_i}) \cos[2\pi(f_c + f_{d_i})t + \phi_{o_i}] + n(t) \tag{2.2}$$

where the received carrier signal cos(.) of amplitude A and initial phase ϕ_o is modulated with a pseudorandom code $C_i(t - t_{\tau_i})$ and binary data bit $D_i(t - t_{\tau_i})$. t_{τ_i} is the signal transit time from satellite i to the receiver. f_c and f_d represent the carrier and Doppler frequencies, respectively. $n(t)$ is additive noise. m is the number of visible satellites. The received signal is multiplied in a mixer by the reference signal $2\cos(2\pi f_1 t)$. The output of the mixer is given by

$$\begin{aligned} r(t) \times 2\cos(2\pi f_1 t) = & \sum_{i=1}^{m} A_i D_i(t - t_{\tau_i}) C_i(t - t_{\tau_i}) (\cos[2\pi(f_c + f_1 + f_{d_i})t + \phi_{o_i}] \\ & + \cos[2\pi(f_c - f_1 + f_{d_i})t + \phi_{o_i}]) + n(t) \end{aligned} \tag{2.3}$$

Thus, the mixing process produces an upper sideband and a lower sideband, as given by the first and second terms of the right hand side of Equation 2.3, respectively. The upper sideband is eliminated via a bandpass filter that follows the mixer. As noted previously, mixing generates harmonics which are also removed by this filter. In addition, a filter generally precedes the mixer to remove any image frequencies that would mix to the lower sideband and cannot, therefore, be removed at IF [62]. It should be noted that if there are more than one stage of down conversion, they are evenly distributed in the RF chain, meaning down conversion alternates with amplification.

2.3.6 Sampling

After front end conditioning, the analog signal is sampled by a sample and hold circuit. Bandpass sampling is performed in modern receivers which also aliases the IF into the baseband. Thus, bandpass sampling combines the sampling task with down conversion. Another advantage of bandpass sampling is that the sample rate is mainly governed by the signal bandwidth, not by its highest frequency component as in baseband sampling [62]. The sampling frequency of a receiver is determined by the receiver frequency plan, IF and the front end bandwidth. The sampled signal is discrete in time, but continuous in magnitude. It is then quantized and digitized by the ADC. The rounding or truncation error between the analog input to the ADC and the digitized output is called quantization error. The degradation caused by the quantization error is minimum when the signal plus noise is spread over full quantization levels i.e., the ratio of the maximum analog to digital (A/D) threshold to RMS noise level varies between one and two. The minimum value of the degradation ($<$ 1.5 dB) decreases with increasing number of bits and higher front end bandwidths. Modern receivers use an automatic gain control (AGC) in front of the ADC to drive noise to the A/D threshold levels or vice versa for quantization level control. The AGC also helps increase the dynamic range and prevents interference from capturing the sampling process [63].

2.3.7 Conversion to Baseband

Conversion to baseband refers to the process of converting the IF signal to the in-phase and quadraphase components of the signal envelope which remain modulated with code, Doppler and data bits. It can be carried out by analog mixing before sampling; during sampling (IF sampling); or after sampling i.e., during digital signal processing. In the first two methods an intentional frequency offset may exist at baseband in addition to the Doppler frequency to simplify front end frequency plans [63]. If baseband conversion is performed before A/D conversion, then two ADCs are required, one for each of the in-phase and quadraphase arms. If the in-phase and quadraphase components are generated during sampling then only one ADC is required. There are also no gain and phase imbalances as both components are generated in the same circuitry, unlike analog baseband conversion. However, the ADC needs to sample at a rate twice the sampling frequency. While hardware receivers can implement either of the three methods, software receivers perform baseband conversion as part of the digital signal processing since they receive digital IF samples as inputs from the front end.

2.3.8 Reference Oscillator and Synthesizer

As illustrated in Figure 2.3, a reference oscillator drives a synthesizer which, in turn, generates receiver frequency plans using custom chips or programmable dividers [63]. Frequency plans include generating local oscillators for down conversion and analog baseband conversion and generating local clocks for digital signal processing circuitry. Thus, the analog front end and multi-channel digital baseband circuits are run by the same reference oscillator. Three types of reference oscillators are used in GNSS receivers: Atomic clocks, temperature compensated crystal oscillators (TCXO) and oven-controlled crystal oscillators (OCXO). TCXOs and OCXOs are more common in GNSS receivers than atomic clocks because of their lower price and size. In TCXO the output signal from a temperature sensor is used to vary a reactance in the oscillator circuit to provide a stable clock signal. In OCXO, all temperature sensitive components including the crystal are kept in an oven whose temperature is maintained at a level where the crystal frequency versus temperature curve has a zero slope [64]. While an OCXO can provide more than one thousand times better frequency stability than a TCXO, its power, size, warm up time and price are more than those of a TCXO. The frequency stability of a crystal oscillator is characterized by long term systematic variations and short term random fluctuations. Systematic variations are caused by aging and measured per hour, month or day. On the other hand, random fluctuations are triggered by noise sources such as thermal, shot and flicker noise and significant over few seconds or minutes [65]. Random variations are modeled in the frequency domain by PSD functions and in the time domain by the Allan variances [66].

2.4 Multi-Channel Digital Signal Processing

After sampling and discretization, the digital signal processing commences. It is split into multiple channels, with each channel tracking one satellite. The functions that are discussed next are identical to all channels.

2.4.1 Doppler Removal

This process, when performed simultaneously with baseband conversion, is called carrier wipeoff as the output signal after Doppler removal has a frequency independent of the carrier frequency (or IF) plus Doppler. Its frequency is equal to the difference between the true Doppler (f_d) and the receiver's estimate of the Doppler ($\tilde{f}_d$). After

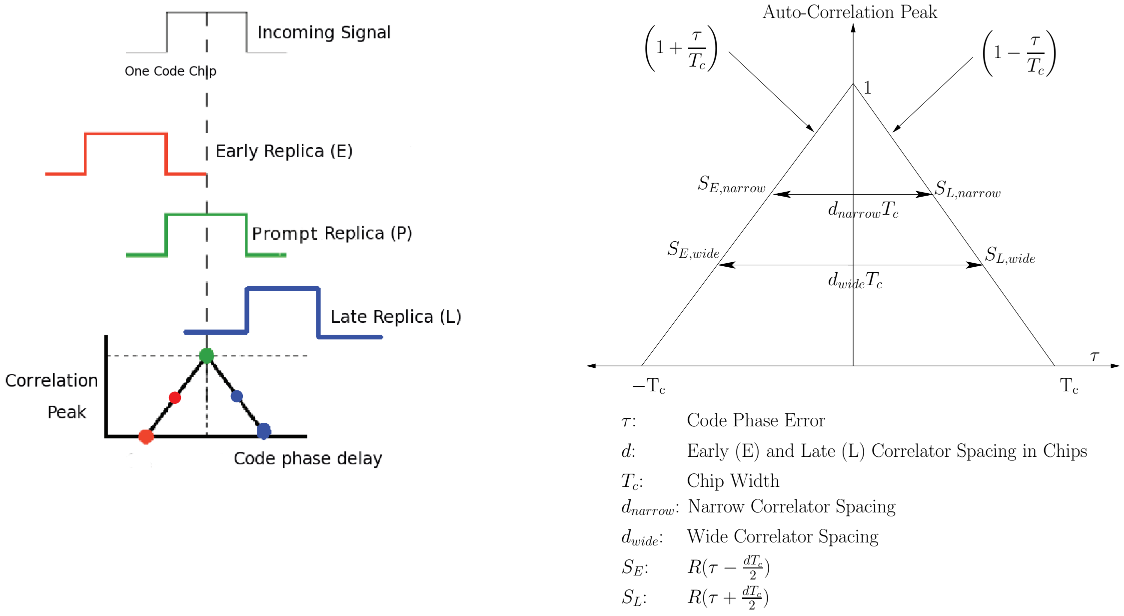
Doppler removal, the outputs of the i^{th} in-phase and quadraphase channels are given as $A_iD_{i,\,k}C_{i,\,k}\cos[2\pi(f_{d_{i,\,k}} - \tilde{f}_{d_{i,\,k}})t_k + \Delta\phi_{o_{i,\,k}}]$ and $A_iD_{i,\,k}C_{i,\,k}\sin[2\pi(f_{d_{i,\,k}} - \tilde{f}_{d_{i,\,k}})t_k + \Delta\phi_{o_{i,\,k}}]$, respectively, where t_k is the k^{th} time instant. $\Delta\phi_{o_{i,\,k}}$ is the phase error at the beginning of time t_k.

2.4.2 Code Wipeoff and Correlation

Carrier wipeoff is followed by code wipeoff. During the code wipeoff process above in-phase and quadraphase signal samples are multiplied by the corresponding replica code samples and accumulated over time. M accumulated samples are coherently added over an integration time T_{coh} to form the in-phase (I) and quadraphase (Q) correlator outputs. The mathematical expressions of a correlator pair at time epoch k and for the i^{th} satellite are given below [63]

$$\begin{aligned}
I_{m_{i,\,k}} &= \frac{A_i}{\sqrt{2}} D_{i,\,k} R\left(\tau_{i,\,k} + \delta T_c\right) \sum_{j=1}^{M} \cos[2\pi(f_{d_{i,\,j}} - \tilde{f}_{d_{i,\,j}})t_j + \Delta\phi_{o_{i,\,j}}] + \eta_{i,\,k} \\
&= \frac{A_i}{\sqrt{2}} M D_{i,\,k} \frac{\sin(\pi\Delta f_{d_{i,\,k}} T_{coh})}{\pi\Delta f_{d_{i,\,k}} T_{coh}} R\left(\tau_{i,\,k} + \delta T_c\right) \cos(\Delta\Phi_{i,\,k}) + \eta_{i,\,k} \\
Q_{m_{i,\,k}} &= \frac{A_i}{\sqrt{2}} M D_{i,\,k} \frac{\sin(\pi\Delta f_{d_{i,\,k}} T_{coh})}{\pi\Delta f_{d_{i,\,k}} T_{coh}} R\left(\tau_{i,\,k} + \delta T_c\right) \sin(\Delta\Phi_{i,\,k}) + \eta_{i,\,k} \qquad (2.4)
\end{aligned}$$

where A = amplitude of the digital signal; M = number of signal samples in a correlation interval; D = navigation data bit; T_{coh} = coherent integration time; Δf_d = Doppler frequency error (f_d - $\tilde{f}_d$); τ = code phase error (estimated code phase - true code phase); $\Delta\phi$ = carrier phase error at the end of T_{coh}. η is the correlator noise which is normally distributed with mean zero and variance σ^2_{IQ}. The noise statistics remain the same across all correlators. σ^2_{IQ} is derived in Appendix A. The derivation relaxes the white noise assumption [63] of the bandlimited pre-correlation noise and takes into account correlations among adjacent noise samples. However, as the pre-correlation noise time constant is much smaller than the coherent integration time, η is assumed to be white. The subscript m stands for E, P and L, denoting early, prompt and late correlators, respectively. The receiver internally generates three versions of the replica PRN code for signal tracking. The prompt code is aligned with the received code when a satellite signal is tracked whereas the late code is shifted behind and the early code is shifted ahead of the prompt code, as illustrated in Figure 2.4 (a). Each code, when correlated with the in-phase and quadraphase signal samples, generates one correlator pair. Thus, three correlator pairs (early, prompt and late) are generated.



(a) (b)

Figure 2.4: BPSK Code Auto-Correlation Peak for Infinite Pre-Correlation Bandwidths

Details of how satellite signals are tracked with these correlator pairs will be discussed later in this chapter. The code phase offset δ can take on three values: $-\frac{d}{2}$, 0, $\frac{d}{2}$ for E, P and L, respectively. d is the early and late correlator spacing in chips. T_c is the chip width. R is the code auto-correlation function. Figure 2.4 (b) depicts the ideal code auto-correlation peak of an infinite bandwidth GPS receiver whose mathematical expression is given by

$$R = \begin{cases} 1 - \dfrac{|\tau|}{T_c} & \text{if } |\tau| < T_c \\ 0 & \text{otherwise} \end{cases}$$

It should be noted that a finite pre-correlation bandwidth tends to round the corners of the correlation peak. However, its effect is negligible if the pre-correlation bandwidth is several times the code chipping rate which is feasible in the case of the C/A-code. But it is generally avoided with the P(Y)-code as front end bandwidths several times the P(Y)-code chipping rate would make the receiver very susceptible to RF interference. The chip of a band-limited signal can be approximated as a trapezium. The correlation function under the trapezium expression can be found in [67].

Equation 2.4 reveals that the post-correlation signal power of a channel is approximately $A^2M^2/2$ while it is $A^2/2$ prior to correlation (assuming the pre-correlation bandwidth $B >> 1/T_c$ [63]). On the other hand, if $N_0/2$ is the PSD of the digital noise samples, then the pre-correlation noise variance is $N_0B/2$. The post-correlation noise variance is shown in Appendix A to be $N_0M^2/(2T_{coh})$. For the receiver used to collect C/A-code signals for experimental validation in Chapter 6, A of a channel at some time instant is 200; sampling frequency, $f_s = 43.428$ MHz; $B \approx 21$ MHz; if T_{coh} is equal to one C/A-code epoch i.e., 1 ms, then $M = 43428$; $N_0/2 = 1.795$. Substituting these values, pre-correlation and post-correlation signal to noise ratios (SNR) are obtained as -31 dB and 11 dB, respectively. Thus, the signal that was below the noise floor before correlation is raised above noise by exploiting the signal correlation property, as elucidated earlier in this chapter. In other words, correlation serves as a noise filter by amplifying the signal power more than that of the noise or by reducing the post-correlation bandwidth.

When the prompt code is aligned with the received code, the auto-correlation function of the early correlator stays on the rising edge of the correlation peak and that of the late correlator on the falling edge of the peak, as is evident in Figure 2.4. With a perfectly aligned prompt code, the early and late correlators are equal in magnitude in the absence of noise. Consequently, the difference between the two correlator powers is used by the receiver baseband software to continuously align the prompt code with that of the incoming signal and forms the basis for code phase tracking. It will be addressed in detail when discussing signal tracking algorithms.

The early and late correlator spacing d is an important design parameter and typically takes values between zero and one chip width. A wide correlator spacing is around one chip width (or $d = 1$). Narrow correlator spacings with d less than one are also used. Narrow spacings improve the code ranging precision because the variance of the code ranging error is directly related to d [58]. In addition, narrow correlators are inherently more robust to multipath-induced peak distortions because of the following reason. Additional delay of the multipath signal causes its correlation peak to shift to the right of the direct-ray correlation peak. As a result, the lower regions of the direct signal correlation function are mostly distorted by the multipath signals, with the peak being impacted less. Narrow correlator spacings can, therefore, be selected to operate above the largely distorted portion of the composite correlation peak which may not be possible with a wide correlator. However, narrow correlators require wide pre-correlation bandwidths to operate near the peak. Otherwise, the rounded corners of the correlation peak would mute their advantages. This requirement discourages the use of narrow correlators in P(Y)-code receivers [58].

It is important to note that the coherent integration time (T_{coh}) of the correlation process is determined by four factors: Signal coherence, navigation data bit boundaries, C/N$_0$ and tracking loop bandwidths [67]. In simple words, coherent integration implies that the signal phase error is not eliminated during the integration process. Therefore, changing signal phase error during correlation causes the summed samples to interfere with each other which results in the sinc function of Equation 2.4. In order to minimize the loss in signal power due to the sinc term, a constant phase error must be maintained. This is termed signal coherence. It may not be fully attainable in practice, however. To limit the signal power loss to a factor of 2, the following condition is imposed: $T_{coh} < 0.443/|\Delta f_d|$, where Δf_d is the Doppler error [67]. This implies that T_{coh} should be small at the beginning of tracking loop closure and increased later on as time progresses and the Doppler error is reduced. Coherently integrating signal samples over multiple data bits also leads to signal power loss and degraded acquisition and tracking performance due to data bit sign changes. In addition, correlator outputs spanning multiple data bits preclude navigation message demodulation. This results in data bit durations placing a limit on the maximum value of T_{coh} for data-carrying signals without bit wipeoff. While longer coherent integration is desirable at low C/N$_0$, it is not permissible beyond a data bit interval or beyond a point where signal coherence fails (whichever is minimum). As a result, common practice is to integrate low C/N$_0$ signals non-coherently after coherent summation. Non-coherent integration sums the samples $I^2 + Q^2$ and, thus, eliminates signal phase error and data bits, albeit at a price of larger noise than coherent summation. Finally, $1/T_{coh}$ should be at least twice or larger than the tracking loop bandwidths to satisfy the Nyquist frequency criteria. The correlator outputs are used for signal acquisition and tracking. In what follows, signal acquisition and tracking methodologies are briefly discussed.

2.5 Signal Acquisition

In order to track a GPS satellite, the receiver must have knowledge of the following signal parameters: The signal code phase at the time of transmission, Doppler frequency (and received carrier phase in the case of carrier phase tracking). Once these quantities are known with sufficient accuracy, the receiver decodes the navigation data from the prompt correlator outputs. It retrieves satellite ephemeris, clock parameters, status information, etc. from the navigation data. When operating in a weak signal environment where decoding the navigation message is not feasible, the receiver can receive this information via an external radio link upon request. This is called assisted-GNSS which is finding increasing use in urban environments. Examples include phone-based and in-car-based navigation systems [60]. The tracked code/carrier phase and Doppler frequencies are used by the receiver to form the range and range rate measurements to the satellites. Thus, with four or more satellites in view, the receiver can trilaterate its position and correct its clock drift using the measurements and satellite data.

The estimation of above three signal parameters is carried out in two phases: Acquisition and tracking. Acquisition is a one-time process that acquires the code phase and Doppler frequency with coarse accuracy through a global search process. Tracking, on the other hand, continuously estimates and fine-tunes these parameters until the satellite signal is lost in the event of which signal has to be re-acquired before tracking commences. It should be noted that the carrier phase cannot be acquired. A coarse estimate of the Doppler frequency is generally sufficient for initiating the carrier phase tracking operation in good signal environments.

During acquisition, the receiver conducts a search over a two-dimensional search space comprised of the Doppler frequency and code phase error (f_d, τ). Without any prior information about satellite and user positions, it has to examine all PRN codes and all possible combinations of the states (1023 chips for the C/A code) of each PRN code and the Doppler frequencies (approximately ±10 KHz). This is denoted as cold start during which the receiver may take several minutes to find visible satellites. However, after four satellites are acquired and the satellite almanac is recovered, it can determine which satellites are visible. When the receiver knows the user GPS time, satellite almanac and its position within 100 km of last fix, but not the satellite ephemeris, a warm start is executed. During a hot start, all of the time, satellite almanac and ephemeris and receiver position (within 100 km of last fix) are known. This is possible if the receiver is powered up within (2 - 3) hours of its previous operation, provided it is not transported hundreds of miles when switched off.

Most receivers search the code phase in increments of 0.5 chips and the Doppler frequency in steps of few tens to few hundreds of Hz. Acquisition can be performed in both time and frequency domains. In the time domain the magnitude (or envelope) of the prompt I and Q correlator outputs is determined and compared to a threshold for each possible combination of τ and f_d. The threshold is computed based on the desired probability of false alarm and the receiver's estimate of the noise floor. One technique for estimating the rms noise is to correlate the signal with an unused PRN code. The signal integration time or dwell time T_{coh} per search bin can vary from less than 1 ms to

20 ms depending on the expected C/N$_0$. A typical search sequence is as follows. First, the baseband process selects a Doppler frequency. Keeping the Doppler constant, it then varies the code phase states from the first state to the last and computes the correlation envelope for each state/search bin. If the envelope sample of a search bin exceeds the threshold, the corresponding up/down counter for that bin is incremented by one. Thus, after the last code phase state is reached, the Doppler frequency is shifted to the next bin in the search pattern. This process is repeated every T_{coh} sec in the case of a multiple trial search. The signal is detected in a bin if its counter reaches a maximum value determined by simulation. This value is a trade-off between search speed and probability of detection [68]. It should be noted that a variable dwell time search is faster and superior to a fixed dwell time search. When the signal is declared present, additional high resolution search follows to find the code phase and Doppler frequency. Some receivers use a serial search process for acquisition while others employ a parallel or multi-correlator process to improve acquisition times. It should be noted that low C/N$_0$ signals can be acquired by searching multiple code phase and Doppler frequency combinations in parallel over longer integration periods.

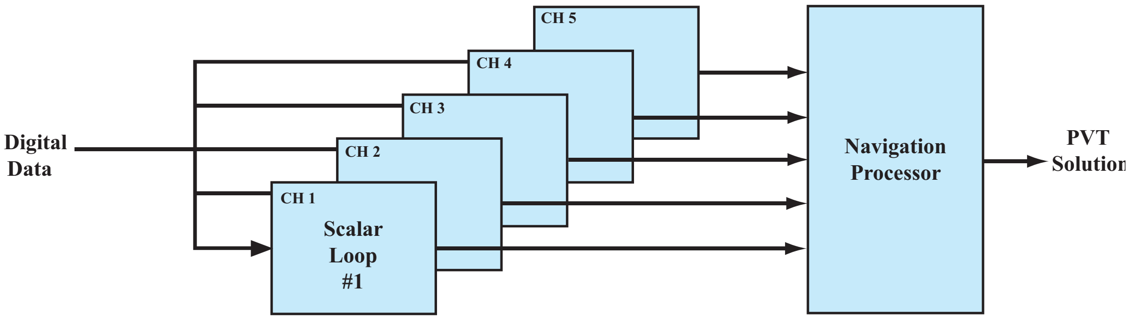
Modern GNSS receivers including software radios also perform a fast search process using frequency domain methods. In this approach first the input signal samples of length one code epoch or higher (denoted as $\{x_n\}_{n=1}^{k}$) are transformed to the frequency domain using the fast Fourier transform (FFT). k is the number of samples. The FFT of $\{x_n\}$ is represented as $\{X_n\}_{n=1}^{k}$, where half of k frequencies contains useful information; the other half is redundant. Then the complex conjugate of $\{X_n\}$ is taken and the output becomes $\{X_n^*\}$. The receiver also generates a set of complex reference signals $\{\tilde{x}_{n,\,i}\}_{n=1}^{k}$. Each set denoted by the second subscript i in $\tilde{x}_{n,\,i}$ is generated using a candidate Doppler frequency. The complex reference signal in a given set is generated by point-by-point multiplication of the local code with the sine and cosine maps of the locally generated continuous wave (CW) signal. Following this, a k-point multiplication operation is performed with $\{X_n^*\}$ and the FFT of each set of $\{\tilde{x}_{n,i}\}$. Next, the inverse FFT of the result is obtained and denoted as $\{r_{n,\,i}\}$. The absolute values of $\{r_{n,\,i}\}$ are then computed which are $k \times M$ in numbers, M being the number of candidate Doppler frequencies. The maximum of $k \times M$ absolute values gives the code phase and Doppler frequency of the acquired signal when it crosses a pre-determined threshold. In the time domain the incoming signals and the local reference signals are correlated multiple times by sliding the replica code by each of its phase states from start to end. In the frequency domain this time consuming operation is replaced with a k-point multiplication, thus expediting the search process. Interested readers may refer to [69] for a detailed treatment of various FFT-based acquisition algorithms.

2.6 Signal Tracking and Navigation

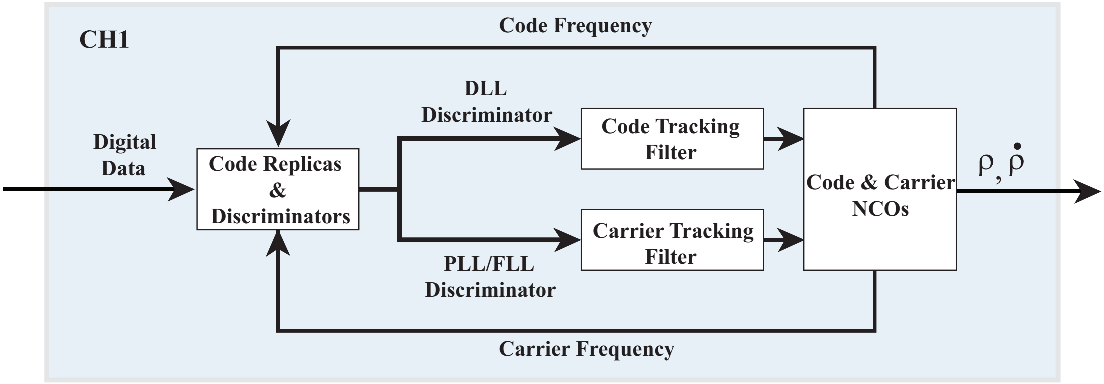
After satellite signal acquisition, signal tracking and navigation commence. Two different tracking architectures have been studied in this thesis, namely scalar and vector tracking. In the scalar architecture code tracking loops, also called delay locked loops (DLL), are used for continuously tracking pseudorandom code phase offset (τ) from the truth. Carrier tracking loops are used for estimating the Doppler error (Δf_d) and initial carrier phase offset ($\Delta \phi_o$). Carrier tracking loops which estimate only Δf_d are called frequency locked loops (FLL). Carrier tracking loops which estimate both Δf_d and $\Delta \phi_o$ together as a single quantity ($\Delta \phi$) are called phase locked loops (PLL). Similarly, for vector tracking loops there are vector DLL (VDLL), vector FLL (VFLL) [10] and vector PLL (VPLL) [12], respectively. There is a fundamental difference in the way scalar and vector architectures implement their signal tracking as well as PVT estimation or navigation algorithms. The basics of the scalar architecture is discussed in this chapter. Discussion on the vector architecture is reserved for the next chapter.

2.6.1 Scalar Tracking Architecture

As mentioned earlier and also depicted in Figure 2.5, the scalar architecture performs signal tracking and PVT estimation separately. Scalar tracking loops in a given receiver are independent of each other (each tracking one satellite) and have their own loop filters. The outputs of the loop filters provide corrections to their respective numerically controlled oscillators (NCO) which are of two types - code NCO and carrier NCO. The code NCO outputs the code phase to the code generator/coder which produces three shifted versions of the replica code (E, P, L). The carrier NCO generates inphase and quadraphase components of the baseband CW signal envelope through the cosine and sine mapping functions, respectively. The replica code and carrier signals are correlated with incoming signals, resulting in three correlator pairs (I_E, Q_E, I_P, Q_P, I_L, Q_L), as discussed previously. The correlator pairs are then passed to discriminator functions to generate estimates of the code phase and Doppler shift (or carrier phase) errors. The estimated errors drive their respective tracking loop filters. Thus, a code/carrier tracking loop can be modeled as a closed loop feedback control system which continuously tracks the code phase offset/Doppler shift (or carrier phase offset)



Scalar Receiver Architecture



Internals of a Scalar Loop

Figure 2.5: Scalar Architecture

of a visible satellite. Once the code and carrier tracking loops are locked (indicated by the estimated C/N$_0$ for DLL and FLL and by the phase lock indicator for PLL [63]), the navigation data can be recovered by the receiver baseband software. From the signal transmit time retrieved using the navigation data and tracked code phase and the signal receive time determined by the navigation process, the code pseudorange measurements to the satellite are computed. The estimated Doppler frequency provides pseudorange rate measurements. If the PLL is in lock, then the carrier phase (or accumulated delta range) measurements are also generated with an integer ambiguity which is resolved in the navigation filter. It should be noted that all measurements are generated by the navigation process. It receives the signal transmit time and carrier phase or carrier Doppler from the receiver baseband process corresponding to a measurement epoch. Then it forms the code pseudoranges (or carrier phase measurements) and pseudo range rates of all tracked satellites and processes them together to estimate the user position, velocity, clock bias and clock drift. This completes the basic signal tracking and PVT estimation methodology of a conventional GPS receiver with scalar tracking loops.

A DLL is usually aided by the Doppler estimate of the carrier tracking loop so that it needs to track only the residual code phase error. This simplifies the DLL loop filter design, making a narrow bandwidth first order DLL sufficient in most cases. The term "bandwidth" is a key loop design parameter and will be discussed later in this section. The PLL, although the most fragile component of the tracking loops, produces highly accurate Doppler estimates [68]. An FLL-generated Doppler is noisier than that from a PLL, but FLLs are more robust than PLLs when tracking weak signals and high user dynamics. A well-designed GNSS receiver, therefore, closes the carrier tracking loop as a wideband FLL. Then, it gradually reduces the FLL bandwidth and transitions to a wideband PLL. The PLL bandwidth is narrowed as tracking proceeds. When the PLL loses lock, the receiver switches back to FLL and stays in this mode until signal strength/receiver dynamics once again becomes conducive to carrier phase tracking. Weak signal tracking being a primary objective of this research, both first and second order FLLs as well as first and second order DLLs are considered herein.

It is important that a GNSS receiver is equipped with a means to detect code and carrier lock. This is to avoid contaminating the navigation processor with incorrect measurements. It also allows the receiver to transition from FLL to PLL and vice versa. In addition, code tracking depends on carrier frequency tracking as the latter ensures signal coherence over a coherent integration interval, meaning the phase error remains almost constant over that interval. Code lock is determined by estimating $\mathrm{C/N_0}$ and comparing it to a minimum value. The minimum $\mathrm{C/N_0}$ can be obtained by matching the discriminator pull-in range to the 3-σ tracking noise jitter [67]; both pull-in range and jitter are discussed later in this chapter. Different $\mathrm{C/N_0}$ measurement techinques have been thoroughly discussed in [70]. The estimated $\mathrm{C/N_0}$ also serves as a frequency lock detector. The $\mathrm{C/N_0}$ level needed to maintain carrier frequency tracking is similar to that needed for code tracking [67]. Carrier phase lock can be detected by a phase lock indicator [63] and also by the estimated $\mathrm{C/N_0}$ if loss of lock is caused by weak signals [67]. The key components of a tracking loop and the navigation filter are described in brief next.

2.6.1.1 Discriminators

It is evident from Equation 2.4 that the correlators are a function of all of three parameters of interest τ, Δf_d and $\Delta\phi$. Therefore, linear/nonlinear mathematical functions are applied to the correlator outputs to produce quantities that are proportional to either τ, Δf_d or $\Delta\phi$. These functions are called discriminators. Each tracking loop has its own discriminator functions which are discussed below. It should also be noted that both scalar and vector loops use discriminators.

DLL Discriminators: As mentioned earlier, the code tracking loop attempts to zero out the difference between E and L correlator powers to align the prompt code with that of the received signal. The E and L correlator outputs are processed by a DLL discriminator function to form a code phase error signal which is fed to the DLL loop filter, as illustrated in Figure 2.5. It is apparent from Equation 2.4 that the unknown nuisance parameters D and $\Delta\phi$ need to be eliminated to extract τ from the E and L correlators. If $\Delta\phi$ and D are well known, they can be easily removed before discriminator processing. The corresponding DLL is called a coherent DLL as the receiver's estimate of the baseband CW signal is assumed to be coherent with the received signal [62]. For a coherent DLL, the early-minus-late discriminator is used which requires only the in-phase E and L correlators [58]. While noise of this discriminator is lowest of all DLL discriminators, coherent code tracking is reliant on carrier phase tracking and cannot, therefore, be performed when carrier phase lock is lost. As code tracking is more robust than carrier phase tracking, non-cohrenet DLLs

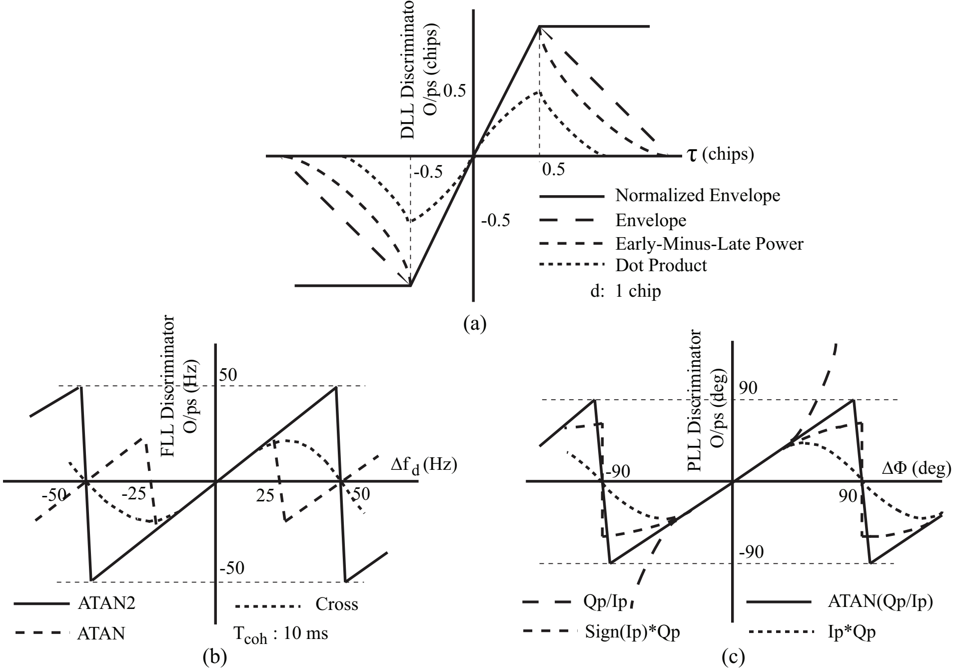


Figure 2.6: Discriminator Outputs in the Absence of Input Noise

have been devised which estimate the code phase error regardless of carrier phase tracking and knowledge of data bits. This is accomplished by eliminating the nuisance parameters (D and $\Delta\phi$) through a non-linear operation which also operates on the received noise, thus resulting in additional discriminator noise termed squaring loss [62]. Various non-coherent DLL discriminators include dot product, early-minus-late power, early-minus-late envelope and early-minus-late envelope normalized. They are illustrated in Figure 2.6 (a). Their mathematical expressions can be found in [68]. For a unit chip correlator spacing all of these discriminators except the dot product are linear when the input code phase error is within ±0.5 chips, assuming no noise in the correlator samples. The normalized envelope discriminator is independent of the signal amplitude unlike others. It also remains constant at an error input between $|0.5|$ chips and $|1.5|$ chips, thus providing a greater working range. It becomes unstable at ±1.5 chip error. Extended range discriminators are also possible. For example, [39] discusses two non-coherent DLL discriminators, flattop and linear, which have an extended operating range of ±5 chips.

The noise variances (σ_τ^2) at the output of the early-minus-late (or coherent), dot product and early-minus-late power discriminators can be derived theoretically and are given below [71]

$$\begin{aligned}\sigma_\tau^2 =& \frac{d}{4C/N_0T}, \quad \text{(early-minus-late)}\\ \sigma_\tau^2 =& \frac{d}{4C/N_0T}\left[1+\frac{1}{C/N_0T}\right], \quad \text{(dot product)}\\ \sigma_\tau^2 =& \frac{d}{4C/N_0T}\left[1+\frac{2}{(2-d)C/N_0T}\right], \quad \text{(early-minus-late power)}\end{aligned} \tag{2.5}$$

where d is early and late correlator spacing in chips and T_{coh} is the coherent integration time. The term within square brackets is called squaring loss. The early-minus-late power discriminator is implemented in this research for both DLL and VDLL. Its mathematical expression is $D_R = (I_E^2+Q_E^2)-(I_L^2+Q_L^2) = (AM)^2 2(2-d)\tau + \zeta$. The subscript R stands for range because the DLL discriminator outputs are proportional to the difference between the true pseudorange measurements and the receiver's estimate of the pseudorange measurements, as discussed later in this thesis. ζ is the discriminator noise. A is the signal amplitude. M is the number of samples in a coherent integration interval T_{coh}.

Carrier Frequency Discriminators: A carrier frequency discriminator extracts the Doppler frequency error by operating on two adjacent pairs of prompt I and Q samples. Different types of carrier frequency discriminators include the cross product, decision-directed cross product, cross-over-dot product, two-quadrant and four-quadrant ATAN discriminators [67]. Three of them are depicted in Figure 2.6 (b). The decision-directed cross product, cross-over-dot product and two-quadrant ATAN discriminators are Costas discriminators. They are insensitive to the phase reversals in the I and Q signals due to bit transitions, provided each of the two sample pairs does not straddle data bit boundaries. With the cross product and four-quadrant ATAN discriminators both pairs of the I and Q samples should be collected within the same data bit period to eliminate errors caused by bit sign changes. References [67] and [68] summarize the key characteristics of these discriminators.

It is important to note that the pull-in range of a carrier frequency discriminator (i.e., the range of frequency error within which the error would reduce to zero in the absence of noise) varies with the coherent integration interval T_{coh} of an I and Q sample pair. The discriminator input-output curves also repeat every $1/4T_{coh}$ or $1/2T_{coh}$ Hz, resulting in multiple crossings of the frequency error axis, each at an integer multiple of $\pm 1/4T_{coh}$ Hz. With a small T_{coh}, the second null point occurs at a large frequency offset which will result in the loss of code lock [63]. However, with a long coherent interval, the discriminator output reduces to zero at a relatively small frequency offset. For example, when T_{coh} is equal to 10 ms, the second null point of the ATAN discriminator occurs at 25 Hz which results in a false lock. This means a frequency error larger than 25 Hz converges to 25 Hz instead of approaching zero. After transitioning from FLL to PLL, the PLL would also false lock at this frequency offset. However, a false lock will lead to continuous parity failure during frame synchronization. False lock can also be detected by a discrepancy between code and carrier Doppler [63]. To avoid false locks, an FLL can be closed with a small coherent interval (i.e., a higher pre-detection bandwidth) which is increased as tracking progresses. It should be noted that this work uses the cross product discriminator for both FLL and VFLL. If $(I_{P_1},\ Q_{P_1})$ and $(I_{P_2},\ Q_{P_2})$ denote two adjacent paris of correlator samples taken at time t_1 and t_2, respectively, with $t_2 > t_1$, then the mathematical form of the cross discriminator is given by $D_{RR} = (I_{P_1}Q_{P_2} - I_{P_2}Q_{P_1})/(t_2 - t_1) = \pi(AM)^2\Delta f_d + \xi$. The subscript RR represents range rate as Doppler frequencies are proportional to the range rate measurements. ξ is the discriminator noise.

Carrier Phase Discriminators: As is evident, a carrier phase discriminator determines total carrier phase error at the end of a coherent integration interval by using a non-linear function of the prompt correlator pair. To eliminate errors due to data bit transitions, the I and Q samples should not be formed across bit boundaries for navigation message-modulated signals. Commonly used carrier phase discriminators and their characteristics are discussed in [67] and [68]. Among them, the Q-channel and four quadrant ATAN discriminators are linear over the entire input error range of $\pm 180^{o}$. Their noise variances do not include squaring loss and are, therefore, independent of the coherent integration time unlike their counterparts. However, they are not Costas discriminators which include IQ product, decision-directed Q, Q-over-I and two-quadrant ATAN discriminators (Figure 2.6 (c)). Costas discriminators can track phase errors between $\pm 90^{o}$. For pilot signals without modulated data bits the non-Costas Q-channel and four-quadrant ATAN discriminators are preferred as they are sensitive to the full range of phase error [67].

Apart from a few discriminators such as the normalized envelope for DLLs, cross-over-dot and ATAN for FLLs and Q-over-I and ATAN for PLLs, discriminator outputs are dependent on the signal amplitude. Consequently, knowledge of the incoming signal amplitude is required for these discriminators. This is to normalize their outputs so that tracking loop gains and bandwidths do not vary with signal amplitudes. Such variations are undesirable because they affect the noise rejection and dynamic performance of the tracking loops and may even cause higher order (≥ 3) loops to become unstable [63]. The signal amplitude of each channel can be estimated using a Kalman filter (KF) and a separate amplitude discriminator, as described in [47].

It should be noted that the discriminator input-output characteristics are linear over a finite range of values of τ, Δf_d or $\Delta\phi$. When the error input lies within this linear region, non-linear correlation and discriminator operations can be replaced with a simple summation for subsequent analysis of tracking loop performance. One input to the summation block would be the received signal code/carrier phase or Doppler frequency minus its acquired value (zero for the carrier phase). The other input would be the reverse of its tracking loop estimate. The output of the summation block would, therefore, be τ, Δf_d or $\Delta\phi$. Thus, under the assumption of linearity, a tracking loop can be represented by a transfer function block diagram which is illustrated in Figure 2.7. The quotation marks in the first summation block of the figure stand for linearization

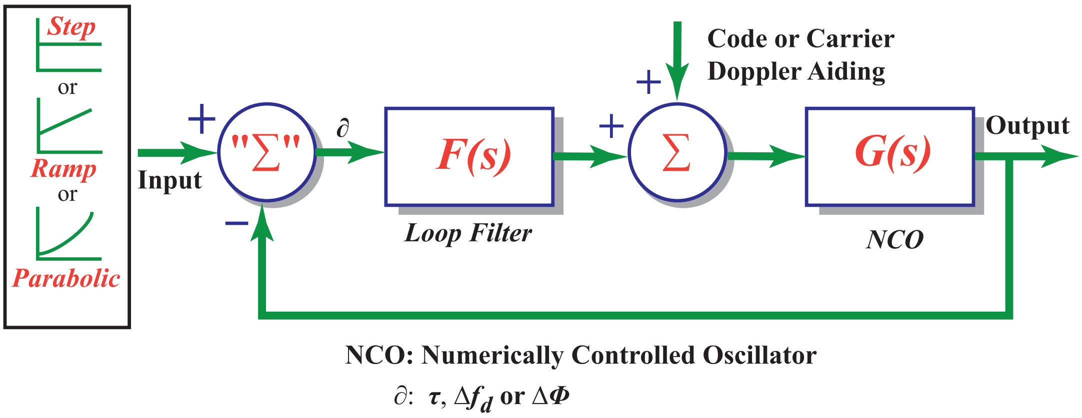


Figure 2.7: Scalar Tracking Loop Block Diagram

of the non-linear correlators and discriminators. The structure of the tracking loop filter is discussed below.

2.6.1.2 Loop Filter

As depicted in Figure 2.7, the discriminator output is fed to a loop filter with transfer function $F(s)$. The objectives of the loop filter are two-fold. First, it filters noise in the discriminator outputs. Second, it shapes the output transient and steady state response. Loop filters of type zero, first and second are normally used in GNSS receivers. More specifically, $F(s)$ takes on the following form: $\frac{\sum_{i=1}^{n} c_{i-1} s^{i-1}}{s^{n-1}}$ for DLL/PLL, where $n = 1, \ldots, 3$ and $\frac{\sum_{i=1}^{n} c_{i-1} s^{i-1}}{s^{n}}$ for FLL, where $n = 1,\ 2$. Higher order filters are generally avoided because a higher order loop becomes unstable after a certain noise bandwidth which is defined later. For example, a third order tracking loop is stable for noise bandwidths larger than 18 Hz [68]. Typical values of the filter coefficients ($\{c_{i-1}\}$) that are used in GNSS receivers can be determined from the steady state gain and error covariance of a KF of the same order as that of the tracking loop. The loop filter is, therefore, sub-optimal in nature [18] and its coefficients can be specified using the loop order and noise bandwidth. While a higher order loop is prefereble for better dynamic performance, choosing a suitable loop noise bandwidth is a trade off between noise rejection and dynamic response. This is because narrowing the noise bandwidth results in better noise rejection, but compromises the loop's ability to handle user dynamics. It is important to note that the NCO in Figure 2.7 receives a frequency input and provides a frequency output for an FLL and a code/carrier phase output for a DLL/PLL. Thus, it is modeled as an integrator for a DLL/PLL and as unity for an FLL. As a result, an FLL has the same order as that of its loop filter whereas a DLL/PLL is of one order higher than that of its loop filter.

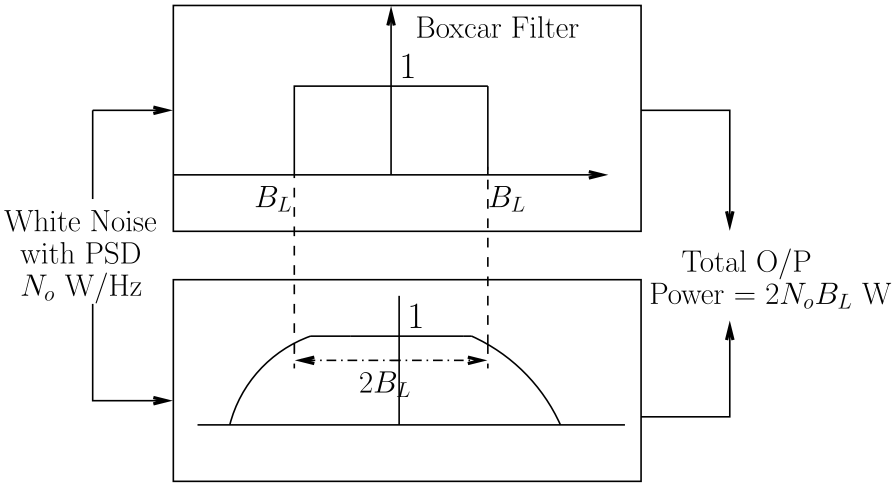
A first order loop is generally sufficient for a DLL aided by a PLL/FLL. For carrier tracking, the loop order and noise bandwidth are chosen based on application. For example, if the carrier tracking loop is not externally aided and is subject to high dynamics, then a third-order PLL is selected for carrier phase tracking which can withstand acceleration stress with zero steady state error. To minimize its jerk stress, the noise bandwidth should be as wide as possible i.e., 18 Hz for a third-order loop [68]. However, if it also needs to operate in weak signal environments, then an optimal design would be to vary its noise bandwidth adaptively as a function of C/N_0. This can be performed either by changing the noise bandwidth based on a lookup table which will list desired noise bandwidths for different C/N_0s or by implementing a KF for each loop. The bandwidth of a KF is adjusted automatically through the Kalman gain as its measurement noise covariance matrix is a function of C/N_0. Reference [72] describes such an extended Kalman filter (EKF)-based tracking algorithm that estimates code phase, carrier phase, Doppler frequency, rate of change of Doppler, signal amplitude and data bit sign. Another approach to maintaining carrier tracking at lower C/N_0 is to reduce the carrier tracking loop bandwidths by deriving external Doppler aiding from an INS. This type of GNSS/INS integration is called tight integration where the INS tracks most of the user dynamics [25]. Thus, the GNSS tracking loop needs to track only residual dynamics due to INS error and colored noises such as clock jitter, vibration-induced noise, etc.. This allows it to reduce the noise bandwidth further at low C/N_0 environments. While varying the noise bandwidth adaptively with C/N_0, however, caution must be exercised to ensure that the Kalman filter gain of the integration filter/navigation processor is matched to the tracking loop bandwidth. External aiding of code tracking is used when the carrier cannot be tracked as the code is usually aided by the carrier. The noise bandwidth of a tracking loop is discussed next.

2.6.1.3 Noise Bandwidth

The closed loop transfer function of a tracking loop under linear operating conditions is given by (Figure 2.7)

$$H(s) = \frac{F(s)G(s)}{1 + F(s)G(s)} \tag{2.6}$$

The lower panel of Figure 2.8 shows a representative frequency response of $|H(j\omega)|^2$ which has gradual roll offs, unlike the sharp edges of a boxcar filter shown in the upper



B_L: One Sided Noise Bandwidth

Figure 2.8: Noise Equivalent Bandwidth

panel of the same figure. If N_0 is the PSD of the white noise input to $H(j\omega)$, then the noise power at the tracking loop output is given as $N_0/\pi\int_{\omega=0}^{\infty}|H(j\omega)|^2d\omega$. For ease of representation and analysis, the term tracking loop noise bandwidth is introduced and defined as follows. It is the bandwidth of a boxcar filter with the same DC gain as that of the scalar tracking loop that would allow the same amount of white noise as $H(j\omega)$ [62]. Assuming the boxcar filter has the same DC gain as that of $H(j\omega)$, the one-sided noise bandwidth of $H(j\omega)$ becomes

$$B_L = \frac{1}{2\pi|H(j0)|^2}\int_{\omega=0}^{\infty}|H(j\omega)|^2d\omega \tag{2.7}$$

Similarly, in the discrete domain the noise bandwidth is given by

$$B_L = \frac{1}{2\pi|H_d(e^{j0T})|^2}\int_{0}^{\pi/T}H_d(e^{j\omega T})\times H_d(e^{-j\omega T})d\omega \tag{2.8}$$

where $H_d(z)$ is the discrete closed loop transfer function.

A tracking loop with one-sided noise bandwidth B_L will have an output noise power of $2|H(j0)|^2N_0B_L$, with N_0 representing the input white noise PSD. Thus, the noise bandwidth quantitatively represents the tracking loop's ability to reject noise. It also has a one-to-one correspondence with the loop natural frequency [18] which enables it to control the loop dynamic response. In traditional scalar tracking loops the noise bandwidth together with the loop order determines all of loop characteristics including transient response and steady state tracking error.

2.6.1.4 NCO

The loop filter output drives an NCO which accumulates phase at its clocking rate based on its input frequency. When its accumulator overflows, a new cycle is generated. Minimum phase increment of an N-bit NCO over time $1/f_c$ is 2^{-N} rad, where f_c is the clocking frequency. Therefore, the phase increment caused by an input frequency f over time $1/f_c$ is $2^{-N} \times \lfloor f/f_c \times 2^N \rfloor$ rad. $\lfloor a \rfloor$ denotes the largest integer not exceeding a. The maximum frequency that can be generated by an NCO is half of its clocking frequency. This is determined by the Nyquist criteria. For the carrier NCO sine and cosine mapping functions convert the NCO phase into appropriate trigonometric functions. On the other hand, the phase of the code NCO is directly used to generate the code phase. When the code NCO overflows, subsequent code generator output is incremented by one chip. Thus, the code generator keeps track of the integer number of chips of the current code epoch and the code NCO provides the fractional chip phase of the current epoch. Typically, the frequency generated in the carrier NCO is the Doppler frequency plus an intentional frequency offset (if applicable) whereas the code NCO generates the entire code frequency i.e., the code chipping rate plus Doppler [63].

2.6.1.5 Tracking Loop Performance Metric

While the true tracking loop performance can be determined by Monte Carlo simulations under various user dynamics and C/N$_0$ conditions, total phase (for DLL/PLL) or frequency (for FLL) jitter is generally used to predict tracking loop error based on theory [68]. The standard deviation of tracking error caused by white noise is called white noise jitter. If σ_{w_i} is the standard deviation of the input white noise, then the white noise jitter σ_{w_o} is $\sqrt{2B_L}\sigma_{w_i}$, where B_L is the one-sided loop noise bandwidth. The standard deviation of tracking error caused by colored noises is called colored noise jitter. Different sources of colored noises include short term frequency instability of the receiver oscillator, vibration-induced oscillator error, satellite clock instability and error due to propagation delay in the ionosphere [64]. If $S_{c_j}(f)$ is the PSD of the i^{th} colored noise, then the corresponding jitter σ_{c_j} is given by

$$\sigma_{c_j}^2 = 2\int_0^\infty |1 - H(j2\pi f)|^2 S_{c_j}(f)df \tag{2.9}$$

Thus, the colored noise jitter σ_{c_o} due to all colored noises is the root sum square of all σ_{c_j}. A close observation of the mathematical expressions of the white and colored noise jitters reveals the following. The white noise jitter arises from the tracking loop's inability to *eliminate* white input noise. On the other hand, the colored noise jitter results from the tracking loop's inability to *track* colored noises. Additionally, the tracking loop's inability to quickly respond to abrupt changes in user motion or any steady state error inflicted by user dynamics is quantified by the term dynamic stress error σ_{dyn}. For example, the peak transient error resulted from a phase velocity, phase acceleration or phase jerk step in the third order loop is approximated by the steady state error of a first, second or third order loop due to a phase velocity, phase acceleration or phase jerk step function, respectively. Total 1-σ phase/frequency jitter is obtained by adding one third of dynamic stress error to the root sum square of white and colored noise jitters. Mathematically, it is represented by

$$\sigma_{tr} = \sqrt{\sigma_{w_o}^2 + \sigma_{c_o}^2} + \frac{\sigma_{dyn}}{3} \tag{2.10}$$

Rules of thumb threshold of total jitter at which a tracking loop loses lock are defined in [68]. For the PLL the 1-σ threshold is 15° while for the FLL it is $0.0833/T_{coh}$. The DLL 1-σ threshold is $d/6$ in chips. They are determined by the respective discriminator pull-in ranges. It is shown in [75] that total phase jitter is a reliable metric for assessing tracking loop performance when the loop noise bandwidth is not too small ($>$ 5 Hz). That is, total phase jitter can reliably predict the tracking threshold C/N$_0$ which is very close to that predicted by bit error rate when the bandwidth is not too small.

2.6.1.6 Generation of Pseudorange, Delta Pseudorange and Carrier Phase Measurements

The receiver baseband process uses code and carrier accumulators to keep track of code and carrier phases of all visible satellites. These accumulator outputs are used to form the pseudorange, delta pseudorange and carrier phase (integrated Doppler) measurements.

Code Accumulator: Each channel of the receiver keeps track of the corresponding satellite signal transmit time by using a code accumulator which accumulates the receiver-generated replica code phase [68]. The code accumulator has three counters. The first counter, also called the Z counter, increments every 1.5 sec. The second counter increments with each replica code chip and the third counter tracks fractions of a chip. The code accumulator is updated after each NCO update. As there is a one-to-one relationship between the transmitted code phase and satellite signal transmit time, the code accumulator provides signal transmit time corresponding to the time at which the signal is received (or processed). This is because the receiver-generated replica code is synchronized to the transmitted code using DLLs. However, the code accumulator needs to be propagated to the desired receiver time epoch to obtain the transmit time for that epoch. This is needed to account for a time skew between the NCO update schedule and receiver time epochs. The baseband process sends the signal transmit time and receive time tag to the navigation process which generates the pseudorange measurements [68].

Carrier Accumulator: The carrier accumulator tracks the integer number of carrier cycles and fractions of a cycle since carrier lock is established. It is updated at the NCO update rate. The delta pseudorange measurements are derived from the change in phase in the carrier accumulator during a specified time. The delta pseudorange can also be extracted from the code tracking loop, but it is about a thousand times and a hundred times less noisy when obtained from a PLL and an FLL, respectively [68]. Usually, the receiver baseband processor propagates the carrier phase output of the accumulator forward to the nearest measurement epoch; forms the delta pseudorange measurements; and then sends delta pseudorange and carrier phase measurements (if the PLL is locked) to the navigation processor.

2.6.1.7 Navigation Processor

The pseudorange (ρ_i) and pseudorange rate ($\dot{\rho}_i$) measurements between the user and satellite i as generated by the navigation processor at receive time $\hat{t}_r$ have the following form [73]:

$$\rho_i = c(\hat{t}_r - \hat{t}_{\tau_i}) = |\underline{r}_i - \underline{r}_u| + b + \epsilon_{p_i} \tag{2.11}$$

$$\begin{aligned} \dot{\rho}_i &= \lambda/(\Delta t) \times (\hat{\phi}_{\hat{t}_r} - \hat{\phi}_{\hat{t}_r - \Delta t)}) \\ &= (\dot{\underline{r}}_i - \dot{\underline{r}}_u).\frac{\underline{r}_i - \underline{r}_u}{|\underline{r}_i - \underline{r}_u|} + \dot{b} + \epsilon_{\dot{\rho}_i} \\ &= (\dot{\underline{r}}_i - \dot{\underline{r}}_u).\underline{\tilde{u}}_i + \dot{b} + \epsilon_{\dot{\rho}_i} \end{aligned} \tag{2.12}$$

where c is the speed of light; $\hat{t}_{\tau_i}$ is the transmit time obtained from the code accumulator; the "hat" sign of $\hat{t}_r$ indicates that it is a receiver-generated estimate of actual t_r; λ is the carrier wavelength; $\phi_{\hat{t}_r}$ and $\phi_{\hat{t}_r - \Delta t}$ are carrier phases output by the carrier accumulator corresponding to times $\hat{t}_r$ and $\hat{t}_{r\,-\,\Delta t}$, respectively. $\underline{r}_i$ is the satellite position vector at transmit time; $\underline{r}_u$ is the user position vector at receive time; $\underline{\tilde{u}}_i$ is the unit vector from satellite i to the user; b is the receiver clock bias in m; and ϵ_{ρ_i} represents the error due to satellite clock, ionospheric and tropospheric delays, receiver noise, etc.. The (.) operation in Equation 2.12 represents the scalar product. $\dot{b}$ is the receiver clock drift in m/s. $\epsilon_{\dot{\rho}_i}$ denotes unmodeled error and noise in $\dot{\rho}_i$. The navigation processor predicts the ρ and $\dot{\rho}$ for each tracked satellite based on its knowledge of $\underline{r}_i$, $\underline{\dot{r}}_i$, $\underline{r}_u$, b, ϵ_{ρ_i}, $\underline{\dot{r}}_u$, $\dot{b}$ and $\epsilon_{\dot{\rho}_i}$. It then subtracts the predicted values from the measurements to form the pseudorange and pseudorange rate residuals, as given below

$$\Delta\rho_i = \rho - \tilde{\rho}_i = [-\underline{\tilde{u}}_i^T \quad 1] \begin{bmatrix} \Delta\underline{r} \\ \Delta b \end{bmatrix} + \Delta\epsilon_{\rho_i} \tag{2.13}$$

$$\Delta\dot{\rho}_i = \dot{\rho} - \tilde{\dot{\rho}}_i = [-\underline{\tilde{u}}_i^T \quad 1] \begin{bmatrix} \Delta\underline{\dot{r}} \\ \Delta\dot{b} \end{bmatrix} + \Delta\epsilon_{\dot{\rho}_i} \tag{2.14}$$

where $\Delta\underline{r} = \underline{r} - \underline{\tilde{r}}$, $\Delta\underline{\dot{r}} = \underline{\dot{r}} - \underline{\tilde{\dot{r}}}$, $\Delta b = b$ - $\tilde{b}$, $\Delta\dot{b} = \dot{b}$ - $\tilde{\dot{b}}$. The superscript tilde represents the receiver's *a priori* estimate of user position, velocity and clock parameters. Satellite position error is usually ignored or taken into account in $\Delta\epsilon_\rho$. Satellite clock error is corrected using the clock coefficients transmitted in the navigation message. The navigation processor employs a least squares estimator or an EKF algorithm to estimate $\Delta\underline{r}$, Δb, $\Delta\underline{v}$ and $\Delta\dot{b}$. Then, the estimated Δ terms are used to correct the *a priori* estimate of user PVT. It should be noted that the $\Delta\epsilon$ terms are generally modeled as Gaussian noises with appropriate means and variances based on residual error models. Representative error models can be obtained from the user equivalent range error (UERE) given in [76]. As the work in this thesis processes code pseudorange and Doppler measurements, processing of carrier phase measurements is not addressed here.

2.7 Conclusion

In this chapter a brief overview of the basics of GPS signal tracking has been provided. The following topics have been touched upon: GPS signal structure, receiver front end signal processing, signal acquisition and tracking. Under signal tracking the key components of the conventional tracking loop architecture, namely scalar tracking have been discussed. In the next chapter an advanced tracking loop architecture called vector tracking which is the focus of the research in this thesis is covered.

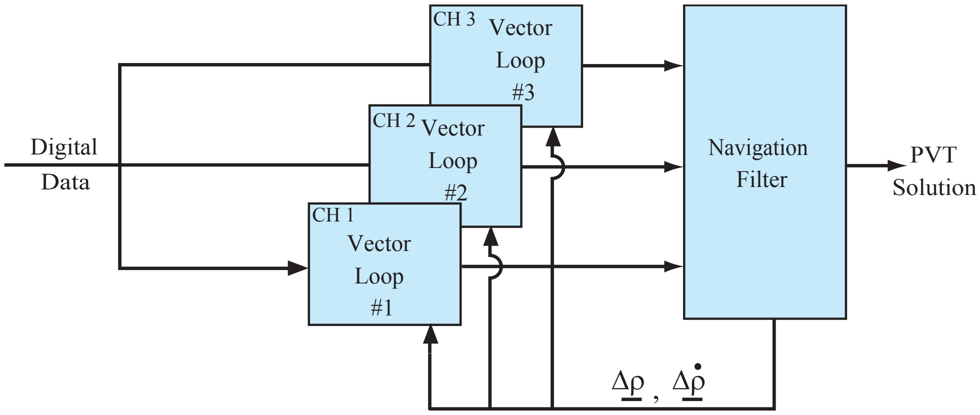
Chapter 3

Basics of Vector Tracking Architecture

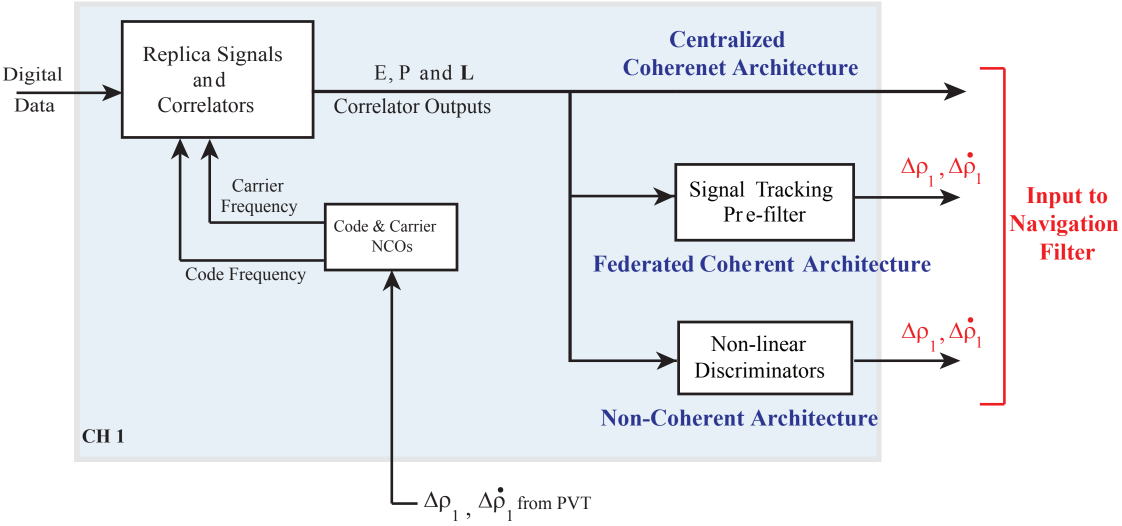
The previous chapter discussed the basics of a conventional GPS receiver architecture. In this chapter the key features of an advanced architecture called vector tracking (or vector tracking loops) are presented. The primary objective of this chapter is to lay the foundation for subsequent analyses of vector loops in Chapters 4 through 6. It also highlights the differences between scalar and vector tracking architectures.

3.1 Vector Tracking Architecture

Figure 3.1 shows a high level schematic and implementation of a vector tracking loop architecture. As is evident in the figure, the cascaded structure of tracking and navigation filters of the conventional receiver is replaced in this architecture with a central navigation filter. The number of loops shown is for illustration purposes. At least four loops are required and as many as possible can be added depending on processor capability. It should be noted that at least four satellites must be acquired and tracked independently to estimate the user position and velocity with sufficient accuracy before closing the vector tracking loops [63]. In other words, the vector loops are initialized by the scalar loops. In the vector architecture the navigation filter closes the signal tracking loops by generating corrections for the code and carrier NCOs. Having received frequency corrections, the NCOs operate the same way as that in scalar loops. The navigation filter accepts either correlator or discrminator outputs as its inputs, unlike the traditional scalar loop filter which receives inputs only from the discriminator functions. Based on the input to the navigation filter, two different architectures



Vector Receiver Architecture



Internals of a Vector Loop

Figure 3.1: Vector Architecture

can be realized: Coherent and non-coherent architectures. The coherent architecture can have two implementations: Centralized coherent architecture where the navigation filter receives three correlator pairs as input from each satellite channel; Federated coherent architecture where the navigation filter is preceded by a bank of pre-filters. In the non-coherent architecture the navigation filter inputs averaged discriminator outputs. Different possible inputs to the navigation filter are shown in Figure 3.1. A brief discussion of vector tracking loop filters, NCO corrections for vector loops and vector tracking loop jitter is provided below. It should be noted that the non-coherent vector architecture is implemented in this thesis with the same discriminator functions (early-minus-late power discriminator for VDLL and cross discriminator for VFLL) as

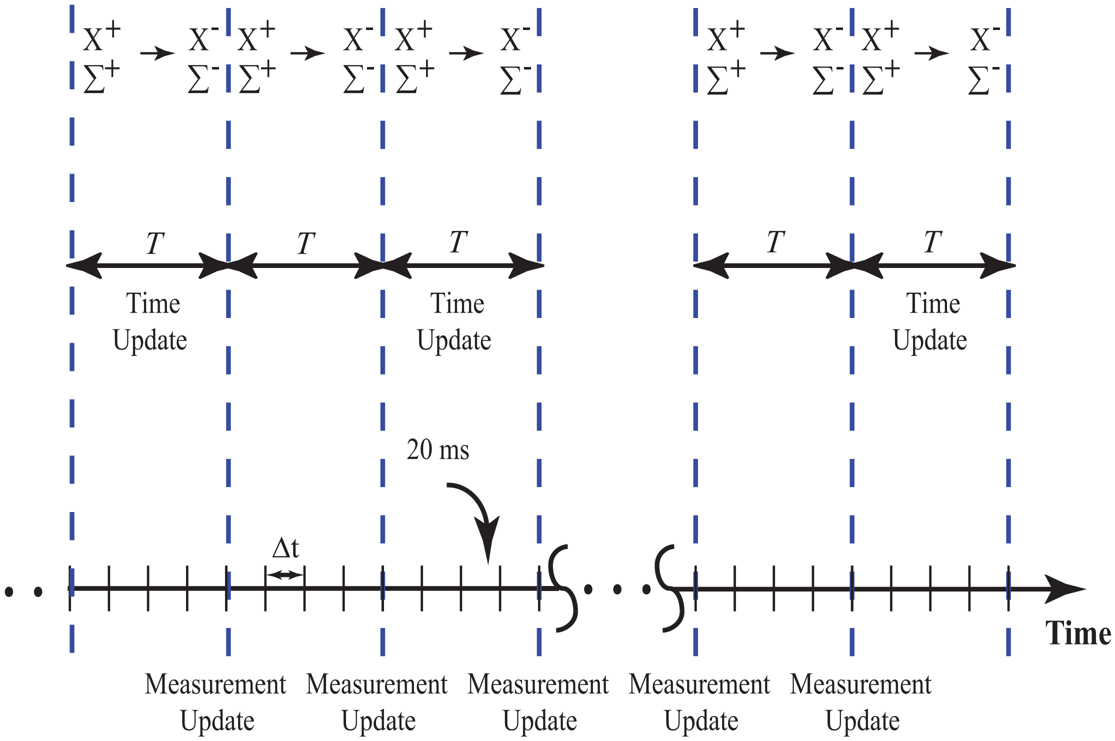


Figure 3.2: Vector Algorithm Time Line

its scalar counterparts. The mathematical models of non-coherent VDLL and VFLL are also presented.

3.1.1 Navigation Filter

The vector loop navigation filter is an EKF which combines information from all satellites to estimate the following state vector: ΔX =$[\Delta x\ \Delta\dot{x}\ \Delta\ddot{x}\ \Delta y\ \Delta\dot{y}\ \Delta\ddot{y}\ \Delta z\ \Delta\dot{z}\ \Delta\ddot{z}\ \Delta b\ \Delta\dot{b}]^T$. The vector $[x, y, z, b]^T$ represents the x, y, z components of user position and clock bias and $[\dot{x},\ \dot{y},\ \dot{z}, \dot{b}]^T$ denotes three components of user velocity and clock drift, respectively. $[\ddot{x}, \ddot{y}, \ddot{z}]^T$ are three components of acceleration. The Δ denotes errors or deviations of these variables from their true values. All positions and bias are in units of m; velocities and drift are in units of m/s. The acceleration is expressed in m/s^2. The position, velocity, acceleration and receiver clock parameters are predicted every data bit interval (Δt = 20 ms) to apply NCO corrections and updated every T sec using the estimated EKF states, as shown by the vector algorithm time line in Figure 3.2. T usually varies from 20 ms to 1 sec depending on implementations, noise levels and user dynamics. The estimated position, velocity and clock terms are predicted over time T sec using the following equation:

$$\begin{bmatrix} \tilde{x}^-_{k+1} \\ \tilde{\dot{x}}^-_{k+1} \\ \tilde{\ddot{x}}^-_{k+1} \\ \tilde{y}^-_{k+1} \\ \tilde{\dot{y}}^-_{k+1} \\ \tilde{\ddot{y}}^-_{k+1} \\ \tilde{z}^-_{k+1} \\ \tilde{\dot{z}}^-_{k+1} \\ \tilde{\ddot{z}}^-_{k+1} \\ \tilde{b}^-_{k+1} \\ \tilde{\dot{b}}^-_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & T^2/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathrm{T} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}^+_{k} \\ \tilde{\dot{x}}^+_{k} \\ \tilde{\ddot{x}}^+_{k} \\ \tilde{y}^+_{k} \\ \tilde{\dot{y}}^+_{k} \\ \tilde{\ddot{y}}^+_{k} \\ \tilde{z}^+_{k} \\ \tilde{\dot{z}}^+_{k} \\ \tilde{\ddot{z}}^+_{k} \\ \tilde{b}^+_{k} \\ \tilde{\dot{b}}^+_{k} \end{bmatrix}$$

$$\text{or,} \quad \tilde{X}^-_{k+1} = F\tilde{X}^+_k \tag{3.1}$$

where k and $k+1$ stand for k^{th} and $(k+1)^{\text{th}}$ time epochs. X_k = true user position, velocity, clock bias and clock drift at time k; $\tilde{X}^{(-/+)}_k$ = predicted/estimated user position, velocity and clock variables at t_k. $(\Delta\tilde{X}^-_{k+1})$ and $(\Delta\tilde{X}^+_{k+1})$ denote *a priori* and *a posteriori* estimates of the filter state vector ΔX_{k+1}, respectively, where $\Delta X_{k+1} = X_{k+1}$ - $\tilde{X}^-_{k+1}$. $\Delta\tilde{X}^-_{k+1}$ is given by

$$\begin{aligned} \Delta\tilde{X}^-_{k+1} &= \tilde{X}^-_{k+1} - \tilde{X}^-_{k+1} \\ &= 0 \end{aligned}$$

The *a posteriori* error covariance matrix Σ^+_k at t_k is predicted over time T as follows:

$$\Sigma^-_{k+1} = F\Sigma^+_k F^T + Q \tag{3.2}$$

Where Q is the process noise covariance matrix and accounts for uncertainties in the kinematic prediction model F. Q can be determined by trial and error or from the expected level of variations in the acceleration, clock bias and drift [10]. The above two equations indicate that the mean of the filter state vector is zero until the measurement update; only the covariance is updated during the time update.

It should be noted that the time update intervals of different satellites are skewed with respect to one another as they are synchronized with their respective data bit intervals. But the measurement updates take place at fixed receiver epochs. In other words, the small vertical lines in Figure 3.2 are skewed relative to one another when observed across satellite channels whereas the long vertical lines are the same for all satellites.

With data-modulated signals the coherent integration time of the E & L correlators and that of the P correlator in the coherent implemetation is 20 ms. In the non-coherent implementation the P correlator generates two pairs of I and Q outputs in 20 ms, with each pair having a coherent integration time of 10 ms. The two pairs of the P correlator are required by the Doppler frequency discriminator to generate the Doppler frequency error. Depending on the filter input, two implementations are feasible, namely coherent (centralized and federated) and non-coherent. Although this thesis uses a non-coherent implementation, a brief discussion of the coherent architecure is provided below.

3.1.1.1 Coherent Architecture

In the coherent architecture, all of code phase, carrier phase and Doppler frequency are tracked and code tracking relies on carrier phase tracking. Three correlator pairs of each channel are input to the navigation filter. The observation matrix is derived by differentiating Equation 2.4 with respect to τ, Δf_d and $\Delta\phi_0$ and then relating τ and Δf_d to user position, velocity and clock error. It should be noted that in coherent VFLL $\Delta\phi_0$ is estimated independently for each channel or in scalar mode while Δf_d is estimated in vector mode. To maintain carrier phase tracking, the filter should be updated at 20 ms or faster [74]. Thus, a distinct problem with coherent tracking is high processing load resulted from large measurement vector (six inputs per channel) and fast update rates. As an alternative, a bank of signal tracking pre-filters - one for each channel - are used (Figure 3.1). Each pre-filter inputs E, P and L correlator pairs at 20 ms and estimates code phase tracking error, carrier frequency error and initial carrier phase offset. At a slower rate (on the order of one or two Hz) it generates pseudorange and pseudorange rate residuals from the estimated code phase and Doppler frequency errors, respectively, and input them to the navigation filter. The pre-filter states are reset to zero after each measurement update of the navigation filter to avoid cascading problems. This is known as federated-zero-reset architecture [74]. In the federated architecture $\Delta\phi_0$ of a channel is estimated by its pre-filter. In the centralized coherent architecture the navigation filter state vector defined above is augmented by $\Delta\phi_0$s of all channels, although they are estimated in scalar mode. The major benefits of the coherent architecture are the absence of unmodeled non-linearities due to discriminator functions and lower measurement noise covariance. However, it necessitates carrier phase tracking which makes it unsuitable for low C/N$_0$ applications.

3.1.1.2 Non-Coherent Architecture

In the non-coherent architecture code phase and Doppler frequency are tracked. In this thesis the non-coherent architecture is implemented. In this architecture non-linear code phase and carrier frequency discriminator outputs are accumulated every 20 ms and a number of such outputs are averaged over T sec before using for the measurement updates. The averaged discriminator outputs are indirectly related to the errors in the position, velocity and clock parameters through the code phase errors (τ) and carrier Doppler frequency errors (Δf_d), as given by the equation below

$$\begin{bmatrix} D_{R_{1,\,k+1}}/c_{1,\,k+1} \\ \vdots \\ D_{R_{j,\,k+1}}/c_{j,\,k+1} \\ D_{RR_{1,\,k+1}}/d_{1,\,k+1} \\ \vdots \\ D_{RR_{j,\,k+1}}/d_{j,\,k+1} \end{bmatrix} = \begin{bmatrix} \tau_{1,\,k+1} \\ \vdots \\ \tau_{j,\,k+1} \\ \Delta f_{d_{1,\,k+1}} \\ \vdots \\ \Delta f_{d_{j,\,k+1}} \end{bmatrix} = \tilde{C}_{k+1} \begin{bmatrix} x_{k+1} - \tilde{x}^-_{k+1} \\ \dot{x}_{k+1} - \tilde{\dot{x}}^-_{k+1} \\ \ddot{x}_{k+1} - \tilde{\ddot{x}}^-_{k+1} \\ y_{k+1} - \tilde{y}^-_{k+1} \\ \dot{y}_{k+1} - \tilde{\dot{y}}^-_{k+1} \\ \ddot{y}_{k+1} - \tilde{\ddot{y}}^-_{k+1} \\ z_{k+1} - \tilde{z}^-_{k+1} \\ \dot{z}_{k+1} - \tilde{\dot{z}}^-_{k+1} \\ \ddot{z}_{k+1} - \tilde{\ddot{z}}^-_{k+1} \\ b_{k+1} - \tilde{b}^-_{k+1} \\ \dot{b}_{k+1} - \tilde{\dot{b}}^-_{k+1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,\,k+1} \\ \vdots \\ \epsilon_{j,\,k+1} \\ \xi_{1,\,k+1} \\ \vdots \\ \xi_{j,\,k+1} \end{bmatrix}$$

$$\underline{e}_{k+1} = \tilde{C}_{k+1}[X_{k+1} - \tilde{X}^-_{k+1}] + [\epsilon_{1,\,k+1} \ldots \xi_{j,\,k+1}]^T \tag{3.3}$$

where j = number of visible satellites

ζ_i and ξ_i are the code and Doppler measurement noise samples

D_{R_i} = averaged code phase or DLL discriminator output of satellite i

D_{RR_i} = averaged carrier frequency discriminator output of satellite i

c_i = scaling constant of $D_{R_i} = -(AM)^2_i(2-d)$

d_i = scaling constant of D_{RR_i}

$= -\pi\frac{(AM)^2_i}{2}R^2(\tau_i)\frac{\Delta t}{2}$, i=1, ..., j

$\mathrm{A}/\sqrt{2}$ = signal amplitude

M = number of signal samples over a code epoch

d = spacing between early and late correlators

$R(\tau)$ = C/A-code autocorrelation function

Δt = accumulation interval for E & L correlators = 20 ms

$\tilde{C}_{k+1} = [\tilde{C}_{x,\,k+1} \quad \tilde{C}_{y,\,k+1} \quad \tilde{C}_{z,\,k+1} \quad \tilde{C}_{clk}]$

$$\tilde{C}_{m,\,k+1} = \begin{bmatrix} \tilde{u}_{m_{1},\,k+1} \times (f_{code}/c) & 0 & 0 \\ \vdots & \vdots & \vdots \\ \tilde{u}_{m_{j},\,k+1} \times (f_{code}/c) & 0 & 0 \\ 0 & \tilde{u}_{m_{1},\,k+1} \times (f_{carr}/c) & 0 \\ \vdots & \vdots & \vdots \\ 0 & \tilde{u}_{m_{j},\,k+1} \times (f_{carr}/c) & 0 \end{bmatrix}_{2j\times 3}$$

$m = x,\, y,\, z$

$$\tilde{C}_{clk} = \begin{bmatrix} -f_{code}/c & 0 \\ \vdots & \vdots \\ -f_{code}/c & 0 \\ 0 & -f_{carr}/c \\ \vdots & \vdots \\ 0 & -f_{carr}/c \end{bmatrix}_{2j\times 2}$$

$\tilde{u}_{m_{i},\,k+1}$ = m component of the unit LOS vector from satellite i to user at t_{k+1}

$= \dfrac{m_{s_{i},\,k+1} - \tilde{m}^{-}_{k+1}}{\tilde{R}^{-}_{i,\,k+1}}$

m = $x,\, y,\, z$

$\tilde{R}^{-}_{i,\,k+1}$ = predicted geometric range of satellite i at time $k+1$

f_{code} = code chipping rate

f_{carr} = carrier frequency

c = speed of light

If $W''_{k,\,2j\times 2j}$ is appropriately scaled measurement noise covarience matrix as given in [47], $\underline{e}_{k+1}$ is the residual measurement vector from Equation 3.3 and K_{k+1} is the Kalman gain, then the measurement update equations at $(k+1)^{\text{th}}$ time epoch are as follows:

$$\begin{aligned} K_{k+1} &= \Sigma^-_{k+1}(\tilde{C}^-_{k+1})^T(\tilde{C}^-_{k+1}\Sigma^-_{k+1}(\tilde{C}^-_{k+1})^T + W''_k)^{-1} \qquad (3.4) \\ \Delta\tilde{X}^+_{k+1} &= K_{k+1}\underline{e}_{k+1} \\ \Sigma^+_{k+1} &= (I_{11\times 11} - K_{k+1}\tilde{C}^-_{k+1})\Sigma^-_{k+1} \qquad (3.5) \end{aligned}$$

So, the predicted and estimated user position, velocity, acceleration, clock bias and clock drift at t_{k+1} are

$$\begin{aligned} \tilde{X}^-_{k+1} &= F\tilde{X}^+_k \\ \tilde{X}^+_{k+1} &= \tilde{X}^-_{k+1} + \Delta\tilde{X}^+_{k+1} \qquad (3.6) \end{aligned}$$

As an alternative to non-coherent averaging, a pre-filter can be used in each channel prior to the navigation filter to reduce the discriminator noise levels [10]. In this thesis, however, discriminator outputs are averaged to reduce noise. The averaging interval or measurement update time T of the VDLL is varied in this work from 0.1 sec to 6 sec while the VFLL update interval is fixed at 0.1 sec. It is also important to note that the navigation filter state in this thesis does not include the acceleration terms.

3.1.2 Amplitude Estimator

A separate EKF algorithm with a measurement model described in [47] is used to estimate the signal amplitudes of all satellite channels. Both the initial error covariance and process noise covariance matrices are assumed to be the identity matrix. As for the prediction model, the estimated amplitude is considered constant between two successive measurement updates.

3.1.3 Code and Carrier NCO Corrections for Vector Loops

The predicted/estimated state vector is used to form the pseudo ranges and range rates to generate the following clock corrections every Δt sec. These corrections ensure that the code and carrier NCO frequencies are aligned with the predicted and estimated states of the navigation filter. It should be noted that the NCO correction times of a satellite are aligned with its data bit intervals whereas measurement updates take place at suitably determined receiver time epochs. The NCO frequency corrections are

$$\Delta f_{carr_{i,\,k+1}} = \frac{f_{carr}}{c}(-\tilde{\dot{R}}^+_{i,\,k+1} - \tilde{\dot{b}}^+_{k+1} + \dot{b}_{s_i,\,k+1}) + \Delta t \times \tilde{\dot{f}}^+_{d_i,\,carr_{k+1}} \qquad (3.7)$$

$$= \tilde{f}^{+}_{d_i,\, carr_{k+1}} + \Delta t \times \tilde{\dot{f}}^{+}_{d_i,\, carr_{k+1}}$$

$$\Delta f_{code_{i,\, k+1}} = \tilde{f}^{+}_{\tau_i,\, code_{k+1}} + \frac{f_{code}}{f_{carr}} \times \left(\tilde{f}^{+}_{d_i,\, carr_{k+1}} + \frac{\Delta t}{2} \times \tilde{\dot{f}}^{+}_{d_i,\, carr_{k+1}} \right) \tag{3.8}$$

where

i = satellite index

$\tilde{f}^{+}_{\tau_i,\, code_{k+1}} = \frac{1}{\Delta t} \frac{f_{code}}{c} (\rho_{rec_{i,\, k+1}} - \tilde{\rho}^{+}_{i,\, k+1})$; $\Delta t = 20$ ms

$= \frac{1}{\Delta t} \frac{f_{code}}{c} \times [c(\hat{t}_{r_{i,\, k+1}} - \hat{t}_{\tau_{i,\, k+1}}) - c(\hat{t}_{r_{i,\, k+1}} - \tilde{t}_{\tau_{i,\, k+1}})]$

$= \frac{1}{\Delta t} \frac{f_{code}}{c} \times [c(\tilde{t}_{\tau_{i,\, k+1}} - \hat{t}_{\tau_{i,\, k+1}})]$

$\rho_{rec_{i,\, k+1}}$ = receiver-generated i^{th} code pseudorange at t_{k+1}. It is derived using the signal transmit time output from the code accumulator and the signal receive time

$\tilde{\rho}^{+}_{i,\, k+1}$ = filter-estimated pseudorange of satellite i at t_{k+1}

$\tilde{f}^{+}_{d_i,\, carr_{k+1}}$ = estimated i^{th} carrier Doppler frequency at t_{k+1}

$\tilde{\dot{f}}^{+}_{d_i,\, carr_{k+1}}$ = estimated rate of change of carrier Doppler at t_{k+1}

$\tilde{\dot{R}}^{+}_{i,\, k+1}$ = estimated geometric range rate of satellite i at t_{k+1}

$\dot{b}_{s_i,\, k+1}$ = i^{th} satellite clock drift at t_{k+1}

$\hat{t}_{r_{i,\, k+1}}$ = signal receive time at t_{k+1} as determined by the receiver

$\hat{t}_{\tau_{i,\, k+1}}$ = receiver-generated transmit time of the signal received at $t_{r_{i,\, k+1}}$; it is derived from the code accumulator

$\tilde{t}_{\tau_{i,\, k+1}}$ = estimated signal transmit time

Here, superscript (+) denotes corrections after a measurement update. It is replaced with (−) when applying corrections after a time update. The above carrier frequency correction closes the carrier tracking loop in the form of VFLL since no corrections are applied to compensate the carrier phase offset. For coherent implementations corrections corresponding to $\Delta\phi_0$ are fed back to the carrier NCO separately. It should be noted that the code clock correction has two components. The second term is the code Doppler frequency plus the change in Doppler in $\Delta t/2$ sec while $f_{\tau,\, code_{i,\, k+1}}$ aligns the code phase corresponding to that of the estimated transmit time in Δt sec. The terms $\Delta t \times \tilde{\dot{f}}^{+}_{d_i,\, carr_{k+1}}$ and $\Delta t/2 \times \tilde{\dot{f}}^{+}_{d_i,\, carr_{k+1}}$ in $\Delta f_{carr_{i,\, k+1}}$ and $\Delta f_{code_{i,\, k+1}}$, respectively, ensure that the receiver-generated pseudorange and range rate measurements correspond to the predicted states of the navigation filter at the time of measurement updates. This

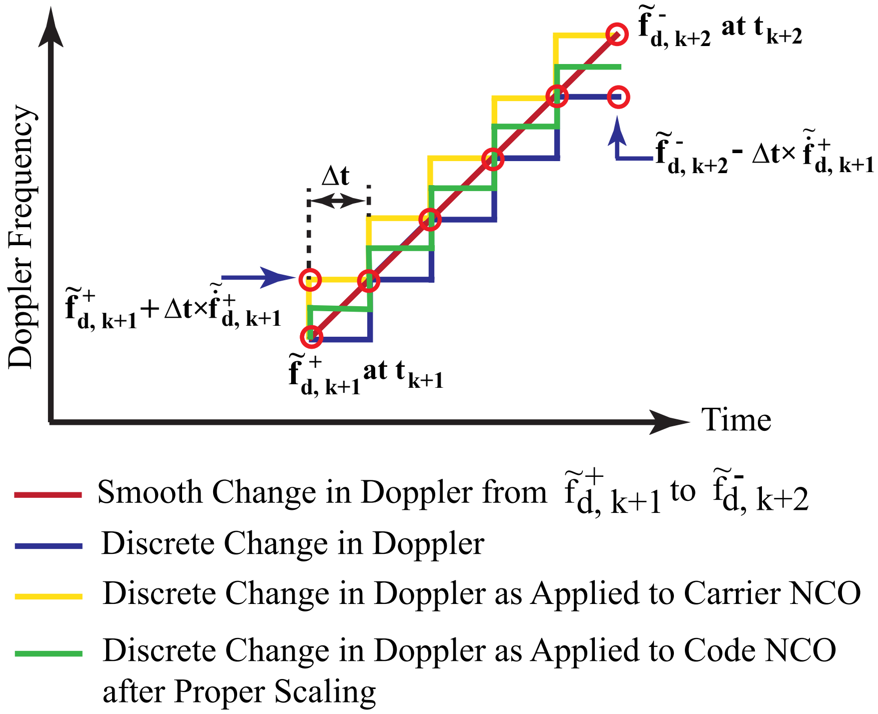


Figure 3.3: Doppler Frequency Corrections for Vector Loop NCOs; Vertical Steps Correspond to NCO Frequency Updates

is illustrated in Figure 3.3 by the propagation of the Doppler frequency from t_{k+1} to t_{k+2}. In the figure $\tilde{f}^{+}_{d,\,k+1}$ and $\tilde{f}^{-}_{d,\,k+2}$ represent the Doppler frequencies of a channel corresponding to the estimated and predicted state vectors of the navigation filter at t_{k+1} and t_{k+2}, respectively. Since the Doppler corrections are applied to the NCO at Δt sec, the upper step corrections to the NCO in yellow will align the carrier NCO Doppler frequency at the time of the measurement update with that obtained from the predicted state vector. On the other hand, the bottom step corrections in dark blue will be less than the desired value by the amount $\Delta t \times \dot{\tilde{f}}^{+}_{d,\,k+1}$. Likewise, the middle step corrections in green after scaling properly for the code Doppler will align the code phase at the time of measurement updates with that predicted from the navigation filter. This is because the area under the ramp change in Doppler and that under the green line are equal over each Δt sec interval. Thus, the Doppler frequency is propagated Δt sec and $\Delta t/2$ sec ahead of current time before applying NCO corrections. However, this can be avoided if $\dot{\tilde{f}}^{+}_{d,\,k+1}$ is negligible compared to $\tilde{f}^{+}_{d,\,k+1}$. It is also important to note that the above corrections assume that the change in $\dot{\tilde{f}}^{+}_{d,\,k+1}$ from t_{k+1} to t_{k+2} is insignificant. This is a reasonable assumption as it will be shown in the next chapter that $\ddot{\tilde{f}}^{+}_{d,\,k+1}$ is negligible.

3.1.4 Vector Loop Performance Metric

In [10] VFLL tracking error at steady state is calculated in terms of frequency jitter (due to white noise and clock error) and dynamic stress. For this purpose, the satellite geometry is assumed to be fixed because vector loop tracking error is a function of satellite geometry. To compute the frequency jitter, first the steady state Kalman gain is computed for a given level of user dynamics represented by the acceleration terms of the process noise covariance matrix. Then, the dynamics are removed by reducing the acceleration process noise terms to zero. Next, the steady state error covariance matrix is determined by using the computed steady state Kalman gain and the general equation for covariance update. The general update equation is used as the filter is no longer optimal with the process noise acceleration terms reduced to zero. The diagonal terms of the steady state error covariance matrix obtained this way represents the state error variances only due to measurement noise and receiver clock error. Hence, by projecting the velocity and clock drift variances onto the LOS, the frequency jitter of each channel is calculated. Since the frequency jitter of one channel varies from that of another, the maximum of all frequency jitters is used to determine vector loop tracking thresholds due to white noise and oscillator frequency instabilities. In this regard, it should be noted that the vector loop threshold is the $\mathrm{C/N_0}$ at which the maximum frequency jitter crosses the rule-of-thumb threshold defined in [68]. Unlike scalar tracking, it is a function of satellite geometry and user dynamics. With regard to dynamic stress error, the work in [10] assumes two separate EKFs. One has perfect knowledge of user dynamics while the other does not. Then the steady state error between the states of the two EKFs is calculated. This error is the dynamic stress error of the EKF with imperfect knowledge of user dynamics.

3.2 Conclusion

This chapter has discussed the basics of vector tracking. The mathematical model of the vector loop navigation filter presented in this chapter serves as a framework for the analyses of subsequent chapters. In the next chapter vector loop parametric models are developed from the existing vector architecture equations and performance benefits of the vector loops are explained with the help of these models. Subsequently, preliminary guidelines to obtain comparable performance of scalar and vector tracking architectures are presented.

Chapter 4

Vector Loop Parametric Models

4.1 Introduction

Chapters 2 and 3 discussed two basic types of GPS signal tracking architectures: Scalar and vector. It is apparent from their tracking methodologies that the vector architecture eliminates independent (or scalar) tracking loops by processing all satellite signals in unison. As shown in [10] and [11], such combined signal processing in vector tracking loops leverages the redundancy of an overdetermined system (when more than four satellites are in view) to attain higher tracking thresholds. However, the current literature (in particular, open literature) lacks a comprehensive analysis which could illuminate *detailed* internal workings of the vector architecture. In response to this challenge, the existing vector architecture equations are recast in this chapter into mathematical models that are easy to analyze. These models are then used as an analysis tool to offer a different, more intuitive perspective on the workings of the vector architecture. More importantly, they are used to explain why the vector architecture exhibits superior performance in low C/N_0 or GNSS-distressed environments.

The remaining chapter is organized as follows: The vector loop equations are first recast into a discrete parametric model from which transfer functions and noise bandwidths of vector loops are derived. Following this, a theoretical analysis of the discrete model is provided to explain the benefits of vector tracking. Next, simulation studies to support the theoretical analysis of the discrete parametric model are presented. Finally, preliminary guidelines for achieving comparable performance of scalar and vector loops are developed.

4.2 Parametric Models of Vector Tracking Loops

The navigation filter model and NCO corrections of the vector loops as described by Equations 3.3 through 3.8 in the previous chapter are rearranged into a discrete parametric model to gain insights into vector tracking's superiority. Then, under the assumption of quasi-stationary geometry, the discrete model is recast into a transfer function formulation. The phrase "quasi-stationary" geometry is taken to mean that the LOS vectors to satellites do not change over a multiple of T sec of Figure 3.2. The transfer function model derived in this manner is particularly useful for calculating vector loop noise bandwidths which serve as a good starting point for designing scalar loop noise bandwidths and, thus, facilitates a fair performance comparison of the two architectures. Furthermore, a close study of the discrete model reveals the key roles LOS geometry and the filter gain play in deciding several important properties of vector loops.

4.2.1 Discrete Parametric Model

The discrete parametric model has two components: The VFLL discrete parametric model and the VDLL discrete parametric model. Each of them is explained in detail below.

4.2.1.1 VFLL Discrete Parametric Model

The VFLL parametric model is described by the following equations.

$$\underline{\tilde{\dot{f}}}^{-}_{d,\,carr_{k+1}} = \underline{\tilde{\dot{f}}}^{+}_{d,\,carr_{k}} + T\underline{\tilde{\ddot{f}}}^{-}_{d,\,carr_{k+1}} \tag{4.1}$$

$$\underline{\tilde{\dot{f}}}^{+}_{d,\,carr_{k+1}} = \underline{\tilde{\dot{f}}}^{-}_{d,\,carr_{k+1}} + \dot{\tilde{G}}_{d,\,k+1}K_{v,\,k+1}\underline{e}_{k+1} + \tilde{G}_{d,\,k+1}K_{a,\,k+1}\underline{e}_{k+1} \tag{4.2}$$

$$\underline{\tilde{f}}^{-}_{d,\,carr_{k+1}} = \underline{\tilde{f}}^{+}_{d,\,carr_{k}} + T\underline{\tilde{\dot{f}}}^{-}_{d,\,carr_{k+1}} - \frac{1}{2}T^{2}\underline{\tilde{\ddot{f}}}^{-}_{d,\,carr_{k+1}} \tag{4.3}$$

$$\underline{\tilde{f}}^{+}_{d,\,carr_{k+1}} = \underline{\tilde{f}}^{-}_{d,\,carr_{k+1}} + \tilde{G}_{d,\,k+1}K_{v,\,k+1}\underline{e}_{k+1} \tag{4.4}$$

where

$\underline{\tilde{f}}^{+}_{d,\,carr_{k+1}}$ = estimated Doppler frequency vector at $k+1$

= $[\tilde{f}^{+}_{d_1,\,carr_{k+1}} \;\cdots\; \tilde{f}^{+}_{d_j,\,carr_{k+1}}]^{T}$

j = number of visible satellites

$\tilde{\underline{f}}^{-}_{d,\, carr_{k+1}}$ = predicted Doppler frequency vector at $k+1$

$\dot{\tilde{\underline{f}}}^{(-/+)}_{d,\, carr_{k+1}}$ = predicted/estimated rate of change of the Doppler at $k+1$

$\ddot{\tilde{\underline{f}}}^{-}_{d,\, carr_{k+1}}$ = predicted Doppler acceleration at $k+1$

$\tilde{G}_{d,\, k+1}$ = $\frac{f_{carr}}{c} \times$ geometry or LOS matrix; f_{carr} = carrier frequency

$\dot{\tilde{G}}_{d,\, k+1}$ = $\frac{(\tilde{G}_{d,\, k+1} - \tilde{G}_{d,\, k})}{T}$ = rate of change of $\tilde{G}_{d,\, k+1}$

$K_{v,\, k+1}$ = rows of the Kalman gain matrix K_{k+1} corresponding to the velocity states

$K_{a,\, k+1}$ = rows of the Kalman gain matrix K_{k+1} corresponding to the acceleration states

$\underline{e}_{k+1}$ = residual measurement vector at $k+1$ (Equation 3.3)

T = measurement update interval

Equations 4.1 through 4.4 form the discrete parametric model for VFLL, parameterized by the geometry matrix and Kalman gain. As the Kalman gain depends on P, Q, W and C matrices (Equations 3.1 through 3.4), the model parameters include estimated state uncertainties, *a priori* knowledge of model dynamics, C/N$_0$ and LOS geometry. It is instructive to observe from the $\underline{e}_{k+1}$ term of Equation 4.4 that the nonzero off-diagonal elements of the $\tilde{G}_{d,k+1}$ and $K_{v,\, k+1}$ matrices allow one satellite to *aid* another. A detailed discussion of this *internal aiding* among loops (which, in part, explains the better performance of the vector tracking architecture) is reserved for the latter part of this chapter.

Derivation of VFLL Discrete Parametric Model

In order to develop a parametric model for the VFLL, the following assumptions are made. First, the model is valid over a small time interval which usually lasts few seconds. Furthermore, T is small enough to assume $\tilde{G}_{d\, k+1}$ - $\tilde{G}_{d,\, k} = T\dot{\tilde{G}}_{d\, k+1}$. This is a valid assumption as T generally varies between 0.1 sec and 1 sec for the VFLL. Second, the geometry matrices obtained using the predicted and estimated user positions are equal or $\tilde{G}^{-}_{d,\, k+1} = \tilde{G}^{+}_{d,\, k+1} = \tilde{G}_{d,\, k+1}$. Third, the user and satellite accelerations are constant over the time period during which the model is valid i.e., $\underline{a}_{k+1} = \underline{a}_{k}$ and $\underline{a}_{s_i,\, k+1} = \underline{a}_{s_i,\, k}$. $\underline{a}$ is a 4×1 column vector containing the x, y and z components of user acceleration and the rate of user clock drift. The rate of drift of the user clock is

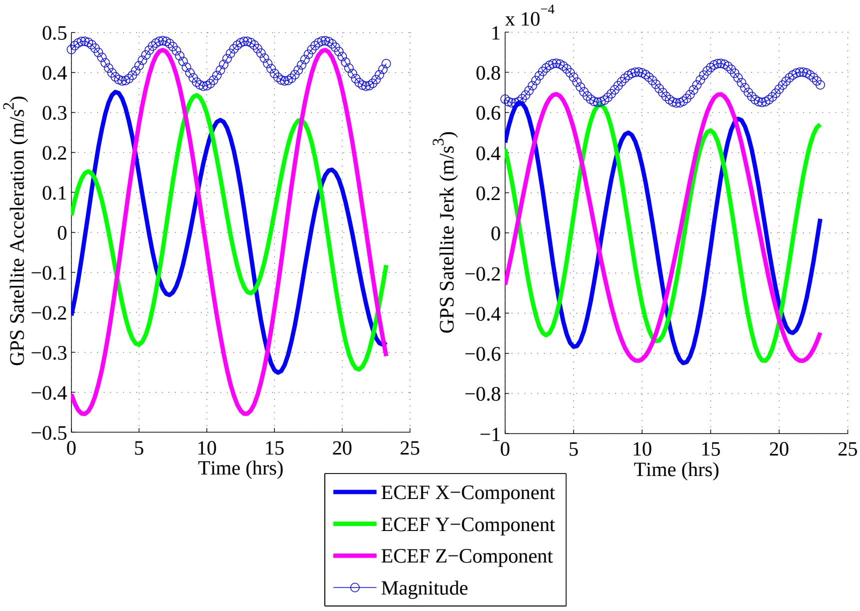


Figure 4.1: Acceleration and Jerk Components of a GPS satellite

usually not modeled, reducing the fourth element of $\underline{a}$ to zero. $\underline{a}_{s_i}$ is a 4-tuple column vector comprising the acceleration and clock drift rate of a GPS satellite, with i being the satellite index. As the GPS satellites are run by precise atomic clocks which are corrected on a regular basis by ground monitoring stations, their clock drift rates are assumed to be zero. Figure 4.1 depicts the acceleration and jerk of a GPS satellite in the earth centered earth fixed (ECEF) coordinate frame, as calculated from its precise ephemeris. It is evident from the figure that the satellite jerk magnitude is in the sub-millimeter level, thereby justifying the assumption of constant satellite acceleration over a short duration. It should be noted that the accelerations of all GPS satellites, although not shown here, vary within the range shown in the figure.

Next, the Doppler frequency for the i^{th} carrier NCO, as given by the first term of Equation 3.7, is rewritten as

$$\Delta \tilde{f}^{+}_{carr_{i,\,k+1}} = \frac{f_{carr}}{c}\left(-\dot{\tilde{R}}^{+}_{i,\,k+1} - \dot{\tilde{b}}^{+}_{k+1} + \dot{b}_{s_i,\,k+1}\right)$$

$$\text{or,}\quad \tilde{f}^{+}_{d_i,\,carr_{k+1}} = \frac{f_{carr}}{c}\left[\tilde{u}_{x_{i,\,k+1}} \quad \tilde{u}_{y_{i,\,k+1}} \quad \tilde{u}_{z_{i,\,k+1}} \quad -1\right] \begin{bmatrix} \dot{\tilde{x}}^{+}_{k+1} - \dot{x}_{s_i,\,k+1} \\ \dot{\tilde{y}}^{+}_{k+1} - \dot{y}_{s_i,\,k+1} \\ \dot{\tilde{z}}^{+}_{k+1} - \dot{z}_{s_i,\,k+1} \\ \dot{\tilde{b}}^{+}_{k+1} - \dot{b}_{s_i,\,k+1} \end{bmatrix}$$

$$
\begin{aligned}
\text{or,}\quad \tilde{f}^{+}_{d_i,\,carr_{k+1}} &= \tilde{G}_{d_i,\,k+1}(\tilde{V}^{+}_{k+1} - V_{s_i,\,k+1}) \\
\text{or,}\quad \tilde{f}^{+}_{d_i,\,carr_{k+1}} &= \tilde{G}_{d_i,\,k+1}\tilde{V}^{+}_{k+1} - c_{v_i,\,k+1} \qquad (4.5)
\end{aligned}
$$

where

$\tilde{G}_{d_i,\,k+1}$ = $f_{carr}/c\times$ i^{th} row of the geometry matrix

$\tilde{u}_{m_{i,\,k+1}}$ = m component of the LOS unit vector from satellite i to user at time $k+1$

$= \dfrac{m_{s_i,\,k+1} - \tilde{m}^{-}_{k+1}}{\tilde{R}^{-}_{i,\,k+1}}$, $m = x,\, y,\, z$

V_{k+1} = a column vector containing the x, y and z components of user velocity and user clock drift

$V_{s_i,\,k+1}$ = a column vector containing the x, y and z components of the velocity and clock drift of satellite i

$c_{v_i,\,k+1}$ = $\tilde{G}_{d_i,\,k+1}V_{s_i,\,k+1}$

Now, stacking all Doppler frequencies together in vector $\underline{f}_d$, Equation 4.5 can be expressed in a more compact form

$$
\tilde{\underline{f}}^{+}_{d,\,carr_{k+1}} = \tilde{G}_{d,\,k+1}\tilde{V}^{+}_{k+1} - C_{v,\,k+1} \qquad (4.6)
$$

Similarly, the predicted Doppler frequencies are given as

$$
\tilde{\underline{f}}^{-}_{d,\,carr_{k+1}} = \tilde{G}_{d,\,k+1}\tilde{V}^{-}_{k+1} - C_{v,\,k+1} \qquad (4.7)
$$

where $C_{v,\,k+1} = [c_{v_1,\,k+1}\ \dots\ c_{v_j,\,k+1}]^T$; $c_{v_i,\,k+1} = \tilde{G}_{d_i,\,k+1}V_{s_i,\,k+1}$; j is the number of visible satellites. Next, taking the first derivative of the expression $\underline{f}_d = G_dV - C_v$, the predicted and estimated rates of change of the Doppler at $k+1$ are given by

$$
\begin{aligned}
\tilde{\dot{\underline{f}}}^{-}_{d,\,carr_{k+1}} &= \frac{(\tilde{G}_{d,\,k+1} - \tilde{G}_{d,\,k})}{T}\tilde{V}^{-}_{k+1} + \tilde{G}_{d,\,k+1}\tilde{\underline{a}}^{+}_{k} - \frac{\Delta C_{v,\,k+1}}{T} - C_{a,\,k+1} \qquad (4.8) \\
\tilde{\dot{\underline{f}}}^{+}_{d,\,carr_{k+1}} &= \frac{(\tilde{G}_{d,\,k+1} - \tilde{G}_{d,\,k})}{T}\tilde{V}^{+}_{k+1} + \tilde{G}_{d,\,k+1}\tilde{\underline{a}}^{+}_{k+1} - \frac{\Delta C_{v,\,k+1}}{T} - C_{a,\,k+1} \qquad (4.9)
\end{aligned}
$$

where $\Delta C_{v,\,k+1} = [\Delta c_{v_1,\,k+1}\ \dots\ \Delta c_{v_j,\,k+1}]^T$; $\Delta c_{v_i,\,k+1} = (\tilde{G}_{d_i,\,k+1}$ - $\tilde{G}_{d_i,\,k})V_{s_i,\,k+1}$. $C_{a,\,k+1}$ $= [c_{a_1,\,k+1}\ \dots\ c_{a_j,\,k+1}]^T$; $c_{a_i,\,k+1} = \tilde{G}_{d_i,\,k+1}\underline{a}_{s_i,\,k+1}$. It should be noted that as the user acceleration is assumed to be constant, $\tilde{\underline{a}}^{-}_{k+1}$ of Equation 4.8 is replaced with $\tilde{\underline{a}}^{+}_{k}$. Subtracting Equation 4.8 from Equation 4.9, the following equation is obtained

$$
\tilde{\dot{\underline{f}}}^{+}_{d,\,carr_{k+1}} = \tilde{\dot{\underline{f}}}^{-}_{d,\,carr_{k+1}} + \frac{(\tilde{G}_{d,\,k+1} - \tilde{G}_{d,\,k})}{T}(\tilde{V}^{+}_{k+1} - \tilde{V}^{-}_{k+1}) + \tilde{G}_{d,\,k+1}(\tilde{\underline{a}}^{+}_{k+1} - \tilde{\underline{a}}^{+}_{k}) \qquad (4.10)
$$

The Kalman gain rows (Equation 3.4) corresponding to user position, velocity and acceleration are extracted into three matrices $K_{p,\,k+1}$, $K_{v,\,k+1}$ and $K_{a,\,k+1}$, respectively. Since the rate of user clock drift is usually not modeled, $K_{a,\,k+1}$ is augmented with a fourth row whose all elements are zero. This is required to match the dimensions of $\tilde{G}_{d,\,k}$ and $K_{a,\,k+1}$ for matrix multiplication purposes. Thus, the velocity and acceleration components of Equation 3.6 can be written as

$$\tilde{V}^{+}_{k+1} = \tilde{V}^{-}_{k+1} + K_{v,\,k+1}\underline{e}_{k+1} \tag{4.11}$$

$$\tilde{\underline{a}}^{+}_{k+1} = \tilde{\underline{a}}^{+}_{k} + K_{a,\,k+1}\underline{e}_{k+1} \tag{4.12}$$

Substituting the above two equations into Equation 4.10, the expression of Equation 4.2 is obtained below

$$\begin{aligned}
\tilde{\dot{\underline{f}}}^{+}_{d,\,carr_{k+1}} &= \tilde{\dot{\underline{f}}}^{-}_{d,\,carr_{k+1}} + \frac{(\tilde{G}_{d,\,k+1} - \tilde{G}_{d,\,k})}{T} K_{v,\,k+1}\underline{e}_{k+1} + \tilde{G}_{d,\,k+1} K_{a,\,k+1}\underline{e}_{k+1} \\
\tilde{\dot{\underline{f}}}^{+}_{d,\,carr_{k+1}} &= \tilde{\dot{\underline{f}}}^{-}_{d,\,carr_{k+1}} + \dot{\tilde{G}}_{d,\,k+1} K_{v,\,k+1}\underline{e}_{k+1} + \tilde{G}_{d,\,k+1} K_{a,\,k+1}\underline{e}_{k+1} \qquad (4.13)
\end{aligned}$$

Next, Equation 4.1 is derived as follows. The rate of change of the Doppler at time k is given by

$$\tilde{\dot{\underline{f}}}^{+}_{d,\,carr_{k}} = \frac{(\tilde{G}_{d,\,k} - \tilde{G}_{d,\,k-1})}{T}\tilde{V}^{+}_{k} + \tilde{G}_{d,\,k}\tilde{\underline{a}}^{+}_{k} - \frac{\Delta C_{v,\,k}}{T} - C_{a,\,k} \tag{4.14}$$

Subtracting Equation 4.14 from Equation 4.8 and assuming $(\tilde{G}_{d,\,k} - \tilde{G}_{d,\,k-1}) \approx (\tilde{G}_{d,\,k+1} - \tilde{G}_{d,\,k}) = T\dot{\tilde{G}}_{d,\,k+1}$ lead to

$$\begin{aligned}
\tilde{\dot{\underline{f}}}^{-}_{d,\,carr_{k+1}} = \tilde{\dot{\underline{f}}}^{+}_{d,\,carr_{k}} &+ \dot{\tilde{G}}_{d,\,k+1}(\tilde{V}^{-}_{k+1} - \tilde{V}^{+}_{k}) + T\dot{\tilde{G}}_{d,\,k+1}\tilde{\underline{a}}^{+}_{k} \\
&- \frac{1}{T}(\Delta C_{v,\,k+1} - \Delta C_{v,\,k}) - (C_{a,\,k+1} - C_{a,\,k}) \qquad (4.15)
\end{aligned}$$

Now, using $\tilde{V}^{-}_{k+1} - \tilde{V}^{+}_{k} = T\tilde{\underline{a}}^{+}_{k}$, the above equation reduces to

$$\tilde{\dot{\underline{f}}}^{-}_{d,\,carr_{k+1}} = \tilde{\dot{\underline{f}}}^{+}_{d,\,carr_{k}} + 2T\dot{\tilde{G}}_{d,\,k+1}\tilde{\underline{a}}^{+}_{k} - \frac{1}{T}(\Delta C_{v,\,k+1} - \Delta C_{v,\,k}) - (C_{a,\,k+1} - C_{a,\,k}) \tag{4.16}$$

Next, utilizing the assumption $(\tilde{G}_{d,\,k} - \tilde{G}_{d,\,k-1}) \approx (\tilde{G}_{d,\,k+1} - \tilde{G}_{d,\,k}) = T\dot{\tilde{G}}_{d,\,k+1}$ and definitions of ΔC_v on page 65, the third term of Equation 4.16 becomes

$$\frac{1}{T}(\Delta C_{v,\,k+1} - \Delta C_{v,\,k}) = \begin{bmatrix} \dot{\tilde{G}}_{d_1,\,k+1} V_{s_1,\,k+1} \\ \vdots \\ \dot{\tilde{G}}_{d_j,\,k+1} V_{s_j,\,k+1} \end{bmatrix} - \begin{bmatrix} \dot{\tilde{G}}_{d_1,\,k+1} V_{s_1,\,k} \\ \vdots \\ \dot{\tilde{G}}_{d_j,\,k+1} V_{s_j,\,k} \end{bmatrix}$$

$$= \begin{bmatrix} \dot{\tilde{G}}_{d_1,\,k+1}(V_{s_1,\,k+1} - V_{s_1,\,k}) \\ \vdots \\ \dot{\tilde{G}}_{d_j,\,k+1}(V_{s_j,\,k+1} - V_{s_j,\,k}) \end{bmatrix}$$

$$= T \begin{bmatrix} \dot{\tilde{G}}_{d_1,\,k+1}\underline{a}_{s_1,\,k+1} \\ \vdots \\ \dot{\tilde{G}}_{d_j,\,k+1}\underline{a}_{s_j,\,k+1} \end{bmatrix} \tag{4.17}$$

Since the satellite acceleration is assumed to be constant (i.e., $\underline{a}_{s_1,\,k} \approx \underline{a}_{s_1,\,k+1}$), the fourth term of Equation 4.16 reduces to

$$(C_{a,\,k+1} - C_{a,\,k}) = T \begin{bmatrix} \dot{\tilde{G}}_{d_1,\,k+1}\underline{a}_{s_1,\,k+1} \\ \vdots \\ \dot{\tilde{G}}_{d_j,\,k+1}\underline{a}_{s_j,\,k+1} \end{bmatrix} \tag{4.18}$$

Thus, the last three terms of Equation 4.16 are combined as

$$\begin{aligned} & 2T\dot{\tilde{G}}_{d,\,k+1}\tilde{\underline{a}}_k^+ - \frac{1}{T}(\Delta C_{v,\,k+1} - \Delta C_{v,\,k}) - (C_{a,\,k+1} - C_{a,\,k}) \\ & = 2T\dot{\tilde{G}}_{d,\,k+1}\tilde{\underline{a}}_k^+ - 2T \begin{bmatrix} \dot{\tilde{G}}_{d_1,\,k+1}\underline{a}_{s_1,\,k+1} \\ \vdots \\ \dot{\tilde{G}}_{d_j,\,k+1}\underline{a}_{s_j,\,k+1} \end{bmatrix} \end{aligned} \tag{4.19}$$

Now, taking the first derivative of the expression $\dot{\underline{f}}_d = \dot{G}_d V + G_d\underline{a} - \dot{C}_v$, ignoring second and higher derivatives of G_d and assuming the user accleration ($\underline{a}$) and satellite acceleration ($\underline{a}_{s_i}$) are constant, one obtains the following

$$\ddot{\underline{f}}_d = 2\dot{G}_d\underline{a} - 2 \begin{bmatrix} \dot{G}_{d_1}\underline{a}_{s_1} \\ \vdots \\ \dot{G}_{d_j}\underline{a}_{s_j} \end{bmatrix} \tag{4.20}$$

Comparing the above two equations, the last three terms of Equation 4.16 reduce to

$$2T\dot{\tilde{G}}_{d,\,k+1}\tilde{\underline{a}}_k^+ - \frac{1}{T}(\Delta C_{v,\,k+1} - \Delta C_{v,\,k}) - (C_{a,\,k+1} - C_{a,\,k}) = T\tilde{\ddot{\underline{f}}}^{-}_{d,\,carr_{k+1}} \tag{4.21}$$

It is important to note that $\tilde{\ddot{\underline{f}}}^{-}_{d,\,carr_{k+1}}$ is obtained by ignoring the second and higher order changes in the LOS vectors. Stated differently, it is assumed that the LOS vectors change linearly over time. This assumption is reasonable given the slow change of the LOS/geometry matrix. An upper bound on the change in LOS is derived in Appendix B. Thus, Equation 4.16 results in Equation 4.1

$$\tilde{\dot{\underline{f}}}^{-}_{d,\, carr_{k+1}} = \tilde{\dot{\underline{f}}}^{+}_{d,\, carr_{k}} + T\tilde{\ddot{\underline{f}}}^{-}_{d,\, carr_{k+1}} \tag{4.22}$$

Following this, Equations 4.3 and 4.4 are derived. For this purpose, Equation 4.8 is rearraned as follows:

$$\begin{aligned}
\tilde{\dot{\underline{f}}}^{-}_{d,\, carr_{k+1}} &= \frac{(\tilde{G}_{d,\, k+1} - \tilde{G}_{d,\, k})}{T}\tilde{V}^{-}_{k+1} + \tilde{G}_{d,\, k+1}\tilde{\underline{a}}^{+}_{k} - \frac{\Delta C_{v,\, k+1}}{T} - C_{a,\, k+1} \\
\text{or,}\quad \tilde{G}_{d,\, k+1}\tilde{V}^{-}_{k+1} &= \tilde{G}_{d,\, k}\tilde{V}^{-}_{k+1} + T\tilde{\dot{\underline{f}}}^{-}_{d,\, carr_{k+1}} - T\tilde{G}_{d,\, k+1}\tilde{\underline{a}}^{+}_{k} + \Delta C_{v,\, k+1} + TC_{a,\, k+1} \\
\text{or,}\quad \tilde{G}_{d,\, k+1}\tilde{V}^{-}_{k+1} - C_{v,\, k+1} &= (\tilde{G}_{d,\, k}\tilde{V}^{+}_{k} - C_{v,\, k}) + T\tilde{\dot{\underline{f}}}^{-}_{d,\, carr_{k+1}} + (\tilde{G}_{d,\, k}\tilde{V}^{-}_{k+1} - \tilde{G}_{d,\, k}\tilde{V}^{+}_{k}) \\
&\quad -T\tilde{G}_{d,\, k+1}\tilde{\underline{a}}^{+}_{k} + (\Delta C_{v,\, k+1} + TC_{a,\, k+1} + C_{v,\, k} - C_{v,\, k+1})
\end{aligned}$$

Using Equations 4.6 and 4.7 and $\tilde{V}^{-}_{k+1} - \tilde{V}^{+}_{k} = T\tilde{\underline{a}}^{+}_{k}$, one gets

$$\begin{aligned}
\text{or,}\quad \tilde{\underline{f}}^{-}_{d,\, carr_{k+1}} &= \tilde{\underline{f}}^{+}_{d,\, carr_{k}} + T\tilde{\dot{\underline{f}}}^{-}_{d,\, carr_{k+1}} + (T\tilde{G}_{d,\, k}\tilde{\underline{a}}^{+}_{k} - T\tilde{G}_{d,\, k+1}\tilde{\underline{a}}^{+}_{k}) \\
&\quad + (\Delta C_{v,\, k+1} + TC_{a,\, k+1} + C_{v,\, k} - C_{v,\, k+1}) \\
\text{or,}\quad \tilde{\underline{f}}^{-}_{d,\, carr_{k+1}} &= \tilde{\underline{f}}^{+}_{d,\, carr_{k}} + T\tilde{\dot{\underline{f}}}^{-}_{d,\, carr_{k+1}} - T^{2}\dot{\tilde{G}}_{d,\, k+1}\tilde{\underline{a}}^{+}_{k} \\
&\quad + (\Delta C_{v,\, k+1} + TC_{a,\, k+1} + C_{v,\, k} - C_{v,\, k+1})
\end{aligned} \tag{4.23}$$

With the help of the definitions of its elements given on page 65, The last term of the above equation is given by

$$\begin{aligned}
\Delta C_{v,\, k+1} + TC_{a,\, k+1} + C_{v,\, k} - C_{v,\, k+1} &= \Delta C_{v,\, k+1} + TC_{a,\, k+1} - (C_{v,\, k+1} - C_{v,\, k}) \\
&= \Delta C_{v,\, k+1} + TC_{a,\, k+1} - \\
&\qquad \begin{bmatrix} \tilde{G}_{d_1,\, k+1}V_{s_1,\, k+1} - \tilde{G}_{d_1,\, k}V_{s_1,\, k} \\ \vdots \\ \tilde{G}_{d_j,\, k+1}V_{s_j,\, k+1} - \tilde{G}_{d_j,\, k}V_{s_j,\, k} \end{bmatrix} \\
\text{or,}\quad \Delta C_{v,\, k+1} + TC_{a,\, k+1} + C_{v,\, k} - C_{v,\, k+1} &= \Delta C_{v,\, k+1} + TC_{a,\, k+1} - \\
&\qquad \begin{bmatrix} (\tilde{G}_{d_1,\, k+1} - \tilde{G}_{d_1,\, k})V_{s_1,\, k+1} + \tilde{G}_{d_1,\, k}(V_{s_1,\, k+1} - V_{s_1,\, k}) \\ \vdots \\ (\tilde{G}_{d_j,\, k+1} - \tilde{G}_{d_j,\, k})V_{s_j,\, k+1} + \tilde{G}_{d_j,\, k}(V_{s_j,\, k+1} - V_{s_j,\, k}) \end{bmatrix}
\end{aligned}$$

$$\text{or,}\quad \Delta C_{v,\,k+1} + TC_{a,\,k+1} + C_{v,\,k} - C_{v,\,k+1} = \Delta C_{v,\,k+1} + TC_{a,\,k+1} - \begin{bmatrix} (\tilde{G}_{d_1,\,k+1} - \tilde{G}_{d_1,\,k})V_{s_1,\,k+1} + T\tilde{G}_{d_1,\,k}\underline{a}_{s_1,\,k+1} \\ \vdots \\ (\tilde{G}_{d_j,\,k+1} - \tilde{G}_{d_j,\,k})V_{s_1,\,k+1} + T\tilde{G}_{d_j,\,k}\underline{a}_{s_j,\,k+1} \end{bmatrix}$$

As $\underline{a}_{s_i,\,k+1} \approx \underline{a}_{s_i,\,k}$

$$\Delta C_{v,\,k+1} + TC_{a,\,k+1} + C_{v,\,k} - C_{v,\,k+1} = \Delta C_{v,\,k+1} + TC_{a,\,k+1} - T\begin{bmatrix} \dot{\tilde{G}}_{d_1,\,k+1}V_{s_1,\,k+1} \\ \vdots \\ \dot{\tilde{G}}_{d_j,\,k+1}V_{s_j,\,k+1} \end{bmatrix} - T\begin{bmatrix} \tilde{G}_{d_1,\,k}\underline{a}_{s_1,\,k} \\ \vdots \\ \tilde{G}_{d_j,\,k}\underline{a}_{s_j,\,k} \end{bmatrix}$$

$$\text{or,}\quad \Delta C_{v,\,k+1} + TC_{a,\,k+1} + C_{v,\,k} - C_{v,\,k+1} = \Delta C_{v,\,k+1} + TC_{a,\,k+1} - (\Delta C_{v,\,k+1} + TC_{a,\,k}) \tag{4.24}$$

Substituting the above into Equation 4.23 and using Equations 4.18, 4.19 and 4.21 result in Equation 4.3 below

$$\begin{aligned} \underline{\tilde{f}}^{-}_{d,\,carr_{k+1}} &= \underline{\tilde{f}}^{+}_{d,\,carr_k} + T\dot{\underline{\tilde{f}}}^{-}_{d,\,carr_{k+1}} - T^2\dot{\tilde{G}}_{d,\,k+1}\underline{\tilde{a}}^{+}_{k} + (\Delta C_{v,\,k+1} - \Delta C_{v,\,k+1}) \\ &\qquad + T(C_{a,\,k+1} - C_{a,\,k}) \\ &= \underline{\tilde{f}}^{+}_{d,\,carr_k} + T\dot{\underline{\tilde{f}}}^{-}_{d,\,carr_{k+1}} - T^2\dot{\tilde{G}}_{d,\,k+1}\underline{\tilde{a}}^{+}_{k} + T(C_{a,\,k+1} - C_{a,\,k}) \\ &= \underline{\tilde{f}}^{+}_{d,\,carr_k} + T\dot{\underline{\tilde{f}}}^{-}_{d,\,carr_{k+1}} - T^2\dot{\tilde{G}}_{d,\,k+1}\underline{\tilde{a}}^{+}_{k} + T^2\begin{bmatrix} \dot{\tilde{G}}_{d_1,\,k+1}\underline{a}_{s_1,\,k+1} \\ \vdots \\ \dot{\tilde{G}}_{d_j,\,k+1}\underline{a}_{s_j,\,k+1} \end{bmatrix} \\ &= \underline{\tilde{f}}^{+}_{d,\,carr_k} + T\dot{\underline{\tilde{f}}}^{-}_{d,\,carr_{k+1}} - \frac{1}{2}T^2\ddot{\underline{\tilde{f}}}^{-}_{d,\,carr_{k+1}} \end{aligned} \tag{4.25}$$

Next, taking the difference between Equations 4.6 and 4.7 and using Equation 4.11, Equation 4.4 is obtained as follows:

$$\begin{aligned} \underline{\tilde{f}}^{+}_{d,\,carr_{k+1}} &= \underline{\tilde{f}}^{-}_{d,\,carr_{k+1}} + \tilde{G}_{d,\,k+1}(\tilde{V}^{+}_{k+1} - \tilde{V}^{-}_{k+1}) \\ &= \underline{\tilde{f}}^{-}_{d,\,carr_{k+1}} + \tilde{G}_{d,\,k+1}K_{v,\,k+1}\underline{e}_{k+1} \end{aligned} \tag{4.26}$$

4.2.1.2 VDLL Discrete Parametric Model

It should be noted that the code NCO corrections given by Equation 3.8 have two components: A frequency correction to align the code phase to the corresponding predicted or estimated pseudorange and the code Doppler frequency. Since the code Doppler can be obtained by appropriately scaling the carrier Doppler, it is assumed that the VDLL is aided by the VFLL loop. Thus, a parametric model of only the first component is developed for the VDLL. Mathematically, the first component $f^{+}_{\tau_i,\,code_{k+1}}$ is given by

$$\tilde{f}^{+}_{\tau_i,\,code_{k+1}} = \frac{\Delta\widetilde{cp}_{i,\,k+1}}{\Delta t} = \frac{1}{\Delta t}\frac{f_{code}}{c}(\rho_{rec_{i,\,k+1}} - \tilde{\rho}^{+}_{i,\,k+1}) \tag{4.27}$$

$\widetilde{cp}_{i,\,k+1}$ is the receiver-predicted code phase of satellite i at t_{k+1} and $\Delta\widetilde{cp}_{i,k+1}$ denotes its change over time Δt sec. After some algebra (see the derivation next for details), the above equation reduces to

$$\underline{\tilde{f}}^{+}_{\tau,\,code_{k+1}} = \frac{\Delta\underline{\widetilde{cp}}_{k+1}}{\Delta t} = \frac{1}{\Delta t}\tilde{G}_{\tau,\,k+1}K_{p,\,k+1}\underline{e}_{k+1} \tag{4.28}$$

where
$\underline{\tilde{f}}^{+}_{\tau,\,code_{k+1}}$ $=[\tilde{f}^{+}_{\tau_1,\,code_{k+1}} \dots \tilde{f}^{+}_{\tau_j,code_{k+1}}]^T$
j = number of visible satellites
$\underline{\widetilde{cp}}_{k+1}$ = receiver-predicted code phase vector of all satellites
$\tilde{G}_{\tau,\,k+1}$ = $f_{code}/f_{carr}\tilde{G}_{d,\,k+1}$; f_{code} = code chipping rate
$K_{p,\,k+1}$ = rows of the Kalman gain matrix K_{k+1} corresponding to the position states

Again from Equation 4.28 it can be seen that the error in one channel contributes to correcting other channels, resulting in internal aiding among loops. Later, Equations 4.1 - 4.4 and 4.28 are used to develop a transfer function representation of the vector architecture and subsequently an approximate expression of its noise bandwidths is derived.

Derivation of VDLL Discrete Parametric Model

First, Equation 4.27 is rewritten below

$$\tilde{f}^{+}_{\tau_i,\,code_{k+1}} = \frac{\Delta\widetilde{cp}_{i,\,k+1}}{\Delta t} = \frac{1}{\Delta t}\frac{f_{code}}{c}(\rho_{rec_{i,\,k+1}} - \tilde{\rho}^{+}_{i,\,k+1}) \tag{4.29}$$

$\widetilde{cp}_{i,k+1}$ is the receiver-predicted code phase of satellite i at t_{k+1} and $\Delta\widetilde{cp}_{i,\,k+1}$ denotes its change over time Δt. Since the NCO command is applied at every Δt sec based on the predicted states of the navigation filter, $\rho_{rec_{i},\,k+1}$ in the above equation is equal to the navigation filter-predicted pseudorange at t_{k+1}. Hence, the above equation reduces to

$$\tilde{f}^{+}_{\tau_i,\,code_{k+1}} = \frac{1}{\Delta t}\frac{f_{code}}{c}(\tilde{\rho}^{-}_{i,\,k+1} - \tilde{\rho}^{+}_{i,\,k+1})$$

Now, considering only the first order term of the Taylor series expansion of $(\tilde{\rho}^{-}_{i,\,k+1} - \tilde{\rho}^{+}_{i,\,k+1})$, the following equation is obtained

$$\tilde{f}^{+}_{\tau_i,\,code_{k+1}} = -\frac{1}{\Delta t}(-\tilde{G}_{\tau_i,\,k+1})(\tilde{P}^{+}_{i,\,k+1} - \tilde{P}^{-}_{i,\,k+1}) \tag{4.30}$$

where

$\tilde{G}_{\tau_i,\,k+1}$ = $f_{code}/f_{carr}\tilde{G}_{d_i,\,k+1}$; $\tilde{G}_{d_i,\,k+1}$ is defined below Equation 4.5
$\tilde{P}_{i,\,k+1}$ = $[\tilde{x}_{k+1} \quad \tilde{y}_{k+1} \quad \tilde{z}_{k+1} \quad \tilde{b}_{k+1}]^T$
$-[x_{s_i,\,k+1} \quad y_{s_i,\,k+1} \quad z_{s_i,\,k+1} \quad b_{s_i,\,k+1}]^T$
or, $\tilde{P}_{i,\,k+1}$ = $\tilde{P}_{u,\,k+1}$ - $P_{s_i,\,k+1}$
$\tilde{P}_u$ = predicted/estimated user position and clock bias
P_{s_i} = position and clock bias of the i^{th} satellite

Note that $P_{s_i,\,k+1}$ in Equation 4.30 cancels out as it is the same for both predicted and estimated pseudoranges. Substituting Equation 3.6 into Equation 4.30 and considering only the position related part of the Kalman gain i.e., K_p, one can obtain

$$\tilde{f}^{+}_{\tau_i,\,code_{k+1}} = \frac{\Delta\widetilde{cp}_{i,\,k+1}}{\Delta t} = \frac{1}{\Delta t}\tilde{G}_{\tau_i,\,k+1}K_{p_i,\,k+1}\underline{e}_{k+1}$$

It is important to note that the above equation holds only after measurement updates. If the NCO corrections are generated as discussed in Chapter 3, then the receiver-generated pseudorange measurement and the predicted pseudorange measurement of the navigation filter will match each other after a time update, resulting in zero $\underline{\tilde{f}}^{-}_{\tau_i,\,code_{k+1}}$.

Now, representing all channels together by $\underline{\tilde{f}}^{+}_{\tau,\,code_{k+1}}$ and all $\widetilde{cp}_i$s by $\underline{\widetilde{cp}}$, Equation 4.28 is obtained

$$\underline{\tilde{f}}^{+}_{\tau,\,code_{k+1}} = \frac{\Delta\underline{\widetilde{cp}}_{k+1}}{\Delta t} = \frac{1}{\Delta t}\tilde{G}_{\tau,\,k+1}K_{p,\,k+1}\underline{e}_{k+1} \tag{4.31}$$

4.2.2 Transfer Function Model

The transfer function representation of vector loops is particularly useful for deriving the vector loop noise bandwidths. Vector loop bandwidths are an important design parameter. This is because they can be used as a good starting point for designing scalar loop noise bandwidths to achieve scalar loop performance comparable to that of vector architecture. The bandwidths also help understand some of the key features of the vector architecture, as will be discussed later in this chapter.

A close observation of the discrete parametric models reveals that the model parameters vary with changes in satellite visibility, C/N$_0$ ratios and LOS geometry. In order to derive a transfer function formulation from such a time varying model for offline analysis, the entire time interval of interest can be divided into a number of sub-intervals, with each sub-interval having a separate transfer function model. The duration (T_{sub}) of each sub-interval is chosen such that the following conditions are satisfied:

1. Number of satellites (j) does not change in T_{sub}.
2. If $\dot{u}_{\text{avg}}$ denotes the average of all element absolute values of the rate of change of the geometry matrix $\dot{\tilde{G}}$ and u_{avg} represents the mean of the absolute values of all elements of the geometry matrix $\tilde{G}$, then T_{sub} is chosen such that $\dot{u}_{\text{avg}}T_{\text{sub}}$ is less than $u_{\text{avg}}/10$. The factor of 10 is an empirical value based on observation of LOS dynamics. In a qualitative sense, T_{sub} is chosen such that the LOS vectors do not change significantly over that sub-interval. Thus, the $\tilde{G}$ matrix can be represented by a set of piecewise constant matrices. Each matrix is valid over a sub-interval and obtained by calculating $\tilde{G}$ at the beginning of that sub-interval.
3. C/N$_0$ does not vary more than (2 - 3) dB-Hz over T_{sub}.

To formulate transfer function models in real time, T_{sub} can be chosen to be a few ($\backsim$ 5) seconds over which all the above conditions will generally be valid. Further, the following assumptions apply, which help formulate the transfer function models:

1. If the initial transients have died down and all the above conditions are satisfied, then the Kalman gain matrix can be approximated to be a constant matrix over a sub-interval. The constant matrix is equal to the actual Kalman gain at the beginning of T_{sub}.
2. It is assumed that $\dot{\underline{f}}_d >> \ddot{\underline{f}}_d$ so that the last term of Equations 4.1 and 4.3 can be ignored. A justification for this assumption is provided next. For this purpose the mathematical expressions of $\dot{\underline{f}}_d$ and $\ddot{\underline{f}}_d$ from page 67 are rewritten below

$$\dot{\underline{f}}_d = \dot{G}_d V \ + G_d \underline{a} \ - \ \dot{C}_v \tag{4.32}$$

$$\ddot{\underline{f}}_d = 2\dot{G}_d \underline{a} - 2 \begin{bmatrix} \dot{G}_{d1} \underline{a}_{s_1} \\ \vdots \\ \dot{G}_{d_j} \underline{a}_{s_j} \end{bmatrix} \tag{4.33}$$

where $G_d = \dfrac{f_{carr}}{c} \times$ LOS/geometry matrix; $G_{d_i} = f_{carr}/c \times$ i^{th} row of the geometry matrix; c is the speed of light; f_{carr} = carrier frequency; V is a 4×1 column vector containing the x, y and z components of user velocity and clock drift; $\underline{a}_{s_i}$ is a 4×1 column vector consisting of the x, y and z components of the acceleration and clock drift rate of i^{th} satellite; $\underline{a}$ is a column vector of user acceleration and clock drift rate; C_v is defined below Equation 4.7. The user clock drift rate is assumed to be zero and user acceleration is modeled as a constant vector.

It should be noted that $\dot{G}_d V$ in $\dot{\underline{f}}_d$ is usually very small due to slow rate of change of the LOS vectors. A conservative upper bound on the elements of the rate of change of the LOS vectors is derived in Appendix B as 4.78×10^{-4} sec^{-1}. It becomes 0.0025 Hz/m when multiplied by the factor f_{carr}/c for the L1 carrier frequency. Hence, $\dot{G}_d V$ is not significant unless the magnitude of V is on the order of the speed of sound which is unlikely in most cases. The last term ($\dot{C}_v$) of $\dot{\underline{f}}_d$ is the rate of change of Doppler due to satellite motion. It is less than 1 Hz/sec for a stationary receiver [69] and depending on the LOS rate of change it can be a few Hz/sec (< 15) for a moving receiver. Thus, $\dot{\underline{f}}_d$ is mainly determined by $G_d \underline{a}$ and $\dot{C}_v$ when it is not small (i.e., when each element $>> 1$ Hz/sec).

Substituting the upper bound for the LOS rate of change and the maximum satellite acceleration (Figure 4.1) into the second term of $\ddot{\underline{f}}_d$ (Equation 4.33), an upper bound for all elements of this term can be obtained. More specifically, each element of the second term is a dot product of its rate of change of the LOS vector multiplied by $\dfrac{f_{carr}}{c}$ and the corresponding satellite acceleration vector. Thus, it is bounded by the product of the magnitude of the $\dfrac{f_{carr}}{c} \times$LOS rate vector and the magnitude of satellite acceleration. An upper bound for the magnitude of the $\dfrac{f_{carr}}{c} \times$LOS rate vector is $\sqrt{3} \times 0.0025$ Hz/m and that of the satellite acceleration is 0.5 m/sec^2. Hence, each element of the second term of Equation 4.33 is bounded by the value 0.0044 Hz/sec^2. Being very small, the second term of Equation 4.33 can be ignored for all practical purposes. With regard to the first term of $\ddot{\underline{f}}_d$, it should be noted that it depends on user acceleration $\underline{a}$. However, as argued above, since the rate of change of the $f_{carr}/c \times$LOS vector is very small (each element is less than 0.0025 Hz/m), this term is not significant unless $\underline{a}$ is on the order of few g (= 9.81 m/s^2). Moreover, each component of the LOS vector is typically much larger than its rate of change, resulting in the elements of $G_d\underline{a}$ of $\dot{\underline{f}}_d$ being much larger in magnitude than the corresponding elements of $\dot{G}_d\underline{a}$ of $\ddot{\underline{f}}_d$. So, $\dot{\underline{f}}_d$ and $\ddot{\underline{f}}_d$ ($\approx 2\dot{G}_d\underline{a}$) can be comparable (i.e., on the same order) if ($\dot{G}_dV$ - $\dot{C}_v$) of $\dot{\underline{f}}_d$ almost cancels out $G_d\underline{a}$. However, $\dot{C}_v$ and $\dot{G}_dV$ cannot be large, as discussed before. This implies that the elements of $G_d\underline{a}$ are of small to moderate values ($\backsim$ 15) for $\dot{\underline{f}}_d$ and $\ddot{\underline{f}}_d$ to be comparable. Since the elements of $\dot{G}_d\underline{a}$ are much smaller than those of $G_d\underline{a}$, both $\dot{\underline{f}}_d$ and $\ddot{\underline{f}}_d$ will be small when comparable and can be ignored. Therefore, the assumption $\dot{\underline{f}}_d >> \ddot{\underline{f}}_d$ is a valid assumption.

3. As stated before, the elements of $\dot{G}_d$ are always less than 0.0025 Hz/m which is a highly conservative upper bound. The term $\dot{\tilde{G}}_{d,\,k+1}K_{v,\,k+1}\underline{e}_{k+1}$ of Equation 4.2 can, thus, be neglected assuming each element of the $K_{v,\,k+1}\underline{e}_{k+1}$ vector remains within a few meters which are again conservative if dynamics are correctly modeled.

4. Finally, it is found from the simulation studies discussed later in this chapter and from experimental data (chapter 6) that the contributions of the code phase errors D_{Ri} to correcting the Doppler frequencies (see $\underline{e}_{k+1}$ of Equations 3.3 and 4.4) are at least an order or two less than that of the Doppler frequency errors D_{RR_i}. Likewise, D_{Ri} supersedes D_{RR_i} when updating τ (Equations 3.3 and 4.28). Consequently, the VDLL and VFLL are decoupled by assuming that D_{Ri} corrects only τ and D_{RR_i} corrects only f_d.

Next, the VFLL and VDLL transfer function models for each sub-interval are developed below.

4.2.2.1 VFLL Transfer Function Model

Applying the above assumptions and replacing $\tilde{\underline{f}}_{d,\,carr_{k+1}}$ with $\tilde{\underline{f}}_{d,\,k+1}$ for ease of notation, the VFLL paramtric model (Equations 4.1 - 4.4) reduces to

$$\dot{\tilde{\underline{f}}}^{-}_{d,\,k+1} = \dot{\tilde{\underline{f}}}^{+}_{d,\,k} \tag{4.34}$$

$$\dot{\tilde{\underline{f}}}^{+}_{d,\,k+1} = \dot{\tilde{\underline{f}}}^{-}_{d,\,k+1} + G_{d,\,l}K_{a,\,l}\underline{e}_{d,\,k+1} \tag{4.35}$$

$$\tilde{\underline{f}}^{-}_{d,\,k+1} = \tilde{\underline{f}}^{+}_{d,\,k} + T\dot{\tilde{\underline{f}}}^{-}_{d,\,k+1} \tag{4.36}$$

$$\tilde{\underline{f}}^{+}_{d,\,k+1} = \tilde{\underline{f}}^{-}_{d,\,k+1} + G_{d,\,l}K_{v,\,l}\underline{e}_{d,\,k+1} \tag{4.37}$$

where $\underline{e}_{d,\,k+1}$ consists of only the frequency error components of $\underline{e}_{k+1}$ i.e., $\underline{e}_{d,k+1} = [\Delta f_{d_1,\,k+1} \ldots \Delta f_{d_j,\,k+1}]^T$ and j = number of visible satellites. $G_{d,\,l}$ is the $\tilde{G}_d$ matrix for any l^{th} sub-interval spanning time instants, say $[k+1,\ \ldots,\ k+n]$; $G_{d,\,l} = \tilde{G}_{d,\,k+1}$. The Kalman gain matrices for the l^{th} sub-interval are denoted as $K_{a,\,l}$ and $K_{v,\,l}$; $K_{v,\,l}$ $= K_{v,\,k+1}$ and $K_{a,\,l} = K_{a,\,k+1}$. It should be noted that $K_{v,\,l}$ and $K_{a,\,l}$ have only those elements that correspond to Δf_d. Next, substituting Equation 4.34 into Equation 4.35 and Equations 4.34 and 4.36 into Equation 4.37, the following equations are obtained

$$\tilde{\dot{\underline{f}}}^{+}_{d,\,k+1} = \tilde{\dot{\underline{f}}}^{+}_{d,\,k} + G_{d,\,l}K_{a,\,l}\underline{e}_{d,\,k+1} \tag{4.38}$$

$$\tilde{\underline{f}}^{+}_{d,\,k+1} = \tilde{\underline{f}}^{+}_{d,\,k} + T\tilde{\dot{\underline{f}}}^{+}_{d,\,k} + G_{d,\,l}K_{v,\,l}\underline{e}_{d,\,k+1} \tag{4.39}$$

Now, dropping the superscripts from above equations and expanding $\underline{e}_{d,\,k+1}$, one gets

$$\tilde{\dot{\underline{f}}}_{d,\,k+1} = \tilde{\dot{\underline{f}}}_{d,\,k} + G_{d,\,l}K_{a,\,l}(\underline{f}_{d,\,k+1} - \tilde{\underline{f}}_{d,\,k} - T\tilde{\dot{\underline{f}}}_{d,\,k}) \tag{4.40}$$

$$\tilde{\underline{f}}_{d,\,k+1} = \tilde{\underline{f}}_{d,\,k} + T\tilde{\dot{\underline{f}}}_{d,\,k} + G_{d,\,l}K_{v,\,l}(\underline{f}_{d,\,k+1} - \tilde{\underline{f}}_{d,\,k} - T\tilde{\dot{\underline{f}}}_{d,\,k}) \tag{4.41}$$

where $\underline{f}_{d,\,k+1}$ is the incoming Doppler frequency vector. When T, the time interval between two updates $(t_{k+1} - t_k)$, is small (i.e., $T \rightarrow 0$), the continuous Kalman gains are given as $K_{v_{cont},\,l} = K_{v,\,l}/T$ and $K_{a_{cont},\,l} = K_{a,\,l}/T$ [78]. Next, rearranging the terms of the above equations and taking the limit on both sides lead to

$$\begin{aligned}
\lim_{T\rightarrow 0} \frac{\tilde{\dot{\underline{f}}}_{d,\,k+1} - \tilde{\dot{\underline{f}}}_{d,\,k}}{T} &= \lim_{T\rightarrow 0} \frac{G_{d,\,l}K_{a,\,l}}{T}(\underline{f}_{d,\,k+1} - \tilde{\underline{f}}_{d,\,k} - T\tilde{\dot{\underline{f}}}_{d,\,k}) \\
\dot{\tilde{\dot{\underline{f}}}}_{d} &= G_{d,\,l}K_{a_{cont},\,l}(\underline{f}_d - \tilde{\underline{f}}_d) \qquad (4.42) \\
\lim_{T\rightarrow 0} \frac{\tilde{\underline{f}}_{d,\,k+1} - \tilde{\underline{f}}_{d,\,k}}{T} &= \tilde{\dot{\underline{f}}}_{d,k} + \lim_{T\rightarrow 0} \frac{G_{d,\,l}K_{v,\,l}}{T}(\underline{f}_{d,\,k+1} - \tilde{\underline{f}}_{d,\,k} - T\tilde{\dot{\underline{f}}}_{d,\,k}) \\
\dot{\tilde{\underline{f}}}_{d} &= \tilde{\dot{\underline{f}}}_{d} + G_{d,\,l}K_{v_{cont},\,l}(\underline{f}_d - \tilde{\underline{f}}_d) \qquad (4.43)
\end{aligned}$$

Taking the Laplace transform of the above equations, one obtains

$$s\tilde{\dot{\underline{f}}}_d(s) = G_{d,\,l}K_{a_{cont},\,l}(\underline{f}_d(s) - \tilde{\underline{f}}_d(s)) \tag{4.44}$$

$$s\tilde{\underline{f}}_d(s) = \tilde{\dot{\underline{f}}}_d(s) + G_{d,\,l}K_{v_{cont},\,l}(\underline{f}_d(s) - \tilde{\underline{f}}_d(s)) \tag{4.45}$$

Substituting Equation 4.44 into Equation 4.45 and rearranging the terms, the VFLL transfer function model for the l^{th} sub-interval is obtained as follows:

$$s\tilde{\underline{f}}_d(s) = \frac{G_{d,\,l}K_{a_{cont},\,l}}{s}[\underline{f}_d(s) - \tilde{\underline{f}}_d(s)] + G_{d,\,l}K_{v_{cont},\,l}[\underline{f}_d(s) - \tilde{\underline{f}}_d(s)]$$

$$\begin{aligned}
\text{or, } \left[sI_{j\times j} + \frac{G_{d,\,l}K_{a_{cont},\,l}}{s} + G_{d,\,l}K_{v_{cont},\,l}\right]\underline{\tilde{f}}_{d}(s) &= \left[\frac{G_{d,\,l}K_{a_{cont},\,l}}{s} + G_{d,\,l}K_{v_{cont},\,l}\right]\underline{f}_{d}(s)\\
\text{or, } \underline{\tilde{f}}_{d}(s) = \left[sI_{j\times j} + \frac{G_{d,\,l}K_{a_{cont},\,l}}{s} + G_{d,\,l}K_{v_{cont},\,l}\right]^{-1} &\left[\frac{G_{d,\,l}K_{a_{cont},\,l}}{s} + G_{d,\,l}K_{v_{cont},\,l}\right]\underline{f}_{d}(s)\\
\text{or, } \underline{\tilde{f}}_{d}(s) &= [H_{VFLL_{cont},\,l}]_{j\times j}\underline{f}_{d}(s)
\end{aligned} \tag{4.46}$$

$H_{VFLL_{cont},\,l}$ in Equation 4.46 represents the $j \times j$ transfer function matrix of VFLL in the continuous domain for sub-interval l. It maps the true Doppler $\underline{f}_{d}(s)$ to the estimated Doppler $\underline{\tilde{f}}_{d}(s)$. It is instructive to note that it represents a sub-optimal implementation because of the underlying assumptions that are used to determine the transfer function models.

Equations 4.40 and 4.41 indicate that the discrete-domain or z-domain representation of the transfer function matrix requires that $\underline{\dot{\tilde{f}}}_{d,\,k+1}$ and $\underline{\tilde{f}}_{d,\,k+1}$ are grouped together as a single vector and, thus, a joint transfer function model of $\underline{\dot{\tilde{f}}}_{d,\,k+1}$ and $\underline{\tilde{f}}_{d,\,k+1}$ can be derived. This is shown below.

$$\begin{bmatrix}\underline{\tilde{f}}_{d,\,k+1}\\ \underline{\dot{\tilde{f}}}_{d,\,k+1}\end{bmatrix} = \begin{bmatrix} I_{j\times j} - G_{d,\,l}K_{v,\,l} & T(I_{j\times j} - G_{d,\,l}K_{v,\,l})\\ -G_{d,\,l}K_{a,\,l} & I_{j\times j} - TG_{d,\,l}K_{a,\,l}\end{bmatrix}\begin{bmatrix}\underline{\tilde{f}}_{d,\,k}\\ \underline{\dot{\tilde{f}}}_{d,\,k}\end{bmatrix} + \begin{bmatrix}G_{d,\,l}K_{v,\,l}\\ G_{d,\,l}K_{a,\,l}\end{bmatrix}\underline{f}_{d,\,k+1}$$

Taking the Z transform gives

$$\begin{aligned}
z\begin{bmatrix}\underline{\tilde{f}}_{d}(z)\\ \underline{\dot{\tilde{f}}}_{d}(z)\end{bmatrix} &= \begin{bmatrix} I_{j\times j} - G_{d,\,l}K_{v,\,l} & T(I_{j\times j} - G_{d,\,l}K_{v,\,l})\\ -G_{d,\,l}K_{a,\,l} & I_{j\times j} - TG_{d,\,l}K_{a,\,l}\end{bmatrix}\begin{bmatrix}\underline{\tilde{f}}_{d}(z)\\ \underline{\dot{\tilde{f}}}_{d}(z)\end{bmatrix} + z\begin{bmatrix}G_{d,\,l}K_{v,\,l}\\ G_{d,\,l}K_{a,\,l}\end{bmatrix}\underline{f}_{d}(z)\\
\text{or, } \begin{bmatrix}(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} & -T(I_{j\times j} - G_{d,\,l}K_{v,\,l})\\ G_{d,\,l}K_{a,\,l} & (z-1)I_{j\times j} + TG_{d,\,l}K_{a,\,l}\end{bmatrix}&\begin{bmatrix}\underline{\tilde{f}}_{d}(z)\\ \underline{\dot{\tilde{f}}}_{d}(z)\end{bmatrix} = z\begin{bmatrix}G_{d,\,l}K_{v,\,l}\\ G_{d,\,l}K_{a,\,l}\end{bmatrix}\underline{f}_{d}(z)\\
\text{or, } \begin{bmatrix}\underline{\tilde{f}}_{d}(z)\\ \underline{\dot{\tilde{f}}}_{d}(z)\end{bmatrix} &= \begin{bmatrix}(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} & -T(I_{j\times j} - G_{d,\,l}K_{v,\,l})\\ G_{d,\,l}K_{a,\,l} & (z-1)I_{j\times j} + TG_{d,\,l}K_{a,\,l}\end{bmatrix}^{-1} z\begin{bmatrix}G_{d,\,l}K_{v,\,l}\\ G_{d,\,l}K_{a,\,l}\end{bmatrix}\underline{f}_{d}(z)
\end{aligned} \tag{4.47}$$

Using the following matrix inverse formula [77], the z-domain transfer function formulation of the VFLL is obtained.

$$\begin{bmatrix} A & D\\ C & B\end{bmatrix}^{-1} = \begin{bmatrix}[A - DB^{-1}C]^{-1} & -E\Delta^{-1}\\ -\Delta^{-1}F & \Delta^{-1}\end{bmatrix} \tag{4.48}$$

if A^{-1} and B^{-1} exist, where $\Delta = B - CA^{-1}D$, $E = A^{-1}D$ and $F = CA^{-1}$. Thus, VFLL transfer function matrix in the z-domain becomes

$$\underline{\tilde{f}}_{d}(z) = [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} + T(I - G_{d,\,l}K_{v,\,l})((z-1)I_{j\times j}+$$

$$\begin{aligned}
&TG_{d,\,l}K_{a,\,l})^{-1}G_{d,\,l}K_{a,\,l}]^{-1}zG_{d,\,l}K_{v,\,l}\underline{f}_d(z)+\\
&[T((z-1)I_{j\times j}+G_{d,\,l}K_{v,\,l})^{-1}(I_{j\times j}-G_{d,\,l}K_{v,\,l})((z-1)I_{j\times j}+TG_{d,\,l}K_{a,\,l}+\\
&TG_{d,\,l}K_{a,\,l}((z-1)I_{j\times j}+G_{d,\,l}K_{v,\,l})^{-1}(I_{j\times j}-G_{d,\,l}K_{v,\,l}))^{-1}]zG_{d,\,l}K_{a,\,l}\underline{f}_d(z)\\
\text{or,}\quad &\tilde{\underline{f}}_d(z)=[H^{(1)}_{VFLL_{discr},\,l}+H^{(2)}_{VFLL_{discr},\,l}]_{j\times j}\underline{f}_d(z)\\
\text{or,}\quad &\tilde{\underline{f}}_d(z)=[H_{VFLL_{discr},\,l}]_{j\times j}\underline{f}_d(z) \qquad (4.49)
\end{aligned}$$

It is apparent that the above formulation would be complex to deal with when calculating noise bandwidths numerically. As an alternative, the s-domain formulation can be used for numerically calculating the noise bandwidths. It will be shown later that the noise bandwidths calculated from s and z-domain transfer function matrices converge to each other for very small T. For a medium value of T (~ 0.1 sec) the z-domain bandwidths can be approximately found by scaling up the s-domain bandwidths by an empirical factor.

It is imperative to note that the primary objective of this chapter is to illuminate the internal workings of the vector architecture and to validate the findings of the theoretical analysis discussed later in this chapter with simulation studies. While this can be accomplished with the above complex VFLL model, it is possible to remove some complexities for simulation purposes and yet to validate the theoretical analysis. To this end, simulation studies are performed with a slowly changing user geometry with pseudolites. It will also be evident later that adopting a simplified model allows one to gain insights into some features of the vector architecture which would not have been easy to do with the above complex model. Transfer function model for the simulated user-pseudolite geometry is derived below from Equations 4.41 and 4.46. Details of this geometry will be discussed during simulation studies.

VFLL Transfer Function Model for Slowly Changing Geometry

The following aspects of the simulation geometry need to be taken into account for developing a transfer function representation. First, the signals are transmitted by ground-based pseudolites that are stationary. This is a reasonable approximation for the following reasons. While not strictly true, satellite motion alone over a short time interval will not affect the LOS appreciably. Moreover, the contributions of satellite motion to the Doppler frequency and its rate of change can be calculated using the satellite ephemeris known from the Navigation message and removed. In addition, the $\underline{e}_{d,\,k+1}$ term in Equations 4.35 and 4.37, being the difference between the received and predicted Doppler frequencies, is not affected by the Doppler component due to satellite motion. This is true, however, provided the Doppler due to satellite motion is predicted correctly.

Second, the user velocity is constant and the rate of change of the LOS vectors due to user motion is insignificant. Thus, the user acceleration and the rate of change of the Doppler frequency are not modeled with this geometry and the $\frac{G_{d,\,l}K_{a_{cont},\,l}}{s}$ term of Equation 4.46 reduces to zero. With this the transfer function model for this geometry becomes

$$\begin{aligned}\underline{\tilde{f}}_{d}(s) &= [sI_{j\times j} + G_{d,\,l}K_{v_{cont},\,l}]^{-1}G_{d,\,l}K_{v_{cont},\,l}\underline{f}_{d}(s)\\ \underline{\tilde{f}}_{d}(s) &= [H'_{VFLL_{cont},\,l}]_{j\times j}\underline{f}_{d}(s)\end{aligned} \tag{4.50}$$

where ($\prime$) in the superscript denotes the transfer function matrix for the simulation geometry.

In addition to the s-domain transfer function model, the z-domain formulation for the slowly changing geometry is provided below. Since $\dot{\underline{\tilde{f}}}_{d}$ is zero, Equation 4.41 becomes

$$\underline{\tilde{f}}_{d,\,k+1} = \underline{\tilde{f}}_{d,\,k} + G_{d,\,l}K_{v,\,l}(\underline{f}_{d,\,k+1} - \underline{\tilde{f}}_{d,\,k}) \tag{4.51}$$

Taking the Z transform, the above equation reduces to

$$z\underline{\tilde{f}}_{d}(z) = \underline{\tilde{f}}_{d}(z) + G_{d,\,l}K_{v,\,l}(z\underline{f}_{d}(z) - \underline{\tilde{f}}_{d}(z)) \tag{4.52}$$

After rearranging terms, the z-domain transfer function matrix for sub-interval l is given by

$$\begin{aligned}\underline{\tilde{f}}_{d}(z) &= [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l}]^{-1}zG_{d,\,l}K_{v,\,l}\underline{f}_{d}(z)\\ \underline{\tilde{f}}_{d}(z) &= [H'_{VFLL_{discr},\,l}]_{j\times j}\underline{f}_{d}(z)\end{aligned} \tag{4.53}$$

Equations 4.50 and 4.53, in addition to facilitating numerical calculation of vector loop noise bandwidth, allows the derivation of a theoretical expression of the noise bandwidth. Note that the transfer function model derived here represents a *sub-optimal* vector architecture which is used as an *analysis tool* during simulation studies. It is also verified during simulation studies that the sub-optimal architecture closely follows the original optimal implementation with continuously changing geometry matrix and Kalman gains. While the above model is developed for stationary pseudolites, it can be valid for the GPS satellites. This requires that the satellite contributions to the Doppler frequencies as calculated using the geometry matrix and satellite ephemeris are removed. The transfer function model is then developed only for the remaining Doppler component which results from user motion with a constant velocity.

4.2.2.2 VDLL Transfer Function Model

The VDLL transfer function model is derived for sub-interval l which spans time instants $[k+1,\ k+2,\ \ldots,\ k+n]$. In order to derive the transfer function model, it is important to take a close look at the implementation of Equation 4.31 in the vector architecture. For this purpose, Equation 4.31 is first rewritten below

$$\underline{\tilde{f}}^{+}_{\tau,\,code_{k+1}} = \frac{\Delta\underline{\widetilde{cp}}_{k+1}}{\Delta t} = \frac{1}{\Delta t}\tilde{G}_{\tau,\,k+1}K_{p,\,k+1}\underline{e}_{k+1} \tag{4.54}$$

For the decoupled VDLL (assumption 4 on page 74) $\underline{e}_{k+1}$ comprises only the DLL discriminator outputs i.e., $\underline{e}_{k+1} = \underline{\tau}_{k+1} = \underline{cp}_{k+1} - \underline{\widetilde{cp}}_{k+1}$, where $\underline{cp}_{k+1}$ is the true code phase vector at t_{k+1} and $\underline{\widetilde{cp}}_{k+1}$ is the receiver-predicted code phase vector at t_{k+1}. Substituting $\underline{e}_{k+1}$ into the above equation, replacing $K_{p,\,k+1}$ with $K_{p,\,l}$ and using $\tilde{G}_{\tau,\,k+i} = G_{\tau,\,l}$, where $1 \leq i \leq n$, one can write

$$\Delta\underline{\widetilde{cp}}_{k+1} = G_{\tau,\,l}K_{p,\,l}(\underline{cp}_{k+1} - \underline{\widetilde{cp}}_{k+1}) \tag{4.55}$$

It should be noted that $K_{p,\,l}$ contains only those elements of the previous $K_{p,\,k+1}$ (see the definition below Equation 4.28) that correspond to the code phase error. The solid green line in Figure 4.2 depicts how code phase corrections are applied for any satellite i in the vector architecture. The navigation filter generates the code phase correction after every measurement update (i.e., at t_k, t_{k+1} and so on). However, since only a frequency correction can be applied to the NCO, the code phase correction is divided by Δt to generate a frequency correction. This frequency correction steers the code phase to the desired value after Δt sec. This is depicted in the figure by the slanted portions of the green line. So, the estimated code phase vector at t_{k+1} ($\underline{\widetilde{cp}}_{k+1} + \Delta\underline{\widetilde{cp}}_{k+1}$) is obtained Δt sec later i.e., at (t_{k+1} + Δt). After this, no more code phase correction is applied until the next measurement update. This is because, as explained above Equation 4.31, the receiver-generated and the navigation filter-predicted pseudorange measurements equal each other after time updates. As a result, $\underline{\tilde{f}}^{-}_{\tau,\,code_{k+1}}$ is zero, where superscript (-) denotes corrections after time updates. This is indicated by the flat segments of the green line. The inclined light brown line shows the ramp change in code phase from $\underline{\widetilde{cp}}_{k}$ as determined by the code NCO frequency. Thus, the receiver-predicted code phase vector at t_{k+2} is given by

$$\begin{aligned}
\underline{\widetilde{cp}}_{k+2} &= [\underline{\widetilde{cp}}_{k+1} + \Delta\underline{\widetilde{cp}}_{k+1}] + \Delta\underline{\widetilde{cp}}_{dopp_{k+2}} \\
&= [\underline{\widetilde{cp}}'_{k+1} + \Delta\underline{\widetilde{cp}}_{dopp_{k+1}} + \Delta\underline{\widetilde{cp}}_{k+1}] + \Delta\underline{\widetilde{cp}}_{dopp_{k+2}}
\end{aligned}$$

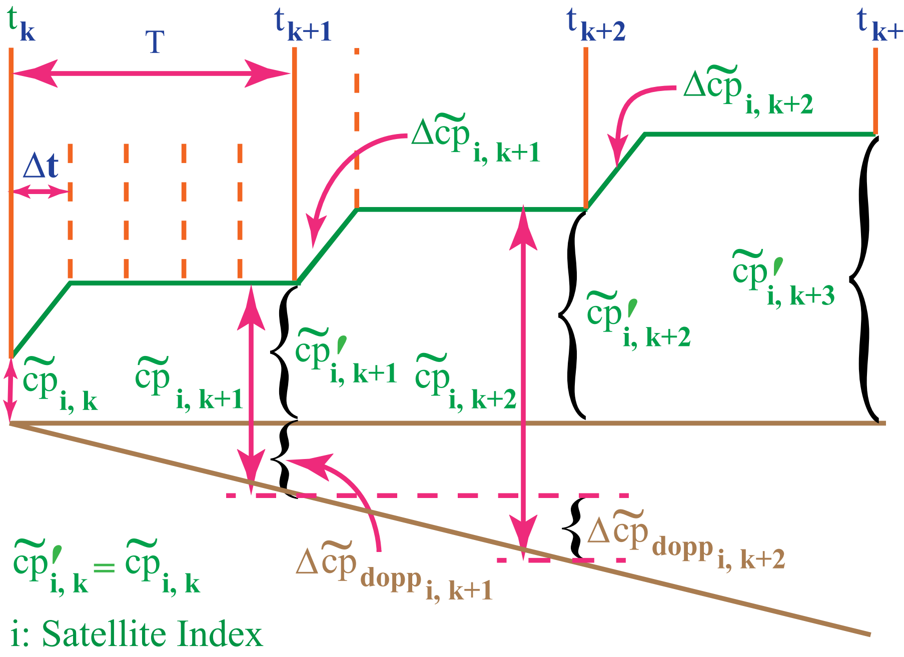


Figure 4.2: Implementation of VDLL Code Phase Corrections

$$\begin{aligned} &= [\underline{\tilde{cp}}'_{k+1} + \Delta\underline{\tilde{cp}}_{k+1}] + \Delta\underline{\tilde{cp}}_{dopp_{k+1}} + \Delta\underline{\tilde{cp}}_{dopp_{k+2}} \\ &= \underline{\tilde{cp}}'_{k+2} + \Delta\underline{\tilde{cp}}_{dopp_{k+1}} + \Delta\underline{\tilde{cp}}_{dopp_{k+2}} \end{aligned} \tag{4.56}$$

where $\underline{\tilde{cp}}'_{k+2} = (\underline{\tilde{cp}}'_{k+1} + \Delta\underline{\tilde{cp}}_{k+1})$, $\underline{\tilde{cp}}'_{k+1} = (\underline{\tilde{cp}}'_{k} + \Delta\underline{\tilde{cp}}_{k})$ and $\underline{\tilde{cp}}'_{k} = \underline{\tilde{cp}}_{k}$. $\Delta\underline{\tilde{cp}}_{dopp_{k+1}}$ and $\Delta\underline{\tilde{cp}}_{dopp_{k+2}}$ denote increments in the receiver-predicted code phase vector from t_k to t_{k+1} and from t_{k+1} to t_{k+2}, respectively, depending on the code NCO frequencies. It should be noted that the code NCO frequencies are updated at every Δt sec based on the VFLL-predicted Doppler frequencies.

The incoming code phase vector at t_{k+2} is given as

$$\begin{aligned} \underline{cp}_{k+2} &= \underline{cp}_{k} + [\Delta\underline{cp}_{dopp_{k+1}} + \Delta\underline{cp}_{dopp_{k+2}}] \\ &= \underline{cp}_{k} + [\Delta\underline{cp}_{dopp_{k+1}} + \Delta\underline{cp}_{dopp_{k+2}}] - [\Delta\underline{\tilde{cp}}_{dopp_{k+1}} + \Delta\underline{\tilde{cp}}_{dopp_{k+2}}] \\ &\quad + [\Delta\underline{\tilde{cp}}_{dopp_{k+1}} + \Delta\underline{\tilde{cp}}_{dopp_{k+2}}] \end{aligned} \tag{4.57}$$

It is important to note that the Delta terms without tilde in the above equation denote increments in the received code phase due to the code Doppler frequency. Hence, the second and third terms together represent the accumulated code phase error due to the error in the predicted Doppler frequency. Next, Replacing the first three terms with $\underline{cp}'_{k+2}$, the above equation reduces to

$$\underline{cp}_{k+2} = \underline{cp}'_{k+2} + \Delta\underline{\tilde{cp}}_{dopp_{k+1}} + \Delta\underline{\tilde{cp}}_{dopp_{k+2}} \tag{4.58}$$

Now, substituting Equations 4.56 and 4.58 into Equation 4.55 for time step t_{k+2}, one gets

$$\Delta\underline{\widetilde{cp}}_{k+2} = G_{\tau,\,l}K_{p,\,l}(\underline{cp}'_{k+2} - \underline{\widetilde{cp}}'_{k+2}) \tag{4.59}$$

Replacing $\Delta\underline{\widetilde{cp}}_{k+2}$ with $\underline{\widetilde{cp}}'_{k+3} - \underline{\widetilde{cp}}'_{k+2}$ (see Figure 4.2), the following equation is obtained

$$\underline{\widetilde{cp}}'_{k+3} - \underline{\widetilde{cp}}'_{k+2} = G_{\tau,\,l}K_{p,\,l}(\underline{cp}'_{k+2} - \underline{\widetilde{cp}}'_{k+2}) \tag{4.60}$$

Similar to the VFLL the continuous Kalman gain for sub-interval l is given as $K_{p_{cont},\,l} = K_{p,\,l}/(T)$ as $T \to 0$. Now, allowing T to go to zero, Equation 4.60 can be written as

$$\begin{aligned}\lim_{T\to 0}\frac{\underline{\widetilde{cp}}'_{k+3} - \underline{\widetilde{cp}}'_{k+2}}{T} &= \lim_{T\to 0} G_{\tau,\,l}\frac{K_{p,\,l}}{T}(\underline{cp}'_{k+2} - \underline{\widetilde{cp}}'_{k+2})\\ \dot{\underline{\widetilde{cp}}}' &= G_{\tau,\,l}K_{p_{cont},\,l}(\underline{cp}' - \underline{\widetilde{cp}}')\end{aligned} \tag{4.61}$$

Taking the Laplace transform and following the same steps as in the VFLL, the following VDLL transfer function matrix results

$$\begin{aligned}\underline{\widetilde{cp}}'(s) &= \left(I_{j\times j} + \frac{G_{\tau,\,l}K_{p_{cont},\,l}}{s}\right)^{-1}\frac{G_{\tau,\,l}K_{p_{cont},\,l}}{s}\underline{cp}'(s)\\ \underline{\widetilde{cp}}'(s) &= [H_{VDLL_{cont},\,l}]_{j\times j}\underline{cp}'(s)\end{aligned} \tag{4.62}$$

Thus, Equation 4.62 represents the VDLL transfer function model for the l^{th} time interval. It is sub-optimal in nature as it replaces the geometry matrix and Kalman gains with piecewise constant matrices.

Next, the z-domain transfer function is developed from Equation 4.60. Taking the Z transform of Equation 4.60, the following transfer function model is obtained

$$\begin{aligned} z\underline{\widetilde{cp}}'(z) - \underline{\widetilde{cp}}'(z) &= G_{\tau,\,l}K_{p,\,l}(\underline{cp}'(z) - \underline{\widetilde{cp}}'(z))\\ \text{or,}\quad [(z-1)I_{j\times j} + G_{\tau,\,l}K_{p,\,l}]\underline{\widetilde{cp}}'(z) &= G_{\tau,\,l}K_{p,\,l}\underline{cp}'(z)\\ \text{or,}\quad \underline{\widetilde{cp}}'(z) &= [(z-1)I_{j\times j} + G_{\tau,\,l}K_{p,\,l}]^{-1}G_{\tau,\,l}K_{p,\,l}\underline{cp}'(z)\\ \text{or,}\quad \underline{\widetilde{cp}}'(z) &= [H_{VDLL_{discr},\,l}]_{j\times j}\underline{cp}'(z)\end{aligned} \tag{4.63}$$

It should be noted that the same transfer function models hold for the slowly changing geometry used for simulation studies. Equations 4.53 and 4.63 are used to numerically and theoretically calculate the vector loop noise bandwidths next.

4.2.2.3 Vector Loop Noise Bandwidths

Two definitions of vector loop noise bandwidths are proposed in this thesis. One is for vector loop performance analysis and the other is for designing the scalar loops and comparing them with vector loops. The basic difference between them is as follows: The first definition considers the noise contributions of all channels to a specific channel i.e., the output channel is fixed. The second definition takes into account the noise contributions of a specific channel to all channels i.e., the input channel is fixed. However, both definitions become equivalent if the transfer function matrix is a symmetric matrix. This is possible only if all channels have equal noise variances. Both definitions are provided below.

Vector Loop Noise Bandwidth for Performance Analysis

For performance analysis of the vector architecture, the one-sided vector loop noise bandwidth of satellite i in the continuous domain for sub-interval l is defined as follows:

$$
\begin{aligned}
B_{Lm_{cont},\,l}(i) &= \frac{1}{\sum_{n=1}^{j} N_{0_m,\,nn}} \times \frac{1}{2\pi}\int_0^{\infty} [H_{m_{cont},\,l}(j\omega)]_i (N_{0_m}) [H_{m_{cont},\,l}(-j\omega)]_i^T d\omega \\
\text{or,}\quad B_{Lm_{cont},\,l}(i) &= \frac{1}{\sum_{n=1}^{j} N_{0_m,\,nn}} \times \frac{1}{2\pi}\sum_{n}^{j} N_{0_m,\,nn}\int_0^{\infty} |H^{(i,\,n)}_{m_{cont},\,l}(j\omega)|^2 d\omega \\
&= \frac{1}{\sum_{n=1}^{j} N_{0_m,\,nn}} \times \sum_{n}^{j} N_{0_m,\,nn} Bw^{(i,\,n)}_{cont} \qquad (4.64)
\end{aligned}
$$

where

m = VFLL or VDLL

$H_{m_{cont},\,l}(jw)$ = VFLL or VDLL transfer function matrix in the continuous domain for sub-interval l

$$
= \begin{bmatrix}
H^{(1,\,1)}_{m_{cont},\,l}(jw) & \dots & H^{(1,\,n)}_{m_{cont},\,l}(jw) & \dots & H^{(1,\,j)}_{m_{cont},\,l}(jw) \\
\vdots & & \ddots & & \vdots \\
H^{(i,\,1)}_{m_{cont},\,l}(jw) & \dots & H^{(i,\,n)}_{m_{cont},\,l}(jw) & \dots & H^{(i,\,j)}_{m_{cont},\,l}(jw) \\
\vdots & & \ddots & & \vdots \\
H^{(j,\,1)}_{m_{cont},\,l}(jw) & \dots & H^{(j,\,n)}_{m_{cont},\,l}(jw) & \dots & H^{(j,\,j)}_{m_{cont},\,l}(jw)
\end{bmatrix}
$$

j = number of visible satellites; note that j in the argument of $H(.)$ denotes the complex operator

$[H_{m_{cont},\,l}(jw)]_i$ = i^{th} row of $H_{m_{cont},\,l}(jw) = [H^{(i,\,1)}_{m_{cont},\,l}(jw) \dots H^{(i,\,j)}_{m_{cont},\,l}(jw)]$

N_{0_m} = PSD matrix for m = VFLL or VDLL; it is a diagonal matrix of dimension $j \times j$

$N_{0_m,\,nn}$ $= n^{\text{th}}$ diagonal element of N_{0_m}

N_{0_m} $= W_{m_d}T$, where W_{m_d} is the navigation filter measurement noise covariance and T is the measurement update interval

A close observation of the above equation reveals that the second part of the equation is equal to the output noise power of channel i in vector tracking mode. It is divided by the sum of the input noise PSDs of all channels to obtain the noise bandwidth of that channel. It is also evident that the noise bandwidth of one channel in vector tracking mode depends on all other channels and their noise levels (unless all noise variances are the same). A graphical representation of the above equation is provided in Figure 4.3. It shows that the above-defined noise bandwidth is the bandwidth of a boxcar filter whose input noise PSD is the sum of the PSD of all noise inputs to the vector loops and output noise power is the same as that of channel i in vector tracking mode.

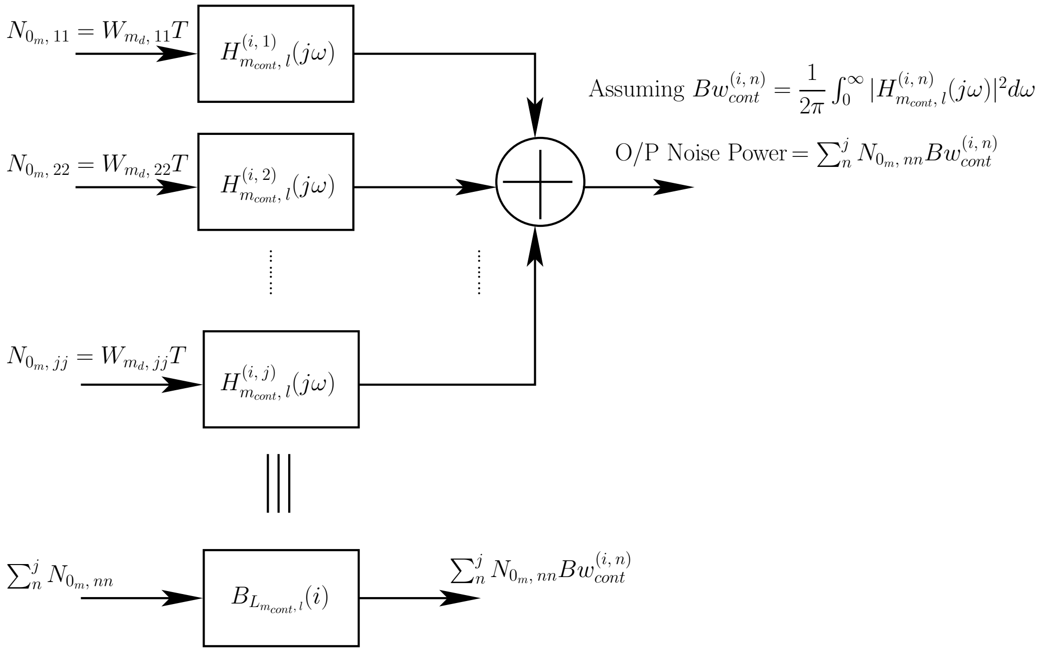


Figure 4.3: Graphical Representation of Vector Loop Noise Bandwidth Used for its Performance Analysis

Similarly, the vector loop noise bandwidth in the discrete domain is given as

$$\begin{aligned} B_{Lm_{discr},\,l}(i) &= \frac{1}{\sum_{n=1}^{j} W_{m_d,\,nn}T} \times \frac{1}{2\pi}\int_0^{\pi/T} [H_{m_{discr},\,l}(e^{j\omega T})]_i (W_{m_d}T)[H_{m_{discr},\,l}(e^{-j\omega T})]_i^T d\omega \\ &= \frac{1}{\sum_{n=1}^{j} W_{m_d,\,nn}T} \times \sum_{n}^{j} (W_{m_d,\,nn}T) Bw_{discr}^{(i,\,n)} \end{aligned} \tag{4.65}$$

It should be noted that the subscript "*cont*" in Equation 4.64 is replaced with "*discr*" in the above equation. W_{m_d} represents the discrete-time measurement noise covariance. The above definition is used to numerically and theoretically calculate the vector loop noise bandwidth for performance analysis.

The reason for choosing the above definition of noise bandwidth for vector loop performance analysis is that it reflects the amount of noise filtered through a given channel in vector tracking mode for a given set of input noise PSDs. However, as will be shown later in this chapter, a channel that has a high input noise PSD may not have a small noise bandwidth. Therefore, this definition of noise bandwidth cannot be used for designing the scalar loops because it may assign a higher bandwidth to a channel with higher noise PSD. It cannot also be used as the noise bandwidth of a scalar tracking loop when comparing it with its vector counterpart. This is because this bandwidth is calculated considering the noise contributions of all channels to a particular channel. To properly select the noise bandwidth of a scalar tracking loop for comparison purposes, one needs to find the noise bandwidth that the vector loops assign to the input noise of that channel alone. In view of this, the following definition of vector loop noise bandwidth is adopted for comparing scalar loops with vector loops and for designing scalar loop noise bandwidths for better performance.

Vector Loop Noise Bandwidth for Comparison and Design of Scalar Loops

The vector loop noise bandwidth used for selecting the appropriate scalar loop noise bandwidth is defined as follows:

$$\begin{aligned} B'_{Lm_{cont},\,l}(i) &= \frac{1}{j} \times \frac{1}{2\pi}\sum_{n}^{j}\int_0^{\infty} |H_{m_{cont},\,l}^{(n,\,i)}(j\omega)|^2 d\omega \\ &= \frac{1}{j}\sum_{n}^{j} Bw_{cont}^{(n,\,i)} \end{aligned} \tag{4.66}$$

A pictorial representation of the above definition is given in Figure 4.4. Let $H_{m_{cont},\,l}^{(n,\,i)}$ be the $(n,\ i)^{\text{th}}$ element of the transfer function matrix $H_{m_{cont},\,l}(j\omega)$, where

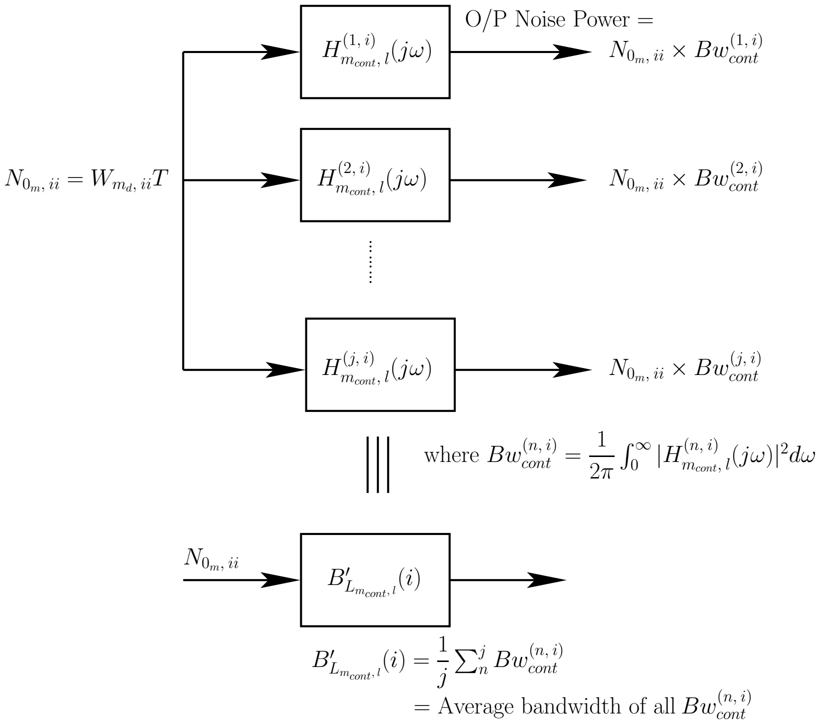


Figure 4.4: Graphical Representation of Vector Loop Noise Bandwidth Used for Comparison and Design of Scalar Tracking Loops

n is the row index and i is the column index. $H^{(n,\,i)}_{m_{cont},\,l}$ is multiplied with the input to channel i to determine a fraction of the output of channel n. $Bw^{(n,\,i)}_{cont} = 1/(2\pi)\int_0^\infty |H^{(n,\,i)}_{m_{cont},\,l}(j\omega)|^2 d\omega$ is, therefore, the noise bandwidth of $H^{(n,\,i)}_{m_{cont},\,l}$ which determines how much of the noise in channel i will filter through channel n and contribute to its output noise power. Likewise, $Bw^{(k,\,i)}_{cont}$ determines the contribution of the noise in channel i to determining the output noise power of channel k. Therefore, the average of all $Bw^{(n,\,i)}_{cont}$, where $n = 1, ..., j$, yields the average noise bandwidth of the vector loops with respect to the noise in channel i. It will be verified later with simulations that this noise bandwidth varies inversely with the input noise variance. It is proposed in this thesis that the corresponding scalar loop noise bandwidth is set to this value for comparison with vector loops. The scalar loop noise bandwidth can also be varied in the vicinity of this value to obtain best possible performance. Thus, the above vector loop noise bandwidth not only provides a framework for systematic performance comparison of both tracking loops, but can be used as a good candidate for designing the scalar loop noise bandwidth.

The discrete equivalent of Equation 4.66 is

$$\begin{aligned} B'_{Lm_{discr},\,l}(i) &= \frac{1}{j} \times \frac{1}{2\pi} \sum_{n}^{j} \int_{0}^{\pi/T} |H^{(n,\,i)}_{m_{discr},\,l}(e^{j\omega T})|^2 d\omega \\ &= \frac{1}{j} \sum_{n}^{j} Bw^{(n,\,i)}_{discr} \end{aligned} \tag{4.67}$$

It is important to note that the two definitions of vector loop noise bandwidths are identical if all channels are assumed to have the same noise variance. This is because the first definition is the weighted average of the noise bandwidths of individual transfer function elements in a row of the transfer function matrix. On the other hand, the second definition takes the average of the noise bandwidths of all transfer function elements in a column of the transfer function matrix. If the noise variance is identical in all channels, it can be easily shown that the transfer function matrix reduces to a symmetric matrix. This results in the same noise bandwidth from both definitions.

Numerical Computation of Noise Bandwidth

For simulation purposes the vector loop noise bandwidths are numerically calculated. This is done by substituting vector loop transfer functions (Equations 4.53 and 4.63) into Equations 4.65 and 4.67; calculating the integrand; and then integrating it numerically from 0 to π/T. As the transfer function model includes a matrix inverse operation, it is reduced to the following form to simplify the calculation of the matrix inverse. The simplified form is derived for the discrete transfer function models. It also holds for the continuous-domain transfer function models.

$$H_{VDLL_{discr},\,l} = ((z-1)I_{j\times j} + G_{\tau,\,l}K_{p,\,l})^{-1}G_{\tau,\,l}K_{p,\,l} \tag{4.68}$$

$$\begin{aligned} &= \frac{1}{(z-1)}\left(I_{j\times j} - \frac{G_{\tau,\,l}K_{p,\,l}}{(z-1)} + \frac{(G_{\tau,\,l}K_{p,\,l})^2}{(z-1)^2} - \dots\right)G_{\tau,\,l}K_{p,\,l} \\ &= \frac{1}{(z-1)}\left(G_{\tau,\,l}K_{p,\,l} - \frac{(G_{\tau,\,l}K_{p,\,l})^2}{(z-1)} + \frac{(G_{\tau,\,l}K_{p,\,l})^3}{(z-1)^2} - \dots\right) \\ &= \frac{1}{(z-1)}G_{\tau,\,l}\left(K_{p,\,l} - \frac{(K_{p,\,l}G_{\tau,\,l})K_{p,\,l}}{(z-1)} + \frac{(K_{p,\,l}G_{\tau,\,l})^2K_{p,\,l}}{(z-1)^2} - \dots\right) \\ &= \frac{1}{(z-1)}G_{\tau,\,l}\left(I_{4\times 4} - \frac{K_{p,\,l}G_{\tau,\,l}}{(z-1)} + \frac{(K_{p,\,l}G_{\tau,\,l})^2}{(z-1)^2} - \dots\right)K_{p,\,l} \\ &= \frac{1}{(z-1)}G_{\tau,\,l}\left(I_{4\times 4} + \frac{K_{p,\,l}G_{\tau,\,l}}{(z-1)}\right)^{-1}K_{p,\,l} \\ &= G_{\tau,\,l}\left((z-1)I_{4\times 4} + K_{p,\,l}G_{\tau,\,l}\right)^{-1}K_{p,\,l} \end{aligned} \tag{4.69}$$

In the above equations j (> 4) represents the number of visible satellites. The dimension of $G_{\tau,\,l}$ is $j \times 4$ and that of $K_{p,\,l}$ is $4 \times j$. Four rows of the Kalman gain matrix correspond to the x, y and z components of user position and clock bias. So, the dimension of $G_{\tau,\,l}K_{p,\,l}$ is $j \times j$ which means the original transfer function model (Equation 4.68) requires the inverse of a $j \times j$ matrix and it increases with the number of visible satellites. On the other hand, the dimension of $K_{p,\,l}G_{\tau,\,l}$ is 4×4 and it remains constant for any number of visible satellites. Thus, the computational load involved is greatly reduced when using Equation 4.69 instead of Equation 4.68 for calculating the noise bandwidth.

Using the same steps as above, Equation 4.53 for the VFLL reduces to

$$\begin{aligned} H'_{VFLL_{discr},\,l} &= [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l}]^{-1} zG_{d,\,l}K_{v,\,l} \\ &= G_{d,\,l}\left((z-1)I_{4\times 4} + K_{v,\,l}G_{d,\,l}\right)^{-1} zK_{v,\,l} \end{aligned} \tag{4.70}$$

Note that the dimensions of $G_{d,\,l}$ and $K_{v,\,l}$ are $j \times 4$ and $4 \times j$, respectively, where j is the number of visible satellites. Equations 4.69 and 4.70 are used to numerically calculate the vector loop noise bandwidths for the simulation geometry.

Next, a simplified form of Equation 4.49 which is the general discrete transfer function model of VFLL is derived below, although it is not used for bandwidth calculation in this thesis. The first part of the equation is simplified as

$$\begin{aligned} H^{(1)}_{VFLL_{discr},\,l} = [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} + T(I - G_{d,\,l}K_{v,\,l})((z-1)I_{j\times j} + \\ TG_{d,\,l}K_{a,\,l})^{-1}G_{d,\,l}K_{a,\,l}]^{-1} zG_{d,\,l}K_{v,\,l} \end{aligned} \tag{4.71}$$

where the dimension of $K_{a,\,l}$ is $4 \times j$. Following the same steps as in Equation 4.69, the third term within square brackets reduces to

$$\begin{aligned} H^{(1)}_{VFLL_{discr},\,l} = [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} + T(I - G_{d,\,l}K_{v,\,l})G_{d,\,l}((z-1)I_{4\times 4} + \\ TK_{a,\,l}G_{d,\,l})^{-1}K_{a,\,l}]^{-1} zG_{d,\,l}K_{v,\,l} \end{aligned} \tag{4.72}$$

Next, with some algebra the following simplified form is obtained:

$$\begin{aligned} H^{(1)}_{VFLL_{discr},\,l} &= [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} + T(I - G_{d,\,l}K_{v,\,l})G_{d,\,l}((z-1)I_{4\times 4} + \\ &TK_{a,\,l}G_{d,\,l})^{-1}K_{a,\,l}]^{-1} zG_{d,\,l}K_{v,\,l} \\ &= [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} + T(G_{d,\,l} - G_{d,\,l}K_{v,\,l}G_{d,\,l})((z-1)I_{4\times 4} + \\ &TK_{a,\,l}G_{d,\,l})^{-1}K_{a,\,l}]^{-1} zG_{d,\,l}K_{v,\,l} \\ &= [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} + TG_{d,\,l}(I_{4\times 4} - K_{v,\,l}G_{d,\,l})((z-1)I_{4\times 4} + \end{aligned}$$

$$TK_{a,\,l}G_{d,\,l})^{-1}K_{a,\,l}]^{-1}zG_{d,\,l}K_{v,\,l}$$

Replacing $T(I_{4\times4} - K_{v,\,l}G_{d,\,l})((z-1)I_{4\times4} + TK_{a,\,l}G_{d,\,l})^{-1}$ with $A_{4\times4}$, one gets

$$\begin{aligned} H^{(1)}_{VFLL_{discr},\,l} &= [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l} + G_{d,\,l}A_{4\times4}K_{a,\,l}]^{-1}zG_{d,\,l}K_{v,\,l} \\ &= [(z-1)I_{j\times j} + G_{d,\,l}(K_{v,\,l} + A_{4\times4}K_{a,\,l})]^{-1}zG_{d,\,l}K_{v,\,l} \end{aligned}$$

Substituting $B_{4\times j}$ for $(K_{v,\,l} + A_{4\times4}K_{a,\,l})$ yields

$$H^{(1)}_{VFLL_{discr},\,l} = \frac{1}{(z-1)}\left[I_{j\times j} + \frac{G_{d,\,l}B_{4\times j}}{(z-1)}\right]^{-1} zG_{d,\,l}K_{v,\,l}$$

Again, using the same steps as in Equation 4.69 gives

$$H^{(1)}_{VFLL_{discr},\,l} = \frac{1}{(z-1)}G_{d,\,l}\left[I_{4\times4} + \frac{BG_{d,\,l}}{(z-1)}\right]^{-1} zK_{v,\,l} \tag{4.73}$$

Next, substituting B and A into the above equation the final simiplifed form results, as shown below

$$\begin{aligned} H^{(1)}_{VFLL_{discr},\,l} = G_{d,\,l}[(z-1)I_{4\times4} + (K_{v,\,l}G_{d,\,l} + T(I_{4\times4} - K_{v,\,l}G_{d,\,l})((z-1)I_{4\times4} + \\ TK_{a,\,l}G_{d,\,l})^{-1}K_{a,\,l}G_{d,\,l})]^{-1}zK_{v,\,l} \end{aligned} \tag{4.74}$$

Using similar steps, the second part of Equation 4.49 reduces to

$$\begin{aligned} H^{(2)}_{VFLL_{discr},\,l} = TG_{d,\,l}[(z-1)I_{4\times4} + K_{v,\,l}G_{d,\,l}]^{-1}(I_{4\times4} - K_{v,\,l}G_{d,\,l})[(z-1)I_{4\times4} + \\ TK_{a,\,l}G_{d,\,l} + TK_{a,\,l}G_{d,\,l}[(z-1)I_{4\times4} + K_{v,\,l}G_{d,\,l}]^{-1}(I_{4\times4} - K_{v,\,l}G_{d,\,l})]^{-1}zK_{a,\,l} \end{aligned} \tag{4.75}$$

A close observation of the above equations indicates that they involve the inverse of 4×4 matrices instead of $j \times j$ matrices, unlike the original form.

Approximate Theoretical Expression of Vector Loop Noise Bandwidth for Performance Analysis

An approximate theoretical expression of the vector loop noise bandwidth for the transfer function model with constant-velocity user motion can be derived. This helps better understand the operating principles of vector loops. In order to do this, the following approach is adopted.

Theoretical Expression for Discrete-Domain Noise Bandwidth: To derive an approximate theoretical expression for the discrete-domain noise bandwidth, the Kalman gains of Equations 4.53 and 4.63 are first replaced with least squares estimation gains over time interval l as

$$K_{v,\,l} =(G_{d,\,l}^T G_{d,\,l})^{-1} G_{d,\,l}^T \tag{4.76}$$

$$K_{p,\,l} =(G_{\tau,\,l}^T G_{\tau,\,l})^{-1} G_{\tau,\,l}^T \tag{4.77}$$

Thus, $G_{d,\,l}K_{v,\,l}$ and $G_{\tau,\,l}K_{p,\,l}$ are projection operators on $\Re_{G_{d,\,l}}$ and $\Re_{G_{\tau,\,l}}$, respectively, where $\Re$ stands for the range or column space of a matrix. It should also be noted that they are idempotent matrices. Hence, mathematically,

$$G_{d,\,l}K_{v,\,l} = (G_{d,\,l}K_{v,\,l})^2 = \ldots = (G_{d,\,l}K_{v,\,l})^n \tag{4.78}$$

$$G_{\tau,\,l}K_{p,\,l} = (G_{\tau,\,l}K_{p,\,l})^2 = \ldots = (G_{\tau,\,l}K_{p,\,l})^n \tag{4.79}$$

Using Equation 4.78, the VFLL transfer function model of Equation 4.53 for constant velocity user motion reduces to

$$\begin{aligned}
[H'_{VFLL_{discr},\,l}]_{j\times j} =& [(z-1)I_{j\times j} + G_{d,\,l}K_{v,\,l}]^{-1} z G_{d,\,l}K_{v,\,l} \\
=& \left(I + \frac{G_{d,\,l}K_{v,\,l}}{(z-1)}\right)^{-1} \frac{z}{(z-1)} G_{d,\,l}K_{v,\,l} \\
[H'_{VFLL_{discr},\,l}]_{j\times j} =& \left(I - \frac{G_{d,\,l}K_{v,\,l}}{(z-1)} + \left(\frac{G_{d,\,l}K_{v,\,l}}{(z-1)}\right)^2 - \ldots\right) \frac{z}{(z-1)} G_{d,\,l}K_{v,\,l} \\
=& \left(\frac{G_{d,\,l}K_{v,\,l}}{(z-1)} - \left(\frac{G_{d,\,l}K_{v,\,l}}{(z-1)}\right)^2 + \left(\frac{G_{d,\,l}K_{v,\,l}}{(z-1)}\right)^3 - \ldots\right) z \\
& \text{Applying the idempotent property of } G_{d,\,l}K_{v,\,l} \\
=& \left(\frac{G_{d,\,l}K_{v,\,l}}{(z-1)} - \frac{G_{d,\,l}K_{v,\,l}}{(z-1)^2} + \frac{G_{d,\,l}K_{v,\,l}}{(z-1)^3} - \ldots\right) z \\
=& \left(I - \frac{I}{(z-1)} + \frac{I}{(z-1)^2} - \ldots\right) \frac{z}{(z-1)} G_{d,\,l}K_{v,\,l} \\
=& \left(I + \frac{I}{(z-1)}\right)^{-1} \frac{z}{(z-1)} G_{d,\,l}K_{v,\,l} \\
=& \frac{(z-1)}{z} I \frac{z}{(z-1)} G_{d,\,l}K_{v,\,l} \\
=& G_{d,\,l}K_{v,\,l}
\end{aligned} \tag{4.80}$$

Substituting the above mathematical expression of the VFLL transfer function model into Equation 4.65, using Equation 4.78 and assuming all channels have the same noise variance σ^2, the one-sided VFLL noise bandwidth for channel i over interval l is obtained as follows:

$$
\begin{aligned}
BW_{VFLL_{discr},\,l}(i) =& \frac{1}{\sum_{n=1}^{j} W_{VFLL_d,\,nn}T} \times \\
\frac{1}{2\pi}\int_0^{\pi/T} & [H'_{VFLL_{discr},\,l}(e^{j\omega T})]_i (W_{VFLL_d}T)[H'_{VFLL_{discr},\,l}(e^{-j\omega T})]_i^T d\omega \\
=& \frac{1}{2\pi \times j\sigma^2}\int_0^{\pi/T} [G_{d,\,l}K_{v,\,l}]_i \sigma^2 I [G_{d,\,l}K_{v,\,l}]_i^T d\omega \\
=& \frac{1}{2\pi \times j}[G_{d,\,l}K_{v,\,l}]_{ii}\int_0^{\pi/T} d\omega \\
=& \frac{1}{2T \times j}[G_{d,\,l}K_{v,\,l}]_{ii}
\end{aligned}
$$

where j is the number of visible satellites. $[G_{d,\,l}K_{v,\,l}]_i$ represents the i^{th} row of $G_{d,\,l}K_{v,\,l}$ and $[G_{d,\,l}K_{v,\,l}]_{ii}$ is the i^{th} diagonal element of $G_{d,\,l}K_{v,\,l}$. The idempotent property of Equation 4.78 results in $[G_{d,\,l}K_{v,\,l}]_{ii} = [G_{d,\,l}K_{v,\,l}]_i[G_{d,\,l}K_{v,\,l}]_i^T$. Thus, the VFLL bandwidths for all channels are given by

$$
[BW_{VFLL_{discr},\,l}]_{j\times 1} = \frac{\text{diag}(G_{d,\,l}K_{v,\,l})}{2T \times j} \tag{4.81}
$$

Likewise, for the VDLL

$$
\begin{aligned}
[H_{VDLL_{discr},\,l}]_{j\times j} =& [(z-1)I_{j\times j} + G_{\tau,\,l}K_{p,\,l}]^{-1}G_{\tau,\,l}K_{p,\,l} \\
=& \frac{G_{\tau,\,l}K_{p,\,l}}{z}
\end{aligned} \tag{4.82}
$$

$$
\begin{aligned}
[BW_{VDLL_{discr},\,l}]_{j\times 1} =& \frac{\text{diag}(G_{\tau,\,l}K_{p,\,l})}{2\pi \times j}\int_0^{\pi/T} \frac{1}{e^{j\omega T}}\frac{1}{e^{-j\omega T}} d\omega \\
=& \frac{\text{diag}(G_{\tau,\,l}K_{p,\,l})}{2T \times j}
\end{aligned} \tag{4.83}
$$

As the elements of the last column of $G_{d,\,l}$ (or $G_{\tau,\,l}$) are identical, it can be shown that the diagonal elements of $(G_{d,\,l}K_{v,\,l})$ or $(G_{\tau,\,l}K_{p,\,l})$ are inversely related to the number of satellites. A simple proof of this is given in Appendix B. The above property of $(G_{d,\,l}K_{v,\,l})$ or $(G_{\tau,\,l}K_{p,\,l})$ implies that the output noise power of each channel and the noise bandwidth of vector loops are narrowed down when more satellites are visible. In addition, it was noted earlier (see below Equations 4.4 and 4.28) that satellites aid each other in vector tracking mode. Thus, drawing a parallel with tight and ultra-tight integration [41], it can be concluded that narrower noise bandwidths along with aiding from more satelites would result in higher tracking thresholds of the vector loops as more satellites are visible. However, in practice, noise in individual satellite channels would place a limit to the tracking threshold improvements which was not considered in this analysis.

The above analysis assumes that all satellite channels have the same noise variance. This assumption can be relaxed by replacing the Kalman gains with weighted least squares gains. That is

$$\begin{aligned} K_{v,\,l} =& (G_{d,\,l}^T W_{VFLL_d}^{-1} G_{d,\,l})^{-1} G_{d,\,l}^T W_{VFLL_d}^{-1} \\ K_{p,\,l} =& (G_{\tau,\,l}^T W_{VDLL_d}^{-1} G_{\tau,\,l})^{-1} G_{\tau,\,l}^T W_{VDLL_d}^{-1} \end{aligned}$$

where W_{VFLL_d} is the discrete-time measurement noise covariance for the VFLL. Since the weighted least squares gains also satisfy the idempotent property, the same transfer function models as Equations 4.80 and 4.82 can be obtained with these gains. Then, following the same steps as before, the VFLL noise bandwidths can be derived as follows:

$$\begin{aligned} [BW_{VFLL_{discr},\,l}]_{j\times 1} =& \frac{\text{diag}(G_{d,\,l}(G_{d,\,l}^T W_{VFLL_d}^{-1} G_{d,\,l})^{-1} G_{d,\,l}^T)}{2T\sum_{i=1}^{j} W_{VFLL_{d},\,ii}} \\ =& \frac{\text{diag}(G_{d,\,l}(G_{d,\,l}^T W_{VFLL_d}^{-1} G_{d,\,l})^{-1} G_{d,\,l}^T W_{VFLL_d}^{-1} W_{VFLL_d})}{2T\sum_{i=1}^{j} W_{VFLL_{d},\,ii}} \end{aligned}$$

where $W_{VFLL_{d},\,ii}$ is the i^{th} diagonal element of W_{VFLL_d}. Substituting $K_{v,\,l}$ for $(G_{d,\,l}^T W_{VFLL_d}^{-1} G_{d,\,l})^{-1} G_{d,\,l}^T W_{VFLL_d}^{-1}$ gives

$$[BW_{VFLL_{discr},\,l}]_{j\times 1} = \frac{\text{diag}(G_{d,\,l} K_{v,\,l} W_{VFLL_d})}{2T\sum_{i=1}^{j} W_{VFLL_{d},\,ii}}$$

As W_{VFLL_d} is a diagonal matrix, the above equation can be written as

$$[BW_{VFLL_{discr},\,l}]_{j\times 1} = \frac{\text{diag}(W_{VFLL_d})}{2T\sum_{i=1}^{j} W_{VFLL_{d},\,ii}} * \text{diag}(G_{d,\,l} K_{v,\,l}) \tag{4.84}$$

where the operator $(*)$ denotes element-by-element multiplication of two vectors. Likewise, for the VDLL

$$[BW_{VDLL_{discr},\,l}]_{j\times 1} = \frac{\text{diag}(W_{VDLL_d})}{2T\sum_{i=1}^{j} W_{VDLL_{d},\,ii}} * \text{diag}(G_{\tau,\,l} K_{p,\,l}) \tag{4.85}$$

It is non-trivial to theoretically prove that the above noise bandwidths reduce with more number of visible satellites. An alternative approach to confirming this is through simulation. Preliminary simulations have shown that both the numerator of the above equation (i.e., output noise power) and the noise bandwidths reduce with more satellites in view.

Theoretical Expression for Continuous-Domain Noise Bandwidth: An approximate theoretical expression of the continuous-domain VFLL noise bandwidth is also derived below for constant-velocity user motion with pseudolites. For this purpose Equation 4.51 is rewritten as follows:

$$\underline{\tilde{f}}_{d,\,k+1} = \underline{\tilde{f}}_{d,\,k} + G_{d,\,l}K_{v,\,l}(\underline{f}_{d,\,k+1} - \underline{\tilde{f}}_{d,\,k})$$

Using the first order Taylor series expansion for small T

$$\underline{\tilde{f}}_{d,\,k} + T\underline{\dot{\tilde{f}}}_{d,\,k} \approx \underline{\tilde{f}}_{d,\,k} + G_{d,\,l}K_{v,\,l}(\underline{f}_{d,\,k} + T\underline{\dot{f}}_{d,\,k} - \underline{\tilde{f}}_{d,\,k})$$

Since the signals are assumed to be transmitted by stationary pseudolites; the user velocity is constant; and the rate of change of LOS due to user motion is negligible, $\underline{\dot{f}}_{d,\,k} \approx [0\ 0\ ...\ 0]^T$. Thus, the above equation becomes

$$T\underline{\dot{\tilde{f}}}_{d,\,k} \approx G_{d,\,l}K_{v,\,l}(\underline{f}_{d,\,k} - \underline{\tilde{f}}_{d,\,k})$$

Taking the Laplace transform, Equation 4.50 is obtained below, where the continuous Kalman gain $K_{v_{cont},\,l}$ is replaced with $K_{v,\,l}/T$

$$\begin{aligned} Ts\underline{\tilde{f}}_{d}(s) &\approx G_{d,\,l}K_{v,\,l}(\underline{f}_{d}(s) - \underline{\tilde{f}}_{d}(s)) \\ \underline{\tilde{f}}_{d}(s) &\approx \left(sI_{j\times j} + \frac{G_{d,\,l}K_{v,\,l}}{T}\right)^{-1}\frac{G_{d,\,l}K_{v,\,l}}{T}\underline{f}_{d}(s) \end{aligned} \tag{4.86}$$

Next, using the idempotent property of $G_{d,\,l}K_{v,\,l}$ as before, the above equation reduces to

$$\begin{aligned} \underline{\tilde{f}}_{d}(s) &\approx \left(I_{j\times j} + \frac{G_{d,\,l}K_{v,\,l}}{Ts}\right)^{-1}\frac{G_{d,\,l}K_{v,\,l}}{Ts}\underline{f}_{d}(s) \\ &\approx \left(I + \frac{I}{Ts}\right)^{-1}\frac{1}{Ts}G_{d,\,l}K_{v,\,l}\underline{f}_{d}(s) \\ &\approx \frac{G_{d,\,l}K_{v,\,l}}{1+Ts}\underline{f}_{d}(s) \end{aligned} \tag{4.87}$$

Thus, the VFLL noise bandwidths for identical noise variance across channels are

$$\begin{aligned} [BW_{VFLL_{cont},\,l}]_{j\times 1} &\approx \frac{\text{diag}(G_{d,\,l}K_{v,\,l})}{2\pi \times j}\int_0^\infty \frac{1}{1+(\omega T)^2}d\omega \\ &\approx \frac{\text{diag}(G_{d,\,l}K_{v,\,l})}{4T \times j} \end{aligned} \tag{4.88}$$

where j is the number of visible satellites. It is important to note that the above bandwidths are half of the discrete-domain noise bandwidths given by Equation 4.81. Similarly, the VDLL noise bandwidths can be derived by assuming $K_{p_{cont},\,l} \approx K_{p,\,l}/T$ and by using the idempotent property of $G_{\tau,\,l}K_{p,\,l}$. They are also half of their discrete-domain counterparts.

Discrete-Domain and Continuous-Domain Noise Bandwidths: Few Important Properties

Although the noise bandwidth calculated using the discrete transfer function model is the actual noise bandwidth, calculating the VFLL noise bandwidth using Equations 4.74 and 4.75 is complex as it involves the inverse of five 4×4 matrices in the z domain. As an alternative, this work proposes to calculate the VFLL noise bandwidth from its Laplace transfer function model and to multiply with it an empirical factor to approximately determine the actual discrete-domain noise bandwidth. In order to find the factor, it is important to investigate how the continuous-domain noise bandwidth differs from its discrete equivalent. To this end, the noise bandwidths calculated from the continuous and discrete transfer function models for the simulation geometry with pseudolites are compared below for different values of the navigation filter measurement update interval T. While the comparison results in a value for the above-mentioned multiplication factor, it is important to note the limitation of that. The transfer function model considered in this analysis is valid for constant-velocity user motion. Therefore, it is different from the transfer function given by the sum of Equations 4.74 and 4.75 which holds for user motion with a constant acceleration. So, the factor proposed below may not hold for Equations 4.74 and 4.75 and the actual factor can only be found by comparing the noise bandwidths from Equations 4.46 and those from the sum of Equations 4.74 and 4.75. Nevertheless, it is worth studying how the continuous-domain and discrete-domain noise bandwidths are related to each other and vary with T in the case of transfer functions models developed for constant-velocity user motion.

It should be noted that the continuous Kalman gains are required to calculate the noise bandwidth from the Laplace transfer function model. Since the continuous gains are unknown, they are approximated by dividing the discrete Kalman gains by T. The continuous-domain noise bandwidths obtained this way are approximate unless T is small enough compared to the channel noise variance $W_{VFLL_d,\,ii}$. i is the channel index and ii stands for the i^{th} diagonal element of the discrete-time measurement noise covariance W_{VFLL_d}.

Figure 4.5 depicts the ratio of the discrete-domain noise bandwidth to the approximate continuous-domain noise bandwidth as a function of T. The results of six different channels, each representing one pseudolite, are shown in six sub-figures. Each sub-figure is plotted with continuous-time noise PSD ($N_{0_{VFLL},\,i}$) as a running parameter. The subscript i represents the i^{th} channel. $N_{0_{VFLL},\,i}$ is assumed to be the same for all i. The steps for obtaining these results are as follows. First, the value of $W_{VFLL_d,\,ii}$

for $T = 0.1$ sec is multiplied by 0.1 to obtain $N_{0_{VFLL,\,i}}$. The reason behind selecting the value of $W_{VFLL_d,\,ii}$ at $T = 0.1$ sec is that 0.1 sec is considered the measurement update interval for simulation studies. Hence, an estimate of $W_{VFLL_d,\,ii}$ for $T = 0.1$ sec is readily available from the vector loop implementation. Following this, $W_{VFLL_d,\,ii}$ for different values of T is calculated by dividing $N_{0_{VFLL,\,i}}$ by the corresponding T. Next, the continuous-time process noise PSD is calculated from the discrete-time process noise covariance Q at $T = 0.1$ sec. From this, Q for other values of T is determined. Then, using the discrete-time process and measurement noise covariances for a given T and the filter models, the discrete-time steady state Kalman gain is calculated for a particular geometry matrix and the initial error covariance as the identity matrix. This Kalman gain is used in the transfer function models to calculate the discrete-domain noise bandwidths for that T using Equation 4.65 or 4.67 (note that both equations are equivalent in this case because of the same noise variance in all channels). Similarly, the continuous-domain noise bandwidths can be calculated using Equation 4.64 or 4.66. Ideally, the Kalman gains should be derived from the vector loop implementation with different update rates corresponding to the respective geometry matrix. However, in view of the fact that the geometry matrix does not change much over the simulation duration, above sub-optimal method is a reasonable approximation. It is easy to implement and serves the purpose of understanding the sensitivity of noise bandwidths to different underlying parameters such as noise covariance and update rates.

In this analysis T is varied from 0.01 sec to 0.2 sec. Values of T less than 0.01 sec are not considered because T is larger than 0.02 sec in vector loops. On the other hand, for T larger than 0.2 sec, the bandwidth ratio can be predicted from Figure 4.5 to approach 2. Hence, results for those values of T are also omitted. In order to consider the effect of noise levels, $W_{VFLL_d,\,ii}$ at $T = 0.1$ sec is varied in Figure 4.5 from 0.01 ($C/N_0 \backsim 50$ dB-Hz) to 100 ($C/N_0 \backsim 10$ dB-Hz) in steps of 10, each giving one value of $N_{0_{VFLL,\,i}}$.

It is evident from Figure 4.5 that the ratio of discrete-domain to approximate continuous-domain noise bandwidths depend on the magnitudes of T as well as $N_{0_{VFLL,\,i}}$. For a fixed T the bandwidth ratio is closer to one as $N_{0_{VFLL,\,i}}$ increases. However, for all combinations of T and $N_{0_{VFLL,\,i}}$ the ratio remains between 1 and 2 and decreases as T decreases for a given $N_{0_{VFLL,\,i}}$.

From the results of Figure 4.5, the following rule of thumb is proposed to determine the factor that relates the discrete-domain noise bandwidth to the approximate continuous-domain noise bandwidth. For T larger than 0.15 sec the factor is 2 if

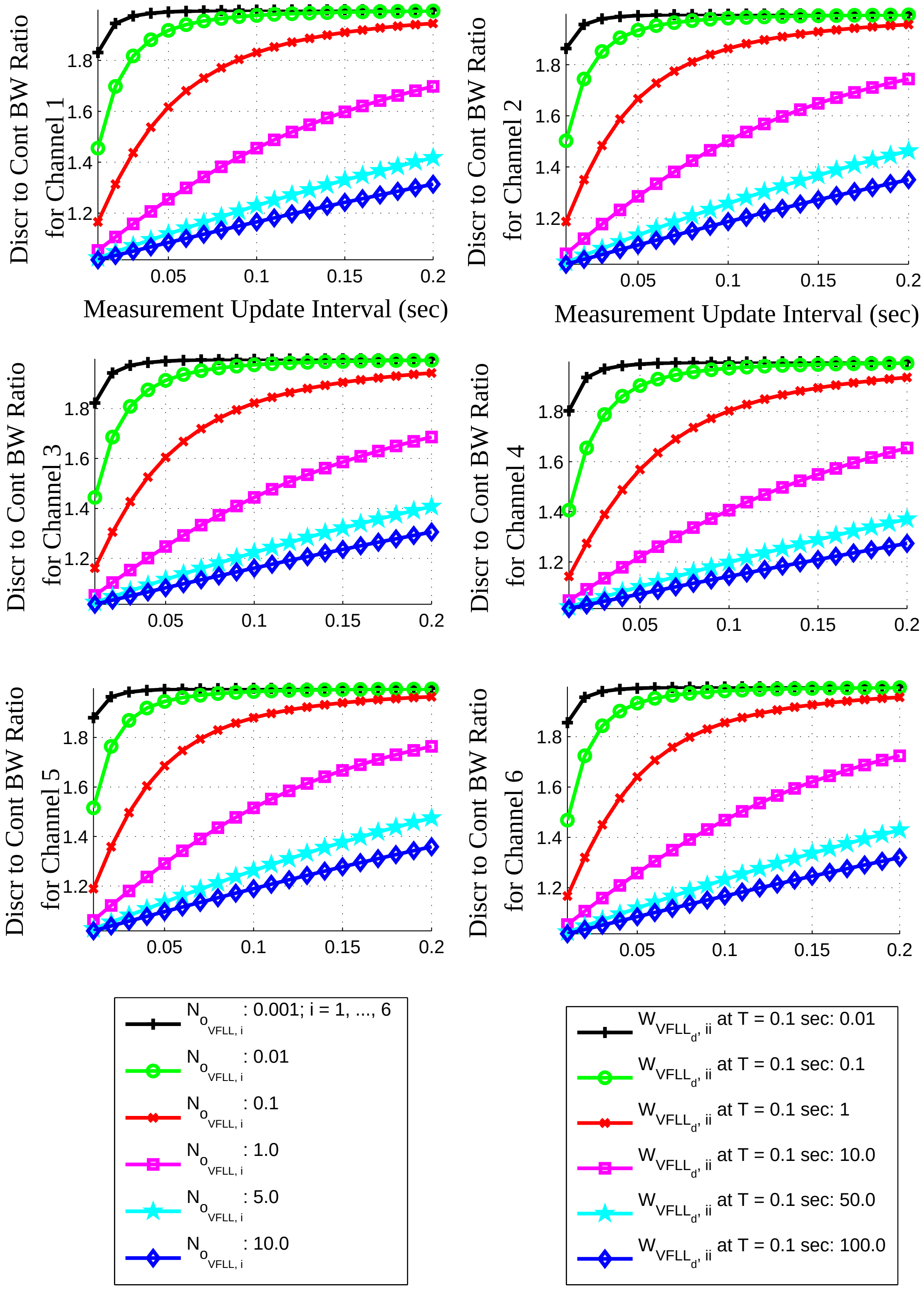


Figure 4.5: Discrete-Domain to Continuous-Domain Noise Bandwidth Ratios of VFLL for Different Measurement Update Rates; All Channels have the Same Noise Variance.

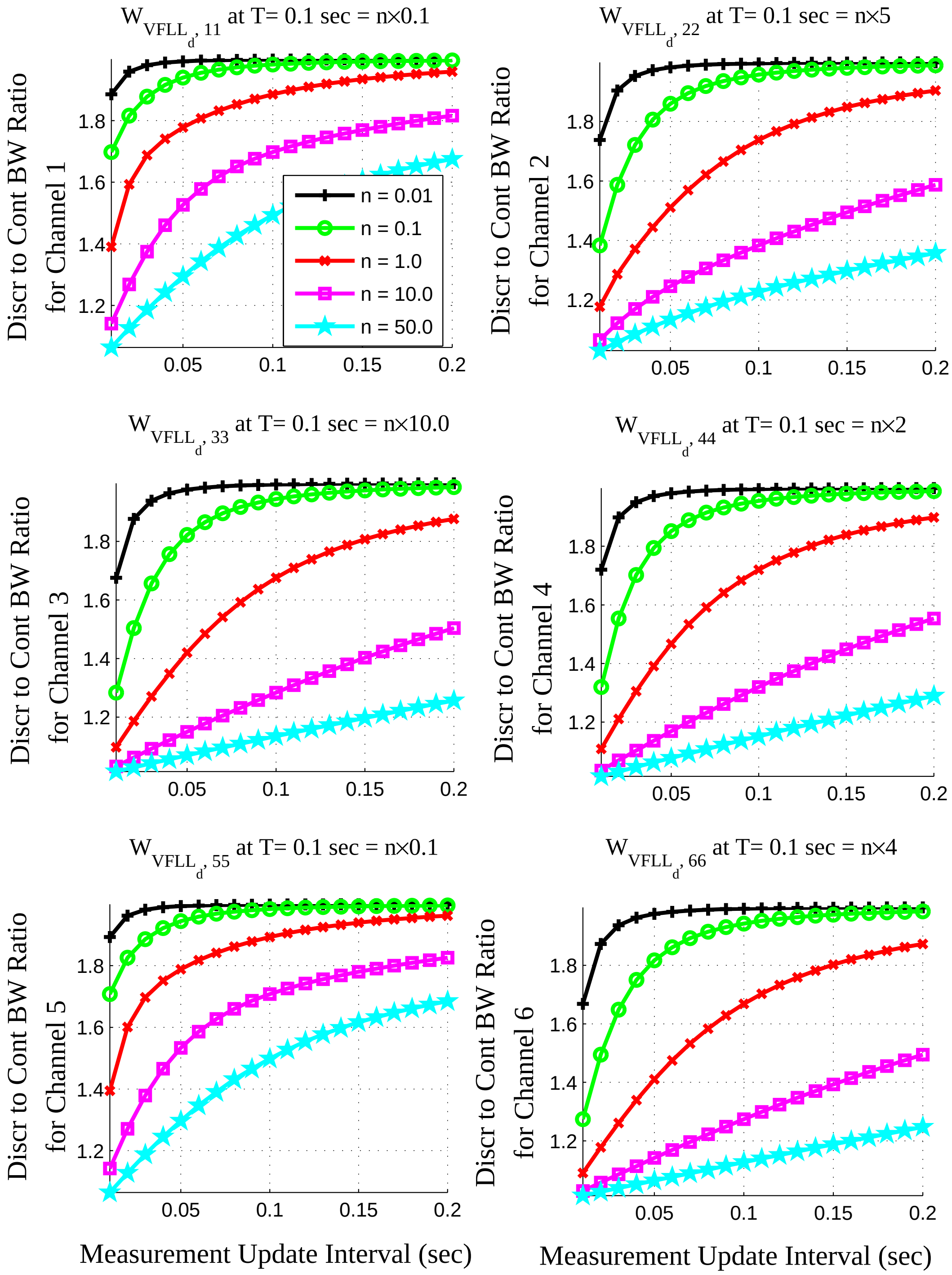


Figure 4.6: Discrete-Domain to Continuous-Domain Noise Bandwidth Ratios of VFLL for Different Measurement Update Rates; In this Figure the Noise Variance Varies across Channels, unlike Figure 4.5. The Bandwidths are Calculated Using the First Definition.

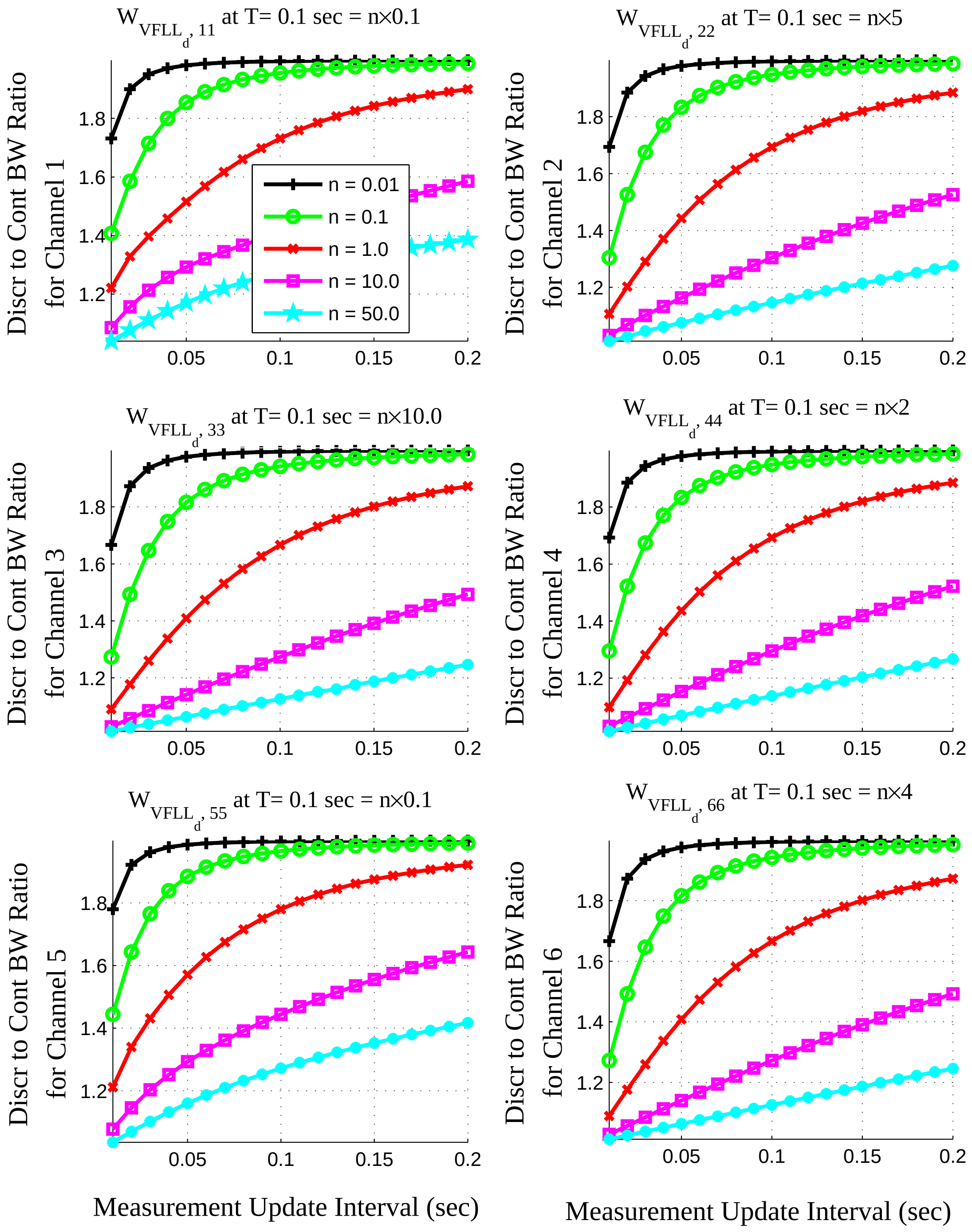


Figure 4.7: Discrete-Domain to Continuous-Domain Noise Bandwidth Ratios of VFLL for Different Measurement Update Rates; In this Figure the Noise Variance Varies across Channels, unlike Figure 4.5. The Bandwidths are Calculated Using the Second Definition.

$N_{0_{VFLL,\,i}}$ is less than 5. For T between 0.02 sec and 0.15 sec it is 1.5 if $N_{0_{VFLL,\,i}}$ is greater than 0.1 and less than 5. For T between 0.02 sec and 0.15 sec it is 2 if $N_{0_{VFLL,\,i}}$ is less than or equal to 0.1. For $N_{0_{VFLL,\,i}}$ greater than 5 the factor is 1 for all values of T. It is important to note that the value of the factor obtained from the theoretical expressions of the noise bandwidths is two. This factor is multiplied with the VFLL noise bandwidth calculated from the Laplace transfer function model to obtain the approximate discrete-domain VFLL noise bandwidth. The analysis above, however, considered only one LOS geometry i.e, a particular $G_{d,\,l}$ and the same noise variance for all channels. Thorough studies with different geometries and different noise levels across channels are, therefore, needed to validate this rule of thumb. To this end, preliminary results for a set of different noise variances across channels are provided in Figures 4.6 and 4.7 for two types of bandwidths, respectively. W_{VFLL_d} at $T = 0.1$ sec in this case is $n\times$diag([0.1 5.0 10.0 2.0 0.1 4.0]), where $n = 0.01$, 0.1, 1.0, 10.0 and 50. These figures support the rule of thumb developed above for determining the bandwidth ratio.

In addition to the bandwidth ratios, the actual values of the approximate continuous-domain and discrete-domain bandwidths are worthy of study. They are shown in Figure 4.8 for the first case with identical noise variance across channels. In this figure $W_{VFLL_d,\,ii}$ at $T = 0.1$ sec is varied from 0.01 to 10, encompassing a C/N$_0$ variation from about 50 dB-Hz to about 20 dB-Hz. The process noise covariance is diag([0.1 0.1 0.1]) at $T = 0.1$ sec. Three observations from this figure are worth noting. First, for a given noise level the noise bandwidth decreases as T increases. For small noise levels the rate of decrease is fast up to $T = 0.05$ sec and slows down after that. The change in the rate of decrease diminishes for larger noise. Second, the noise bandwidth becomes less sensitive to changes in the input noise variance as the value of T goes up, meaning the noise bandwidth for higher values of T is mainly determined by T. Third, all channels have more or less similar bandwidths in this case because of identical measurement noise variance across channels.

In addition, a preliminary sensitivity analysis of the noise bandwidth with respect to different process noise levels is performed. It shows that the noise bandwidths increase slowly as the process noise covariance is made higher for a given measurement noise level and small T. For instance, a hundredfold increase in the first two diagonal elements of the process noise covariance at $T = 0.1$ sec from 0.1 to 10 results in 6 times as large noise bandwidth when $T = 0.02$ sec whereas the increase is 3 times when $T = 0.1$ sec. Both times $W_{VFLL_d,\,i}$ is 30 at $T = 0.1$ sec (C/N$_0$ $\backsim$ 15 dB-Hz). It is also

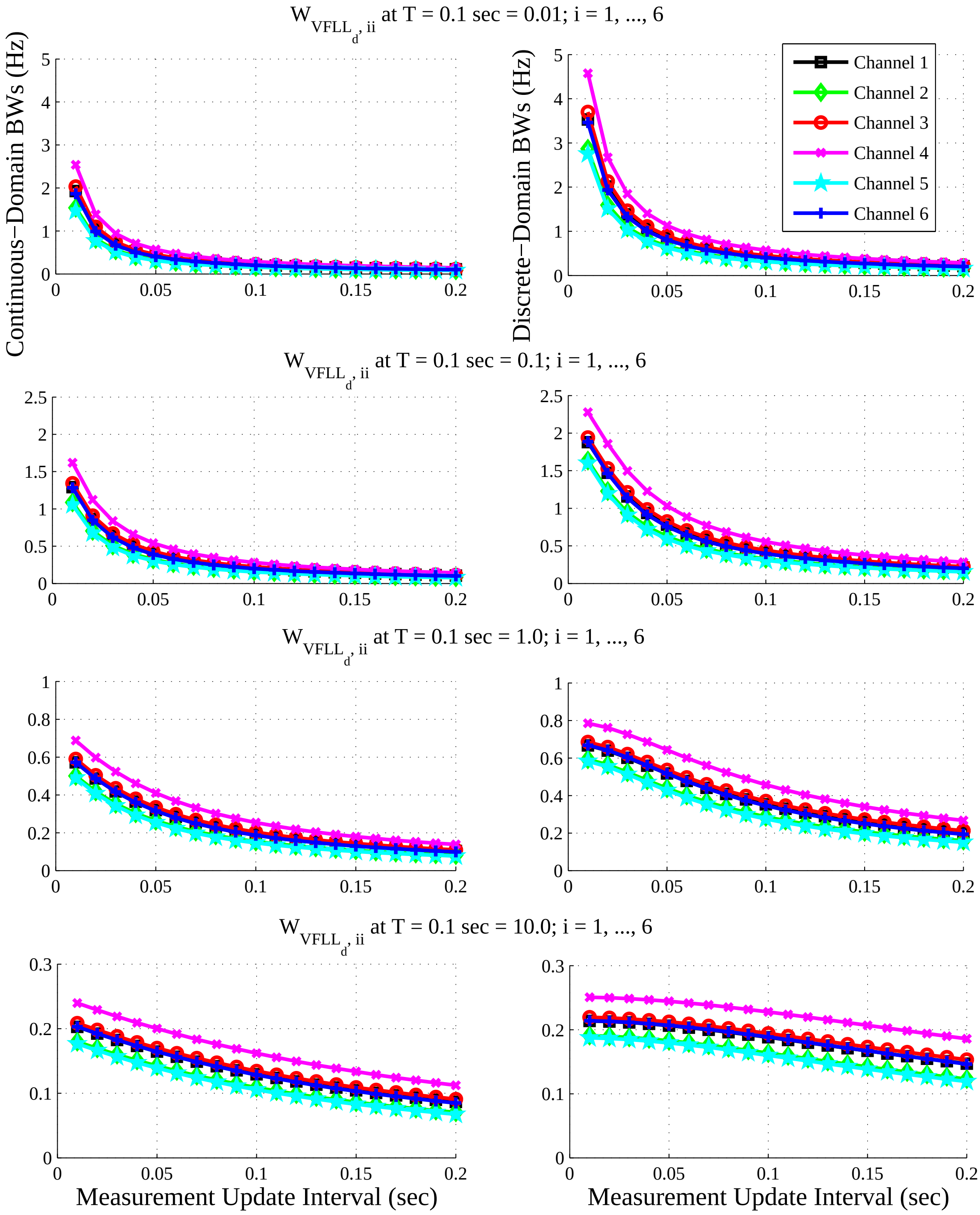


Figure 4.8: Discrete-Domain and Continuous-Domain Noise Bandwidths of VFLL for Different Measurement Update Rates

noted that the bandwidth increase is more at higher measurement noise levels.

Next, Figures 4.9 and 4.10 illustrate the actual values of the continuous-domain and discrete-domain noise bandwidths as calculated from the two definitions of noise bandwidths, respectively, for the second case with different noise variances across channels. Two values of n are considered: 0.01 and 10. The process noise covariance is retained the same as the previous case. While the figures show the same monotonically decreasing trend as in Figure 4.8, the following points deserve a mention. Figure 4.9 shows that the noise bandwidths calculated from the first definition (i.e., Equations 4.64 and 4.65) can be more for channels with higher input noise PSD. For example, channel 3 has the highest input noise variance of all channels. Its bandwidth is higher than all channels except channel 6. There also are exceptions as channel 6 has a higher noise bandwidth than that of channel 3. Channel 4 whose noise variance is lower than that of channel 2 has a higher noise bandwidth than that of channel 2. The above findings lead to the following conclusion. A channel that has a high input noise variance may or may not have a small noise bandwidth when it is calculated using the first definition of noise bandwidths. This is possible because the noise bandwidth of a channel, in addition to its own noise, Kalman gains and LOS geometry, depends on noise contributions from all other channels. It is, therefore, evident that Equations 4.64 and 4.65 are not suitable for designing the corresponding scalar loop noise bandwidths.

Figure 4.10 depicts the noise bandwidths that are obtained from Equations 4.66 and 4.67 i.e., from the second definition. It is clear from the figure that a channel with high input noise variance is assigned a lower noise bandwidth, thus enabling this bandwidth to be a good candidate for designing the scalar loop noise bandwidth. The inverse relationship between the bandwidth and input noise could be achieved in this case because the input channel is fixed, unlike the previous definition where all channel inputs contribute to determining the noise bandwidth. Thus, the noise bandwidth obtained from the second definition is a function of the LOS geometry and Kalman gains only that decrease with larger noise and increase with higher uncertainties in the dynamic model. Consequently, this bandwidth can be expected to shrink with higher noise levels and expand with large dynamics. It is also important to note the difference in a channel noise bandwidth when its noise level is different from those of the other channels. For example, consider the top right plots of Figures 4.8 and 4.10. While all channels in that plot of Figure 4.8 have a bandwidth of about 0.5 Hz at $T = 0.1$ sec, channel 1 in the top right plot of Figure 4.10, which has the same noise level as that of the channels in the corresponding plot of Figure 4.8, has a bandwidth of 4 Hz at

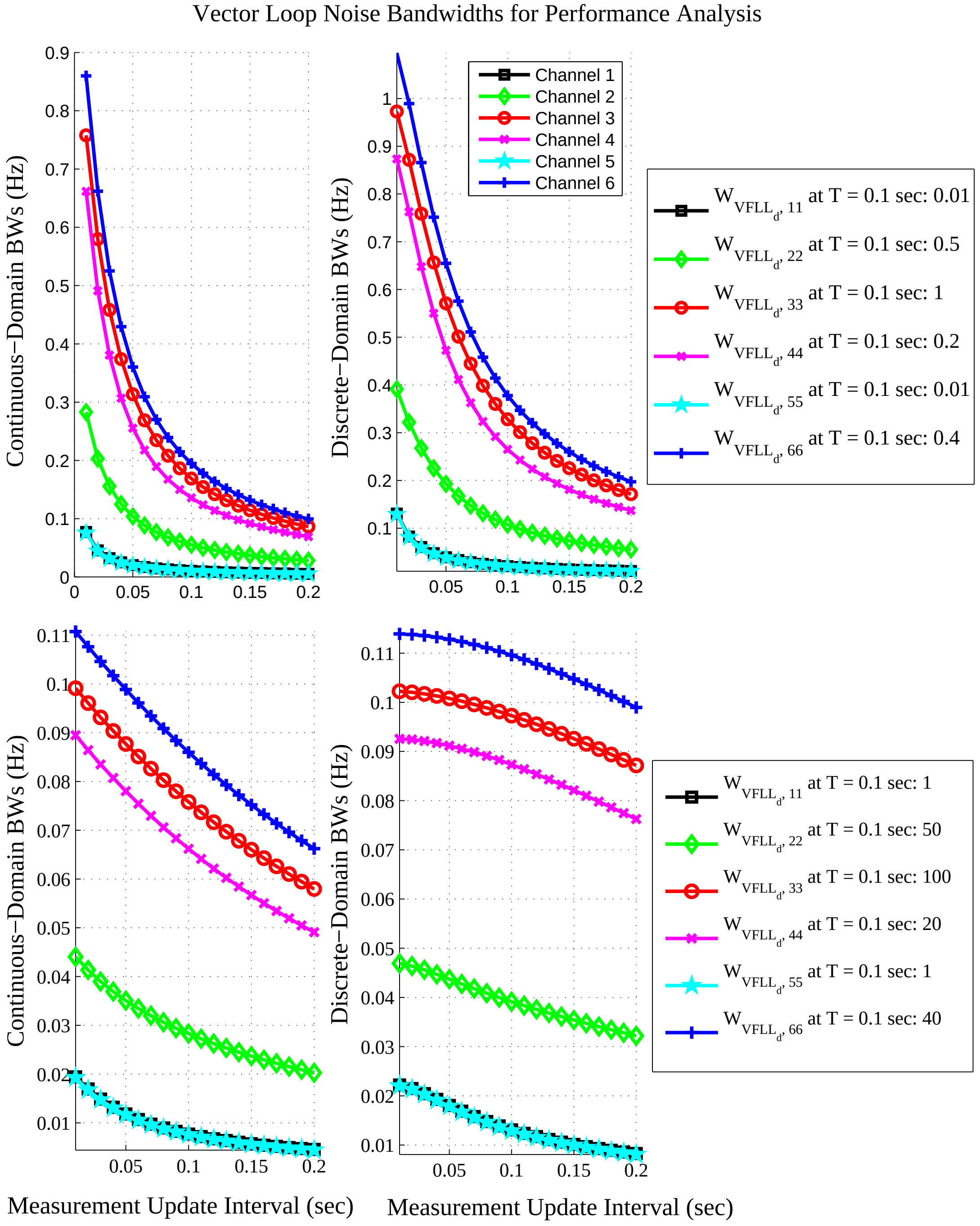


Figure 4.9: Discrete-Domain and Continuous-Domain Noise Bandwidths of VFLL for Different Measurement Update Rates; In this Figure the Noise Variance Varies across Channels, unlike Figure 4.8. The Bandwidths are Calculated Using the First Definition.

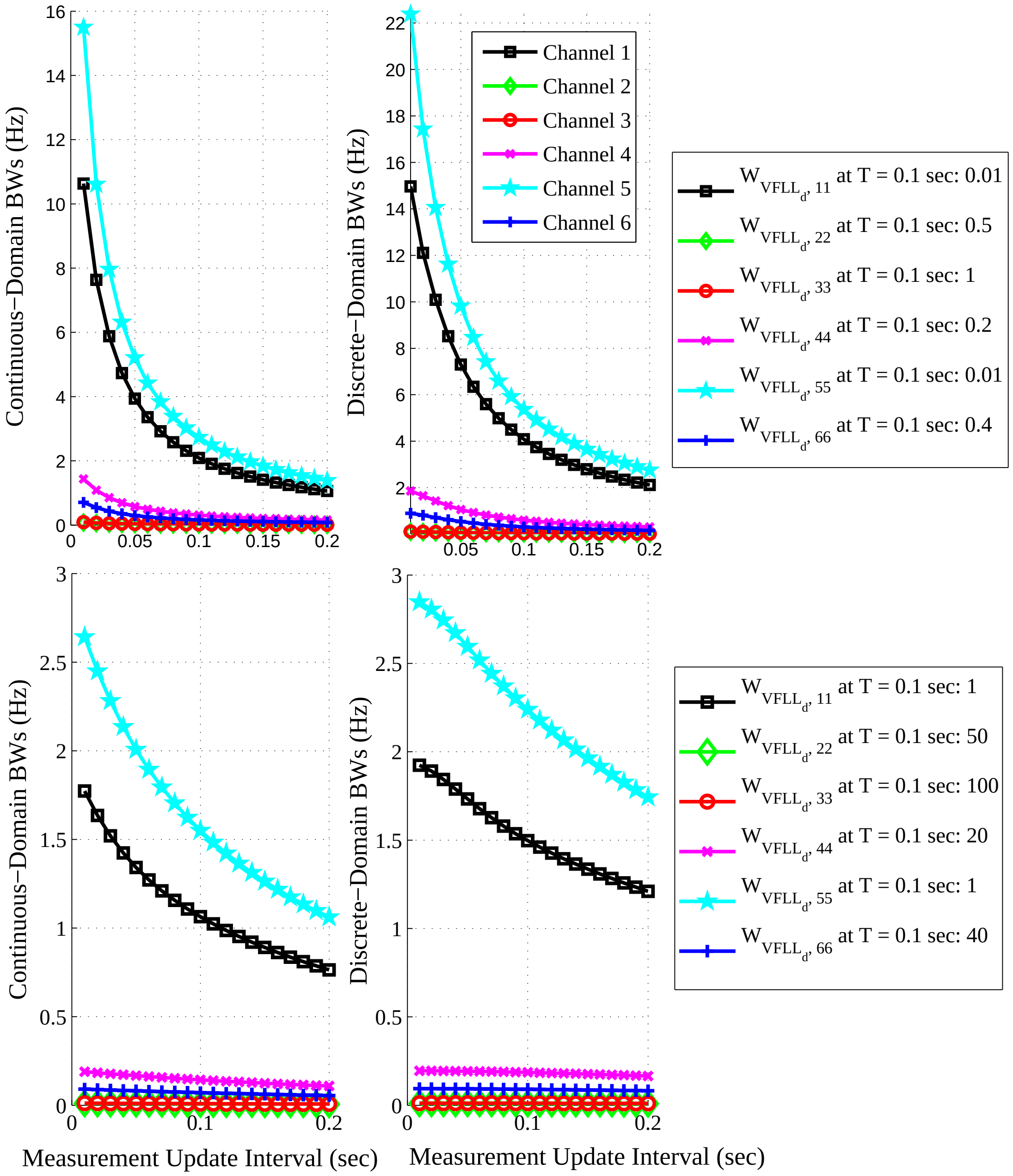


Figure 4.10: Discrete-Domain and Continuous-Domain Noise Bandwidths of VFLL for Different Measurement Update Rates; In this Figure the Noise Variance Varies across Channels, unlike Figure 4.8. The Bandwidths are Calculated Using the Second Definition.

$T = 0.1$ sec. Thus, the bandwidth assigned to a channel not only depends on its own noise level, but also on the noise levels of other channels.

The above analysis of vector loop noise bandwidths reveals some interesting properties which not only shed light on the internal workings of the vector architecture, but also justify the worth of vector loop noise bandwidths to design the scalar loops. For simulation studies discussed in the next section, all channel inputs are assumed to have the same noise variance. Hence, both definitions of vector loop noise bandwidths result in the same bandwidths. In the simulation studies the discrete-domain vector loop noise bandwidths are used to select the scalar loop noise bandwidths for fair performance comparison of two architectures. However, the FLL noise bandwidths are not set equal to the vector loop values. Instead, they are varied to obtain the best possible scalar loop performance while not moving far away from the VFLL noise bandwidths. This is prompted by the fact that vector loops benefit from internal aiding among channels while individual scalar loops do not.

4.2.2.4 Sample Transfer Function Models

To conclude the discussion of transfer function models, four VFLL models in the continuous domain which are obtained with approximate continuous Kalman gains are displayed below. They pertain to the constant-velocity user motion with pseudolites. The first two transfer function models each have identical noise variance across channels. Their measurement update intervals (T) are 0.1 sec and 0.02 sec, respectively. Individual channel noise variances of the two models are 0.01 and 0.05, both having the same continuous-time noise PSD of 0.001. These models correspond to the upper left plot of Figure 4.8. The third and fourth transfer function models are illustrated for the same two measurement update intervals, but with different noise variances across channels. The continuous-time noise PSDs of both models are $N_{0_{VFLL_d}} = \text{diag}([0.01\ 0.5\ 1.0\ 0.2\ 0.01\ 0.4]) \times 0.1$. They correspond to the upper left plots of Figures 4.9 and 4.10. From these models the following important observations are made:

1. The update interval has a direct bearing on the locations of poles and zeros of each transfer function element as they are close to $1/T$ in each model. In addition, the magnitudes of poles and zeros vary inversely with measurement noise variances when T is small. In this context, the difference in pole and zero locations in the second and fourth transfer function models where $T = 0.02$ sec is noteworthy. As an aside, it is instructive to note that the magnitudes of poles and zeros increase slowly with higher process noise for small T ($\backsim 0.02$ sec).

2. Although each transfer function element is of third order because of three independent states (x and y coordinates of user velocity and clock drift), two zeros are located very close to two poles, resulting in effectively first order transfer function elements. The pole-zero cancelation can probably be attributed to the nearly idempotent property of the $G_{d,\,l}K_{v,\,l}$ matrix. With least squares gains the idempotent property results in the theoretical expression of the continuous-domain transfer function matrix having a single pole at $1/T$ (Equation 4.87). Since the Kalman gains are not significantly different from the least squares gains in this case, $G_{d,\,l}K_{v,\,l}$ is approximately idempotent up to some power, resulting in closely located poles and zeros.

3. In the first two transfer function models the amplitude of a transfer function element can be large or small in any row or column i.e., large amplitudes are evenly distributed. This is because the channel noise variances are identical. In contrast, the amplitude of each transfer function element of the remaining two models is inversely related to the corresponding channel input noise variance. For example, columns 1 and 5 of these models have large amplitudes in almost all transfer function elements while those of columns 2 and 3 are substantially small (note their PSDs above). This is reflected in higher noise bandwidths of channels 1 and 5 when they are derived using the second definition (see Figure 4.10).

4. Both absolute and relative magnitudes of measurement noise variances affect the locations of transfer function poles and zeros at small T while the relative magnitudes have a strong influence on determining the transfer function amplitudes.

Having discussed the vector loop parametric models and noise bandwidths, an explanation for vector loops' superior performance is provided next using the developed discrete parametric model.

$$[H'_{VFLL_{cont},l}]_{j\times j} = \frac{1}{(s+9.99)(s+9.99)(s+9.98)}\times$$

$$\begin{bmatrix}
5.11(s+9.99)(s+9.99) & 1.39(s+9.99)(s+9.97) & -1.56(s+10.01)(s+9.99) & 2.34(s+9.99)(s+9.99) & 3.74(s+9.99)(s+9.98) & -1.03(s+10.00)(s+9.98) \\
1.39(s+9.99)(s+9.97) & 4.00(s+9.99)(s+9.98) & 3.22(s+9.99)(s+9.99) & -1.80(s+10.00)(s+9.99) & 2.89(s+9.99)(s+9.98) & 0.31(s+9.98)(s+9.98) \\
-1.56(s+10.01)(s+9.99) & 3.22(s+9.99)(s+9.99) & 5.41(s+9.99)(s+9.99) & -0.88(s+10.02)(s+9.99) & 0.50(s+9.99)(s+9.96) & 3.31(s+9.99)(s+9.98) \\
2.34(s+9.99)(s+9.99) & -1.80(s+10.00)(s+9.99) & -0.88(s+10.02)(s+9.99) & 6.85(s+9.99)(s+9.99) & 0.02(s+10.01)(s+10.01) & 3.46(s+9.99)(s+9.98) \\
3.74(s+9.99)(s+9.98) & 2.89(s+9.99)(s+9.98) & 0.50(s+9.99)(s+9.96) & 0.02(s+10.01)(s+10.01) & 3.75(s+9.99)(s+9.98) & -0.91(s+10.00)(s+9.98) \\
-1.03(s+10.00)(s+9.98) & 0.31(s+9.98)(s+9.98) & 3.31(s+9.99)(s+9.98) & 3.46(s+9.99)(s+9.98) & -0.91(s+10.00)(s+9.98) & 4.85(s+9.99)(s+9.98)
\end{bmatrix}$$

Figure 4.11: Transfer Funtion Model with Equal Noise Variance of All Channels for T = 0.1 sec

$$[H'_{VFLL_{cont},l}]_{j\times j} = \frac{1}{(s+49.31)(s+48.82)(s+47.86)}\times$$

$$\begin{bmatrix}
24.85(s+49.03)(s+48.37) & 7.05(s+48.92)(s+46.55) & -7.31(s+50.69)(s+48.67) & 11.21(s+49.07)(s+49.07) & 18.34(s+49.00)(s+48.02) & -4.96(s+50.01)(s+48.03) \\
7.05(s+48.92)(s+46.55) & 19.56(s+48.90)(s+48.18) & 15.66(s+48.66)(s+48.66) & -8.57(s+49.98)(s+48.57) & 14.25(s+48.93)(s+47.72) & 1.55(s+47.69)(s+47.69) \\
-7.31(s+50.69)(s+48.67) & 15.66(s+48.66)(s+48.66) & 26.29(s+49.02)(s+48.48) & -3.96(s+52.55)(s+49.04) & 2.57(s+48.53)(s+45.68) & 16.21(s+49.11)(s+47.96) \\
11.21(s+49.07)(s+49.07) & -8.57(s+49.98)(s+48.57) & -3.96(s+52.55)(s+49.04) & 33.14(s+49.25)(s+48.42) & 0.11(s+49.66)(s+49.66) & 16.99(s+49.22)(s+47.76) \\
18.34(s+49.00)(s+48.02) & 14.25(s+48.93)(s+47.72) & 2.57(s+48.53)(s+45.68) & 0.11(s+49.66)(s+49.66) & 18.43(s+48.99)(s+47.90) & -4.35(s+50.13)(s+47.91) \\
-4.96(s+50.01)(s+48.03) & 1.55(s+47.69)(s+47.69) & 16.21(s+49.11)(s+47.96) & 16.99(s+49.22)(s+47.76) & -4.35(s+50.13)(s+47.91) & 23.74(s+49.23)(s+47.87)
\end{bmatrix}$$

Figure 4.12: Transfer Funtion Model with Equal Noise Variance of All Channels for T = 0.02 sec

$$[H'_{VFLL_{cont},l}]_{j\times j} = \frac{1}{(s+9.99)(s+9.83)(s+9.70)}\times$$

$$\begin{bmatrix}
8.80(s+9.89)(s+9.72) & -0.17(s+10.09)(s+9.75) & -0.15(s+10.01)(s+9.78) & 0.40(s+9.97)(s+9.59) & 1.29(s+9.74)(s+9.26) & -0.17(s+10.00)(s+9.91) \\
-8.26(s+10.09)(s+9.75) & 0.93(s+9.98)(s+9.76) & 0.71(s+9.99)(s+9.80) & -0.06(s+10.00)(s+6.25) & 15.42(s+9.94)(s+9.73) & 1.17(s+9.99)(s+9.86) \\
-15.21(s+10.01)(s+9.78) & 1.43(s+9.99)(s+9.80) & 1.43(s+9.99)(s+9.82) & 2.73(s+9.99)(s+9.90) & 16.06(s+9.97)(s+9.77) & 3.34(s+9.99)(s+9.84) \\
8.01(s+9.97)(s+9.59) & -0.02(s+10.00)(s+6.25) & 0.55(s+9.99)(s+9.90) & 5.75(s+9.99)(s+9.76) & -7.29(s+10.01)(s+9.62) & 2.81(s+9.99)(s+9.81) \\
1.29(s+9.74)(s+9.26) & 0.31(s+9.94)(s+9.73) & 0.16(s+9.97)(s+9.77) & -0.36(s+10.01)(s+9.62) & 8.48(s+9.89)(s+9.71) & 0.12(s+9.97)(s+9.97) \\
-6.89(s+10.00)(s+9.91) & 0.94(s+9.99)(s+9.86) & 1.33(s+9.99)(s+9.84) & 5.61(s+9.99)(s+9.81) & 4.60(s+9.97)(s+9.97) & 4.12(s+9.99)(s+9.82)
\end{bmatrix}$$

Figure 4.13: Transfer Funtion Model with Different Noise Variances across Channels for $T = 0.1$ sec

$$[H'_{VFLL_{cont},l}]_{j\times j} = \frac{1}{(s+48.89)(s+37.54)(s+32.86)} \times$$

$$\begin{bmatrix}
38.58(s+41.47)(s+33.50) & -0.51(s+58.80)(s+34.50) & -0.52(s+50.95)(s+35.66) & 1.73(s+47.42)(s+29.49) & 10.51(s+34.12)(s+20.48) & -0.53(s+49.66)(s+41.05) \\
-25.34(s+58.80)(s+34.50) & 3.44(s+47.64)(s+35.03) & 2.51(s+48.52)(s+36.27) & -0.92(s+49.89)(s+5.28) & 61.84(s+44.97)(s+33.98) & 3.84(s+48.82)(s+38.93) \\
-51.91(s+50.95)(s+35.66) & 5.02(s+48.52)(s+36.27) & 4.92(s+48.81)(s+37.03) & 8.65(s+48.67)(s+40.58) & 59.58(s+47.14)(s+35.26) & 11.20(s+48.88)(s+37.91) \\
34.66(s+47.42)(s+29.49) & -0.37(s+49.89)(s+5.28) & 1.73(s+48.67)(s+40.58) & 20.82(s+48.85)(s+34.91) & -28.88(s+50.73)(s+30.31) & 9.72(s+48.88)(s+36.62) \\
10.51(s+34.12)(s+20.48) & 1.24(s+44.97)(s+33.98) & 0.60(s+47.14)(s+35.26) & -1.44(s+50.73)(s+30.31) & 37.50(s+41.35)(s+33.26) & 0.35(s+46.72)(s+44.62) \\
-21.19(s+49.66)(s+41.05) & 3.07(s+48.82)(s+38.93) & 4.48(s+48.88)(s+37.91) & 19.45(s+48.88)(s+36.62) & 13.88(s+46.72)(s+44.62) & 14.04(s+48.89)(s+37.33)
\end{bmatrix}$$

Figure 4.14: Transfer Funtion Model with Different Noise Variances across Channels for $T = 0.02$ sec

4.2.3 Vector Tracking's Superiority: An Explanation

A close study of the vector loop discrete parametric model shows that the vector tracking architecture derives its benefits from two types of constraints: *Position/velocity constraint* and *least squares constraint*. In what follows, these constraints are explained in the context of VFLL using Figures 4.15 and 4.16 and Equation 4.4. The same treatment can also be repeated with VDLL after some modifications.

Figure 4.15 provides an intuitive explanation of the following concepts: *Velocity constraint* of the Doppler frequencies and tracking thresholds of both scalar and vector architectures. The Doppler frequency of each satellite is represented in the figure by a vector in the three dimensional coordinate system. The three orthogonal axes of the coordinate frame are the norm of user velocity, user clock drift and the norm of satellite velocity, respectively. It should be noted that velocity norms are used and satellite clock drifts are ignored in this analysis to reduce the dimensionality of the coordinate frame. Since, at a given instant of time, all Doppler frequencies have the same horizontal coordinates, the frequency vectors in Figure 4.15 are ideally constrained to lie on a straight line perpendicular to the horizontal plane. This is termed the *velocity constraint* of the Doppler frequencies. By estimating user velocity and clock drift, the vector architecture makes use of this constraint while independent scalar tracking loops do not. However, because of noise in the received signals, the actual Doppler frequency is assumed to lie on an uncertainty sphere centered on the true Doppler as shown in Figure 4.15 by the outermost solid circle. Tracking loops, therefore, include low pass filters to reduce noise in the estimated Doppler. The low pass filters, in turn, define loop noise bandwidths which can be varied to achieve a trade off between good noise rejection and fast dynamic response. Thus, for a given C/N_0 it is always possible to find an optimum noise bandwidth for which the tracking jitter or error in the estimated Doppler is minimum.

In the figure the dashed circle represents the minimum error sphere for a scalar tracking loop whereas the innermost solid circle represents the same for an individual loop in vector tracking mode. The solid circle on the borders of the two differently shaded regions denotes the rule-of-thumb tracking threshold. Therefore, the shaded region inside this circle depicts locked loops, implying that loss of lock occurs if the estimated Doppler lies outside this region. Since a scalar tracking loop requires to track full signal dynamics, narrowing its noise bandwidth improves resistance to noise at the expense of dynamic response. This trade off limits its ability to track low C/N_0 signals and high dynamics at a given C/N_0. On the other hand, Equation 4.4 shows that

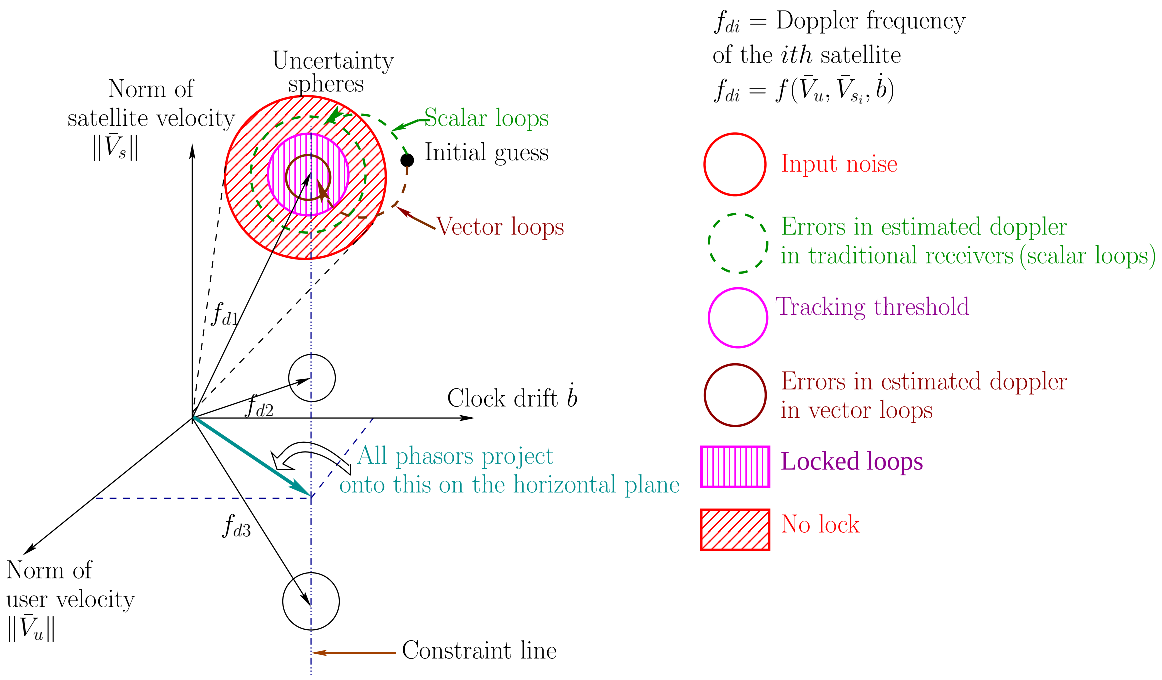


Figure 4.15: Velocity Constraint: All Doppler Frequencies are Generated by the Same User Velocity and Clock Drift

the vector tracking architecture allows tracking loops to *aid* each other. Therefore, their noise bandwidths can be reduced further without incurring as large dynamic stresses as in scalar loops. Thus, *internal aiding* in vector loops helps maintain lock at a lower C/N$_0$. It also helps withstand high user dynamics. User dynamics cause different Doppler shifts to different satellites depending on LOS geometry. Satellites with large Doppler errors are aided by others to maintain tracking while satellites with small Doppler errors may or may not require aiding depending on noise levels. The superiority of vector loops is summarized in the figure by showing the smallest solid circle in the locked region at a certain C/N$_0$ whereas the dashed circle lies outside it.

Now with the help of Figure 4.16 it will be shown intuitively how internal aiding is materialized in the vector architecture and under what scenarios it will help maintain lock. First, however, some clarifications of Figure 4.16 are in order to better understand subsequent theory of internal aiding. The figure provides a graphical interpretation of the correction term ($= \tilde{G}_{d,\,k+1} K_{v,\,k+1} \underline{e}_{d,\,k+1}$) of Equation 4.4. It will subsequently be referred to as $\delta \underline{f}$. For clarity a hypothetical case of only three visible satellites is considered, although the minimum number should be more than four. The elements

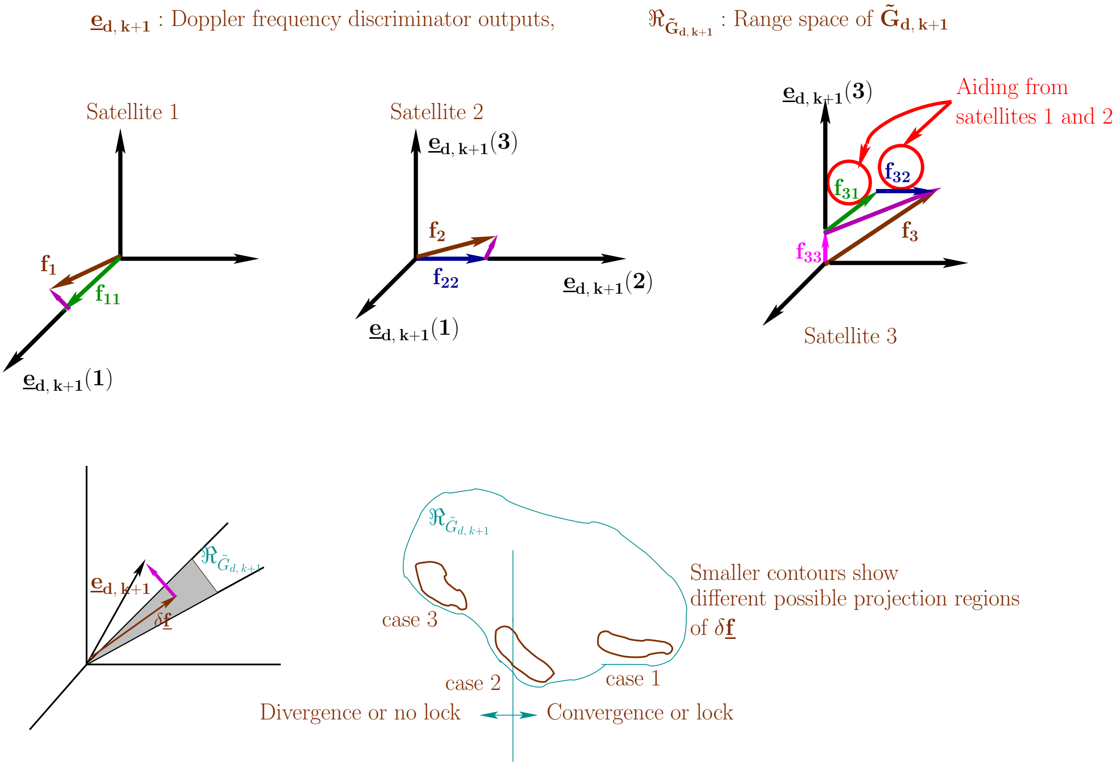


Figure 4.16: Least Squares Constraint: This Constraint Enables Internal Aiding among Tracking Loops and Guarantees Convergence when Good Signal-Strength Satellites or Satellites with Small Doppler Errors (e.g., Satellites 1 and 2) Aid Low C/N$_0$ Satellites or Satellites with Large Doppler Errors (Satellite 3) in such a Way that Minimizes their Errors.

of $\underline{e}_{d,\,k+1}$, in spite of being linearly dependent, are represented as orthogonal axes for simplicity. The large contour on the bottom right corner represents the range space of the geometry matrix $\tilde{G}_{d,\,k+1}$ and shows different projection regions of $\delta\underline{f}$. It is assumed for this analysis that the contributions of satellite motion to the Doppler frequency, being known, have been removed from the $\underline{\tilde{f}}^{-}_{d,\,carr_{k+1}}$ and $\underline{\tilde{f}}^{+}_{d,\,carr_{k+1}}$ terms of Equation 4.4. With this background the reasons of internal aiding and its effectiveness in certain scenarios is explained below.

In an over determined system where the number of measurements (i.e., number of visible satellites) exceeds the number of independent variables to be estimated (number of user velocity components and clock drift = 4), the Doppler frequencies (after removing the contributions of satellite motion) are linearly dependent. They are, therefore, constrained to lie in a subspace defined by the range space of the geometry matrix. This constraint is exploited by the recursive least squares algorithm of the Kalman filter and, hence, termed *least squares constraint* here. It will be shown that this constraint along with previously mentioned *velocity constraint* enables internal aiding among vector loops. This is clarified by explaining below the contributions of the correction term ($\delta \underline{f}$) of Equation 4.4.

$\delta \underline{f}$: Internal aiding in VFLL is realized through the term $\delta \underline{f}$. In order to show this, first the expression of $\delta \underline{f}$ is rewritten below:

$$\begin{aligned}
\delta \underline{f} &= [f_1 ... f_n]^T \\
\delta \tilde{V}_{k+1} &= \tilde{V}^{+}_{k+1} - \tilde{V}^{-}_{k+1} = K_{v,\,k+1} \underline{e}_{d,\,k+1} \\
\delta \underline{f} &= \tilde{G}_{d,\,k+1} \delta \tilde{V}_{k+1} \\
&= (\tilde{G}_{d,\,k+1} K_{v,\,k+1}) \underline{e}_{d,\,k+1} \\
&= A \underline{e}_{d,\,k+1} \qquad (4.89) \\
f_{ij} &= A(i,j) \underline{e}_{d,\,k+1}(j) \qquad (4.90) \\
f_i &= \sum_{j=1}^{n} f_{ij}, \quad 1 \le j \le n
\end{aligned}$$

where $\tilde{V}^{-/+}_{k+1}$ = predicted/estimated user velocity and clock drift at t_{k+1}; n = number of visible satellites; f_i is the i^{th} element of $\delta \underline{f}$; f_{ij} is the contribution or aiding of j^{th} loop to the correction of i^{th} loop. In the figure the length of f_i denotes its relative magnitude and for clarity f_{ij} is shown to be added as vectors. A close observation of the above equations reveals the following. First, a velocity correction term $\delta \tilde{V}_{k+1}$ is estimated and then, it is projected onto $\Re_{\tilde{G}_{d,\,k+1}}$ to find $\delta \underline{f}$. Since $\tilde{G}_{d,\,k+1}$ is a four-column rectangular matrix with more than four rows, the elements of $\delta \underline{f}$ are linearly dependent and belong to a particular subspace defined by $\Re_{\tilde{G}_{d,\,k+1}}$. In this context, note that the residual measurement vector $\underline{e}_{d,\,k+1}$ should ideally be in $\Re_{\tilde{G}_{d,\,k+1}}$ as the true Doppler frequency vector $\underline{f}_{true,d_{k+1}} \in \Re_{\tilde{G}_{d,\,k+1}}$ and $\underline{\tilde{f}}^{-}_{d,\,carr_{k+1}} \in \Re_{\tilde{G}_{d,\,k+1}}$, where contributions of satellite motion to the Doppler vectors are assumed to be removed. However, the elements of $\underline{e}_{d,\,k+1}$ usually lie away from $\Re_{\tilde{G}_{d,\,k+1}}$ mainly due to noise or sometimes due to discriminator nonlinearities. So, $\delta \underline{f}$ is the least squares projection of $\underline{e}_{d,\,k+1}$ on $\Re_{\tilde{G}_{d,\,k+1}}$. The recursive least squares algorithm of the EKF ensures that, given an $\underline{e}_{d,\,k+1}$, $\delta \underline{f}$ is the best possible estimate of the true frequency correction vector ($= \underline{f}_{true,d_{k+1}} - \underline{\tilde{f}}^{-}_{d,\,carr_{k+1}}$) on $\Re_{\tilde{G}_{d,\,k+1}}$. This implies that the elements of $\underline{e}_{d,\,k+1}$ which closely satisfy the linear dependence of $\Re_{\tilde{G}_{d,\,k+1}}$ will define the projection region on $\Re_{\tilde{G}_{d,\,k+1}}$. Because for a given $\underline{e}_{d,\,k+1}$, those elements would change less after projection to obtain the best possible estimate $\delta \underline{f}$ on $\Re_{\tilde{G}_{d,\,k+1}}$. As a result, the remaining outlying elements of $\underline{e}_{d,\,k+1}$ are pulled into that projection region, resulting in internal aiding.

A geometric interpretation of the above explanation is as follows. By definition, f_{ii} and $\underline{e}_{d,\,k+1}(i)$ are related as $f_{ii} = c\underline{e}_{d,k+1}(i)$, where from Equation 4.90 $c = A(i,i)$. It follows from the definition and the positive definite property of A that $0 < c < 1$ and hence $\underline{e}_{d,\,k+1}(i) > f_{ii}$. So, in the figure f_{ii} is drawn in the direction of $\underline{e}_{d,\,k+1}(i)$ and smaller in length than $\underline{e}_{d,\,k+1}(i)$. It is assumed that satellites 1 and 2 correspond to either good C/N$_0$ satellites or satellites with small frequency errors or errors that are corrupted by relatively small noise samples. As a consequence, $\underline{e}_{d,\,k+1}(i)$, $i = 1, 2$ closely satisfy the linear dependence of the Doppler frequencies and, therefore, f_1 and f_2 change marginally from $\underline{e}_{d,\,k+1}(1)$ and $\underline{e}_{d,\,k+1}(2)$, respectively, after projection. On the other hand, satellite 3 does not satisfy the linear dependence due to either of the two reasons: First, more noise or second, a large frequency error. Sometimes the error can even be large enough to lie outside the discriminator pull-in range and, thus, produces a wrong discriminator output. This is possible with a long coherent integration time ($\geq$ 10 ms) which reduces noise, but at the same time narrows down the pull-in range. This will be explained in detail later with simulation studies. In order to show that more noise results in smaller c as $K_{v,\,k+1}$ is inversely related to measurement noise, f_{33} is drawn as a small fraction of $\underline{e}_{d,\,k+1}(3)$. Now note that, with the help of aiding from satellites 1 and 2, f_3 is significantly "bent" from $\underline{e}_{d,\,k+1}(3)$ to be pulled onto $\Re_{\tilde{G}_{d,\,k+1}}$. However, such internal aiding guarantees convergence if there is a sufficient number of satellites which correctly define the projection region (cases 1 and 2) and pulls the outlying ones into that region. Case 2 indicates that some satellites may be in the divergence region in the beginning, but are pulled later in the correct direction. Finally, case 3 illustrates an extreme situation when none of the satellites lie near the correct region and hence vector loops, in spite of having internal aiding, do not maintain lock.

Thus, from the above analysis the following conclusion is made: In the vector architecture a low C/N$_0$/blocked satellite is aided by good signal strength satellites to bridge a momentary outage. In the presence of multiple low C/N$_0$ satellites vector tracking narrows down the noise bandwidths by allowing longer integration times of all channels and decreasing the Kalman gain adaptively with enhanced noise levels. However, this does not affect its dynamic performance as satellites with large Doppler offsets are aided by others to satisfy the linear dependence of all Doppler frequencies. Finally, the same explanation of inter loop aiding also applies to tracking high user dynamics.

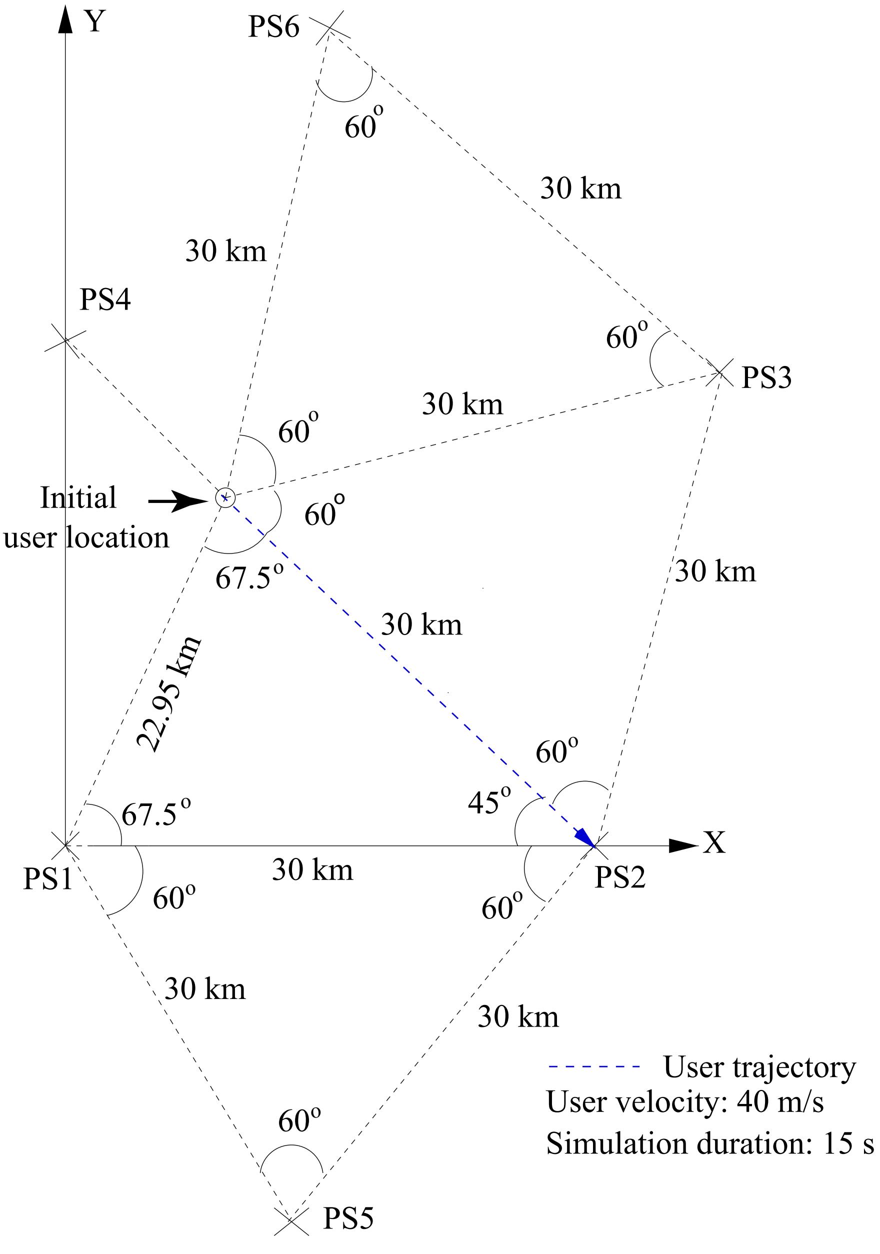


Figure 4.17: Description of User Geometry for Simulation Studies; PS Stands for Ground-Based Pseudolites

4.3 Simulation Studies

Since the vector tracking algorithm is too complex to analyze with real data and, as shown earlier when explaining vector tracking's superiority, LOS geometry is one of the key parameters that has an effect on vector loop performance, simulation studies involving pseudolites used in navigation of a ground vehicle (i.e., two dimensional motion) are considered. This allows easy control of LOS and user geometry which is described below. Note that both scalar and vector architectures are implemented with this user geometry.

4.3.1 Two Dimensional User Geometry

The user geometry shown in Figure 4.17 involves six ground-based pseudolites and a ground vehicle navigating in a straight line towards the second pseudolite. The pseudolites are assumed to have the same signal structure as GPS except that they include only dummy navigation message to simulate the adverse effect of data bit transitions in tracking loop performance. In this context, note that decoding the navigation message is not required since the pseudolite locations are precisely known. As shown in the figure, all six pseudolites are evenly distributed around the user to provide a low dilution of precision ($\backsim$ 2). The user, on the other hand, moves with a constant velocity of 40 m/s, resulting in the following initial Doppler frequencies for the six pseudolites, respectively: [76 210 102 -210 166 -108] Hz. Although 40 m/s is high for ground applications, this velocity is considered in the simulation studies to obtain fairly large Doppler frequencies that are easy to analyze. Since the pseudolites are stationary, the user motion alone causes a slow change in LOS geometry, with the maximum rate of change of the Doppler frequency given as 0.3 Hz/sec. An important advantage of such benign, slowly changing geometry is that it allows one to use the VFLL transfer function models described by Equations 4.50 and 4.53. The noise bandwidths of the VFLL and VDLL transfer function models set the benchmark for the scalar loop noise bandwidths and, thus, serve as a way to systematically compare the two architectures.

4.3.2 Simulation Results

For simulation purposes a high fidelity RF front end simulator has been developed which allows generation of GPS signals with different C/N_0s. Using this simulator, 17 dB-Hz IF data is generated for the user geometry described above. Only wide band

Gaussian thermal noise is added to IF samples and a total of sixty Monte Carlo runs are simulated. Each run is with a different initial state from a random number generator. At 17 dB-Hz the FLL thermal jitter with the cross frequency discriminator, a 2 Hz noise bandwidth and a 10 ms accumulation interval exceeds the rule-of-thumb tracking threshold of [68], implying that longer accumulation intervals or non-coherent averaging is required to maintain lock at this $\mathrm{C/N_0}$. Furthermore, in an effort to create a situation where scalar loops with a 10 ms or a longer accumulation interval will never pull in to the truth, the following initial Doppler errors for pseudolite 1 through pseudolite 6 are considered: [27, 59, 33, -44, 48, -18] Hz. These errors are obtained by projecting a velocity error of 7 m/s on the LOS. Since the frequency pull-in range with a 10 ms accumulation interval is ±50 Hz [67], it is apparent that the second loop will not be able to maintain lock because a smaller accumulation interval will increase the thermal jitter and a longer one narrows the pull-in range. In addition, the fourth and fifth pseudolites can also cross the 50 Hz threshold because of noise. The purpose of this study is to show how vector tracking loops, with identical design parameters as in the scalar tracking loops, can maintain lock at 17 dB-Hz and to validate the previous theoretical analysis of vector loop internal aiding. The afore-mentioned design parameters include nonlinear discriminator types, accumulation intervals, noise bandwidths and loop update rates.

Figure 4.18 shows the Doppler frequency tracking performance of the scalar and vector architectures for a particular Monte Carlo run. The thick solid line in the figure denotes the actual Doppler frequency after RF front end processing. It includes the receiver clock drift which is added to the geometric Doppler frequency during frequency down conversion. The thin black line represents the actual vector loop performance. The thin solid green line shows the performance of the vector tracking loops with constant parameters (i.e., a sub-optimal vector architecture with the decoupled VDLL and VFLL, piecewise constant geometry matrices and constant Kalman gain). It should be noted that in this study $K_{v,\,l}$ of Equation 4.51 and $K_{p,\,l}$ of Equation 4.60 are replaced with K_v and K_p, respectively, for any sub-interval l. The constant Kalman gains (K_v and K_p) are obtained by averaging the steady state Kalman gains over a time interval of 5 sec. The reason for replacing all sub-interval Kalman gains with constant gains is that the Kalman gains are found to remain almost constant due to slowly changing user-pseudolite geometry after initial transients have died down (the maximum rate of change of the Doppler frequency is only 0.3 Hz/s). Each piecewise constant geometry matrix is assumed to be valid for $T_{\mathrm{sub}} = 1$ sec (i.e., for ten update intervals). All underlying assumptions discussed before to select T_{sub} are satisfied.

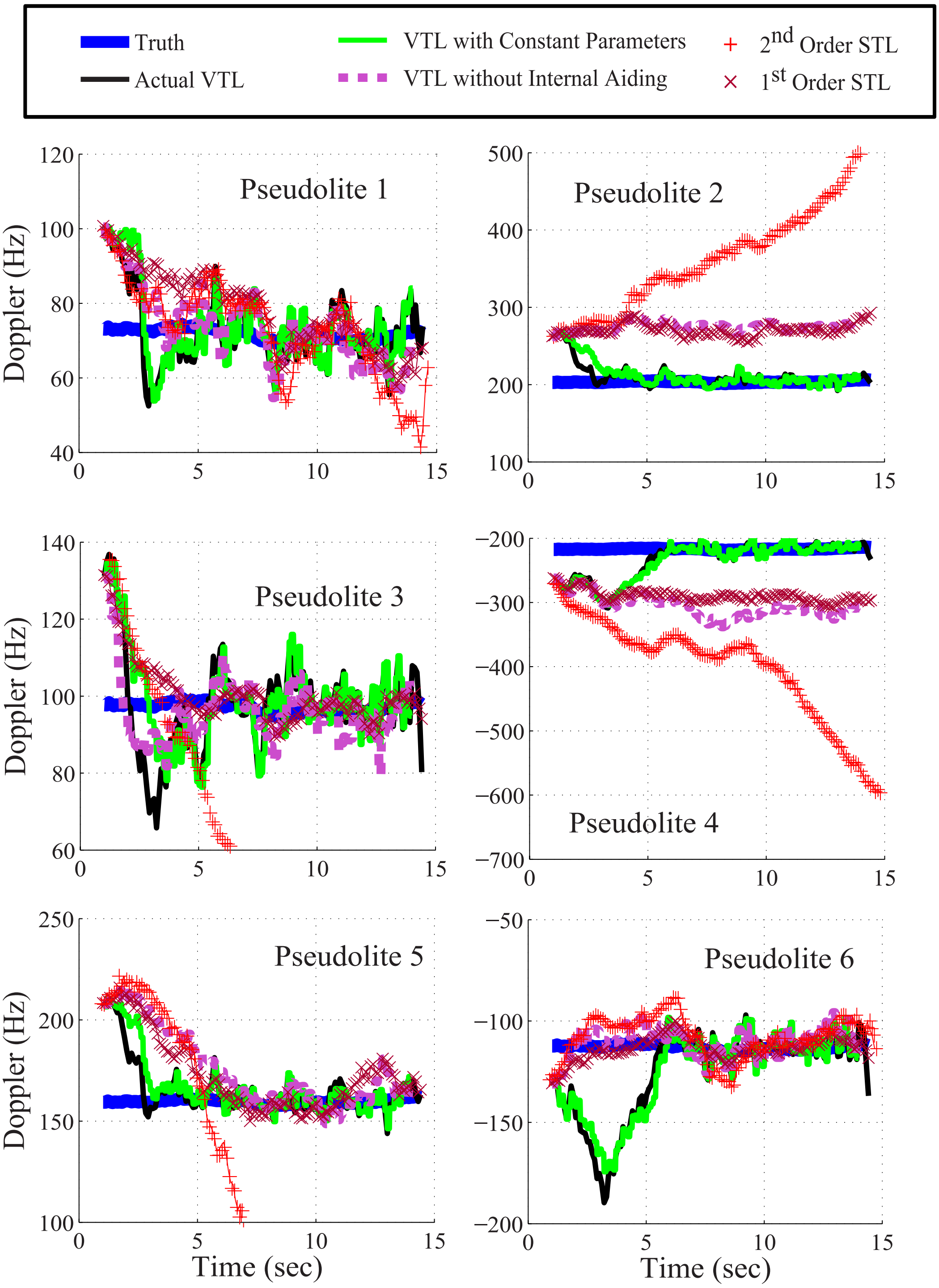


Figure 4.18: Doppler Plots of Scalar and Vector Tracking Loops. VTL: Vector Tracking Loops. STL: Scalar Tracking Loops

The line with 'X' markers represents first order scalar loops and the one with '+' markers illustrates the performance of second order scalar loops. Note that a 10 (20) ms accumulation interval for P (E & L) correlators and a non-coherent averaging time/loop update rate of 0.1 sec to reduce the thermal jitter in the scalar tracking loops are used. The dashed line, on the other hand, shows the performance of the vector architecture without internal aiding (i.e., a sub-optimal architecture where corrections from other loops are dropped by zeroing the off-diagonal matrix entries of Equation 4.37).

The following conclusions are drawn from Figure 4.18:

1. the close proximity of the thin solid green and black lines justifies that, with respect to quasi-stationary geometry, the sub-optimal architecture is a valid approximation to the actual implementation. This validates the use of the sub-optimal architecture as subsequent analysis tool. An advantage of the sub-optimal architecture is that, in the continuous and discrete domains, it can be represented by a set of transfer function matrices (Equations 4.50, 4.53, 4.62 and 4.63), where each set corresponds to one piecewise constant geometry matrix. Thus, instead of a single transfer function for each loop, a 6×6 transfer function matrix for all six loops together is obtained. Consequently, the noise bandwidth of an individual loop in vector tracking is determined by averaging the noise bandwidths of all transfer function elements in the corresponding row of the transfer function matrix. This noise bandwidth is identical with two definitions of vector loop noise bandwidths discussed earlier and valid only when all loops have similar noise statistics as is the case in this simulation. Now considering all transfer function matrices over the simulation duration of 15 sec, 15 noise bandwidths are obtained. With quasi-stationary geometry the 15 values do not differ significantly from each other. For example, the VFLL noise bandwidth of the first loop varies from 0.42 Hz to 0.32 Hz over 10 sec, as illustrated in Figure 4.19. This allows one to determine an effective noise bandwidth of each loop by averaging the 15 values.

   The significance of the vector loop noise bandwidths lies in the fact that they set the benchmark for the scalar loop noise bandwidths which are actually varied in the vicinity of vector loop prescribed values to obtain the best possible performance. This ensures similar scalar and vector loop noise bandwidths while comparing the two architectures. The FLL bandwidths in the continuous domain are selected as [0.5 1.0 0.5 1.0 1.0 0.5] Hz for pseudolites 1 through 6, respec-

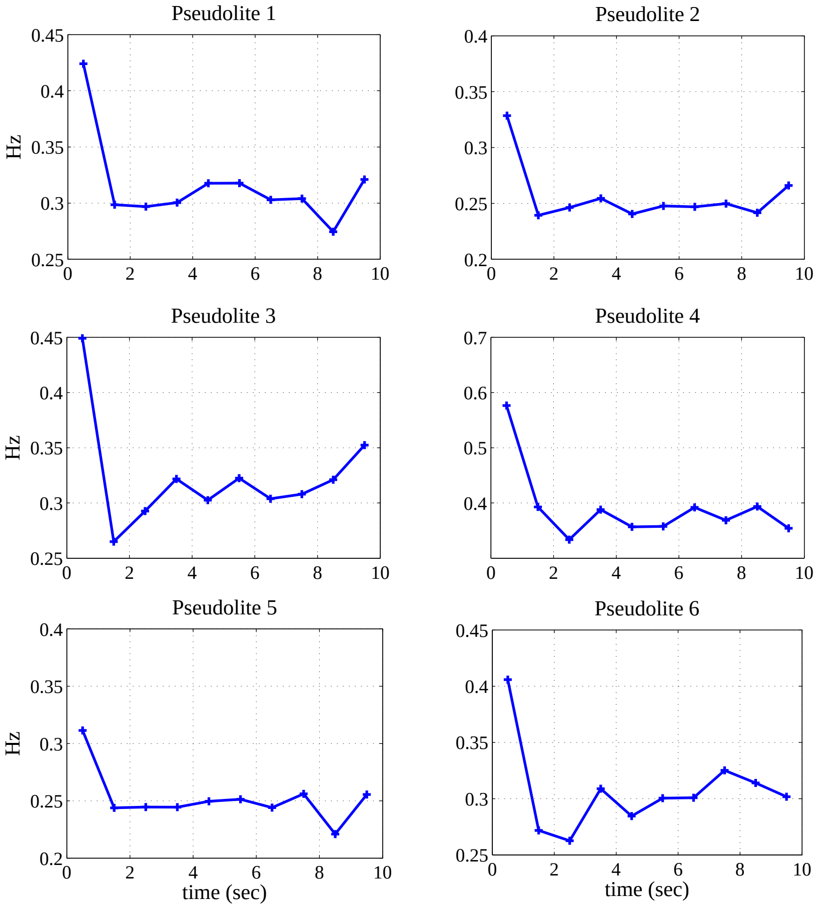


Figure 4.19: Evolution of VFLL Noise Bandwidths in Time for Simulation Geometry

tively. In the discrete domain the FLL noise bandwidths become 0.55, 1.24, 0.55, 1.24, 1.24 and 0.55 for first order loop channels, respectively. They become 0.47, 0.88, 0.47, 0.88, 0.88 and 0.47 for second order loop channels, respectively. It is important to note that the FLLs for channels 1, 3 and 6 have noise bandwidths very close to the corresponding VFLL values whereas the FLLs for the remaining channels have noise bandwidths two to four times the VFLL values. The reason is as follows: It is observed that channels 1, 3 and 6 do not need internal aiding in vector tracking mode in almost all of 60 simulations while channels 2, 4 and 5 need aiding from other channels. So, the FLL noise bandwidths of channels 1, 3 and 6 are set close to vector loop values while those of channels 2, 4 and 5 are set to 1 Hz (i.e., almost two to four times the VFLL values). All DLL bandwidths are also matched to the VDLL bandwidths (~ 0.06 Hz).

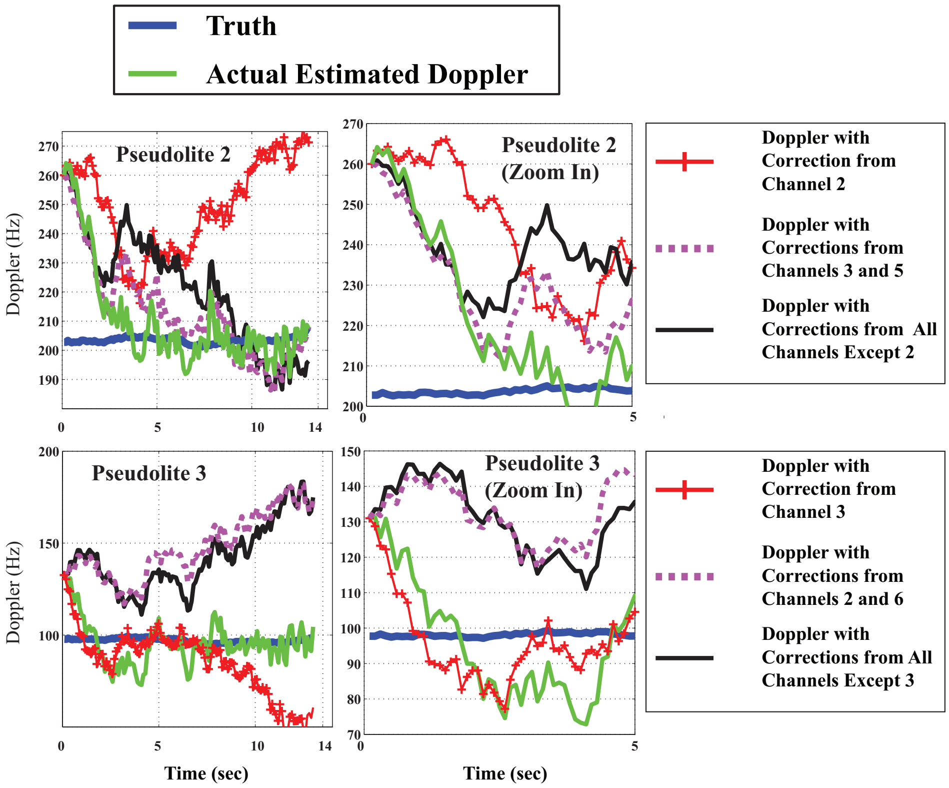
2. The dashed line is equivalent to a first order scalar DLL and FLL except that, unlike the line with 'X' markers, its noise bandwidths are not varied in accordance with vector loop prescribed values. Since it is obtained by turning off inter loop aiding in the sub-optimal parametric model, this implies that the sub-optimal model allows switching back and forth between vector and scalar implementations.

3. The optimal and sub-optimal vector architectures perform better than their scalar equivalents except for pseudolite 6 for which vector loops exhibit large transients before settling down.

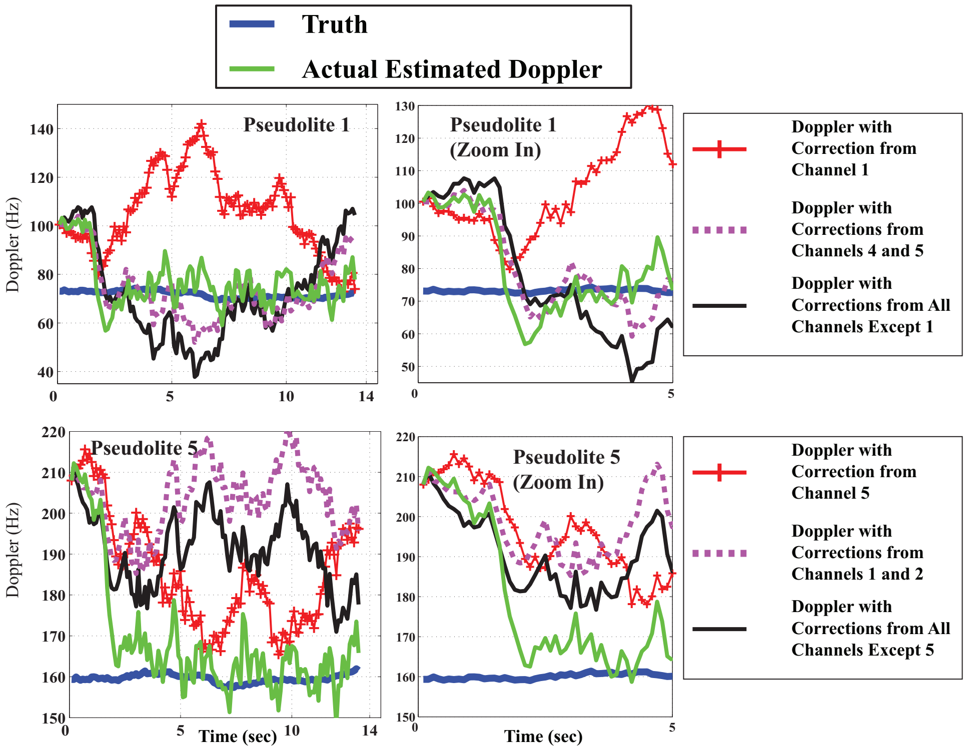
Figure 4.18 shows that both scalar loops fail to converge for pseudolites 2 and 4 because the Doppler frequency errors exceed the nonlinear discriminator pull-in range. Figure 4.20 explains the reasons for vector tracking loop performance with the help of its sub-optimal parametric model described by Equation 4.37. Using this equation, the effect of aiding from different loops is quantitatively shown and, thus, the role of internal aiding in vector tracking is validated with simulations. For simplicity of notation first the equation for the quasi-stationary simulation geometry is rewritten as follows:

$$\begin{aligned}
\underline{\tilde{f}}^{+}_{d,\,k+1} &= \underline{\tilde{f}}^{-}_{d,\,k+1} + \tilde{G}_{d,\,l} K_{v,\,l} \underline{e}_{d,\,k+1} \\
&\text{Since } \underline{\tilde{f}}^{-}_{d,\,k+1} = \underline{\tilde{f}}^{+}_{d,\,k} \text{ within any } l^{\text{th}} \text{ sub-interval and } K_{v,\,l} = K_v \\
\underline{\tilde{f}}^{+}_{d,\,k+1} &= \underline{\tilde{f}}^{+}_{d,\,k} + \tilde{G}_{d,\,l} K_{v} \underline{e}_{d,\,k+1} \\
\underline{\tilde{f}}^{+}_{d,\,k+1} &= \underline{\tilde{f}}^{+}_{d,\,k} + A \underline{e}_{d,\,k+1}
\end{aligned} \tag{4.91}$$

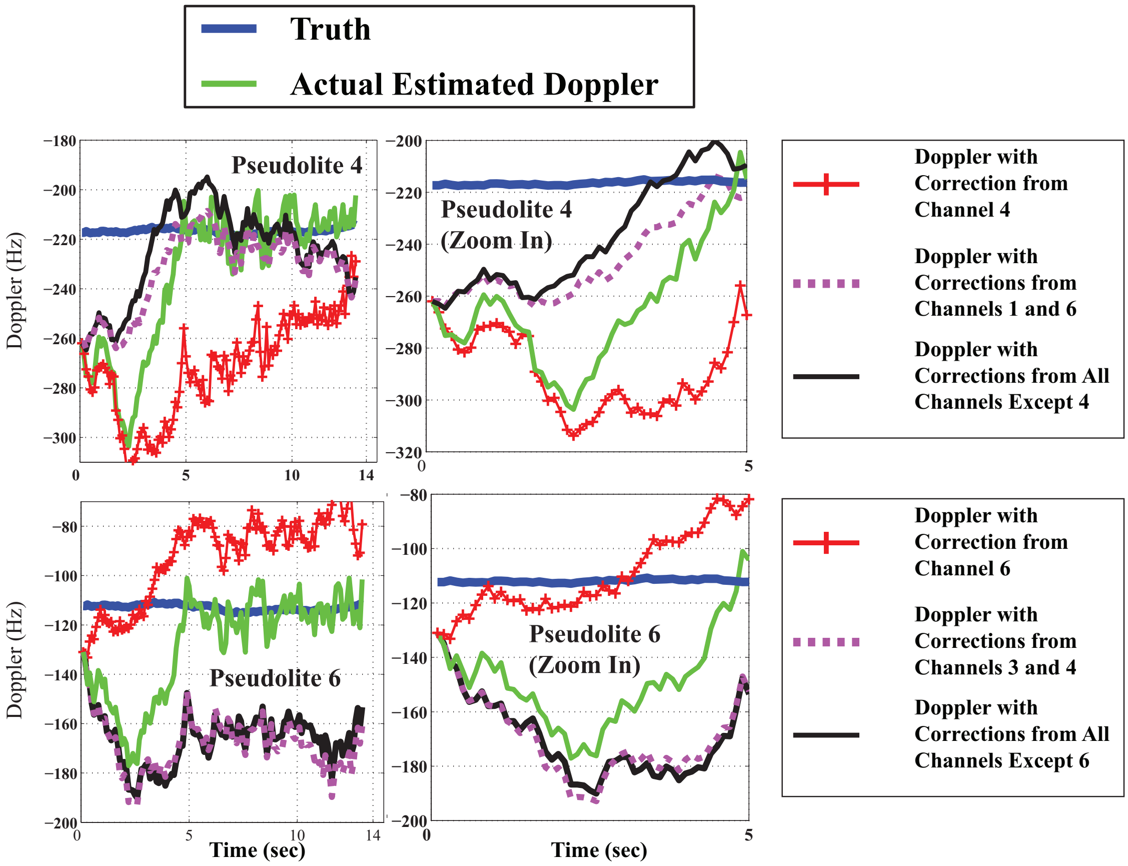
where A is a square matrix of dimension $n \times n$ and n is the number of visible satellites. It is imperative to note that within a sub-interval $\underline{\tilde{f}}^{-}_{d,\,k+1} = \underline{\tilde{f}}^{+}_{d,\,k}$ because the geometry matrix is constant within a given sub-interval and the user velocity is constant. When transitioning from one sub-interval to another, the geometry matrix is assumed to remain unchanged till the first time update of the new sub-interval, thus, still satisfying $\underline{\tilde{f}}^{-}_{d,\,k+1} = \underline{\tilde{f}}^{+}_{d,\,k}$. This approximation is reasonable as the change in $\underline{\tilde{f}}^{-}_{d,\,k+1}$ from $\underline{\tilde{f}}^{+}_{d,\,k}$ due to the change in the geometry matrix during sub-interval transitions is negligible compared to the correction from the measurement update. The new geometry matrix is used from the first measurement update of the new sub-interval i.e., with the term $\tilde{G}_{d,\,l} K_v \underline{e}_{d,\,k+1}$.



(a) Pseudolites 2 and 3



(b) Pseudolites 1 and 5



(c) Pseudolites 4 and 6

Figure 4.20: Internal Aiding in Vector Tracking Loops: (a) Pseudolites 2 and 3; (b) Pseudolites 1 and 5; (c) Pseudolites 4 and 6

For any pseudolite i the Doppler frequency is given by

$$\underline{\tilde{f}}^{+}_{d,\,k+1}(i) = \underline{\tilde{f}}^{+}_{d,\,k}(i) + \sum_{j}^{6} A(i,j)\underline{e}_{d,k+1}(j)$$

$$\underline{\tilde{f}}^{+}_{d,\,k+1}(i) = \underline{\tilde{f}}^{+}_{d,\,k}(i) + \sum_{j}^{6} \Delta f_{ij} \tag{4.92}$$

Δf_{ij} is referred to as the correction term of i^{th} loop contributed by j^{th} loop. The actual estimated Doppler frequency which is generated by combining corrections from all loops is depicted in Figure 4.20 (a) - (c) by the thin solid green line. The line with '+' markers would be the estimated Doppler frequency if only the contribution of an individual loop to correcting its own error was considered and those of others were ignored (i.e., Δf_{ii} for pseudolite i in Equation 4.92 is only considered). The black line, on the other hand, results from considering the total corrections of all loops except the one being estimated. The dashed line shows the estimated Doppler when considering corrections from only those two loops that are most correlated with the loop under interest. The two most correlated loops are determined by the two elements of a row of $\tilde{G}_{d,\,l}K_v$ which are second and third largest after its diagonal entry. Mathematically, for the i^{th} pseudolite the solid green and black lines and the line with '+' markers are obtained by the following equations, respectively

$$\begin{aligned}
\tilde{\underline{f}}^{+}_{d,\,k+1}(i) &= \tilde{\underline{f}}^{+}_{d,\,k}(i) + \sum_{j=1}^{6} \Delta f_{ij} \\
\tilde{\underline{f}}^{+}_{d,\,k+1}(i) &= \tilde{\underline{f}}^{+}_{d,\,k}(i) + \sum_{j=1,\,j\neq i}^{6} \Delta f_{ij} \\
\tilde{\underline{f}}^{+}_{d,\,k+1}(i) &= \tilde{\underline{f}}^{+}_{d,\,k}(i) + \Delta f_{ii}
\end{aligned}$$

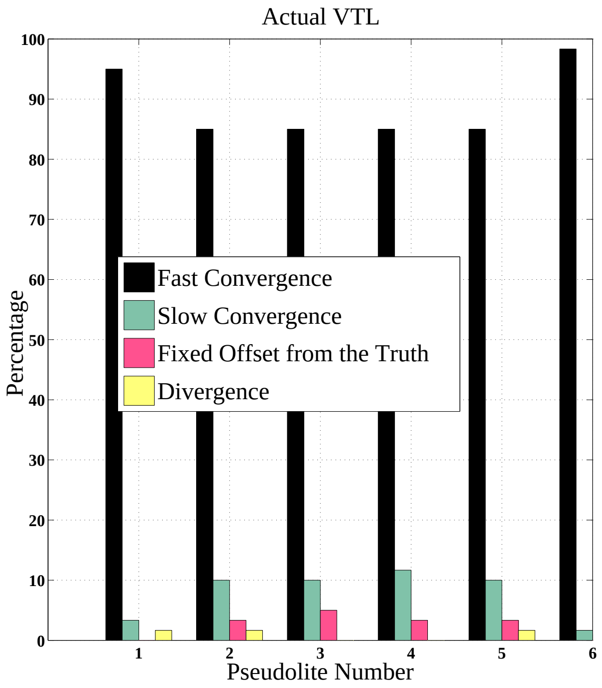
Since $\Delta f_{ii} = f_{ii} = c\underline{e}_{d,k+1}(i)$, where $0 < c < 1$ (Equation 4.90), all the flat segments of the line with '+' markers indicate that the corresponding $\underline{e}_{d,\,k+1}(i)$ is negligible. Further, as $\Delta f_{ij} \propto \underline{e}_{d,\,k+1}(j)$, this implies that the loop not only has negligible self-corrections in those segments, but also its contributions to other loops are then insignificant. This interpretation will be used in the explanation below. If l^{th} and m^{th} pseudolites are most correlated with i^{th} pseudolite, then the dashed line follows the equation given below

$$\tilde{\underline{f}}^{+}_{d,\,k+1}(i) = \tilde{\underline{f}}^{+}_{d,\,k}(i) + \sum_{j=l,\,j=m} \Delta f_{ij}$$

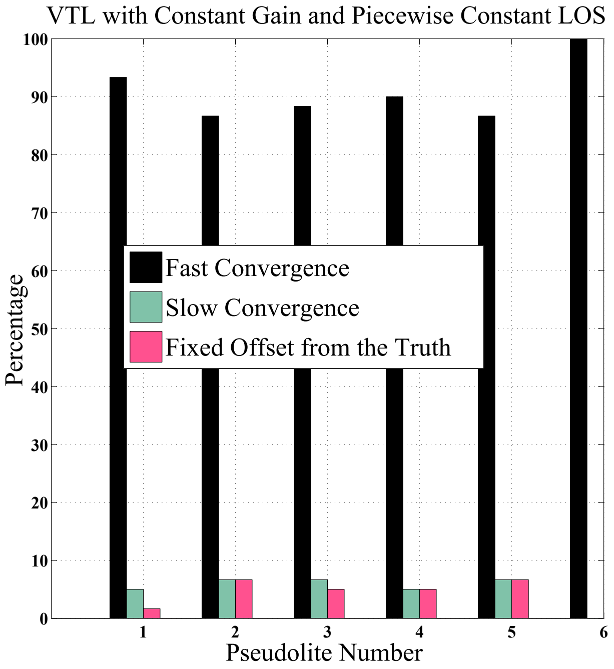
It is imperative to observe that none of these lines except the one corresponding to the actual estimated Doppler frequency converge to the true value as they apply only partial corrections. The significance of these lines, in particular, lies during the transient period. Since all of them start with the same initial guess, the proximity of the solid green line to a particular line during a transient illustrates which loop is instrumental in driving the actual estimated Doppler (solid green line) to the truth. For example, the solid green, black and dashed lines for pseudolite 2 are nearly coincident with each other which implies that the loop is aided by the ones that are most correlated with it (i.e. pseudolites 3 and 5). Since pseudolite 5 generates almost negligible self-corrections in the beginning, as reflected by the initial flat portions of the line with '+' markers for this pseudolite (i.e. Δf_{55} and $\underline{e}_{d,k+1}(5) \approx 0$), it can be concluded that most of the corrections for pseudolite 2 are generated by pseudolite 3 which also shows rapid convergence. Thus, internal aiding drives the Doppler frequency of the second loop within its pull-in range. Later, the second loop aids itself to ensure fast convergence. Likewise, pseudolite 1 helps pseudolite 5 (notice the jump in its dashed line around the same time the line with '+' markers of pseudolite 1 starts to move downward).

The adverse effect of internal aiding is evident when the diverging 4th element of $\underline{e}_{d,k+1}$ (see its effect on the line with '+' markers for pseudolite 4) propagates into pseudolite 6 and delays its convergence. This explains why in Figure 4.18 vector loops show large transients with pseudolite 6. Further, note that the diverging 4th element of $\underline{e}_{d,k+1}$ prevents scalar loops from pulling in as it causes the Doppler error to cross the 50 Hz threshold while at the same time vector loops pull in due to internal aiding. Figure 4.20, therefore, supports the theoretical analysis presented with the help of Figures 4.15 and 4.16. It also clearly illustrates the drawback of internal aiding in vector tracking loop performance.

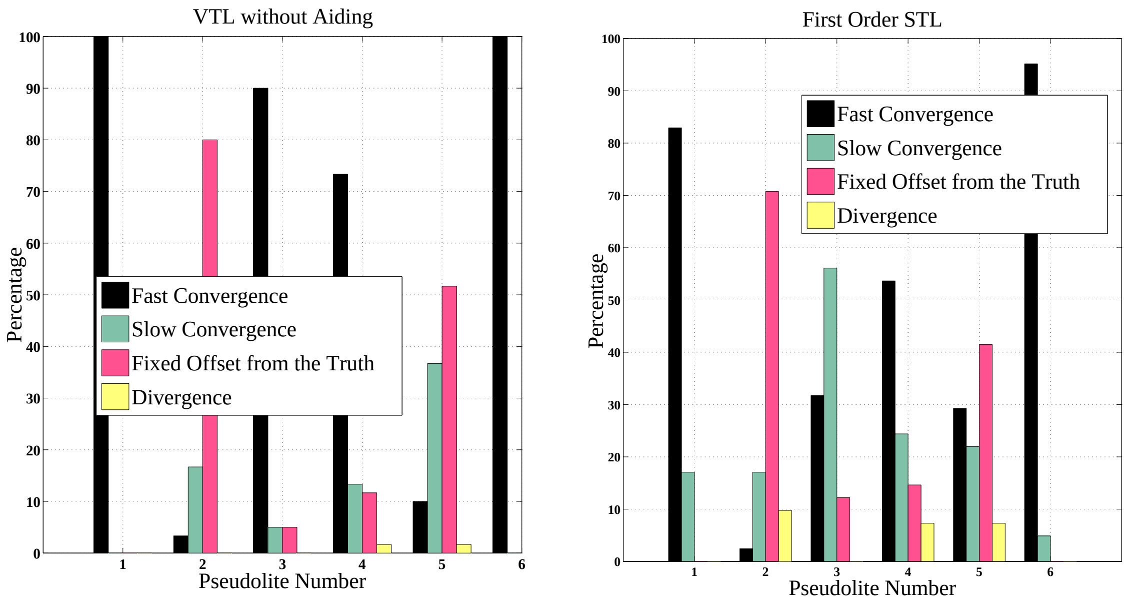
Next, the histogram plots of Figure 4.21 summarize the Doppler tracking performance of the scalar and vector architectures with sixty Monte Carlo runs. As for the legend, "fast convergence" implies convergence of the estimated Doppler frequency to the true Doppler frequency within the first five senconds. "Fixed offset" means a fixed bias from the truth as illustrated by the first order scalar loop performance of pseudolites 2 and 4 in Figure 4.18. "Divergence" denotes monotonically increasing errors similar to the ones for the second order scalar loops in pseudolites 2 and 4 in Figure 4.18. The histogram plots reinforce earlier observations that the optimal and sub-optimal vector architectures are both significantly better than the scalar loops when the initial Doppler errors are large. Although the performance of the optimal and sub-optimal architectures is almost comparable, the optimal architecture results in divergence in a small number of runs but the sub-optimal architecture does not. While the actual reason behind such inconsistency is unknown, this may be caused by weakly coupled VDLL and VFLL in the optimal architecture, which results in slighly higher chance of error propagation among channels. In the sub-optimal architecture VDLL and VFLL are decoupled. It is noteworthy that the first order scalar loops perform better than the second order scalar loops. A probable explanation for this is that the transfer function elements of the vector loops are of first order. So the vector loop prescribed noise bandwidths may be better matched to the first order scalar loops than to the second order scalar loops. It is found that the number of divergence in the scalar loops is reduced if the noise bandwidths are increased.



(a) Actual VTL

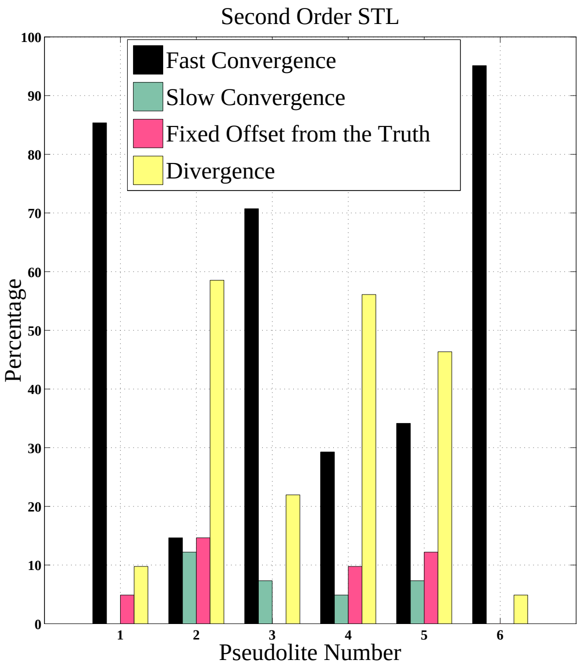


(b) VTL with Constant Kalman Gain and Piecewise Constant LOS



(c) VTL without Aiding

(d) First Order STL



(e) Second Order STL

Figure 4.21: Histogram Plots; VTL: Vector Tracking Loops; STL: Scalar Tracking Loops

4.4 Preliminary Guidelines for Achieving Better Performance of Scalar Loops

The theoretical analysis and simulation results discussed before illuminate two key features of the vector architecture which can be leveraged to design a more robust scalar tracking loop. They are inter-satellite or internal aiding of channels in vector tracking mode and vector loops' ability to adjust noise bandwidths with input noise levels, user dynamics and number of visible satellites.

As demostrated earlier in Chapter 1, the conventional scalar receiver architecture loses track of signals in low $\mathrm{C/N_0}$ and/or high dynamic environments. Among various approaches used to deal with the problem of frequent signal outage, adaptive tightly coupled GNSS/INS integration is one where Doppler information derived from the INS is used to aid GNSS tracking loops [79]. The tracking loop bandwidths, on the other hand, are varied adaptively depending on $\mathrm{C/N_0}$ to facilitate better INS calibration. Further, to ensure that the GNSS/INS integration Kalman filter is not destabilized by colored noise, the Kalman filter noise bandwidth is tuned to the tracking loop noise bandwidth by appropriate design. This is done by scaling up the Kalman filter measurement noise covariance using tracking loop bandwidth when the bandwidth is below a certain threshold. Although not detailed in [79], the threshold bandwidth can be selected to correspond to the filter measurement update rate.

The guidelines proposed in this thesis for achieving better performance of the scalar architecture are akin to tight GNSS/INS integration in that they attempt to replicate the effect of vector loop internal aiding in the scalar architecture. The proposed method, however, offers a departure from the usual approach that selects the tracking loop noise bandwidth somewhat heuristically (wide at good $\mathrm{C/N_0}$ and narrow at low $\mathrm{C/N_0}$). It proposes a more systematic way of selecting the bandwidth. This is achieved by setting the vector loop-derived noise bandwidth as the corresponding scalar loop bandwidth for a given noise level, user dynamics and tracking loop update rate.

Recall from the discussion under vector loop transfer function models that for a given level of user dynamics and measurement noise, the vector loop transfer function models can be derived over a number of sub-intervals in a given time interval. The duration of each sub-interval is determined using some criteria for offline analysis. For real-time implementation the duration of a transfer function model can be a few seconds. From these transfer function models vector loop noise bandwidths can be obtained using the second definition of noise bandwidths. For a given level of measurement noise and dynamic uncertainties, the noise bandwidth is a function of the order of vector loop transfer function elements (or the number of independent states (velocity, acceleration, jerk) estimated), measurement update rate and user-satellite geometry.

The following parameters of the scalar tracking loops are proposed to be designed with the help of vector loop transfer function models. The FLL bandwidths are adjusted corresponding to the FLL update rate which is set as 20 ms. The FLL filter order can also be set equal to that of the corresponding diagonal element of the vector loop transfer function matrix (after canceling closely located poles and zeros). The DLLs are generally aided by the FLLs. So, their bandwidths are set inside the tracking algorithm. The selection of FLL bandwidths can be carried out in the scalar loop navigation filter at each filter measurement update and fed back to the tracking loops. Since this is done at a relatively low rate (few seconds), individual scalar loop-estimated C/N_0 should be used to scale the scalar loop bandwidth if C/N_0 changes between two successive updates from the navigation filter.

The preliminary guidelines for designing a scalar loop which possesses some of the attractive features of vector loops and not all the complexity are presented below. This leverages vector loop transfer function models developed earlier:

1. The vector loop Kalman gain and noise bandwidths are calculated in real time using a separate algorithm inside the scalar loop navigation filter routine. A rough description of the algorithm inputs and outputs is as follows: At each filter update interval the algorithm inputs the geometry matrix, process noise covariance (from the expected level of dynamics) and receiver estimated C/N_0s for calculating the measurement noise covariance. Then it computes the steady state VFLL Kalman gain corresponding to the FLL update rate. This Kalman gain is used in the VFLL transfer function model. The initial error covariance is assumed to be the identity matrix for computing the Kalman gain of the first VFLL transfer function model and equal to the steady state error covariance of the previous model after that. Since C/N_0s and geometry matrix are available only at the filter measurement update rate which is lower than the FLL update rate, the computation of VFLL Kalman gain assumes that C/N_0s and geometry matrix do not change between two successive filter updates. Thus, the time interval over which the VFLL transfer function model is valid is automatically determined by the scalar loop navigation filter measurement update rate. The algorithm outputs the FLL filter order as determined from the VFLL transfer function matrix and

the vector loop noise bandwidths corresponding to the FLL update rate. The navigation filter sends out this information to the tracking loops. The previous analysis of vector loop noise bandwidths suggests that for an update rate of 20 ms the continuous and discrete bandwidths are almost similar (within 1.5 times of each other). As a result, it is sufficient to calculate the continuous-domain noise bandwidths instead of complex discrete-domain bandwidths.

2. The navigation filter also predicts the Doppler frequencies by projecting the satellite and user velocity and acceleration on the LOS vectors and feeds back the predicted Dopplers and Doppler rates to individual scalar tracking loops as aiding. The Doppler rates are used to predict the Doppler frequencies used for aiding. When there is a change in user dynamics, good C/N_0 channels are likely to track it. If poor C/N_0 channels are unable to track the change, they may lose lock but can be expected to restore tracking after the next Doppler update from the navigation filter. The Doppler information can also be derived from an external source to better track user dynamics, as is done in tightly coupled GNSS/INS integration.

3. To avoid cascading problems i.e., positive feedback caused by destabilized navigation Kalman filter, the filter noise bandwidths are matched to the scalar tracking loop bandwidths using the method of [79].

It is instructive to note that the above modifications can be carried out on receivers that allow tracking bandwidths and NCO inputs to be controlled externally. This should not be too costly as most receiver tracking loops are now implemented in software. While it can be expected that the above designed scalar loops will be less complex in terms of implementation and yet will offer performance comparable with vector loops to some extent, the latter will always have an edge. This is because, unlike vector loops, individual scalar loops will handle any residual error on their own. In addition, the measurement update rate of the scalar loop navigation filter is lower than that of the vector loops, resulting in less accurate Doppler information between two updates.

4.5 Conclusion

This chapter presented a methodology for analyzing vector tracking loops and comparing their performance to scalar architectures. Based on this comparison, the following conclusions are reached: First, vector tracking is superior to scalar tracking because of internal aiding among loops which enables better performance of the vector tracking architecture. This implies that Doppler aiding can be used to achieve similar performance with scalar loops. Furthermore, vector loop noise bandwidths as derived from their transfer function model can also be a good starting point for designing scalar loops with equivalent performance. Based on these findings, preliminary guidelines to achieve comparable scalar and vector loop performance are developed. Second, internal aiding, in addition to being responsible for good dynamic/noise rejection performance, points to an important pitfall of such a coupled architecture; an error developed in one loop can easily propagate into others and corrupt them. Such propagation of errors is, to some extent, apparent in some of the simulation results where the scalar loops settle down in the vicinity of the true value, but individual vector tracking loops have wider variations or settle down late due to errors in others. Therefore, an extensive study of the integrity risk posed by vector tracking is of paramount importance and will be the focus of the next two chapters of this thesis.

Chapter 5

Integrity Analysis of Vector Tracking Architecture

5.1 Introduction

While the vector tracking architecture has proved to be an effective solution for dealing with the challenge of frequent loss or attenuation of GPS signals in urban areas, issues with its fault detection capability or integrity monitoring have yet to be explored in great depth. The integrity studies of the vector architecture in this chapter, therefore, aim at making contributions towards enabling its use in high-integrity navigation systems which lie in the heart of all safety and liability-critical applications. The use of vector tracking architecture with integrity monitoring will hold promise in applications such as driver-assist systems [80], law enforcement operations, road tolling/congestion charging schemes [81], etc..

In broad terms integrity relates to the level of trust placed in a system. It is an important navigation performance parameter in safety and liability-critical operations. Integrity is quantitatively represented by three sub-parameters: Integrity risk, alert limits and time-to-alert. Integrity risk is the the probability of occurring an event when a navigation system will generate misleading information (MI). In more concrete terms, this is an event where the system generates a solution with larger than acceptable position error without issuing a timely alert. Depending on the severity of consequences, it is also called hazardously misleading information (HMI) in certain applications. Integrity risk is mathematically expressed by the probability of HMI (or MI). The second sub-parameter is called alert limits which are of two types: Vertical alert limits (VAL) and horizontal alert limits (HAL). Alert limits are maximum position errors that are permissible in a given application from a safety or liability standpoint. The third sub-parameter time-to-alert is the maximum allowable time between the instant when position error exceeds either or both alert limits and the instant when an alarm is issued to the user. Once these sub-parameters are specified for an application, an intergrity monitoring system can be designed to continuously check system anomalies. The monitoring system also outputs position error bounds called protection levels that protect the actual position error with a certain probablity and must remain below the alert limits for integrity monitoring to be available. Among different architectures that have been proposed over the years, receiver autonomous integrity monitoring (RAIM) is a solely receiver-based integrity monitoring technique that uses redundant pseudorange measurements to satellites to perform consistency checks.

Although there is an extensive body of knowledge on RAIM for scalar architectures [52]- [55], it is not directly applicable to the vector architecture. This is because, as shown in the previous chapter, satellites aid each other in vector tracking mode, meaning an individual satellite tracking loop not only applies correction to its own error, but also to those of other loops. Such inter-satellite aiding or internal aiding, in addition to being responsible for vector loops' superior performance, allows easy propagation of errors (and similarly faults) among loops. Consequently, the pseudorange measurements output from a vector tracking receiver are all affected by a failure in one channel and, thus, violate the basic single fault assumption of the traditional or scalar loop-based RAIM algorithms. Such propagation of errors/faults among the pseudorange measurements of a vector tracking receiver is demonstrated in Figure 5.1. The fault and the intermediate frequency data for this purpose are generated using a high fidelity receiver front end simulator. The figure shows the pseudorange measurement noise in four channels of a vector tracking receiver before and after a ramp error of slope 1 m/sec is injected into channel 2 at t = 20 sec. It is clearly evident from the figure that the fault not only affects the measurements in channel 2, but also manifests itself in the measurements of other channels.

While one can argue that independent scalar loop-based RAIM algorithms can run alongside vector loops, this architecture will not work under weak signal environments. In these environments scalar loops, unlike their vector counterpart, easily lose the ability to track GPS signals. Further, with multiple GNSS Systems on the horizon, the vector architecture holds potential for tracking very weak satellite signals by fusing multi-GNSS signals. This is attributed to its inherent ability to improve track-

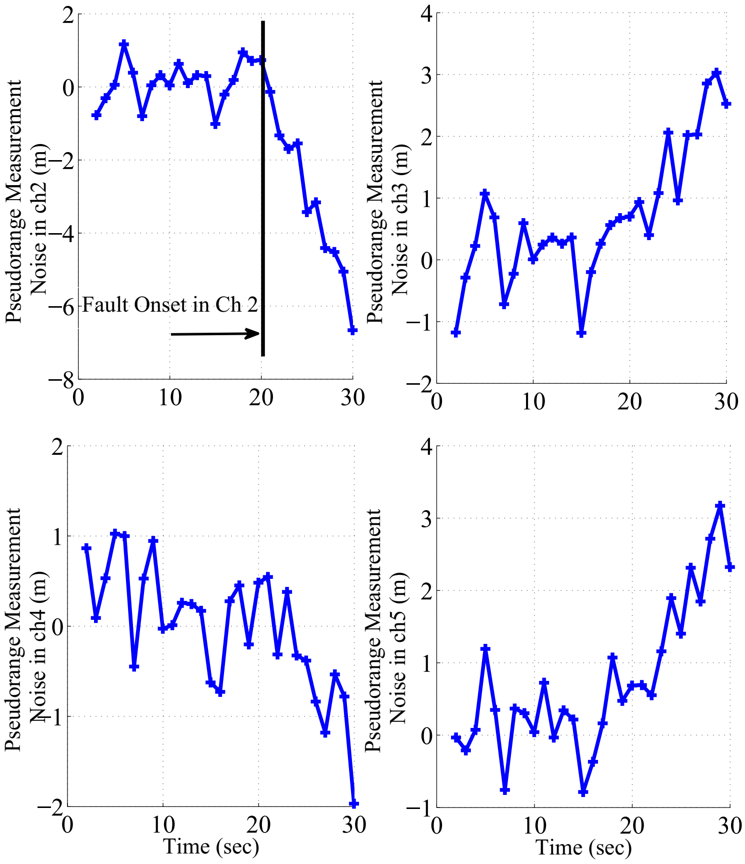


Figure 5.1: Propagation of Fault among Channels

ing thresholds adaptively when more satellites are visible, as concluded in Chapter 4. Hence, in view of these issues, studies of integrity risk in vector tracking are warranted as the literature offers meagre information about it [12], [57].

The remaining chapter is organized as follows: First the objectives of this chapter are highlighted. After that the scalar loop RAIM algorithm which will be used as a benchmark later in the chapter is discussed. Then, vector loop RAIM algorithms for coherent and non-coherent architectures are presented. Following this, the vector loop RAIM algorithm for the non-coherent architecture is validated with simulation studies. Finally, a summary and concluding remarks close the chapter.

5.2 Objectives

The objectives of this chapter are twofold. First, it is to design a RAIM algorithm for vector tracking. Second, it is to assess the performance of vector loop RAIM by comparing it to a traditional RAIM (scalar loop RAIM) with simulation studies. With respect to the first objective, the approach in this thesis involves the use of correlator or nonlinear discriminator outputs to form test statistics for the vector loop RAIM algorithms. The advantage of this approach is that the discriminator outputs (for non-coherent architecture) or the code phase errors extracted from the correlator outputs (for coherent architecture) directly translate to the pseudorange residuals. This is because they are proportional to the code phase alignment errors of the replica code with the received code. While the received code corresponds to the actual pseudorange measurements, the phase of a replica code in vector tracking mode is adjusted based on the predicted pseudorange output of the vector loop navigation filter. In addition, similar to the pseudorange measurements of scalar architectures, the assumption of a single failure applies to the vector loops at the correlator or discriminator level. Therefore, the test statistics of the vector loop RAIM algorithms can be formed by appropriately using the discriminator or correlator outputs or, in other words, from the difference between the actual and the receiver-predicted pseudorange measurements. Moreover, it will be shown that the protection levels can be calculated for the federated coherent and non-coherent vector architectures in a way similar to the scalar architectures. This implies that schemes equivalent to the standard scalar loop RAIM [52] - [54] will hold true for the vector architecture when implemented with the discriminator or correlator outputs. In this context, it is important to note that [12] uses the discriminator outputs as inputs to its vector loop RAIM algorithm. But neither does it point out the drawbacks of vector loop pseudorange measurements, nor does it address the challenges associated with using the discriminator outputs. Further, it lacks a thorough performance comparison of scalar and vector loop RAIMs.

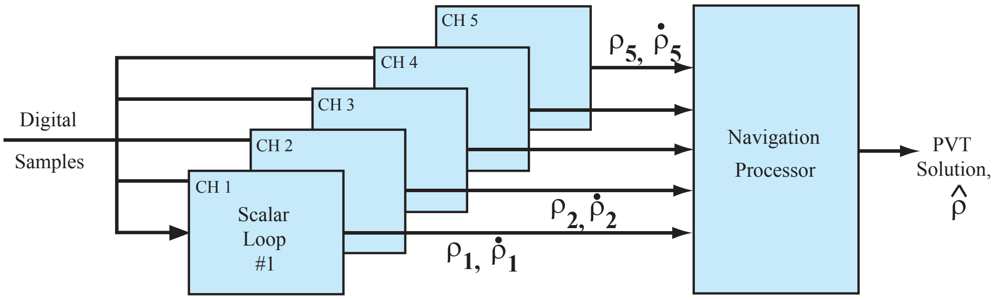
With respect to the second objective, this chapter focuses on identifying conditions under which scalar and vector loop RAIM algorithms can be studied systematically and comparably. In this regard, the tracking loop noise statistics are related to those of the pseudorange noise used in scalar loop RAIM because the tracking loop noise statistics are used in the vector loop RAIM. Correlator noise statistics are also derived in Appendix A removing the white noise assumption [63] of the bandlimited pre-correlation noise (when the two-sided pre-correlation bandwidth is less than the sampling frequency). Next, the calculation of protection levels is modified to account for Kalman filter implementations as opposed to least squares estimation of user PVT. It should be noted that the protection level calculation with the Kalman filter implementation presents a challenge that past faults affect the present state estimate, unlike least squares estimation. This is compounded by the fact that user position is updated at a *slower* rate than the user velocity to perform integrity monitoring in the vector architecture with relatively noisy discriminator outputs. The discriminator outputs need to be averaged over a few seconds to reduce noise. All this results in faults in the pseudorange rate measurements contributing to the position error even when the position is not updated, thus making the computation of protection levels complicated. It is addressed in detail later in this chapter. Then, for given probabilities of false alarm and missed detection (or integrity risk) the performance of both non-coherent scalar and vector algorithms is studied with different values of the following parameters: Ramp type faults and C/N_0. To compare performance, the fault detection times since the onset of the fault and protection levels of the two tracking architectures are noted. Finally, based on the performance results of simulation studies, conclusions are drawn on the integrity monitoring capability of vector tracking as compared to its scalar counterpart. Thus, this chapter lays the groundwork for further research on vector loop integrity monitoring algothims. In the next chapter the vector loop RAIM algorithm will be validated with GPS measurements.

5.3 Scalar and Vector Loop RAIM Algorithms

In this chapter and the next chapter, the scalar loop RAIM algorithm is used as a benchmark against which vector loop RAIM performance is evaluated. Therefore, first, the scalar loop RAIM algorithm is described. Following the scalar loop RAIM, a discussion of the vector loop RAIM algorithms is provided.

5.3.1 Scalar Loop RAIM

The RAIM algorithm performs the following functions. First, it employs a fault detection test to detect system anomalies. Second, it provides vertical and horizontal position error bounds called vertical protection level (VPL) and horizontal protection level (HPL), respectively. When no fault is present, the VPL and HPL guarantee that true position errors are bounded with a certain probability. The protection levels are determined from the integrity risk and must be bounded by the corresponding alert limits for RAIM to be considered available. The time-to-alert requirement also needs to be met by the RAIM algorithm. In what follows, pseudorange error models; fault detection methods employed in this thesis; and methods to calculate vertical and horizontal position errors and protection levels pertaining to EKF-based navigation filter



Scalar Receiver Architecture

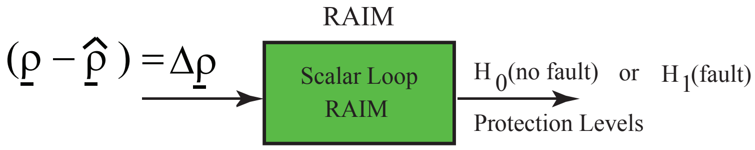


Figure 5.2: Scalar Loop RAIM

are presented. While they are discussed in the context of scalar loop RAIM, it will be justified later that the same formulations generally hold for the vector architecture if the vector loop RAIM inputs are suitably chosen. Only major changes required for the vector architecture will be addressed when discussing the vector loop RAIM.

5.3.1.1 Pseudorange Error Models

The linearized pseudorange measurement vector or pseudorange residual vector ($\Delta\underline{\rho}_k$) at time epoch k is defined as

$$\Delta\underline{\rho}_k = H_k(X_{p_k} - \tilde{X}^-_{p_k}) + \underline{\epsilon}_k \tag{5.1}$$

where $\Delta\underline{\rho}_k = \underline{\rho}_k$ - $\underline{\hat{\rho}}_k$. $\underline{\rho}_k$ is the pseudorange measurement vector output by the scalar tracking loops and $\underline{\hat{\rho}}_k$ is the predicted pseudorange measurement vector of the navigation filter. They are depicted in Figure 5.2. H_k is the pseudorange measurement model matrix. X_{p_k} is the 4×1 vector of true user position and clock bias and $\tilde{X}^-_{p_k}$ is an *a priori* estimate of X_{p_k}. $\underline{\epsilon}_k$ is the $n \times 1$ vector of pseudorange error with mean zero under fault-free conditions and error covariance W_k. Each element of $\underline{\epsilon}_k$ corresponds to one satellite and comprises unmodeled propagation delays (i.e., unmodeled ionospheric and troposheric errors along the LOS), residual satellite clock and ephemeris errors, multipath and receiver noise. Typical error budgets for stand-alone single-frequency GPS receivers (without external differential corrections) can be found in [76] while the same for differentially corrected receivers can be found in [82]. The approach to representing unmodeled errors as uncorrelated Gaussian distributions works well for a snapshot fault detection test statistic. It also works for the test statistic that is formed with past and present measurement residuals provided measurement samples are sufficiently spaced apart to treat time-correlated errors as uncorrelated.

Unmodeled atmospheric delays, errors in broadcast satellite ephemeris and clock can also be modeled based on prior knowledge and estimated along with user position and clock bias using the approach of [83]. Accounting for these unmodeled time-correlated errors is particularly helpful when forming the vector loop fault detection test statistic using past and present pseudorange residuals derived from the discriminator outputs. This is because in the vector architecture successive measurements cannot be adequately spaced apart to counteract the effect of time-correlated errors as the user position has to be updated within few seconds to maintain code tracking. Furthermore, mitigating the effect of multipath at the navigation filter level with short measurement update intervals presents its own challenge. The measurement differencing method of [11] cannot be adopted to deal with multipath because, with this method, the statistical independence of test statistic and position error cannot be established. The method of state augmentation to deal with multipath is also not desirable as the number of states will be more than the number of measurements. It is, therefore, imperative to mitigate multipath as much as possible using advanced antenna design or enhanced receiver design [84] to perform integrity monitoring with vector loops. Time-correlated errors are assumed to be negligible in this thesis, however. Analyzing vector loop RAIM performance in the presence of time-correlated errors will be part of future work. Thus, W_k in this thesis comprises only receiver noise variances.

When comparing the scalar and vector architectures, the variances and models of all the above error sources except that of receiver noise remain the same. The receiver noise differs in the two architectures because of the fundamental difference in the way they implement their signal tracking algorithms. Therefore, to ensure a systematic performance comparison of the two architectures, pseudorange noise statistics from $\mathrm{C/N_0}$ and various other tracking loop parameters are derived. Thus, this method allows the tracking loop noises to directly relate to their pseudorange counterparts, as described below.

In a scalar receiver architecture independent code tracking loops are used to align the locally generated replica C/A-codes with the corresponding received C/A-codes. Such alignments allow the receiver to obtain coarse and fine estimates of the signal transmit time from the decoded navigation message and the replica code phase output of the tracking loops, respectively [62]. Then, from the difference between the signal receive and transmit times it generates the pseudorange measurements. This has been discussed in Chapter 2. Assuming zero error in the time derived from the navigation message, the standard deviation of the pseudorange noise is given by the code phase jitter of the tracking loop in the units of meters. Similar to the carrier phase jitter defined in [64], the code phase jitter is expressed in chips as

$$\sigma_{\tau_{i,k}} = \sqrt{\sigma^2_{w_{o_{i,k}}} + \sigma^2_{c_{o_{i,k}}}} + \frac{\sigma_{dyn_{i,k}}}{3} \tag{5.2}$$

where i is the satellite number, k is the time epoch. σ_{w_o} is the jitter on the output of the code tracking loop due to white noise, σ_{c_o} is the jitter due to colored noises and σ_{dyn}, also known as the dynamic stress error, represents the code phase tracking error from abrupt LOS dynamics. Each of these terms specific to the scalar loop architecture used in this thesis is described below.

White Noise Jitter

In this thesis a non-coherent scalar tracking architecture is implemented. It uses the early-minus-late power discriminator to estimate the code phase error (τ) and the cross discriminator to estimate the residual Doppler shift (Δf_d) [68]. As for the tracking loop filters, a zero order code tracking loop or DLL loop filter and a second order FLL loop filter are considered in this chapter.

Recall from Chapter 2 that the tracking loop generated replica code and carrier signals are correlated with incoming signals to generate three correlator pairs (I_E, Q_E, I_P, Q_P, I_L, Q_L), where I and Q stand for the inphase and quadrature components of a correlator pair, respectively. The subscripts E, P and L represent early, prompt and late versions of the corrletor pairs, respectively. The mathematical expressions of a correlator pair at time epoch k and for the i^{th} satellite are given below [63]

$$\begin{aligned} I_{m_{i,\,k}} &= \frac{A_i}{\sqrt{2}} M D_{i,\,k} \frac{\sin(\pi \Delta f_{d_{i,\,k}} T_{coh})}{\pi \Delta f_{d_{i,\,k}} T_{coh}} R\left(\tau_{i,\,k} + \delta T_c\right) \cos(\Delta\Phi_{i,\,k}) + \eta_{i,\,k} \\ Q_{m_{i,\,k}} &= \frac{A_i}{\sqrt{2}} M D_{i,k} \frac{\sin(\pi \Delta f_{d_{i,\,k}} T_{coh})}{\pi \Delta f_{d_{i,\,k}} T_{coh}} R\left(\tau_{i,\,k} + \delta T_c\right) \sin(\Delta\Phi_{i,\,k}) + \eta_{i,\,k} \end{aligned} \tag{5.3}$$

where $m = E, P, L$. A = amplitude of the digital intermediate frequency (IF) signal, M = number of IF samples in a correlation interval, D = navigation data bit, T_{coh} = coherent integration time, R = C/A-code autocorrelation function. $\Delta\phi_{i,\,k} = k^{\text{th}}$ carrier phase error at the end of T_{coh}. The code phase offset δ can take on three values: $-\frac{d}{2}$, 0, $\frac{d}{2}$ for E, P, L, respectively, where d is the early and late correlator spacing. T_c is the code chip width. η is the correlator noise which is normally distributed with mean zero and variance σ_{IQ}^2. The noise statistics remain the same across all correlators. σ_{IQ}^2 is derived in Appendix A. The derivation relaxes the white noise assumption [63] of the bandlimited pre-correlation noise and takes into account correlations among adjacent IF noise samples. However, as the pre-correlation noise time constant is much smaller than the coherent integration time, η is assumed to be white.

A coherent averaging time of T_a ms is chosen in this implementation. The early (and late) correlator outputs derived every 1 ms are averaged over T_a ms to smooth noise, thus, producing one pair of averaged early (and late) correlator outputs every T_a ms. On the other hand, two pairs of prompt correlator outputs are generated every T_a ms, with each pair having a smoothing/averaging time of $T_a/2$ ms. The averaged correlator noise variances are: $\sigma_{m_{i,k}}^2 = \frac{\sigma_{IQ_{i,k}}^2}{T_a}$, $m = E$, L and $\sigma_{p_{i,k}}^2 = \frac{2\sigma_{IQ_{i,k}}^2}{T_a}$.

With a non-coherent tracking architecture, the averaged correlator pairs are passed to nonlinear discriminator functions to generate independent estimates of the code phase and Doppler shift errors. The nonlinear discriminator outputs are accumulated every T_a ms and l such outputs are averaged over lT_a ms before sending to the tracking loop filters. Therefore, the early-minus-late power discriminator output $D_{\tau_{i,k}}$ at time epoch k (t_k - $t_{k-1} = lT_a$ ms) for the i^{th} satellite is

$$\begin{aligned} D_{\tau_{i,k}} &= A^2M^2(2-d)\tau_{i,k} + \zeta_{i,k} \\ \text{or, } D_{\tau_{i,k}}/(A^2M^2(2-d)) &= \tau_{i,k} + \zeta_{i,k}/(A^2M^2(2-d)) \\ \text{or, } \bar{D}_{\tau_{i,k}} &= \tau_{i,k} + \bar{\zeta}_{i,k} \end{aligned} \tag{5.4}$$

It can be shown that the normalized discriminator ($\bar{D}_{\tau_{i,k}}$) noise variance is $\sigma_{\bar{\zeta}_{i,k}}^2 = (4/l)d(2-d)[(2-d)\dfrac{A^2M^2}{4} + 2\sigma_{EL_{i,k}}^2]\sigma_{EL_{i,k}}^2/(A^2M^2(2-d))^2$. It is important to note that, although ζ is derived from a nonlinear discriminator operation, it remains white as the correlator noise samples used to derive ζ in any two intervals are independent of one another. Now if B_L is the one-sided noise bandwidth of the DLL, the code phase jitter due to white noise is given by $\sigma_{w_{o_{i,\,k}}}^2 = 2B_L(lT_a)\sigma_{\bar{\zeta}_{i,k}}^2$.

Colored Noise Jitter

The code phase error τ in Equation 5.4 has both colored noise ζ_{c_o} and the part of the white noise $\bar{\zeta}$ that is filtered through the tracking loop, denoted as ζ_{w_o}. The colored noise arises from the satellite clock phase jitter, receiver clock jitter, vibration and ionospheric delay [64]. Following the approach of [64], an analytical error model for each of these sources can be used to calculate the total colored noise jitter in DLLs. While this approach provides an accurate estimate of the colored jitter, it requires good knowledge of each of the error sources. As an alternative, a variant of the phase jitter estimation methodology developed in [64] is used. It is assumed that the dynamic stress error is zero as it will be shown later that the dynamic stress error in carrier aided DLLs is usually negligible.

The jitter estimation method is based on measuring the tracking error power σ_D^2 at the discriminator output. In a locked DLL the tracking power is governed only by noises, as shown below

$$\begin{aligned}
\bar{D}_{\tau_k} &= \bar{\zeta}_k - \zeta_{w_{o_{k-1}}} - \zeta_{c_{o_{k-1}}} \\
\sigma_{D_k}^2 &= E[(\bar{\zeta}_k - \zeta_{w_{o_{k-1}}} - \zeta_{c_{o_{k-1}}})^T(\bar{\zeta}_k - \zeta_{w_{o_{k-1}}} - \zeta_{c_{o_{k-1}}})] \\
\sigma_{D_k}^2 &= E[\bar{\zeta}_k^T \bar{\zeta}_k] + E[\zeta_{w_{o_{k-1}}}^T \zeta_{w_{o_{k-1}}}] - E[\bar{\zeta}_k^T \zeta_{w_{o_{k-1}}}] \\
&\quad - E[\bar{\zeta}_k \zeta_{w_{o_{k-1}}}^T] + E[\zeta_{c_{o_{k-1}}}^T \zeta_{c_{o_{k-1}}}] \\
\sigma_{D_k}^2 &= \sigma_{\bar{\zeta}_k}^2 + \sigma_{w_{o_{k-1}}}^2 - 2E[\bar{\zeta}_k^T \zeta_{w_{o_{k-1}}}] + \sigma_{c_{o_{k-1}}}^2
\end{aligned} \tag{5.5}$$

For simplicity of notation the satellite index i has been dropped from the above equations. As there is no correlation between the white and colored noises, the colored and white noise correlation terms are zero. At steady state $\sigma_{c_{o_{k-1}}}^2 = \sigma_{c_{o_k}}^2$ and $\sigma_{w_{o_{k-1}}}^2 = \sigma_{w_{o_k}}^2$. $E[\bar{\zeta}_k^T \zeta_{w_{o_{k-1}}}] = 0$ as $\bar{\zeta}_k$ and $\zeta_{w_{o_{k-1}}}$ are independent. $\sigma_{\bar{\zeta}_k}^2$ and $\sigma_{w_{o_k}}^2$ were defined when deriving the white jitter. Now re-arranging Equation 5.5, the colored jitter is found as

$$\sigma_{c_{o_k}}^2 = \sigma_{D_k}^2 - \sigma_{\bar{\zeta}_k}^2 - \sigma_{w_{o_k}}^2 \tag{5.6}$$

Samples of the normalized discriminator outputs are collected after the loop has reached steady state and these samples are used to compute σ_D^2. Then from σ_D^2 σ_c^2 is computed using the above equation.

Dynamic Stress

It is stated in [68] that the dynamic stress error of a carrier aided code tracking loop can usually be neglected. To justify this claim, an upper bound for the dynamic stress error (σ_{dyn_u}) of an FLL-aided DLL is derived. It is assumed that the FLL is of second order and experiences a constant jerk stress error equal to its tracking threshold defined in [68]. So, the FLL dynamic stress error is $0.25/T_a = 25$ Hz, where $T_a = 10$ ms. An FLL-aided DLL, therefore, needs to track a scaled down value of 25 Hz i.e., 25×C/A-code chipping rate/carrier frequency which reduces to $25 \times 1.023 \times 10^6/1.575 \times 10^9 = 0.0162$ Hz for L_1 carrier frequency and C/A-code. The corresponding dynamic stress error (σ_{d_u}) of a first order DLL with 1 Hz noise bandwidth is $0.0162/(1/0.25) = 0.00406$ chip (or 1.191 m) while that for a higher order DLL is zero. Note that $1/0.25$ is the loop natural frequency of a first order loop. The rule-of-thumb tracking threshold of a DLL is equal to the early-late correlator spacing d, in chips [68]. Hence, with the standard correlator spacing of 1 chip, σ_{dyn} of a first order DLL is only 1.12% of its tracking threshold. It should also be noted that the above upper bound is fairly conservative. This is because the actual dynamic stress error of an FLL would be smaller than its tracking threshold in order for it to maintain lock. The dynamic stress error of a first order DLL, though small, can be computed explicitly when infomation about the receiver's motion is available from another sensor such as an INS [64].

5.3.1.2 Fault Detection Methods

The fault detection method for RAIM is usually designed as a binary hypothesis test in which a test statistic is compared against a pre-determined threshold to detect faults. While there is a large body of work on least squares estimation-based RAIM algorithms [52]- [55], [85] is one of the few works dealing with KF-based RAIM. The test statistic in [85] is derived recursively using the Kalman gain and state error covariance matrix and is shown to follow a generalized chi-squared distribution. Although this thesis uses EKF implementations for both scalar and vector loop navigation filters, the weighted least squares-based fault detection method is adopted here. The pseudorange residuals for the weighted least squares method used in this thesis are formed by projecting the EKF-predicted position error onto the LOS vectors. Then, the weighted least squares estimate of the error in the predicted position is used for fault detection purposes rather than the EKF-estimated position error. The EKF-estimated position error computed from the same pseudorange residuals is used to update the user position and, therefore, to calculate the protection levels.

The rationale behind using the weighted least squares method for fault detection purposes is as follows. A fault detection test relies on statistical consistency checks rather than on the accuracy of the position estimate. So, even for an EKF a least squares estimation-based fault detection test can be formulated provided the protection level calculations account for the EKF position error. It can be argued that as long as the underlying error distributions are modeled correctly and the test statistic and position error are statistically independent, either method (KF or least-squares) should detect fault and allow integrity risk computation. While it is not clear at this point if the performance (e.g., fault detection time) of the KF method will be superior to that of the least squares method, the latter is favored here because of the following advantages: First, under no fault, the statistical properties of the test statistic based on weighted least squares method are independent of measurement noise statistics whereas those of the KF-based test statistic in [85] are not. Second, the weighted least squares test statistic and its noise parameters are easier to compute. Additionally, the statistical independence between the weighted least squares test statistic and EKF position error can be proven easily using the recursive implementation of the EKF, unlike the batch method needed in [85].

Two fault detection methods based on current (snapshot) and past residuals are considered in this thesis. Each method will have a different approach to bounding the EKF position error, as will be discussed later.

Snapshot Fault Detection Method: In this method the weighted RAIM algorithm presented in [53] is considered for fault detection purposes. It uses a scalar test statistic to flag system fault. The statistic is determined using a snapshot approach. This means that, at a given time epoch, the statistic relies on the pseudorange measurement outputs of only that epoch. The pseudorange measurements are also appropriately weighted to improve fault detection performance. The test statistic z in weighted RAIM is defined as the square root of a scalar measure called the weighted sum of the squared errors ($WSSE$). $WSSE$ at time epoch k is given by

$$WSSE_k = z_k^2 = \Delta\underline{\rho}_k^T W_k^{-1}(I - D_k)\Delta\underline{\rho}_k \tag{5.7}$$

where $\Delta\underline{\rho}_k$ is the $n \times 1$ vector of linearized pseudorange measurement residuals at time epoch k and n is the number of visible satellites. They are formed by taking the difference between the pseudorange measurements and the EKF-predicted pseudoranges. W_k^{-1} is the inverse of the pseudorange error ($\underline{\epsilon}_k$) covariance matrix. I is the $n \times n$ identity matrix. $D_k = H_k(H_k^T W_k^{-1} H_k)^{-1} H_k^T W_k^{-1}$ and H_k is the pseudorange measurement model matrix at the k^{th} time instant. The derivation of Equation 5.7 is given next. $\Delta\underline{\rho}_k$ is related to the error (ΔX_k) in the predicted position vector as follows:

$$\Delta\underline{\rho}_k = H_k \Delta X_k + \underline{\epsilon}_k \tag{5.8}$$

So, the weighted least squares estimate of ΔX is

$$\Delta\tilde{X}_k = (H_k^T W_k^{-1} H_k)^{-1} H_k^T W_k^{-1} \Delta\underline{\rho}_k \tag{5.9}$$

Thus, the test statistic is given as

$$WSSE_k = (\Delta\underline{\rho}_k - H_k\Delta\tilde{X}_k)^T W_k^{-1} (\Delta\underline{\rho}_k - H_k\Delta\tilde{X}_k)$$

Substituting for $\Delta\tilde{X}_k$

$$\begin{aligned} WSSE_k =& (\Delta\underline{\rho}_k - H_k(H_k^T W_k^{-1} H_k)^{-1} H_k^T W_k^{-1} \Delta\underline{\rho}_k)^T \times \\ & W_k^{-1}(\Delta\underline{\rho}_k - H_k(H_k^T W_k^{-1} H_k)^{-1} H_k^T W_k^{-1} \Delta\underline{\rho}_k) \\ =& (\Delta\underline{\rho}_k - D_k\Delta\underline{\rho}_k)^T W_k^{-1} (\Delta\underline{\rho}_k - D_k\Delta\underline{\rho}_k) \\ =& \Delta\underline{\rho}_k^T (I - D_k)^T W_k^{-1} (I - D_k) \Delta\underline{\rho}_k \\ =& \Delta\underline{\rho}_k^T (W_k^{-1} - D_k^T W_k^{-1} - W_k^{-1} D_k + D_k^T W_k^{-1} D_k) \Delta\underline{\rho}_k \end{aligned}$$

Since $D_k^T W_k^{-1} = W_k^{-1} D_k = D_k^T W_k^{-1} D_k$, the above equation is simplified as

$$WSSE_k = \Delta\underline{\rho}_k^T W_k^{-1} (I - D_k) \Delta\underline{\rho}_k \tag{5.10}$$

It can be shown that, under fault-free conditions, $WSSE_k$ is chi-squared distributed with $n - 4$ degrees of freedom (DOF). When a fault of magnitude b occurs in the i^{th} satellite, $WSSE_k$ has a non-central chi-square distribution with $(n - 4)$ DOF and non-centrality parameter $\lambda = S_{ii} b^2$, where S_{ii} is the i^{th} diagonal element of $W_k^{-1}(I - D_k)$.

As noted earlier, the fault detection test is designed as a binary hypothesis test with the following decision rule. At a given time if the test statistic z is below a certain threshold T, there is no fault, as shown in Figure 5.3. Conversely, if the statistic is above the threshold, system operation is considered unsafe. The threshold is determined from the system continuity requirement. That is, under fault-free conditions and over an intended period of operation, z can exceed the threshold with a given probability of false alarm P_{FA}. Mathematically, P_{FA} is expressed as [53]

$$1 - P_{\text{FA}} = \frac{1}{2^a \Gamma(a)} \int_0^{T^2} e^{-x/2} x^{a-1} dx \tag{5.11}$$

where a is $(n-4)/2$; Γ is the Gamma funtion. T^2 is obtained from the chi-square table using P_{FA} and a.

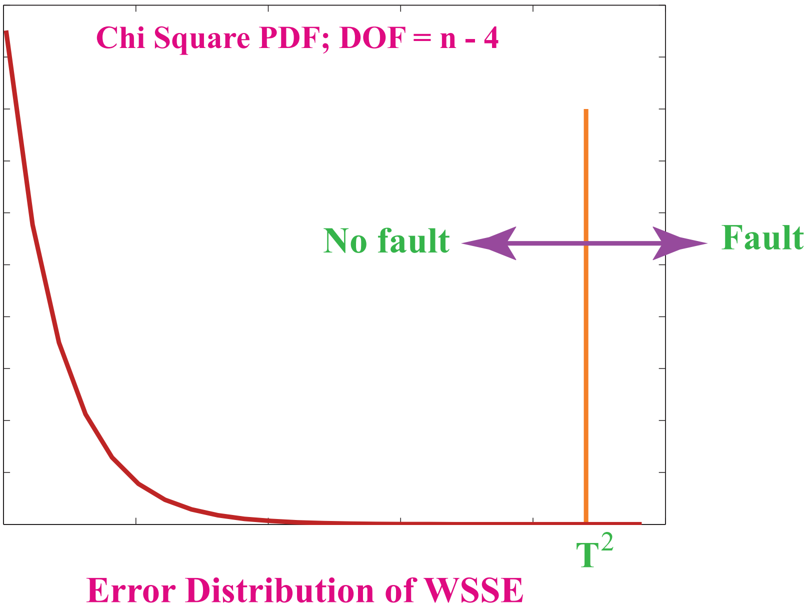


Figure 5.3: Chi-Square PDF (Assuming n - 4 = 1)

Fault Detection Method Using Current and Past Snapshot Test Statistics: As will be shown later, the mean of EKF position error under fault depends on current as well as past fault magnitudes, in contrast with the least squares position estimation method. Therefore, to bound the mean of the position error for computing protection levels, an assumption that the fault magnitude remains constant or grows with time is made. This is needed so that the previously described snapshot method can be used to compute protection levels. This approach is employed in the simulation studies of this chapter. Further, in an effort to relax this assumption, a fault detection method can be used whose test statistic is the sum of the snapshot test statistics of current epoch plus a number of past epochs. The number of past epochs $(N-1)$ is determined by the number of epochs required to calculate the mean position error discussed later. The test statistic is given by

$$z_k^2 = \sum_{i=k-(N-1)}^{k} \Delta\underline{\rho}_i^T W_i^{-1}(I - D_i)\Delta\underline{\rho}_i \tag{5.12}$$

where $\Delta\underline{\rho}_i$, W_i and D_i are defined below Equation 5.7. Note that each term of the above equation is chi-squared distributed under no fault conditions and non-central chi-squared distributed under fault. They are independent of one another as the pseudorange measurement noise vector $\underline{\epsilon}_i$ at time t_i is assumed to be Gaussian-distributed and uncorrelated in time. The assumption of uncorrelated noise holds when the time interval between two successive measurement vectors ($t_i - t_{i-1}$) is chosen to be longer than the scalar tracking loop time constant. In addition, time correlated errors such as atmospheric delays, ephemeris errors and multipath must be appropriately modeled, as discussed before. Thus, the underlying Gaussian noise vector in each term of the above equation is independent of the other, making individual terms independent of one another. As a result, z_k^2 is chi-squared distributed with DOF $N \times (n-4)$ under no fault and non-central chi-squared distributed under fault with DOF $N \times (n-4)$ and non-centrality parameter given by the sum of the non-centrality parameters of all terms.

5.3.1.3 Vertical and Horizontal Position Errors of EKF

An EKF algorithm is used as the scalar loop navigation filter to estimate the errors or deviations of position, velocity and clock parameters from their true values. It should be noted that the user acceleration is not modeled in the filter. If X_k = true user position, velocity, clock bias and clock drift at time k and $\tilde{X}_k^{(-/+)}$ = predicted/estimated user position, velocity and clock variables at t_k, then $(\Delta\tilde{X}_k^-)$ and $(\Delta\tilde{X}_k^+)$ denote *a priori* and *a posteriori* estimates of filter state vector ΔX_k, respectively. ΔX_k is defined as follows: $\Delta X_k = X_k$ - $\tilde{X}_k^- = [\Delta x\ \Delta\dot{x}\ \Delta y\ \Delta\dot{y}\ \Delta z\ \Delta\dot{z}\ \Delta b\ \Delta\dot{b}]_k^T$. The vector $[x, y, z, b]^T$ represents the x, y, z components of user position and clock bias and $[\dot{x},\ \dot{y},\ \dot{z}, \dot{b}]$ denotes three components of user velocity and clock drift, respectively. Similar to the vector loop navigation filter described in Chapter 3, $\Delta\tilde{X}_k^-$ and $\Delta\tilde{X}_k^+$ are given by

$$\Delta\tilde{X}_k^- = 0 \tag{5.13}$$

$$\Delta\tilde{X}_k^+ = K_{\mathrm{stl}_k}[\Delta\underline{\rho}_k^T\ \Delta\underline{\dot{\rho}}_k^T]^T \tag{5.14}$$

where K_{stl_k} represents the EKF Kalman gain for the scalar architecture and $[\Delta\underline{\rho}_k^T \Delta\underline{\dot{\rho}}_k^T]^T$ is the linearized measurement or innovations vector. The measurement noise covariance matrix at time t_k for $\Delta\underline{\rho}_k$ is given by W_k which is defined in relation to the discussion of Equation 5.1 earlier. The measurement noise covariance matrix for $\Delta\underline{\dot{\rho}}_k$ is denoted by W_k'. W_k' is determined by the FLL jitter defined in [75]. The initial state error covariance matrix is the identity matrix of dimension 8×8. The process noise covariance matrix Q is chosen to optimize filter performance. Time-correlated errors are not modeled in this thesis. This assumption holds for the simulation studies of this chapter as well as for the GPS data of the next chapter which was generated by a Spirent simulator using broadcast ephemeris and clock of satellites and standard atmospheric

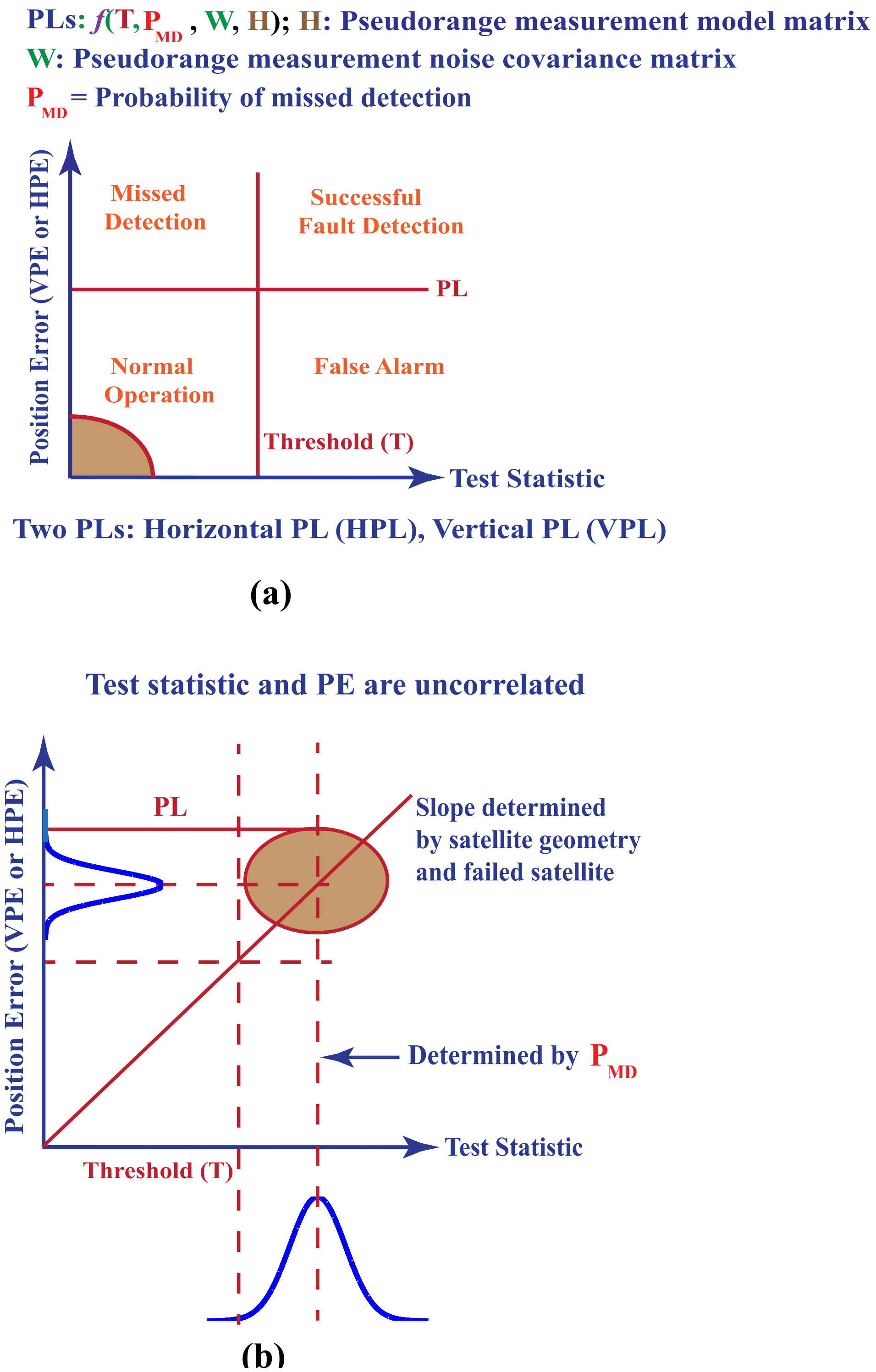


Figure 5.4: (a) Shows Four Quadrants of Operation on the Statistic-Position Error Plane; (b) Pictorially Describes the Relationship between Protection Levels and Test Statistic at a Given Epoch under Fault; PL: Protection Level

models.

The x, y and z axes position errors are $\epsilon_{m_k} = \Delta m_k - \Delta\tilde{X}_k^+(l)$, where $m = x, y, z$ and $l = 1, 3, 5$, respectively. If Σ_k^+ is the *a posteriori* error covariance matrix at t_k, then the variances of the x, y and z axes position errors are given by $\sigma^2_{m_k} = \Sigma_k^+(l,l)$; $m = x, y, z$ and $l = 1, 3, 5$, respectively. Thus, under fault-free conditions and at time t_k, the vertical position error (VPE) is Gaussian-distributed with zero mean and variance $\sigma^2_{z_k}$. For purposes of calculating the horizontal protection level (HPL), which will be discussed later, the horizontal position error (HPE) at no fault is usually approximated to be Gaussian-distributed with zero mean and variance $(\sigma^2_{x_k} + \sigma^2_{y_k})$ [55]. To justify this approximation, it is shown in Appendix C that, beyond a certain M, the probability that the HPE is greater than k ($> M$) is bounded above by the corresponding probability of a Gaussian random variable with zero mean and variance $(\sigma^2_{x_k} + \sigma^2_{y_k})$. This implies that the HPL calculated using the above Gaussian distribution assumption would bound the actual HPL (obtained from the true distribution) because the HPL is larger than M. Thus, the use of a Gaussian distribution for the HPE in place of a complex non-Gaussian one is justified. Moreover, the Gaussian assumption does not exhibit any numerical problems that may arise with the actual non-Gaussian distribution.

It can be shown from Equations 5.7 and 5.14 that the distribution of operation points in the statistic-position error plane (for both vertical and horizontal dimensions) is an ellipse [55]. Under no fault, the ellipse is centered near the origin (see Figure 5.4 (a)). When fault occurs in any one satellite, the center of the ellipse moves up along a straight line depending on the fault magnitude, satellite geometry and the failed satellite. This is shown in Figure 5.4 (b). The slope of the line (also called failure mode slope) is determined by the mean of VPE (or HPE) and the square root of the non-centrality parameter of WSSE. For a fault of magnitude b in satellite i, $b^2 S_{ii_k}$ is the non-centrality parameter of $WSSE_k$ or z_k^2; S_{ii_k} is the i^{th} diagonal element of $W_k^{-1}(I - D_k)$. $b\tilde{K}_{\text{stl}_k}(5,i)$ and $b(\sqrt{\tilde{K}^2_{\text{stl}_k}(1,i) + \tilde{K}^2_{\text{stl}_k}(3,i)})$ are defined as upper bounds on the means of the VPE and HPE, respectively, which are used to calculate the maximum failure mode slopes and protection levels. The derivations of $\tilde{K}_{\text{stl}_k}(l,i)$, $l = 1, 3, 5$ and bounds on the means of VPE and HPE are given below.

$\tilde{K}_{\text{stl}_k}(l,i)$ and Bounds on Means of VPE and HPE under Fault

In order to derive $\tilde{K}_{\text{stl}_k}(l,i)$, it is assumed that a failure has occurred in satellite i. The failure started at t_m and b_i is the fault magnitude (or bias) at t_{k+m-1} for the psedorange measurement and $\dot{b}_i$ is the corresponding fault magnitude of the pseudorange rate measurement. Due to the failure, the pseudorange measurement noise at t_k is distributed as $\mathcal{N}(b_{k-m+1},\, W_k)$. With these assumptions the mean $\underline{\mu}_k$ of the *a posteriori* state estimate error ($\underline{\epsilon}_{X_k}$) will be derived for t_k ($> t_m$). Let F_{k-1} and C_k be the state transition and measurement model matrices of the EKF, respectively. Now using the filter time and measurement update equations, the mean of $\underline{\epsilon}_{X_k}$ can be written as [86]

$$\begin{aligned}\underline{\mu}_k &= (I - K_{\mathrm{stl}_k} C_k) F_{k-1} \underline{\mu}_{k-1} - K_{\mathrm{stl}_k} [0 \;\ldots\; 0 \; b_{k-m+1} \;\ldots\; 0 \; \dot{b}_{k-m+1} \;\ldots\; 0]^T \\ &= A_k \underline{\mu}_{k-1} - K_{\mathrm{stl}_k} \underline{f}_{i,\, k-m+1} \end{aligned} \tag{5.15}$$

where $\underline{f}_{i,\, k-m+1}$ is a column vector such that $\underline{f}_{i,\, k-m+1}(i) = b_{k-m+1}$; $\underline{f}_{i,\, k-m+1}(i+n) = \dot{b}_{k-m+1}$ and $\underline{f}_{i,\, k-m+1}(j) = 0$ for $j \neq i$ and $j \neq i+n$; n is the number of visible satellites. It is assumed that the filter estimate was unbiased under no-fault conditions. So, $\underline{\mu}_{m-1}$ = 0, $\underline{\mu}_m = -K_{\mathrm{stl}_m} \underline{f}_{i,\, 1}$ and so on. Thus, $\underline{\mu}_k$ becomes

$$\begin{aligned}\underline{\mu}_k =& A_k A_{k-1} \ldots A_{m+1} (-K_{\mathrm{stl}_m} \underline{f}_{i,\, 1}) + A_k \ldots A_{m+2} (-K_{\mathrm{stl}_{m+1}} \underline{f}_{i,\, 2}) + \\ & \ldots + A_k (-K_{\mathrm{stl}_{k-1}} \underline{f}_{i,\, k-m}) + (-K_{\mathrm{stl}_k} \underline{f}_{i,\, k-m+1}) \\ =& \prod_{j=k}^{m+1} A_j (-K_{\mathrm{stl}_m} \underline{f}_{i,\, 1}) + \prod_{j=k}^{m+2} A_j (-K_{\mathrm{stl}_{m+1}} \underline{f}_{i,\, 2}) + \ldots \\ & + A_k (-K_{\mathrm{stl}_{k-1}} \underline{f}_{i,\, k-m}) + (-K_{\mathrm{stl}_k} \underline{f}_{i,\, k-m+1}) \end{aligned} \tag{5.16}$$

Since for a stable filter $|\mathrm{eig}(A_j)| < 1$ for any j, $\prod_{j=k}^{m+1} A_j \rightarrow 0$ as $k \rightarrow \infty$. This implies that the last N terms of the above equation will effectively determine $\underline{\mu}_k$, where N is determined by ($|\mathrm{eig}(A_j)|$) and is inversely related to C/N$_0$. Thus, considering the last N terms of the above equation gives

$$\begin{aligned}\underline{\mu}_k &= -\sum_{j=0}^{N-2} \left(\left[\prod_{p=k}^{k-j} A_p \right] K_{\mathrm{stl}_{k-j-1}} \right) \underline{f}_{i,\, k-m-j} - K_{\mathrm{stl}_k} \underline{f}_{i,\, k-m+1} \\ &= -\sum_{j=0}^{N-2} B_{k-j} \underline{f}_{i,\, k-m-j} - K_{\mathrm{stl}_k} \underline{f}_{i,\, k-m+1} \end{aligned} \tag{5.17}$$

N is obtained from the following inequality in this thesis:

$$\frac{\max_{i=1,\,3,\,5} \{ |\mathrm{eig}(A_k)|_i \}}{p} > \max_{i=1,\,3,\,5} \{ |\mathrm{eig}(\prod_{j=k}^{k-(N-2)} A_j)|_i \} \tag{5.18}$$

That is, (maximum of absolute eigenvalues of A_k corresponding to the position states)/p > maximum of absolute eigenvalues of ($\prod_{j=k}^{k-(N-2)} A_j$) corresponding to the position states. p is heuristically selected as 200 for C/N$_0$ > 40 dB-Hz and 600 otherwise. A lower value of p is chosen at high C/N$_0$ because $(\prod_{j=k}^{k-(N-2)} A_j) \to 0$ faster when C/N$_0$ is high. The eigenvalues corresponding to the position states are obtained by matching the eigenvalues to the matrix diagonal elements. Thus, $\underline{\mu}_k$ is computed over a moving window of length N. The reason for choosing the eigenvalues to determine N is that the eigenvalues are a measure of how fast $(\prod_{j=k}^{k-(N-2)} A_j) \to 0$.

Note that the disadvantage of past faults affecting the current state error is that past bias rates will also contribute to determining the current position error, although their effect is attenuated by the term $(I - K_{\mathrm{stl}_k} C_k)$. If b_i is the fault magnitude (or bias) at t_{i+m-1}, its rate of change $\dot{b}_i$ is approximated as follows: $\dot{b}_i = (b_i - b_{i-1})/(t_{i+m-1} - t_{i+m-2}) = (b_i - b_{i-1})/T$, where T is a few seconds; and $\dot{b}_{k-m-(N-2)} = \dot{b}_{k-m-(N-3)}$ i.e., the bias rates of the last two time epochs in the calculation of $\underline{\mu}_k$ are identical. The above approximation is reasonable for step faults, ramp faults and sinusoidal faults with time periods on the order of hours, which can be approximated by a piecewise ramp or step over a few seconds. The l^{th} element of $\underline{\mu}_k$ is given by

$$\begin{aligned}
\underline{\mu}_k(l) = &- \sum_{j=0}^{N-2} (B_{k-j}(l,\, i) b_{k-m-j} + B_{k-j}(l,\, i+n) \dot{b}_{k-m-j}) - \\
&(K_{\mathrm{stl}_k}(l,\, i) b_{k-m+1} + K_{\mathrm{stl}_k}(l,\, i+n) \dot{b}_{k-m+1}) \\
= &- (B_{k-(N-2)}(l,\, i) b_{k-m-(N-2)} + B_{k-(N-2)}(l,\, i+n) \dot{b}_{k-m-(N-2)}) - \\
&\sum_{j=0}^{N-3} (B_{k-j}(l,\, i) b_{k-m-j} + B_{k-j}(l,\, i+n) \dot{b}_{k-m-j}) - \\
&(K_{\mathrm{stl}_k}(l,\, i) b_{k-m+1} + K_{\mathrm{stl}_k}(l,\, i+n) \dot{b}_{k-m+1})
\end{aligned}$$

Substituting for $\dot{b}$

$$\begin{aligned}
\underline{\mu}_k(l) = &- B_{k-(N-2)}(l,\, i) b_{k-m-(N-2)} - \\
&B_{k-(N-2)}(l,\, i+n)/T (b_{k-m-(N-3)} - b_{k-m-(N-2)}) - \\
&\sum_{j=0}^{N-3} (B_{k-j}(l,\, i) b_{k-m-j} + B_{k-j}(l,\, i+n)/T (b_{k-m-j} - b_{k-m-(j+1)})) - \\
&(K_{\mathrm{stl}_k}(l,\, i) b_{k-m+1} + K_{\mathrm{stl}_k}(l,\, i+n)/T (b_{k-m+1} - b_{k-m}))
\end{aligned}$$

Denoting the coefficient of bias b_{k-m-j} in $\underline{\mu}_k(l)$ as $h_{l,\,j+2}$, the above equation reduces to

$$\underline{\mu}_k(l) = \sum_{j=-1}^{N-2} h_{l,\,j+2} b_{k-m-j} \tag{5.19}$$

where

For $N >= 4$

$$\begin{aligned}
h_{l,\,N} &= -\,(B_{k-(N-2)}(l,i) - B_{k-(N-2)}(l,\ i+n)/T) + B_{k-(N-3)}(l,\ i+n)/T \\
h_{l,\,N-1} &= -\,B_{k-(N-2)}(l,\ i+n)/T - (B_{k-(N-3)}(l,\ i) + B_{k-(N-3)}(l,i+n)/T) + \\
&\quad B_{k-(N-4)}(l,i+n)/T \\
h_{l,\,j+2} &= -\,(B_{k-j}(l,\ i) + B_{k-j}(l,i+n)/T) + B_{k-(j-1)}(l,i+n)/T; j = 1,\ldots,N-4 \\
h_{l,\,2} &= -\,(B_k(l,\ i) + B_k(l,i+n)/T) + K_{\mathrm{stl}_k}(l,\ i+n)/T \\
h_{l,\,1} &= -\,(K_{\mathrm{stl}_k}(l,\ i) + K_{\mathrm{stl}_k}(l,\ i+n)/T)
\end{aligned}$$

For $N = 3$

$$\begin{aligned}
h_{l,\,3} &= -\,(B_{k-1}(l,\ i) - B_{k-1}(l,i+n)/T) + B_k(l,i+n)/T \\
h_{l,\,2} &= -\,B_{k-1}(l,i+n)/T - (B_k(l,\ i) + B_k(l,i+n)/T) + K_{\mathrm{stl}_k}(l,\ i+n)/T \\
h_{l,\,1} &= -\,(K_{\mathrm{stl}_k}(l,\ i) + K_{\mathrm{stl}_k}(l,\ i+n)/T)
\end{aligned}$$

For $N = 2$

$$\begin{aligned}
h_{l,\,2} &= -\,(B_k(l,\ i) - B_k(l,i+n)/T) + K_{\mathrm{stl}_k}(l,\ i+n)/T \\
h_{l,\,1} &= -\,(K_{\mathrm{stl}_k}(l,\ i) + K_{\mathrm{stl}_k}(l,\ i+n)/T) - B_k(l,i+n)/T
\end{aligned}$$

For $N = 1$

$$h_{l,\,1} = -\,K_{\mathrm{stl}_k}(l,\ i)$$

As stated earlier, Equation 5.19 holds if the bias rate does not change over the time interval between two position updates, which typically varies between (2 - 6) sec for integrity monitoring with the vector architecture. It can be argued that a bias whose rate is changing fast is more likely to be detected soon and, therefore, presents a lower integrity risk. Next, one of the following approaches can be used to derive a bound on $\mu_k(l)$ and $\tilde{K}_{\mathrm{stl}_k}(l,i)$.

1. This approach is suitable for the snapshot fault detection method described earlier. It is used in the simulation studies of this chapter. In this method it is assumed that the fault magnitude remains constant or increases with time i.e., $|b_1| \leqslant |b_2| \leqslant \ldots \leqslant |b_{k-m}| \leqslant |b_{k-m+1}|$, where b_i is the fault magnitude (or bias) at t_{i+m-1}. With this assumption an upper bound on the absolute value of the l^{th} element of $\underline{\mu}_k$ is determined and $\tilde{K}_{\mathrm{stl}_k}(l,i)$ is obtained from that bound. The bound is

$$|\underline{\mu}_k(l)| \leqslant \sum_{j=-1}^{N-2} |h_{l,\,j+2}||b_{k-m+1}|$$

From the above bound $\tilde{K}_{\mathrm{stl}_k}(l,\, i)$ is defined as $(\sum_{j=-1}^{N-2} |h_{l,\, j+2}|)$. $b_{k-m+1}^{\mathrm{worst}}$ is determined by assuming $P\{(z < T)|\text{fault in sat } i\} = P_{MD}$; z is the test statistic; T is the detection threshold and P_{MD} is the probability of missed detection defined later. The failure mode slope for the VPE is $\tilde{K}_{\mathrm{stl}_k}(5,\, i)/\sqrt{S_{ii_k}}$ and for the HPE is $\sqrt{\tilde{K}_{\mathrm{stl}_k}(1,\, i)^2 + \tilde{K}_{\mathrm{stl}_k}(3,\, i)^2}/\sqrt{S_{ii_k}}$; $S_{ii_k}(b_{k-m+1}^{\mathrm{worst}})^2$ is the non-centrality parameter of $WSSE$ corresponding to the probability $P\{(z < T)|\text{fault in sat } i\} = P_{MD}$. The drawback of this method is that it makes an assumption on the temporal patterns of the fault.

2. The second method is more general in the sense that, unlike the first method, it does not make any assumption on the temporal characteristics of fault. In this method the worst case fault magnitudes for epochs $k-(N-1)$ through k are obtained using the approach of [87] and [88]. It maximizes the failure mode slope $\underline{\mu}_k(5)^2/\underline{b}^T\Lambda\underline{b}$ for VPE and $(\underline{\mu}_k(1)^2 + \underline{\mu}_k(3)^2)/\underline{b}^T\Lambda\underline{b}$ for HPE, where $\underline{b}$ $= [b_{k-m-(N-2)} \ldots\ b_{k-m+1}]^T$, $\Lambda = \mathrm{diag}([S_{ii_{k-(N-1)}}, \ldots, S_{ii_k}])$; S_{ii_j} is the i$^{\mathrm{th}}$ diagonal element of $W_j^{-1}(I - D_j)$, assuming satellite i is faulty; W_j is the pseudorange measurement noise covariance at t_j; $D_j = H_j(H_j^{-1}W_j^{-1}H_j)^{-1}H_j^TW_j^{-1}$; H_j is the pseudorange measurement model matrix at t_j. At any time epoch t_k the worst case bias vector with respect to VPE maximizes the corresponding failure mode slope, as given below

$$(\text{maximum failure mode slope})^2 = \frac{\underline{\mu}_k(5)^2_{\max}}{\underline{b}^T\Lambda\underline{b} = s} = \max_{\underline{b}^T\Lambda\underline{b}=s} \frac{\underline{b}^T L \underline{b}}{\underline{b}^T\Lambda\underline{b}} \tag{5.20}$$

where s is the non-centrality parameter of z^2 corresponding to the probability $P\{(z < T)|\text{fault in sat } i\} = P_{MD}$; z is obtained using the second fault detection method (i.e., using current snapshot test statistic and $(N-1)$ past snapshot test statistics). The solution to the above equation is the maximum eigenvalue of $L\Lambda^{-1}$ and the worst case bias vector b^{worst} is the corresponding eigenvector times $\sqrt{s}$. $\tilde{K}_{\mathrm{stl}_k}(5,\, i) = \underline{\mu}_k(5)_{\max}/b_{k-m+1}^{\mathrm{worst}}$. The L matrix is formed from the coefficients of Equation 5.19 as follows: L = $[h_{5,\,N} \ldots\ h_{5,\,1}]^T[h_{5,\,N} \ldots\ h_{5,\,1}]$.

The same approach is also used for the HPE. The only difference is that the L matrix for the HPE is

$$L = \begin{bmatrix} h_{1,\,N} & \ldots & h_{1,\,1} \\ h_{3,\,N} & \ldots & h_{3,\,1} \end{bmatrix}^T \times \begin{bmatrix} h_{1,\,N} & \ldots & h_{1,\,1} \\ h_{3,\,N} & \ldots & h_{3,\,1} \end{bmatrix} \tag{5.21}$$

Then similar to $\tilde{K}_{\mathrm{stl}_k}(5,\, i)$, $(\tilde{K}_{\mathrm{stl}_k}(1,\, i)^2 + \tilde{K}_{\mathrm{stl}_k}(3,\, i)^2)$ are obtained from $(\underline{\mu}_k(1)^2 + \underline{\mu}_k(3)^2)_{\max}$ and $b_{k-m+1}^{\mathrm{worst}}$.

For calculating the protection levels, the above upper bound of $\underline{\mu}_k$ is used to find the means of the VPE and HPE. Thus, the mean of the VPE ($= \underline{\epsilon}_{X_k}(5)$) is $\tilde{K}_{\mathrm{stl}_k}(5, i) b^{\mathrm{worst}}_{k-m+1}$. Using the Taylor series expansion of the mean of the HPE ($= \sqrt{\underline{\epsilon}_{X_k}(1)^2 + \underline{\epsilon}_{X_k}(3)^2}$), it can be shown that the mean is approximately equal to $b^{\mathrm{worst}}_{k-m+1} \times \sqrt{\tilde{K}^2_{\mathrm{stl}_k}(1,\ i) + \tilde{K}^2_{\mathrm{stl}_k}(3,\ i)}$.

It is important to note that the approximation made to the bias rate on page 149 can be relaxed if test statistics formed by the pseudorange rate residuals and velocity states are added to the test statistic of the fault detection method on page 144 and then the second approach above is used to determine the worst case bias and bias rate vector. However, this has not been implemented in this work as the underlying faults for both simulation studies and GPS measurements are assumed to have a ramp profile for which the approximation is valid.

In addition, note that the above method of bounding the position error assumes that the user position and velocity are updated at the same rate. This may not be true with the vector architecture. Any additional change made to the vector architecture for different position and velocity update rates will be discussed under vector loop RAIM. This modification cannot be applied to the scalar architecture. This is because the underlying assumption for that is that the velocity update rates are much faster than the position update rates, which is applicable only to the vector architecture.

5.3.1.4 Protection Levels

As illustrated in Figure 5.2, the scalar loop RAIM algorithm provides the following outputs: Result of the binary hypothesis fault detection test discussed earlier and protection levels when no fault is detected. In the terminology of statistics, a protection level bounds true position error with a given probability. Thus, protection levels together with the threshold T of the fault detection test divide the statistic-position error plane into four quadrants. Each quadrant is assigned a name which is shown in Figure 5.4 (a).

Protection levels are determined from the maximum allowable integrity risk (i.e., probability of MI or HMI). HMI (or MI) exists when the position error exceeds the protection levels without a timely alert to the user. More specifically, HMI exists if the VPE (and/or HPE) exceeds the protection levels for a period longer than the specified time-to-alert. This can happen in two ways. A fault is not detected or it is detected but annunciation takes longer than the time-to-alert limit. While the second possiblity is addressed by suitable system design (e.g., reducing communication lag, etc.), the calculation of protection levels ensures that the probability that a fault can cause integrity risk is always less than the required probability of HMI. Mathematically, the probability of HMI P_{HMI} is expressed as [82]

$$\begin{aligned}
&P\{(|\text{PE}| > \text{PL})|\text{no fault}\}P\{(z < T)|\text{no fault}\}(1 - P_{\text{F}}) + \\
&P\{(|\text{PE}| > \text{PL})|\text{fault in a sat}\}P\{(z < T)|\text{fault in a sat}\}P_{\text{F}} \\
&\qquad\qquad\qquad\qquad\qquad = P_{\text{HMI}}
\end{aligned} \tag{5.22}$$

where PE is either the HPE or VPE and the corresponding PL is HPL or VPL. P_{F} is the prior probability of a satellite failure over a specified time multiplied by the number of visible satellites. $P\{(z < T)|\text{no fault}\}(1 - P_{\text{F}}) \approx 1$. As defined earlier, z is the test statistic used for fault detection and T is the statistic threshold determined from P_{FA}. z and EKF PE are statistically independent, as proved in Appendix C based on the derivation of [89]. The events "no fault" and "fault in a satellite" are mutually exclusive and exhaustive. Now considering the VPE and following the method of [82], the integrity risk P_{HMI} is allocated between the two terms of the above equation as follows:

$$P\{(|\text{VPE}| > \text{VPL})|\text{no fault}\} = 0.5P_{\text{HMI}} \tag{5.23}$$

$$\begin{aligned}
&P\{(|\text{VPE}| > \text{VPL})|\text{fault in a sat}\}P\{(z < T)|\text{fault in a sat}\}P_{\text{F}} \\
&\qquad\qquad\qquad\qquad = P_{MD}P_{F} = 0.5P_{\text{HMI}}
\end{aligned} \tag{5.24}$$

Equation 5.23 denotes the probability that the VPE exceeds the VPL under no fault. On the other hand, Equation 5.24 stands for the probability that the VPE exceeds the VPL under fault, but no alarm is issued i.e., z is less than T. The first two terms of this equation together are known as the probablity of missed detection or P_{MD}.

Above two equations are used to determine the protection levels. If VPL_{NF} is the VPL derived from Equation 5.23 and VPL_{F} is the VPL obtained from Equation 5.24, then the final VPL is $\max\{\text{VPL}_{\text{NF}}, \text{VPL}_{\text{F}}\}$ [82]. VPL_{NF} and VPL_{F} are obtained using the following method. VPL_{NF} is determined from the distribution of position error under fault-free conditions using Equation 5.23. The procedure for finding VPL_{F} is schematically shown in Figure 5.4 (b). A close observation of Equation 5.24 reveals that determining the maximum VPL from this equation is an iterative process. Hence, a conservative approach is adopted which first finds the maximum failure mode slope using one of the methods described on pages (150 - 151). Then, the mean of the VPE is found by multiplying the maximum failure mode slope with the test statistic non-centrality parameter corresponding to P_{MD}. Next, to account for noise, the mean VPE is inflated by the term $K\sigma_z$, where $P\{(|\text{VPE}| > \text{K}\sigma_z) = \text{P}_{MD}$; σ_z = standard deviation

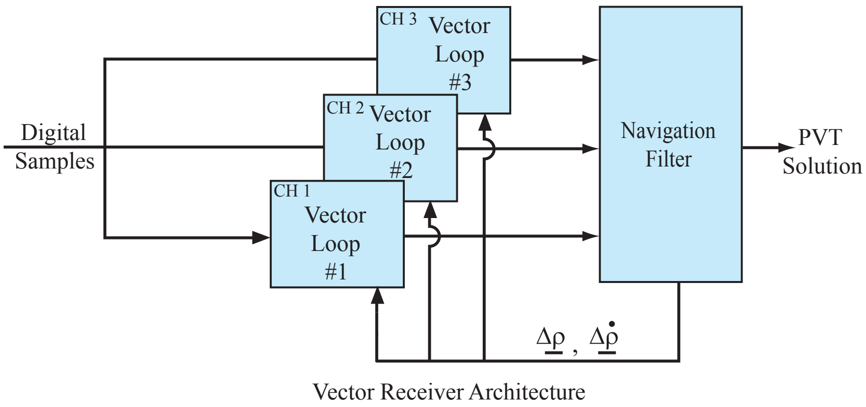


Figure 5.5: Vector Tracking Architecture

of vertical position error. Thus, the VPL for each of the visible satellites is calculated and the maximum of all these VPLs is set as $\mathrm{VPL_F}$. Note that the same approach also applies to calculating the HPL.

5.3.2 Vector Loop RAIM

Having outlined the scalar loop RAIM algorithm implemented in this work, now a test statistic for the vector tracking architecture will be defined. It will also be shown that the same relationship between the test statistic and position error as in scalar loop RAIM is preserved in this formulation. Moreover, position error and protection level computations discussed under scalar loop RAIM will apply to vector loops for federated coherent and non-coherent architectures.

Figure 5.5 shows a simplified representation of vector tracking architecture. As is evident in the figure, the navigation filter accepts inputs from all channels and processes them *together* to generate the pseudorange and pseudorange rate residuals. These residuals, in turn, correct the code and carrier NCOs of the tracking loops. Thus, the replica code phase of an individual channel is adjusted based on the corresponding pseudorange output of the navigation filter. As discussed earlier, a distinct problem with such an implementation is that the pseudorange measurements generated by the navigation filter are not suitable for fault detection purposes. This is because the combined processing of signals in the navigation filter allows a fault in one channel to easily corrupt all the pseudorange measurements, as noted earlier. Thus, the basic single fault assumption of RAIM is violated when it is implemented with the filter generated pseudorange measurements. To circumvent this problem, a function of the correlator outputs instead of the pseudorange measurements is used as input to vector loop RAIM. The motivation behind using the I and Q outputs is that the noise at the correlator level conforms to the single fault assumption of the RAIM algorithms. In what follows, a way to apply the traditional RAIM algorithms to vector loops is developed using the I and Q outputs.

5.3.2.1 Coherent Architecture

The coherent vector tracking architecture has two variants depending on the types of inputs sent to the navigation filter. The centralized coherent architecture directly inputs the I and Q measurements to the navigation filter at a rate of 50 Hz or more. A RAIM scheme for this architecture has been proposed in [57]. However, this approach has the following limitations: The test statistic (maximum magnitude of the residual I and Q outputs among all channels) is assumed to be Gaussian-distributed. This is incorrect as it can be easilty shown that the residuals have a Rayleigh distribution. The relationship between the statistic and the VPE is approximated to be linear based on only simulations. As an alternative, the following scheme is proposed which can overcome the above limitations. Recall from Equation 5.3 that the correlator outputs are a nonlinear function of the code phase alignment errors (τ) of the replica code with the received code and the Doppler frequency offsets (Δf_d). Of the two, τ is of particular interest because it is proportional to the difference between the actual and the receiver-predicted pseudorange measurements. Mathematically, it is expressed *in meters* as follows:

$$\underline{\tau}_m = H_m[X_{p_m} - \tilde{X}^-_{p_m}] + \underline{\epsilon}_{\text{vector},\, m} \tag{5.25}$$

where H_m and X_{p_m} are defined below Equation 5.1. $\epsilon_{\text{vector},\, m}$ is identical with $\underline{\epsilon}_m$ except that the vector receiver noise is different from the scalar architecture. The above equation shows that $\underline{\tau}_m$ has the same mathematical model as the linearized pseudorange measurements of the scalar architecture (Equation 5.1). The only difference is that its measurements have a different noise variance than that of its scalar counterpart. Therefore, for fault detection purposes, τ is first extracted from the I and Q outputs using a non-linear discriminator function, as described by Equation 5.4. Then, the same method as that of the non-coherent vector architecture (discussed later) is followed to form a test statistic for the centralized coherent architecture. However, it is not clear at this point how the protection levels will be calculated for this architecture. This is because the inputs to the navigation filter are correlator outputs (six per channel) which are non-linear functions of τ and Δf_d and, therefore, non-linear functions of the fault.

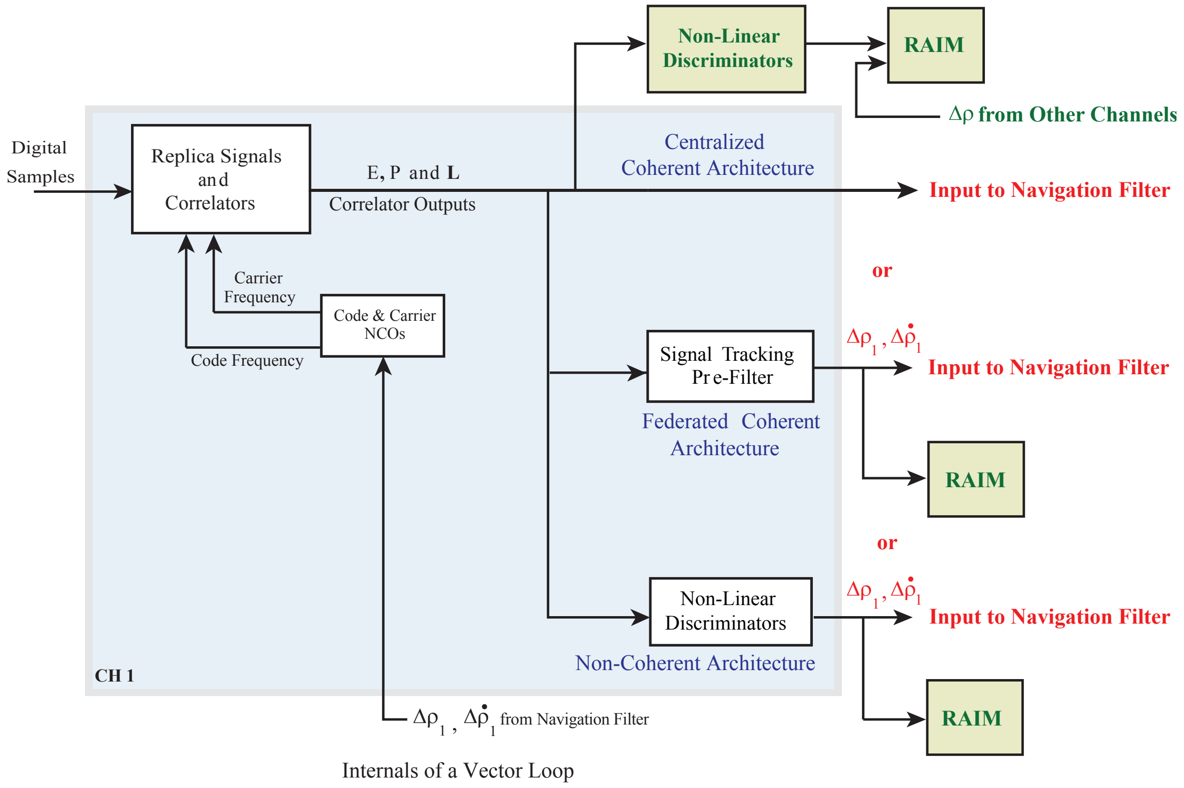


Figure 5.6: Vector Loop RAIM

A second variant of the coherent architecture is called federated coherent architecture. Despite having some loss in performance, it is usually preferred to a centralized one. This is because it partially gets rid of the high computational load of the centralized formulation by distributing computations into a two-stage cascaded structure. Depicted in Figure 5.6 by the top white block on the right, a bank of signal tracking pre-filters - one for each channel - precedes the navigation filter in the federated architecture. They receive the I and Q measurements at 50 Hz and output the pseudorange and pseudorange rate residuals at a slower rate (typically 1 - 2 Hz) to the navigation filter. The pre-filter pseudorange residuals and their error variances can be utilized to design a fault detection algorithm. In other words, these residuals and their error variances are substituted in Equations 5.7 and 5.12 for $\Delta\underline{\rho}_k$ and W_k, respectively, to obtain a test statistic for the federated coherent architecture. Note that, in order to reduce noise, the residuals may need to be averaged over a period of time before they are used for fault detection. The various issues pertaining to the averaging operation are discussed under the non-coherent architecture below. The protection level calculations will also be the same as the non-coherent architecture.

5.3.2.2 Non-Coherent Architecture

In the non-coherent architecture the I and Q measurements are fed to a pair of non-linear discriminator functions (see Figure 5.6) which generate independent estimates of τ and Δf_d (see Equation 5.4 for τ). Recalling the one-to-one correspondence between τ and the pseudorange residuals, τ (i.e., the code phase discriminator outputs) in meters is used to form a test statistic for the non-coherent vector architecture. In this regard, the code phase discriminator outputs (converted into meters) of all channels are stacked into a vector $\Delta\underline{\bar{\rho}}_m$, with m being the time index. The advantages of $\Delta\underline{\bar{\rho}}_m$ are twofold: It is a vector of pseudorange residuals and complies with the single fault assumption of RAIM. However, unlike its scalar loop counterpart $\Delta\underline{\rho}_m$, it is corrupted by large noise. The reason for this is that $\Delta\underline{\bar{\rho}}_m$ is derived from the input of the navigation filter before the navigation filter would bring its noise levels down. Therefore, to reduce noise so as to make $\Delta\underline{\bar{\rho}}_m$ on par with $\Delta\underline{\rho}_m$, the following method is used.

Similar to the non-coherent scalar architecture described previously, the discriminator outputs in the vector architecture are produced every T_a ms. M such outputs (i.e., M samples of $\Delta\underline{\bar{\rho}}_m$) are averaged over MT_a ms, where $m = j - M + 1, \ldots, j$ and the difference between the time points of the first and last samples is t_k - t_{k-1} $= MT_a \times 10^{-3}$ sec$= T$. Thus, the interval between t_k and t_{k-1} has M samples and, therefore, $k = j/M + 1$. M is a trade off between LOS dynamics and noise levels. The receiver noise variance of the i^{th} element of the averaged vector $(\Delta\underline{\hat{\rho}}_k)$ is $(l/M)\sigma^2_{\bar{\zeta}_{i,k}}$. Recall that $\sigma^2_{\bar{\zeta}_{i,k}}$ is the averaged normalized discriminator input noise variance described in relation to Equation 5.4 earlier and l is the number of discriminator outputs averaged in scalar loops. It is also important to observe that the input noise $(\underline{v}_m)$ of $\Delta\underline{\bar{\rho}}_m$ is white i.e., it does not have any colored components or dynamic stress. Therefore, if M is chosen such that $(l/M)\sigma^2_{\bar{\zeta}_{i,k}}$ is less than the white noise jitter plus colored noise jitter of the scalar loops, $\Delta\underline{\hat{\rho}}_k$ will have lower noise levels than $\Delta\underline{\rho}_k$. But the choice of a large M is always constrained by high LOS dynamics. This is explained by writing the mathematical expression of $\Delta\underline{\hat{\rho}}_k$ below

$$\begin{aligned}
\Delta\underline{\hat{\rho}}_k &= 1/M \sum_{m=j-M+1}^{j} \Delta\underline{\bar{\rho}}_m \\
&= 1/M \sum_{m=j-M+1}^{j} [H_m(X_{p_m} - \tilde{X}_{p_m}) \; + \; \underline{v}_m]
\end{aligned}$$

$$= 1/M \sum_{m=j-M+1}^{j} [H_m \Delta X_{p_m} + \underline{v}_m] \tag{5.26}$$

where H_m and ΔX_{p_m} are defined after Equation 5.1. If both ΔX_{p_m} and H_m in the above equation change significantly during the averaging interval, the mathematical model of $\Delta\underline{\bar{\rho}}_k$ cannot be retained in $\Delta\underline{\hat{\rho}}_k$, needed to form the test statistic. Since the velocity is updated multiple times between two subsequent position updates i.e., over the averaging interval T, velocity error can be assumed to move up and down the zero line unless there is a modeling error. As a result, ΔX_{p_m} remains almost constant over the averaging interval. If the constant vector is ΔX_{p_j}, the above equation becomes

$$\Delta\underline{\hat{\rho}}_k = 1/M[\sum_{m=j-M+1}^{j} H_m]\Delta X_{p_j} + 1/M \sum_{m=j-M+1}^{j} \underline{v}_m \tag{5.27}$$

To replace the term $[\sum_{m=j-M+1}^{j} H_m]$ with H_j, the following condition is imposed on the value of M. Considering only the first order variations of the LOS vectors i.e., constant $\dot{H}$, Equation 5.27 reduces to

$$\begin{aligned}
&\approx H_j \Delta X_{p_j} + 1/M[-\dot{H}T_a\{\Delta X_{p_j} + 2\Delta X_{p_j} + \ldots + (M-1)\Delta X_{p_j}\} + \sum_{m=j-M+1}^{j} \underline{v}_m] \\
&= H_j \Delta X_{p_j} + 1/M[-\dot{H}T_a\{1 + 2 + \ldots + (M-1)\}\Delta X_{p_j} + \sum_{m=j-M+1}^{j} \underline{v}_m] \\
&= H_j \Delta X_{p_j} + 1/M[-(\dot{H}T_a)M\frac{(M-1)}{2}\Delta X_{p_j} + \sum_{m=j-M+1}^{j} \underline{v}_m] \\
&= H_j \Delta X_{p_j} - (\dot{H}T_a)\frac{(M-1)}{2}\Delta X_{p_j} + 1/M \sum_{m=j-M+1}^{j} \underline{v}_m \qquad (5.28)
\end{aligned}$$

To ensure that the first term of the above equation is much greater than the second term, the following condition is imposed: $\|H_j\| >> (M-1)/2\|\dot{H}T_a\|$, where $\|.\|$ is the matrix L_2 norm which is the maximum singular value of a matrix. From this an *empirical* upper bound on M is determined as $M_{u_1} = 2 \times (1/100)\|H_k\|/\|\dot{H}T_a\| + 1$.

Next, an upper bound on M is derived by taking into account a more stringent condition that due to unmodeled dynamics velocity error is biased i.e., it moves back and forth a constant value and remains bounded over an averaging interval. Bounded velocity error is achieved by injecting fictitious process noise. Using $H_m = H_j - \dot{H}(j -$

$m)T_a$, Equation 5.26 becomes

$$\begin{aligned}
\Delta\underline{\hat{\rho}}_k &= 1/M \sum_{m=j-M+1}^{j} [H_m \Delta X_{p_m} \; + \; \underline{v}_m] \\
&= \frac{1}{M} \sum_{m=j-M+1}^{j} [H_j - \dot{H}(j-m)T_a]\Delta X_{p_m} \; + \; \frac{1}{M} \sum_{m=j-M+1}^{j} \underline{v_m} \\
&= \frac{1}{M} \times H_j \sum_{m=j-M+1}^{j} \Delta X_{p_m} - \frac{1}{M} \times \dot{H}T_a[(M-1)\Delta X_{p_{j-M+1}} + \\
&\qquad (M-2)\Delta X_{p_{j-M+2}} + \ldots + \Delta X_{p_{j-1}}] + \frac{1}{M} \sum_{m=j-M+1}^{j} \underline{v_m}
\end{aligned}$$

Substituting $\Delta X_{p_m} = \Delta X_{p_j} - (j-m)\underline{b}T_a$, where $\underline{b}$ is the upper bound on the velocity error, one can write

$$\begin{aligned}
\Delta\underline{\hat{\rho}}_k &= \frac{1}{M} \times H_j \sum_{m=j-M+1}^{j} \left[M\Delta X_{p_m} - \frac{M(M-1)}{2}\underline{b}T_a \right] \; + \\
\frac{1}{M} \times \dot{H}T_a & \left[-\frac{M(M-1)}{2}\Delta X_{p_m} \; + \; \frac{M(M-1)(2M-1)}{6}\underline{b}T_a \right] + \frac{1}{M} \sum_{m=j-M+1}^{j} \underline{v_m} \qquad (5.29)
\end{aligned}$$

To make sure the second term is negligible compared to the first term so that the mathematical model $H_j \sum_{m=j-M+1}^{j} \Delta X_{p_m}$ can be preserved, M is chosen such that $\|H_j\| >> (M-1)/2 \times \|\dot{H}T_a\|$ and $1/2 \times \|H_j\| >> (2M-1)/6 \times \|\dot{H}T_a\|$. If $M'_{u_1} = 2 \times (1/100)\|H_j\|/\|\dot{H}T_a\| + 1$ and $M''_{u_1} = 1/2 \times (3/(100)\|H_j\|/\|\dot{H}T_a\| + 1)$, the upper bound on M (M_{u_1}) is given by $\min\{M'_{u_1},\ M''_{u_1}\} = M''_{u_1}$. It should be noted that M'_{u_1} is the same as previously defined M_{u_1}.

Further, note that both the test statistic and the user position are estimated at time interval MT_a. In the vector architecture the estimated user position is also fedback to correct any code phase alignment error of the replica C/A-code with the received C/A-code. Hence, a second upper bound on M is given by the minimum required update rate or maximum update interval of the code NCOs. For deriving this bound the LOS vectors are assumed to be constant; the position error is assumed to be increasing with time; and the velocty error is assumed to remain mostly on one side of the zero line. This bound accounts for a bias in the velocity error which is smaller than its standard deviation. The minimum update rate is determined using the assumption that the maximum code phase error is one tenth of the early-late correlator spacing d (discriminator pull-in range). Let $\sigma_{d_{\text{code, i}}}$ be the standard deviation of the error in Hz in the estimated code Doppler frequency of channel i and $\sigma_{p_{\mathrm{code,\ i}}}$ be the correponding standard deviation of the code phase error in sec after a position update. Then the maximum update interval $T_{\max}$ is given by $\min_i\{d/10/\text{code chipping rate} - \sigma_{p_{\mathrm{code,\ i}}}) \times (\text{code chipping rate}/\sigma_{d_{\mathrm{code,\ i}}})\}$, where $(\sigma_{d_{\mathrm{code,\ i}}}/\text{code chipping rate})$ is the i^{th} code phase alignment error in sec over 1 sec [69]. So, $T_{\max}$ is the time in which the error in the estimated Doppler will misalign the replica and received codes of one of the channels by $d/10$ chips. Thus, a second upper bound on M is $M_{u_2} = T_{\max}/T_a$ and the final upper bound on M is $\min\{M_{u_1}, M_{u_2}\}$. In addition to the above bounds, a lower bound on M is chosen by trial and error such that the position update interval is at least 1 sec. It is apparent that at high $\mathrm{C/N_0}$ M will remain well between the prescribed bounds. However, as $\mathrm{C/N_0}$ degrades, M needs to increase, but M_{u_2} will decrease, placing a constraint on M. It is important to note that the Doppler frequencies are estimated by projecting the estimated user velocity and clock drift on individual LOS. Therefore, $\sigma_{d_{\mathrm{code}},i}$ is the standard deviation of the projection of the error vector in the estimated velocity and clock drift on i^{th} LOS. $\sigma_{p_{\mathrm{code,\ i}}}$ is found the same way using the position and clock bias error standard deviations.

The averaged $\Delta\underline{\hat{\rho}}_k$ and its error covariance are substituted in Equations 5.7 and 5.12 to form a test statistic for vector loops. Note that other scalar loop-based RAIM algorithms (e.g., parity space RAIM) can also be used with $\Delta\underline{\hat{\rho}}_k$. Furthermore, the pseudorange residual model of $\Delta\underline{\hat{\rho}}_k$ ensures the same relationship between the statistic and position error as in scalar loops when a failure occurs in a satellite. The previous definition of protection levels and the method for calculating them also remain valid for this formulation except for the following change that needs to be made to account for different position and velocity update rates: Different update rates are required to reduce noise in the code phase discriminators for fault detection purposes, as elucidated above.

Modification to the Calculation of Vector Loop Position Error Bounds: The method to bound the means of VPE and HPE discussed under the scalar loop RAIM assumes that user position and velocity are updated at the same rate. Unfortunately, in order to perform integrity monitoring, this assumption needs to be relaxed for the vector architecture for which the same method applies otherwise. The vector loop position update interval, however, is a few seconds and the velocity update interval is a fraction of a second (0.1 sec in this thesis). Hence, it can be assumed that the mean velocity error between two position update intervals remains almost constant at the value corresponding to the previous position update epoch, although it is not accurate. However, this enables implementation of the previous formulation for different position and velocity update rates.

Alternatively, the following modification can be applied to Equation 5.16 to account for different update rates. It complies with the previous assumption of piecewise ramp faults. All notations defined under the scalar loop RAIM hold except that the subscript "stl" is changed to "vtl." Recall that the fault in satellite i started at t_m. Thus, the mean state error at t_m is given by

$$\underline{\mu}_m = -K_{\mathrm{vtl}_m}\underline{f}_{i,\,1} \tag{5.30}$$

where $\underline{f}_{i,\,1}$ is a column vector such that $\underline{f}_{i,\,1}(i) = b_1$; $\underline{f}_{i,\,1}(i+n) = \dot{b}_1$ and $\underline{f}_{i,\,1}(j) = 0$ for $j \neq i$ and $j \neq i+n$; n is the number of visible satellites. b_1 is a bias at t_m and $\dot{b}_1$ is its rate of change. Collecting the velocity terms of the above equation yields

$$\underline{\mu}_{m,\,v} = -K_{\mathrm{vtl}_m,\,v_1} b_1 q_i - K_{\mathrm{vtl}_m,\,v_2}\dot{b}_1 q_i \tag{5.31}$$

where q_i is an $n \times 1$ column vector whose i^{th} row is 1 and all other rows are zero. If there are p velocity updates between t_m and t_{m+1}; $(K_{\mathrm{vtl}_m,\,v_2} C_{m,\,v})$ is not changed significantly during velocity updates; and $\dot{b}_1$ remains approximately constant over $T = (t_{m+1} - t_m)$, then the mean velocity error $\underline{\mu}_{m_p,\,v}$ after p velocity updates is recursively obtained as

$$\begin{aligned}\underline{\mu}_{m_p,\,v} &= (I - K_{\mathrm{vtl}_m,\,v_2} C_{m,\,v})^p(-K_{\mathrm{vtl}_m,\,v_1} b_1 q_i)\\ &\quad + \left(I + \sum_{j=1}^{p}(I - K_{\mathrm{vtl}_m,\,v_2} C_{m,\,v})^j\right)(-K_{\mathrm{vtl}_m,\,v_2}\dot{b}_1 q_i)\end{aligned} \tag{5.32}$$

where $C_{m,\,v}$ contains the portion of the measurement model matrix C_m corresponding to the velocity states and pseudorange rate measurements. User velocity is modeled as a constant vector, resulting in the identity state transition matrix for the velocity vector. Since the eigenvalues of $(I - K_{\mathrm{vtl}_m,\,v_2} C_{m,\,v})$ are less than one for this implementation of the EKF, $(I - K_{\mathrm{vtl}_m,\,v_2} C_{m,\,v})^p \to 0$ if p is sufficiently large. Thus, the above equation reduces to

$$\begin{aligned}\underline{\mu}_{m_p,\,v} &\approx \left(I + \sum_{j=1}^{p}(I - K_{\mathrm{vtl}_m,\,v_2} C_{m,\,v})^j\right)(-K_{\mathrm{vtl}_m,\,v_2}\dot{b}_1 q_i)\\ &= A'_m(-K_{\mathrm{vtl}_m,\,v_2}\dot{b}_1 q_i)\end{aligned} \tag{5.33}$$

As the mean velocity error changes over T, the change in the mean position error over this time interval can be obtained as $(\delta t/2)\underline{\mu}_{m,\,v} + (T - \delta t/2)(\underline{\mu}_{m_p,\,v})$, assuming the mean velocity error changes almost linearly over δt before reaching the steady state value $\underline{\mu}_{m_p,\, v}$ i.e., $(I - K_{\mathrm{vtl}_{m},\, v_2} C_{m,\, v})^j \rightarrow 0$ in δt. δt is chosen such that the eigenvalues of $(I - K_{\mathrm{vtl}_{m},\, v_2} C_{m,\, v})$ are at least 100 times the corresponding eigenvalues of $(I - K_{\mathrm{vtl}_{m},\, v_2} C_{m,\, v})^j$ and is found to be inversely related to C/N$_0$. Note that p should be large to ensure that $(I - K_{\mathrm{vtl}_{m},\, v_2} C_{m,\, v})^p \approx [0]_{n\times n}$. The value of p is in the range of 20 to 60 in the vector loop implementation of this thesis where velocities are updated at 0.1 sec and position update intervals vary from 2 sec to 6 sec depending on C/N$_0$. At very low C/N$_0$ ($\backsim$ 15 dB-Hz) δt is sometimes found to be greater than T and assumed equal to T then. But the $(I - K_{\mathrm{vtl}_{m},\, v_2} C_{m,\, v})^p(-K_{\mathrm{vtl}_{m},\, v_1})$ term, being small, can still be ignored for this case.

Next with both position and velocity updates the mean state error at t_{m+1} is

$$\underline{\mu}_{m+1} = -K_{\mathrm{vtl}_{m+1}} \underline{f}_{i,\, 2} + (I - K_{\mathrm{vtl}_{m+1}} C_{m+1}) \begin{bmatrix} 1 & \delta t/2 & & \dots & 0 \\ 0 & 0 & & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & \vdots & \delta t/2 \\ 0 & 0 & & \dots & 0 \end{bmatrix} (-K_{\mathrm{vtl}_{m}} \underline{f}_{i,\, 1}) +$$

$$(I - K_{\mathrm{vtl}_{m+1}} C_{m+1}) \begin{bmatrix} 0 & T - \delta t/2 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \vdots & T - \delta t/2 \\ 0 & 0 & \dots & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A'_m(1,\, 1) & 0 & A'_m(1,\, 2) & 0 & A'_m(1,\, 3) & 0 & A'_m(1,\, 4) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A'_m(2,\, 1) & 0 & A'_m(2,\, 2) & 0 & A'_m(2,\, 3) & 0 & A'_m(2,\, 4) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A'_m(3,\, 1) & 0 & A'_m(3,\, 2) & 0 & A'_m(3,\, 3) & 0 & A'_m(3,\, 4) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A'_m(4,\, 1) & 0 & A'_m(4,\, 2) & 0 & A'_m(4,\, 3) & 0 & A'_m(4,\, 4) \end{bmatrix} (-K'_{\mathrm{vtl}_{m}} \underline{f}_{i,\, 1})$$

$$\begin{aligned} \text{or,}\quad \underline{\mu}_{m+1} &= (I - K_{\mathrm{vtl}_{m+1}} C_{m+1}) F'_m \underline{f}_{i,\, 1} - K_{\mathrm{vtl}_{m+1}} \underline{f}_{i,\, 2} \\ &= \underline{\mu}'_{m+1} - K_{\mathrm{vtl}_{m+1}} \underline{f}_{i,\, 2} \end{aligned} \tag{5.34}$$

The difference between K'_{vtl_m} and K_{vtl_m} is that in K'_{vtl_m} the elements of the velocity rows of K_{vtl_m} corresponding to the pseudorange measurements are zero. This is needed because Equation 5.33 shows that the mean velocity error after p velocity updates depends only on the bias rate $\dot{b}_1$. It should be noted that the second component of F'_m first accounts for p velocity updates in T. The above arrangement of the elements of A'_m in F'_m is required because, in the filter, each position state is followed by the corresponding velocity state. The change in the mean position error is calculated. Following the same steps as with $\underline{\mu}_{m_p,\,v}$, $\underline{\mu}_{m+1_p,\,v}$ results as

$$\begin{aligned} \underline{\mu}_{m+1_p,\,v} &= (I - K_{\mathrm{vtl}_{m+1},\,v_2} C_{m+1,\,v})^p (\underline{\mu}'_{m+1,\,v} - K_{\mathrm{vtl}_{m+1},\,v_1} b_2 q_i) + A'_{m+1}(-K_{\mathrm{vtl}_{m+1},\,v_2} \dot{b}_2 q_i) \\ &\approx A'_{m+1}(-K_{\mathrm{vtl}_{m+1},\,v_2} \dot{b}_2 q_i) \end{aligned} \tag{5.35}$$

The change in mean position error is $(\delta t/2)\underline{\mu}_{m+1,\,v} + (T - \delta t/2)(\underline{\mu}_{m+1_p,\,v})$. Thus, $\underline{\mu}_{m+2}$ is given by

$$\begin{aligned} \underline{\mu}_{m+2} =& (I - K_{\mathrm{vtl}_{m+2}} C_{m+2}) F''_{m+1} (I - K_{\mathrm{vtl}_{m+1}} C_{m+1}) F'_m \underline{f}_{i,\,1} - \\ & (I - K_{\mathrm{vtl}_{m+2}} C_{m+2}) F'_{m+1} \underline{f}_{i,\,2} - K_{\mathrm{vtl}_{m+2}} \underline{f}_{i,\,3} \end{aligned} \tag{5.36}$$

where F''_{m+1} in the first term is the same as the state transition matrix F_{m+1} except that the element T is replaced with $\delta t/2$ and the elements of its velocity rows are all zero. This is because the mean velocity error before the measurement update at t_{m+2} is given by the second term; no contribution comes from the first term (see Equation 5.35). Repeating the above steps and replacing F_{k-1} in A_k's of Equation 5.16 with F''_{k-1}, one can write

$$\begin{aligned} \underline{\mu}_k =& \prod_{j=k}^{m+2} A_j (I - K_{\mathrm{vtl}_{m+1}} C_{m+1}) F'_m \underline{f}_{i,\,1} + \\ & \prod_{j=k}^{m+3} A_j (I - K_{\mathrm{vtl}_{m+2}} C_{m+2}) F'_{m+1} \underline{f}_{i,\,2} + \dots \\ & + (I - K_{\mathrm{vtl}_k} C_k) F'_{k-1} \underline{f}_{i,\,k-m} + (-K_{\mathrm{vtl}_k} \underline{f}_{i,\,k-m+1}) \\ =& \prod_{j=k}^{m+2} A_j (-K''_{\mathrm{vtl}_m} \underline{f}_{i,\,1}) + \prod_{j=k}^{m+3} A_j (-K''_{\mathrm{vtl}_{m+1}} \underline{f}_{i,\,2}) + \dots \\ & + (-K''_{\mathrm{vtl}_{k-1}} \underline{f}_{i,\,k-m}) + (-K_{\mathrm{vtl}_k} \underline{f}_{i,\,k-m+1}) \end{aligned} \tag{5.37}$$

where $K''_{\mathrm{vtl}_j} = -(I - K_{\mathrm{vtl}_{j+1}} C_{j+1}) F'_j$. The above equation is used to find the bounds of the mean VPE and HPE instead of Equation 5.16 for different position and velocity update rates.

5.4 Simulation Studies

In order to validate the vector loop RAIM algorithm discussed above, simulation studies involving pseudolites used in navigation of a ground vehicle (i.e., two dimensional motion) are performed. Simulation studies are resorted to because, to begin with, the vector architecture-based RAIM would be complex to analyze with real data. Therefore, at this point two-dimensional simulation studies with a simple, benign geometry will help gain better insights into the integrity monitoring capability of this architecture. The same user-pseudolite geometry as described in Chapter 4 is considered. According to this geometry six pseudolites are evenly distributed around a ground vehicle that is moving along a straight line towards pseudolite 2 (Figure 5.7). A vehicle speed of 40 m/sec, though high for ground application, is chosen to generate fairly large Doppler frequencies that are easy to analyze. The pseudolite locations are fixed and assumed to be precisely known, thereby obviating the need for decoding the navigation message. The pseudolites have the same signal structures as GPS L1 signals. For simulation purposes a high fidelity RF front end simulator has been developed which allows generation of GPS signals with different $\mathrm{C/N_0}$s. Using this simulator, separate 50 dB-Hz and 30 dB-Hz IF data for the user geometry mentioned above are generated. It is also assumed that all pseudolite signals have the same $\mathrm{C/N_0}$ at the receive end, neglecting the signal propagation loss. Only wide band Gaussian thermal noise is added to IF samples to simulate different $\mathrm{C/N_0}$ signals.

To simulate a threat model or failure mode for this geometry, the clock of pseudolite 2 is assumed to exhibit a ramp error that starts in the middle of the simulation and continues till the end. Two drift magnitudes are considered: 0.5 m/sec and 1 m/sec. The onset times of the fault at 50 dB-Hz and 30 dB-Hz are 20 sec and 16 sec, respectively. As it is noticed from the simulation results that a longer time is needed to detect fault at lower $\mathrm{C/N_0}$, simulation durations of 30 sec and 40 sec at 50 dB-Hz and 30 dB-Hz, respectively, are used. Thus, 30 Monte Carlo runs are simulated for each combination of $\mathrm{C/N_0}$ and drift. For fault detection purposes the following RAIM parameters are used: Probability of false alarm $P_{\mathrm{FA}} = 10^{-5}$; probability of HMI $P_{\mathrm{HMI}} = 10^{-7}$; prior probability of pseudolite failure $P_{\mathrm{F}} = 10^{-4}$. To compare the performance of scalar and vector loop RAIM algorithms, the fault detection times from the onset of the fault and the HPLs of the two architectures are calculated.

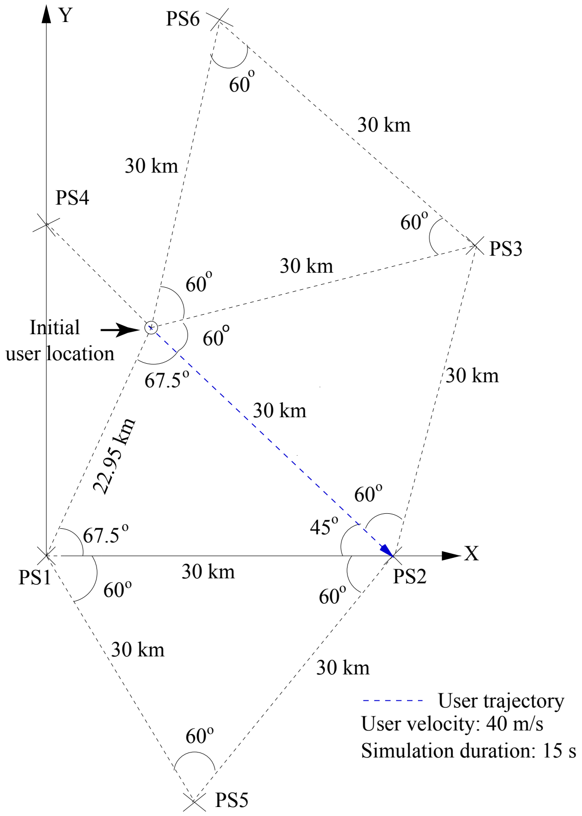


Figure 5.7: Description of User Geometry

5.4.1 Simulation Results

As mentioned earlier, non-coherent scalar and vector architectures are implemented in this thesis. Tables 5.1 and 5.2 list the values of the various scalar and vector loop design parameters that allow a systematic performance comparison of the two tracking loops. A two-sided DLL bandwidth of 2 Hz is assumed. Hence, the scalar loop pseudorange measurements are sampled at a rate equal to the inverse of the DLL bandwidth i.e., 0.5 sec to avoid the effect of time-correlated noise. $\sigma^2_{\bar{\zeta}_{i,k}}$ is the normalized discriminator noise variance defined below Equation 5.4. Since the atmospheric delays, satellite clock jitter or mutipath are not simulated, receiver dynamics (ramp in Doppler) are fully tracked by the second order FLLs, and receiver clock instabilities are small, no colored jitter or dynamic stress is assumed to be present in the scalar loop pseudorange residuals. With $T_a = 20$ ms M_{u_1} is found to be 867. The value of M_{u_2} is 1331 at 50 dB-Hz and 541 at 30 dB-Hz. The mathematical expressions of M_{u_1} and M_{u_2} are given below Equation 5.29. Based on the calculated M_{u_1} and M_{u_2}, M is chosen to be 50 and 150 at 50 dB-Hz and 30 dB-Hz, respectively, which also satisfy the lower bound

Table 5.1: Scalar Loop Design Parameters

Design parameters	STL	
	50 dB-Hz	30 dB-Hz
Discriminator output interval (T_a)	20 ms	20 ms
Discriminator averaging interval (lT_a)	0.1 sec	0.1 sec
Position update rate	1 sec DLL BW ($2B_n$): 2 Hz Pseudorange sample rate (T_ρ): 0.5 sec	3 sec DLL BW: 2 Hz Pseudorange sample rate: 0.5 sec
Velocity update rate	0.5 sec	0.5 sec
Test statistic derived from	Pseudorange residuals	Pseudorange residuals
Statistic update interval or averaging interval of pseudorange residuals (pT_ρ)	1 sec	3 sec
Channel i statistic noise variance at epoch k $\left(\frac{2B_n l T_a \sigma^2_{\bar{\zeta}_{i,k}}}{p}\right)$	$\frac{2 \times 0.1}{2}\sigma^2_{\bar{\zeta}_{i,k}}$	$\frac{2 \times 0.1}{6}\sigma^2_{\bar{\zeta}_{i,k}}$

constraint.

The test statistic is calculated using the snapshot fault detection method described in Section 5.3.1.1. The threshold T of the test statistic is determined from the chi-square table as 5.089 This corresponds to a $P_{\text{FA}} = 10^{-5}$ and DOF = number of pseudolites - number of variables to be estimated = 3. Note that since a two-dimensional motion is simulated, the estimated variables have three elements: The x and y coordinates of user position and clock bias.

Figure 5.8 shows the mean and 1-σ bound of the test statistic as it evolves in time. Table 5.3 summarizes the mean and standard deviation of the fault detection times of the two architectures for each pair of C/N$_0$ and drift. A close observation of the test statistic in Figure 5.8 reveals that, for the threshold chosen in this analysis, there is no statistically significant difference in the fault detection times of the two architectures, as is evident from Table 5.3. It is imperative to note the difference (more noticeable at 30 dB-Hz) in the time at which the mean test statistic crosses

Table 5.2: Vector Loop Design Parameters

Design parameters	VTL	
	50 dB-Hz	30 dB-Hz
Discriminator output interval (T_a)	20 ms	20 ms
Discriminator averaging interval	Code (MT_a): 1 sec Doppler (lT_a): 0.1 sec	Code: 3 sec Doppler: 0.1 sec
Position update rate	1 sec	3 sec
Velocity update rate	0.1 sec	0.1 sec
Test statistic derived from	Discriminator outputs	Discriminator outputs
Statistic update interval (MT_a)	1 sec	3 sec
Channel i statistic noise variance at epoch k ($\frac{l}{M}\sigma^2_{\bar{\zeta}_{i,k}}$)	$\frac{1}{10}\sigma^2_{\bar{\zeta}_{i,k}}$	$\frac{1}{30}\sigma^2_{\bar{\zeta}_{i,k}}$

the threshold and the mean fault detection time reported in Table 5.3. This occurs because the mean fault detection time in Table 5.3 is the mean of all times at which the tests statistics of individual runs cross the threshold. On the other hand, the mean test statistic of Figure 5.8 is obtained by averaging the test statistics of all runs at a particular time epoch. So, for the mean test statistic, averaging is done along the vertical axis whereas for the mean fault detection time averaging is performed along the horizontal threshold.

For calculating the protection level, Equation 5.22 and the slope of the statistic and position error as discussed earlier are used. The mean velocity error between position update intervals is assumed constant i.e., the modification (see Section 5.3.2.2) due to different position and velocity update rates is not applied. As for the vector architecture, any scalar loop parameters are replaced with their vector loop equivalents as it was explained earlier that the scalar loop definition of protection levels and the statistic-position error relationship hold for the vector loop RAIM. Figure 5.9 shows the HPE and the corresponding HPL for both architectures. It is important to note that for both C/N$_0$s the HPL is determined by the second part of Equation 5.22. However,

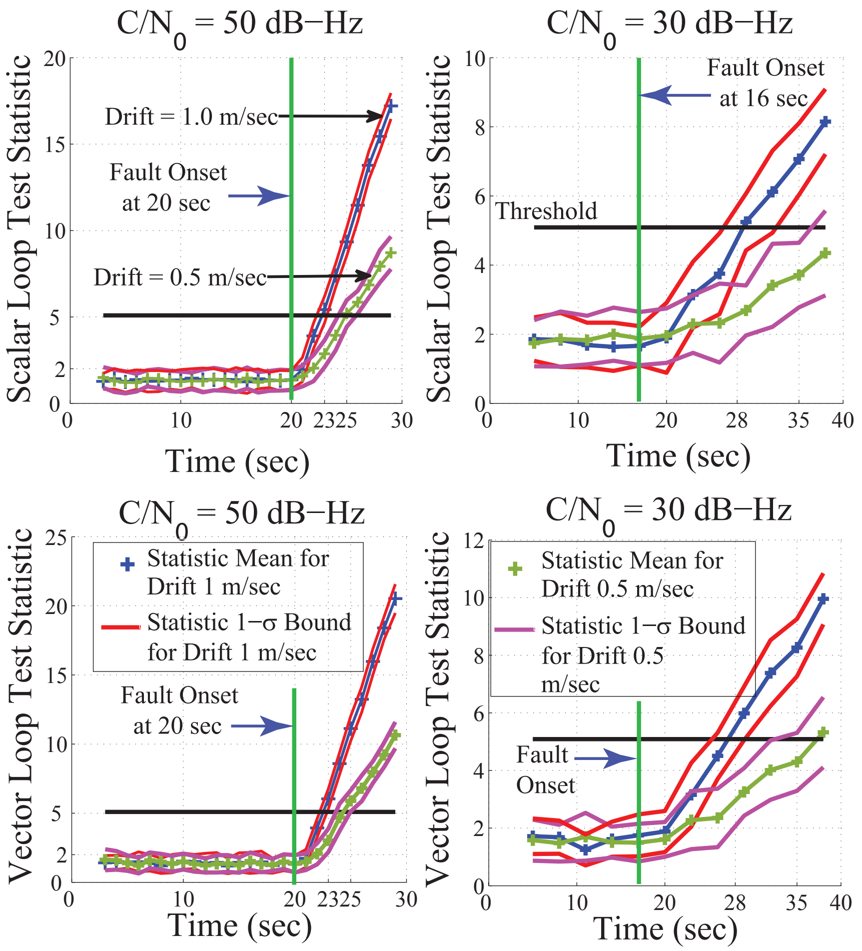


Figure 5.8: Evolution of Test Statistic with Time

the difference in the two HPLs obtained from the first and second parts of Equation 5.22 is small at 50 dB-Hz.

Figure 5.9 also shows the HPE before the fault is detected. Note that the measurement update intervals at 50 dB-Hz and 30 dB-Hz are 1 sec and 3 sec, respectively and the HPEs are calculated accordingly. As a result, the HPEs are spaced apart at 30 dB-Hz by 3 sec whereas because of faster update rates they appear to be clustered together at 50 dB-Hz. Also note that the HPE does not cross the HPL in any of the simulations for both architectures. Figures 5.8 and 5.9 illustrate that by suitably choosing the design parameters of the scalar and vector architectures, similar integrity monitoring performance of the two architectures is achieved. However, it should be noted that the receiver clock jitter of the scalar loops is not accounted for in this preliminary study because simulated clock instabilities are small. Accounting for receiver clock jitter may result in different protection levels of the two architectures especially

Table 5.3: Fault Detection Times Since the Onset of Fault

Drift (m/sec)	STL FDT (sec) (mean ± std dev)		VTL FDT (sec) (mean ± std dev)	
	50 dB-Hz	30 dB-Hz	50 dB-Hz	30 dB-Hz
0.5	5.23 ± 0.63	19.26 ± 3.43 (no detection in 16 runs)	4.76 ± 0.63	18.83 ± 3.63 (no detection in 12 runs)
1.0	3.27 ± 0.52	14.1 ± 2.01	3 ± 0.45	12 ± 2.13

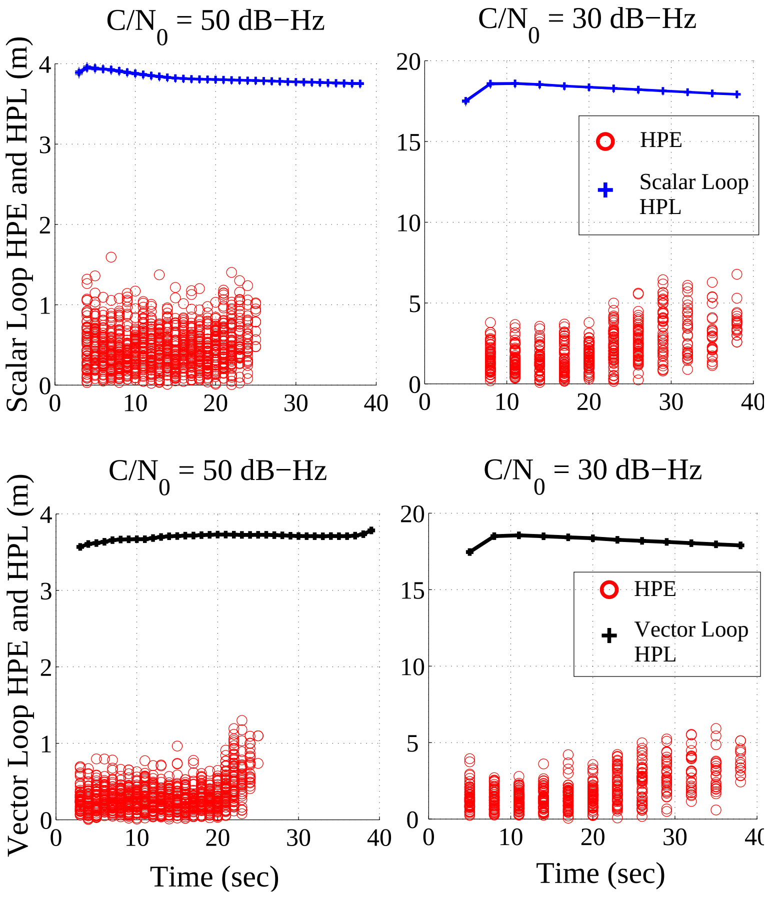


Figure 5.9: HPL and HPE of Scalar and Vector Loops

at higher C/N$_0$. In this context, it is important to observe that the scalar loop HPEs are larger than the vector loop HPEs, although their white input noise covariances are matched to each other. Calculating the variance of the HPE from all Monte Carlo runs, it is found that the scalar loop sample variance is higher than the EKF-estimated variance. For the vector loops the sample variance and EKF values are consistent. Using the Scalar loop sample variance for protection levels, an increase of about 0.9 m in the protection level is noted at 50 dB-Hz. At 30 dB-Hz the increase is about 0.3 m. A possible explanation for such a discrepancy in the scalar loop EKF variance could be that the scalar loop clock jitter was not taken into account, whose effect, being small in this simulation, is more pronounced at higher C/N$_0$. During the validation of the vector loop RAIM algorithm with GPS measurements in the next chapter, the colored jitter of the scalar loops will be computed.

5.5 Conclusion

The scalar loop RAIM is not directly applicable to the vector architecture as the pseudorange measurements output from a vector tracking receiver are all affected by a failure in one channel. In this chapter an alternative way to apply ideas from scalar loop RAIM to the vector architecture is developed. This is achieved by deriving the input to the RAIM algorithm from the input (i.e., the correlator outputs) of the vector loop navigation filter rather than from its output (i.e., pseudorange measurements). In addition, protection level calculations are modified for the EKF implementation of the navigation filter. Next with simulation studies and proper design of the pertinent scalar and vector loop parameters, similar fault detection performance of both architectures is demonstrated.

The next step is validate this algorithm with GPS measurements which will help make a definitive statement about the integrity monitoring capability of the vector loops. In addition, the following aspects associated with vector loop RAIM will be explored: The gain in vector loop RAIM performance because of the ability of the vector loops to bridge through momentary satellite blockage and the issues of large noise involved with vector loop RAIM. Large noise may, in fact, preclude the use of vector loop RAIM at very low C/N$_0$, even though the vector loops continue to track signals at such low C/N$_0$. Therefore, the prospect of using vector loop RAIM at low C/N$_0$ requires further analysis and will be carried out in greater detail in the next chapter.

Chapter 6

Validation of Vector Loop RAIM with GPS Measurements

In this chapter the vector loop integrity monitoring algorithm developed in the previous chapter is validated with GPS measurements. For this purpose, a scenario of an aircraft approach is created using Spirent's SimGEN software package [90]. The scenario simulates an aircraft making a non-precision or lateral navigation (LNAV) approach to a runway. Only the final approach phase of the aircraft is simulated. The simulation duration is about 5 minutes. The RF signals are assumed to be healthy during the first one and a half minutes of the simulation, thereby ensuring that the receiver tracking loops have sufficient time to reach steady state. Over the remaining duration, the receiver is subjected to adverse conditions resulting from pseudorange drifts with/without elevated noise floor caused by intentional interference. This enables testing the receiver performance under deep signal fades and signal anomalies.

This chapter is organized as follows: First, the motivation behind scenario selection is discussed. Then, the key features of the simulation including generation of low C/N_0 data are discussed. Subsequently, results with the high and low C/N_0 data are presented, which highlight key elements of vector loops and vector loop RAIM performance in GPS-distressed environments.

6.1 Motivation for Scenario Selection

As noted above, the final approach phase of an aircraft making an instrument approach to a runway is considered for validation of vector loop integrity monitoring. The safety-critical application of an aircraft approach is chosen because well-defined

RAIM parameters are available for this application. Additionally, the noise floor is assumed to be elevated halfway through the approach due to intentional interference. In their November 2010 report, the national space-based PNT advisory board committee of the USA described interference to GPS as a serious threat and called for immediate countermeasures to locate, mitigate and shut down a source of interference [16]. Among a number of recommended actions to counter the threat of RF interference, they proposed hardening of GPS antennas and receivers which includes algorithms that can quickly recover from RF interference. In the report it was emphasized that these receivers should be equipped with integrity algorithms to “insure they do not display Hazardous and Misleading Information during periods of interference.” Within this context, low C/N_0 data during an approach is generated in this study by adding wideband Gaussian noise caused by intentional interference. These data sets are used to study the vector loop integrity monitoring algorithm developed in this thesis.

While incidents of deliberate jamming of GPS receivers on-board an aircraft are unknown to the author, an indepedent GPS risk assessment study conducted by the John Hopkins University Applied Physics Laboratory (JHU/APL) concluded that its possibility cannot be dismissed unless advanced mitigation techniques are employed [92]. The six-month long study was carried out under cosponsorship from the Federal Aviation Administration (FAA), the Air Transport Association (ATA) and the Aircraft Owners and Pilots Association (AOPA). In the study, an example aircraft trajectory was created at John F. Kennedy Airport, New York and the GPS receiver technology was assumed consistent with the Wide Area Augmentation System (WAAS) specification. The analysis showed that a hundred watt jammer placed under the flight path at approximately 20 nautical miles away from the landing point will deny GPS for the entire approach and landing trajectory and a one watt jammer can deny GPS within a ten-nautical mile radius.

The simuated approach path considered in this study and injection of fault and noise along the approach are depicted in Figure 6.1. The entire approach path is shown in black in the top sub-figure. Only the green segment of the path i.e. the final approach phase is simulated in this study. The bottom sub-figure shows the injection of fault and one case of elevated noise floor along the approach course. Four cases of elevated noise floor are considered in this analysis, which are discussed later in this chapter.

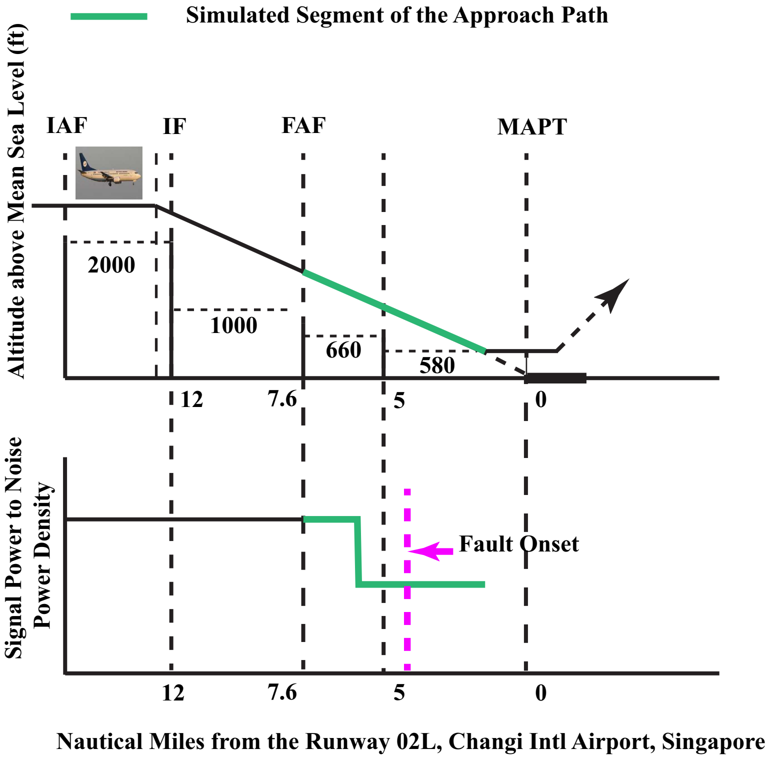


Figure 6.1: Simulated Approach Course and Signal Anomalies along the Approach

6.2 Key Characteristics of Simulation

Important features of the simulation are as follows:

- Date of simulation: 17th April, 2010.
- Time: (7:20 - 7:25:11) pm.
- Aircraft: Cessna Caravan.
- Location: Changi International Airport, Singapore.
- Aircraft Trajectory: The final approach phase to runway 02L (from final approach fix to decision height) is described by a number of waypoints in the SimGEN scenario. The instrument approach chart for this runway is provided in Appendix D.
- GPS satellite orbits and clocks are simulated using the navigation message for the above date as obtained from a continuously operating reference station (CORS).

- Atmospheric delays: The tropospheric delay is modeled by the STANAG model. The model type for the ionospheric delay is chosen as Terrestrial [90].
- The aircraft antenna pattern is input as a comma separated variable (CSV) file in SimGEN. It is generated by combining a Cessna Caravan's roll and pitch plane antenna patterns given in [91]. The patterns are illustrated in [91] for four different azimuth angles. To determine the antenna gains for all values of azimuth and elevation, the following is done. Each of the four vertical halves - two from each of roll and pitch plane patterns - determines a quarter segment of the entire 3D pattern. That is, each quarter segment of the 3D antenna pattern is obtained by revolving the corresponding vertical half about its vertical axis by 45^o on either side of its horizontal axis.
- As for signal anomaly or fault, pseudorange ramps of 0.5 m/s and 1 m/s are injected at two and a half minutes into the simulation.
- C/N_0 of the healthy signals varies between (44 - 49) dB-Hz. In order to simulate intentional interference, the noise floor is elevated at various points in the approach by adding artificial Gaussian noise to the collected IF data.
- IF data was collected using a custom receiver front end with an IF of 12.012 MHz and a sampling frequency of 43.428 MHz. The two-sided front end bandwidth is approximately 21 MHz.
- The limitations of the simulation are that the vibration-induced PSD of the receiver clock and multipath signals are not simulated.

GPS satellite visibility during the approach is shown in Figure 6.2. A total of six data sets are collected - two without fault, two with a ramp fault of slope 1 m/s and two with a ramp fault of slope 0.5 m/s. The fault is injected into the highest elevation satellite PRN 28 because it was judged that it would have the most detrimental effect on protection levels. In all data sets signals have C/N_0 in the range of 44 dB-Hz to 49 dB-Hz. Low C/N_0 data are then generated by adding artificial Gaussian noise to the digital IF data, as discussed below.

6.2.1 Low C/N_0 Data

It is assumed that the GPS receiver is subject to intentional interference halfway through the approach. This results in elevated noise floor in the data. In this analysis

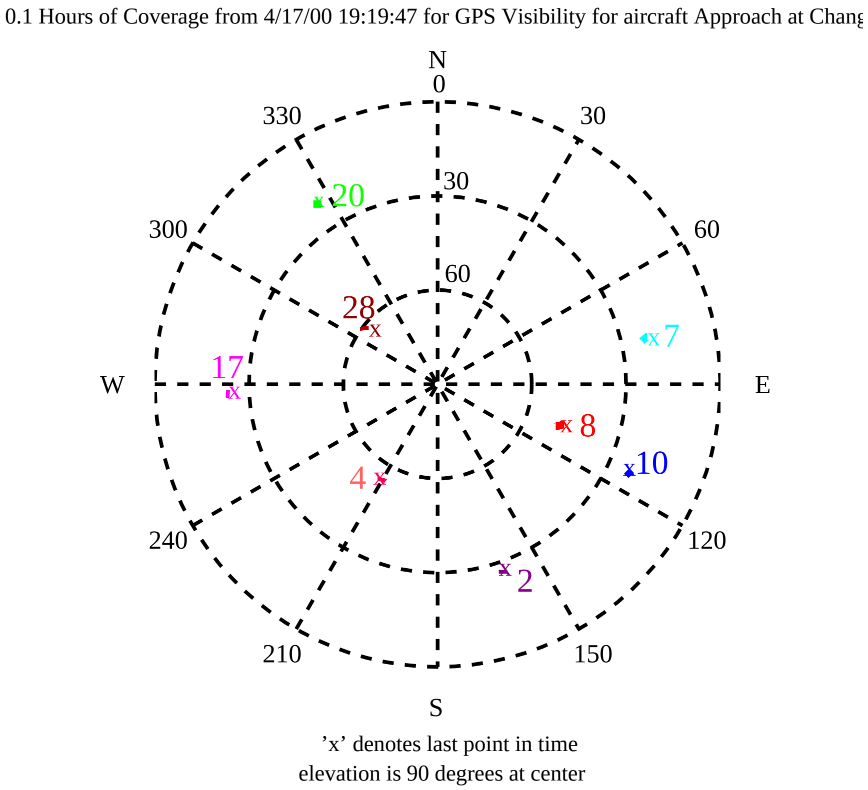


Figure 6.2: Skyplot of GPS Satellite Visibility during Approach

four cases of signal degradation/elevated noise floor and their effect on scalar and vector loop RAIMs are evaluated. The degraded C/N$_0$s are illustrated in Figure 6.3. Cases I and II have a C/N$_0$ degradation of about 20 dB-Hz while cases III and IV assume a degradation of 30 dB-Hz. In cases I and III the noise floor is enhanced about a minute before fault initiation and held constant at the low value until the end. In cases II and IV C/N$_0$ is first ramped down to a low value; held constant at that value for about a minute; and then ramped back up to the original level. Restoration of signal power is immediately followed by fault onset in these cases. The change in the C/N$_0$ level is assumed to take place over a period of six seconds in all cases. Five Monte Carlo runs are created for each of cases I and III - one without fault and four with faults (two with 1 m/s ramp fault and two with 0.5 m/s ramp fault). Noting the sensitivity of the test statistics and HPLs to the changing noise floor during the restoration of C/N$_0$ levels, ten simulations are created for each of cases II and IV - two without fault (one from each data set collected from the Spirent simulator) and four with each fault magnitude (two from each data set). Thus, the elevated noise floor accompanied with

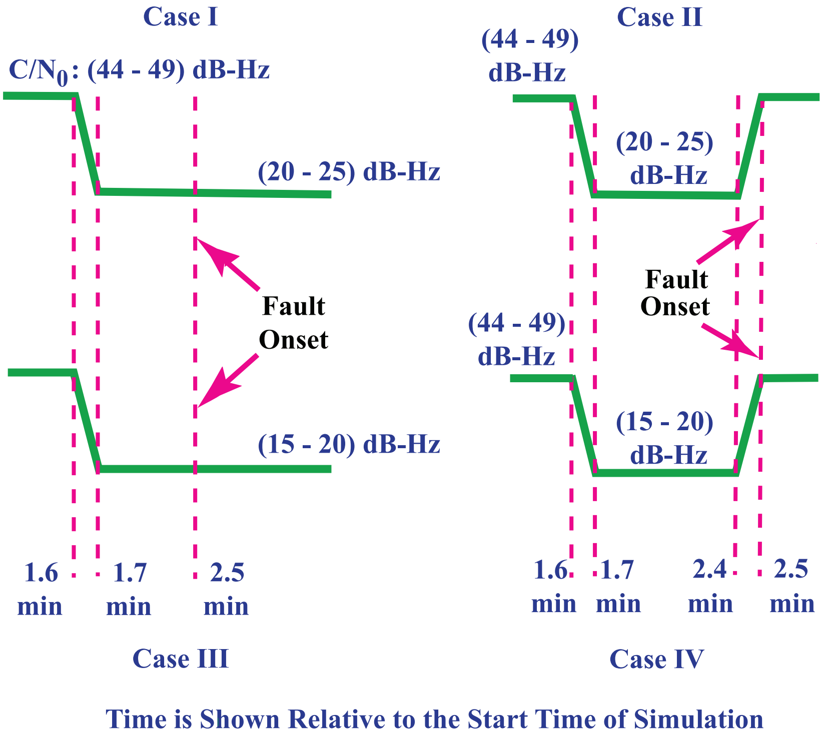


Figure 6.3: Low C/N_0 Data

faults subjects the scalar and vector receiver algorithms to an extremely adverse situation to test the performance of their integrity monitoring algorithms under degraded signal environments.

6.3 Validation Results of RAIM Algorithm

In this section results with the collected Spirent data and the above-mentioned low C/N_0 data are presented. RAIM performance of the scalar and vector architectures are compared with respect to fault detection times and HPL. It is assumed that the aircraft executes an LNAV approach. According to International Civil Aviation Organization (ICAO) specifications, RAIM parameters for an LNAV approach are as follows: Probability of false alarm $P_{\mathrm{FA}} = 1 \times 10^{-5}$/hr, $P_{\mathrm{HMI}} = 1 \times 10^{-7}$/hr, HAL $= 556$ m. In addition, the prior probability of satellite failure P_F is assumed 10^{-5}/satellite/hr [82]. From the above specifications P_{FA}, P_{HMI} and P_F are calculated for a duration of 5 minutes and for the number of satellites visible during the approach after applying a masking angle of 5^o above the local horizon. For calculating the HPL, method 1 on page 150 (section 5.3.1.3) of Chapter 5 is used.

6.3.1 Scalar and Vector Loop Design Parameters

The scalar loop and vector loop design parameters are shown in Tables 6.1 and 6.2. A first order DLL loop filter and a second order FLL loop filter are considered for the scalar loops. Note that almost all scalar loops simuated here are found to lose lock with the (15 - 20) dB-Hz data in which the C/N$_0$s of only two satellites are above 17 dB-Hz all the times. So, scalar loop results are not shown for these data sets. In order to compute the colored noise variance of the scalar loop pseudorange measurements using Equation 5.6, the averaged discriminator samples are collected every 0.1 sec over a period of 15 sec. The scalar loop white noise variance is multiplied by a factor of 3.5 for the (44 - 49) dB-Hz data set. This is done to account for small residual atmospheric errors which are comparable to noise at high C/N$_0$. For the vector loops the multiplication factor is 4. It should be noted that the position update interval of the scalar loops for the (25 - 30) dB-Hz data is 6 sec whereas that for the vector loops is 4 sec. This is because, for the same duration of data, the scalar loop test statistic is

Table 6.1: Scalar Loop Design Parameters

Design parameters	STL	
	(44 - 49) dB-Hz	(25 - 30) dB-Hz
Discriminator output interval (T_a)	20 ms	20 ms
Discriminator averaging interval (lT_a)	0.1 sec	0.1 sec
Position update rate	2 sec DLL BW ($2B_n$): 4 Hz Pseudorange sample rate (T_ρ): 0.25 sec	6 sec DLL BW: 2 Hz Pseudorange sample rate: 1 sec
Velocity update rate	2 sec	6 sec
Test statistic derived from	Pseudorange residuals	Pseudorange residuals
Statistic update interval or averaging interval of pseudorange residuals (pT_ρ)	2 sec	6 sec
White noise variance of channel i statistic at epoch k $\left(\dfrac{2B_n lT_a\sigma^2_{\bar{\zeta}_{i,k}}}{p}\right)$	$\dfrac{4\times 0.1}{8}\sigma^2_{\bar{\zeta}_{i,k}}$	$\dfrac{2\times 0.1}{6}\sigma^2_{\bar{\zeta}_{i,k}}$

Table 6.2: Vector Loop Design Parameters

Design parameters	VTL	
	(44 - 49) dB-Hz	(25 - 30) dB-Hz
Discriminator output interval (T_a)	20 ms	20 ms
Discriminator averaging interval	Code (MT_a): 2 sec Doppler (lT_a): 0.1 sec	Code: 4 sec Doppler: 0.1 sec
Position update rate	2 sec	4 sec
Velocity update rate	0.1 sec	0.1 sec
Test statistic derived from	Discriminator outputs	Discriminator outputs
Statistic update interval (MT_a)	2 sec	4 sec
Channel i statistic noise variance at epoch k ($\frac{l}{M}\sigma^2_{\bar{\zeta}_{i,k}}$)	$\frac{1}{20}\sigma^2_{\bar{\zeta}_{i,k}}$	$\frac{1}{40}\sigma^2_{\bar{\zeta}_{i,k}}$

Design parameters	VTL
	(15 - 20) dB-Hz
Discriminator output interval (T_a)	20 ms
Discriminator averaging interval	Code (MT_a): 6 sec Doppler (lT_a): 0.1 sec
Position update rate	6 sec
Velocity update rate	0.1 sec
Test statistic derived from	Discriminator outputs
Statistic update interval (MT_a)	6 sec
Channel i statistic noise variance at epoch k ($\frac{l}{M}\sigma^2_{\bar{\zeta}_{i,k}}$)	$\frac{1}{60}\sigma^2_{\bar{\zeta}_{i,k}}$

found to detect fault with the six-second update interval but not with the four-second interval. However, the vector loops are able to detect fault over the same duration with the four-second update interval. The modification to account for different position and velocity update rates of the vector architecture, as described in Section 5.3.2.2, is applied to vector loop protection level calculations of this chapter.

As an aside, it is important to note that an aviation receiver certified to execute non-precision approaches updates the position and protection levels at 1 Hz. However, such fast update rates are not possible in this study under degraded signal environments. Fast updates are also not possible in good signal environments. This is because the scalar architecture has high colored noise due to the large drift of the receiver clock, which needs averaging of a number of RAIM input samples. These samples should be spaced apart by more than the loop filter time constant. As for the vector architecture, the discriminator outputs used in vector loop RAIM have large noise which also needs RAIM input samples to be averaged over at least 2 sec.

6.3.2 Results with Spirent Data

As noted earlier, a distinct problem with the vector architecture is that all of its pseudorange measurements get corrupted by a fault in a satellite channel, rendering the vector loop pseudorange measurements unusable for integrity monitoring purposes. As a remedy, use of the code phase discriminator outputs is proposed to perform integrity monitoring with the vector architecture. Before delving deep into the performance of the vector loop integrity monitoring algorithm developed in Chapter 5, some analysis results are presented below. This is to show what the vector loop pseudorange noise and pseudorange residuals look like when a fault occurs in a channel and to study whether the code phase discriminator outputs comply with the single fault assumption. All these results aim at providing insights into the workings of the vector architecture under fault. It is important to note that the pseudorange noise mentioned above indicates the difference between the vector or scalar loop-generated pseudorange measurements and the geometric pseudoranges (i.e. geometric ranges plus atmospheric delays and clock bias). On the other hand, pseudorange residuals denote the difference between pseudorange measurements and predicted or estimated pseudoranges.

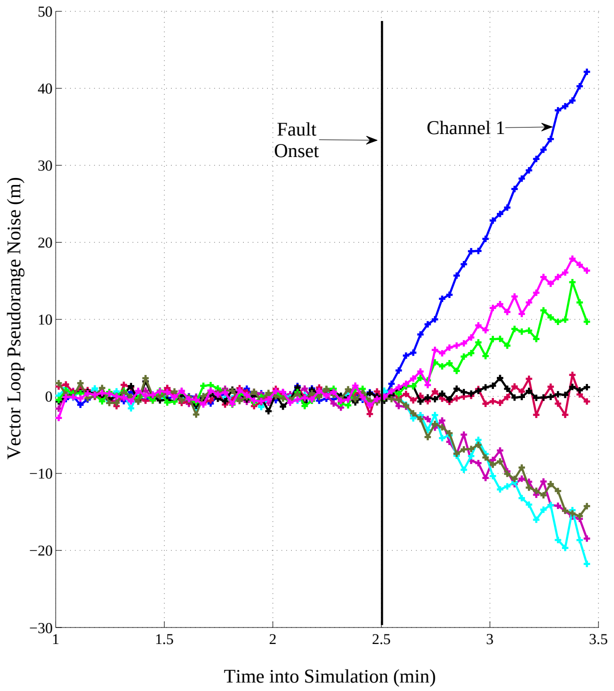


Figure 6.4: Noise in Vector Loop Pseudorange Measurements

6.3.2.1 Vector and Scalar Loop Pseudorange Noise and Residuals

Figure 6.4 shows the receiver input noise plus the noise in the vector loop-generated pseudorange measurements when a fault is injected in channel 1. They are formed by multiplying the code phase discriminator outputs with the speed of light and then removing the fault from the discriminator outputs of the faulty channel. Thus, they reflect the difference between the geometric pseudoranges plus receiver input noise and the pseudorange measurements generated by the vector tracking receiver. It is evident from the figure that the effect of the fault in channel 1 propagates to almost all channels and corrupts their measurements. This reinforces the earlier finding that the vector loop pseudorange measurements do not comply with the single fault assumption needed for protection level calculations of the RAIM algorithms. Thus, they are not suitable for RAIM purposes. Nevertheless, it would be interesting to study how their residuals behave under fault, which is explored next.

Two types of vector loop pseudorange residuals are depicted in Figure 6.5 when a ramp fault of slope 1 m/s is injected into channel 1 at 2.5 min into simulation. The vector loop pseudorange measurements are generated by taking the difference between the signal receive time and the receiver-estimated signal transmit time and multiplying the difference by the speed of light. The top left subplot of the figure shows the difference between the vector loop pseudorange measurements and the pseudoranges predicted by the navigation filter (i.e., pseudoranges generated by projecting the predicted state vector onto LOS). The top right subplot shows the difference between the vector loop pseudorange measurements and the pseudoranges estimated by the navigation filter (i.e., pseudoranges generated with the estimated state vector). The distinction between the two subplots is that the residuals in the first one are obtained after time updates while those in the second one are obtained after measurement updates. The residual magnitudes are less in the first subplot because the replica code phase based on which the vector loop pseudorange measurements are generated closely follows the navigation filter prediction. It is corrected after each measurement update to accommodate the difference shown in the second subplot. It is interesting to note that none of the residuals exhibit any anomaly (e.g., drifts) due to the fault except that the estimated pseudorange residuals vary over a wider range after the occurrence

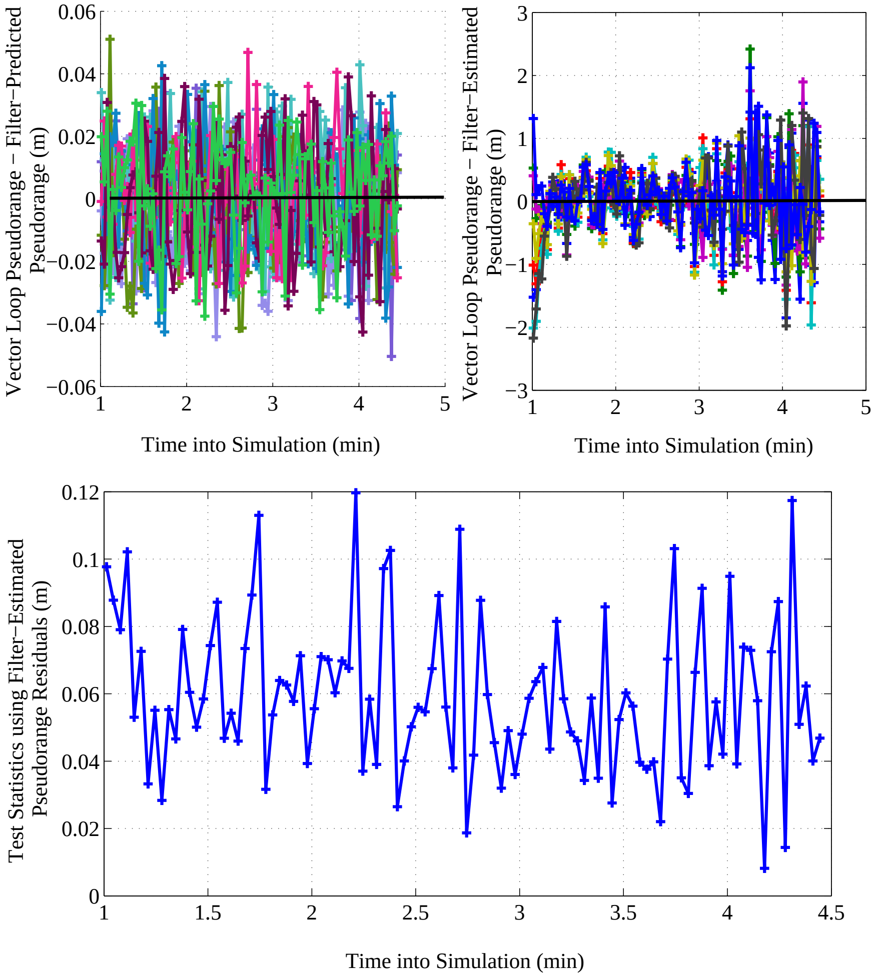


Figure 6.5: Vector Loop Pseudorange Residuals and Test Statistics Derived from the Pseudorange Residuals

of the fault. The absence of inconsistency in the pseudorange residuals of the vector loops results from the fact that individual code phases are adjusted by projecting the navigation filter-estimated state vector onto the LOS vectors. This means that the replica code phases from which pseudorange measurements are derived are aligned in accordance with the estimated pseudoranges. Similar to the predicted/estimated pseudoranges, vector loop pseudorange measurements are, therefore, derived as projections from the position domain to the range domain. As a result, when fault occurs, both pseudorange measurements and predicted/estimated pseudoranges deviate from the true geometric pseudoranges in such a way that the pseudorange residuals vector will continue to move around the zero line and follow the linearized measurement model. However, their noises are correlated with each other at all times because, as mentioned above, the replica code phases are corrected by projecting the estimated user position onto LOS vectors. Ignoring the correlation of noise, an attempt has been made to see how the fault detection algorithm works with the pseudorange residuals. As shown in Figure 6.5, the test statistics formed by the estimated pseudorange residuals do not indicate any fault. So, no anomaly is detected by the fault detection algorithm which performs a consistency check of the pseudorange residuals based on the measurement model.

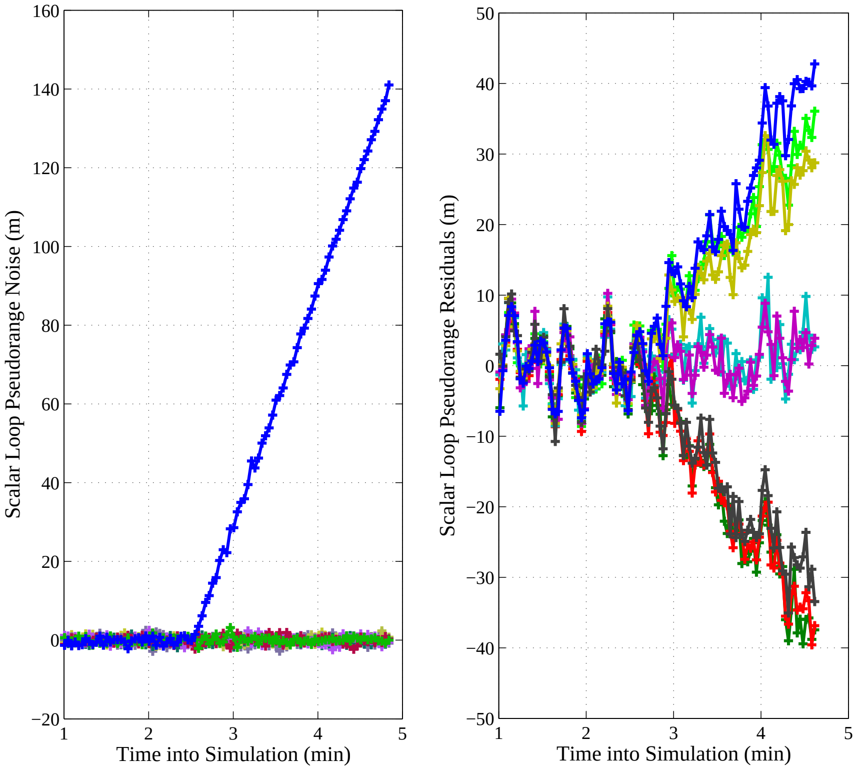


Figure 6.6: Scalar Loop Pseudorange Noise and Residuals

In order to contrast the vector loop pseudorange noise and residuals with its scalar counterparts, the scalar loop pseudorange noise and residuals are shown in Figure 6.6. The true geometric range, satellite clock error, estimated receiver clock bias and the atmospheric delays are first subtracted from the scalar loop pseudorange measurements to generate scalar loop pseudorange noise. The resulting difference, however, exhibits a common non-zero mean which is probably caused by the error in the estimated clock bias and is, thus, removed. The pseudorange residuals are obtained by taking the difference between the pseudorange measurements and the navigation filter-predicted pseudoranges. As expected, the scalar loop pseudorange noises exhibit a linear drift only in the faulty channel and all pseudorange residuals start drifting apart after the fault is injected into channel 1, unlike the vector architecture.

Thus, this analysis shows that the vector loop pseudorange measurements not only violate the single fault assumption, but in the presence of a fault their residuals do not show any inconsistency which can be exploited by the fault detection algorithm to detect faults.

6.3.2.2 Error in Code Phase Discriminator Outputs

As discussed in Chapter 5, the vector loop pseudorange errors or code phase discriminator outputs shown in Figure 6.4 have the same mathematical model as that of the scalar loop pseudorange residuals shown in Figure 6.6. In addition, discriminator noises are uncorrelated with one another and it can be predicted from theory that fault at the discriminator level will affect the anomalous channel alone. This means that the discriminator noise/error of the faulty channel will have a non-zero mean due to fault while those of the other channels will have zero mean, unlike the vector loop pseudorange measurements. Thus, the code phase discriminator outputs are expected to exhibit inconsistency during faults which can be used to do integrity monitoring with the vector architecture. The ability of the code phase discriminator outputs to detect fault is demonstrated in the simulation studies of Chapter 5. In this chapter, before focusing on vector loop RAIM performance with GPS data, an analysis of the error characteristics of the code phase discriminator outputs is performed. This analysis serves to verify if the single fault assumption holds for the code phase discriminator outputs.

In order to verify if a fault remains confined only to the faulty channel at the discriminator level, error in the code phase discriminator outputs is calculated and plotted in Figure 6.7. From Equation 5.25 of Chapter 5, the discriminator error

($\underline{\epsilon}_{\text{vector},\,m}$) results as

$$\begin{aligned} \underline{\tau}_m &= H_m[X_{p_m} - \tilde{X}^-_{p_m}] + \underline{\epsilon}_{\text{vector},\,m} \\ \text{or,}\quad \underline{\epsilon}_{\text{vector},\,m} &= \underline{\tau}_m - H_m[X_{p_m} - \tilde{X}^-_{p_m}] \end{aligned} \tag{6.1}$$

where $\underline{\tau}_m$ is the discriminator output vector; X_{p_m} is the true user position and clock bias vector and $\tilde{X}^-_{p_m}$ is its prediction at time t_m. while the true user position is known, the true clock bias is an unknown quantity. However, it can be determined after subtracting the position error from the discriminator outputs; it will be a common error that is observed in all channels. The clock error is, thus, extracted by determining the mean error across channels. Once the clock error is removed, the result obtained is shown in the left subplots of Figure 6.7. To compare with the scalar loops, the same operation

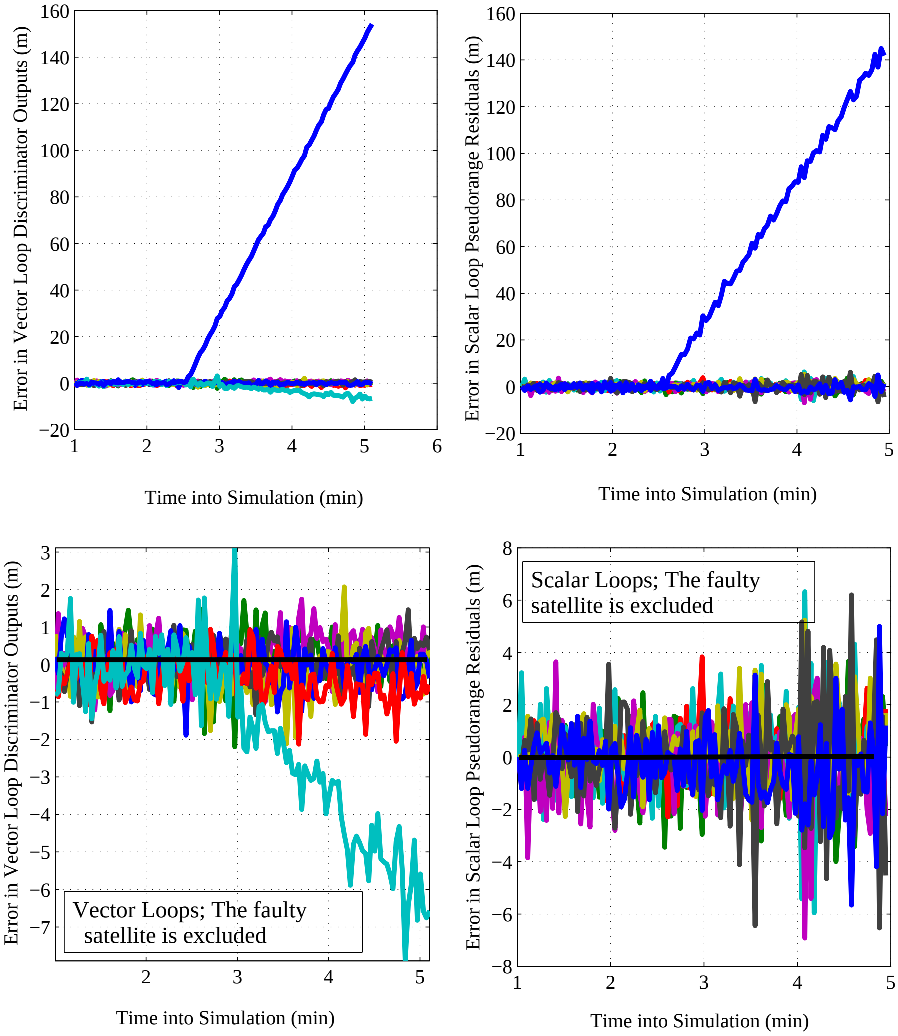


Figure 6.7: Error in Vector Loop Code Phase Discriminator Outputs and in Scalar Loop Pseudorange Residuals

is performed with the scalar loop pseudorange residuals and the scalar loop errors are shown alongside the vector loop plots in Figure 6.7. It is evident that the vector loop discriminator error in the faulty channel grows linearly with time at the rate of 1 m/s (i.e., at the same rate as that of the injected fault).

An issue that is important to vector loop RAIM is highlighted by the slowly growing discriminator output error clearly evident in the lower left plot of Figure 6.7. As will be shown, this has to do with how the correlation function is performed. This will be discussed in passing only here as a direction for future research and does not affect the performance of vector loop RAIM in this analysis. To this end, note that, while all other healthy channels have zero-mean errors up to 3 minutes, error in channel 8 starts drifting slowly after that. In contrast, the scalar loop errors of the healthy channels always have zero mean. The drift of channel 8 in vector loops, however, is very slow compared to the actual fault and it is found from other data sets that it depends on the magnitude of fault drift. That is, with a fault of lower slope, the drift of channel 8 is slower. Surprisingly, the drift is found to be present only in channel 8 in all of the collected data.

Since the slow drift in channel 8 is not present in the scalar architecture, it was anticipated that the reason could lie in the different working principles of the two tracking loops. The difference in the workings of the scalar and vector loops under fault is that the replica codes of all channels of the latter start drifting from their respective incoming codes when a fault occurs. This happens because the replica codes in vector tracking mode are aligned using the estimated pseudoranges from the navigation filter. Thus, the operating points of the early, late and prompt correlators move away from their locked positions. If the code autocorrelation function is distorted from its ideal triangular shape of unity slope, then the discriminator outputs $\underline{\tau}_m$ will not match the first term of Equation 6.1 and a systematic error will appear in the term $\underline{\epsilon}_{\text{vector},\, m}$. To verify this, the autocorrelation functions of all channels are reconstructed using three more early-late correlator pairs in addition to the tracking correlator pair whose outputs are used to zero the discriminator output [93]. The reconstructed autocorrelation functions of channel 8 and another channel are shown in Figure 6.8. The line with '+' markers in the figure depicts the trace of the correlator pairs as fault occurs whereas the dark solid line indicates an ideal autocorrelation function with unity slope on either side. The dashed line in the right subplot also shows the trace of the tracking correlators of channel 8 for a longer duration. It is apparent that the correlation function of channel 8 has a different slope than unity. After accounting for the different slope of the autocorrelation function in Equation 6.1, the errors in the code phase discriminator outputs are shown in Figure 6.9. The absence of the drift of channel 8 in the figure justifies that it was caused by the slightly distorted autocorrelation function. Thus, a

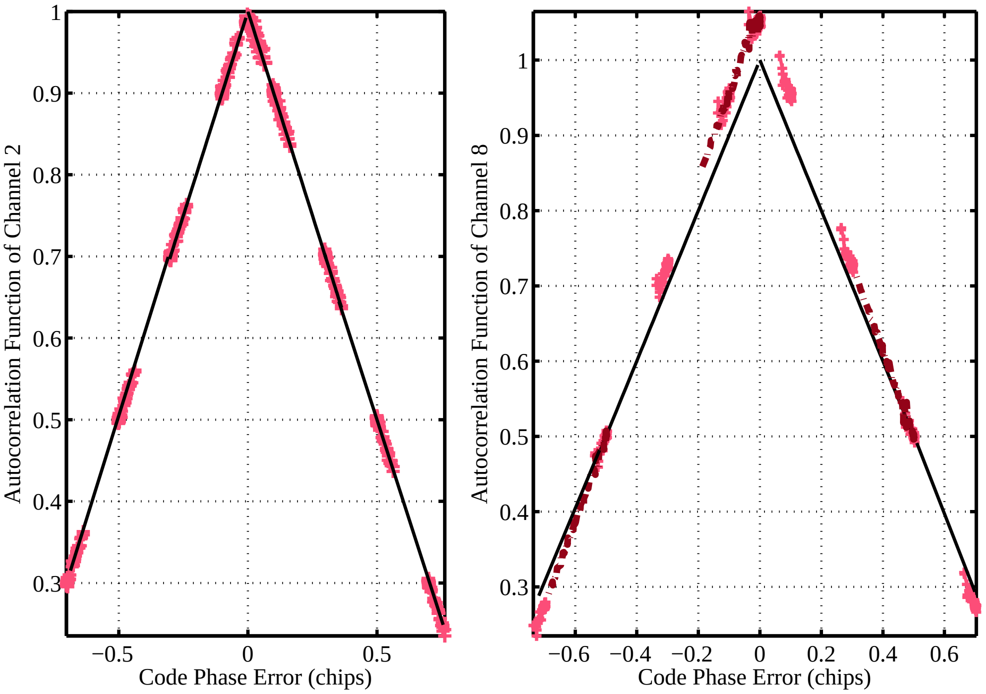


Figure 6.8: Reconstructed Autocorrelation Functions of Channels 2 and 8

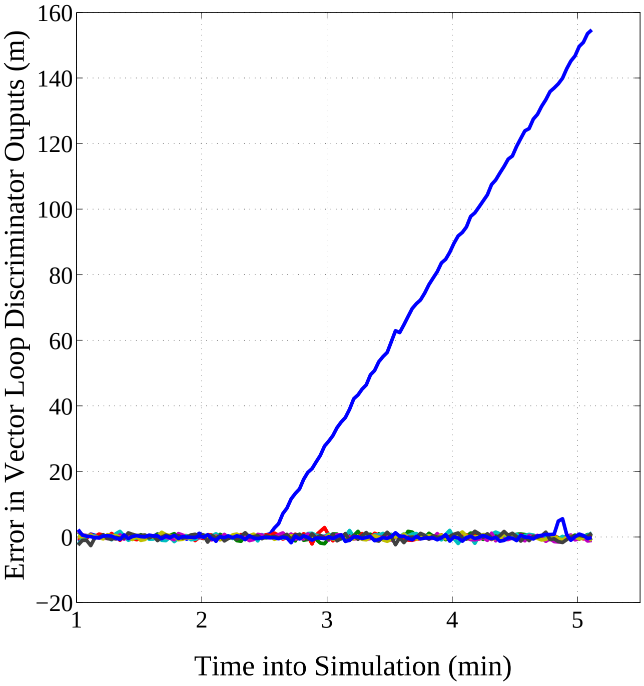


Figure 6.9: Error in Vector Loop Code Phase Discriminator Outputs after Correcting the Slope of the Autocorrelation Function

fault triggers drifts of the replica code phases which may lead to an increasing error in the discriminator output of a healthy channel if its autocorrelation function is distorted from the ideal shape. It is also important to note that in scalar loops and in vector loops under no fault conditions, the changed slope will not cause any tracking error because the change is identical on either side. In addition, the drift, being small, does not affect the RAIM performance of vector loops in this study. However, the above analysis indicates that vector loop RAIM performance can be sensitive to the shape of the autocorrelation function and, therefore, the validity of the algorithm and its performance under multipath signals deserve extensive studies. Next, the performance of the RAIM algorithm designed in Chapter 5 is discussed.

6.3.2.3 RAIM Performance with Spirent Data

The scalar and vector loop RAIM performance is compared in terms of fault detection

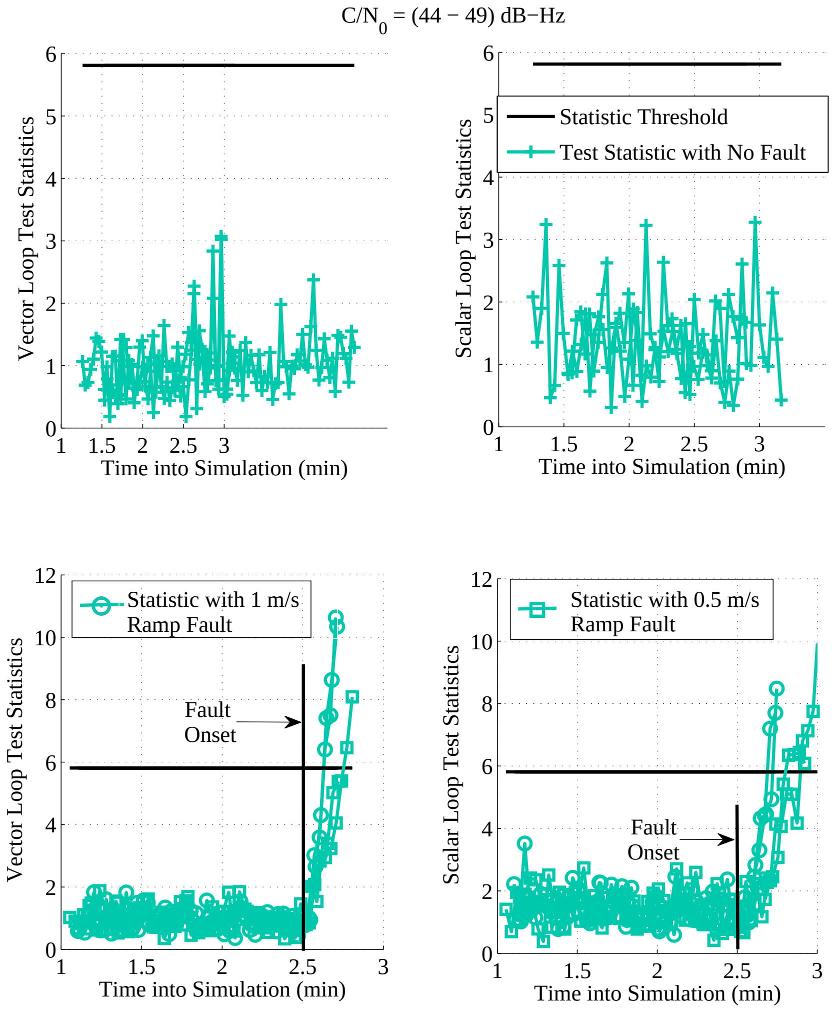


Figure 6.10: Evolution of Scalar and Vector Loop Test Statistics for Spirent Data

times and HPLs. The vector loop discriminator averaging interval is 2 sec at (44 - 49) dB-Hz. That is, the number of discriminator samples (M) averaged is $2/T_a = 2/(20\times10^{-3}) = 100$. M_{u_1} and M_{u_2} are two upper bounds for the number of discriminator samples that can be averaged. They are defined below Equation 5.29 in Chapter 5. M_{u_1} is calculated as 4397 and remains the same for all C/N$_0$s as it mainly depends on LOS dynamics. M_{u_2} for the (44 - 49) dB-Hz data sets is 11500. So, M is below M_{u_1} and M_{u_2}, as required. The test statistics for the six data sets (two without fault, two with a ramp fault of slope 1 m/s and two with a ramp fault of slope 0.5 m/s) are shown in Figure 6.10. Figure 6.11, on the other hand, illustrates the HPL and HPE for the same data sets. HPEs are plotted until a fault is detected. The left hand subplots show results for the vector loops and the right hand subplots show results for the scalar loops. The number of previous time epoch terms needed for HPL calculations is 12. Criterion for determining this number is provided in Section 5.3.1.3 while discussing

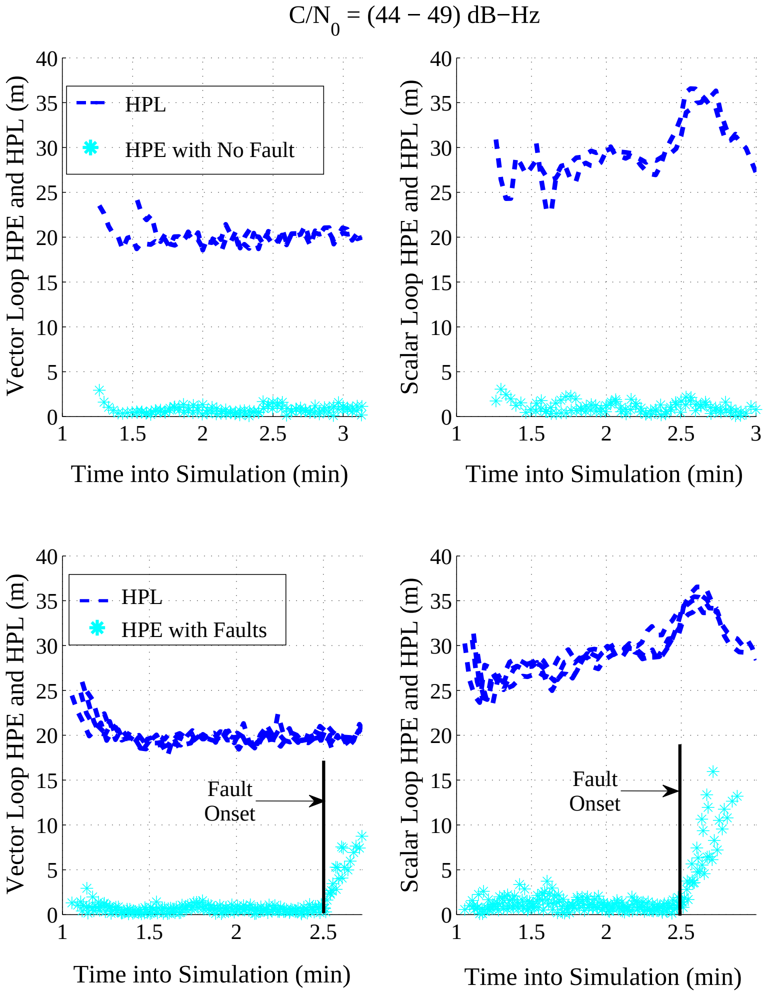


Figure 6.11: Scalar and Vector Loop Horizontal Protection Levels and Position Errors for Spirent Data

bounds on EKF position error.

For these data sets the vector architecture detects fault about 4 sec before the scalar architecture. As for the protection levels, the vector loop HPL is about 5 m to 10 m lower than that of the scalar loops. The reason for better performance of the vector loops is that the vector loop test statistic and position estimate are not affected by colored noises but the scalar loop test statistic and position estimate are. It is found that the colored noise contribution to the scalar loop pseudorange measurements is significant. This is because the receiver is driven by a TCXO whose clock drift and bias are found to be quite large. The results for four cases of low C/N$_0$ data are presented next.

6.3.3 Results with Low C/N$_0$ Data

The RAIM performance of the scalar and vector loops for the generated low C/N$_0$ data is provided below.

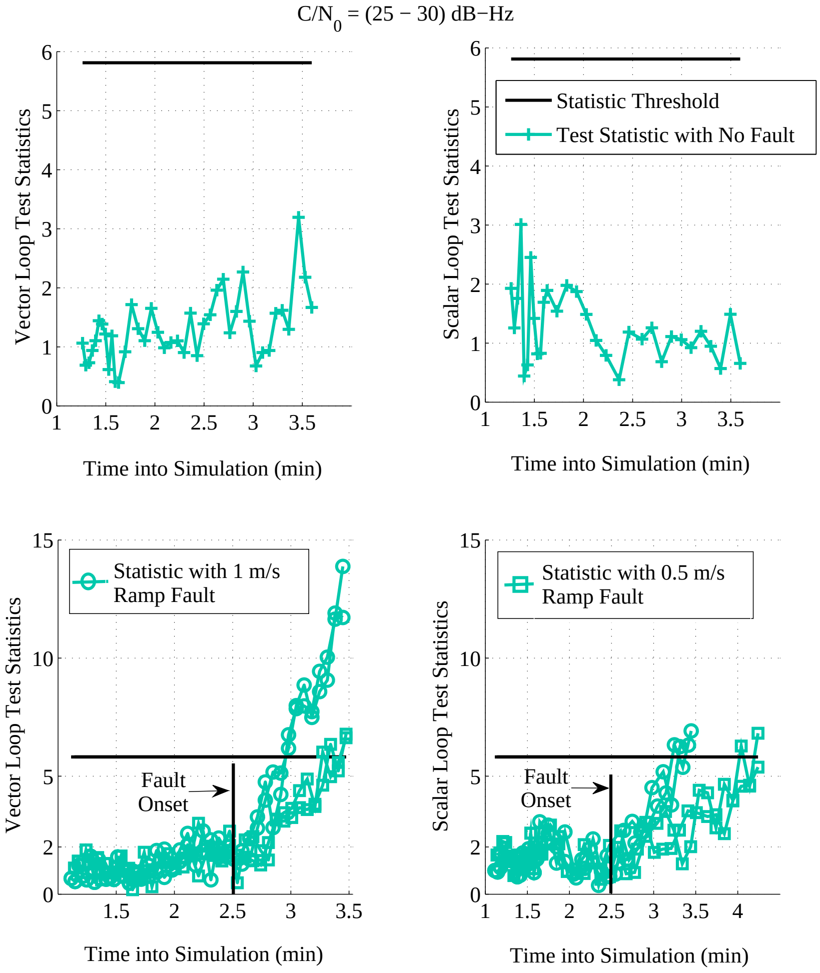


Figure 6.12: Scalar and Vector Loop Test Statistics for Case I

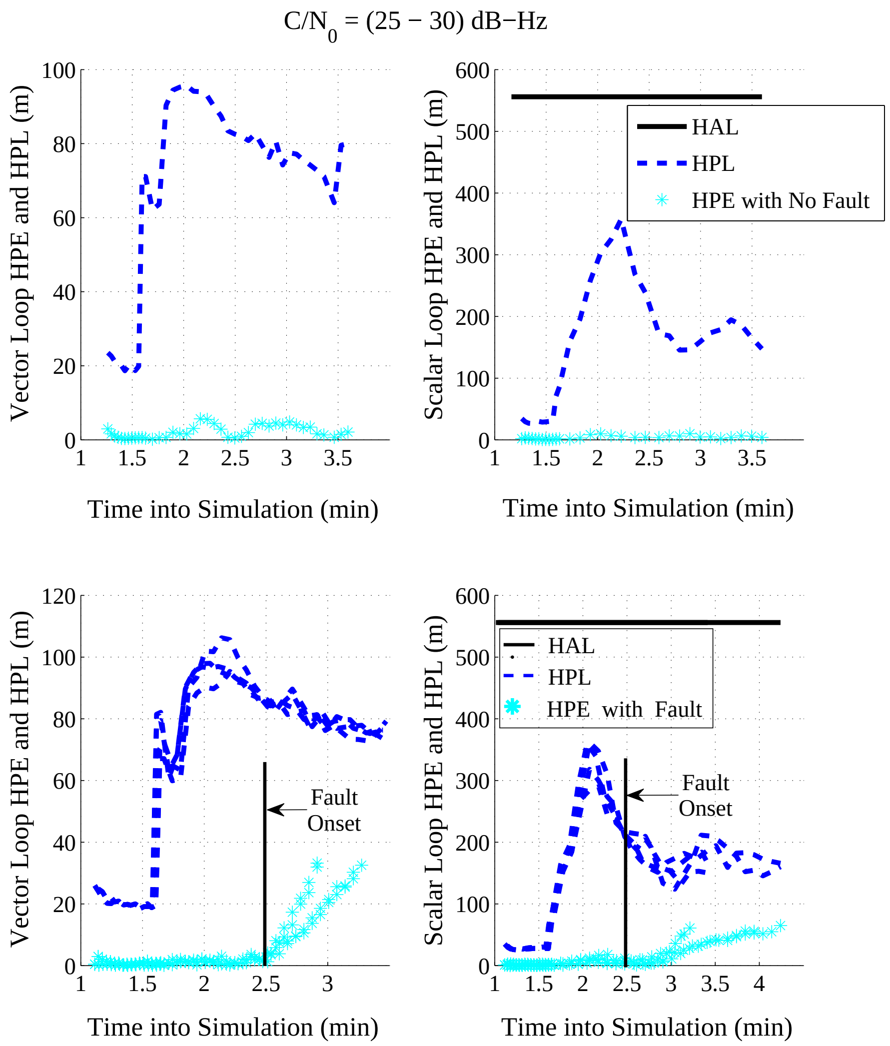


Figure 6.13: Scalar and Vector Loop Horizontal Protection Levels and Position Errors for Case I

6.3.3.1 Case I

Five data sets are generated for this case where the C/N$_0$ level is decreased to (25 - 30) dB-Hz at 1.6 min into the simulation and held constant at that level until the end. The number of discriminator samples (M) averaged to generate the test statistic at the low C/N$_0$ level is 200 while M_{u_1} and M_{u_2} are 4397 and 2000, respectively. For calculating the HPL, 24 previous epoch terms are considered. It is important to note that the number of previous time epochs needed to calculate the HPL increases as C/N$_0$ decreases. The vector loop fault detection times are about 16 sec and 42 sec lower than the scalar loops. The vector loop HPLs are about half of the scalar loop

HPLs. Comparing with the previous high C/N$_0$ results, it can be concluded that the improvement offered by the vector loop RAIM has increased as C/N$_0$ levels are decreased.

6.3.3.2 Case II

In this case C/N$_0$ is lowered to the level of (25 - 30) dB-Hz for a period of one minute and then restored back to the original level. The fault initiates immediately after the signal power is restored. A total of ten simulations are created for this case: Two without fault, four with 1 m/s ramp fault and four with 0.5 m/s ramp fault. The test statistics are shown in Figure 6.14. HPLs and HPEs are shown in Figure 6.15. Note that the scalar loop HPL and HPE plot is broken into two segments for clarity. In this case the vector loops detect 1 m/s ramp fault 8 sec and 0.5 m/s fault 4 sec earlier than the scalar loops.

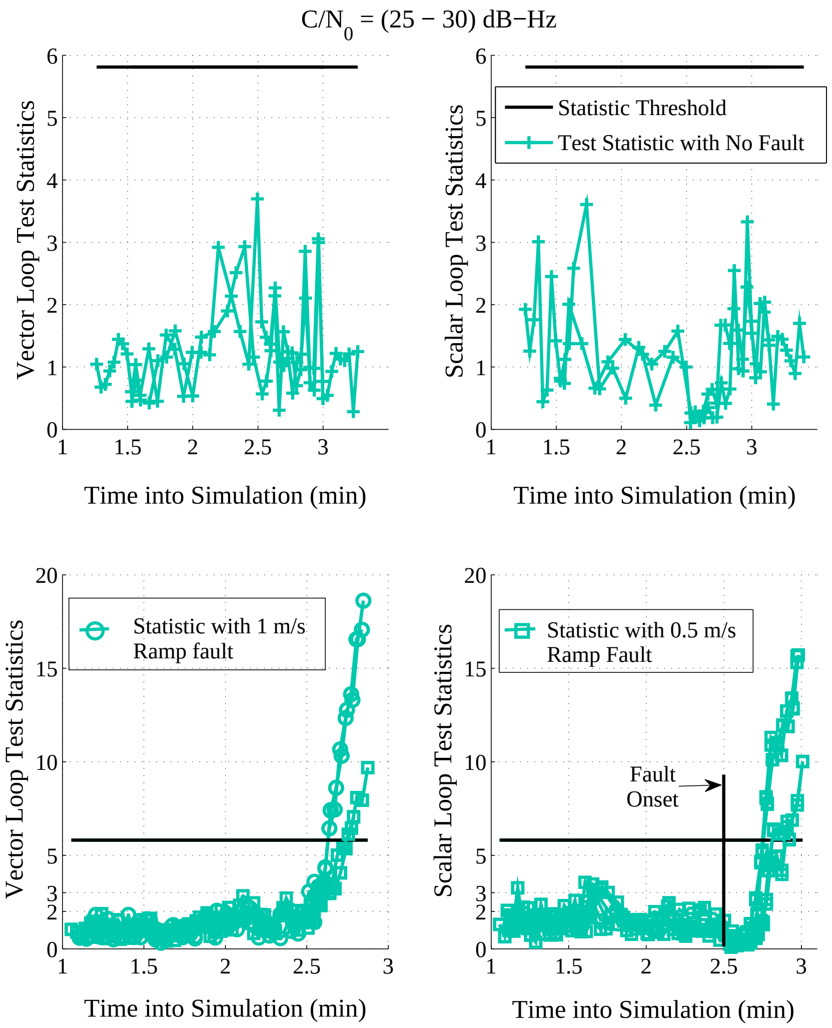


Figure 6.14: Scalar and Vector Loop Test Statistics for Case II

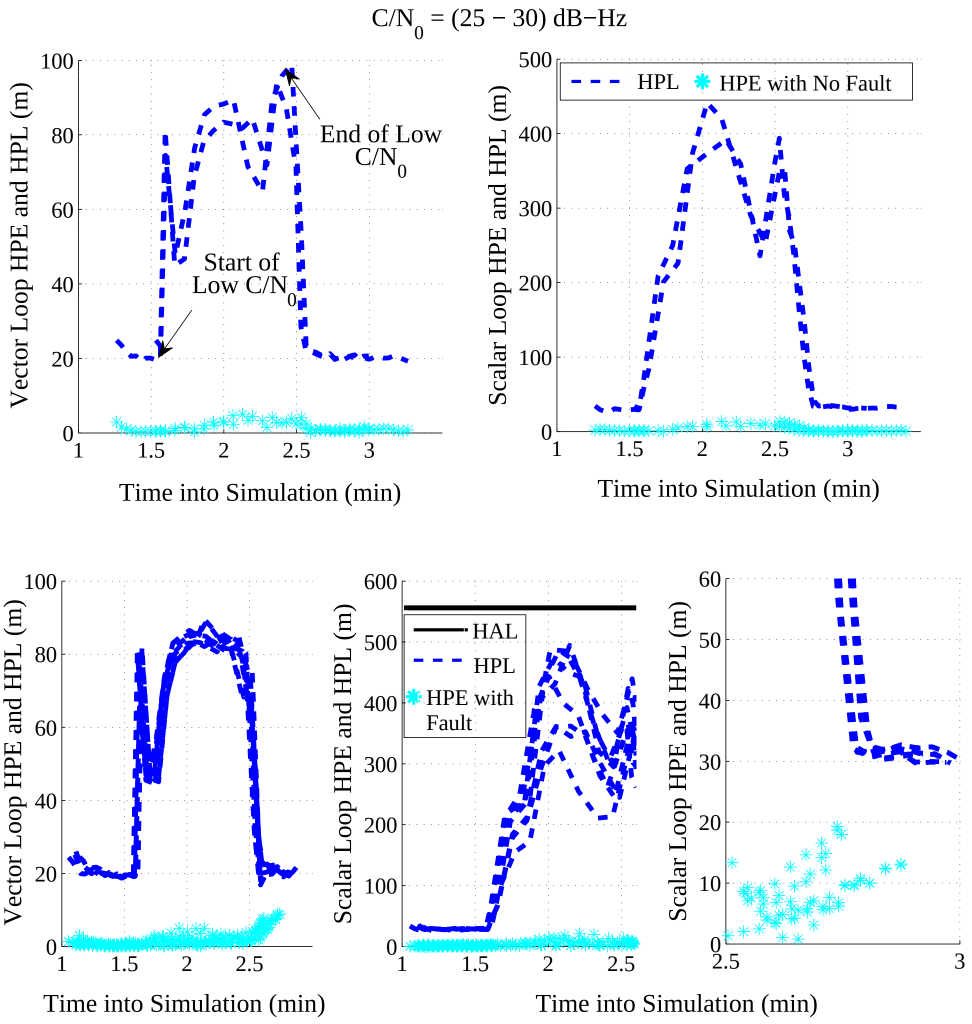


Figure 6.15: Scalar and Vector Loop HPL and HPE for Case II; Note that Scalar Loop Results under *Fault* are Divided into Two Subplots on the Bottom Right Corner for Visual Clarity; The First Subplot Shows Results from 1 min to 2.5 min into Simulation and the Second One Shows Results 2.5 min onwards

6.3.3.3 Case III

For case III C/N$_0$ is degraded to the level of (15 - 20) dB-Hz at about 1.6 min into simulation and maintained at that level till the end. Since most of the scalar loops are found to lose lock at the degraded signal level, only vector loop results are provided. The number of discriminator samples averaged to form the test statistic at (15 - 20) dB-Hz is 300. The corresponding value of M_{u_2} is 500. The number of previous time epoch terms used to calculate the HPL is 29. The performance of vector loop RAIM

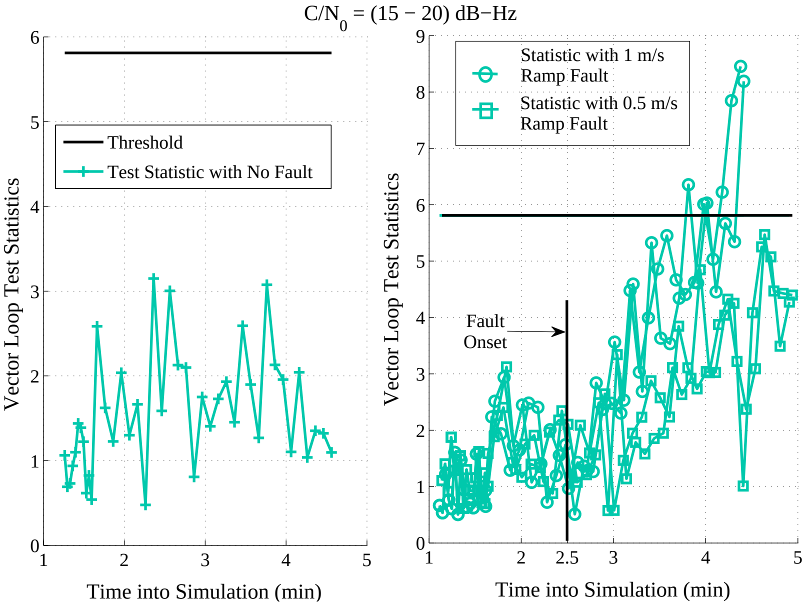


Figure 6.16: Vector Loop Test Statistics for Case III

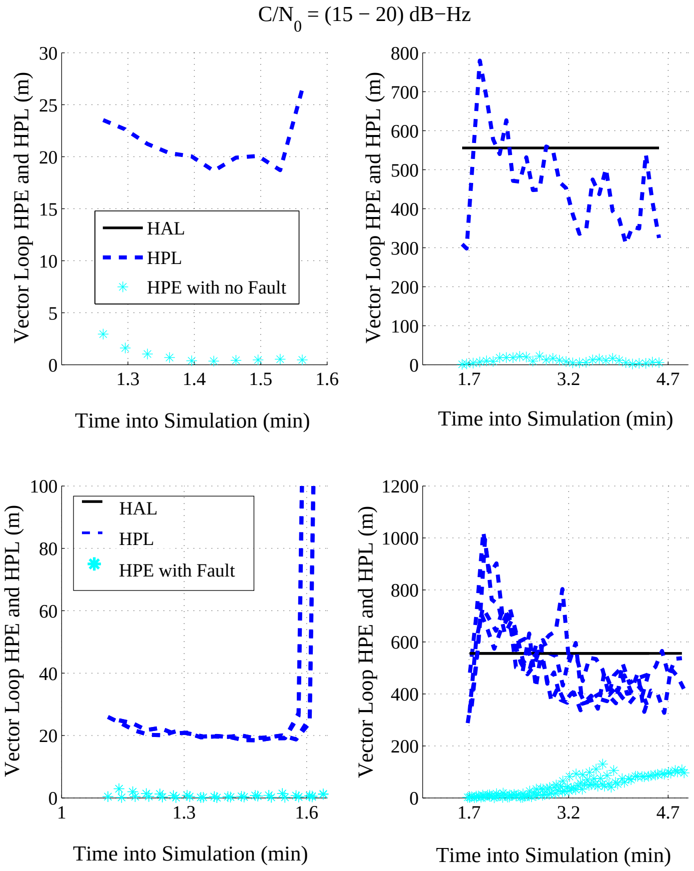


Figure 6.17: Vector Loop HPL and HPE for Case III

is shown in Figures 6.16 and 6.17. As the HPL changes over a wide range due to change in C/N_0, the HPL plot is split into two subplots for visual clarity. The results show that although fault detection is possible at such low C/N_0 environments, the HPL exceeds the HAL many times, causing RAIM to be unavailable then.

6.3.3.4 Case IV

This case is similar to case II. The only difference is that C/N_0 is degraded to the level of (15 - 20) dB-Hz instead of (25 - 30) dB-Hz. Fault occurs after the signal level is restored. Results of all of the 10 simulations are presented in Figures 6.18 and 6.19. Although the HPL plot has been broken into three subplots to clearly show both small and large values of the HPL, the separation between the HPL and HPE does not appear to be wide enough in the bottom right subplot of Figure 6.19. Actually, the HPEs that are comparatively large have a correspondingly high HPL. This is not apparent because all simulation results have been plotted together. In order to clearly illustrate the separation between the HPE and HPL, this segment of the figure is magnified in Figure 6.20. In this figure values of some of the corresponding HPE and HPL that have the same X coordinates are displayed using datatips. For example, when the HPE is 10.76 m (second left datatip in the middle), the HPL is 62.07 m.

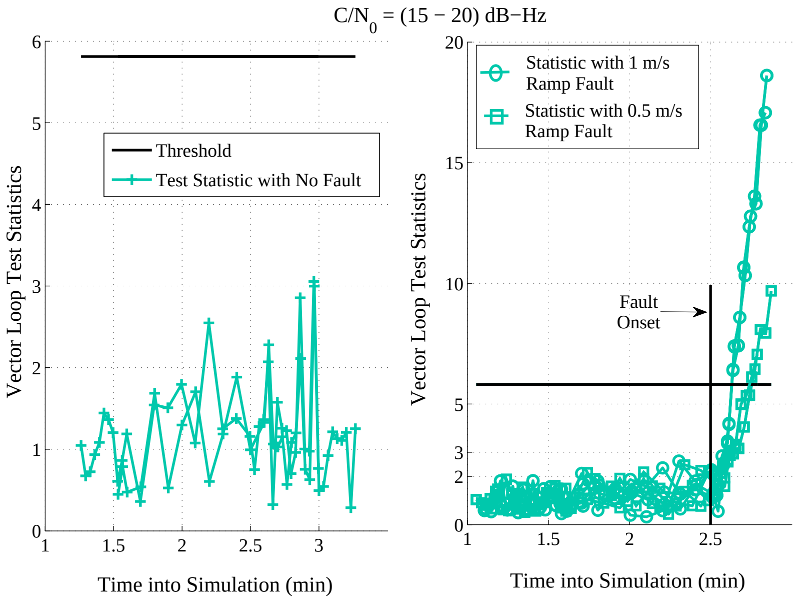


Figure 6.18: Vector Loop Test Statistics for Case IV

As for vector loop RAIM performance, during the period when C/N$_0$ is in the range of 15 dB-Hz to 20 dB-Hz, the HPL exceeds the HAL, resulting in RAIM unavailability. However, RAIM becomes available immediately after the signal power is restored from the degraded condition.

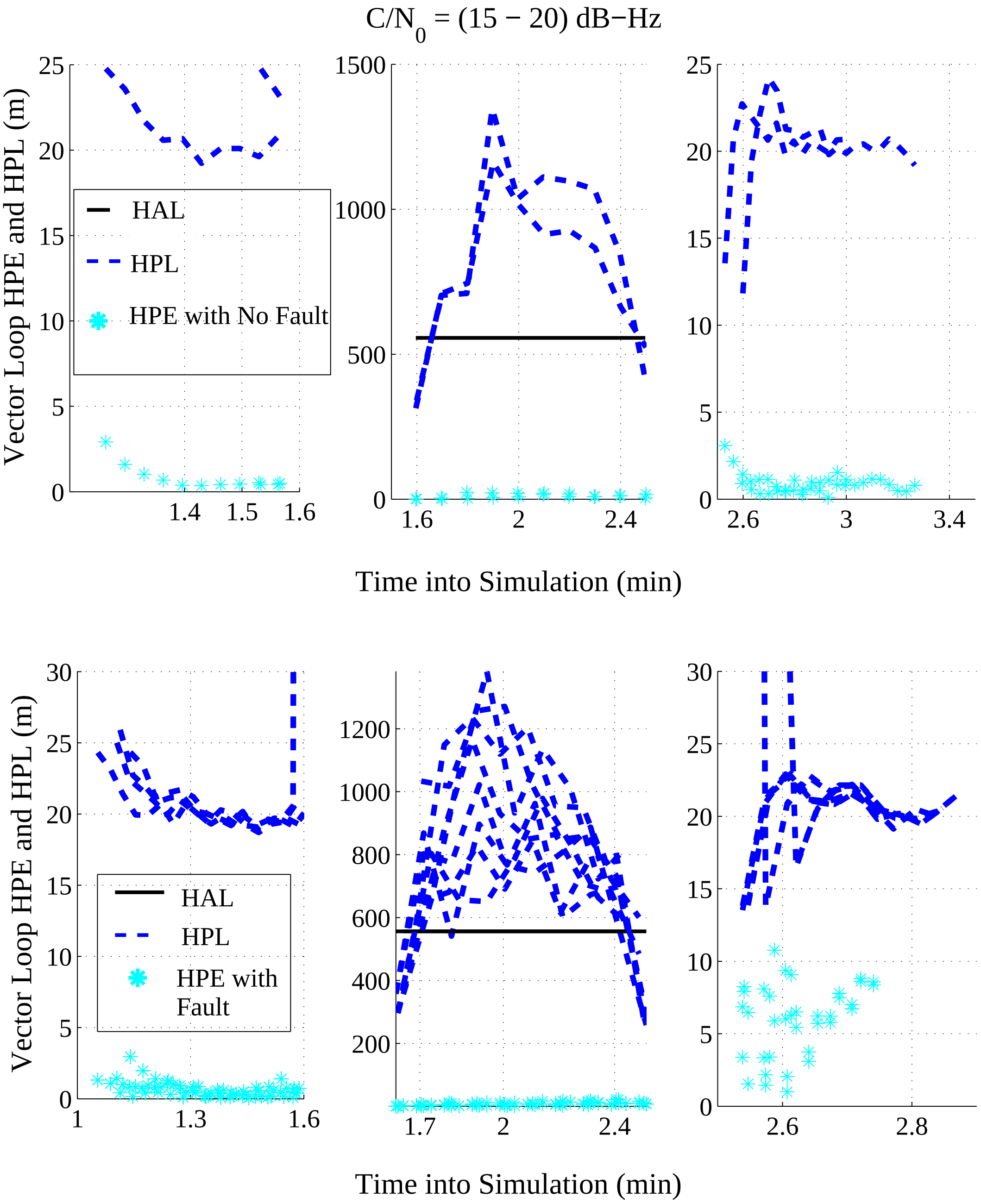


Figure 6.19: Vector Loop HPL and HPE for Case IV; Note that the Entire Time Interval is Divided into Three Sub-intervals

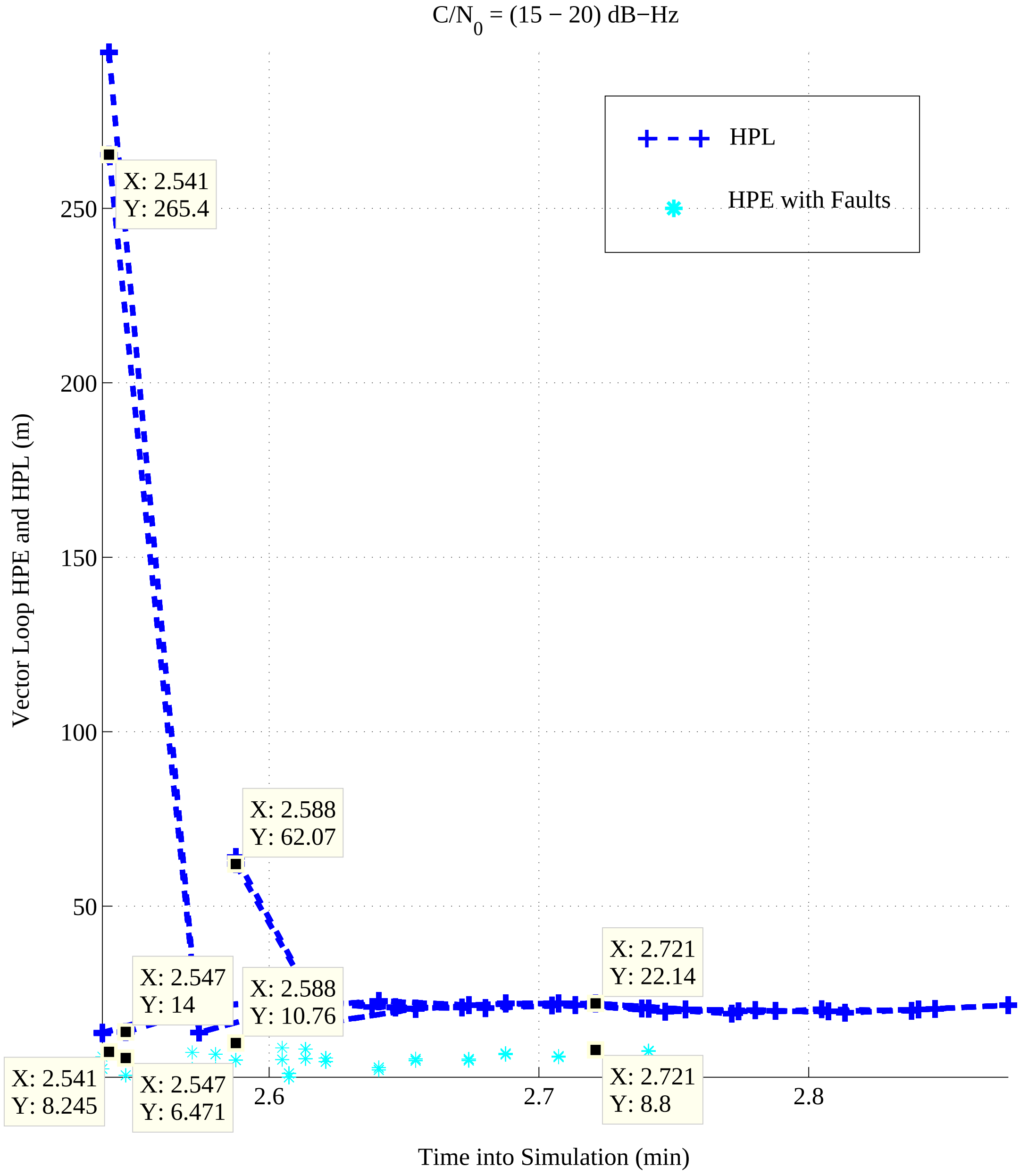


Figure 6.20: Zoom in View of Vector Loop HPL and HPE under *Fault* for Case IV for Sub-interval (2.6 - 3) min into Simulation

6.3.4 Sensitivity of Protection Levels to Number of Terms

It was discussed in Chapter 5 that a number of previous epoch terms in addition to the current epoch term need to be considered to calculate the protection levels for the EKF implementation of the navigation filter (see Section 5.3.1.3). A criterion (Equation 5.18) was also proposed to determine the number of terms N. To verify the validity of the criterion, the sensitivity of the HPL to N is shown in Figures 6.21 and 6.22 for one run of the collected Spirent data ($C/N_0 = (44 - 49)$ dB-Hz) and one run of the first set of low C/N_0 data ($C/N_0 = (25 - 30)$ dB-Hz), respectively. The HPLs are shown only for the vector architecture. The figures also show the HPLs that would be obtained with a least squares algorithm for comparison purposes. For the Spirent data N is found to be 13 or more using Equation 5.18. Since the corresponding HPL is not distinguishable from that with $N = 25$ when HPLs with different N are plotted together, the difference between the HPL with $N = 13$ and that with $N = 25$ is shown separately in the right subplot of Figure 6.21. The maximum difference between the two HPLs is 0.1138 m. Hence, $N = 13$ is sufficient for $C/N_0 = (44 - 49)$ dB-Hz. It is also important to note that the EKF HPL is greater than the HPL obtained from the least squares algorithm.

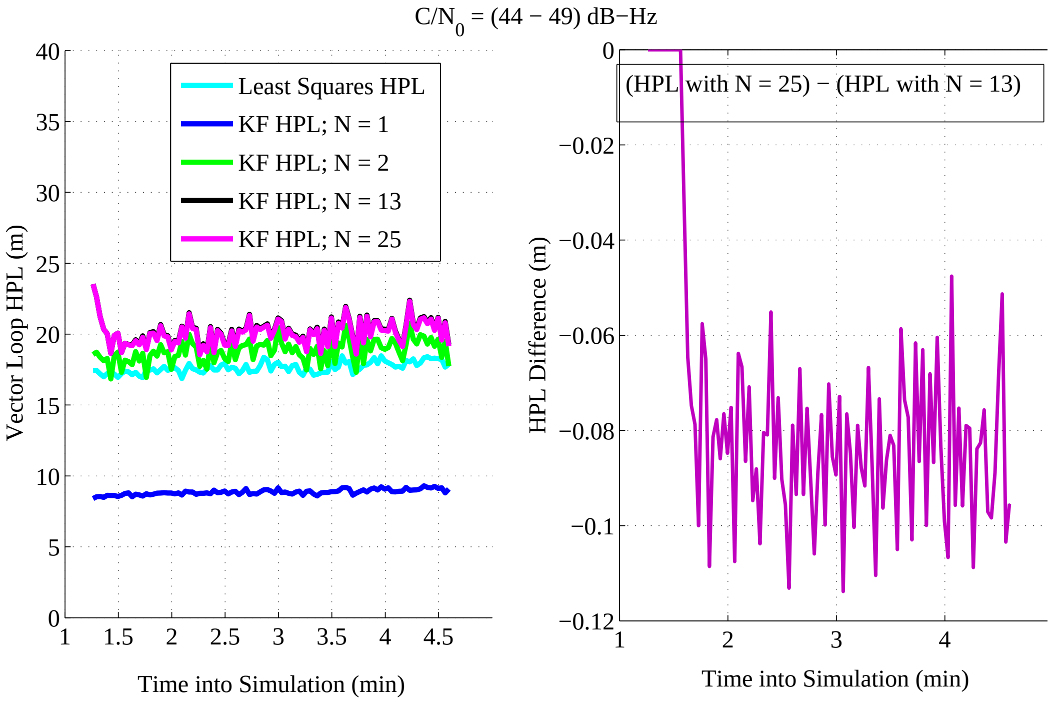


Figure 6.21: Sensitivity of HPL to the Number of Terms at $C/N_0 = (44 - 49)$ dB-Hz

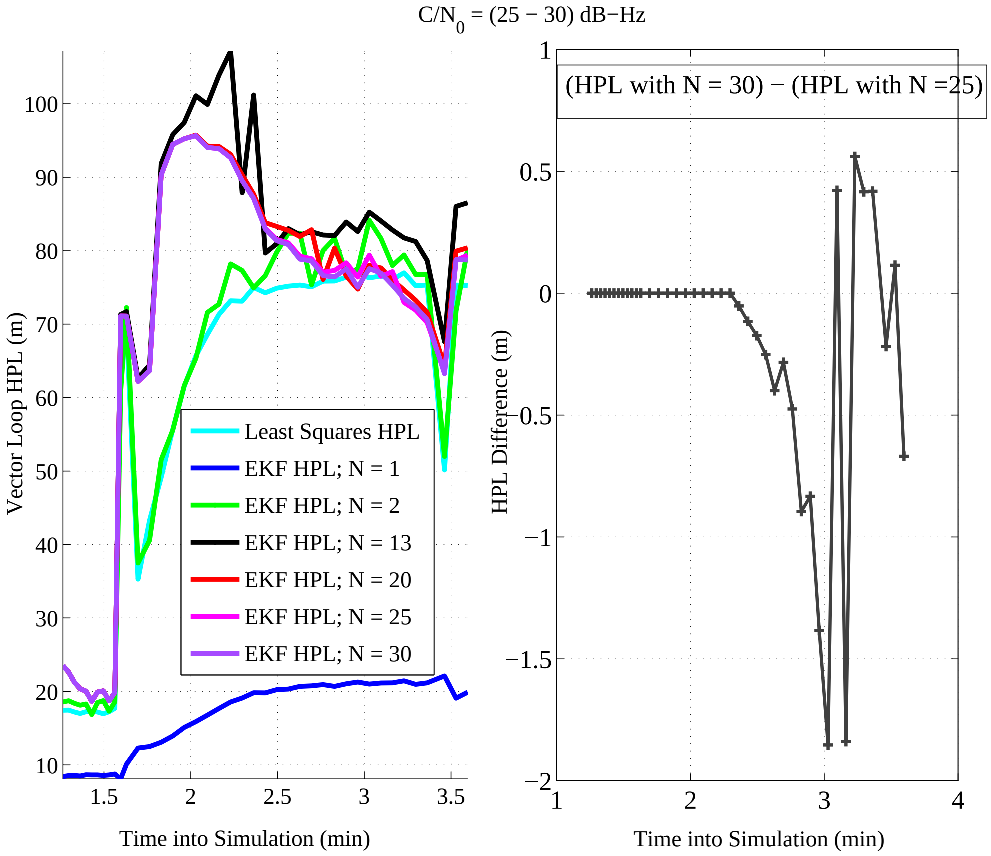


Figure 6.22: Sensitivity of HPL to the Number of Terms at C/N$_0$ = (25 - 30) dB-Hz

Next, Figure 6.22 shows the sensitivity of HPL to N for a dataset where the C/N$_0$ level is degraded to (25 - 30) dB-Hz at 1.6 min into simulation. Following Equation 5.18, N is found to be greater than 20 for this data. $N = 25$ is considered for protection level calculations for this C/N$_0$ level. The right subplot of Figure 6.22 shows the difference between the HPL with $N = 30$ and that with $N = 25$. Since the maximum difference between the two HPLs is on the order of 2 m, a higher value of N may be necessary. This, in turn, suggests that a more stringent criterion to determine N at low C/N$_0$ than Equation 5.18 is required.

6.4 Conclusion

In this chapter the vector loop RAIM algorithm developed in Chapter 5 is validated with GPS measurements. Data for this purpose was generated for the final approach phase of an aircraft using a Spirent simulator. Artificial Gaussian noise was added to the data to simulate intentional interference. With the generated data scalar and vector loop RAIM performance is compared. From the performance results the following conclusions are drawn: Vector loop RAIM produces lower protection levels than the scalar loop RAIM when the receiver is driven by a clock with large bias and drift. The performance improvement increases as the signal level degrades. At low C/N_0 such as below 20 dB-Hz when scalar loops lose track of signals, vector loop RAIM may be able to detect fault. But it can also become unavailable due to protection levels being greater than the alert limits. Because of its ability to track weak signals, however, it can provide integrity monitoring immediately after the C/N_0 level is restored. In a nutshell, the analysis results of this chapter show that the vector loop RAIM has the potential to improve the availability of the integrity monitoring algorithm in both good and bad signal environments. However, the performance of the vector loop RAIM can be sensitive to multipath, which requires extensive studies.

Chapter 7

Conclusions and Recommendations for Future Research

7.1 Summary

The aim of this thesis was to explore some of the key features of the vector architecture. While this architecture has proved to be a good candidate for dealing with weak signals, scant information is available about its integrity monitoring. The ability of the vector architecture to track weak signals, when combinded with integrity, can find use in safety-critical and liability-critical applications that are inherently susceptible to signal blockage or signal attenuation. Some of these include personal navigation system for visually impaired individuals and road tolling operation. In response, the research in this thesis first shed light on the internal workings of the vector architecture as the open literature offers meagre information in this regard. Then, leveraging the insights gained on internal operations of vector loops, a vector loop integrity monitoring algorithm was designed. The key points that can be drawn from this research are as follows:

1. Internal Doppler aiding among loops in vector tracking mode is the key to vector tracking's superiority. Utilizing some constraints that are inherent to the implementation of the vector architecture, good C/N_0 satellites aid low C/N_0 satellites in vector tracking mode. This finding suggests that scalar loops with Doppler aiding can provide similar performance as the vector architecture.
2. An important result of the vector loop transfer function models developed in Chapter 4 is the vector loop noise bandwidth. Two types of vector loop noise

bandwidths were defined: One is for performance analysis of the vector architecture and the other is for comparison with the scalar loops. The second bandwidth was shown to have the capability of adaptively adjusting its value with noise levels and user dynamics. Thus, it can be a good candidate for systematically designing a Doppler-aided scalar loop noise bandwidth, which is currently performed somewhat heuristically. Preliminary guidelines for designing a scalar loop based on vector loop noise bandwidth were developed in Chapter 4.

3. It was shown in Chapter 4 that a transient error in one channel easily propagates to other channels because of internal aiding among vector loops. Then, Chapters 5 and 6 showed that all of the vector loop pseudorange measurements get corrupted by a fault in a single channel. In addition, noise in the vector loop pseudorange measurements is correlated across channels. This is because the replica code phases from which the vector loop pseudorange measurements are derived are corrected by projecting the estimated user position onto LOS vectors. Thus, error in the estimated position gets translated to all the pseudorange measurements. Further, the analysis in Chapter 6 revealed that the vector loop pseudorange residuals do not show any inconsistency when a fault occurs in a channel. All these findings point that the vector loop pseudorange measurements are not suitable for designing an integrity monitoring or RAIM algorithm, unlike the scalar loop pseudorange measurements.

4. As an alternative, the code phase discriminator outputs which have the same mathematical model as the pseudorange residuals were used to design a RAIM algorithm with the vector architecture. The code phase discriminator outputs have uncorrelated noise across channels and comply with the single fault assumption (assuming ideal autocorrelation functions) as they are derived from the input to the vector loop navigation filter. They exhibit inconsistency under faulty conditions, which can be exploited by the fault detection algorithm to detect faults.

5. The protection level calculations of RAIM with the EKF implementation of the navigation filter required an altogether different treatment. This is because in the EKF past faults affect current estimates, unlike the least squares estimation algorithms.

6. Performance comparison of vector loop RAIM with scalar RAIM revealed that the

former offers lower protection levels in both good and bad signal environments. Moreover, because of the vector loops' ability to track weak signals, they can offer integrity monitoring as soon as signal levels are restored if integrity is not available at low signal powers. Thus, the vector loop RAIM will have higher availability than scalar loop RAIM in urban navigation where traditional scalar loop receivers suffer frequent signal attenuation or outages.

7.2 Future Research

The research contained in this thesis can be extended to several directions. They are noted below:

1. While the vector architecture is superior to the scalar architecture, its benefits come at the cost of complexity. In Chapter 4 preliminary guidelines for designing scalar loops that can achieve performance comparable to the vector loops were proposed. The next step is to implement the guidelines for an application in poor signal environments and study how the designed scalar architecture performs compared to the vector architecture and also in terms of computational load.

2. Although the vector loop RAIM has the potential to improve RAIM availability, injection of fault into real GPS measurements for the validation of vector RAIM with real data presents a significant challenge. The reason for this is that the fault should be injected at the system level rather than at the pseudorange level as with scalar loop RAIM. Data for the results shown in this thesis was generated using a Spirent simulator. Although this data is a good representation of the real world environment, it uses standard error models for simulating error sources such as atmospheric delays and multipath. Thus, a comprehesive performance analysis of vector loop RAIM requires real GPS measurements. One way of injecting fault into real GPS data, though somewhat complex, could be as follows. First, the real IF data can be processed using a Vector PLL and DLL or a deeply integrated GNSS/INS system with code and carrier phase tracking. Vector tracking or deep integration is recommended so that low C/N_0 signals can be tracked. Once the tracking loops are in lock, the replica code and carrier phases are aligned with the incoming code and carrier phases with sufficient accuracy and, hence, can be treated as equivalent to the incoming signal phases. Next, a drift or bias can be introduced to the code and carrier phases of the faulty channel and the signal

samples of this channel can be reconstructed using the modified phase. Then the signal samples of all channels can be added back to form the IF samples. While adding the signal samples, caution must be exercised to take into account the relative time skew of the received satellite signals. An easy but computationally expensive way would be to write all the signal samples of individual channels in separate files and then add them together on a point-by-point basis.

3. It was concluded in Chapter 6 that a fault in a channel triggers drifts of all replica code phases in vector tracking mode. In other words, all replica codes in vector tracking mode start drifting from their respective incoming codes under fault. As a result, if the code autocorrelation function of a channel is distorted from the ideal triangular shape, error in the code phase discriminator ouput of that channel can grow with time, resulting in violation of the single fault assumption even at the code phase discriminator level. This will make the developed vector loop RAIM algorithm ineffective. A potential cause for the distortions of the autocorrelation function is multipath. Thus, vector loop RAIM performance deserves extensive studies in multipath environment.

4. As noted earlier, with an EKF-based navigation filter past faults affect the current estimate. This complicates the computation of the protection levels because a number of past epoch terms have to be considered in addition to the current epoch term. Please refer to Section 5.3.1.3 of Chapter 5 for details. The formulation provided in this thesis holds for step, ramp and sinusoidal faults with time periods on the order of several hours. Suitable modifications are, therefore, needed to relax the assumption on the temporal characteristics of faults. Inclusion of the velocity states in the computation of the fault detection test statistic offers a partial solution, as noted in Section 5.3.1.3. However, even with that modification, the requirement for *different* position and velocity update rates of vector loops would require the above-mentioned assumption to hold true for protection level calculations (see the derivation in section 5.3.2.2).

5. The present method of protection level calculations assumes only user position, velocity, clock bias and drift as the navigation filter state vector. This formulation should be extended to account for user acceleration in order to allow for a better representation of user dynamics.

6. Further, the calculation of the protection levels for the coherent implementa-

tion of the vector tracking architecture will probably require an entirely different approach than that of the noncoherent implementation provided in this thesis.

7. Time correlated errors such as residual atmospheric delays and broadcast ephemeris and clock errors need to be adequately modeled or estimated to implement the second fault detection method of section 5.3.1.2 whose test statistic is the sum of the snapshot test statistics of current epoch plus a number of past epochs. Time correlated errors need to be properly accounted for because in the vector architecture successive measurements cannot be adequately spaced apart to counteract the effect of time-correlated errors as the user position has to be updated every few seconds to maintain code tracking.

8. In Chapter 6 vector loop RAIM was found to be unavailable at C/N_0 between 15 dB-Hz and 20 dB-Hz. Thus, it would be worth studying if the vector loop RAIM availability improves at low C/N_0 with multi-constellation satellite signals.

9. Next, the benefits of vector tracking in non-aviation applications such as Road User Charging (RUC) schemes [56] can be examined. The use of GNSS in such high-integrity applications is favored for its low costs and flexibility. However, frequent signal outages in urban environments remain a significant challenge for these applications and compromise the availability of scalar loop-based integrity monitoring. Under these circumstances, a comprehensive analysis can be carried out to determine whether a vector architecture-based RAIM algorithm can provide better system availability.

10. In view of the challenges associated with vector loop integrity monitoring as detailed in recommendations 2 through 4, 6 and 7 above, it is important to study if a well-designed scalar architecture, when driven by a low-drift clock, can provide similar integrity monitoring performance as the vector architecture. The scalar architecture can be designed following the guidelines proposed in Chapter 4 of this thesis. In addition, the advent of chip-scale atmoic clocks opens up the possibility of future receivers to be run by more accurate clocks.

References

[1] Rife, J. and Pullen, S, "Aviation Applications," Chapter 10 in *GNSS Applications and Methods*, Ed: Gleason, S. and Gebre-Egziabher, D., Artech House, Norwood, MA, 2009, pp. 245-268.

[2] http://www.gps.gov/applications/roads/.

[3] "Seen + Heard - Smooth Start for I-95 Traffic Network," *GPS World*, Vol. 19, No. 7, July 2008, pp. 50.

[4] "GPS for the Visually Impaired," http://en.wikipedia.org/wiki/GPS_for_the_visually_impaired.

[5] http://www.gps.gov/applications/timing/.

[6] http://www.gps.gov/applications/agriculture/.

[7] Diggelen, F. V., Abraham, C., Salas, J. D., and Randy, S., "GNSS Inside Mobile Phones: GPS, GLONASS, QZSS, and SBAS in a Single Chip," *Inside GNSS*, March/April 2011, Vol. 6, No. 2, pp. 50-60.

[8] http://www.berginsight.com/News.aspx?m_m=6&s_m=1.

[9] Gleason, S., Zavorotny, V. and Lowe, S. "Remote Sensing Using Bistatic GNSS Reflections," Chapter 16 in *GNSS Applications and Methods*, Ed: Gleason, S. and Gebre-Egziabher, D., Artech House, Norwood, MA, 2009, pp. 399-436.

[10] Lashley, M., "Modeling and Performance Analysis of GPS Vector Tracking Algorithms," PhD Thesis, Auburn University, 2009.

[11] Spilker Jr., J. J., "Fundamentals of Signal Tracking Theory," Chapter 7 in *Global Positioning System: Theory and Applications*, Vol. I, Ed: Parkinson, B. W. and Spilker Jr., J. J., AIAA, Washington, DC, 1995, pp. 245 - 327.

[12] Henkel, P., Giger, K., and Günther, C., "Multifrequency, Multisatellite Vector Phase-Locked Loop for Robust Carrier Tracking," *IEEE Journal of Selected Topics in Signal Processing,* Vol. 3, No. 4, December 2009, pp. 674-681.

[13] Rizos, C. and Grejner-Brzezinska, D. A., "Geodesy and Surveying," Chapter 14 in *GNSS Applications and Methods*, Ed: Gleason, S. and Gebre-Egziabher, D., Artech House, Norwood, MA, 2009, pp. 347-380.

[14] Chi, A. O., "Atmospheric Sensing Using GNSS Occultations," Chapter 15 in *GNSS Applications and Methods*, Ed: Gleason, S. and Gebre-Egziabher, D., Artech House, Norwood, MA, 2009, pp. 381-398.

[15] Lo, S. C., Lorenzo, D. D., and Enge, P., "Signal Authentication: A Secure Civil GNSS for Today," *Inside GNSS*, September/October 2009, Vol. 4, No. 5, pp. 30-39.

[16] National PNT Advisory Board Comments on Jamming the Global Positioning System - A National Security Threat: Recent Events and Potential Cures, November 4, 2010.

[17] GPS - DOD's profit Center, *InsideGNSS*, January/February 2012, pp. 10.

[18] Jwo, D. J., "Optimisation and Sensitivity Analysis of GPS Receiver Tracking Loops in Dynamic Environments," *IEE Proc. -Radar, Sonar Navigation*, Vol. 148, No. 4, August 2001, pp. 241-250.

[19] Rubin, D. and Young, T., "Femtocells Bringing Reliable Location and Timing Indoors," *Inside GNSS*, Fall 2008, pp. 40-46.

[20] Ward, P., "GPS Satellite Signal Acquisition and Tracking," Chapter 6 in *Understanding GPS: Principles and Application*, Ed: E. D. Kaplan, Artech House Publisher, Boston, MA, 1996, pp. 209-235.

[21] Jones, M., "The Civilian Battlefiled: Protecting Receivers from Interference and Jamming," *Inside GNSS*, March/April 2011, Vol. 6, No. 2, pp. 40-49.

[22] "Satellite Navigation & Space Weather: Understanding the Vulnerability & Building Resilience," *American Meteorological Society Policy Workshop Report*, March 2011.

[23] Kinter, Jr., P. M., Humphreys, T. and Hinks, J., "GNSS and Ionospheric Scintillation: How to Survive the Next Solar Maximum," *Inside GNSS*, July/August 2009, Vol. 4, No. 4, pp. 22-29.

[24] Knight, M. F., "Ionospheric Scintillation Effects on Global Positioning System Receivers," PhD Thesis, University of Adelaide, South Australia, December 2000.

[25] Gebre-Egziabher, D. Petovello, M. and Bevly, D., "Integration of GNSS and INS (Part I and II)," Chapters 6 and 7 in *GNSS Applications and Methods*, Ed: Gleason, S. and Gebre-Egziabher, D., Artech House, Norwood, MA, 2009, pp. 149-190.

[26] Veth, M., Anderson, R., Webber, F., Nielsen, M., "Tightly-Coupled INS, GPS, and Imaging Sensors for Precision Geolocation," *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, San Diego, CA, January 2008, pp. 487-496.

[27] De Lorenzo, David S., Lo, Sherman C., Enge, Per K., Feuerstein, Marty, Bhattacharya, Tarun K., Spain, Steve, Kang, Zhengjiu, "Design and Performance of a Minimum-Variance Hybrid Location Algorithm Utilizing GPS and Cellular Received Signal Strength for Positioning in Dense Urban Environments," *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, Anaheim, CA, January 2009, pp. 784-792.

[28] Zhao, Xing, Goodall, Chris, Syed, Zainab, Wright, Bruce, El-Sheimy, Naser, "Wi-Fi Assisted Multi-sensor Personal Navigation System for Indoor Environments," *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation*, San Diego, CA, January 2010, pp. 236-243.

[29] Nielsen, Michael B, Raquet, John F., Veth, Michael, Pachter, Meir, "Development and Flight Test of a Robust Optical-Inertial Navigation System Using Low-Cost Sensors," *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, Savannah, GA, September 2008, pp. 2650-2661.

[30] Smearcheck, M.A., Veth, M.J., “Performance of a Next Generation Integrated Multi-Sensor Time-Space Position Information (TSPI) System,” *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)*, Portland, OR, September 2010, pp. 2489-2499.

[31] McCroskey, R., Samanant, P., Hawkinson, W., Huseth, S., Hartman, R., “GLANSER An Emergency Responder Locator System for Indoor and GPS-Denied Applications,” *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)*, Portland, OR, September 2010, pp. 2901-2909.

[32] Groves, Paul D., Long, Daniel C., “Inertially-Aided GPS Signal Re-Acquisition in Poor Signal to Noise Environments and Tracking Maintenance Through Short Signal Outages,” *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, Long Beach, CA, September 2005, pp. 2408-2417.

[33] Copps, E. M., Geier, G. J., Fidler, W. C., Grundy, P. A., “OPTIMAL PROCESSING OF GPS SIGNALS,” *NAVIGATION*, Vol. 27, No. 3, Fall 1980, pp. 171-182.

[34] Sennott, James W., Senffner, David, “THE USE OF SATELLITE GEOMETRY FOR PREVENTION OF CYCLE SLIPS IN A GPS PROCESSOR,” *NAVIGATION*, Vol. 30, No. 2, Summer 1992, pp. 217-235.

[35] Zhodzishsky, M., Yudanov, S., Veitsel, V., J. Ashjaee, ”Co-op Tracking for Carrier Phase,” *Proceedings of the 11th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1998)*, Nashville, TN, Sep. 1998, pp. 653-664, Also U.S. patent (6,313,789 B1), June 1998.

[36] Pany, T., Eissfeller, B., “Use of a Vector Delay Lock Loop Receiver for GNSS Signal Power Analysis in Bad Signal Conditions,” *Proceedings of IEEE/ION PLANS 2006*, San Diego, CA, April 2006, pp. 893-903.

[37] Petovello, M.G., Lachapelle, G., “Comparison of Vector-Based Software Receiver Implementations with Application to Ultra-Tight GPS/INS Integration,” *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, Fort Worth, TX, September 2006, pp. 1790-1799.

[38] Benson, D., "Interference Benefits of a Vector Delay Lock Loop (VDLL) GPS Receiver," *Proceedings of the 63rd Annual Meeting of The Institute of Navigation*, Cambridge, MA, April 2007, pp. 749-756.

[39] Gustafson, D., Dowdle, J., "Deeply Integrated Code Tracking: Comparative Performance Analysis," *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, Portland, OR, September 2003, pp. 2553-2561.

[40] Ohlmeyer, E. J., "Analysis of an Ultra-Tightly Coupled GPS/INS System in Jamming," *Proceedings of IEEE/ION PLANS 2006*, San Diego, CA, April 2006, pp. 44-53.

[41] Groves, Paul D., Mather, Christopher J., Macaulay, Alex A., "Demonstration of Non-coherent Deep INS/GPS Integration for Optimised Signal-to-Noise Performance," *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, Forth Worth, TX, September 2007, pp. 2627-2638.

[42] Soloviev, Andrey, Gunawardena, Sanjeev, van Graas, Frank, "Deeply Integrated GPS/Low Cost IMU for Low CNR Signal Processing: Concept Description and In-Flight Demonstration," *NAVIGATION*, Vol. 55, No. 1, Spring 2008, pp. 1-13.

[43] Pany, T., Winkel, J., Riedl, B., Restle, M., Wörz, T., Schweikert, R., Niedermeier, H., Ameres, G., Eissfeller, B., Lagrasta, S., López-Risueño, G.,"Performance of a Partially Coherent Ultra-Tightly Coupled GNSS/INS Pedestrian Navigation System Enabling Coherent Integration Times of Several Seconds to Track GNSS Signals Down to 1.5 dBHz," *Proceedings of the 22nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2009)*, Savannah, GA, September 2009, pp. 919-934.

[44] Abbott, A. S. and Lillo, W. E., "Global Positioning Systems and Inertial Measuring Unit Ultratight Coupling Method," *U. S. Patent 6,516,021, The Aerospace Corporation*, Segundo, CA, 2003.

[45] Buck, Timothy M., Wilmot, Jason, Cook, Michael J, "A High G, MEMS Based, Deeply Integrated, INS/GPS, Guidance, Navigation and Control Flight Management Unit," *Proceedings of IEEE/ION PLANS 2006*, San Diego, CA, April 2006, pp. 772-794.

[46] Horslund, J. M. and Hooker, J. R., "Increase Jamming Immunity by Optimizing Processing Gain for GPS/INS Systems," *U. S. Patent 5,983160, Raytheon Company*, Lexington, MA, 1999.

[47] Crane, R. N., "A Simplified Method for Deep Coupling of GPS and Inertial Data," *Proceedings of the 2007 National Technical Meeting of The Institute of Navigation*, San Diego, CA, January 2007, pp. 311-319.

[48] Jovancevic, A., Ganguly, S., "Real-Time Implementation of a Deeply Integrated GNSS-INS Architecture," *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, Long Beach, CA, September 2005, pp. 503-511.

[49] Soloviev, Andrey, Bruckner, Dean, van Graas, Frank, Marti, Lukas, "Assessment of GPS Signal Quality in Urban Environments Using Deeply Integrated GPS/IMU," *Proceedings of the 2007 National Technical Meeting of The Institute of Navigation*, San Diego, CA, January 2007, pp. 815-828.

[50] Landis, D., Thorvaldsen, T., Fink, B., Sherman, P., Holmes, S., "A Deep Integration Estimator for Urban Ground Navigation," *Proceedings of IEEE/ION PLANS 2006*, San Diego, CA, April 2006, pp. 927-932.

[51] Li, Tao, Petovello, Mark G., Lachapelle, Gerard, Basnayake, Chaminda, "Ultratightly Coupled GPS/Vehicle Sensor Integration for Land Vehicle Navigation," *NAVIGATION*, Vol. 57, No. 4, Winter 2010-2011, pp. 263-274.

[52] Parkinson, Bradford W., Axelrad, Penina, "AUTONOMOUS GPS INTEGRITY MONITORING USING THE PSEUDORANGE RESIDUAL", *NAVIGATION,* Vol. 35, No. 2, Summer 1986, pp. 255-274.

[53] Walter, T., Enge, Per, "Weighted RAIM for Precision Appproach," *Proceedings of the 8th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1995)*, Palm Springs, CA, September 1995, pp. 1995-2004.

[54] Pervan, B. S., Lawrence, D. G. and Parkinson, B. W., "Autonomous Fault Detection and Removal Using GPS Carrier Phase," *IEEE Transactions on AES*, Vol. 34, No. 3, July 1998, pp. 897-905.

[55] Walter, Todd, Blanch, Juan, Enge, Per, Pervan, Boris, Gratton, Livio, "Future Architectures to Provide Aviation Integrity," *Proceedings of the 2008 National*

Technical Meeting of The Institute of Navigation, San Diego, CA, January 2008, pp. 394-401.

[56] Salós, D., Martineau, A., Macabiau, C., Kubrak, D., Bonhoure, B., "Groundwork for GNSS Integrity Monitoring in Urban Road Applications. The Road User Charging Case," *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)*, Portland, OR, September 2010, pp. 1130-1144.

[57] Ding, Y., Sennott, J., "Ultra-Tight Coupling (UTC) Technique in GPS RAIM," *Proceedings of the 17th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2004)*, Long Beach, CA, September 2004, pp. 1803-1812.

[58] Misra, P. and Enge, P., "Signal-to-Noise Ratio and Ranging Precision," Chapter 8 in *Global Positioning System, Signals, Measurements, and Performance*, Ganga-Jamuna Press, Lincoln, MA, 2001, pp. 283-316.

[59] Navstar GPS, Interface Specification, IS-GPS-200, Revision E, Navstar GPS Space Segment/Navigation User Interfaces, June 2010.

[60] Gleason, S. and Gebre-Egziabher, D., "Global Navigation Satellite Systems: Present and Future," Chapter 1 in *GNSS Applications and Methods*, Ed: Gleason, S. and Gebre-Egziabher, D., Artech House, Norwood, MA, 2009, pp. 1-21.

[61] Jaewoo, J., "High Integrity Carrier Phase Navigation Using Multiple Civil GPS Signals," PhD Thesis, Stanford University, August 2000.

[62] Misra, P. and Enge, P., "GPS Receivers," Chapter 8 in *Global Positioning System, Signals, Measurements, and Performance*, Ganga-Jamuna Press, Lincoln, MA, 2001, pp. 317-374.

[63] Van Dierendonck, A. J., "GPS Receivers," Chapter 8 in *Global Positioning System: Theory and Applications*, Vol. I, Ed: Parkinson, B. W. and Spilker Jr., J. J., AIAA, Washington, DC, 1995, pp. 329-407.

[64] Razavi, A., "Sensitivity and Performance Analysis of Doppler Aided GPS Carrier Tracking Loops," Master Thesis, University of Minnesota, Twin cities, December 2004.

[65] Meyr, H. and Ascheid, G., “Phase-Locked Loop Tracking Performance in the Presence of Noise,” Chapter 3 in *Synchronization in Digital Communications: Phase-, Frequency-Locked Loops, and Amplitude Control*, Vol. I, John Wiley & Sons, pp. 99-162.

[66] Spilker, J. J., “Carrier-Phase Tracking and Oscillator-Phase Noise,” Chapter 12 in *Digital Communications by Satellite*, Prentice-Hall, Inc, Englewood, NJ, 1977, pp. 336-428.

[67] Groves, P. D., “Satellite Navigation Processing, Errors, and Geometry,” Chapter 7 in *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Artech House, Boston, London, 2008, pp. 195-278.

[68] Ward, P., “GPS Satellite Signal Acquisition and Tracking,” Chapter 5 in *Understanding GPS: Principles and Application*, Ed: E. D. Kaplan, Artech House Publisher, Boston, MA, 1996, pp. 119-208.

[69] Tsui, J. B., “Acquisition of GPS C/A Code Signals,” Chapter 7 in *Fundamentals of Global Positioning System Receivers: A Software Approach*, Ed: Chang K., John Wiley & Sons, Hoboken, New Jersy, pp. 129-159.

[70] Groves, Paul D., “GPS Signal to Noise Measurement in Weak Signal and High Interference Environments,” *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, Long Beach, CA, September 2005, pp. 643-658.

[71] Van Dierendonck, A. J., Fenton, Pat, Ford, Tom, ”THEORY AND PERFORMANCE OF NARROW CORRELATOR SPACING IN A GPS RECEIVER”, NAVIGATION, Vol. 39, No. 3, Fall 1992, pp. 265-284.

[72] Psiaki, Mark L., Jung, Hee, ”Extended Kalman Filter Methods for Tracking Weak GPS Signals,” *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, Portland, OR, September 2002, pp. 2539-2553.

[73] Axelrad, P. and Brown, R. G., “GPS Navigation Algorithms,” Chapter 9 in *Global Positioning System: Theory and Applications*, Vol. I, Ed: Parkinson, B. W. and Spilker Jr., J. J., AIAA, Washington, DC, 1995, pp. 409-433.

[74] Groves, P. D., "INS/GNSS Integration," Chapter 12 in *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Artech House, Boston, London, 2008, pp. 363-406.

[75] Razavi, A., Gebre-Egziabher, D. and Akos, D. M., "Carrier Loop Architectures for Tracking Weak GPS Signals," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. 44, No. 2, April 2008, pp. 697-710.

[76] US Department of Defense, Appendix A in *Global Positioning System Standard Positioning Service (SPS) Performance Standard*, 4th Edition, Sep. 2008.

[77] Kailath, T., "Some Facts from Matrix Theory," Appendix in *Linear System*, Prentice Hall, New Jersey, 1980, pp. 656.

[78] Zarchan, P. and Musoff, H., "Continuous Polynomial Kalman Filter," Chapter 6 in *Fundamentals of Kalman Filtering: A Practical Approach*, Progress in Aeronautics and Astronautics, Vol. 190, Virginia, 2001, pp. 219-255.

[79] Groves, P.D., Long, D.C., "Adaptive tightly-coupled, a low cost alternative anti-jam INS/GPS integration technique," *Proceedings of the 2003 National Technical Meeting of The Institute of Navigation,* Anaheim, CA, January 2003, pp. 429-440.

[80] Rempfler, A. and Mathis, H., "GPS/GPRS on the Road: Virtual Radar for a Swiss Bus Fleet," *Inside GNSS*, Fall 2007, pp. 32-36.

[81] Cosmen-Schortmann, J., Azaola-Saenz, M., Martinez-Olague, M.A., Toledo-Lopez, M., "Integrity in Urban and Road Environments and its Use in Liability Critical Applications," *Proceedings of IEEE/ION PLANS 2008,* Monterey, CA, May 2008, pp. 972-983.

[82] Walter, T., Enge, P., Blanch, J. and Pervan, B., "Worldwide Vertical Guidance of Aircraft Based on Modernized GPS and New Integrity Augmentations," *Proceedings of the IEEE*, Vol. 26, No. 12, Dec. 2008.

[83] Joerger, Mathieu, Gratton, Livio, Pervan, Boris, Cohen, Clark E., "Analysis of Iridium-Augmented GPS for Floating Carrier Phase Positioning", *NAVIGATION*, Vol. 57, No. 2, Summer 2010, pp. 137-160.

[84] Shivaramaiah, N.C., "Code Phase Multipath Mitigation by Exploiting the Frequency Diversity in Galileo E5 AltBOC," *Proceedings of the 22nd International*

Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009), Savannah, GA, September 2009, pp. 3219-3233.

[85] Joerger, M., Pervan, B., "Integrity Risk of Kalman Filter-Based RAIM," *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2011)*, Portland, OR, September 2011.

[86] Simon, D., "Additional Topics in Kalman Filtering," Chapter 10 in *Optimal State Estimation: Kalman,* H_∞ *and Nonlinear Approaches*, John Wiley & Sons Inc., Hoboken, NJ, pp. 297-332.

[87] Angus, J. E., "RAIM with Multiple Faults", NAVIGATION, Vol. 53, No. 4, Winter 2006-2007, pp. 249-257.

[88] Joerger, M., Neale, J., Pervan, B., Datta-Barua, S., "Measurement Error Models and Fault-Detection Algorithms for Multi-Constellation Navigation Systems," *Proceedings of IEEE/ION PLANS 2010*, Indian Wells, CA, May 2010, pp. 927-946.

[89] Pervan, B., "Navigation Integrity for Aircraft Precision Landing Using the Global Positioning System," PhD Thesis, Stanford University, March 1996.

[90] DGP00686AAA SimGEN Software User manual, Spirent Communications.

[91] Bauregger, Frank N., Walter, Todd, Akos, Dennis, Enge, Per, "A Novel Dual Patch Anti Jam GPS Antenna," *Proceedings of the 58th Annual Meeting of The Institute of Navigation and CIGTF 21st Guidance Test Symposium*, Albuquerque, NM, June 2002, pp. 516-522.

[92] GPS Risk Assessment Study Final Report, the John Hopkins University Applied Physics Laboratory, VS-99-007, January 1999, M8A01 Revised.

[93] Phelts, R. E., "Multicorrelator Techniques for Robust Mitigation of Threats to GPS Signal Quality," PhD Thesis, Stanford University, June 2001.

Appendix A

A.1 Derivation of σ_{IQ}^2

A mathematical expression of σ_{IQ}^2 is derived here considering bandlimited pre-correlation noise. The derivation is given below.

The digital IF signal input to a tracking loop is represented as

$$r_k = AC_kD_k \cos[2\pi(f_I + f_{d_k})t_k + \theta_0] + n_k \tag{A.1}$$

where A is the signal amplitude after the receiver front end processing; C_k is the received C/A code at time t_k; D_k is the received navigation data bit at time t_k; f_I is the IF frequency; f_{d_k} is the Doppler frequency at t_k; θ_0 is a constant phase shift; and n_k is bandpass stationary noise. The satellite index has been dropped for simplicity of notation.

The power spectral density (PSD) function of n_k is given as

$$\Phi_n(f) = \begin{cases} N_0/2 & \text{if } |f \pm f_I| \leqslant \dfrac{B}{2} \\ 0 & \text{otherwise} \end{cases}$$

where is B is the two-sided pre-correlation bandwidth. The low pass equivalent ($n_{l_k} = x_k + jy_k$) of n_k has the following PSD

$$\Phi_{n_l}(f) = \begin{cases} N_0 & \text{if } |f| \leqslant \dfrac{B}{2} \\ 0 & \text{otherwise} \end{cases}$$

Note that approximate values of B and N_0 can be obtained from the PSD spectrum of the IF data as the signal is below the noise floor in IF samples.

n_k is represented in the time domain as

$$\begin{aligned} n_k &= \mathrm{Re}\{n_{l_k} \exp^{j2\pi f_I t_k}\} \\ &= x_k \cos(2\pi f_I t_k) - y_k \sin(2\pi f_I t_k) \end{aligned} \tag{A.2}$$

x_k and y_k are uncorrelated Gaussian noise processes at time t_k with the following statistics: $E[x_k] = E[y_k] = 0$. $E[x_k^2] = \sigma_x^2 = E[y_k^2] = \sigma_y^2 = N_0B$. Hence, $E[n_k] = 0$, $E[n_k^2] = \sigma_n^2 = N_0B$. It can be shown that the autocorreation function of n_k is

$$\phi_{n_m} = E(n_k^T n_{k+m}) = \phi_{x_m} \cos[2\pi f_I(t_{k+m} - t_k)] \tag{A.3}$$

where ϕ_x is the autocorrelation function of x_k.

After the inphase (I) and quadrature (Q) processing and carrier wipe off of the incoming signal [62], the inphase and quadrature channel noises are represented as

$$n_{I_k} = \frac{x_k}{\sqrt{2}} \cos(2\pi \hat{f}_{d_k} t_k + \hat{\theta}_0) + \frac{y_k}{\sqrt{2}} \sin(2\pi \hat{f}_{d_k} t_k + \hat{\theta}_0) \tag{A.4}$$

$$n_{Q_k} = \frac{y_k}{\sqrt{2}} \cos(2\pi \hat{f}_{d_k} t_k + \hat{\theta}_0) - \frac{x_k}{\sqrt{2}} \sin(2\pi \hat{f}_{d_k} t_k + \hat{\theta}_0) \tag{A.5}$$

where $\hat{f}_{d_k}$ denotes estimated Doppler frequency and $\hat{\theta}_0$ is estimated carrier phase offset in the beginning of the k^{th} correlation interval. $\hat{\theta}_0$ is zero when only the Doppler frequency is tracked. For ease of notation the term $(2\pi \hat{f}_{d_k} t_k + \hat{\theta}_0)$ is now represented as $\hat{\theta}_k$. As for the noise statistics, $E[n_{I_k}] = E[n_{Q_k}] = 0$. $E[n_{I_k}^2] = E[n_{Q_k}^2] = N_0B/2$. $E[n_{I_k}^T n_{Q_k}] = 0$.

Carrier wipe off is followed by code wipe off. During the code wipe off process the I and Q samples are multiplied by the corresponding replica C/A code samples. Then, M samples are added over a coherent integration time T to form the I and Q correlator outputs. Similar to n_{I_k} and n_{Q_k}, the I and Q correlator noises are uncorrelated and have identical noise statistics. So, both of them are represented by η_k. Their noise statistics are: $E(\eta_k) = 0$ and $E(\eta_k^2) = \sigma_{IQ}^2$. Next, σ_{IQ}^2 of the I correlator noise is derived. The same derivation also holds for the Q correlator. For convenience $\frac{x_k}{\sqrt{2}}$ is represented as $\tilde{x}_k$ with variance $\sigma_{\tilde{x}}^2 = N_0B/2$.

By definition, σ_{IQ}^2 is

$$\sigma_{IQ}^2 = E[\sum_{i=1}^{M} C_i n_{I_i} \sum_{j=1}^{M} C_j n_{I_j}]$$

$$= E[\sum_{i=1}^{M}\sum_{j=1}^{M} C_i C_j n_{I_i} n_{I_j}] \tag{A.6}$$

Since the C/A code and IF noise are uncorrelated and the summation and the expectation operations are commutative, one writes

$$\begin{aligned}\sigma_{IQ}^2 &= \sum_{i=1}^{M}\sum_{j=1}^{M} E[C_i C_j] E[n_{I_i} n_{I_j}] \\ &= \sum_{i=1}^{M}\sum_{j=1}^{M} R_{ij} \phi_{n_{I_{ij}}} \end{aligned} \tag{A.7}$$

where R is the C/A code autocorrelation function. Now using Equations A.3 and A.4

$$\begin{aligned}\sigma_{IQ}^2 &= \sum_{i=1}^{M}\sum_{j=1}^{M} R_{ij} \phi_{\tilde{x}_{ij}} \cos(\hat{\theta}_i - \hat{\theta}_j) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{M} R_{ij} \phi_{\tilde{x}_{ij}} \cos(\hat{\theta}_{ij}) \end{aligned} \tag{A.8}$$

Based on an ideal correlation of the C/A code, R_{ij} is given by

$$R_{ij} = \begin{cases} 1 - |\tau_{ij}| & \text{if } |\tau_{ij}| < 1 \\ 0 & \text{otherwise} \end{cases}$$

τ_{ij} is the phase difference between the i^{th} and j^{th} samples of the C/A code in chips.

$\phi_{\tilde{x}_{ij}}$ is defined as

$$\begin{aligned}\phi_{\tilde{x}_{ij}} &= \frac{N_0 B}{2} \frac{\sin[\pi B(t_i - t_j)]}{\pi B(t_i - t_j)} \\ &= \frac{N_0 B}{2} \text{sinc}[B(t_i - t_j)] \end{aligned}$$

To simplify subsequent derivations, the pre-corre-lation noise bandwidth B and $(t_i - t_j)$ in the above equation are replaced with f_s/l ($l \geq 1$) and $(i-j)/f_s$, respectively, where f_s is the sampling frequency. Thus, the above equation becomes

$$\phi_{\tilde{x}_{ij}} = \frac{N_0 f_s}{2l} \text{sinc}\left(\frac{i-j}{l}\right) \tag{A.9}$$

Figure A.1 shows a pictorial representation of R_{ij} and normalized $\phi_{\tilde{x}_{ij}}$ for $f_s =$ 20 MHz and $B = 6$ MHz. It is evident from the mathematical expression of R_{ij} and

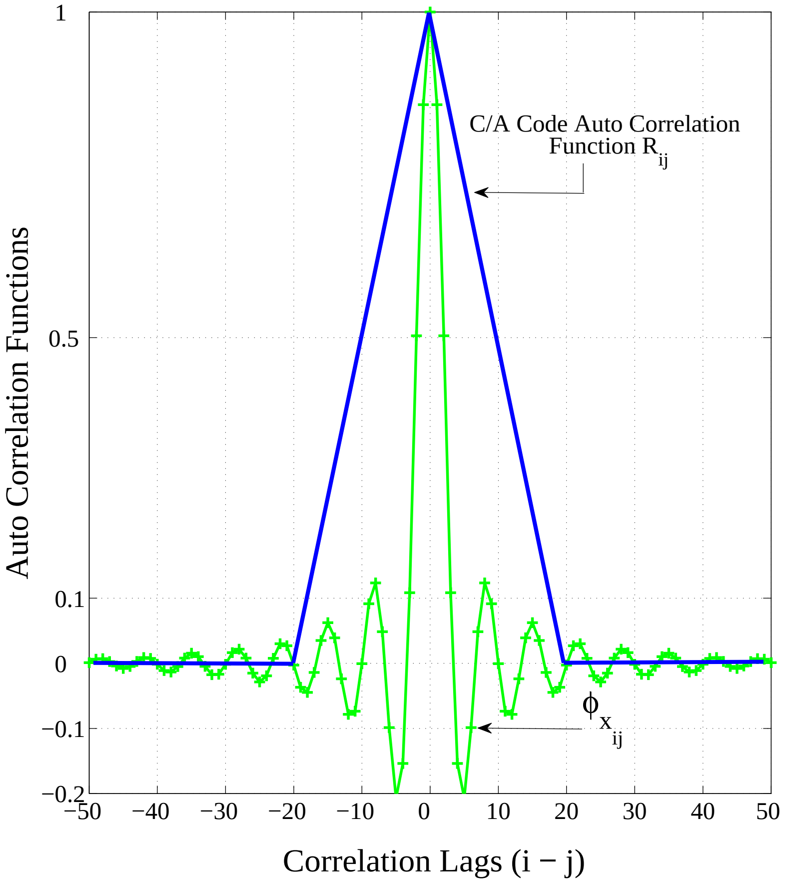


Figure A.1: Auto Correlation Functions

also from the figure that R_{ij} becomes zero when $(i-j)$ exceeds a certain integer value (say $N = \lfloor f_s/f_{\text{code}} \rfloor$, where f_{code} is the C/A code chipping rate and $\lfloor a \rfloor$ denotes the largest integer not exceeding a). As a result, for a given i, j needs to be varied only from max[1,$(i-N)$] to min[$(i+N)$, M]. Thus, after modifying the upper and lower limits of j and substituting R_{ij} and $\phi_{x_{ij}}$, Equation A.8 changes to

$$\sigma_{IQ}^2 = \frac{N_0 f_s}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} (1 - |\tau_{ij}|)\text{sinc}\left(\frac{i-j}{l}\right) \cos(\hat{\theta}_{ij}) \tag{A.10}$$

Although σ_{IQ}^2 can be computed numerically using the above equation, a simplified form of σ_{IQ}^2 is derived next. In order to do so, the following approximation is made to the value of $\cos\hat{\theta}_{ij}$ $(= \cos[2\pi \hat{f}_d (i-j)/f_s])$ in Equation A.10. $\hat{\theta}_{ij}$ becomes maximum when $i - j = \pm N = \pm\lfloor f_s/f_{\text{code}} \rfloor$. Assuming a maximum Doppler shift of 10 KHz for high-speed vehicles including most high-speed aircraft [69] and substituting f_s/f_{code} for

$(i-j)$, the lowest possible value of $\cos(\hat{\theta}_{ij})$ is calculated as $\cos[2\pi \times 10^4/(1.023 \times 10^6)]$ $= 0.998$. Thus, $\cos(\hat{\theta}_{ij})$ varies between 0.998 and 1 over a coherent integration interval. Therefore, is is approximated to be 1 in the derivation.

Now with some algebra Equation A.10 is simplified as follows:

$$\begin{aligned}
\sigma_{IQ}^2 &= \frac{N_0 f_s}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} (1-|\tau_{ij}|)\operatorname{sinc}\left(\frac{i-j}{l}\right) \\
&= \frac{N_0 f_s}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} \operatorname{sinc}\left(\frac{i-j}{l}\right) - \\
&\quad \frac{N_0 f_s}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} |\tau_{ij}|\operatorname{sinc}\left(\frac{i-j}{l}\right)
\end{aligned}$$

Substituting $(i-j) \times f_{\text{code}}/f_s$ for τ_{ij}, one gets

$$\begin{aligned}
\sigma_{IQ}^2 &= \frac{N_0 f_s}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} \operatorname{sinc}\left(\frac{i-j}{l}\right) - \\
&\quad \frac{N_0 f_s}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} |i-j|\frac{f_{\text{code}}}{f_s}\operatorname{sinc}\left(\frac{i-j}{l}\right) \\
&= \frac{N_0 f_s}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} \operatorname{sinc}\left(\frac{i-j}{l}\right) - \\
&\quad \frac{N_0 f_{\text{code}}}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} |i-j|\operatorname{sinc}\left(\frac{i-j}{l}\right) \qquad \text{(A.11)}
\end{aligned}$$

The first term of the above equation is denoted as $\sigma_{IQ_1}^2$ and the second term as $\sigma_{IQ_2}^2$ and simplify them separately. As for $\sigma_{IQ_1}^2$, the improper integral of the sinc function is used as follows:

$$\begin{aligned}
\int_{-\infty}^{\infty} \operatorname{sinc}(x)dx &= 2\int_{0}^{\infty} \operatorname{sinc}(x)dx \\
&= 2\lim_{b\to\infty} \int_{0}^{b} \operatorname{sinc}(x)dx \\
&= 1
\end{aligned}$$

Approximating the above integration by a discrete sum

$$\lim_{c\to\infty} 2\sum_{x_j=1}^{x_j=\,c} \mathrm{sinc}(x_j)\Delta x + \mathrm{sinc}(0)\Delta x = 1$$

where $\{x_j\} \in \mathbf{R}$ represents equally spaced sample points, with increment $\Delta x = x_{j+1}$ - x_j being the same for all j. Dividing both sides of the above equation by Δx leads to

$$\lim_{c\to\infty} 2\sum_{x_j=1}^{x_j=\,c} \mathrm{sinc}(x_j) + 1 = \frac{1}{\Delta x}$$

Substituting $(i-j)/l$ for x_j and $1/l$ for Δx, the above equation reduces to

$$\lim_{c\to\infty} 2\sum_{(i-j)=1}^{(i-j)=\,c} \mathrm{sinc}\left(\frac{i-j}{l}\right) + 1 = l \tag{A.12}$$

It is observed from Figure A.1 that $\phi_{\tilde{x}_{ij}}$ has m oscillations of decreasing amplitude on either side of non-zero R_{ij}, where $m = \lfloor B/(2f_{\text{code}})\rfloor = 3$. $\lfloor a\rfloor$ denotes the largest integer not exceeding a. It is also evident that the decay rate of $\phi_{\tilde{x}_{ij}}$ decreases with large values of $(i-j)$. In addition, it follows from the mathematical expression of $\phi_{\tilde{x}_{ij}}$ that it decays faster as m increases. Therefore, when $m \geqslant 3$ i.e., when the pre-correlation bandwidth is fairly larger than the two-sided C/A code bandwidth, the following approximation is made

$$\lim_{c\to N} 2\sum_{(i-j)\,=\,1}^{(i-j)\,=\,c} \mathrm{sinc}\left(\frac{i-j}{l}\right) + 1 = \sum_{(i-j)\,=\,-N}^{(i-j)\,=\,N} \mathrm{sinc}\left(\frac{i-j}{l}\right) \approx l$$

For a given i the limits of the summation operation are changed as follows:

$$\sum_{j\,=\,i-N}^{j\,=\,i+N} \mathrm{sinc}\left(\frac{i-j}{l}\right) \approx l \tag{A.13}$$

Comparing Equations A.11 and A.13, one can conclude that Equation A.13 holds true for all i except for $i \leq N$ and $i > M$ - N. For these outlying values of i, the left hand side (LHS) of Equation A.13 lies in the range of $[l-\Delta_1,\ l+\Delta_2]$, where Δ_1 and Δ_2 are positive real numbers and $< l/2$. As defined earlier, M is the number of samples that are added over a given coherent integration time T. Without data bit wipe-offs, T can vary from one code epoch to 20 code epochs. So, M is at least 1023 times larger than N which is the number of samples in one code chip. As $M \gg N$, the first N and the last N sample points of T (for which the LHS of Equation A.13 is different from l) will have an insignificant effect on the total sum. Thus, assuming Equation A.13 holds for all i, $\sigma^2_{IQ_1}$ is approximately given as

$$
\begin{aligned}
\sigma^2_{IQ_1} &= \frac{N_0 f_s}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} \text{sinc}\left(\frac{i-j}{l}\right) \\
\sigma^2_{IQ_1} &\approx \frac{N_0 f_s}{2l} \sum_{i=1}^{M} l \\
\sigma^2_{IQ_1} &\approx \frac{M N_0 f_s}{2} \\
\sigma^2_{IQ_1} &\approx \frac{M^2 N_0}{2T} \qquad \text{(A.14)}
\end{aligned}
$$

It should be noted that the above value of $\sigma^2_{IQ_1}$ matches that of σ^2_{IQ} obtained using the white noise assumption [63].

Now $\sigma^2_{IQ_2}$ is simplified as follows:

$$
\begin{aligned}
\sigma^2_{IQ_2} &= \frac{N_0 f_{\text{code}}}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=\min[i+N,\, M]} |i-j| \text{sinc}\left(\frac{i-j}{l}\right) \\
&= \frac{N_0 f_{\text{code}}}{2l} \sum_{i=1}^{M} \sum_{j=\max[1,\, i-N]}^{j=i} \frac{l}{\pi} \sin\left(\pi \frac{i-j}{l}\right) - \\
&\frac{N_0 f_{\text{code}}}{2l} \sum_{i=1}^{M} \sum_{j=i+1}^{j=\min[i+N,\, M]} \frac{l}{\pi} \sin\left(\pi \frac{i-j}{l}\right) \qquad \text{(A.15)}
\end{aligned}
$$

Again, the different effect of N first and N last sample points of T is ignored as $M \gg N$. Now since $l = f_s/B$ and $N = \lfloor f_s/f_{\text{code}} \rfloor$, it can be shown that for $N < i \leqslant M - N$, either term of $\sigma^2_{IQ_2}$ vanishes when B/f_{code} is an even integer. So, $\sigma^2_{IQ_2}$ is approximately zero for even B/f_{code}. Likewise, when B/f_{code} is an odd integer, $\sigma^2_{IQ_2}$ becomes $\dfrac{2MN_0 f_{\text{code}} l}{\pi^2}$. For other positive values of B/f_{code}, $\sigma^2_{IQ_2}$ is given by $\dfrac{MN_0 f_{\text{code}} l}{\pi^2}\left(1 - \cos\dfrac{B\pi}{f_{\text{code}}}\right)$.

In summary, σ_{IQ}^2 is

$$\sigma_{IQ}^2 \approx \begin{cases} \dfrac{MN_0 f_s}{2} & \text{if } \dfrac{B}{f_{\text{code}}} \text{ is even} \\[2ex] \dfrac{MN_0 f_s}{2} - \dfrac{2MN_0 f_{\text{code}} l}{\pi^2} & \text{if } \dfrac{B}{f_{\text{code}}} \text{ is odd} \\[2ex] \dfrac{MN_0 f_s}{2} - \dfrac{MN_0 f_{\text{code}} l}{\pi^2}\left(1 - \cos\dfrac{B\pi}{f_{\text{code}}}\right) & \text{otherwise} \end{cases} \tag{A.16}$$

In the above equation the first term dominates the second term as $\dfrac{B}{f_{\text{code}}}$ or $\dfrac{f_s}{f_{\text{code}}}$ increases.

To estimate σ_{IQ}^2 in real time, an additional correlator pair is used. This correlator pair is shifted from the prompt correlators by multiple chips. The replica signal generated by this correlator pair, when correlated with the incoming signal, results in noise [47]. This is because the signal component correlates to zero due to the large offsets of the replica code from the received code. The variance of the output noise samples is computed and cross-checked.

Appendix B

B.1 Upper Bound on LOS Rate of Change

The LOS vector to satellite i is given by

$$\begin{aligned}&\begin{bmatrix}\dfrac{-(x_{s_i}-x_u)}{\rho_i} & \dfrac{-(y_{s_i}-y_u)}{\rho_i} & \dfrac{-(z_{s_i}-z_u)}{\rho_i}\end{bmatrix}\\ &=\begin{bmatrix}-u_{x_i} & -u_{y_i} & -u_{z_i}\end{bmatrix}\end{aligned}$$

where ρ_i is the pseudorange measurement to satellite i. $[x_{s_i}\ y_{s_i}\ z_{s_i}]^T$ and $[x_u\ y_u\ z_u]^T$ are the x, y and z components of satellite and user positions, respectively, expressed in the ECEF coordinates. u_{x_i}, u_{y_i} and u_{z_i} are the x, y and z components of the LOS vector.

An upper bound for the rate of change of u_{x_i} is derived below. The derivation also holds for the y and z components. The rate of change of u_{x_i} is given by

$$\dot{u}_{x_i} = (\rho_i(\dot{x}_{s_i} - \dot{x}_u) - \dot{\rho}_i(x_{s_i} - x_u))/\rho_i^2 \tag{B.1}$$

where $\dot{\rho}_i$ represents the pseudorange rate for satellite i. $\dot{x}_{s_i}$ and $\dot{x}_u$ denote x-axis velocities of the satellite i and user, respectively. Substituting the maximum values of $\dot{\rho}$, $\dot{x}_s$ and $\dot{x}_u$ and the minimum value of ρ and using the inequality $|x_s - x_u| < \rho$, one can write

$$\begin{aligned}\dot{u}_{x_i} &< (\rho_i(|\dot{x}_{s_i}|_{\max} + |\dot{x}_u|_{\max}) + |\dot{\rho}_i|_{\max}\rho_i)/\rho_i^2\\ &< (|\dot{x}_{s_i}|_{\max} + |\dot{x}_u|_{\max} + |\dot{\rho}_i|_{\max})/\rho_{i\min}\end{aligned} \tag{B.2}$$

Now, substituting $|\dot{x}_s|_{\max} = |\dot{x}_u|_{\max} = 3.874$ km/sec, $|\dot{\rho}|_{\max} = 1.912$ km/sec [69] and $\rho_{\min} = 20193$ km [58], the following upper bound is obtained

$$\dot{u}_{x_{\max}} = 4.78 \times 10^{-4} \quad \text{sec}^{-1} \tag{B.3}$$

Note that the above bound is a very conservative one as it is unlikely that the maxima and minima of all the components of $\dot{u}_x$ will occur simultaneously. Further, the value of $|\dot{x}_u|_{\max}$ assumed here would be too high for most applications.

B.2 Proof that Diagonal Elements of $G_{d,\,l}K_{v,\,l}$ are Inversely Related to Number of Satellites

An arbitrary matrix $P_{3\times 2}$ is defined as $\begin{bmatrix} a_1 & 1 \\ a_2 & 1 \\ a_3 & 1 \end{bmatrix}$. Thus, P represents a geometry with two degrees of freedom and three visible satellites. Now, after some algebraic manipulations, one can write:

$$\begin{aligned} \mathrm{diag}[P(P^TP)^{-1}P^T]_{3\times 1} =& \frac{1}{(a_1-a_2)^2+(a_1-a_3)^2+(a_2-a_3)^2}\times \\ & \begin{bmatrix} (a_1-a_2)^2+(a_1-a_3)^2 \\ (a_2-a_1)^2+(a_2-a_3)^2 \\ (a_3-a_1)^2+(a_3-a_2)^2 \end{bmatrix} \end{aligned} \tag{B.4}$$

If one more satellite is added to the P matrix with the 4th row as $[a_4 \quad 1]$, then:

$$\begin{aligned} \mathrm{diag}[P(P^TP)^{-1}P^T]_{4\times 1} =& \frac{1}{\sum_{i=2}^{4}(a_1-a_i)^2+\sum_{i=3}^{4}(a_2-a_i)^2+(a_3-a_4)^2}\times \\ & \begin{bmatrix} (a_1-a_2)^2+(a_1-a_3)^2+(a_1-a_4)^2 \\ (a_2-a_1)^2+\sum_{i=3}^{4}(a_2-a_i)^2 \\ \sum_{i=1}^{2}(a_3-a_i)^2+(a_3-a_4)^2 \\ \sum_{i=1}^{3}(a_i-a_4)^2 \end{bmatrix} \end{aligned} \tag{B.5}$$

Comparing Equations B.4 and B.5, it is noted that the diagonal elements of $P(P^TP)^{-1}P^T$ are always positive and less than one. Moreover, a close observation of Equation B.5 indicates that, with more satellites in view, both numerators and denominator of Equation B.5 are greater than their counterparts of Equation B.4. So, if k_i/k and $(k_i+c_i)/(k+c)$ are the i^{th} elements of the two column vectors of Equations B.4 and B.5, respectively, then, with simple algebra it can be shown that $k_i/k > (k_i+c_i)/(k+c)$ requires $k_i/k > c_i/c$. Now substituting k, c, k_i and c_i from Equations B.4 and B.5, it can be shown that $k_i/k > c_i/c$ holds for any values of a_1, a_2, a_3 and a_4, meaning the diagonal terms of $P(P^TP)^{-1}P^T$ decrease with the number of satellites. The above proof, after rigorous algebraic manipulations, can also be extended to any dimensions of P to ensure generality.

Appendix C

C.1 Upper Bound on Cumulative Distribution Function of HPE

If ϵ_x and ϵ_y represent the x axis and the y axis position errors, respectively, then the horizontal position error (HPE) is given as HPE $= \sqrt{\epsilon_x^2 + \epsilon_y^2}$. The derivation of the cumulative distribution function (CDF) of the HPE is given below. It is assumed that $\epsilon_x \backsim \mathcal{N}(0,\, \sigma_x^2)$, $\epsilon_y \backsim \mathcal{N}(0,\, \sigma_y^2)$; ϵ_x and ϵ_y are statistically independent; and $\sigma_x^2 > \sigma_y^2$.

It can be shown that the probability density functions (PDF) of $x' = \epsilon_x^2$ and $y' = \epsilon_y^2$ are

$$f_{X'}(x') = \frac{1}{\sqrt{2\pi}\sigma_x}\frac{1}{\sqrt{x'}}\exp^{-x'/(2\sigma_x^2)} \tag{C.1}$$

$$f_{Y'}(y') = \frac{1}{\sqrt{2\pi}\sigma_y}\frac{1}{\sqrt{y'}}\exp^{-y'/(2\sigma_y^2)} \tag{C.2}$$

Considering the one-to-one relationship between the HPE^2 and the HPE, one can write

$$\begin{aligned} P(HPE \leqslant \sqrt{k}) &= P(HPE^2 \leqslant k) \\ &= P(x' + y' \leqslant k) \\ &= \iint\limits_{x'+y' \leqslant k} f_{X'Y'}(x',\, y')dx'dy' \end{aligned} \tag{C.3}$$

where $f_{X'Y'}(x',\, y')$ is the joint PDF of x' and y'. Since x' and y' are independent, $f_{X'Y'}(x',\, y') = f_{X'}(x')f_{Y'}(y')$. Hence, the above equation becomes

$$P(x' + y' \leqslant k) = \iint\limits_{x'+y' \leqslant k} f_{X'}(x')f_{Y'}(y')dx'dy'$$

$$= \int_0^k f_{X'}(x')dx' \int_0^{k-x'} f_{Y'}(y')dy' \tag{C.4}$$

Substituting Equation C.2 in the above equation and replacing y' with $y'' = y'/\sigma_y^2$, the following equation is obtained

$$P(x' + y' \leqslant k) = \int_0^k f_{X'}(x')dx' \int_0^{(k-x')/\sigma_y^2} \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{y''}} \exp^{-y''/2} dy''$$

Note that, after the change of variable from y' to y'', the second integral reduces to the CDF of a chi-square distribution with single DOF which is $P(x \leqslant X) = \mathrm{erf}(\sqrt{X/2})$, where erf is the error function and x is chi squared distributed with single DOF. Substituting this, one gets

$$P(x' + y' \leqslant k) = \int_0^k f_{X'}(x')\mathrm{erf}\left(\sqrt{\frac{k - x'}{2\sigma_y^2}}\right) dx' \tag{C.5}$$

A close observation of Equation C.1 shows that $f_{X'}(x')$ is positive and has an asymptotic discontinuity at $x' = 0$. Therefore, it is required to examine whether the integral of Equation C.5 exists over the integration interval $(0,\ k]$ or not. For this purpose the definition of the improper Riemann integral is used, as given below

$$\int_0^k f_{X'}(x')\mathrm{erf}\left(\sqrt{\frac{k - x'}{2\sigma_y^2}}\right) dx'$$
$$= \lim_{\epsilon \to 0^+} \int_\epsilon^k f_{X'}(x')\mathrm{erf}\left(\sqrt{\frac{k - x'}{2\sigma_y^2}}\right) dx'$$

Note that the error function $\mathrm{erf}(x)$ takes on values between 0 and 1 i.e. $0 \leqslant \mathrm{erf}(x) \leqslant 1$ for any $x \in \mathbf{R}$. This implies that the non-negative function $I(x') = f_{X'}(x')\mathrm{erf}\left(\sqrt{\frac{k - x'}{2\sigma_y^2}}\right)$ is less than $f_{X'}(x')$. Now applying the property of monotonicity to the integration of $I(x')$ and $f_{X'}(x')$ which are Reimann integrable in the range of $0 < \epsilon < k$, one can write

$$0 < \int_\epsilon^k I(x')dx' < \int_\epsilon^k f_{X'}(x')dx'$$

Substituting for $f_{X'}(x')$

$$< \int_\epsilon^k \frac{1}{\sqrt{2\pi}\sigma_x} \frac{1}{\sqrt{x'}} \exp^{-x'/(2\sigma_x^2)} dx'$$

Using the substitution $x'' = x'/\sigma_x^2$

$$< \int_{\epsilon/\sigma_x^2}^{k/\sigma_x^2} \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{x''}} \exp^{-x''/2} dx'' \tag{C.6}$$

As ϵ tends to zero from above, the RHS of the above inequality becomes the CDF of a chi-square distribution with single DOF and exists. Therefore, for $0 < \epsilon < k$

$$\begin{aligned} &\int_{\epsilon/\sigma_x^2}^{k/\sigma_x^2} \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{x''}} \exp^{-x''/2} dx'' \\ &< \lim_{\epsilon \to 0^+} \int_{\epsilon/\sigma_x^2}^{k/\sigma_x^2} \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{x''}} \exp^{-x''/2} dx'' \\ &< \operatorname{erf}\left(\sqrt{\frac{k}{2\sigma_x^2}}\right) \end{aligned} \tag{C.7}$$

Combining Equations C.6 and C.7 gives

$$0 < \int_{\epsilon}^{k} I(x')dx' < \operatorname{erf}\left(\sqrt{\frac{k}{2\sigma_x^2}}\right) \tag{C.8}$$

Considering $\int_{\epsilon}^{k} I(x')dx' = F(k)$ - $F(\epsilon)$, $dF(x')/dx' = I(x') > 0$ and the above inequality, it can be concluded that $G(\epsilon) = F(k)$ - $F(\epsilon)$ is continuous, decreasing and bounded on $(0, k)$ for $0 < \epsilon < k$, where k is fixed. If M is its least upper bound on $(0, k)$, then arbitrarily small δ_1, $\delta_2 > 0$ can always be found such that for $0 < \epsilon < \delta_1$, $M - \delta_2 < G(\delta_1) < G(\epsilon) < M + \delta_2$. Or, for $\epsilon < \delta_1$, $(M - G(\epsilon)) < \delta_2$. As this inequality is true for any infinitesimally small δ_1 and δ_2, one can write $\lim_{\epsilon \to 0^+} G(\epsilon) = \lim_{\epsilon \to 0^+}(F(k) - F(\epsilon))$ $= M$. Hence, $\lim_{\epsilon \to 0^+} \int_{\epsilon}^{k} I(x')dx'$ i.e. the integral of Equation C.5 exists.

Since both integrals of Equation C.6 exist as $\epsilon \to 0^+$, the inequality is preserved under the limit operation. Therefore, from Equations C.6 and C.7 the following inequality is obtained

$$\begin{aligned} \lim_{\epsilon \to 0^+} \int_{\epsilon}^{k} I(x')dx' &< \lim_{\epsilon \to 0^+} \int_{\epsilon/\sigma_x^2}^{k/\sigma_x^2} \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{x''}} \exp^{-x''/2} dx'' \\ P(x' + y' \leqslant k) &< \operatorname{erf}\left(\sqrt{\frac{k}{2\sigma_x^2}}\right) \end{aligned} \tag{C.9}$$

Next it will be investigated if a Gaussian distribution with zero mean and variance $(\sigma_x^2 + \sigma_y^2)$ will bound the tail of the distribution of the HPE. In order to do this, first a lower bound of Equation C.5 is found. Note that $I(x') > f_{X'}(x')\operatorname{erf}(\sqrt{(k - x')/(2\sigma_x^2)})$ as $\sigma_x^2 > \sigma_y^2$. Using the property of monotonicity of integration yields

$$P(x' + y' \leqslant k) > \int_{0}^{k} f_{X'}(x')\operatorname{erf}\left(\sqrt{\frac{k - x'}{2\sigma_x^2}}\right) dx' \tag{C.10}$$

Substituting for $f_{X'}(x')$ from Equation C.1 and replacing x'/σ_x^2 with x', it can be shown that the RHS of the above inequality becomes the CDF of a chi-square distribution with DOF 2. Thus, combining Equations C.9 and C.10 results

$$1 - \exp^{-k/(2\sigma_x^2)} < P(x' + y' \leqslant k) < \mathrm{erf}\left(\sqrt{\frac{k}{2\sigma_x^2}}\right) \tag{C.11}$$

So, the probablity that the HPE is larger than $\sqrt{k}$ is bounded as follows

$$1 - \mathrm{erf}\left(\sqrt{\frac{k}{2\sigma_x^2}}\right) < P(\sqrt{x' + y'} > \sqrt{k}) < \exp^{-k/(2\sigma_x^2)} \tag{C.12}$$

Now, the probability that a Gaussian random variable z with zero mean and variance $(\sigma_x^2 + \sigma_y^2)$ exceeds $\sqrt{k}$ in magnitude is given by

$$P(|z| > \sqrt{k}) = 1 - \mathrm{erf}\left(\sqrt{\frac{k}{2(\sigma_x^2 + \sigma_y^2)}}\right) \tag{C.13}$$

It should be noted that the lower bound $L_B(k)$ of Equation C.12 is always less than $G(k)$ $(= P(|z| > \sqrt{k}))$ of Equation C.13 while the upper bound $U_B(k)$ is equal to $G(k)$ at $k = 0$. It can also be shown that, for some M, $U_B(k)$ is larger than $G(k)$ for $0 < k \leqslant M^2$ and then crosses over to the other side of $G(k)$ for $k > M^2$. The value of M is determined by the absolute and relative magnitudes of σ_x and σ_y. The larger is the ratio of σ_x to σ_y, the greater is M. Therefore, when σ_x and σ_y are close to each other (e.g. $\sigma_x < 2.5\sigma_y$), $G(k)$ will always bound $U_B(k)$ (and $H(k)$ $(= P(\sqrt{x' + y'} > \sqrt{k}))$) after some k. On the other hand, when σ_x exceeds σ_y by a certain margin determined by their magnitudes, $G(k)$ may not bound $U_B(k)$ for any k of practical interest. However, $U_B(k)$ becomes a conservative upper bound of $H(k)$ as $\sigma_x \gg \sigma_y$ and both $G(k)$ and $H(k)$ then approach $L_B(k)$ from above. Thus, it might happen that $G(k)$ would bound $H(k)$ after some k. To verify this, $G(k)$ and $H(k)$ are simulated for different magnitudes of σ_x and σ_y and it is found that $G(k)$ always bounded $H(k)$ after some k, as shown in Figure C.1.

It is important to note that the above derivation assumes that ϵ_x and ϵ_y are uncorrelated. If they are correlated, then they can be uncorrelated by applying a whitening transformation. The transformation matrix is the eigen vectors of the position error covariance matrix in the horizontal plane. Since ϵ_x and ϵ_y are assumed to

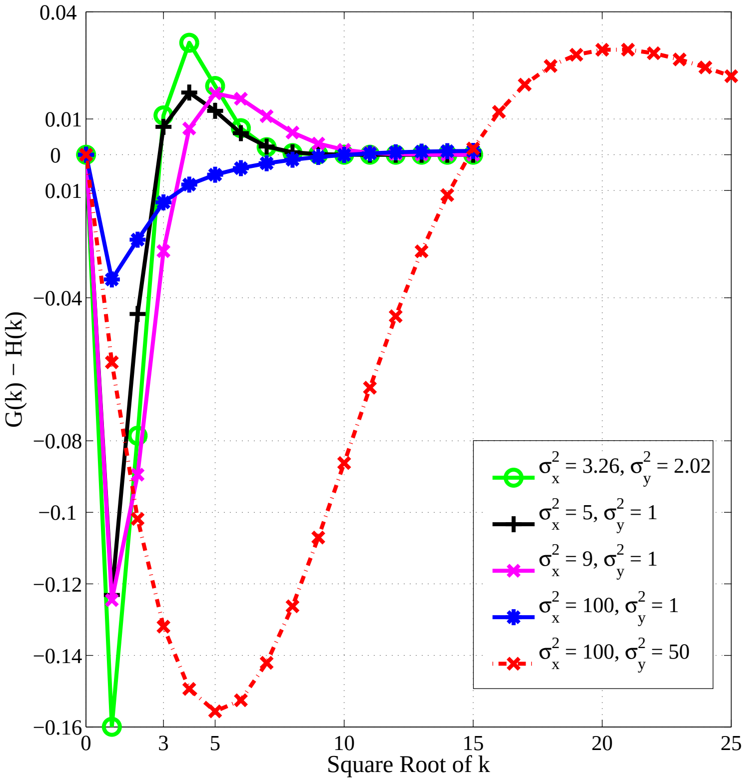


Figure C.1: Difference between $G(k)$ and $H(k)$ for Different Values of k

be Gaussian distributed, the uncorrelated variables ϵ'_x and ϵ'_y can be related to ϵ_x and ϵ_y through an angular rotation θ about the origin, as given below

$$\epsilon_x = \epsilon'_x \cos\theta + \epsilon'_y \sin\theta$$
$$\epsilon_y = \epsilon'_x \sin\theta - \epsilon'_y \cos\theta$$

Using the above transformation, the following relationship is obtained: $x' + y' = \epsilon_x^2 + \epsilon_y^2 = \epsilon_x'^2 + \epsilon_y'^2$. Thus, $P(x' + y' \leq k) = P(\epsilon_x'^2 + \epsilon_y'^2 \leq k)$. Then Equation C.12 can be derived using the same approach as before for the uncorrelated variables ϵ'_x and ϵ'_y. The variances of ϵ'_x and ϵ'_y are the eigen values of the position error covariance matrix and the means are zero.

C.2 Proof that Fault Detection Test Statistic and EKF Position Error are Statistically Independent

C.2.1 Snapshot Fault Detection Method

The statistical independence between current position error and current snapshot test statistic is proved by showing that orthogonal components of the measurement noise vector contribute to the test statistic and position error. This derivation follows from [89]. Recalling the definition of $WSSE$ or z^2 from Chapter 5, one can write

$$\begin{aligned} WSSE_k = z_k^2 = (\Delta\underline{\rho}_k - H_k(H_k^T W_k^{-1} H_k)^{-1} H_k^T W_k^{-1} \Delta\underline{\rho}_k)^T \times \\ W_k^{-1}(\Delta\underline{\rho}_k - H_k(H_k^T W_k^{-1} H_k)^{-1} H_k^T W_k^{-1} \Delta\underline{\rho}_k) \end{aligned} \tag{C.14}$$

where W_k is the pseudorange measurement noise covariance. H_k is the pseudorange measurement model matrix and $\Delta\underline{\rho}_k$ is the predicted pseudorange residual vector. Decomposing W_k^{-1} into $W_k^{-1/2} W_k^{-1/2}$

$$\begin{aligned} z_k^2 = (W^{-1/2}\Delta\underline{\rho}_k - W_k^{-1/2} H_k(H_k^T W_k^{-1/2} W_k^{-1/2} H_k)^{-1} H_k^T W_k^{-1/2} W_k^{-1/2} \Delta\underline{\rho}_k)^T \times \\ (W_k^{-1/2}\Delta\underline{\rho}_k - W_k^{-1/2} H_k(H_k^T W_k^{-1/2} W_k^{-1/2} H_k)^{-1} H_k^T W_k^{-1/2} W_k^{-1/2} \Delta\underline{\rho}_k) \end{aligned} \tag{C.15}$$

Substituting $W_k^{-1/2}\Delta\underline{\rho}_k = \Delta\underline{\rho}_k^*$ and $W_k^{-1/2} H_k = H_k^*$ gives

$$\begin{aligned} z_k^2 = (\Delta\underline{\rho}_k^* - H_k^*(H_k^{*T} H_k^*)^{-1} H_k^{*T} \Delta\underline{\rho}_k^*)^T \times \\ (\Delta\underline{\rho}_k^* - H_k^*(H_k^{*T} H_k^*)^{-1} H_k^{*T} \Delta\underline{\rho}_k^*) \end{aligned} \tag{C.16}$$

Next, decomposing $\underline{\epsilon}_k^* = W_k^{-1/2}\underline{\epsilon}_k$ into two orthogonal components, it will be shown that one component contributes to the test statistic and the other to the EKF position error.

$$\underline{\epsilon}_k^* = \underline{\epsilon}_{\Re_{H_k^*}} + \underline{\epsilon}_{N_{H_k^{*T}}} \tag{C.17}$$

where the first term on the right hand side is in the column space (or range) of H_k^* and the second term is in the null space of H_k^{*T}. Since the measurement noise is injected through $\Delta\underline{\rho}_k$, $\Delta\underline{\rho}_k^*$ is now replaced with $\underline{\epsilon}_k^*$. Thus, the first term in the parentheses of Equation C.16 becomes

$$\underline{r}_k = \Delta\underline{\rho^*}_k - H_k^*(H_k^{*T} H_k^*)^{-1} H_k^{*T} \Delta\underline{\rho}_k^*$$

$$=\underline{\epsilon}_k^* - H_k^*(H_k^{*T}H_k^*)^{-1}H_k^{*T}\underline{\epsilon}_k^* \tag{C.18}$$

Substituting for $\underline{\epsilon}_k^*$ gives

$$\begin{aligned}\underline{r}_k &= \underline{\epsilon}_k^* - H_k^*(H_k^{*T}H_k^*)^{-1}H_k^{*T}\underline{\epsilon}_k^* \\ &=(\underline{\epsilon}_{\Re_{H_k^*}} + \underline{\epsilon}_{N_{H_k^{*T}}}) - H_k^*(H_k^{*T}H_k^*)^{-1}H_k^{*T}(\underline{\epsilon}_{\Re_{H_k^*}} + \underline{\epsilon}_{N_{H_k^{*T}}})\end{aligned} \tag{C.19}$$

Replacing $\underline{\epsilon}_{\Re_{H_k^*}}$ with $H_k^*\underline{y}$ as it lies in the range of H_k^*, one obtains

$$\begin{aligned}\underline{r}_k &= \underline{\epsilon}_k^* - H_k^*(H_k^{*T}H_k^*)^{-1}H_k^{*T}\underline{\epsilon}_k^* \\ &=(H_k^*\underline{y} + \underline{\epsilon}_{N_{H_k^{*T}}}) - H_k^*(H_k^{*T}H_k^*)^{-1}H_k^{*T}(H_k^*\underline{y} + \underline{\epsilon}_{N_{H_k^{*T}}}) \\ &=\underline{\epsilon}_{N_{H_k^{*T}}}\end{aligned} \tag{C.20}$$

Next, the error $\underline{\epsilon}_{X_k}$ in the *a posteriori* state estimate can be written as [86]

$$\underline{\epsilon}_{X_k} = (I - K_kC_k)F\underline{\epsilon}_{X_{k-1}} + (I - K_kC_k)\underline{w}_{k-1} - K_k\begin{bmatrix}\underline{\epsilon}_k \\ \underline{\xi}_k\end{bmatrix} \tag{C.21}$$

where K_k is the Kalman gain; F is the state transition matrix; C_k is the EKF measurement model matrix and $\underline{w}_{k-1}$ is the process noise vector. $\underline{\xi}_k$ is the noise in the pseudorange rate measurements vector. It is important to note that $\underline{\epsilon}_k$ is independent of $\underline{\epsilon}_{X_{k-1}}$, $\underline{\xi}_k$ and $\underline{w}_{k-1}$. The contribution of the measurement noise to the state estimate error is given by

$$\begin{aligned}\underline{\eta}_{X_k} =& K_k\begin{bmatrix}\underline{\epsilon}_k \\ \underline{\xi}_k\end{bmatrix} \\ =& \Sigma_k^- C_k^T\left(C_k\Sigma_k^-C_k^T + \begin{bmatrix}W_k & 0 \\ 0 & W_k'\end{bmatrix}\right)^{-1}\begin{bmatrix}\underline{\epsilon}_k \\ \underline{\xi}_k\end{bmatrix}\end{aligned}$$

where Σ_k^- is the predicted error covariance matrix of the EKF; W_k' is the pseudorange rate measurement noise covariance. Replacing $\begin{bmatrix}W_k & 0 \\ 0 & W_k'\end{bmatrix}$ with W_k'' and $\begin{bmatrix}\underline{\epsilon}_k \\ \underline{\xi}_k\end{bmatrix}$ with $\underline{\zeta}_k$

$$\begin{aligned}\underline{\eta}_{X_k} =& \Sigma_k^-C_k^T(C_k\Sigma_k^-C_k^T + W_k'')^{-1}\underline{\zeta}_k \\ =& \Sigma_k^-C_k^T(W_k'')^{-1/2}(W_k'')^{1/2}(C_k\Sigma_k^-C_k^T + W_k'')^{-1}(W_k'')^{1/2}(W_k'')^{-1/2}\underline{\zeta}_k\end{aligned}$$

Replacing $(W_k'')^{-1/2}C_k$ with C_k^* and $(W_k'')^{-1/2}\underline{\zeta}_k$ with $\underline{\zeta}_k^*$, one gets

$$\underline{\eta}_{X_k} = \Sigma_k^-C_k^{*T}(C_k^*\Sigma_k^-C_k^{*T} + I)^{-1}\underline{\zeta}_k^* \tag{C.22}$$

With some algebra $\underline{\eta}_{X_k}$ becomes

$$\begin{aligned}
\underline{\eta}_{X_k} =& \Sigma_k^- C_k^{*T}(C_k^* \Sigma_k^- C_k^{*T} + I)^{-1} \underline{\zeta}_k^* \\
=& \Sigma_k^- (C_k^{*T} - C_k^{*T}(C_k^* \Sigma_k^- C_k^{*T}) + C_k^{*T}(C_k^* \Sigma_k^- C_k^{*T})(C_k^* \Sigma_k^- C_k^{*T}) - \ldots) \underline{\zeta}_k^* \\
=& \Sigma_k^- (C_k^{*T} - (C_k^{*T} C_k^* \Sigma_k^-) C_k^{*T} + (C_k^{*T} C_k^* \Sigma_k^-)(C_k^{*T} C_k^* \Sigma_k^-) C_k^{*T} - \ldots) \underline{\zeta}_k^* \\
=& \Sigma_k^- (C_k^{*T} C_k^* \Sigma_k^- + I)^{-1} C_k^{*T} \underline{\zeta}_k^* \\
=& \Sigma_k^- (C_k^{*T} C_k^* \Sigma_k^- + I)^{-1} C_k^{*T} \begin{bmatrix} \underline{\epsilon}_k^* \\ \underline{\xi}_k^* \end{bmatrix}
\end{aligned} \tag{C.23}$$

C_k and H_k are given by

$$\begin{aligned}
C_k &= [C_{x,\,k} \quad C_{y,\,k} \quad C_{z,\,k} \quad C_{clk}] \\
H_k &= [H_{x,\,k} \quad H_{y,\,k} \quad H_{z,\,k} \quad [1 \ldots 1]_{1\times j}^T]
\end{aligned}$$

where

$$C_{m,\,k} = \begin{bmatrix} -\tilde{u}_{m_{1,\,k}} & 0 \\ \vdots & \vdots \\ -\tilde{u}_{m_{j,\,k}} & 0 \\ 0 & -\tilde{u}_{m_{1,\,k}} \\ \vdots & \vdots \\ 0 & -\tilde{u}_{m_{j,\,k}} \end{bmatrix}_{2j\times 2}, \quad H_{m,\,k} = \begin{bmatrix} -\tilde{u}_{m_{1,\,k}} \\ \vdots \\ -\tilde{u}_{m_{j,\,k}} \end{bmatrix}_{j\times 1}, \quad C_{clk} = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}_{2j\times 2}$$

$m = x,\, y,\, z$

where $\tilde{u}_{m_{i,\,k}} = m$ component of the LOS vector from satellite i to user at time t_k. It is defined in Chapter 3 while discussing the vector loop navigation filter and in Appendix B. j is the number of visible satellites. Thus, H_k^* and C_k^* become

$$\begin{aligned}
H_k^* &= W_k^{-1/2} H_k = \begin{bmatrix} H_{x,\,k}^* & H_{y,\,k}^* & H_{z,\,k}^* & W_k^{-1/2} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{j\times 1} \end{bmatrix} \\
C_k^* &= \begin{bmatrix} W_k^{-1/2} & [0]_{j\times j} \\ [0]_{j\times j} & (W_k')^{-1/2} \end{bmatrix} C_k
\end{aligned}$$

$$= \begin{bmatrix} I_{j\times j} & [0]_{j\times j} \\ [0]_{j\times j} & (W_k')^{-1/2} \end{bmatrix} \times \begin{bmatrix} H^*_{x,\,k} & \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & H^*_{y,\,k} & \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & H^*_{z,\,k} & \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & W_k^{-1/2} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & H_{x,\,k} & \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & H_{y,\,k} & \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & H_{z,\,k} & \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \end{bmatrix} \tag{C.24}$$

Using the above mathematical expression of C_k^*, it is evident that only the $\underline{\epsilon}_{\Re_{H_k^*}}$ component of $\underline{\epsilon}_k^*$ will contribute to Equation C.23. Hence, $\underline{\eta}_{X_k}$ is given by

$$\underline{\eta}_{X_k} = \Sigma_k^-(C_k^{*T} C_k^* \Sigma_k^- + I)^{-1} C_k^{*T} \begin{bmatrix} \underline{\epsilon}_{\Re_{H_k^*}} \\ \underline{\xi}_k^* \end{bmatrix} \tag{C.25}$$

As $\underline{\epsilon}_{\Re_{H_k^*}}$ and $\underline{\epsilon}_{N_{H_k^{*T}}}$ are orthogonal components of $\underline{\epsilon}_k^*$ and $\underline{\xi}_k^*$ and $\underline{\epsilon}_k^*$ are statistically independent, $\underline{\eta}_{X_k}$ and $\underline{r}_k$ are uncorrelated. Since $\underline{\eta}_{X_k}$ and $\underline{r}_k$ are assumed to be Gaussian-distributed, they are also statistically independent. Thus, the current snapshot test statistic and position error, being corrupted by $\underline{r}_k$ and $\underline{\eta}_{X_k}$, respectively, are statistically independent.

Though not considered in the navigation filter implementation of this thesis, H_k and C_k will, in practice, have additional columns to account for coefficients corresponding to estimated state errors which include ephemeris error, residual ionospheric and troposhperic delays and additional rows corresponding to measurements containing *a priori* information about the state errors. However, since both H_k and C_k will be expanded the same way, the statistical independence between $\underline{r}_k$ and $\underline{\eta}_{X_k}$ will be retained. Moreover, it is envisioned that the future implementation of vector loop integrity monitoring algorithm will have a test statistic that may include the velocity states. For this case, using the same argument as above it can be shown that orthogonal components of $\underline{\xi}_k^*$ will contribute to $\underline{r}_k$ and $\underline{\eta}_{X_k}$.

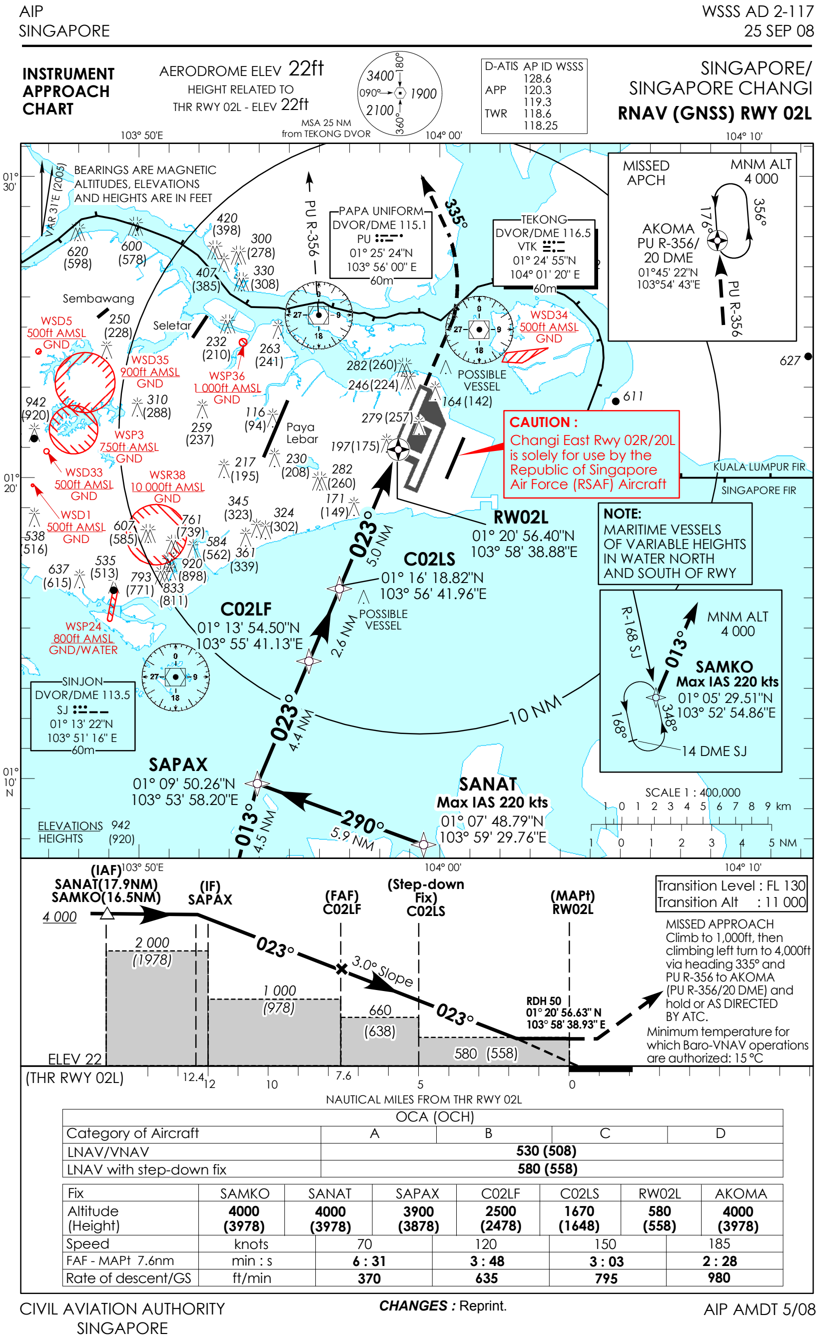
C.2.2 Fault Detection Method Using Current and Past Snapshot Test Statistics

In this fault detection method the test statistic is formed by adding a number of past snapshot test statistics to the current snapshot test statistic. As it has been proved above that current snapshot statistic and position error are independent, it is sufficient to show that current position error is independent of a past snapshot test statistic. For this purpose, $\underline{\eta}_{X_k}$ is propagated to the next time epoch t_{k+1} using Equations C.21 and C.25 as follows:

$$\underline{\eta}_{X_{k+1}} = (I - K_{k+1}C_{k+1})F\Sigma_k^-(C_k^{*T}C_k^*\Sigma_k^- + I)^{-1}C_k^{*T}\begin{bmatrix}\underline{\epsilon}_{\Re_{H_k^*}} \\ \underline{\xi}_k^*\end{bmatrix} \tag{C.26}$$

Again using the orthogonality of $\underline{\epsilon}_{N_{H_k^{*T}}}$ and $\underline{\epsilon}_{\Re_{H_k^*}}$, it is apparent that $\underline{\eta}_{X_{k+1}}$ and $\underline{r}_k$ are uncorrelated. Hence, past snapshot test statistics and current position error are statistically independent.

Appendix D



AIP
SINGAPORE

WSSS AD 2-117
25 SEP 08

INSTRUMENT APPROACH CHART

AERODROME ELEV 22ft
HEIGHT RELATED TO
THR RWY 02L - ELEV 22ft

MSA 25 NM from TEKONG DVOR: 3400, 1900, 2100

D-ATIS	AP ID WSSS 128.6
APP	120.3 119.3
TWR	118.6 118.25

SINGAPORE/
SINGAPORE CHANGI
RNAV (GNSS) RWY 02L

BEARINGS ARE MAGNETIC
ALTITUDES, ELEVATIONS
AND HEIGHTS ARE IN FEET

MISSED APCH
MNM ALT 4 000
AKOMA
PU R-356/
20 DME
01°45' 22"N
103°54' 43"E

CAUTION :
Changi East Rwy 02R/20L is solely for use by the Republic of Singapore Air Force (RSAF) Aircraft

NOTE:
MARITIME VESSELS OF VARIABLE HEIGHTS IN WATER NORTH AND SOUTH OF RWY

Transition Level : FL 130
Transition Alt : 11 000

MISSED APPROACH
Climb to 1,000ft, then climbing left turn to 4,000ft via heading 335° and PU R-356 to AKOMA (PU R-356/20 DME) and hold or AS DIRECTED BY ATC.
Minimum temperature for which Baro-VNAV operations are authorized: 15 °C

OCA (OCH)				
Category of Aircraft	A	B	C	D
LNAV/VNAV	**530 (508)**			
LNAV with step-down fix	**580 (558)**			

Fix	SAMKO	SANAT	SAPAX	C02LF	C02LS	RW02L	AKOMA
Altitude (Height)	**4000 (3978)**	**4000 (3978)**	**3900 (3878)**	**2500 (2478)**	**1670 (1648)**	**580 (558)**	**4000 (3978)**

Speed	knots	70	120	150	185
FAF - MAPt 7.6nm	min : s	**6 : 31**	**3 : 48**	**3 : 03**	**2 : 28**
Rate of descent/GS	ft/min	**370**	**635**	**795**	**980**

CIVIL AVIATION AUTHORITY
SINGAPORE

CHANGES : Reprint.

AIP AMDT 5/08

Figure D.1: Approach Plate of Runway 02L, Changi International Airport, Singapore