VECTOR TRACKING LOOP DESIGN FOR GPS RECEIVERS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

DENİZ ÜZEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2016

Approval of the Thesis

VECTOR TRACKING LOOP DESIGN FOR GPS RECEIVERS

submitted by **DENİZ ÜZEL** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____

Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan _____

Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Buyurman Baykal _____

Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. Tolga Çiloğlu _____

Electrical and Electronics Engineering Dept., METU

Prof. Dr. Buyurman Baykal _____

Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Umut Orguner _____

Electrical and Electronics Engineering Dept., METU

Assist. Prof. Sevinç Figen Öktem _____

Electrical and Electronics Engineering Dept., METU

Assist. Prof. Sevgi Zübeyde Gürbüz _____

Electrical and Electronics Engineering Dept., TOBB ETU

Date: 04.02.2016

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: DENİZ ÜZEL

Signature :

ABSTRACT

VECTOR TRACKING LOOP DESIGN FOR GPS RECEIVERS

Üzel, Deniz
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Buyurman Baykal

February 2016, 110 pages

This study describes the design of a modern GPS receiver architecture based on vector tracking loops. Since the traditional tracking loops process the signals independently, there is no information exchange between channels. Due to that fact, aiding of weaker signals in the presence of relatively strong signals is impossible. On the other hand, vector tracking loops simultaneously process the signals from all visible channels. Therefore, they are able to perform better than the traditional tracking loops in degraded signal environments where carrier to noise ratio is significantly low.

In this thesis, a review of GPS receivers and satellite constellation is firstly presented. Then, the implementation details of traditional tracking loops and Vector Delay/Frequency Lock Loop (VDFLL) algorithm, which is a common type of the vector tracking architectures, are discussed. In addition, a modified version of VDFLL, which has a navigation filter with less number of states, is proposed. The mentioned algorithms are implemented in MATLAB environment. The performances of these algorithms are compared by using a Spirent GPS signal simulator. These performance comparison results indicate that the vector tracking algorithms have improved signal tracking capabilities in low carrier to noise ratio environments. Besides the superiority of the vector tracking algorithms, it is also shown that the modified version of VDFLL algorithm may be an alternative to original VDFLL with the advantage of lower computational load and almost similar performance in degraded signal environments. These algorithms can be used to track GPS signals in challenging GPS signal environments such as urban canyons and indoor areas.

Key words: GPS, Vector Tracking, Scalar Tracking, VDFLL, Extended Kalman Filter

ÖZ

GPS ALICILARI İÇİN VEKTÖR TAKİP DÖGÜSÜ TASARIMI

Üzel, Deniz
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Buyurman Baykal

Şubat 2016, 110 sayfa

Bu çalışma vektör takip döngüsü tabanlı modern GPS alıcısı tasarımının detaylarını kapsamaktadır. Geleneksel takip döngüleri yakalanan sinyalleri birbirlerinden bağımsız şekilde işledikleri için sinyaller arası bilgi aktarımı ve zayıf sinyallerin görece daha güçlü sinyaller tarafından desteklenmesi gibi durumlar mümkün olmamaktadır. Diğer taraftan; vektör takip döngüleri yakalanan tüm GPS uydu sinyallerini aynı anda işlerler. Bu durum GPS sinyallerinin herhangi bir sebepten düşük güce sahip olduğu bölgelerde vektör takip döngülerinin geleneksel takip döngülerine göre daha iyi performans gösterebilmelerine olanak sağlamaktadır.

Bu tezde, ilk olarak, GPS alıcılarının ve GPS uydu sisteminin temel özellikleri anlatılmıştır. Geleneksel ve vektör takip döngülerinin uygulanmalarına yönelik bu tez çalışmasında yapılanlar detaylı şekilde anlatılmıştır. Ayrıca, VDFLL tabanlı fakat VDFLL' e göre daha az işlem yüküne sahip olan yeni bir vektör takip döngü algoritması sunulmuştur. Belirtilen sinyal takip döngüsü algoritmaları MATLAB üzerinde uygulanmıştır. Bu algoritmaların performansları çeşitli yönlerden geleneksel takip döngü performansları ile Spirent GPS sinyal simülatörü kullanılarak karşılaştırılmıştır. Elde edilen sonuçlar ile vektör takip döngüsü algoritmalarının düşük taşıyıcı frekans/gürültü oranına sahip bölgelerde geleneksel takip döngülerinden daha iyi sinyal takibi yapabildikleri doğrulanmıştır. Bunun yanında, sunulan yeni takip döngüsünün VDFLL algoritmasına alternatif olabileceği gösterilmiştir. Bu algoritmalar kapalı alanlar ve kanyonlar gibi GPS sinyalleri açısından zorlayıcı olan bölgelerde GPS sinyal takibi için kullanılabilir.

Anahtar Kelimeler: GPS, Vektör Takip Döngüsü, Geleneksel Takip Döngüsü, VDFLL, Kalman Filtre

Tatlım' a…

ACKNOWLEDGEMENTS

Firstly, I would like to acknowledge the guidance of my supervising advisor Prof. Dr. Buyurman Baykal.

I am thankful to my colleagues, especially Berk Osunluk, Tahsin Zubaroğlu and Eren Kahraman for their friendship, moral support and suggestions throughout this study. I am even more thankful to Yiğiter Yüksel for his encouragement, technical guidance and endless support.

I am also thankful to Aselsan Inc. for the resources and facilities that were made available to me during my studies.

I would like to thank to my family, for their continuous support, not only for this thesis but also for my life.

Lastly, but certainly not least, I would like to thank to my love, Öznur Cansu Seçkin, for her continuous support, patience and endless help. This thesis would not be done without her.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
CDMA	Code Division Multiple Access
C/A	Coarse Acquisition
C/No	Carrier to Noise Ratio
dB	Decibel
DFT	Discrete Fourier Transform
DLL	Delay Lock Loop
ECEF	Earth Centered Earth Fixed
EKF	Extended Kalman Filter
FFT	Fast Fourier Transform
FLL	Frequency Lock Loop
GLONASS	Russian Global Orbiting Navigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HOW	Handover Word
Hz	Hertz
IF	Intermediate Frequency
LHCP	Left Hand Circularly Polarized
LOS	Line of Sight

NCO	Numerically Controlled Oscillator
P	Precise
PLL	Phase Lock Loop
PRM	Power Ratio Method
PRN	Pseudo-Random Noise
RHCP	Right Hand Circularly Polarized
SLL	Scalar Lock Loop
SNR	Signal to Noise Ratio
TLM	Telemetry
VDFLL	Vector Delay/Frequency Lock Loop
VDFLL-M	Vector Delay/Frequency Lock Loop - Modified
VDLL	Vector Delay Lock Loop
VFLL	Vector Frequency Lock Loop
VPLL	Vector Phase Lock Loop
VSM	Variance Summing Method
VTL	Vector Tracking Loop
WGN	White Gaussian Noise

CHAPTER 1

INTRODUCTION

From the very beginning of mankind, navigation is one of the most crucial issues for human beings to maintain their lives. Therefore, even the primitive ones had tried to develop many navigation methods to make the life easier. After the invention of the radio frequency signal, it is discovered that the radio signals can be used for navigation purposes. The idea of satellite based navigation has been supported by United States Department of Defense and has come to life with the name Global Positioning System (GPS) in 1995 [1]. There are other global navigation satellite systems (GNSS) currently in operation such as the Russian Global Orbiting Navigation Satellite System (GLONASS) but, GPS is the most popular one for the users worldwide.

GPS has been designed and developed to provide the position, velocity and time information to any user at any place on earth under any weather conditions continuously (7/24) with acceptable accuracy [2].

Although it has been designed and developed for military purposes, it is being used in many civil applications and gaining more and more popularity in the civil navigation. Increase in the usage of GPS for both military and civil areas has helped to increase the reliability and applicability of it. According to the market researches, commercial sector has the biggest share of the revenue of GPS [3]. Therefore, affordability became as important as the accuracy for satellite navigation systems. By the help of developments of processing units, software defined GPS receivers have

started to be used lately. This improvement has led to low cost receivers and the affordability problem has been solved permanently.

With the increasing popularity and usage area, reliability has become the most questioned issue of GPS. Although, it seems to provide satisfactory navigation solution in most of the applications, the performance of GPS severely degrades in weak signal environments, such as indoors, tunnels and urban canyons.

In a traditional receiver, all available satellites are processed independently without any cooperation [11]. Typically, least squares method is used to determine the receiver position. Basics of the least squares method are the linearized positioning equations that are used to approximate the true position [11]. The traditional tracking loops operate well under high carrier-to-noise ratio (C/N_o) and low receiver dynamics. However, they cannot give acceptable performance in weak signal environments [5]. This is because of no cooperation between channels, which means lock on any satellite does not support tracking of any other satellite. The traditional tracking loop architectures are relatively easy to implement and robust in the sense that error in one channel does not corrupt the other ones. But, in the worst case, once the signal is lost or too weak, tracking channels are not able to track again and require running the acquisition stage to relock the satellite [4].

There are other methods that have many advantages with respect to the traditional tracking loops. Vector Tracking Loop (VTL) is the most effective one which combines tracking stage, positioning stage and movement of the receiver in a Kalman filter using the basic idea that every acquired channel's code phase (or pseudorange) and carrier frequency (pseudorange rate) are related to the user position and velocity in contrast to the traditional loops. In fact, in a traditional receiver, navigation solutions are calculated using pseudoranges and pseudorange rates. On the contrary, by projecting the relative position and velocity between the user and any acquired satellite, code phases and carrier frequencies can also be calculated. That means all acquired signals are correlated with each other through the same user position and velocity [5].

1.1 Scope of the Thesis

This thesis is devoted to design and implement a Vector Tracking Loop algorithm for software defined GPS receivers.

In the literature, there exist several studies investigating many techniques to solve the mentioned performance problems of traditional tracking loop in degraded signal environments [4], [5], [6], [7], [8], [9]. The vector tracking loop based solution methods appear to be the most attractive ones for making the GPS receivers more robust in weak signal environments. In such environments, a GPS receiver captures multiple strong and weak signals. This fact is the basis of the idea of vector tracking loops in which a relatively high power satellite can aid a weak power satellite by combining the tracking and navigation stages. The study given in [10] was the first proposal of a vector tracking loop with this kind of combination. After that, many other works on vector tracking architectures have been implemented; such as, vector delay lock loop (VDLL) and vector delay/frequency lock loop (VDFLL) [4], [5], [6], [7].

In the scope of this thesis, the vector delay/frequency lock loop algorithm given in [5] is implemented. In addition, the mentioned algorithm is modified in the sense of Kalman filtering and implemented. The traditional tracking loop given in [11] is also implemented as a reference performance measurement algorithm. The performance of vector tracking algorithms and the traditional tracking algorithm are compared. The superiority of the vector tracking algorithms over scalar tracking algorithms is shown with various experiments. The advantages of the modified VDFLL algorithm over the VDFLL algorithm, such as lower computational load, are also illustrated.

In all mentioned studies Extended Kalman Filter models have similar size of observation matrix [4], [5]. In this thesis, an Extended Kalman Filter model with fewer observation states is also proposed and performance comparison is done with mentioned vector tracking loops and traditional tracking loop.

Estimation of the C/No ratios is very important to verify the simulation scenarios used to compare the performances of vector and scalar tracking loops. Therefore, the

methods for estimation of C/No ratios explained in [24] and [25] are implemented as a part of the vector tracking algorithms. The C/No ratio estimations can also be used to calculate the covariance matrix of measurement noise which is a future work.

1.2 Outline of the Thesis

Basics of Global Positioning System are given in Chapter 2, which covers the history and main operating principles of GPS. Also, the GPS segments required for full operation are introduced. In addition, basic principles of the satellite constellation are discussed. Lastly in Chapter 2, GPS signal structures are discussed in details.

In Chapter 3, general properties of a software defined GPS receiver are described. As a part of a receiver, antenna and front-end are briefly explained. Also, the software part of the receiver is covered. Acquisition and navigation methods are presented in detail.

Details of the scalar tracking algorithms are explained in Chapter 4, Carrier and code tracking methods are discussed and the discriminators that can be used in these methods are given. Based on the discussed carrier/code tracking methods and the optimum discriminators, the complete scalar tracking block is presented. Finally in Chapter 4, performance effecting parameters of the scalar tracking algorithms are explained briefly.

Chapter 5 focusses on the theoretical explanation of the vector tracking algorithms, especially Vector Delay/Frequency Lock Loop (VDFLL). Vector tracking algorithms are discussed in two main parts as coherent and non-coherent algorithms. Firstly, the coherent algorithms are discussed briefly as they are not in the scope of the thesis. Secondly, mechanisms of VDFLL as a non-coherent algorithm explained in details. Discriminators that can be used in VDFLL algorithms are presented. Also, design method of the navigation filter, which is the most important part of a VDFLL, is discussed in details. In addition, the modified version of the VDFLL algorithm having less navigation filter parameters is proposed and explained as an alternative to original VDFLL algorithm. Lastly in this chapter, some methods for tuning the filter parameters are given.

Performance comparison results of scalar tracking loops and vector tracking loops are presented in Chapter 6. Firstly, the signal acquisition setup and simulation scenarios are explained in details. Then, the superiorities of vector tracking algorithms over scalar tracking algorithms in low C/No ratio environments are verified by using three different datasets. In addition, computational load advantage of modified VDFLL algorithm over VDFLL is illustrated with some computation time results comparison in this chapter.

In the last chapter, Chapter 7, conclusions of the thesis and future works for the improvement of the vector tracking algorithms are presented.

CHAPTER 2

BASICS OF GPS

GPS is a satellite radio navigation and time broadcasting system. It was designed and developed in the 1970's by the United States Department of Defense (DoD). The initial purpose was to provide navigation information to the U.S. military but nowadays, it is the most complex radio navigation signal to date, employing code division multiple access (CDMA) spread spectrum modulation to supply an unaided horizontal position estimate to within 100 meters for civilian users worldwide [30]. The system aims to continuously provide navigation data with high accuracy to any user at any point on earth under any weather condition. So in this section, GPS basics are described briefly such as the system description and signal properties.

2.1 GPS Segments

The overall system of GPS consists of three major segments: space segment, control segment and the user segment. Figure 2.1 shows the interaction of them.

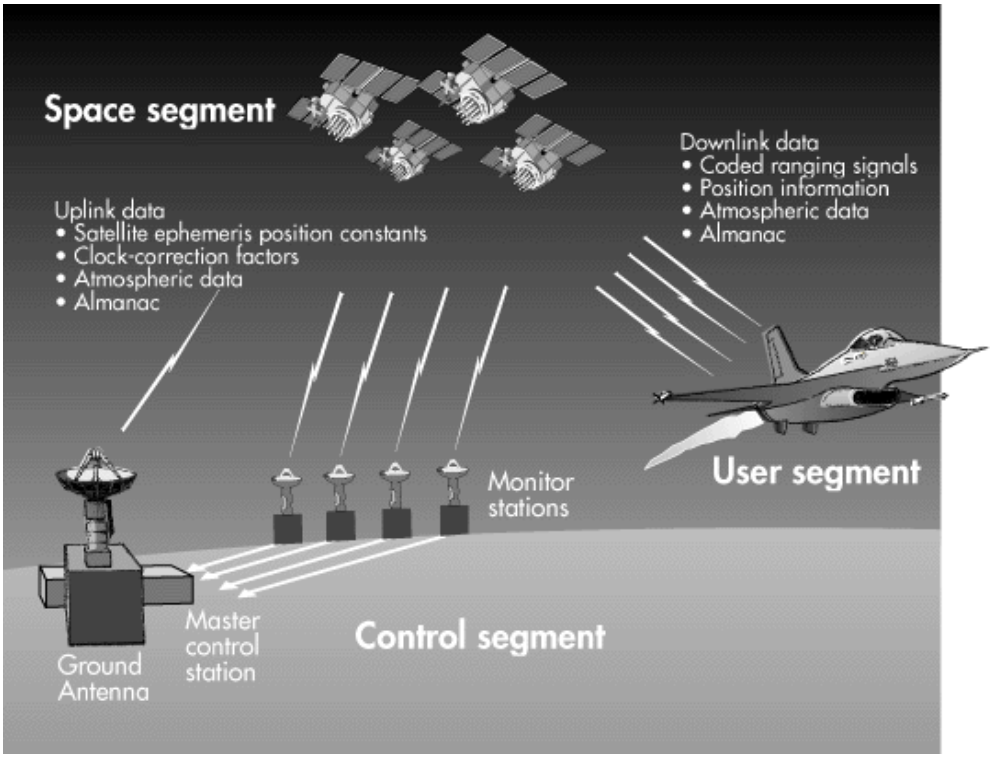

Figure 2.1 GPS Segments [34]

2.1.1 Space Segment

The space segment consists of a minimum of 32 satellites that operate 12-hour Earth-centered orbit [12]. There are six different circular actively used orbits that are 20200 km above the earth at an inclination angle of 55 degrees. Figure 2.2 shows the allocation of the satellites on circular orbits.

Figure 2.2 Space Segment [34]

GPS receivers require at least 4 visible satellites in order to get navigation information and typically the satellites are spaced in orbits as any time minimum of 6 satellites will be visible to users anywhere in the world.

All GPS satellites broadcast some information on two carrier frequencies (L1 and L2), such as synchronized time signals, position information of all other satellites and the orbital parameters [30]. All satellites have a specific pseudo-random code (PRN) and use that code to modulate broadcasting signal. The receiver determines satellite number thanks to that code.

2.1.2 Control Segment

The control segment consists of the master control station, ground antennas and monitor stations controlled by Operational Control System. All GPS satellites are monitored from sixteen monitor stations that are suitably located on precisely known locations. Six of the stations belong to the U.S. Air forces and remaining ones belong to the National Geospatial-Intelligence Agency (NGA). The station on Colorado Springs is both master station and a monitor station [13]. Locations of these stations are shown on Figure 2.3.

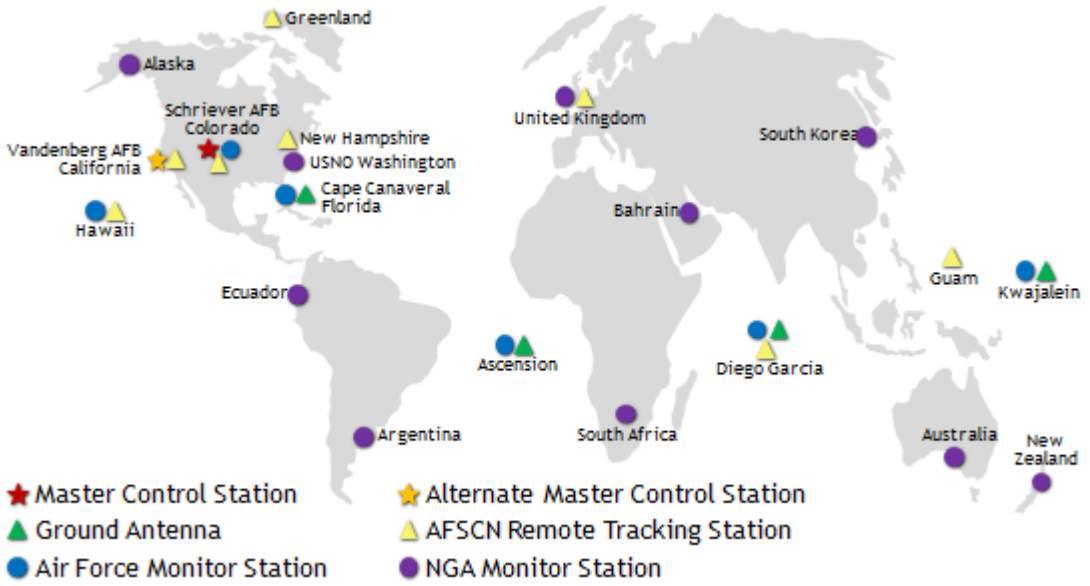

Figure 2.3 Locations of control stations [31]

The master control station controls the overall system by performing primary control segment functions and providing command and control of the GPS constellation. It periodically calculates and uploads the clock corrections and ephemeris information related to any satellites. It also ensures the health and accuracy of the satellites by collecting the information from the monitor stations.

Monitor stations track the satellites while they are visible and collect atmospheric data in order to send the observations to the master station.

2.1.3 User Segment

GPS is mainly used for navigation purposes and any system with a GPS receiver is called a user. GPS receivers are passive devices that just receive the data from the satellites. Therefore, infinite number of users can use GPS at the same time without interfering each other. The application areas can be categorized into two main topics as follows:

1. Military purposes
    - Navigation of military platforms
    - Search and rescue
    - Target pointing
    - Missile guidance
    - INS aiding
2. Civil applications
    - Navigation of civil transport vehicles
    - Cadastral mapping
    - Geodetic and geodynamic measurements

2.2 Basic GPS Principles

Basics of GPS simply depend on a geometric phenomenon called trilateration: the position of a point can be calculated for the case that the distances to 3 different

points with precisely known locations. In summary, if the distance from a point to an object is known, that point must be located on a sphere centered by the mentioned object. If the distances to three different objects are known, point must be located on the intersection of three spheres centered by corresponding objects. Two spheres intersect to make a circle and this circle intersects the third sphere to produce two points. In order to determine which point is the user position, theoretically one more object is needed. However, for the GPS case, it is assumed that the user is close to the earth and only one of the calculated points is close to the earth thus, it is selected as the user position. Figure 2.4 shows the trilateration principle.

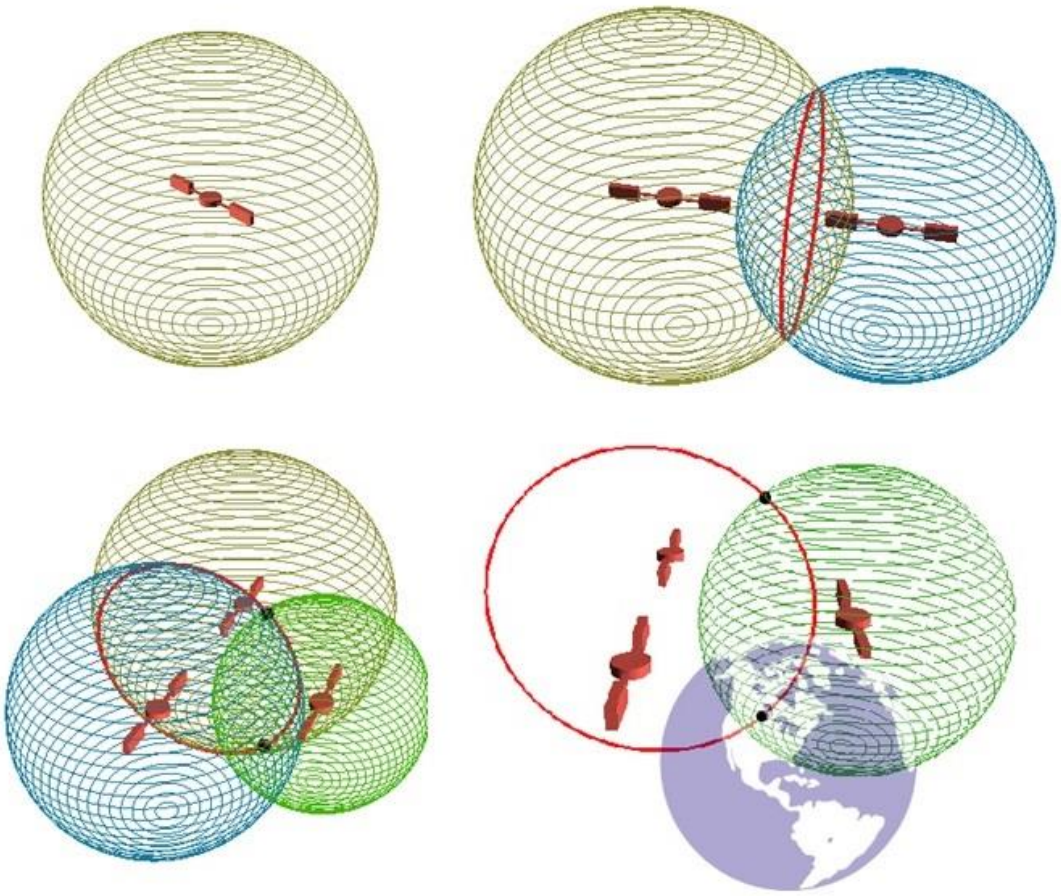

Figure 2.4 Trilateration principle

Each satellite transmits a signal with a time reference associated with it. By decoding the signal that time reference can be obtained. The distance between the satellite and the user can be calculated by measuring time of the signal travelling from the satellite to the user, which is simply multiplication of time difference between satellite and the receiver with speed of light as shown in equation (2.1).

$$\rho_i = c(t_u - t_s) \tag{2.1}$$

where c is the speed of light, ρ_i is the true value of pseudorange from user to satellite i, t_s is the reference time or the true time of transmission and t_u is the true time of reception. Theoretically, it seems very easy to calculate the pseudoranges. However, it is almost impossible to get exact time from the satellite or the user. Thanks to the atomic clocks of the satellites, they provide the clock information precisely. On the other hand, receivers are not able to provide a precise clock data. Due to that clock bias error, another unknown appears which causes requirement of one more visible satellite. As a result, position equations can be written as follows [11]:

$$\rho_1 = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + b \tag{2.2}$$

$$\rho_2 = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + b \tag{2.3}$$

$$\rho_3 = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + b \tag{2.4}$$

$$\rho_4 = \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + b \tag{2.5}$$

where b is the clock bias error expressed in distance, (x_i, y_i, z_i) is the three dimensional position of *ith* satellite and (x_u, y_u, z_u) is the three dimensional position of the user.

2.3 GPS Signal Structure

GPS signal consists of three different signals named as carrier signal, PRN code (C/A and P) and navigation data. In this section details of these signals are presented.

2.3.1 GPS Carrier Signal

The GPS signals consist of civilian and military components. The civilian component is open for access to all users and it can be used for navigation. The military signal which has enhanced positioning capabilities is available only the secure users. The civilian carrier signal is called L_1 and the military carrier signal is called L_2. For the work of the thesis, the civilian component, L_1 is used only. L_1 and L_2 carrier frequencies are derived from a common central frequency (f_o) and the exact frequencies (f_{L1} and f_{L2}) are shown below [11].

$$f_0 = 10.23 MHz \tag{2.6}$$

$$f_{L_1} = 154 f_0 = 1575.42 MHz \tag{2.7}$$

$$f_{L_2} = 120 f_0 = 1227.60 MHz \tag{2.8}$$

L_1 and L_2 carrier signals are modulated with PRN codes and navigation data in order to be able to carry information such as, the satellite clock corrections, orbital parameters. Each satellite has a specific PRN code that prevents interaction or coupling of signals broadcasted from different satellites. All PRN codes are uncorrelated with each other and can be recognized by CDMA technique [11], [13].

The L_1 signal broadcasted from kth satellite can be written as follows [11]:

$$\begin{aligned} S_{L1k} &= \sqrt{2P_c}\,(C^k(t) \oplus D^k(t))\cos(2\pi f L_1 t) \\ &+ \sqrt{2P_p}\,(p^k(t) \oplus D^k(t))\sin(2\pi f L_1 t) \end{aligned} \tag{2.9}$$

Where P_c and P_p are the powers of the signals modulated with the C/A, P code respectively, which are explained in the next section. C^k and D^k are the C/A code sequences and navigation data sequence assigned to the satellite number k, respectively. As clearly seen on equation (2.9), there are in phase and quadrature components that are modulated with a PRN sequence. The in-phase component is only modulated by the C/A code and the quadrature component is modulated by P code. Both in phase and quadrature components are modulated with the same navigation sequence.

The L_2 signal broadcasted from kth satellite can be written as follows [11]:

$$S_{L2k} = \sqrt{2P_p}\,(p^k(t) \oplus D^k(t))\sin(2\pi f L_2 t) \qquad \textbf{(2.10)}$$

As clearly seen on equation (2.10), The L_2 signal is only modulated by P code.

Figure 2.5 shows the generation of L_1 and L_2 signals in a detailed way and Figure 2.6 shows the bandwidth difference of C/A and P code on a power spectral density graph. C/A code has 2.046 MHz null-to-null bandwidth and P code has 20.46 MHz null-to-null bandwidth.

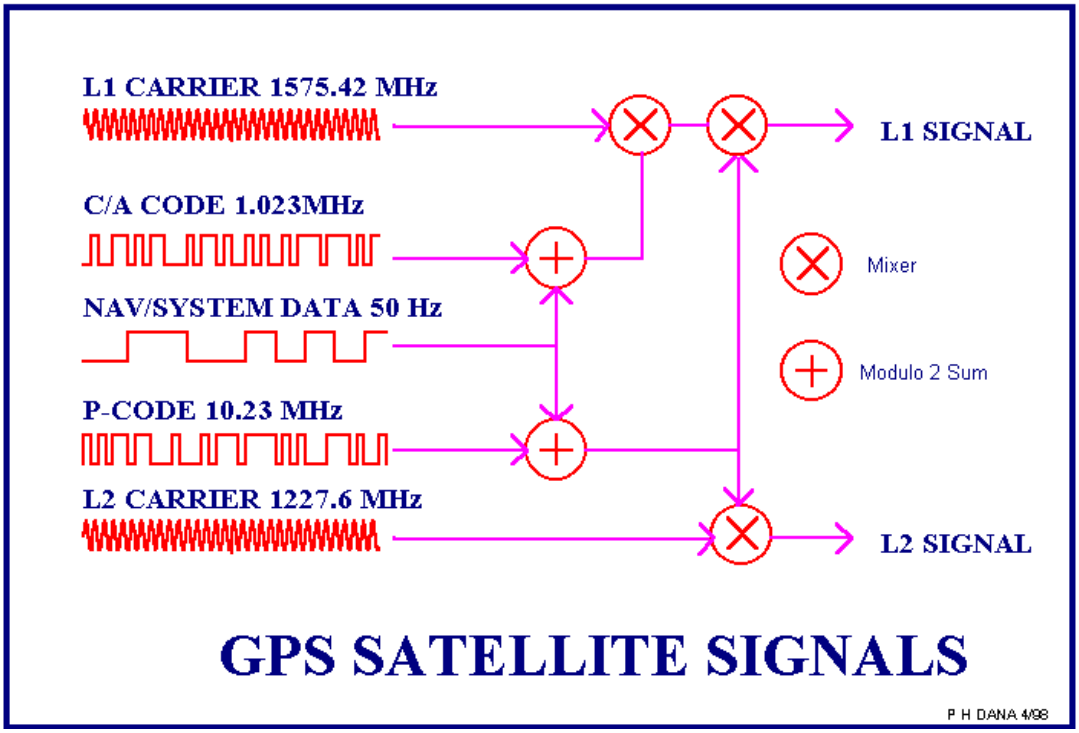

Figure 2.5 GPS Signal Generation [32]

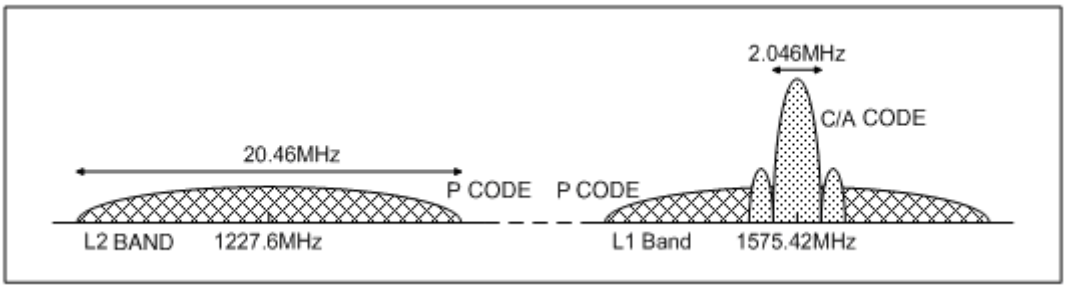

Figure 2.6 C/A and P Code Bandwidth [33]

2.3.2 C/A Code Sequence Properties

The C/A spreading code sequence is also called as pseudo-random noise sequence (PRN), due to their characteristics. Since they were determined by Robert Gold in 1967, they are also called as Gold codes [14]. The individual symbols of C/A codes are called as chips, since they carry no information. A C/A code has 1023 chips with a chipping rate of 1.023MHz which means it repeats in 1 millisecond.

The auto-correlation function of the C/A codes is very narrow and that property is used to track the satellite signal and make pseudorange measurements. Figure 2.7 shows the auto-correlation of PRN1. The maximum of correlation peak is 1023 that is equal to C/A code length. A GPS receiver recognizes the GPS signals with specific satellite number thanks to that property.

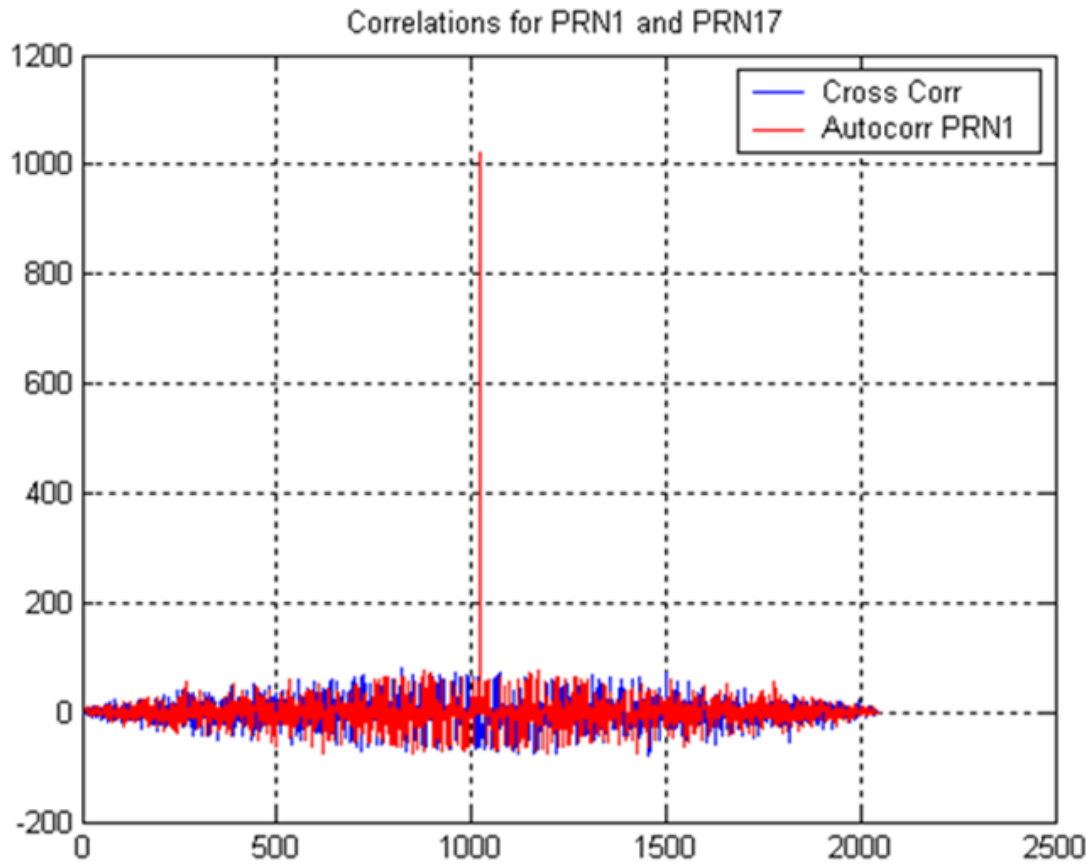

Figure 2.7 Auto-correlation and Cross-correlation of C/A Codes

Auto-correlation property is not enough to track the GPS signals. There is also cross-correlation property of C/A codes that is the different PRN codes cancel out each other. GPS uses code division multiple access (CDMA) and satellites are able to transmit on the same frequency with the help of that property. After recognizing the satellite number by using the auto-correlation property, the receiver easily picks out the corresponding satellite by multiplying the signal by an in-phase replica of the satellite that is generated by the local oscillator. This correlation effectively cancels out all the other satellite signals while removing the C/A code modulation from the signal. Figure 2.7 shows the cross-correlation of PRN1 and PRN17.

2.3.3 Navigation Data

The navigation data is transmitted within L_1 and L_2 carrier signals at a bit rate of 50 bits per second and it contains GPS system time, satellite's current health, its orbit information (ephemeris) and almanac data for the other satellites in use. The user can calculate the precise locations of the other visible satellites and transmission time of each signal using the navigation message.

Navigation data message consists of 5 sub-frames, each having 300 bits, so the main frame is 1500 bit-long. These sub-frames arranged into thirty bits words so a sub-frame has 10 words. Figure 2.8 shows the overall structure of a complete navigation message. First, second and third sub-frames are repeated in all frames. On the other hand, fourth and fifth sub-frames have 25 different types and the information on that sub-frames are meaningful only with those 25 sub-frames together. Thus, with the bit rate of 50 bits per second, a sub-frame is completed in 6 seconds, a frame is completed in 30 seconds and a full navigation data is completed in 12.5 minutes [11].

All sub-frames begin with two specific words, handover word (HOW) and the telemetry (TLM). TLM has 30 bits with a fixed preamble of 8 bits pattern which is very important for positioning calculations. The second word, HOW, starts with the truncated version of the time of week data. In addition, HOW contains two flags for anti-spoofing purposes [13]. Also, the last three bits indicates the sub-frame number.

After TLM and HOW words, there are 8 more words in all sub-frames, which indicate different information. As can be clearly seen on Figure 2.8, first sub-frame has all the clock information and health indicators. The second and third sub-frames contain ephemeris data, which includes data for the orbital calculations of satellites. Finally, the last sub-frames contain almanac and ionospheric correction data. All satellites transmit the same almanac data which is an approximate of ephemeris data for the other satellites in the constellation. It has less accuracy than ephemeris, but it is valid for longer period of time [11].

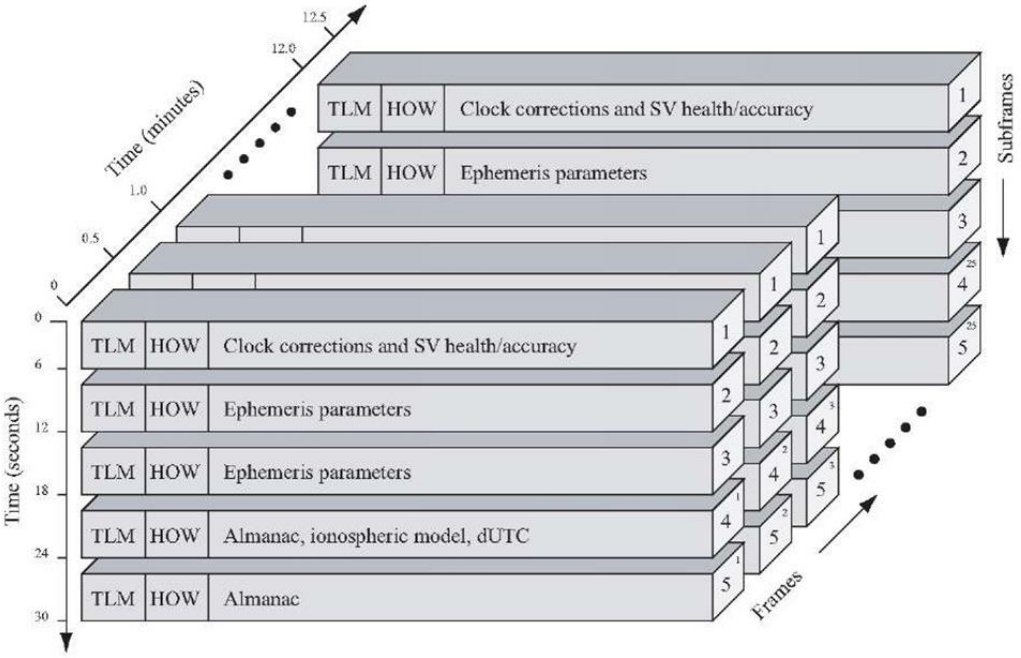

Figure 2.8 Navigation Data Message Structure [11]

CHAPTER 3

GPS RECEIVER ARCHITECTURE

A typical GPS receiver is illustrated in Figure 3.1. It consists of an antenna that picks up the signal transmitted from satellites, a front-end that digitizes and down converts the signals before processing, and an algorithm which handles the acquisition, tracking and navigation part.

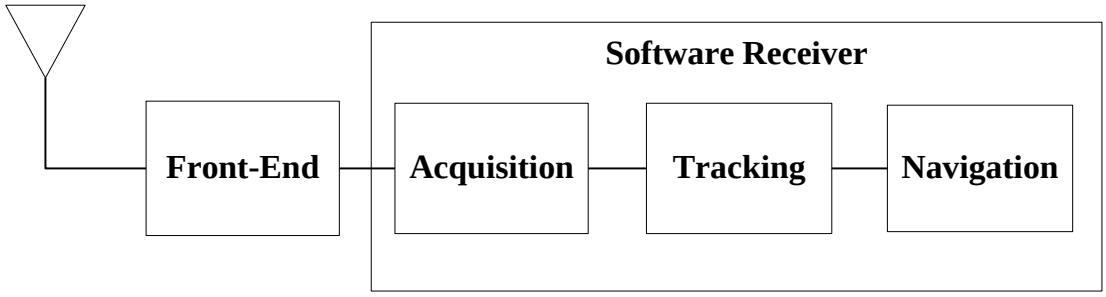

Figure 3.1 Generic GPS Receiver

3.1 GPS Antenna

The performance of a GPS receiver highly depends on the power of incoming signal, and suppressing the unwanted signals, which can be achieved with a high quality antenna. The antenna should be able to accommodate the appropriate bandwidth of the expected signal and filter the other parts. Since the GPS signals are right hand circularly polarized (RHCP), it should also be RHCP to prevent the multipath effect that is the polarization change from RHCP to LHCP in case of reflecting of a signal

from an object. The GPS antennas are very effective to cancel the multipath effect [1]. In addition, the low elevation also causes multipath effect, and to reduce that error, the antenna pattern of a GNSS receiver is designed in such a way that it receives signals only above 10^{O}-20^{O} elevation [11].

3.2 Front-End

A GPS receiver requires the signal received by the antenna to be down converted to an intermediate frequency and digitized in order to be able to process it. Figure 3.2 shows a generic GPS front-end schematic.

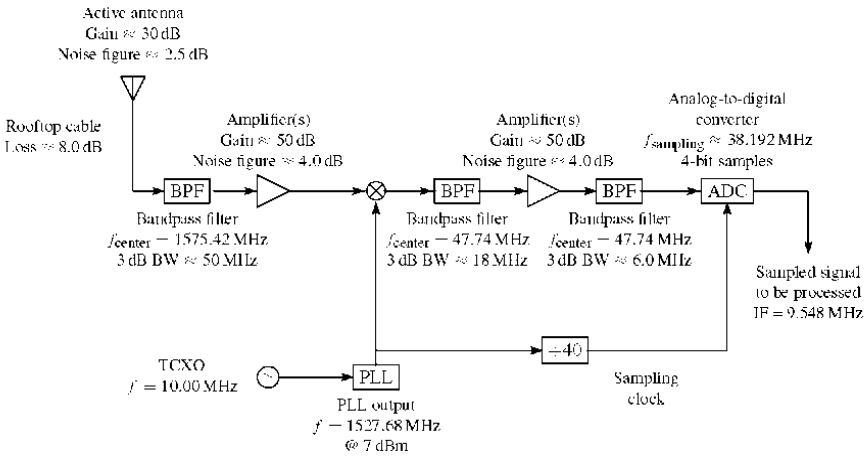

Figure 3.2 Generic GPS Front-End [11]

The first part of the front-end is a filter that allows only selected frequencies of a signal and attenuates the others. The first filter in Figure 3.2 is a bandpass filter with a center frequency 1575.42 MHz and 50 MHz bandwidth. Bandpass filters are preferred since they provide additional frequency selectivity, in contrast to lowpass or highpass filters [11].

The second stage is the amplification. Since the GPS satellites transmit very low power signals, amplifiers are very crucial to increase the signal magnitude to a suitable level for analog to digital conversion. Ideally, an amplifier just raises the

signal magnitude, but in practice, it also adds unwanted noise components to the signal, which is called as noise figure.

Basically, a local oscillator translates the carrier frequency to a lower intermediate frequency. The main goal for this is to convert the frequency to usable ranges to perform the analog to digital conversion [11]. The process is based on the trigonometric identity shown on equation (3.1).

$$\cos(w_1 t)\cos(w_2 t) = \frac{1}{2}\cos((w_1 - w_2)t) + \frac{1}{2}\cos((w_1 + w_2)t) \qquad \textbf{(3.1)}$$

In Figure 3.2, the L_1 frequency 1575.42 MHz is translated to intermediate frequency (IF) 47.74 MHz using the equation (3.1) by selecting w_1 as the L_1 center frequency and w_2 as (1575.42-47.74) 1527.68 MHz, which is the local oscillator frequency [11]. The next bandpass filter in Figure 3.2 is used to get the expected intermediate frequency and filter the high frequency component.

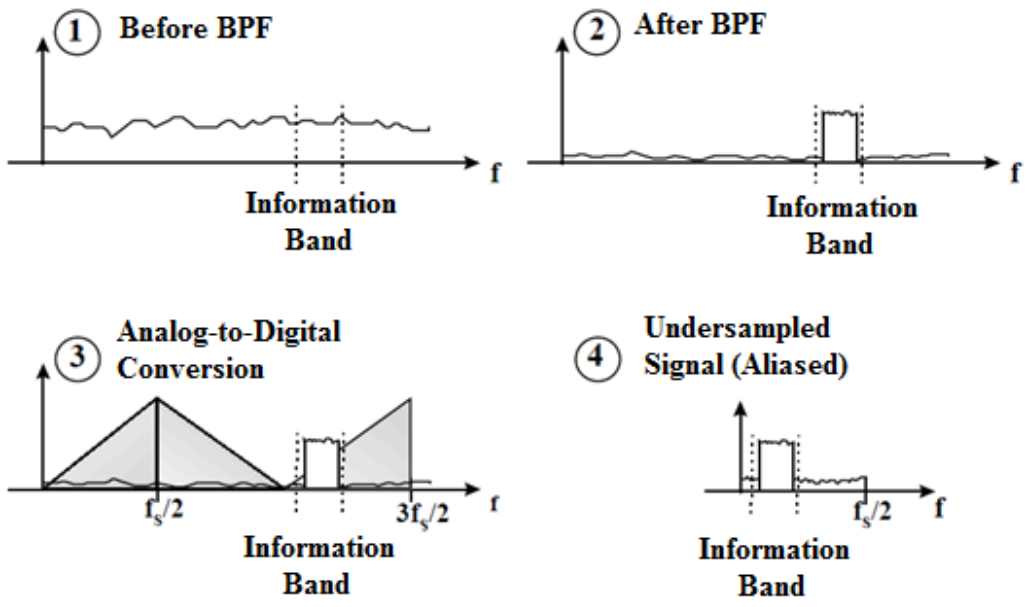

Figure 3.3 Undersampling [15]

Analog to digital conversion is the last stage of front-end. Analog to digital converter (ADC) simply converts the analog signal to digital samples. In Figure 3.2, the signal with frequency at 47.74 MHz sampled with an ADC at a sampling frequency 38.192

MHz and translated to an IF at 9.548 MHz. This method is called as bandpass sampling or undersampling. Figure 3.3 shows a simple representation of undersampling.

3.3 Acquisition

According to the last reports, there are 32 active satellites in the GPS satellite constellation [12]. However, they are not all visible from any point on earth. Therefore, an acquisition process is required to determine which satellites are in the view. In addition, with the acquisition process, code phases or the time alignment of the PRN codes in the current block of data are found. Also, the Doppler shifts caused by the relative motion for all satellites are determined. Actually, acquisition stage roughly estimates these parameters and passes them to tracking block. If there is a priori information about the user position and a satellite almanac, the acquisition process can be speeded up, called as hot start.

Although, the GPS signals are broadcasted with same carrier frequency by all satellites, the relative motion between satellites and the receiver causes a Doppler shift on the carrier frequency. The Doppler shift can cause a shift of ±10kHz at most for the fastest vehicle on earth [11]. Therefore, the acquisition algorithm should search in this range.

There are three different methods for acquisition; traditional method, FFT method and delay and multiply method.

3.3.1 Traditional Method

This method is based on serial search of the signal in time domain by using the auto-correlation property of PRN codes. It is basically hardware and very easy to implement.

$$z(n) = \frac{1}{N}\sum_{m=0}^{N-1} x(m)y(m+n) = \frac{1}{N}\sum_{m=0}^{N-1} x(-m)y(m-n) \qquad \textbf{(3.2)}$$

Basic operation of traditional method is shown on equation (3.2), where *x* is the input signal multiplied with proper carrier frequency, *y* is the corresponding PRN code and *z* is the correlation of them [11].

3.3.2 FFT Method

FFT method is most suitable method for software defined receivers. Thus, this method is selected for the simulations in this thesis. Main difference of the FFT method from traditional method is that its calculations are done on frequency domain. The discrete Fourier transforms of two signals can be calculated as follows [11]:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad and, \quad Y(k) = \sum_{n=0}^{N-1} y(n) e^{-j2\pi kn/N} \tag{3.3}$$

Fourier transform gives the ability to do calculations in frequency domain. The cross correlations of two transformed signals can be found as in equation (3.4).

$$Z(k) = X^{*}(k) Y(k) \tag{3.4}$$

where X^{*} is the complex conjugate of X and Z is the Fourier transform of cross-correlation of x and y. Obviously, the last step is an inverse Fourier transform as shown on Figure 3.4. The absolute value of the output of the inverse Fourier transform represents the correlation between the input and the PRN code. If a peak is presented in the correlation, the index of this peak marks the code phase of the input signal as in Figure 3.5.

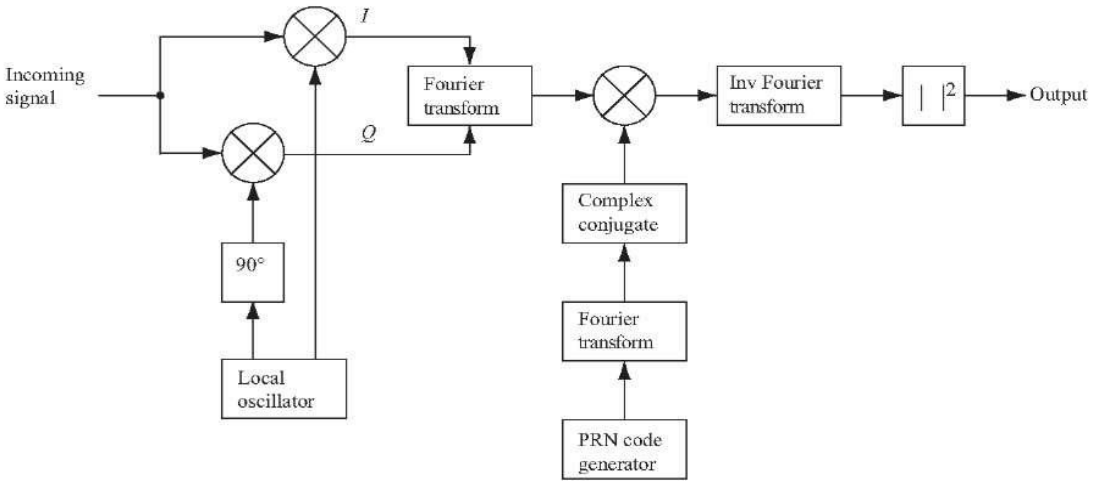

Figure 3.4 Acquisition with FFT method [11]

As shown on Figure 3.5, the correlation result of PRN19 has no peak since it is not visible and PRN21 is visible so a significant peak is present.

Normally, a step size of 500Hz is sufficient to search the ±10kHz frequency range that has cut down the search space to only 41 different carrier frequencies. Therefore, the same step size has been used in the acquisition algorithm of the software GPS receiver used in this thesis.

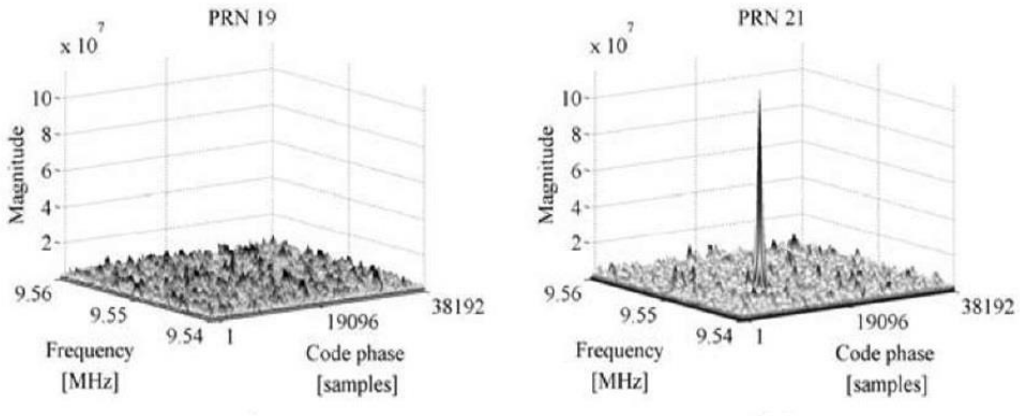

Figure 3.5 Sample Acquisition Results for PRN19 and PRN21 [11]

3.3.3 Delay and Multiply Method

This method mainly aims to get rid of the frequency information in the input signal. After the PRN code phase is found, the frequency can be calculated by using DFT or FFT [16]. The GPS input signal *s(t)* is assumed as complex, thus;

$$s(t) = C_s(t)e^{j2\pi ft} \tag{3.5}$$

where $C_s(t)$ is PRN code of the corresponding satellite. The delayed version can be written as follows:

$$s(t-\tau) = C_s(t-\tau)e^{j2\pi f(t-\tau)} \tag{3.6}$$

where τ represents the delay time. The next step is multiplication of s*(t)* with the complex conjugate of the delayed version of it.

$$s(t)s(t-\tau)^* = C_s(t)C_s(t-\tau)^* e^{j2\pi ft}e^{-j2\pi f(t-\tau)} \equiv C_n(t)e^{j2\pi ft} \tag{3.7}$$

where

$$C_n(t) = C_s(t)C_s(t-\tau) \tag{3.8}$$

As shown on equation (3.8), $C_n(t)$ is the multiplication of a PRN code and its delayed version, and it is also a PRN code that has the same correlation properties and even the same code phase with original one. As can be clearly seen from equation (3.7), there is no frequency component that means there is no need to search for Doppler shift and the code phase can be found easily by using cross-correlation property.

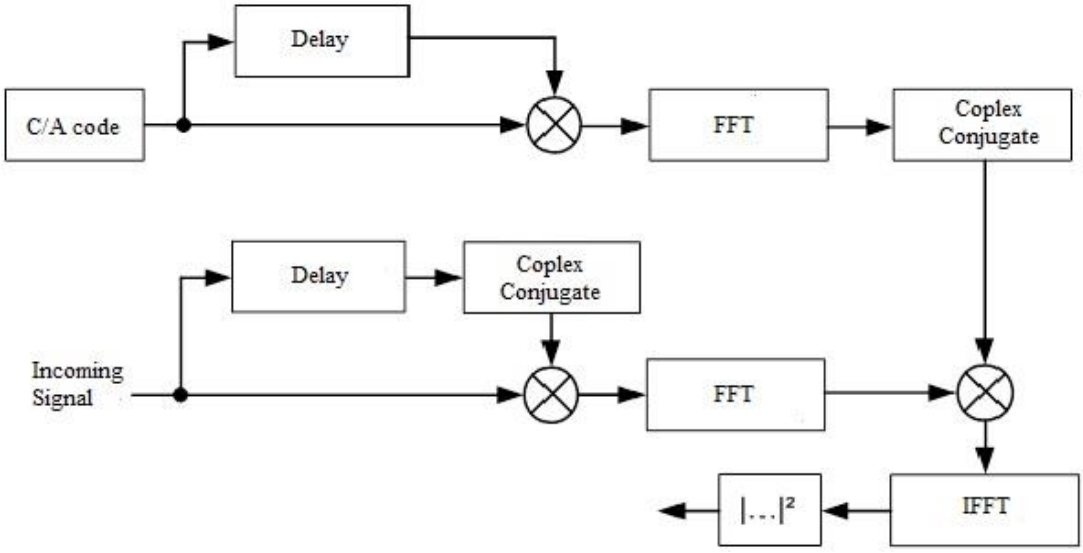

Figure 3.6 Acquisition with Delay and Multiply Method [17]

Figure 3.6 shows a simple block diagram of delay and multiply method.

This approach seems to be very attractive; however, there is a disadvantage that the noise floor increases when two noisy signals are multiplied together. Due to that problem, longer data are required to acquire a certain satellite [16].

3.4 Tracking

After a successful acquisition of the satellites, the receiver passes the next phase, called tracking. A typical GPS receiver uses code and carrier tracking loops for tracking of the satellites. A delay lock loop (DLL) is used to keep the alignment of code phase and a phase lock loop (PLL) or a Costas loop can be used to track the carrier phase. There are different types of tracking loop algorithms such as scalar and vector tracking loop. For the scalar tracking loops, the discriminator functions are used to determine the phase offset of the received and locally generated signal and each acquired satellite is tracked independently that is the reason why they are called as scalar tracking loops. On the other hand, vector tracking loops process the GPS signal in an aggregate manner and mutual aiding of the weaker signals can be achieved by the presence of stronger signals. Since the tracking loops are in the

scope of this thesis, they will be discussed in a more detailed way in Chapter 4 and Chapter 5.

3.5 Navigation

The main purpose of GPS receiver is to determine its own position precise enough. Therefore, navigation phase is another important part of the algorithm. In order to achieve that aim, navigation data message and pseudorange measurement are used.

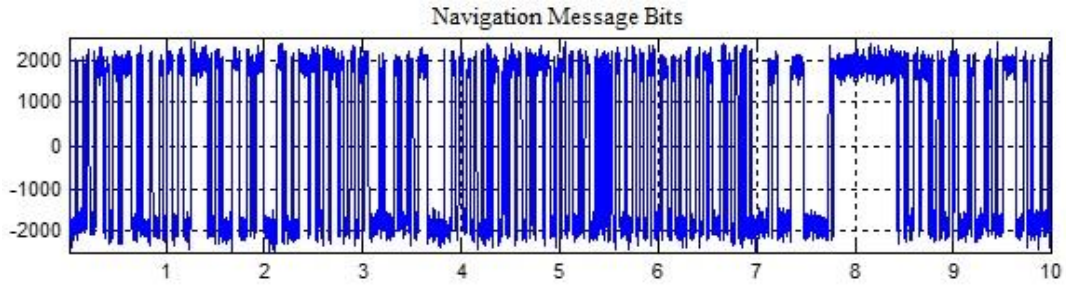

Figure 3.7 Navigation Message Bits

Figure 3.7 shows a sample of navigation message bits as the output of tracking phase. The details of navigation data structure is given in part 2.3.3 and the complete details about navigation data structure and decoding can be found from [11].

3.5.1 Pseudorange Estimation

The Pseudorange estimation can be analyzed in two main parts: initial set of pseudoranges and estimation of subsequent pseudoranges.

To find the initial set of pseudoranges between the receiver and tracked satellites, travel time of the signals must be known. Obviously, result of multiplication of that duration with the speed of light is the pseudoranges. The actual signal transmit time is given on the navigation data, but there is always a clock bias error between the satellite clock and receiver clock which is unpredictable. Therefore, first of all, by checking the TLM words in navigation message, the difference between times of arrivals of different satellites are found. It is also known that the travel time of the

signals from the satellites to the Earth is between 65-83 milliseconds. Thus, by assuming travel time of the first arrived signal as 65 ms, the travel time of other signals can also be found.

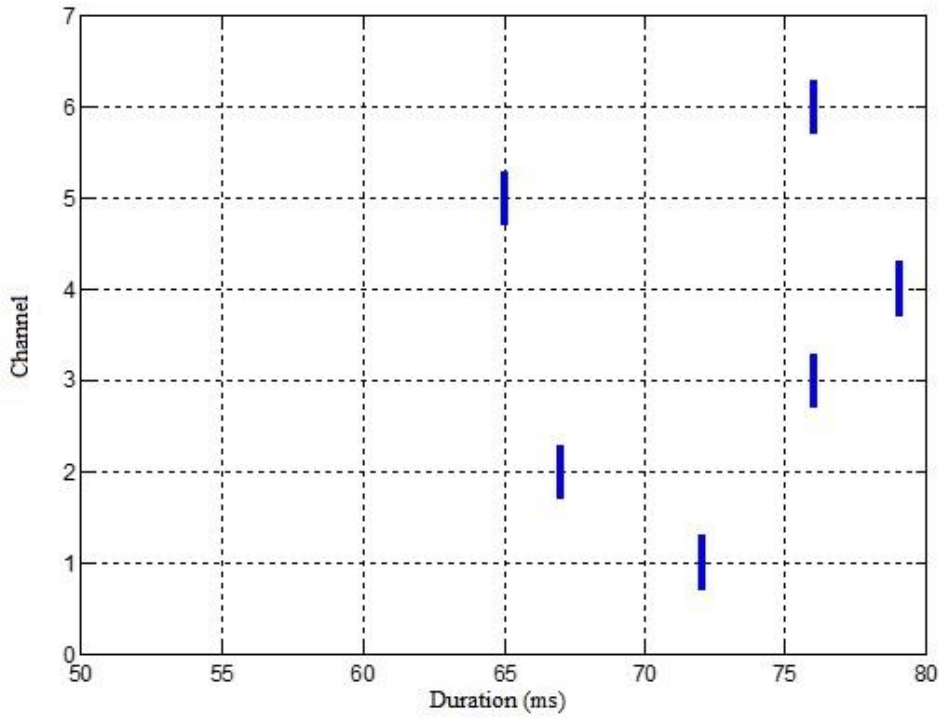

Figure 3.8 Travel time estimation

Figure 3.8 shows the initial estimation of travel times of signals. As explained above, travel time of channel 5 is assumed 65 ms and the others are calculated with respect to it.

After the initial estimation of travel times, to calculate the subsequent pseudoranges two information should be tracked. The first one is the difference in ms between the start of the sub-frame from the reference satellite. The second one is the start of the PRN code, which increases the resolution and gives almost exact pseudorange for corresponding channel [11].

3.5.2 User Position Determination

In order to calculate the user's position, at least pseudorange measurements from four satellites are required as mentioned before. By ignoring the ionosphere, troposphere and the satellite clock offset, the pseudorange measurement can be expressed as in equation (3.9).

$$P_j = \rho_j + cdt_u \tag{3.9}$$

Where ρ_j is the geometrical distance between *jth* satellite and the receiver and dt_u is the receiver clock bias. The geometrical distance ρ_j can be written with respect to the user's coordinates (x_u, y_u, z_u) as follows [1]:

$$\rho_j = \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} \tag{3.10}$$

Substituting the equation (3.10) into (3.9):

$$P_j = \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} + cdt_u \tag{3.11}$$

Where (x_j, y_j, z_j) are the satellite coordinates which are calculated using the ephemeris transmitted in the navigation message. Equation (3.11) can either be solved by using least squares method or using an extended Kalman filter (EKF). Both of these methods are used in this thesis and discussed in details in the following parts.

3.5.2.1 Least Squares Method

Since the equation (3.11) is nonlinear with respect to the satellite coordinates (x_j, y_j, z_j), before using the least squares method, it must be linearized. Actually, the nonlinear part is given in equation (3.10). For linearization, nominal estimation of the receiver's position is chosen first which is generally the center of earth (0, 0, 0).

Let Δx, Δy and Δz are the changes in position, and the position updates can be expressed as follows:

$$x_{u,1} = x_{u,0} + \Delta x_u \tag{3.12}$$

$$y_{u,1} = y_{u,0} + \Delta y_u \tag{3.13}$$

$$z_{u,1} = z_{u,0} + \Delta z_u \tag{3.14}$$

Where $(x_{u,0}, y_{u,0}, z_{u,0})$ are the initial nominal estimation of the user position. Based on these relations, the equation (3.10) can be rewritten as follows:

$$\rho_j = f(x_{u,1}, y_{u,1}, z_{u,1}) = f(x_{u,0} + \Delta x_u, y_{u,0} + \Delta y_u, z_{u,0} + \Delta z_u) \tag{3.15}$$

At this point the pseudorange equations can be linearized by expanding the equation using Taylor series.

$$\begin{aligned} f(x_{u,1}, y_{u,1}, z_{u,1}) = f(x_{u,0}, y_{u,0}, z_{u,0}) + \frac{\delta f(x_{u,0}, y_{u,0}, z_{u,0})}{\delta x_{u,0}} \Delta x_u \ldots \\ \ldots + \frac{\delta f(x_{u,0}, y_{u,0}, z_{u,0})}{\delta y_{u,0}} \Delta y_u + \frac{\delta f(x_{u,0}, y_{u,0}, z_{u,0})}{\delta z_{u,0}} \Delta z_u \end{aligned} \tag{3.16}$$

From the first order Taylor series expansion, the equation (3.16) is obtained by using partial derivatives. The equation truncated at the first order, so it can be used to calculate an approximate positioning solution. The partial derivatives calculated as follows [11]:

$$\frac{\delta f(x_{u,0}, y_{u,0}, z_{u,0})}{\delta x_{u,0}} = -\frac{x_j - x_{u,0}}{\rho_j} \tag{3.17}$$

$$\frac{\delta f(x_{u,0}, y_{u,0}, z_{u,0})}{\delta y_{u,0}} = -\frac{y_j - y_{u,0}}{\rho_j} \tag{3.18}$$

$$\frac{\delta f(x_{u,0}, y_{u,0}, z_{u,0})}{\delta z_{u,0}} = -\frac{z_j - z_{u,0}}{\rho_j} \tag{3.19}$$

Therefore, first order positioning equation is as follows:

$$P_j = \rho_j - \frac{x_j - x_{u,0}}{\rho_j} \Delta x_u - \frac{y_j - y_{u,0}}{\rho_j} \Delta y_u - \frac{z_j - z_{u,0}}{\rho_j} \Delta z_u + c dt_u \tag{3.20}$$

Equation (3.20) can also be in vector form as in equation (3.21).

$$P_j = \rho_j + \begin{bmatrix} -\frac{x_j - x_{u,0}}{\rho_j} & -\frac{y_j - y_{u,0}}{\rho_j} & \frac{z_j - z_{u,0}}{\rho_j} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ cdt_u \end{bmatrix} \quad \textbf{(3.21)}$$

Rewriting it in the form of $Ax = b$ that is suitable for least squares method [11]:

$$\begin{bmatrix} -\frac{x_j - x_{u,0}}{\rho_j} & -\frac{y_j - y_{u,0}}{\rho_j} & \frac{z_j - z_{u,0}}{\rho_j} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ cdt_u \end{bmatrix} = P_j - \rho_j \quad \textbf{(3.22)}$$

Finally, for all satellites it can be written as [11]:

$$Ax = \begin{bmatrix} -\frac{x_1 - x_{u,0}}{\rho_1} & -\frac{y_1 - y_{u,0}}{\rho_1} & \frac{z_1 - z_{u,0}}{\rho_1} & 1 \\ -\frac{x_2 - x_{u,0}}{\rho_2} & -\frac{y_2 - y_{u,0}}{\rho_2} & \frac{z_2 - z_{u,0}}{\rho_2} & 1 \\ -\frac{x_3 - x_{u,0}}{\rho_3} & -\frac{y_3 - y_{u,0}}{\rho_3} & \frac{z_3 - z_{u,0}}{\rho_3} & 1 \\ & \vdots & & \\ -\frac{x_m - x_{u,0}}{\rho_m} & -\frac{y_m - y_{u,0}}{\rho_m} & \frac{z_m - z_{u,0}}{\rho_m} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ cdt_u \end{bmatrix} = \begin{bmatrix} P_1 - \rho_1 \\ P_2 - \rho_2 \\ P_3 - \rho_3 \\ \vdots \\ P_m - \rho_m \end{bmatrix} = b \quad \textbf{(3.23)}$$

The equation (3.22) requires at least four pseudorange measurements to have a solution. Obviously, if there are four pseudorange measurements, the equation can be solved just inverting the matrix A and multiplying with b as in equation (3.24) [13].

$$x = A^{-1}b \quad \textbf{(3.24)}$$

However, if there are more than four pseudorange measurements, the inverse of A matrix will not be unique. In this case, it can be solved using Moore-Penrose pseudoinverse as in equation (3.25) [13].

$$x = \left(A^T A\right)^{-1} A^T b \quad \textbf{(3.25)}$$

The approximate user position is updated by using the equations (3.12), (3.13), and (3.14) with the solution that is found by equation (3.24) or (3.25). Using these new

user position estimates, calculations are repeated starting from the equation (3.23) until the position differences are at meter level.

3.5.2.2 Kalman Filtering Method

The user states like clock errors cannot be modeled in the least squares solution method, since it is a point solution. On the other hand, Kalman Filtering is a recursive algorithm and it has the memory of previous measurements in order to use them to obtain more refined results with current measurements. Details of Kalman filters are explained in Chapter 5 and Appendix A.

In this study, an EKF for Vector Delay/Frequency Lock Loop algorithm is implemented. Therefore, a brief summary of the mentioned EKF is given in this section. Simply, it estimates and tracks the user's state errors and clock bias/drift errors by using the inputs coming from discriminators. The estimated states are used to correct the actual user's states and clock bias/drift which are kept outside of the filter.

The state matrix can be defined as follows:

$$X_E = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta v_x \\ \delta v_y \\ \delta v_z \\ t_b \\ \Delta t_d \end{bmatrix} = \begin{bmatrix} X\ axis\ Position\ Error \\ Y\ axis\ Position\ Error \\ Z\ axis\ Position\ Error \\ X\ axis\ Velocity\ Error \\ Y\ axis\ Velocity\ Error \\ Z\ axis\ Velocity\ Error \\ Clock\ Bias\ Error \\ Clock\ Drift\ Error \end{bmatrix} \quad \textbf{(3.26)}$$

User's states are defined in the Earth Centered-Earth Fixed (ECEF) reference frame.

The process update matrix is written as follows:

$$F_{k,k+1} = \begin{bmatrix} 1 & 0 & 0 & \Delta t_{k+1} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t_{k+1} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t_{k+1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t_{k+1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(3.27)}$$

where,

Δt_{k+1} = duration between t_k and t_{k+1} (in second), also called filter update rate.

The user velocity and position predictions can be corrected by using the errors estimations as follows.

$$X_{k+1} = \hat{X}_{k+1} + \delta X_{k+1} \quad \textbf{(3.28)}$$

$$V_{k+1} = \hat{V}_{k+1} + \delta V_{k+1} \quad \textbf{(3.29)}$$

where,

X_{k+1} and V_{k+1} = user position/velocity vector in ECEF frame, respectively

$\hat{X}_{k+1}$ and $\hat{V}_{k+1}$ = predictions of user position/velocity vector in ECEF frame, respectively

δX_{k+1} and δV_{k+1} = the EKF outputs as user position/velocity error vectors, respectively.

The corrected user positions and velocities can be used directly as navigation output of the receiver.

CHAPTER 4

SCALAR TRACKING LOOP DESIGN

In this chapter, the scalar tracking loop algorithm that is mentioned in part 3.4 is explained in detailes.

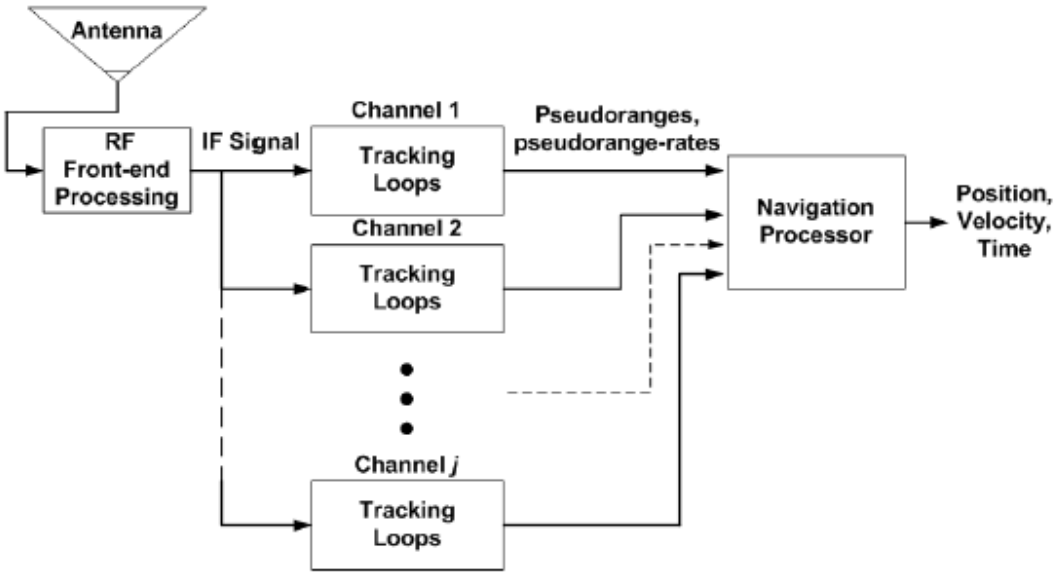

Figure 4.1 Traditional Receiver Architecture [23]

Figure 4.1 shows simple traditional receiver architecture. The traditional tracking loops are also called as scalar tracking loops since each receiver channel is tracked individually. The outputs of each channel as pseudoranges and pseudorange rates are directly input to the navigation processor as shown on Figure 4.1 which is the only

interaction of channels. This causes a certain disadvantage that aiding of weaker signals is not possible by the other signals. As a result, scalar tracking loops operate worse under high user dynamics and low carrier-to-noise (C/No) ratio. However, scalar tracking loops are relatively easier to implement and more robust since the noise in any channel cannot corrupt the others.

4.1 Carrier Frequency Tracking

For the correct decoding of the navigation data an almost exact replica of the carrier signal must be generated. For this purpose, generally, a phase lock loop (PLL) is used. A standard PLL is sensitive to 180º phase shifts which is unacceptable due to the phase shifts caused by navigation data changes. Therefore, generally a Costas loop that is insensitive to this phase reversal is preferred in a GPS receiver [11].

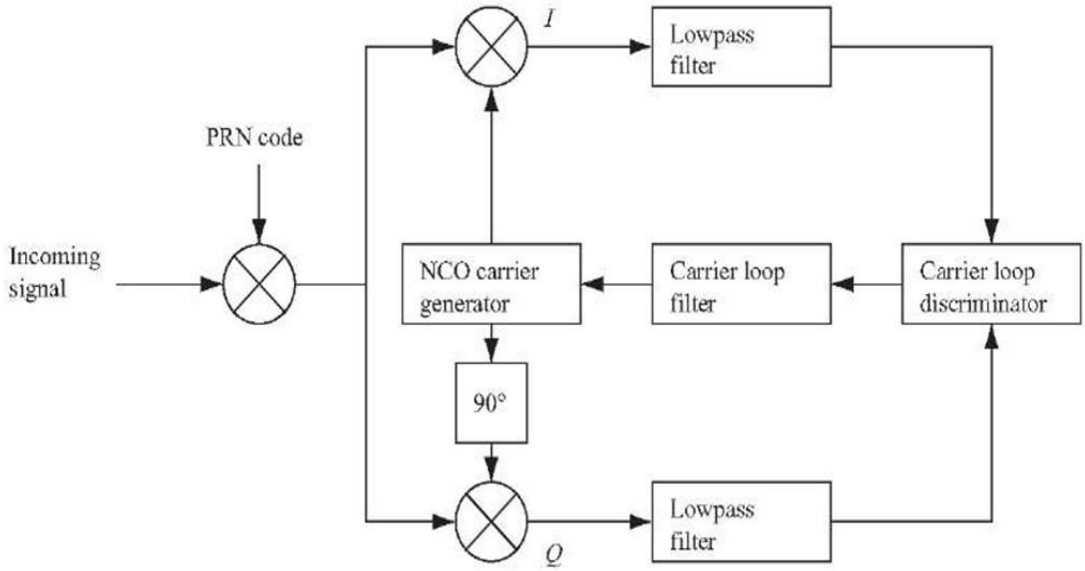

Figure 4.2 Block Diagram of Costas Loop [11]

Figure 4.2 shows the block diagram of a Costas loop. There are two multiplication processes in a Costas loop. The first one is between the incoming signal and locally generated replica and the second one is between the incoming signal and 90º phase shifted version of the locally generated replica. The first multiplication outputs the in

phase (I) arm and the other one outputs the quadrature arm (Q). The aim of the Costas loop is to track the I arm that requires a feedback to the NCO carrier generator. In Figure 4.2, assuming that the PRN code is exactly known, the result of the multiplication on the I arm is as follows:

$$D^j(n)\cos(w_{IF}n)\cos(w_{IF}n+\varphi)=\frac{1}{2}D^j(n)\cos(\varphi)+\frac{1}{2}D^j(n)\cos(2w_{IF}n+\varphi) \quad \textbf{(4.1)}$$

where φ is the phase difference between the generated replica of the carrier signal and incoming signal.

Similarly, the result at the Q arm is as follows:

$$D^j(n)\cos(w_{IF}n)\sin(w_{IF}n+\varphi)=\frac{1}{2}D^j(n)\sin(\varphi)+\frac{1}{2}D^j(n)\sin(2w_{IF}n+\varphi) \quad \textbf{(4.2)}$$

Figure 4.2 shows that the next part is lowpass filtering which is used to surpass the high frequency components. In a software defined GPS receiver *"integrate and dump"* functions, which are just integrators, are used as lowpass filters since, the signals in a software defined receiver are discrete instead of being continuous. The results shown on equation (4.1) and (4.2) on I and Q arms are integrated over an interval.

After the lowpass filtering part, the outputs of I and Q arm can be written as in equation (4.3) and (4.4).

$$I^j=\frac{1}{2}D^j(n)\cos(\varphi) \quad \textbf{(4.3)}$$

$$Q^j=\frac{1}{2}D^j(n)\sin(\varphi) \quad \textbf{(4.4)}$$

The phase error can be calculated as follows:

$$\frac{I^j}{Q^j}=\frac{\frac{1}{2}D^j(n)\cos(\varphi))}{\frac{1}{2}D^j(n)\sin(\varphi))}=\tan(\varphi) \quad \textbf{(4.5)}$$

$$\varphi=\tan^{-1}\left(\frac{I^j}{Q^j}\right) \quad \textbf{(4.6)}$$

The equation (4.6) shows the two-quadrant arctangent non-linear discriminator function which has optimal performance for high and low SNR [13]. There are many other types of the discriminators and some of them are shown on Table 4.1 with their descriptions. Although the arctangent discriminator is the most time consuming discriminator, it is also the most precise one.

Table 4.1 Discriminators for Costas Loop

Discriminator	Description
$\varphi = \tan^{-1}\left(\frac{Q^j}{I^j}\right)$	The output is the phase error.
$\varphi = I^j Q^j$	The output is proportional to $\sin(2\varphi)$
$\varphi = sign(I^j)Q^j$	The output is proportional to $\sin(\varphi)$

Figure 4.3 shows the comparison of the outputs of above mentioned discriminators. It is clearly seen that when the phase error is 0 or ±180º, the discriminator outputs are zero which means they are insensitive to the phase shifts due to navigation data change [13].

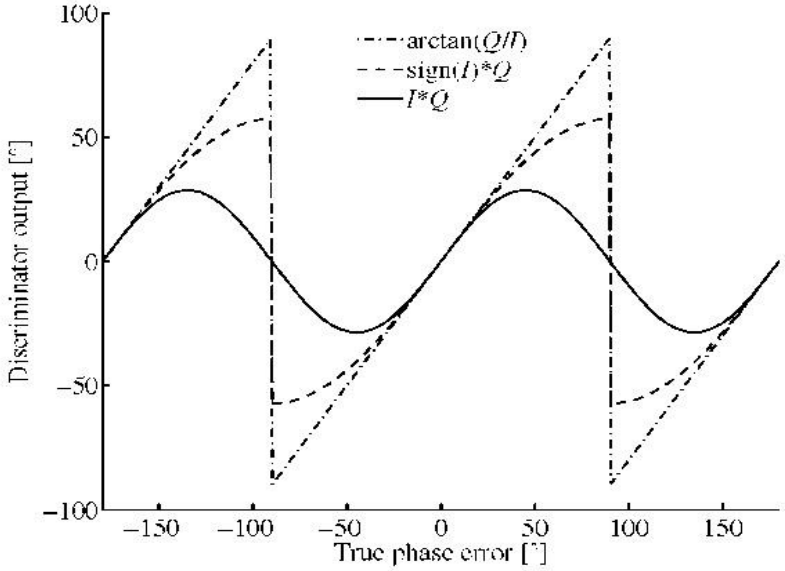

Figure 4.3 Costas loop Discriminator Outputs Comparison [11]

4.2 PRN Code Tracking

The aim of the PRN code tracking loop is to keep track of the code phase of a specific PRN code in the signal and generate a code signal that is the perfect replica of the PRN code of the incoming signal. In GPS receivers, generally, the code tracking loop is implemented as a delay lock loop (DLL) which is also called early-late tracking loop.

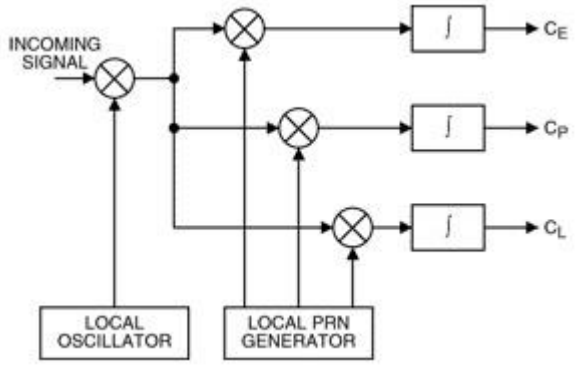

Figure 4.4 Basic DLL Block Diagram [18]

As shown in Figure 4.4, the DLL operates by generating three replicas of the received code, that are called as early, prompt and late of the code. The first step is to convert the incoming signal to baseband by multiplying it with a local replica of the carrier signal which is perfectly aligned. Then, the signal is multiplied with three codes with different code phases. The phase difference between the replicas is normally a half (±1/2) chip. The next phase is to integrate and dump the output signals. The numerical results indicate the correlation of the PRN code replica with the incoming signal [11]. Later on, the three correlation results are compared in order to find the highest correlation result.

Figure 4.5 shows a sample correlation result of a code tracking process. In Figure 4.5 the highest correlation belongs to the prompt code which means the phase is locked. Results with the highest correlation value at early or late code arms mean locally generated PRN code is early or late, respectively.

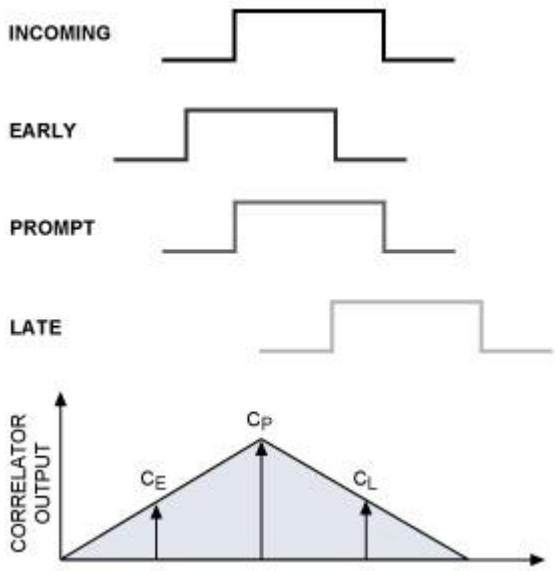

Figure 4.5 DLL Code Tracking, Correlator Outputs [18]

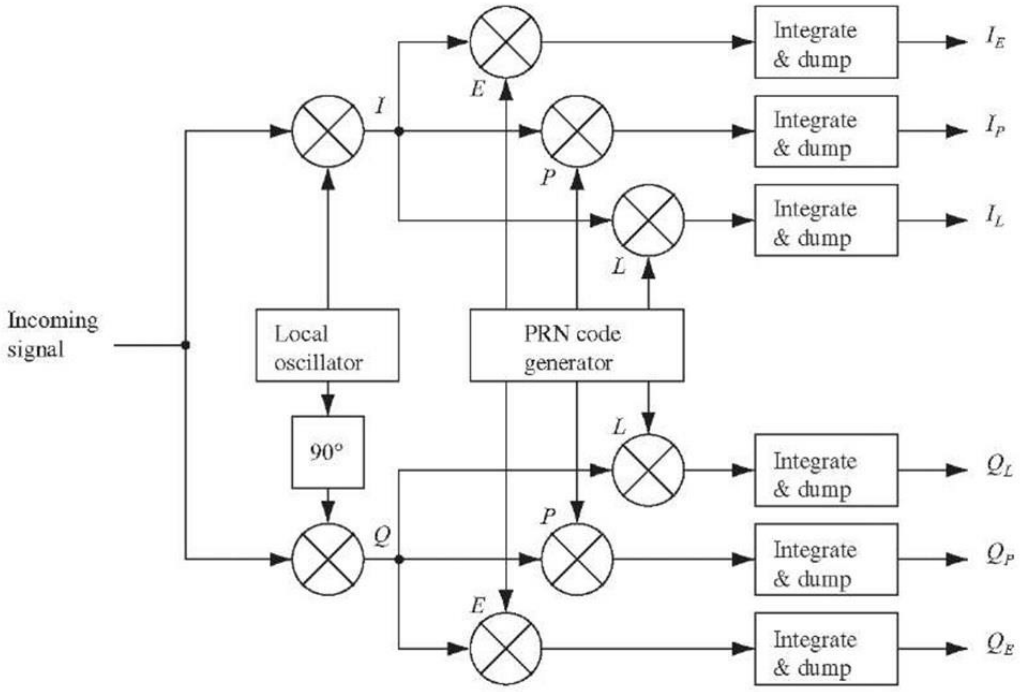

Figure 4.6 DLL with six correlators [11]

A DLL with 3 correlators as in Figure 4.4 operates well in case both carrier frequency and phase are locked almost perfectly. However, in practice, due to noise and user dynamics carrier phase error is inevitable and that may cause loss of DLL

lock. Therefore, in GPS receivers DLL's with 6 correlators are generally preferred as in Figure 4.6 [11].

The DLL shown in Figure 4.6 has the advantage of being independent of the carrier phase. If the local replica of the carrier wave is in-phase with the incoming signal, all energy will be on I arm. On the other hand, if there is a phase error between the input signal and the local replica of the carrier signal, I and Q arms will share the total energy. That information can be used to solve the problems caused by carrier phase error [13].

In order to be able to track the code, PRN code generator requires a feedback from the discriminators as in carrier frequency tracking. Therefore, the discriminator performance determines the DLL performance directly. There are various types of DLL discriminators and some of them are shown on Table 4.2 with their explanations.

Table 4.2 Discriminators for DLL [11]

TYPE	DISCRIMINATOR	DESCRIPTION
Coherent	I_E - I_L	The simplest type. Q arm is not required, but a good operating carrier frequency and phase tracking loop is crucial.
Non-coherent	$(I_E^2 + Q_E^2)$ - $(I_L^2 + Q_L^2)$	Early minus late power, inside $\pm\frac{1}{2}$ chip performance is almost the same with the coherent discriminator.
	$\frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)}$	Normalized early minus late power, this discriminator has the ability to keep track of the code in case the phase error is greater than $\pm\frac{1}{2}$ chip that helps to operate better with noisy signals.
	$I_P\,(I_E - I_L) + Q_P(Q_E - Q_L)$	Dot Product, it uses all correlation results.

The discriminators can be divided into two categories as coherent and non-coherent. Coherent discriminators require a well operating carrier frequency and phase lock loop, simply a PLL. Due to the fact that 3 correlators are enough for the operation of coherent discriminators, they are preferred in simple GPS receivers. On the other hand, non-coherent discriminators do not depend on the performance of carrier tracking loop since they use both I and Q arms. The disadvantage is that they are more complex and require a DLL with 6 correlators.

Figure 4.7 shows the output responses of the discriminators given in Table 4.2. As clearly seen on Figure 4.7, normalized early minus late power discriminator has better performance than the others since it is able to respond phase errors greater than $\pm\frac{1}{2}$ chip. In addition, since the discriminator is normalized, it can be used with signals with different SNR, different signal strengths and amplitudes. Another advantage is that it has the highest gain which increases with decreasing correlator spacing [19]. Therefore, with these advantages, normalized early minus late discriminator is chosen in the DLL implementation of this thesis.

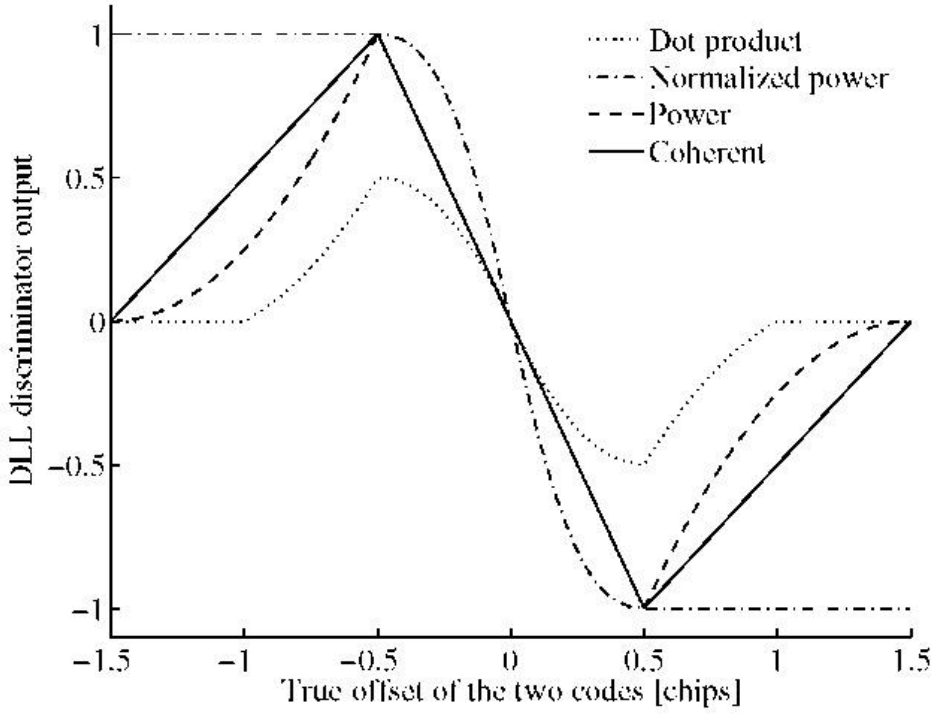

Figure 4.7 DLL Discriminator outputs [11]

4.3 The Complete Tracking Loop, PLL aided DLL

In this section, a combined method of PLL and DLL that minimizes multiplication operations and decreases the computation time is explained.

Figure 4.8 shows the combined block diagram of PLL and DLL. It is seen on the Figure 4.8 that the generated PRN code replica on the code tracking loop is used to wipe off the PRN code in the carrier tracking loop. Similarly, locally generated carrier signal replicas are used to wipe off the carrier signal in the code tracking loop. The block diagram consists of 11 multiplications that are the most time consuming processes of the entire loop. Reducing the number of multiplications can increase the processing rate of the loop [13].

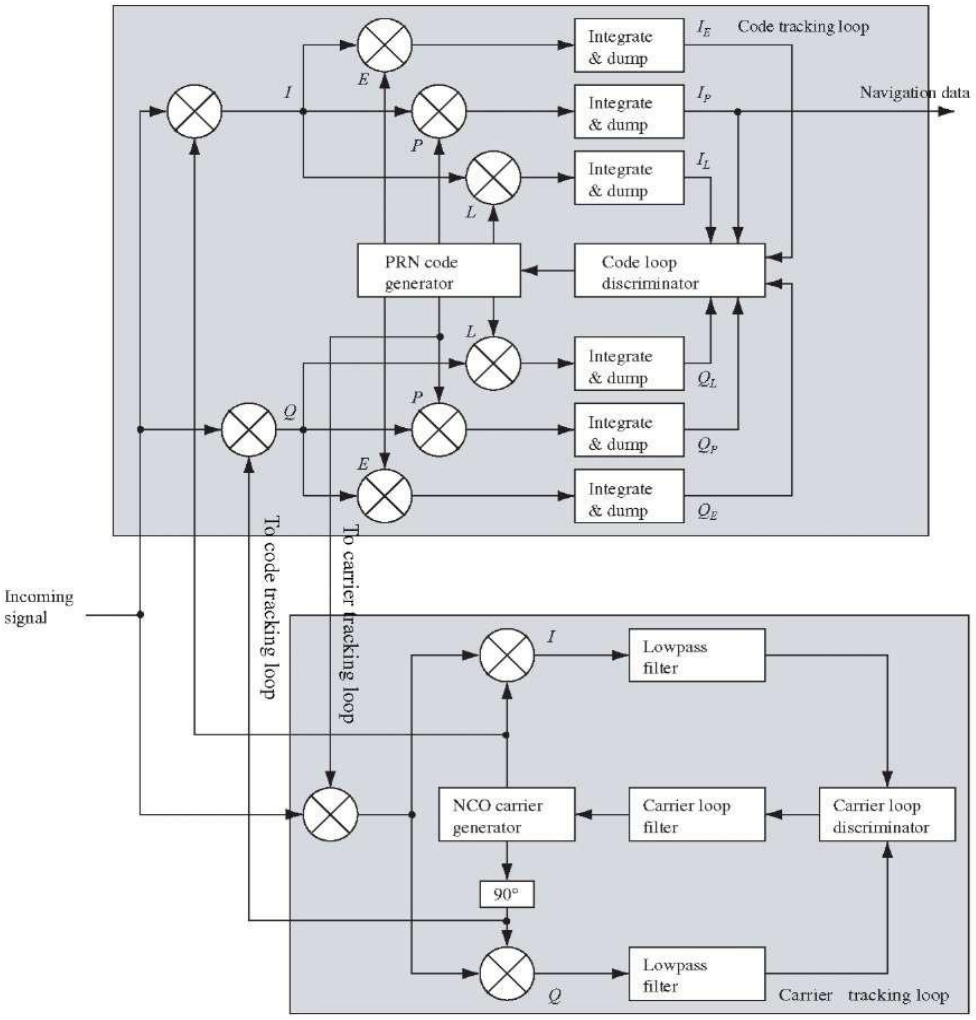

Figure 4.8 Combined PLL and DLL Block Diagram [11]

Figure 4.9 shows the complete block diagram of a tracking loop. It is the optimized version of the combined tracking loop shown on Figure 4.8. I and Q correlation outputs of the code tracking block are input to the carrier loop discriminator. Therefore, Costas loop is embedded in the DLL and three multiplications are eliminated. Computation time is reduced thanks to that optimization.

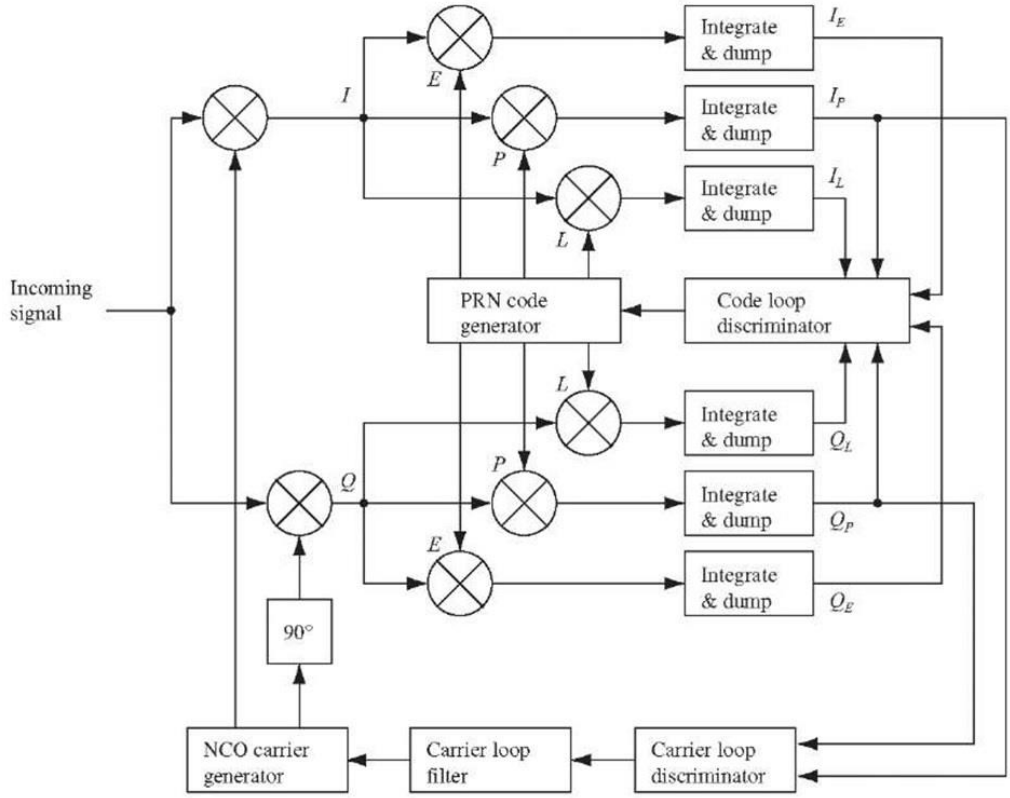

Figure 4.9 Block Diagram of Complete Tracking Loop [11]

4.4 Performance Effecting Parameters

Although there are many parameters affecting the performance of a scalar tracking loop, some of them have higher effect that the others. In this section a few of them is introduced shortly.

4.4.1 Discriminators

As mentioned before, there are many types of discriminators that can be used in a scalar tracking algorithm. They are the most important parameter of the tracking loop

since they affect directly the tracking results. The discriminator used in the carrier or code tracking loop defines the type of the loop such as PLL, FLL and DLL. The discriminators are selected due to application requirements. For instance, the receivers that are specialized for high user dynamics use different discriminators than standard receivers. Their computational burden is also an important factor as the simple receivers use simple discriminators [13]

4.4.2 Loop Filter

Another important factor affecting the loop performance is loop filters which are used to reduce the noise in the tracking loops to produce an accurate estimate of the error. Basically, the performance of a loop filter is defined by its order, damping ratio and noise bandwidth.

For the phase tracking loops, generally a third order filter is selected to track the receiver dynamics. On the other hand, a first order loop is enough for code tracking that is aided by carrier tracking loop.

Damping ratio determines the pace of the filter to reach its settle point and amount of the overshoot.

Finally, the noise bandwidth controls the noise level in the tracking loop. Estimated errors can be decreased by decreasing the noise bandwidth. On the other hand, increasing the noise bandwidth increases the ability for tracking of user dynamics. A third order loop has higher noise bandwidth which is suitable for high user dynamics [13].

4.4.3 Integration Time

Integration time selection is very important for tracking loops. It determines the update rate of the tracking loop. Increasing the integration time increases the information gathered but decreases the ability for tracking of user dynamics [13].

CHAPTER 5

VECTOR TRACKING LOOP DESIGN

In the previous chapter, the basics of the scalar tracking loop design are mentioned in a detailed way. In this chapter, the key features of vector tracking loops are introduced and differences and potential benefits over scalar tracking loops are explained in detail. In addition, implementation details of VTL are presented as the scope of this thesis.

5.1 Vector Tracking Architecture

Figure 5.1 shows the basic implementation of the vector tracking architecture. It is clearly seen in the figure that signal processing is done in an aggregate manner other than the cascaded structure of scalar tracking loop. Therefore, the aiding of the weaker signals by other stronger signals is possible.

The idea of the vector tracking depends on the basic nature of the GPS satellite constellation. According to the principle of GPS, the user dynamics as the position and velocity can be calculated using the code phase and the carrier frequency information of the received signal. Vector tracking algorithms use that simple idea reversely, that the code phase and carrier frequency of the signal can be calculated from the user's estimated position and velocity. By checking the difference between predicted and received signal, errors in each channel are calculated and then used to update the user position and velocity estimations.

As mentioned before, at least four satellites are required for user position calculations and these satellites must be acquired and tracked by a scalar loop in order to estimate the initial user dynamics sufficiently before initializing the VTL [1]. The navigation filter seen on Figure 5.1 generates corrections for the code and carrier NCOs and closes the tracking loop. Different from the scalar tracking loop architecture, either correlator or discriminator outputs are input to the navigation filter. Based on these input types, the VTL architecture can be analyzed under two main denominations as coherent and non-coherent architectures.

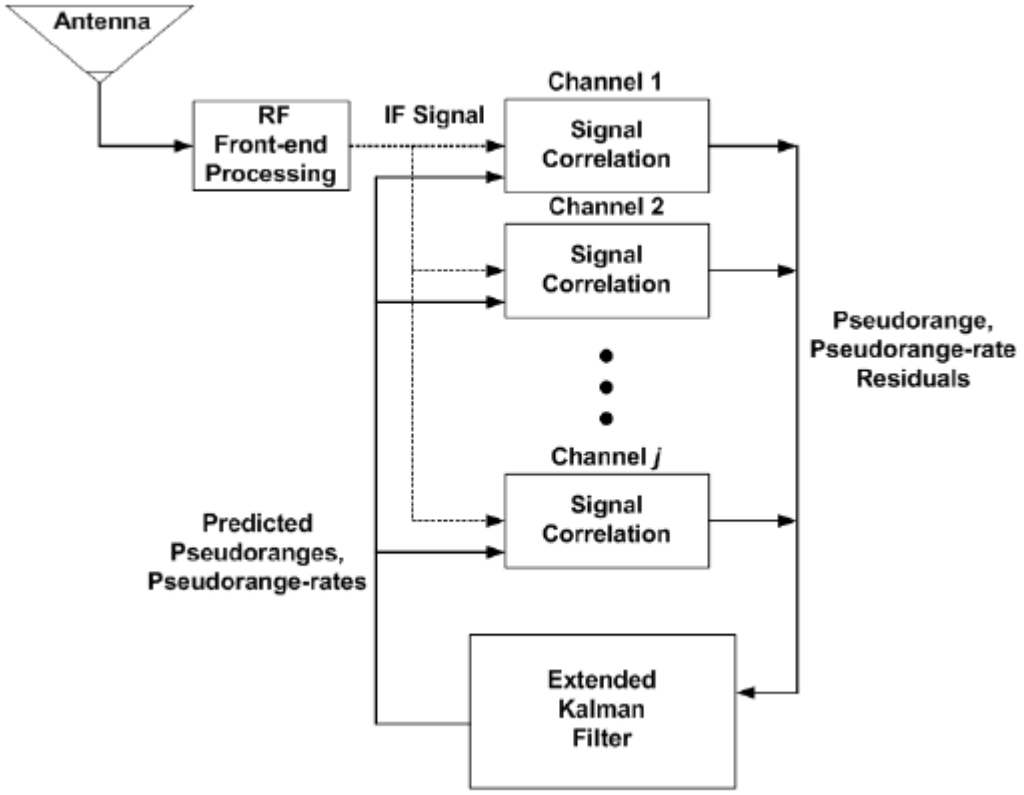

Figure 5.1 Basic Vector Tracking Architecture [23]

5.1.1 Coherent Architecture

As shown in Figure 5.2, the coherent architecture has two types of implementation as centralized coherent architecture and federated coherent architecture. In centralized coherent architecture, three correlator pairs (I_E, Q_E, I_P, Q_P, I_L, Q_L,) from each satellite

are input to the navigation filter. On the other hand, the three correlator pairs are filtered before the navigation filter that is used in the federated coherent architecture.

In the coherent architecture; Doppler frequency, carrier phase and code phase are tracked and code tracking depends on carrier phase tracking.

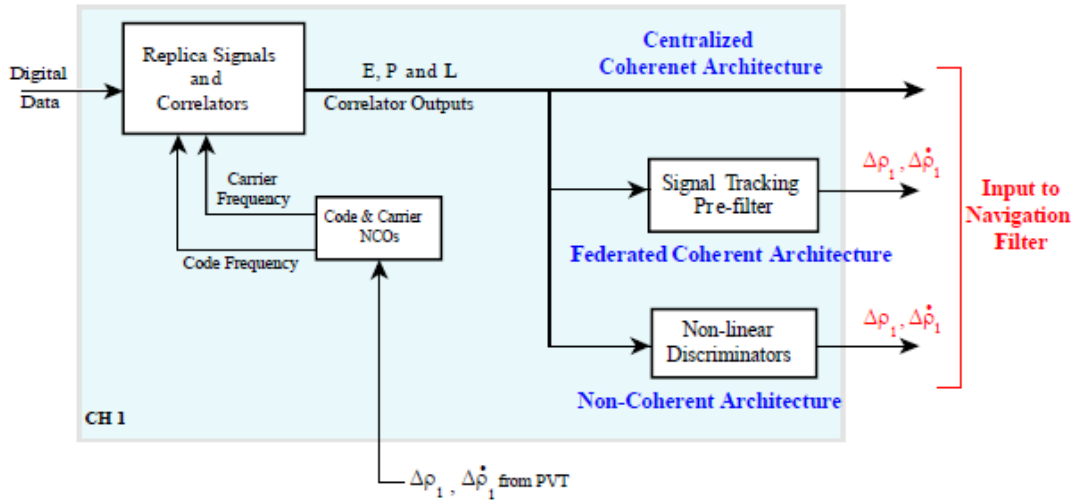

Figure 5.2 Coherent and Non-Coherent VTL [22]

As mentioned before, code phases and carrier frequencies of the acquired signals can be calculated using the receiver's relative position and velocity with respect to the corresponding satellite. The accuracy of any GPS receiver for user position estimation is not accurate enough to calculate carrier phases, but this does not prevent the implementation of a coherent VTL. Figure 5.3 shows a simple coherent VTL architecture block diagram. Due to tracking of carrier phases, coherent vector tracking algorithms are also called as Vector Phase Lock Loop (VPLL).

In Figure 5.3, there are two filters that one of them tracks the carrier phase and frequencies while the other one tracks the code phase. In order to convert the correlator outputs into usable state for the filter, the phase and frequency error discriminators are used. The filter tracking the code phase uses only pseudorange error discriminators.

It is possible to track the carrier/code frequency and phase with only a single filter, but it is more complicated.

The major advantages of coherent architecture are low measurement noise covariance and the absence of the un-modeled nonlinearities caused by discriminators. However, it consists of carrier phase tracking loops that needs more critical conditions than frequency discriminators and does not fit for low C/N_o applications. Therefore, coherent VTL architectures do not have definite advantages over scalar tracking loops and that make them out of the scope of this thesis.

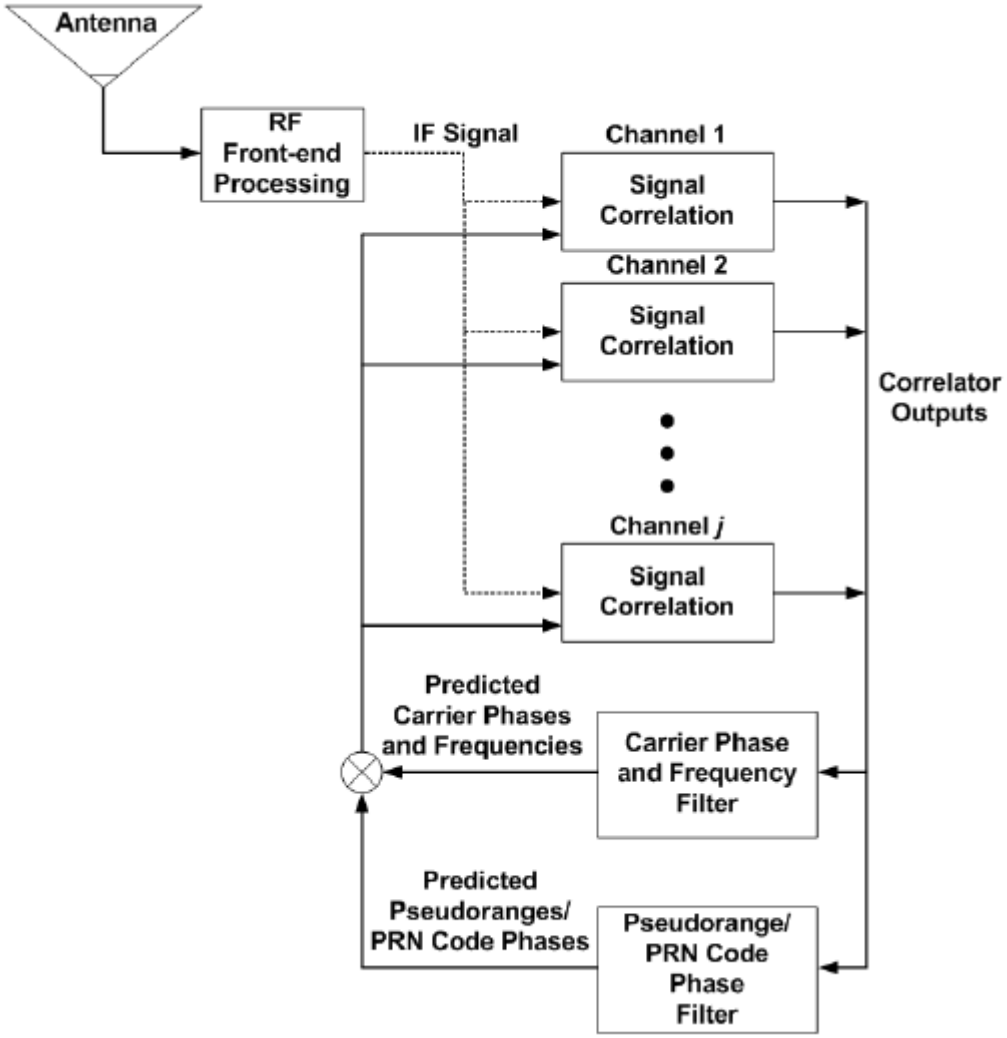

Figure 5.3 A Simple Coherent VTL Architecture [23]

5.1.2 Non-Coherent Architecture

As shown on Figure 5.2, non-coherent architectures depend on non-linear discriminators. Therefore, they are also called as discriminator based vector tracking loops. It should be noted that the implemented VTL in this thesis is also a non-coherent vector tracking architecture.

In the non-coherent architecture, code frequency depends on the Doppler frequency, and the carrier phase is not tracked as opposed to the coherent architecture due to high C/N_o requirement of carrier phase tracking loops.

The signal tracking and estimation of user dynamics are not separate processes as seen in Figure 5.1. An Extended Kalman Filter utilizes these processes simultaneously.

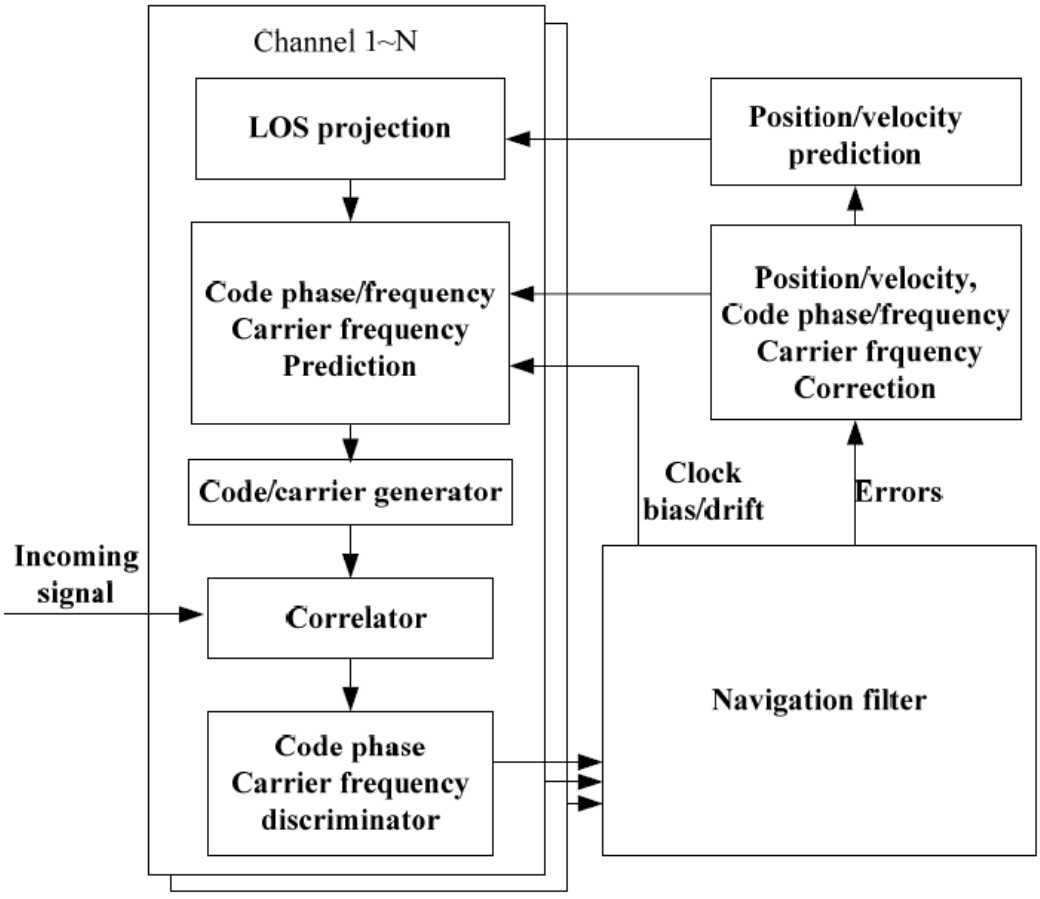

Figure 5.4 VDFLL Architecture Block Diagram [5]

Non-coherent vector tracking architectures have two common types: Vector Delay Lock Loop (VDLL) and Vector Delay/Frequency Lock Loop (VDFLL). In a VDLL, carrier frequency tracking is done by using scalar tracking loops individually in all channels as in the traditional receiver architectures. The navigation filter tracks only the code phases. On the other hand, a VDFLL uses the navigation Kalman Filter to track both carrier frequencies and code phases. The implemented algorithm in this thesis is also a VDFLL. Figure 5.4 shows the block diagram of the VDFLL architecture.

In the following sections, the mechanisms of non-coherent VTL or VDFLL are presented in details.

5.2 Vector Delay/Frequency Lock Loop Algorithm

This section describes the VDFLL that has been implemented in this thesis. In a software defined GPS receiver, local replicas of the incoming signal are used to correlate with a section of received signal in order to calculate the pseudoranges and pseudorange rates of each channel. As mentioned before, VTL algorithms require the initial knowledge of the user position, velocity and clock bias. Therefore, the receiver implemented in this study operates in the scalar mode with scalar DLL and PLL until sufficiently accurate navigation solution is obtained. With these initial knowledge, the VTL algorithm is started and predictions of user position and velocity at the next time epoch are generated using the equations (5.1) and (5.2) [5].

$$\hat{X}_{k+1} = X_k + t_{k,k+1} V_k \quad \textbf{(5.1)}$$

$$\hat{V}_{k+1} = V_k \quad \textbf{(5.2)}$$

where

k = specific time epoch,

V_k/X_k = the user velocity and position respectively,

$t_{k,k+1}$ = the time interval between k^{th} and $k+1^{th}$ epoch,

$\hat{X}_{k+1}$ = the estimation of user position at $k+1^{th}$ time epoch, topping hat on variables denote the estimation of corresponding variable.

The code phases and carrier frequencies for all acquired satellites can be calculated using these estimated user dynamics and satellite dynamics from the ephemeris at the time epoch *k+1* and the replicas of the PRN code and carrier frequency can be created locally. For this purpose, line of sight projections of the relative position and velocity between the user and corresponding satellite must be calculated using the equation (5.3).

$$a_{j,k} = \frac{X_{j,k} - X_k}{norm(X_{j,k} - X_k)} \tag{5.3}$$

where,

$a_{j,k}$ = unit line of sight vector between the user and j^{th} satellite at k^{th} time epoch

$X_{j,k}$ = the position of j^{th} satellite at k^{th} time epoch.

It should be noted that the LOS vectors may have some errors due to inaccuracy in the user position. However, they can be ignored since the distances between the user and satellites are too long.

The next part shown in Figure 5.4 is correlation of the incoming signals and locally generated replicas. The output of the selected discriminators gives the code phase and carrier frequency errors which contain the corresponding line of sight (LOS) projection of the differences between the true position/velocity of the user and estimated ones. Since the code phase error is related with the relative position, it can be written as follows [5]:

$$E_{code,k} = \hat{\varphi}_{j,k} - \varphi_{j,k} + \eta_{j,k} \tag{5.4}$$

$$E_{code,k} = t_{b,k} + \left(X_k - \hat{X}_k\right)^T a_{j,k} + \eta_{j,k} \tag{5.5}$$

where,

$\varphi_{j,k}$ = code phase of j^{th} satellite at k^{th} time epoch (in meter)

$t_{b,k}$ = clock bias of the receiver (in meter)

$\eta_{j,k}$ = noise term on the code of j^{th} satellite signal at k^{th} time epoch (in meter).

On the other hand, carrier frequency error is related with relative velocity and it can be written as shown in equations (5.6) and (5.7) [5].

$$E_{carrier,k} = \hat{f}_{j,k} - f_{j,k} + w_{j,k} \quad \textbf{(5.6)}$$

$$E_{carrier,k} = \Delta t_{d,k} + \left(V_k - \hat{V}_k\right)^T a_{j,k} + w_{j,k} \quad \textbf{(5.7)}$$

where,

$f_{j,k}$ = carrier frequency of j^{th} satellite at k^{th} time epoch (in m/s)

$\Delta t_{d,k}$ = the difference between the clock drifts at k^{th} time epoch and $k\text{-}1^{th}$ time epoch (in m/s)

$w_{j,k}$ = noise term on the carrier signal of j^{th} satellite signal at k^{th} time epoch (in m/s).

As can be understood from the equations shown above, the navigation solution could be obtained by using the user position and velocity errors estimations. Therefore, position and velocity errors are selected as the states of Extended Kalman Filter used as Navigation Filter. If it is assumed that the position at k^{th} time epoch is known, the position error at $k+1^{th}$ time epoch can be calculated by integrating the velocity error between kth and $k+1^{th}$ time epoch. In addition, if the integration time is selected short enough, the velocity error can be assumed as constant during integration. The equation (5.8) shows the model of the position error.

$$\delta X_{k+1} = \Delta t_{k+1} \delta V_k + n_{k+1} \quad \textbf{(5.8)}$$

where,

δX_{k+1} = the position error at time epoch k+1 (in meter)

Δt_{k+1} = duration between t_k and t_{k+1} (in second)

δV_k = the velocity error at time epoch k (in m/s)

n_{k+1} = the un-modelled error term in position which is considered as White Gaussian Noise (WGN) (in meter).

Similarly, the equation (5.9) shows the model of velocity error.

$$\delta V_{k+1} = v_{k+1} \tag{5.9}$$

where,

v_{k+1} = the un-modelled error term in velocity which is considered as White Gaussian Noise (WGN) (in m/s).

As mentioned above, integration time is selected short enough and velocity is assumed to be constant during integration which is selected as 1ms for the implementation in this thesis. Thus, the velocity error is modeled as a WGN sequence.

Based on the above equations and given the position and velocity of the receiver; code phase, code frequency and carrier frequency can be predicted at $k+1^{th}$ time epoch using the following equations respectively [5]:

$$\hat{\varphi}_{j,k+1} = \varphi_{j,k} + \left(\Delta X_{j,k+1} - \Delta t_{k+1} v_k\right)^T a_{j,k+1} + \Delta t_{k+1} c \tag{5.10}$$

$$\hat{f}_{code,j,k+1} = \left(1 + t_{d,k} + \left(V_{j,k} - v_k\right)^T a_{j,k+1}\right) f_{code} / c \tag{5.11}$$

$$\hat{f}_{j,k+1} = \left(1 + t_{d,k} + \left(V_{j,k} - v_k\right)^T a_{j,k+1}\right) f_n / c \tag{5.12}$$

where,

$\Delta X_{j,k+1}$ = the displacement vector of the j^{th} satellite between k^{th} and $k+1^{th}$ time epoch (in meter)

$t_{d,k}$ = clock drift at time epoch k (in m/s)

$V_{j,k}$ = velocity of the j^{th} satellite at time epoch k (in m/s)

f_{code} = the nominal code frequency, 1.023MHz for C/A code which is used in this study

f_n = the nominal carrier frequency, 1575.42 MHz for L_1 which is used in this thesis

c = the speed of light.

The clock bias caused by clock drift can be modeled as follows:

$$t_{b,k+1} = t_{b,k} + \Delta t_{k+1}\left(t_{d,k+1} - t_{d,k}\right) = t_{b,k} + \Delta t_{k+1}\Delta t_{d,k} \quad \textbf{(5.13)}$$

where the clock drift can be modeled as:

$$t_{d,k+1} = t_{d,k} + \Delta t_{d,k} \quad \textbf{(5.14)}$$

$$\Delta t_{d,k} = r_k \quad \textbf{(5.15)}$$

where,

r_k = a WGN sequence.

As can be seen on equation (5.15), the clock drift difference between k^{th} and $k+1^{th}$ time epoch is modeled as a WGN sequence due to the fact that the oscillators used in GPS receivers have relatively stable drift during short period of time such as the selected integration time in the implementation of the algorithm in this study.

Until this point, all sections and operations shown in Figure 5.4 are covered except the navigation filter which is used to estimate the mentioned errors of the user dynamics, and the code and frequency discriminators of which the output are employed as measurements into the navigation filter. In the next sections, the navigation filter and the selected discriminators are discussed in details.

5.2.1 Discriminators

Since the discriminator outputs are the inputs to the navigation filter, the performance of the selected discriminator affects the performance of VDFLL algorithm directly. Table 5.1 shows the selected discriminators for carrier frequency

tracking of VDFLL implementation of this study. The main difference between the scalar tracking loop algorithm discussed in Chapter 4 and VDFLL algorithm is that the scalar tracking loop algorithm tracks the carrier phase, but VDFLL tracks the carrier frequency or the Doppler shift. Therefore, the types of discriminators given in Table 4.1 and Table 5.1 are different.

Table 5.1 Discriminators Used for Carrier Frequency Tracking [13]

Discriminator	Description
$f_{err} = \frac{cross^{\,j}}{(t_2 - t_1)}$	Nearly optimal at low SNR. Low computational load. Dependent to squared signal amplitude.
$f_{err} = \frac{\tan^{-1}\left(\frac{cross^{\,j}}{dot^{\,j}}\right)}{2\pi(t_2 - t_1)}$	Four-quadrant arctangent. Optimal at high and low SNR. Very high computational load. No signal amplitude dependence.
$f_{err} = \frac{cross^{\,j} \times sign\ \left(dot^{\,j}\right)}{2\pi(t_2 - t_1)\left(I^2{}_{P2} + Q^2{}_{P2}\right)}$	Normalized decision directed frequency discriminator. Nearly optimal at high SNR. Moderate computational load.

where,,

$$cross = I_{P1}Q_{P2} - I_{P2}Q_{P1} \quad \textbf{(5.16)}$$

$$dot = I_{P1}I_{P2} + Q_{P2}Q_{P1} \quad \textbf{(5.17)}$$

I_{P1} and I_{P2} = represent the in-phase prompt correlation results over $k\text{-}1^{th}$ to k^{th} and k^{th} to $k+1^{th}$ time epochs, respectively.

Q_{P1} and Q_{P2} = represent the quadrature prompt correlation results over $k\text{-}1^{th}$ to k^{th} and k^{th} to $k+1^{th}$ time epochs, respectively.

t_1 and t_2 = denote the accumulated values over $k\text{-}1^{th}$ to k^{th} and k^{th} to $k+1^{th}$ time epochs, respectively.

Since, there is no Costas loop in the architecture, navigation data bit transition sensitivity must be taken into consideration while selecting the discriminators. All of the discriminators given in Table 5.1 are insensitive to the navigation data bit transition. As shown on Figure 5.5, the four-quadrant arctangent discriminator gives the optimal error output, however, it has very high computational load. Therefore, normalized decision directed discriminator is the one that is most commonly used as in this thesis.

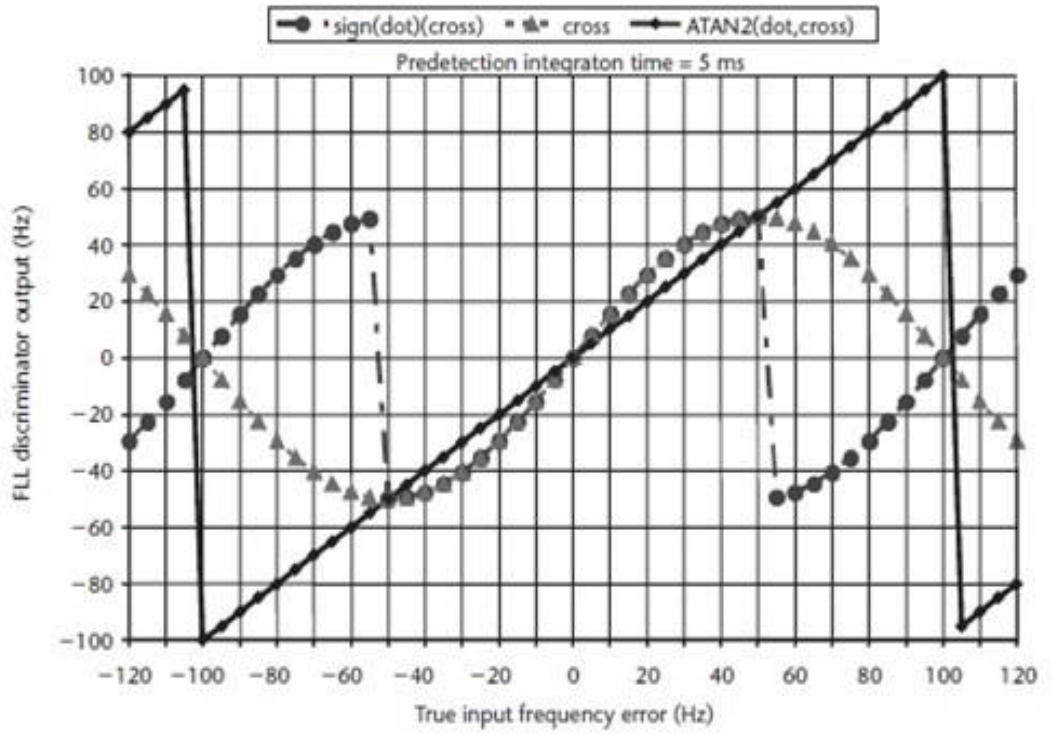

Figure 5.5 FLL Discriminator Outputs [13]

Table 5.2 shows the discriminators that are used to track code phases in VDFLL algorithm. Table 4.2 and Table 5.2 are almost the same except coherent discriminator part, since both of the algorithms track code phases. Figure 4.7 shows the output comparison of the discriminators given above. Normalized early minus late power discriminator is the most commonly used one due to the advantages mentioned in section 4.2. The outputs of the discriminators mentioned above are input to the navigation filter which is discussed in the next section in details.

Table 5.2 Discriminators Used for Code Phase Tracking [11]

DISCRIMINATOR	DESCRIPTION
$(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)$	Early minus late power, inside $\pm\frac{1}{2}$ chip performance is almost the same with the coherent discriminator. It has moderate computational load.
$\frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)}$	Normalized early minus late power, this discriminator has the ability to keep track of the code in case the phase error is greater than $\pm\frac{1}{2}$ chip that helps to operate better with noisy signals.
$I_P (I_E - I_L) + Q_P(Q_E - Q_L)$	Dot Product, it uses all correlation results.

5.2.2 Navigation Filter

The navigation filter shown on Figure 5.4 is an Extended Kalman Filter (EKF) that processes all information from all satellites to estimate the defined state vector. The central EKF can be designed in two different ways: the position-state formulation and pseudorange-state formulation. The position-state formulation is the most commonly used method that the filter states filter are the user's position, velocity and clock errors [23]. Higher order dynamics such as acceleration can also be used, but it is not critical if the integration time is short enough. On the other hand, the states of central filter in the formulation of the pseudorange-states are selected as the pseudorange rates and pseudoranges of the acquired satellites. The basic idea is same in both methods, but the definition of the state vectors of Extended Kalman Filters is different which requires changing all models given in equations above from (5.1) to (5.15). Therefore, in this study, the position-state formulation of the EKF is selected and implemented as can be clearly understood from the defined models.

The error states of the EKF are defined as follows:

$$X_E = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta v_x \\ \delta v_y \\ \delta v_z \\ t_b \\ \Delta t_d \end{bmatrix} = \begin{bmatrix} X\ axis\ Position\ Error \\ Y\ axis\ Position\ Error \\ Z\ axis\ Position\ Error \\ X\ axis\ Velocity\ Error \\ Y\ axis\ Velocity\ Error \\ Z\ axis\ Velocity\ Error \\ Clock\ Bias\ Error \\ Clock\ Drift\ Error \end{bmatrix} \quad \textbf{(5.18)}$$

In (5.18), the error states of the position and velocity are all in ECEF frame. The actual estimates of the user position and velocity are outside of the EKF and they are corrected at each measurement update using the estimated errors by the residuals of the pseudorange and pseudorange rate of each channel. After the update process, the state vector is reset to zeros.

The discrete process equation is shown in equation (5.19) [5].

$$X_{E,k+1} = \begin{bmatrix} \delta x_{k+1} \\ \delta y_{k+1} \\ \delta z_{k+1} \\ \delta v_{x,k+1} \\ \delta v_{y,k+1} \\ \delta v_{z,k+1} \\ t_{b,k+1} \\ \Delta t_{d,k+1} \end{bmatrix} = F_{k,k+1} \begin{bmatrix} \delta x_k \\ \delta y_k \\ \delta z_k \\ \delta v_{x,k} \\ \delta v_{y,k} \\ \delta v_{z,k} \\ t_{b,k} \\ \Delta t_{d,k} \end{bmatrix} + W_k \quad \textbf{(5.19)}$$

where the process update matrix is defined as follows:

$$F_{k,k+1} = \begin{bmatrix} 1 & 0 & 0 & \Delta t_{k+1} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t_{k+1} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t_{k+1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t_{k+1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(5.20)}$$

and W_k is the process noise matrix.

The inputs from the code phase and carrier frequency discriminators are accepted as measurements. Using the relationships between code phase error/user position error and carrier frequency error/user velocity error that are given in equation (5.5) and (5.7) respectively, the measurement equation can be written as follows [5]:

$$Z_k = HX_{E,k} + V_k = \begin{bmatrix} z_{code,1,k} & z_{carrier,1,k} & \cdots & z_{code,n,k} & z_{carrier,n,k} \end{bmatrix}^T_{1x2n} \quad \textbf{(5.21)}$$

where V_k is the measurement noise matrix and the H matrix is defined as follows:

$$H_k = \begin{bmatrix} a_{1,x,k} & a_{1,y,k} & a_{1,z,k} & 0 & 0 & 0 & 1 & 0 \\ a_{2,x,k} & a_{2,y,k} & a_{2,z,k} & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{j,x,k} & a_{j,y,k} & a_{j,z,k} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a_{1,x,k} & a_{1,y,k} & a_{1,z,k} & 0 & 1 \\ 0 & 0 & 0 & a_{2,x,k} & a_{2,y,k} & a_{2,z,k} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & a_{j,x,k} & a_{j,y,k} & a_{j,z,k} & 0 & 1 \end{bmatrix} \quad \textbf{(5.22)}$$

Therefore, the element of the measurement matrix can be written as:

$$z_{code,j,k} = a_{j,x,k}\delta x_k + a_{j,y,k}\delta y_k + a_{j,z,k}\delta z_k + \eta_{j,k} \quad \textbf{(5.23)}$$

$$z_{carrier,j,k} = a_{j,x,k}\delta v_{x,k} + a_{j,y,k}\delta v_{y,k} + a_{j,z,k}\delta v_{z,k} + w_{j,k} \quad \textbf{(5.24)}$$

After the error states are estimated by the EKF, user position/velocity, code phase, code frequency and carrier frequency, which are predicted using the initial values, are corrected using these error states as shown in following equations, respectively [5].

$$X_{k+1} = \hat{X}_{k+1} + \delta X_{k+1} \quad \textbf{(5.25)}$$

$$V_{k+1} = \hat{V}_{k+1} + \delta V_{k+1} \quad \textbf{(5.26)}$$

$$\varphi_{j,k+1} = \hat{\varphi}_{j,k+1} + \delta X^T_{k+1} a + \Delta t_{k+1} c + t_{b,k} \quad \textbf{(5.27)}$$

$$f_{code,j,k+1} = \hat{f}_{code,j,k+1} + \left(t_{d,k+1} + \delta v_{k+1} a\right) f_{code} / c \quad \textbf{(5.28)}$$

$$f_{j,k+1} = \hat{f}_{j,k+1} + \left(t_{d,k+1} + \delta v_{k+1} a\right) f_n / c \quad \textbf{(5.29)}$$

The Vector Delay/Frequency Lock Loop can be start over again by using these corrected values. The user position/velocity shown in equation (5.25) and (5.26) are the navigation output of the VDFLL algorithm.

5.2.3 Navigation Filter with Less State Parameters

The EKF given in section 5.2.2 is modeled with 8 states, which is the most common model used for VDFLL architecture. Number of states can be increased or decreased depending on the application area and hardware resources that the algorithm runs on. Increasing the number of states by adding the higher order derivatives of position as acceleration and jerk may increase the performance and accuracy of the algorithm. However, it significantly increases the computational load of the hardware. The VDLL algorithm requires only the states for the position and clock bias since only the code phases are tracked in vector form. Therefore, EKF can be modeled with only 4 states. On the other hand, VDFLL requires additional states for the velocity and clock drift in order to be able to track the carrier frequency in vector form and can be modeled with 8 states as in section 5.2.2. VDFLL algorithm is deep integration of VDLL and VFLL. However, in this study, the implemented VDFLL algorithm is tested with a navigation filter modeled with 5 states. The user position error estimation states are omitted in this model as shown in (5.30).

$$X_E = \begin{bmatrix} \delta v_x \\ \delta v_y \\ \delta v_z \\ t_b \\ \Delta t_d \end{bmatrix} = \begin{bmatrix} X\ axis\ Velocity\ Error \\ Y\ axis\ Velocity\ Error \\ Z\ axis\ Velocity\ Error \\ Clock\ Bias\ Error \\ Clock\ Drift\ Error \end{bmatrix} \quad \textbf{(5.30)}$$

Therefore, the discrete process equation is written as follows:

$$X_{E,k+1} = \begin{bmatrix} \delta v_{x,k+1} \\ \delta v_{y,k+1} \\ \delta v_{z,k+1} \\ t_{b,k+1} \\ \Delta t_{d,k+1} \end{bmatrix} = F_{k,k+1} \begin{bmatrix} \delta v_{x,k} \\ \delta v_{y,k} \\ \delta v_{z,k} \\ t_{b,k} \\ \Delta t_{d,k} \end{bmatrix} + W_k \quad \textbf{(5.31)}$$

where the process update matrix is defined as follows:

$$F_{k,k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t_{k+1} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(5.32)}$$

and W_k is the process noise matrix.

The inputs from carrier frequency and code phase discriminator are accepted as measurements and using the relationship between code phases/user position and carrier frequencies/user velocity, measurement equation is written as follows:

$$Z_k = HX_{E,k} + V_k = \begin{bmatrix} z_{code,1,k} & z_{carrier,1,k} & \cdots & z_{code,n,k} & z_{carrier,n,k} \end{bmatrix}^T_{1x2n} \quad \textbf{(5.33)}$$

where V_k is the measurement noise matrix and the H matrix is defined as follows:

$$H_k = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \\ a_{1,x,k} & a_{1,y,k} & a_{1,z,k} & 0 & 1 \\ a_{2,x,k} & a_{2,y,k} & a_{2,z,k} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{j,x,k} & a_{j,y,k} & a_{j,z,k} & 0 & 1 \end{bmatrix} \quad \textbf{(5.34)}$$

Therefore, the elements of the measurement matrix can be written as:

$$z_{code,j,k} = \eta_{j,k} \quad \textbf{(5.35)}$$

$$z_{carrier,j,k} = a_{j,x,k}\delta v_{x,k} + a_{j,y,k}\delta v_{y,k} + a_{j,z,k}\delta v_{z,k} + w_{j,k} \quad \textbf{(5.36)}$$

Additionally, the position error estimations are calculated as in equation (5.36).

$$\delta X_{k+1} = \Delta t_{k+1} \delta V_{k+1} \quad \textbf{(5.37)}$$

After the error states of the user velocity and clock bias/drift are estimated by using the EKF, user position/velocity, code phase, code frequency and carrier frequency which are predicted using the initial values are corrected using these error states by using the equations (5.25) to (5.29) given in section 5.2.2, respectively. The algorithm can start over again by using these corrected estimates.

This model may seem to be a Vector Frequency Lock Loop (VFLL), but unlike a VFLL algorithm, there is no scalar DLL in this algorithm and code phase is tracked indirectly by integrating the estimated velocity error to calculate estimated position error as shown in equation (5.36) and estimating the clock bias. Actually, the code phase is not even tracked and just calculated by using the results of carrier frequency tracking loop. This type of algorithm seems to be not preferred in the literature; however, the results shown on Chapter 6 are very promising with the advantage of significantly decreasing the computational load.

5.3 Tuning the Filter Parameters

The process and measurement noise matrices mentioned above are the most important parameters for determination of performance of a Kalman filter [5]. In this thesis, *Q* matrix is used to denote the indefiniteness of user dynamics and *R* matrix is used to denote the un-modeled noise in the discriminator outputs. In this section, initialization and tuning of these two matrices are explained briefly.

Covariance matrix of the process noise W_k can be written as follows:

$$Q_k = \begin{bmatrix} Q_{dynamic} & 0 \\ 0 & Q_{clock} \end{bmatrix} \quad \textbf{(5.38)}$$

where,

$Q_{dynamic}$ = user dynamics part of the covariance matrix

Q_{clock} = clock errors part of the covariance matrix.

They can be written as follows:

$$Q_{dynamic} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_x}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{v_z}^2 \end{bmatrix} \quad \textbf{(5.39)}$$

$$Q_{clock} = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix} \quad \textbf{(5.40)}$$

where,

Q_{11} and Q_{22} = clock bias and drift noise error, respectively.

Clock drift error is modeled as a WGN sequence in equation (5.15) due to stability of the oscillators used in GPS receiver. Therefore, clock bias and drift noise errors can be set by using the variance of clock drift calculated by scalar tracking loop for the initialization of vector tracking loop [5].

User dynamics part of the covariance matrix is determined based on some priori knowledge of the user velocity and acceleration. Therefore, it directly depends on the usage area of the receiver. For an application area with average movements, the order of 10 and 1 are acceptable for positional variances and velocity variances, respectively. However, these values must be adjusted carefully in different application areas by taking into consideration that increasing the $Q_{dynamic}$ widens the loop bandwidth for user dynamics but on the other hand, decreases the accuracy [5].

The covariance matrix of the measurement noise matrix V_k is defined as in (5.40).

$$R_k = \begin{bmatrix} R_{code,1} & 0 & 0 & 0 & 0 \\ 0 & R_{carrier,1} & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & R_{code,n} & 0 \\ 0 & 0 & 0 & 0 & R_{carrier,n} \end{bmatrix}_{2n \times 2n} \quad \textbf{(5.41)}$$

where,

$R_{code,n}$ and $R_{carrier,n}$ = code phase and carrier frequency error terms, respectively.

R_k is diagonal due to the assumption that the outputs of different discriminators are independent. Carrier and code phase error terms can be calculated by using the variances of certain time length of discriminator outputs. Better estimates of true variances can be calculated by using longer time length. On the other hand, using longer time length to get variances narrows the loop bandwidth for the changes of carrier frequency and code phase. For the implementation of VDFLL in this thesis, time length is selected as 100ms which is reasonable for the scenarios with moderate dynamics.

5.4 Summary

In this chapter, some basic information about VTL algorithms is given as coherent and non-coherent VTL architectures. VDFLL as non-coherent architecture is discussed by presenting the implementation details since it is the algorithm that implemented in this thesis. Selected discriminators that are used in the VDFLL are shown and explained. In addition, the details of the navigation filter of VDFLL are presented and a navigation filter with less process states is also proposed. Performance comparison of vector tracking loops implemented by using these two navigation filters and scalar tracking loop mentioned in Chapter 4 are presented in the following chapter. The VDFLL is able to perform in degraded signal environments and can tolerate the momentarily signal blockages by aiding the weaker signals in the presence of other relatively strong signals. It is shown that VDFLL operates better at lower C/N_o ratio and during the satellite blockage than traditional tracking loop mentioned in Chapter 4.

CHAPTER 6

SIMULATION RESULTS

In this chapter, performance comparison of the scalar and vector tracking algorithms, which are explained in Chapter 4 and Chapter 5 respectively, are investigated. The advantages of vector tracking loops during short satellite blockage periods and operation abilities under lower C/N_o ratio levels are shown.

6.1 Signal Acquisition Setup

The scalar tracking loop and vector tracking loop algorithms mentioned above were implemented in MATLAB environment. The open source software receiver given in [11] is used as the base of both scalar and vector tracking algorithms.

In order to be able to process the GPS signals by a software defined receiver implemented in MATLAB environment, the signals must be down-converted to an intermediate frequency and digitized. As mentioned in section 3.2, an RF-Front End is used for this purpose. "*SiGE GN3Sv2 GPS Sampler"* is the RF-Front End used in this thesis. It digitizes the GPS signal as 1-bit in-phase (I) and 1-bit quadrature (Q) with the help of the IC named SE4120L. A suitable GPS antenna is required to pick up the signals. The digitized data are transferred to the computer running MATLAB in order to process it. Figure 6.1 shows the signal acquisition setup and Table 6.1 shows the signal sampling characteristics of this setup.

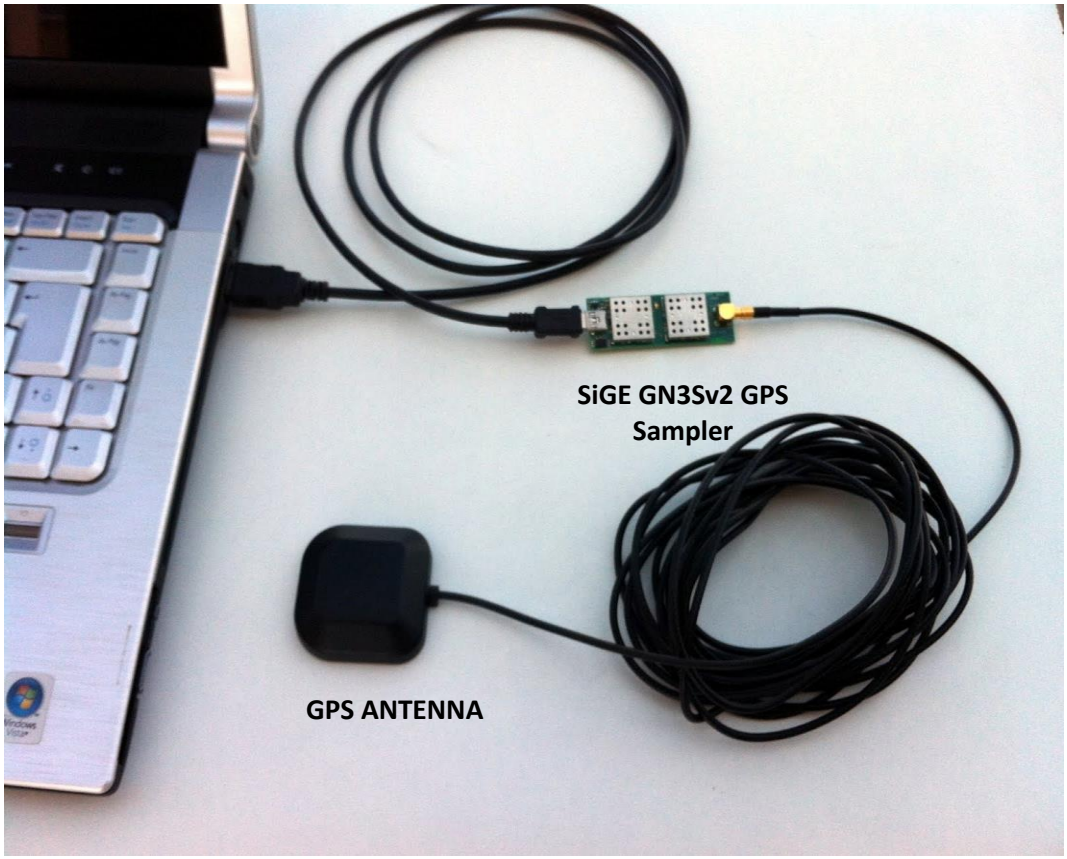

Figure 6.1 GPS Signal Acquisition Setup

Table 6.1 Signal Sampling Characteristics of Front-End [20]

Name	Value
IF	4.092 MHz
Sampling Rate	8.184 MHz
Data Formation	1 bit I / 1 bit Q
Forward Gain	18.5 dB
Noise Figure	1.7 dB

SiGE GN3Sv2 GPS Sampler has the ability to pick up GPS and Galileo signals broadcasted at 1575.42 MHz center frequency. It can be adjusted to pick up only GPS L_1 signal or both GPS and Galileo signals by changing the bandwidth of the bandpass filter to 2.2MHz or 4.4MHz respectively [20]. In this thesis, it is adjusted to pick up GPS signal only, since the Galileo signal is not in the scope. Figure 6.2 shows the frequency spectrum of GPS L_1 and Galileo signals.

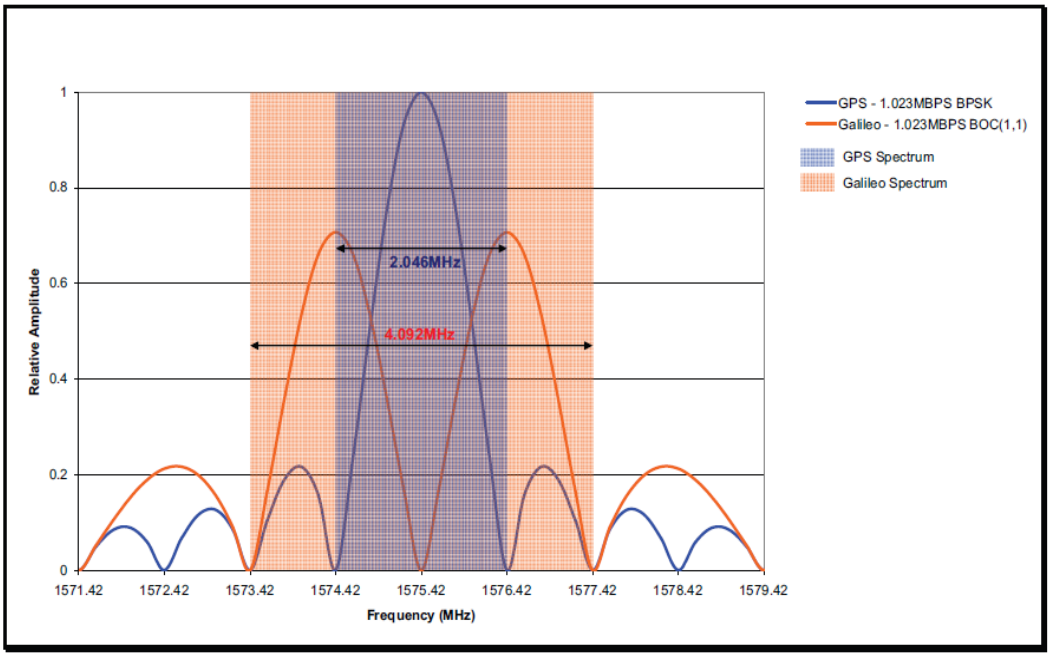

Figure 6.2 Frequency Spectrum of GPS L_1 and Galileo Signals

The signal sampler is able to record GPS signals up to 40 seconds which is enough to extract navigation data and calculate the user position. However, as mentioned in Chapter 5, vector tracking algorithm requires some initial values including code phase and frequency, carrier frequency from each channel, user position and velocity and signal transmit time corresponding to the starting point of the VTL. Therefore, these values should be obtained by the scalar tracking loop before the initialization of VTL which causes the requirement of longer periods of GPS signal records to test the algorithm sufficiently. Due to data record limitations of the signal sampler, using real data records is not suitable to test behavior of VTL algorithm. Thus, simulated data were used, which were generated by the Spirent GSS7700 simulator [21], that

gives ability to know the required initialization values before processing the data by the VTL algorithm.

6.2 Simulation Scenario

In the simulated scenario, a vehicle is moving in a simple circular trajectory as shown in Figure 6.3, with 15 m/sec constant velocity in clockwise direction. The radius of the circle is 100 meters and the altitude is adjusted as 1000 meters. In reality, the given simulated trajectory is located on some suburban areas of Ankara. The latitude and longitude values of the start point shown on Figure 6.3 are 40° N and 33° E, respectively. The simulated vehicle may be assumed as a drone flying in circles.

Figure 6.3 Trajectory of the Simulated Vehicle

In the simulated scenario, there are ten visible satellites with PRN numbers 1, 2, 4, 7, 8, 10, 13, 23, 25, 27 as shown on Figure 6.4. All the satellite signals have almost the same nominal C/No level which is about 45 dB-Hz.

There are three different datasets generated by Spirent GPS simulator with same mentioned trajectory and sky plot. The differences between them are explained in Table 6.2.

Table 6.2 Simulated Datasets

Simulated Data	Description
Dataset 1	All satellite signal powers remain constant at about 45dB-Hz until 7th second, and then all begin dropping at -1dB-Hz/sec.
Dataset 2	All satellite signal powers remain constant at about 45dB-Hz until 12th second, and then only the satellites with PRN number 1, 2, 4, 13, 25 begin dropping at -1dB-Hz/sec.
Dataset 3	All satellite signal powers remain constant at about 45dB-Hz until 12th second, and then at 12th second the signals coming from PRN 2 and PRN 23 are blocked for 10 seconds. After that the signals return to their previous levels.

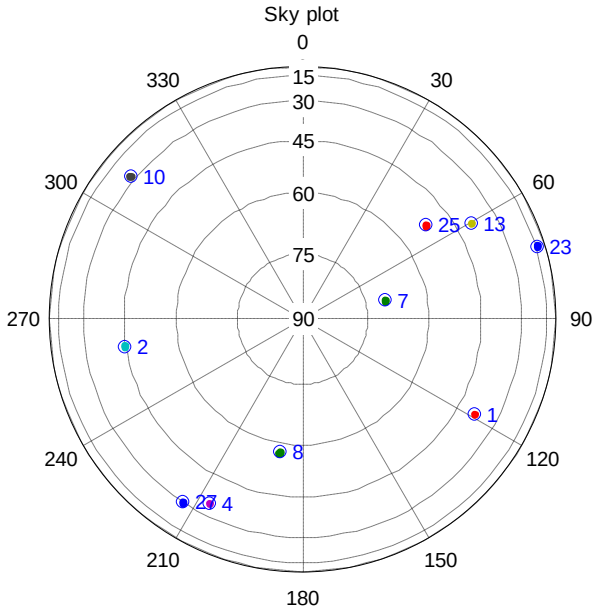

Figure 6.4 Sky Plot Showing the Visible Satellites for All Datasets

In the following section, performance comparisons of scalar and vector tracking loops, which are discussed in Chapter 4 and Chapter 5 respectively, are presented.

The general settings of the vector tracking loops are given in Table 6.3 and the general settings of the scalar tracking loop are given in Table 6.4. Same code phase discriminator is used in both scalar and vector tracking loops for a proper comparison. Carrier tracking discriminators are different since the scalar tracking loop tracks both carrier phase and frequency by using a Costas loop, but vector tracking loop just tracks the carrier frequency.

Table 6.3 Settings of Vector Tracking Loops for All Datasets

Process Noise Covariance Matrix (Q)	Position Error (Variance) (m^2)	20
	Velocity Error (Variance) (m^2/s^2)	3
	Clock Bias (Variance) (m^2)	1e-7
	Clock Drift (Variance) (m^2/s^2)	0.1
Measurement Noise Covariance Matrix (R)	Code Phase Error (Variance) (m^2)	Variance of 100ms discriminator outputs
	Carrier Frequency Error (Variance) (m^2/s^2)	Variance of 100ms discriminator outputs
Code Phase Discriminator		$\frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)}$
Carrier Frequency Discriminator		$\frac{cross \times sign\ (dot)}{2\pi(t_2 - t_1)(I^2{}_{P2} + Q^2{}_{P2})}$

Table 6.4 Settings of Scalar Tracking Loops for All Datasets

Code/Carrier pre-detection time (ms)	1
DLL/FLL/PLL Damping Ratio	0.7
DLL Noise Bandwidth (Hz)	2
DLL Correlator Spacing (chips)	0.5
PLL Noise Bandwidth (Hz)	25
PLL Noise Bandwidth (Hz)	10
Code Phase Discriminator	$\frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)}$
Carrier Phase Discriminator	$\tan^{-1}\left(Q_p / I_p\right)$

6.3 Performance Comparison for the Scalar and Vector Tracking Loops

In this section, performance comparison of scalar tracking loop, VDFLL and VDFLL with navigation filter having less state parameters are given. From this point, VDFLL with navigation filter having less state parameters is called VDFLL-M, meaning modified VDFLL. The three different simulated datasets are explained in Table 6.2. Simulated dataset 1 is used to compare the performances of tracking loops under low C/No ratios. Decreasing the power levels of all visible satellites with a constant predefined ratio gives the ability to see and compare minimum power requirements of all tracking loops. Similarly, simulated dataset 2 is used to compare the performances of tracking loops in the presence of weak and strong satellite signals.

Better performances are expected from the vector tracking loops due to their ability to aid weaker signals in the presence of relatively stronger signals. Finally, simulated dataset 3 is used to assess the ability to tolerate short time of satellite blockages of vector tracking algorithms. All the results presented in the following sections are the average of 20 Monte-Carlo runs.

6.3.1 Results of Simulated Dataset 1

As mentioned in Table 6.2, C/No ratios of the signals in this data set remain constant at about 45 dB-Hz until 7th second and start to decrease at -1 dB-Hz /sec.

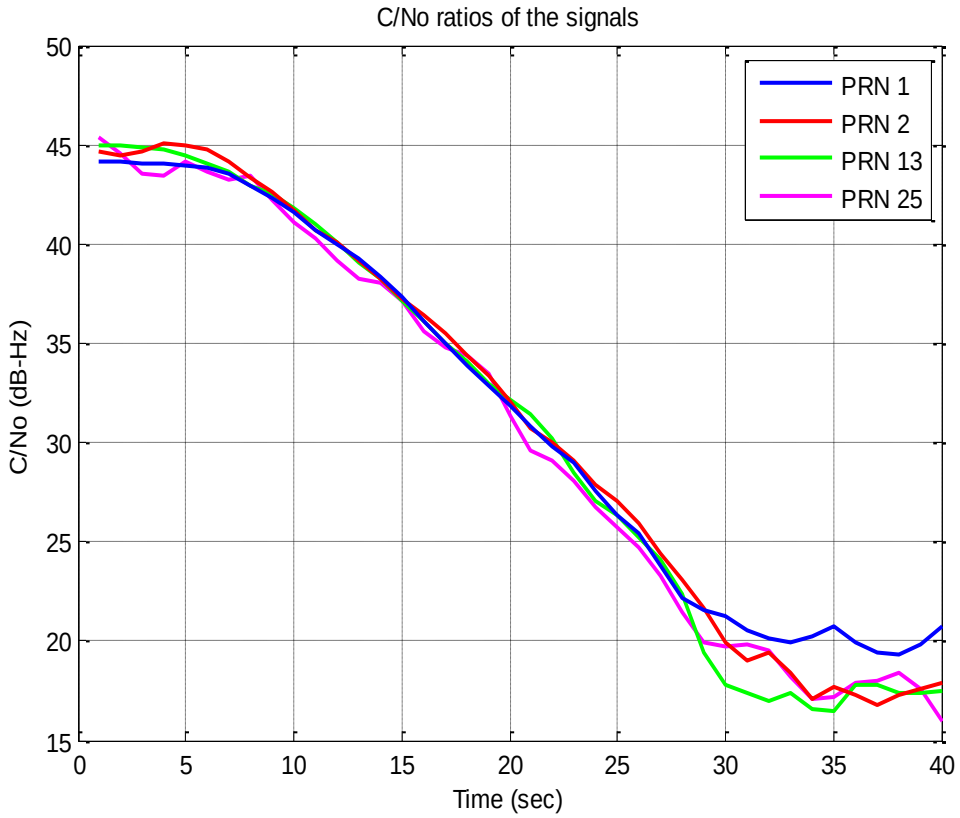

Figure 6.5 C/No Ratio Estimations of Satellites

C/No ratios of some of the selected satellites are shown on Figure 6.5 which verifies the settings of the simulation scenario. Only four of the satellites are shown on Figure 6.5 since the others have similar power characteristics as defined in the simulation. C/No ratio estimations are done by using the power ratio method (PRM)

which is explained in Appendix B. In this thesis, both power ratio method (PRM) and variance summing method (VSM) are implemented, but PRM is generally preferred for the C/No estimations. The entire C/No ratio estimation results shown in this study are calculated by using the PRM.

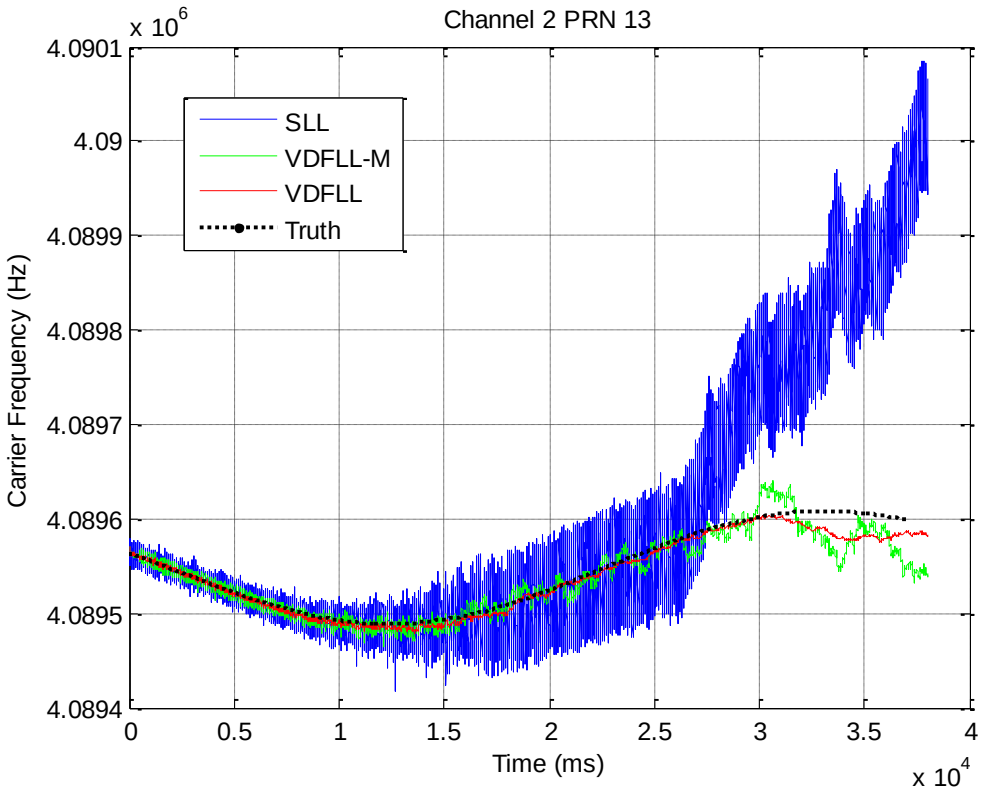

Figure 6.6 Carrier Frequency Tracking Results of PRN13

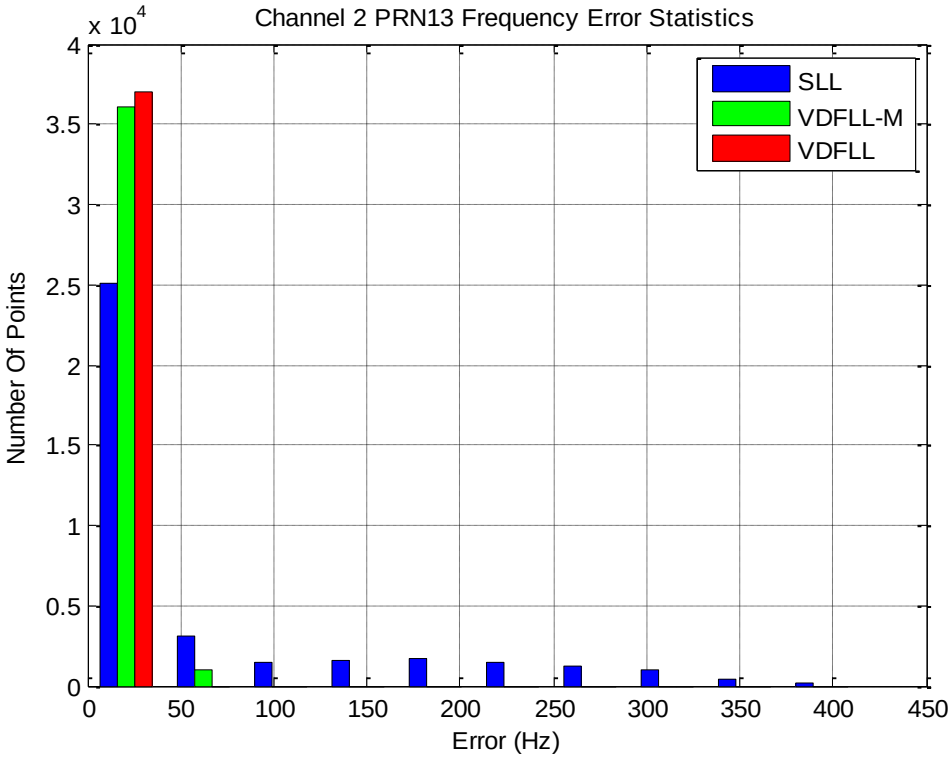

Figure 6.7 Carrier Frequency Error Statistics of PRN13

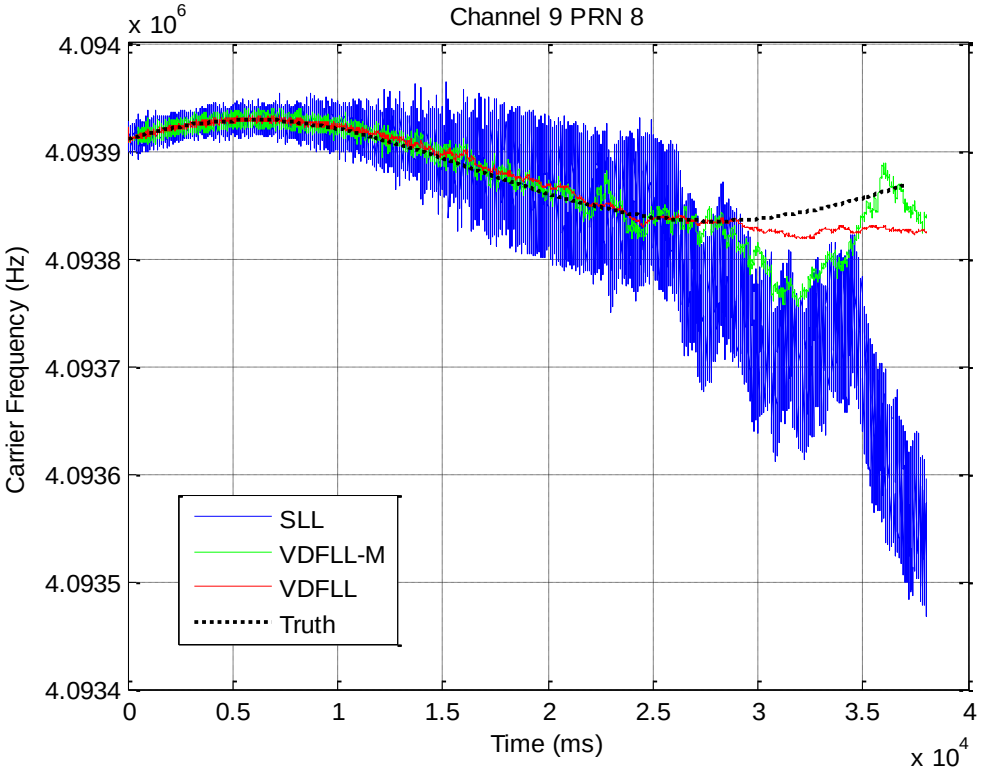

Figure 6.8 Carrier Frequency Tracking Results of PRN8

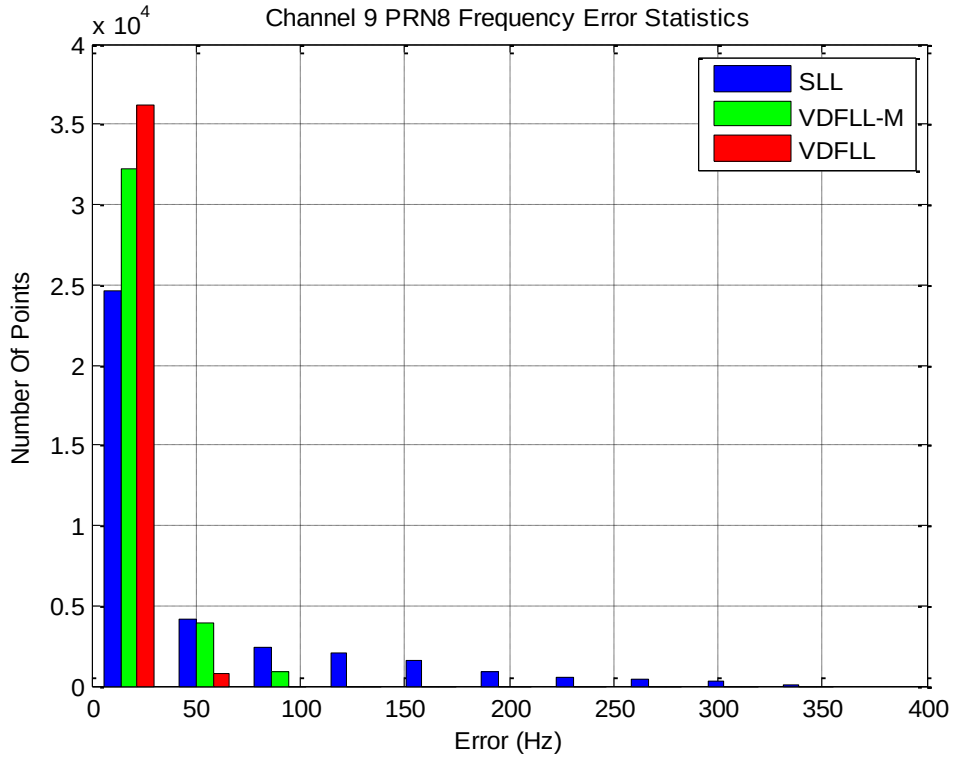

Figure 6.9 Carrier Frequency Error Statistics of PRN8

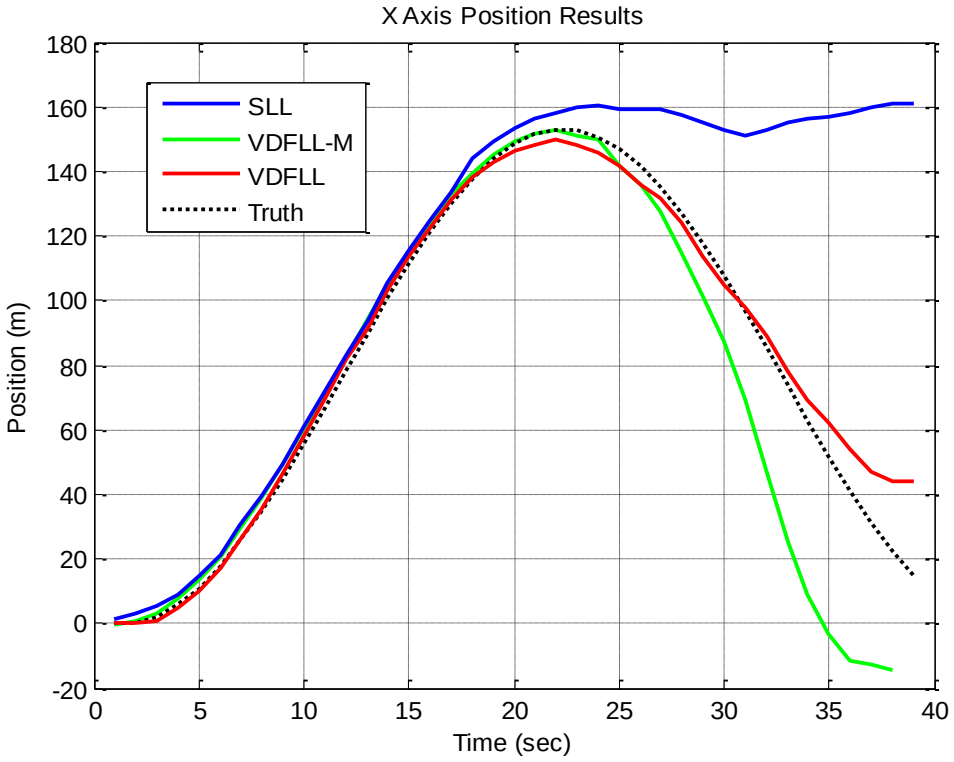

Figure 6.10 X Axis Position Tracking Results

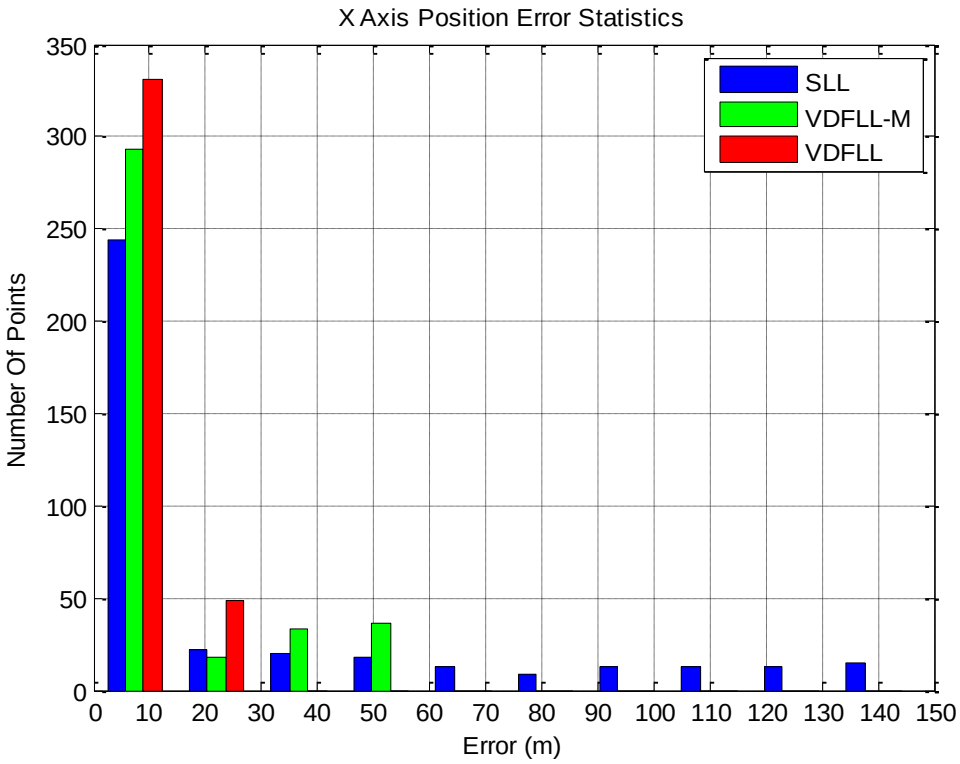

Figure 6.11 X Axis Position Error Statistics

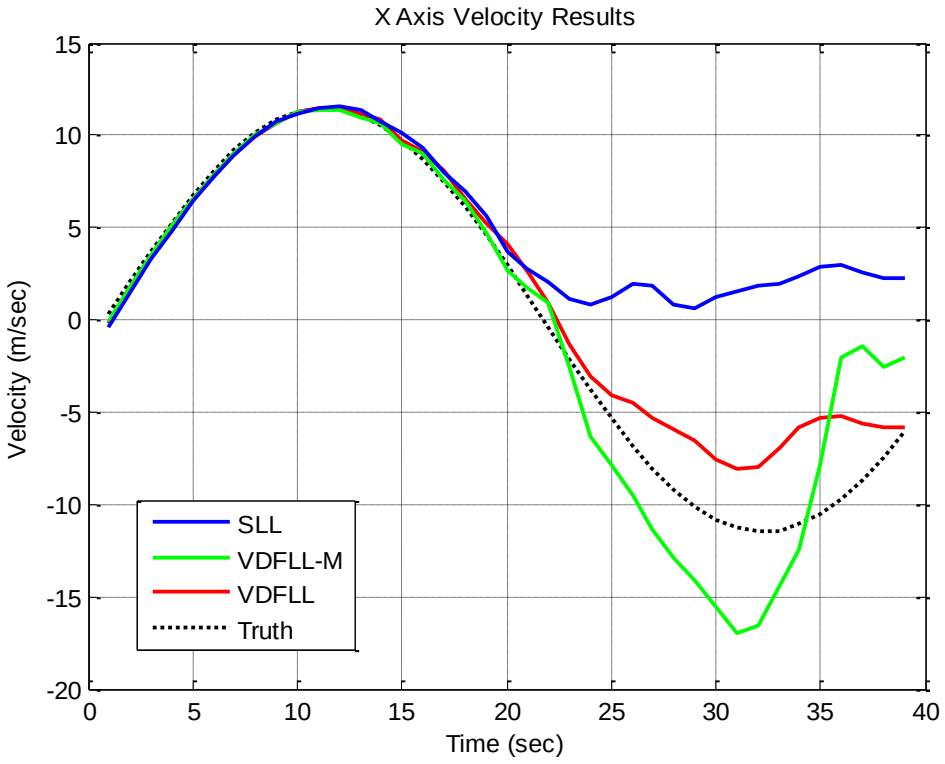

Figure 6.12 X Axis Velocity Tracking Results

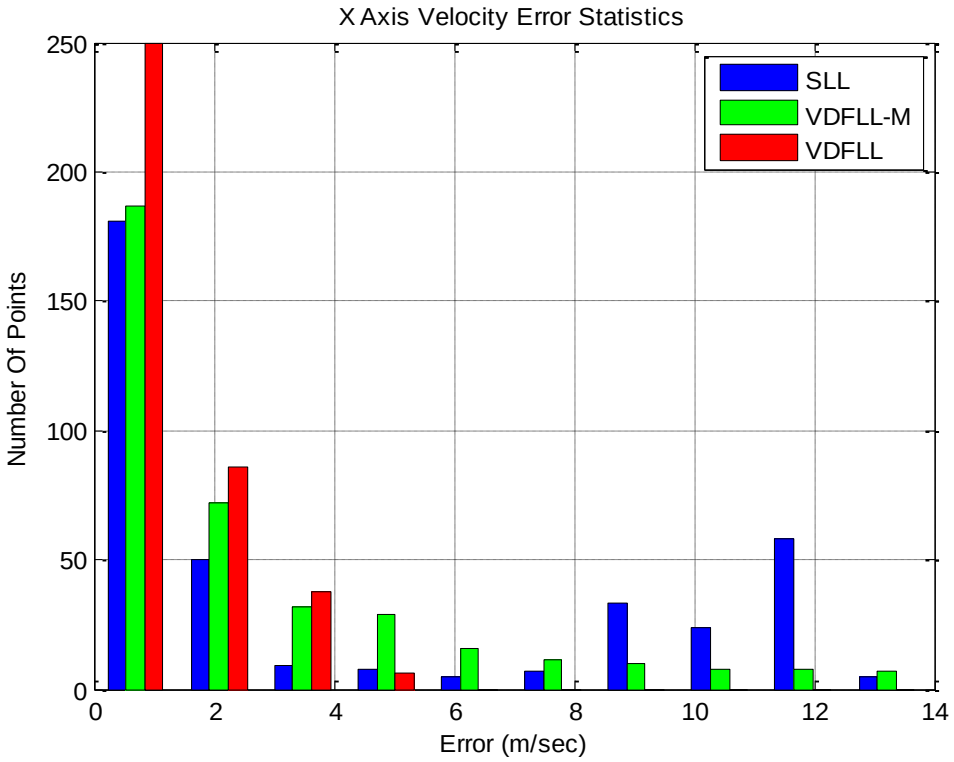

Figure 6.13 X Axis Velocity Error Statistics

The results of SLL, VDFLL and VDFLL-M are demonstrated in above figures in order to verify the superiority of vector tracking loops' performance over SLL in low C/No environment. The frequency results of only two channels (PRN8 and PRN13) are shown since the other channels' results are similar. From Figure 6.6 and Figure 6.8, it can be easily seen that the SLL loses lock at about 26th second, VDFLL-M at about 28th second and VDFLL at about 29th second. This illustrates that vector tracking loops can maintain frequency lock longer than scalar tracking loop. In this experiment, the gain of VDFLL is about 3 dB-Hz and the gain of VDFLL-M is about 2 dB-Hz over SLL with the -1 dB-Hz/sec signal attenuation rate. The frequency error statistics given in Figure 6.7 and Figure 6.9 support the results mentioned above. The X axis navigation solutions are also given in Figure 6.10 and Figure 6.12. It is easily seen that best solution is given by VDFLL and the worst solution is given by SLL. Vector tracking loops' accuracy of position and velocity solutions are improved significantly. The Y and Z axes are not shown here, since the results are similar.

6.3.2 Results of Simulated Dataset 2

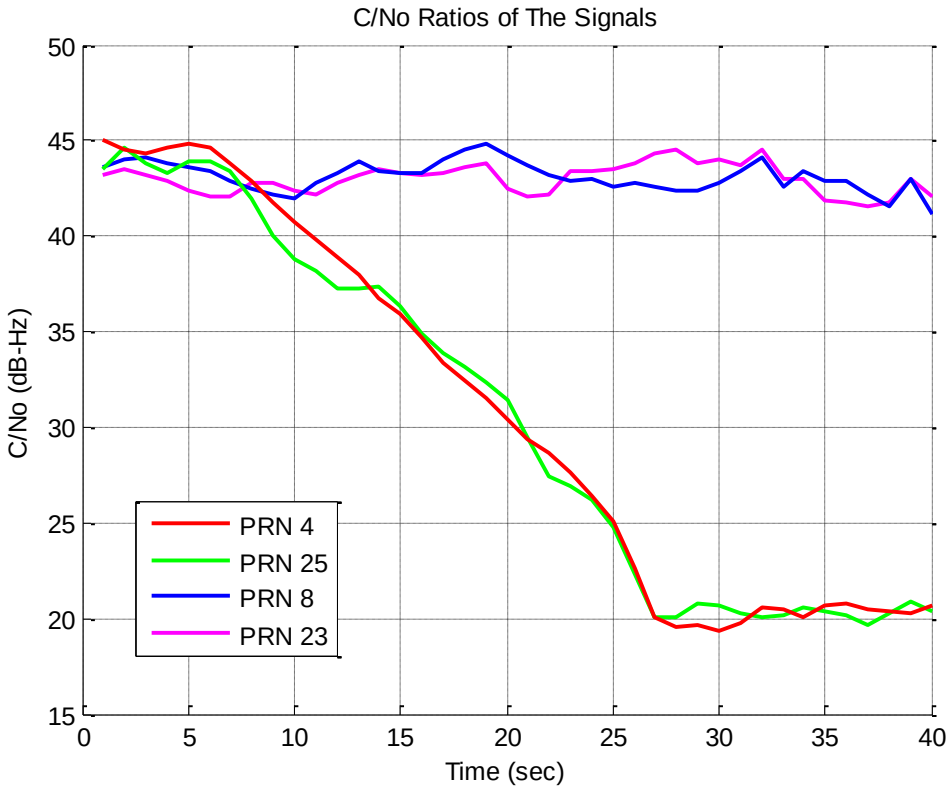

Figure 6.14 C/No Ratio Estimations of Satellites

As mentioned in Table 6.2, C/No ratios of the signals in this data set remain constant at about 45 dB-Hz until 7th second and then only the satellites with PRN number 1, 2, 4, 13, 25 begin dropping at -1dB-Hz/sec as shown on Figure 6.14 which verifies the settings of the scenario. In this scenario, combination of 4 satellites (PRN4, PRN25, PRN8, and PRN23) is used to get the navigation solutions for SLL, VDFLL and VDFLL-M. As mentioned before, according to the GPS operation principle, at least 4 satellites are required to calculate the navigation solutions. Therefore, when the signal power of two satellites becomes too weak, the number of visible satellites decreases to 2 which are not enough to get the navigation solutions for SLL. However, improved results are expected from the vector tracking loops.

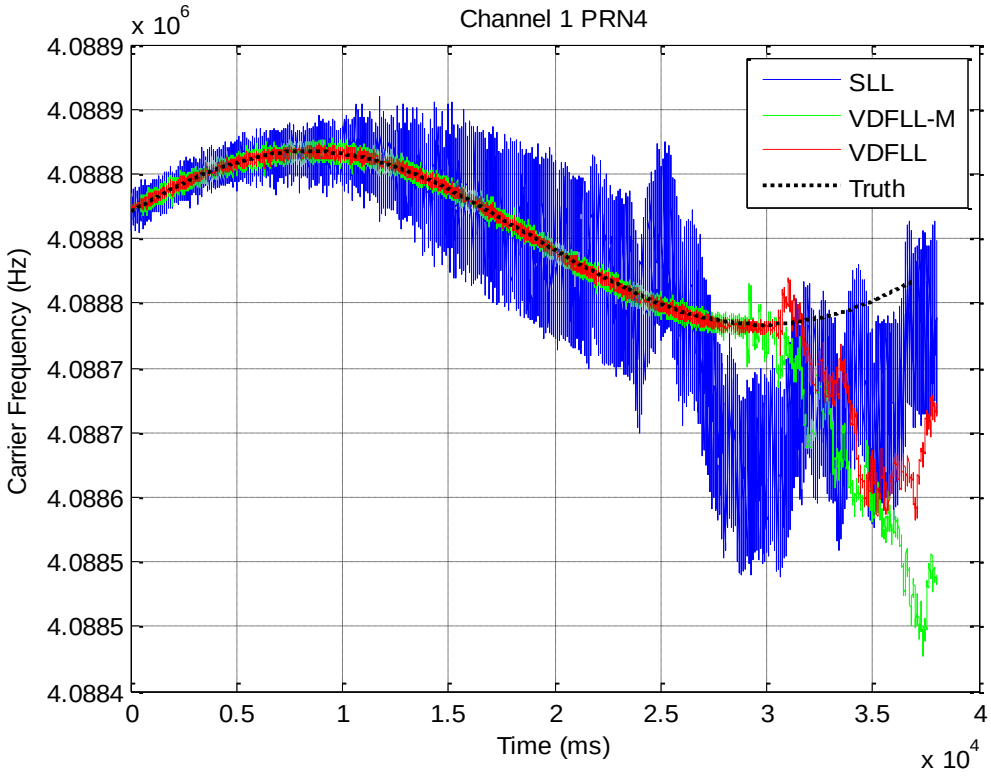

Figure 6.15 Carrier Frequency Tracking Results of PRN4

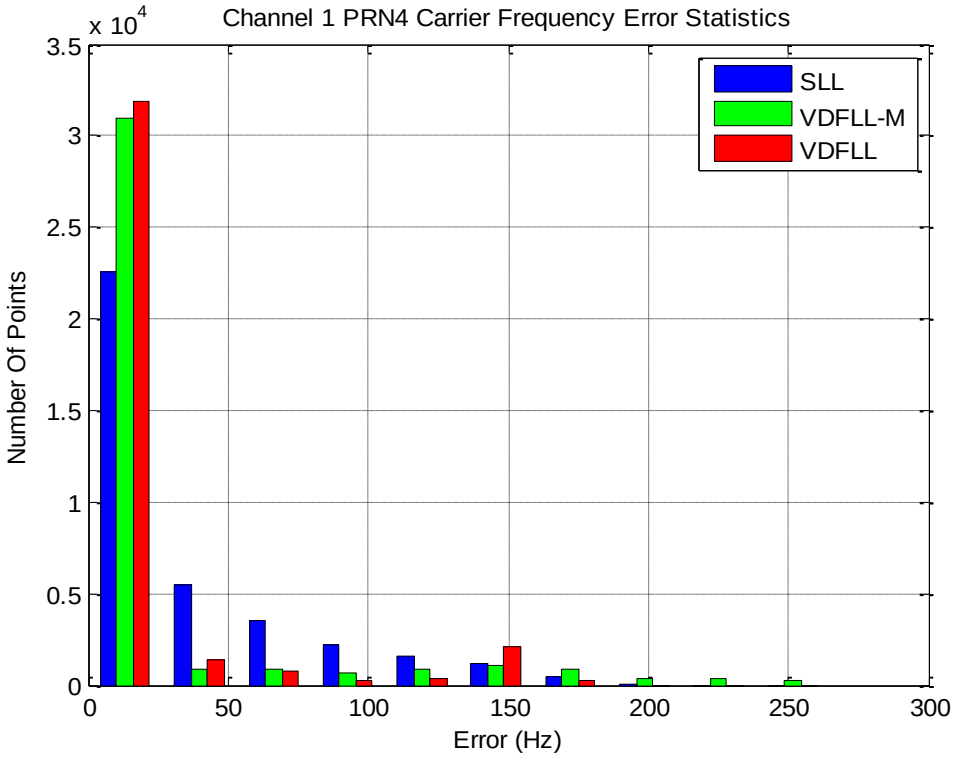

Figure 6.16 Carrier Frequency Error Statistics of PRN4

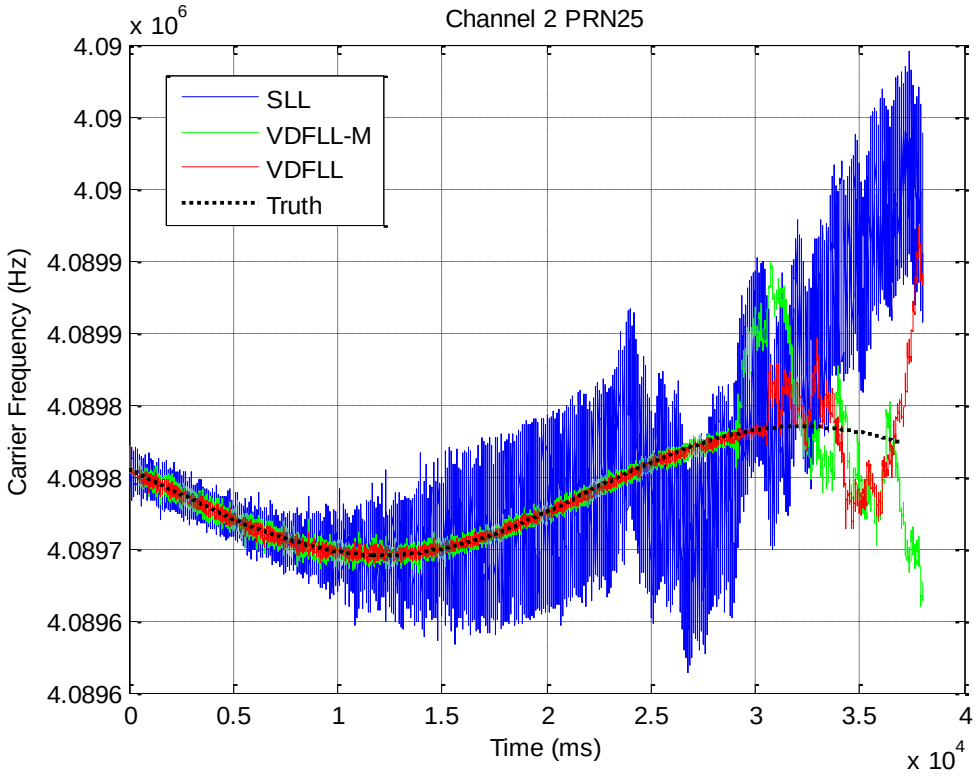

Figure 6.17 Carrier Frequency Tracking Results of PRN25

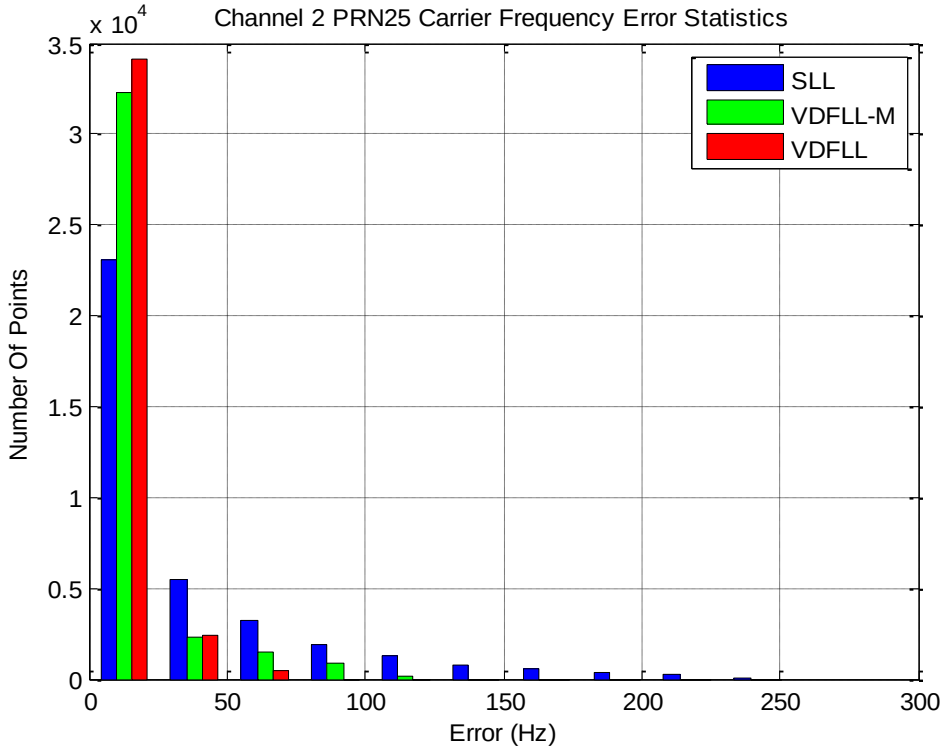

Figure 6.18 Carrier Frequency Error Statistics of PRN25

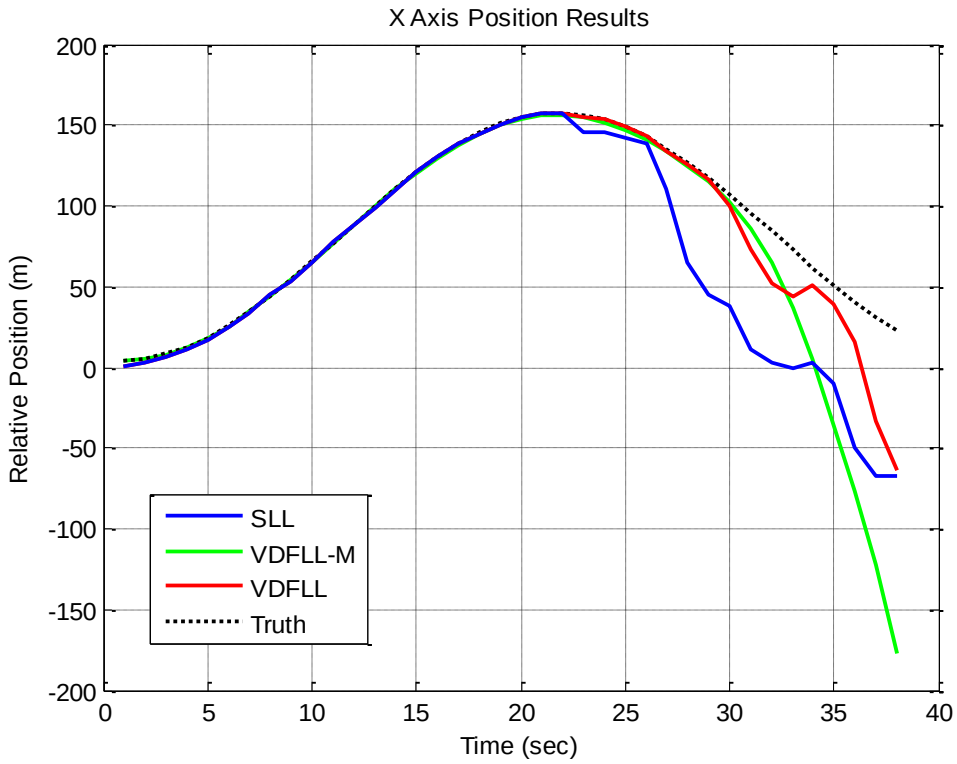

Figure 6.19 X Axis Position Tracking Results

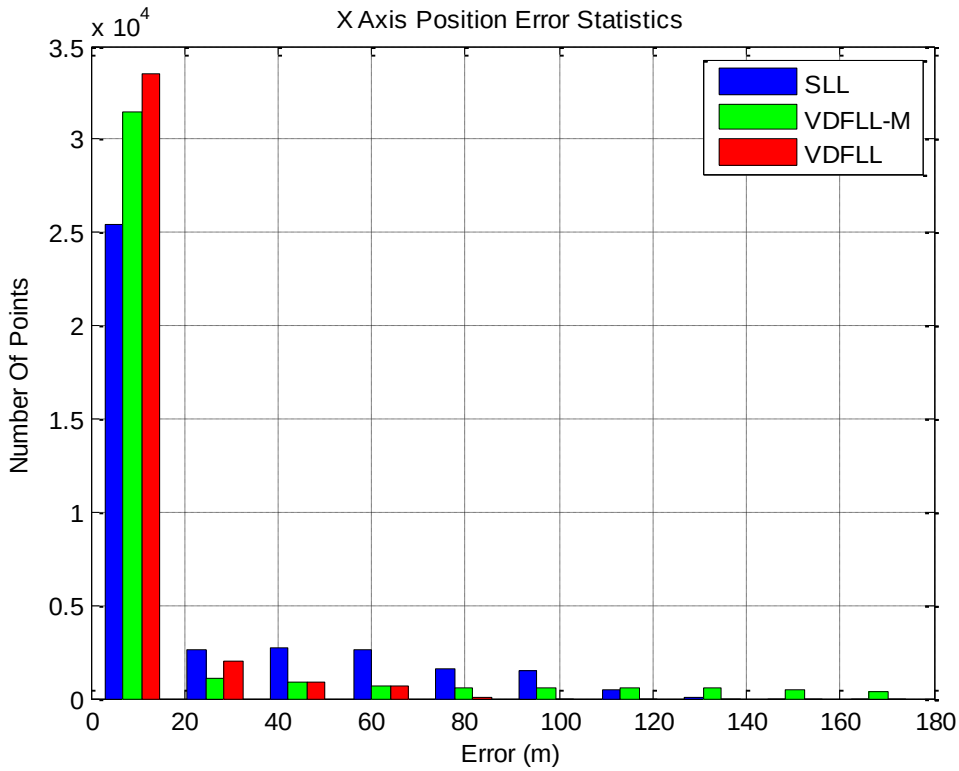

Figure 6.20 X Axis Position Error Statistics

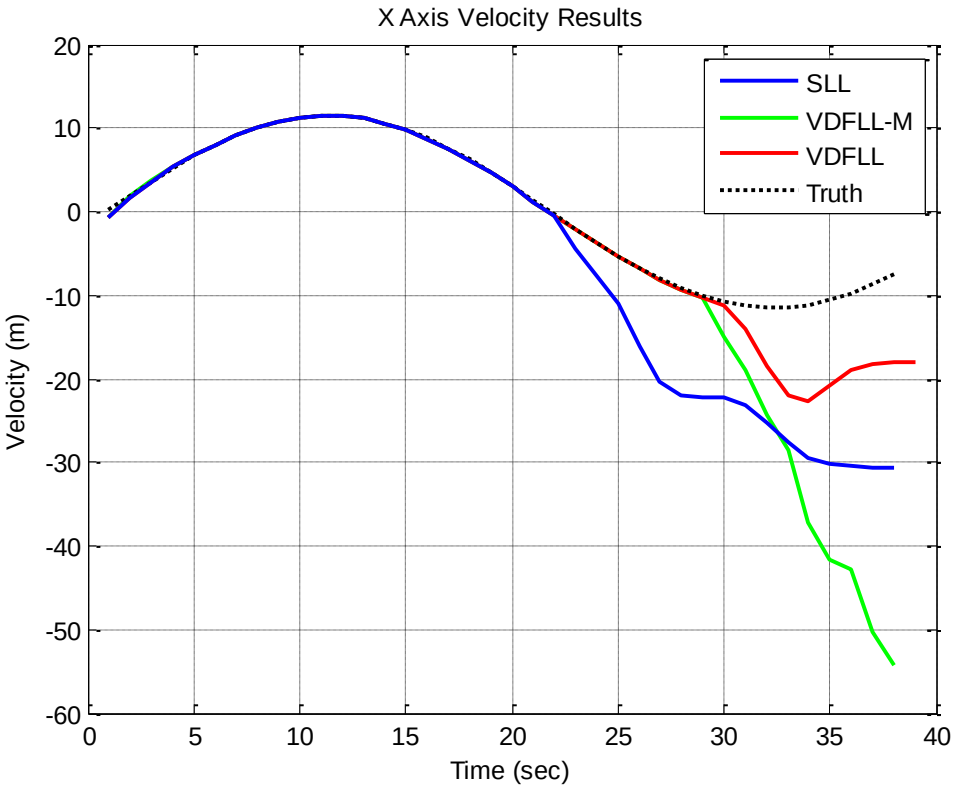

Figure 6.21 X Axis Velocity Tracking Results

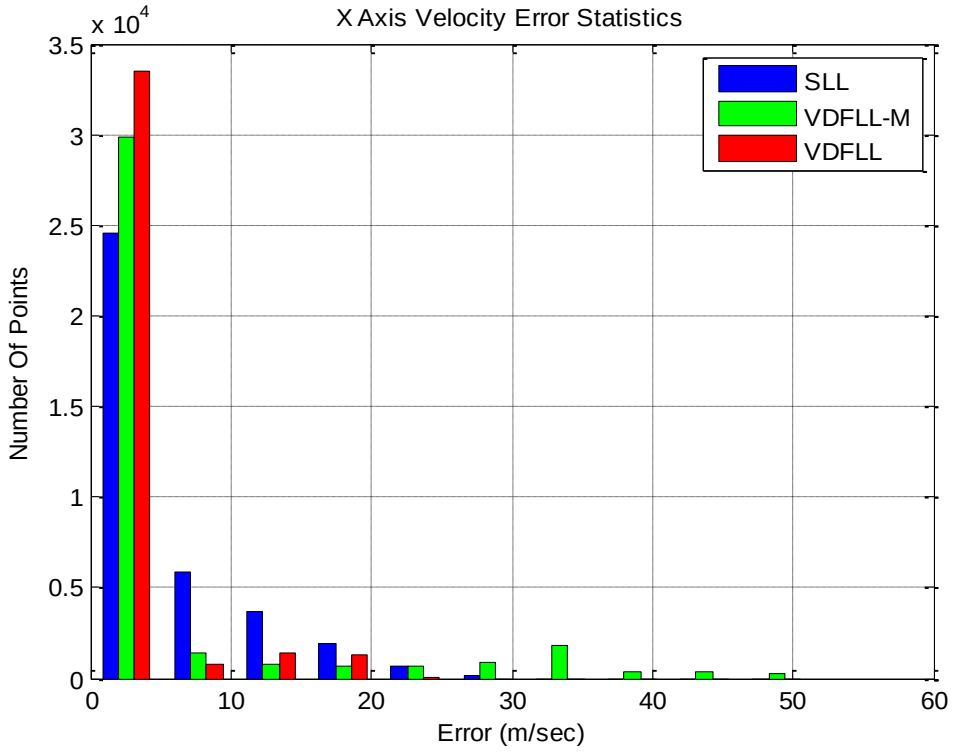

Figure 6.22 X Axis Velocity Error Statistics

Figure 6.15 and Figure 6.17 show the frequency tracking results of PRN4 and PRN25, respectively. Also, the error statistics are shown on Figure 6.16 and Figure 6.18. It can be observed from the frequency results that VDFLL and VDFLL-M maintain lock longer than SLL for PRN4 and PRN25. In Figure 6.15, SLL loses lock at about 24^{th} second and at about 23.5^{th} second in Figure 6.17. On the other hand, VDFLL and VDFLL-M lose lock at about 30.5^{th} second as shown in Figure 6.15. Also, Figure 6.17 VDFLL shows the similar performance and VDFLL-M loses lock at about 29^{th} second with a little bit worse performance. As the signal power decreasing rate is 1 dB-Hz/s, it can be roughly estimated that VDFLL and VDFLL-M are able to work under a 6dB-Hz lower C/No ratio environment than SLL. The error statistics shown on Figure 6.16 and Figure 6.18 verify that VDFLL and VDFLL-M provide better frequency tracking performances than SLL. X axis position and velocity solutions are also given in Figure 6.19 and Figure 6.21, respectively. As a result of the mentioned frequency tracking performances, navigations solutions are affected in same way. The combinations of other satellites generate similar results; therefore, they are not presented here.

6.3.3 Results of Simulated Dataset 3

As mentioned in Table 6.2, C/No ratios of the signals in this data set remain constant at about 45 dB-Hz until 12^{th} second and then at 12^{th} second the signals coming from PRN 2 and PRN 23 are blocked for 10 seconds. After that the signals return to their previous level.

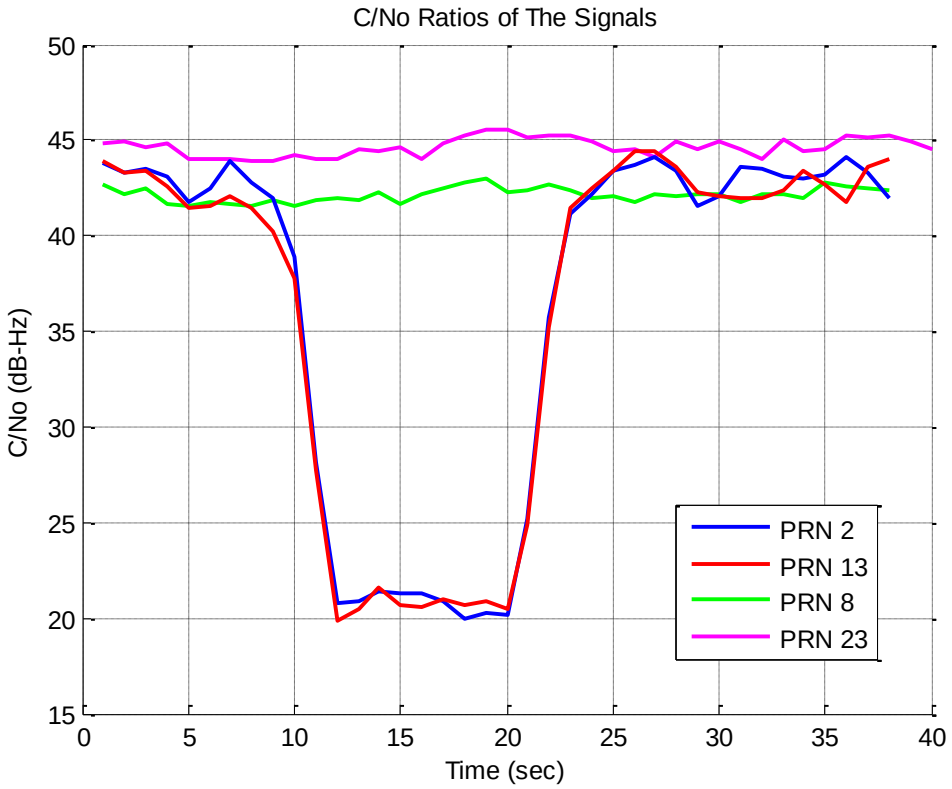

Figure 6.23 C/No Ratio Estimations of Satellites

C/No ratios of some of the selected satellites are shown on Figure 6.23 which verifies the settings of the scenario. In this scenario, a combination of 5 satellites (PRN2, PRN13, PRN8, PRN23 and PRN1) is used to get the navigation solutions for SLL, VDFLL and VDFLL-M. As mentioned before, according to the GPS operation principle, at least 4 satellites are required to calculate the navigation solutions. Therefore, when the signals of two satellites are blocked, only the other 3 satellites are available which causes the situation that prevents SLL to provide navigation solutions. Vector tracking loops can provide navigation solutions under short time of signal blockages.

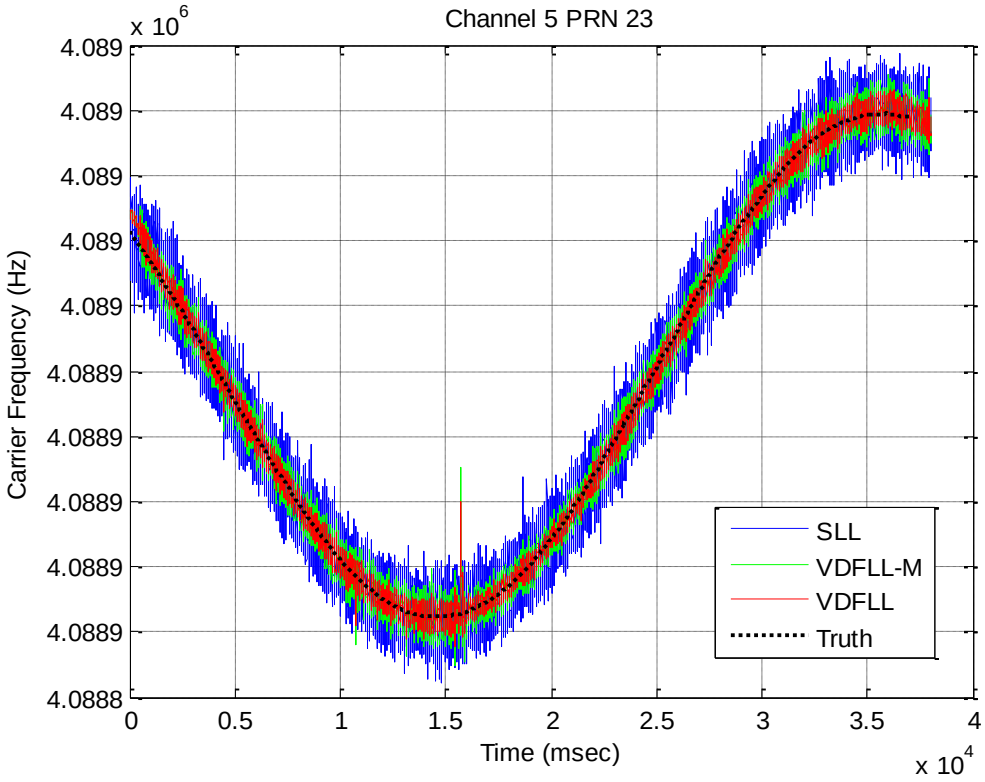

Figure 6.24 Carrier Frequency Tracking Results of PRN23

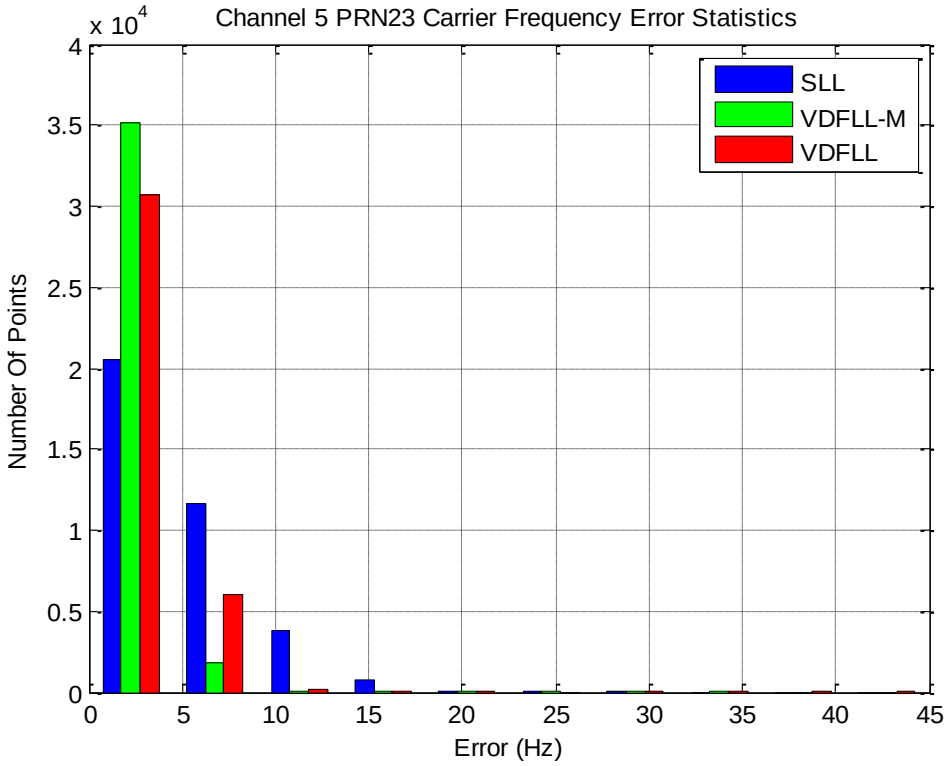

Figure 6.25 Carrier Frequency Error Statistics of PRN23

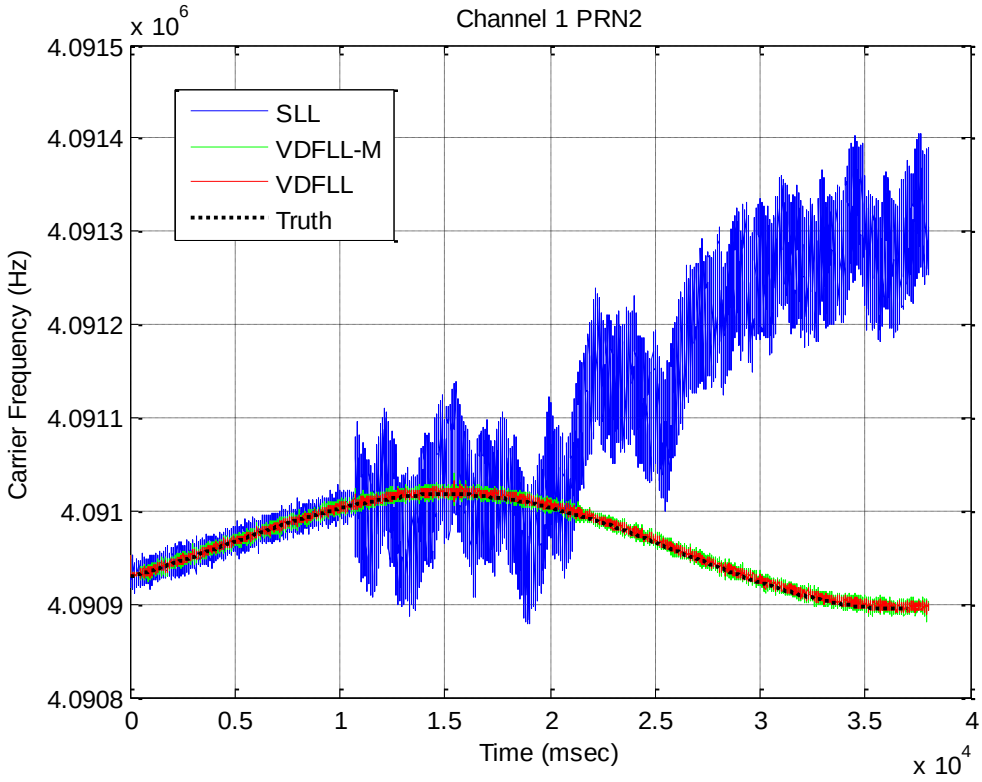

Figure 6.26 Carrier Frequency Tracking Results of PRN2

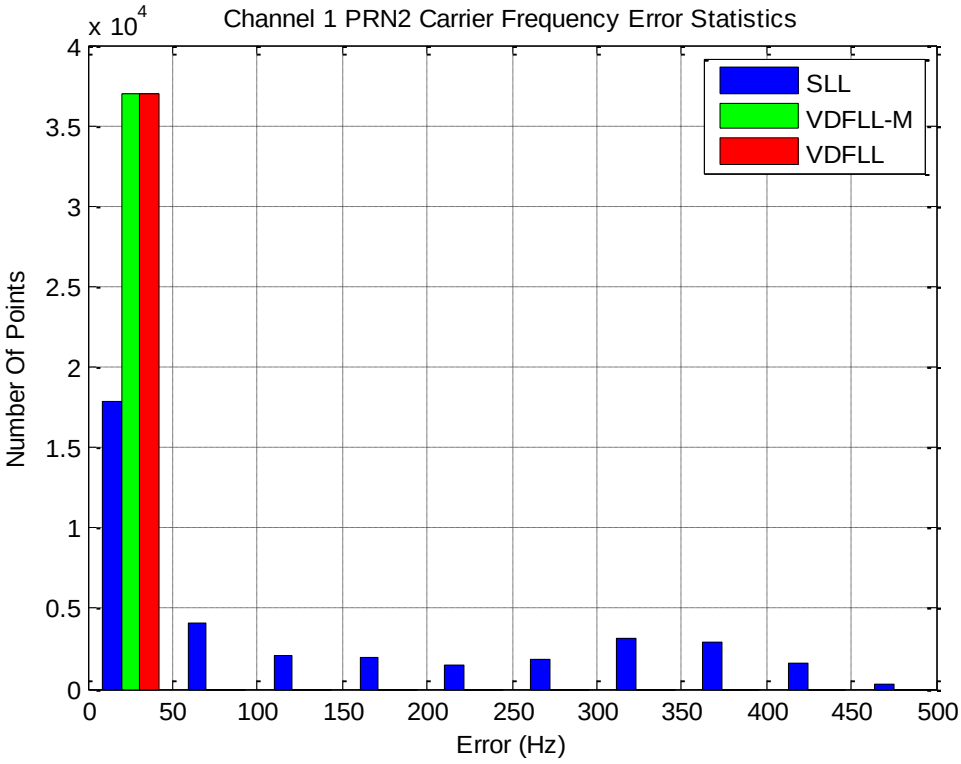

Figure 6.27 Carrier Frequency Error Statistics of PRN2

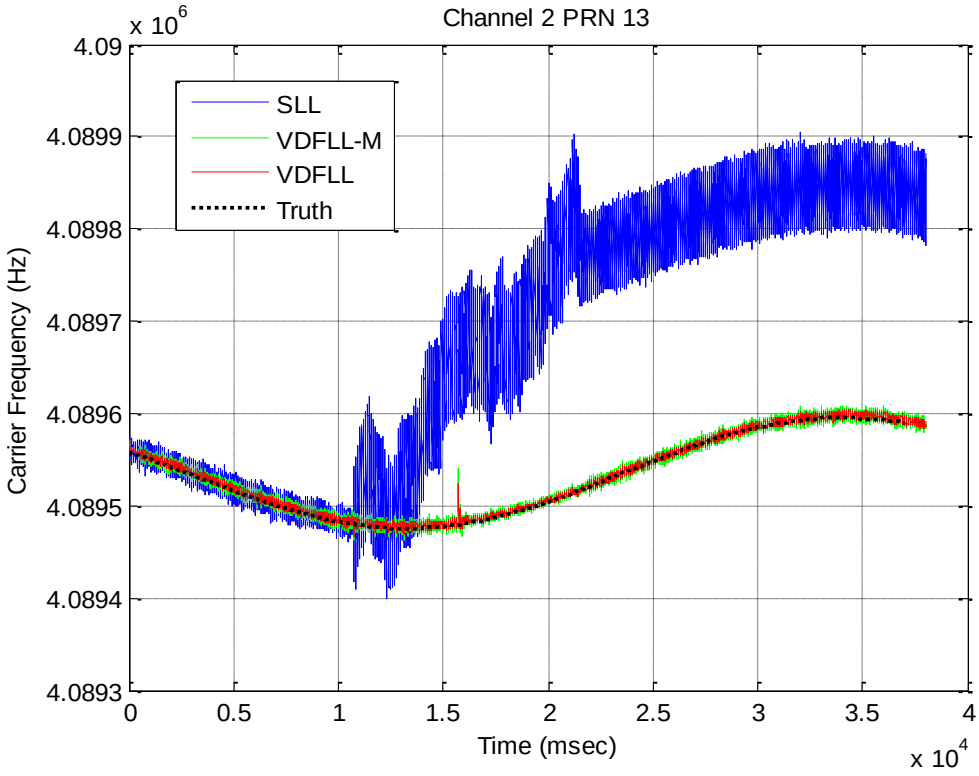

Figure 6.28 Carrier Frequency Tracking Results of PRN13

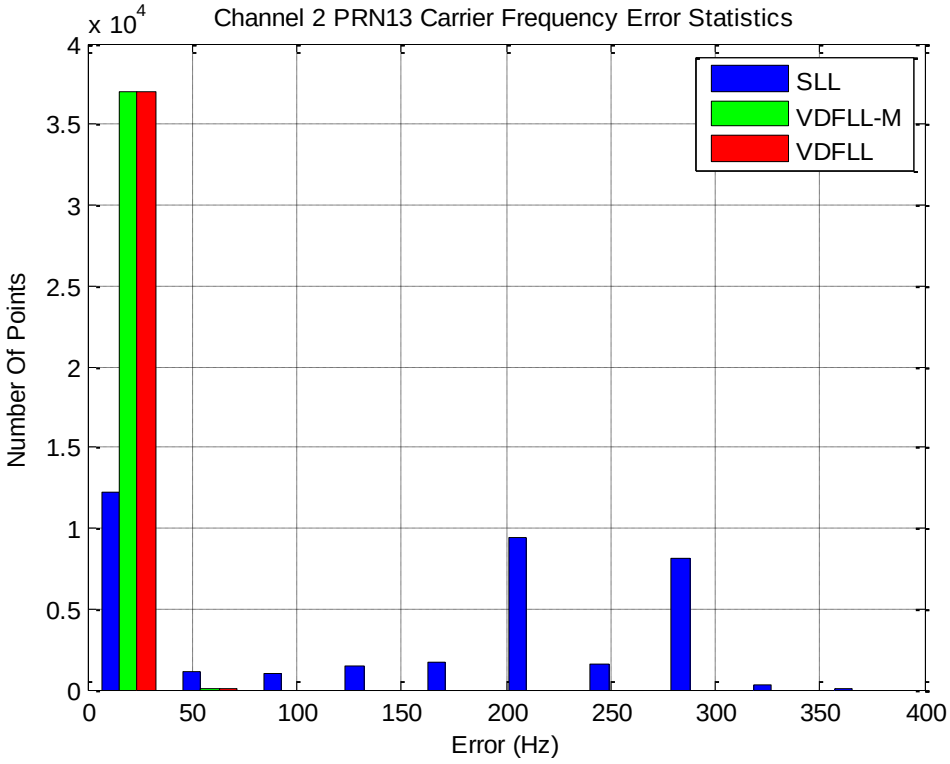

Figure 6.29 Carrier Frequency Error Statistics of PRN13

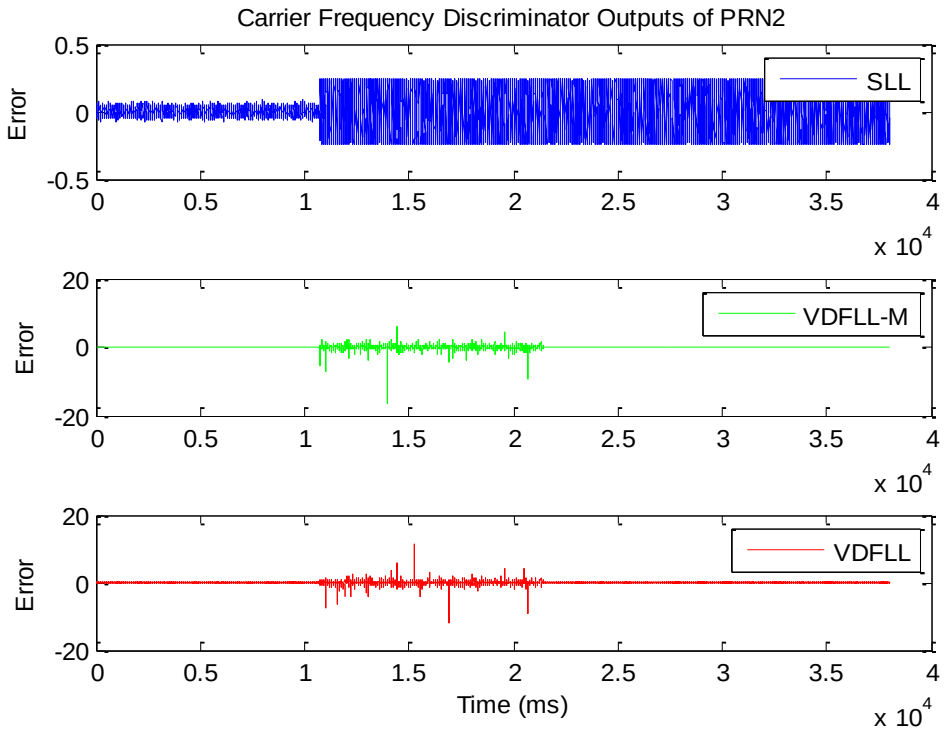

Figure 6.30 Carrier Frequency Discriminator Outputs Comparison for PRN2

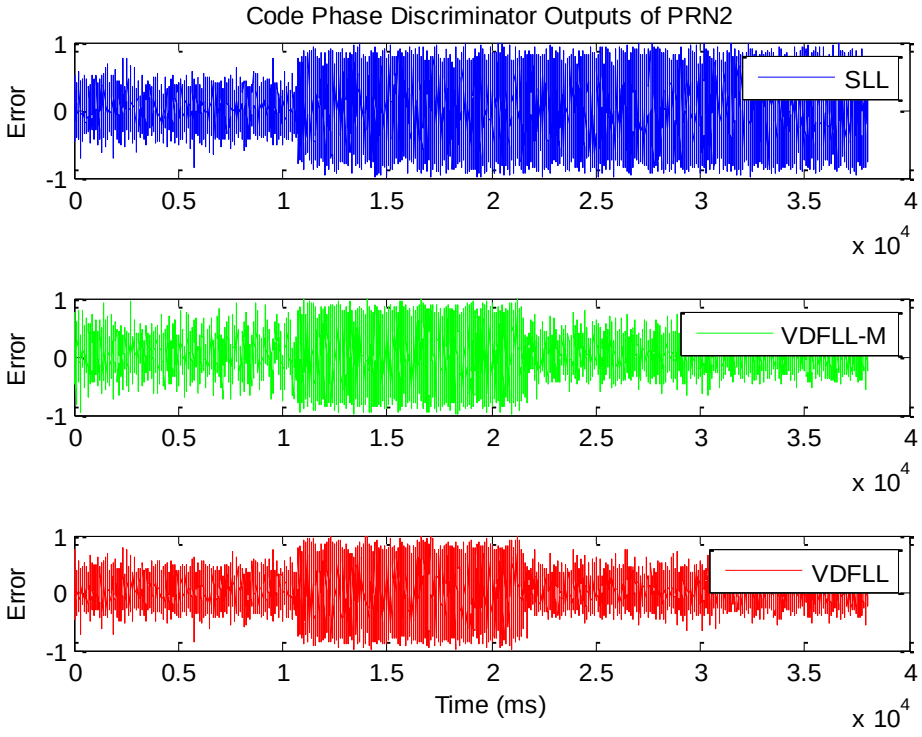

Figure 6.31 Code Phase Discriminator Outputs Comparison for PRN2

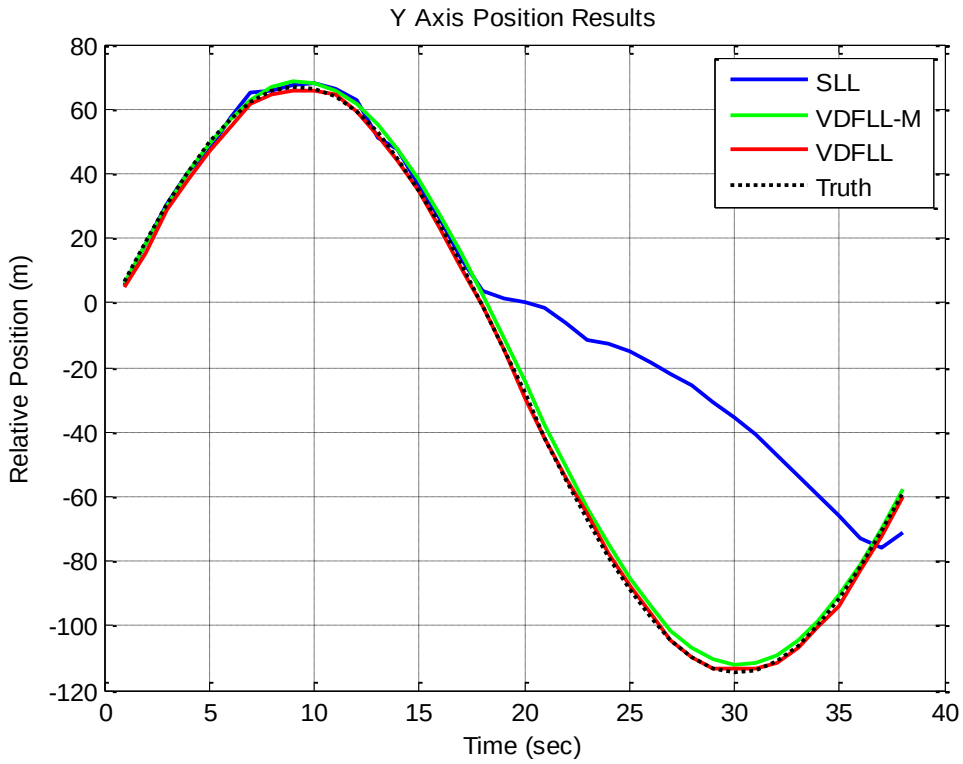

Figure 6.32 Y Axis Position Tracking Results

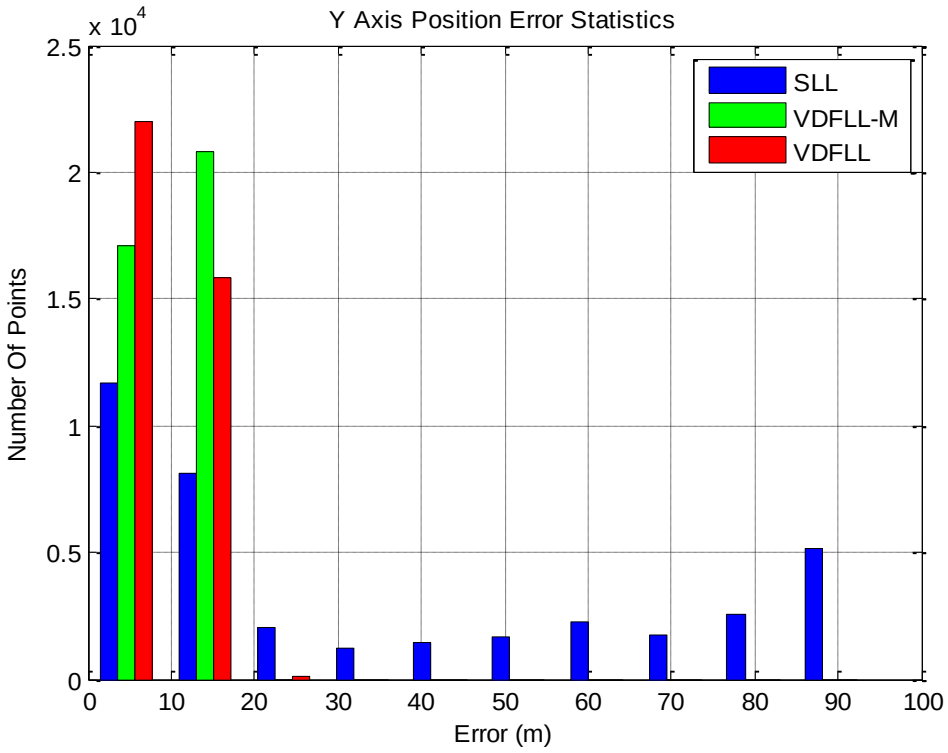

Figure 6.33 Y Axis Position Error Statistics

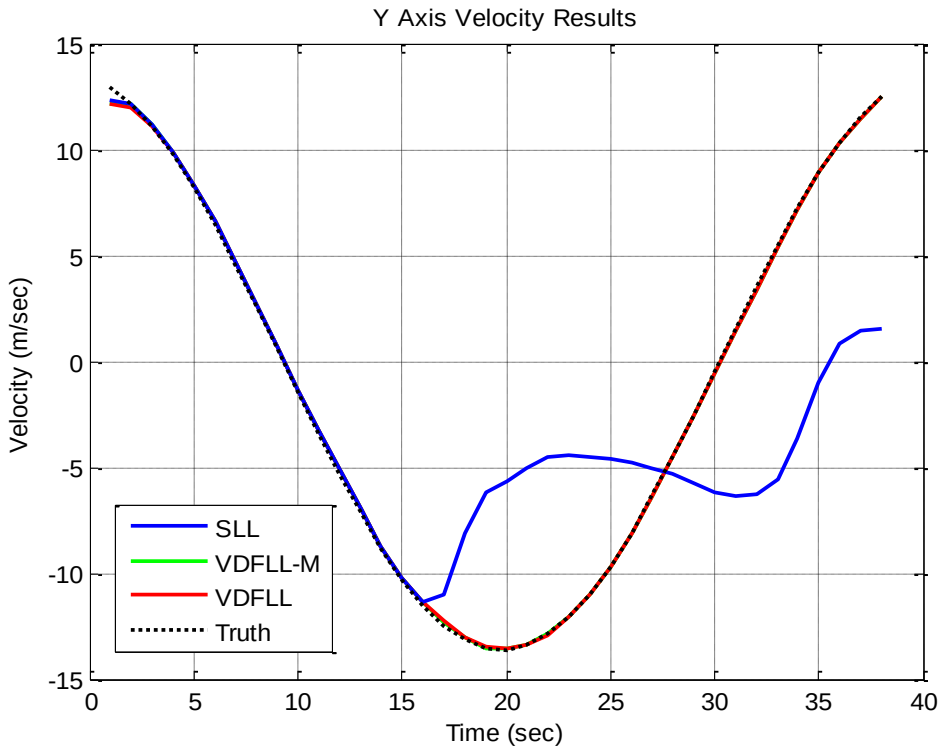

Figure 6.34 Y Axis Velocity Tracking Results

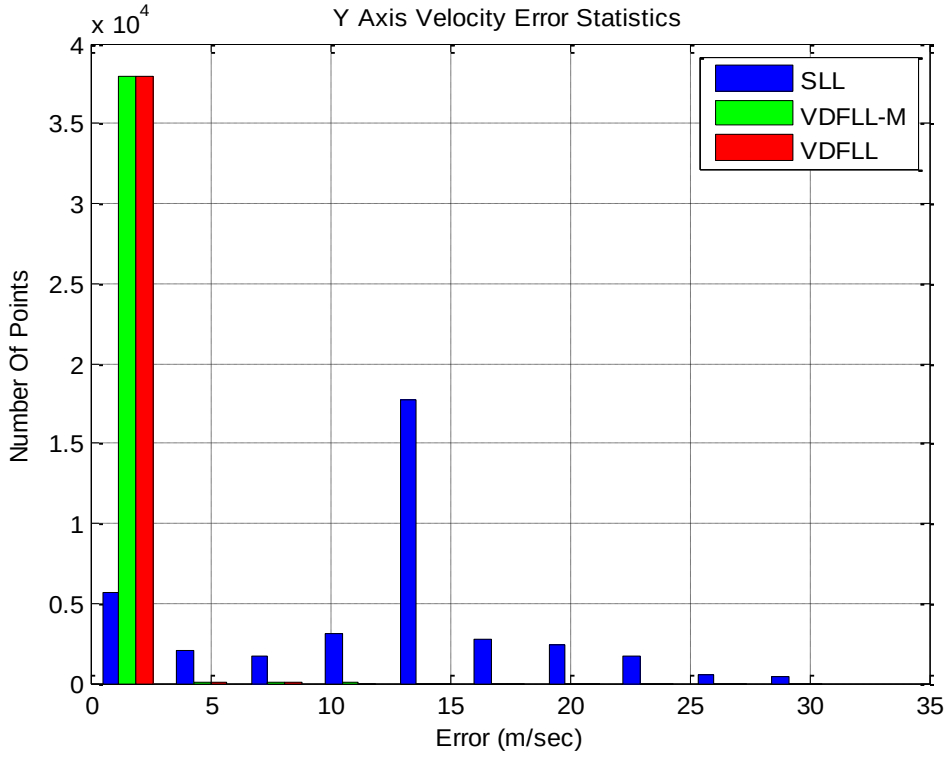

Figure 6.35 Y Axis Velocity Error Statistics

The signal power of PRN23 remains constant at about 45 dB-Hz during the whole scenario as shown on Figure 6.23. Frequency tracking results of PRN23 are given in Figure 6.24 to show that all of the tracking algorithms perform similar with high C/No ratio and no signal blockage. Results are supported with error statistics shown in Figure 6.25. On the other hand, the signal powers of PRN2 and PRN13 are blocked for 10 seconds as shown on Figure 6.23. The frequency tracking results of PRN13 and PRN2 presented in Figure 6.26 and Figure 6.28 show that after a few seconds than start of the signal blockage, SLL loses the track of both satellites and fails to reacquire them when they reappear after 10 seconds. However, VDFLL and VDFLL-M maintain the frequency lock during the blockage. Figure 6.30 and Figure 6.31 show the carrier frequency/code phase discriminator outputs respectively, which give the ability to compare SLL, VDFLL and VDFLL-M tracking states in an easier way. The outputs show that VDFLL and VDFLL-M calculated high phase/frequency errors during satellite blockage and reset to their original value after the blockage period is over. That means VDFLL and VDFLL-M maintain the lock on PRN2 and PRN13 with aiding of strong satellites. Finally, Y Axis position and velocity calculation results are illustrated in Figure 6.32 and Figure 6.34. Since the number of satellites drops to three during the blockage period, SLL cannot provide true navigation solutions as expected. On the other hand, with the help of the Figure 6.32 and Figure 6.34, it can be seen that VDFLL and VDFLL-M are able to provide navigation solutions which also proves their superior performance over SLL. Position and velocity error statistics are also given in Figure 6.33 and Figure 6.35 for the verification of the results.

6.4 Comparison of Computational Load

As mentioned before, VDFLL and VDFLL-M algorithms are implemented on MATLAB in order to process the captured GPS signal. MATLAB is not able process the signal in real-time using the implemented codes. Therefore, offline processing method is used. Comparison of computational time of VDFLL and VDFLL-M on

MATLAB does not provide certain information due to offline processing and characteristics of MATLAB. However, it may prove the intuitive expectation that VDFLL-M is faster than VDFLL. Table 6.5 shows average computational time comparison of VDFLL and VDFLL-M. The results are calculated by using "Run and Time" tool of MATLAB as average of 10 different runs. Same data is used for all calculations, which is 40 seconds long. Given calculation results consist only of the time spent for tracking purposes. Also, MATLAB R2014a runs on a 64bit operating system with 8gb installed memory and 3.3GHz i5 processor.

Table 6.5 Computational Time Comparison of VDFLL and VDFLL-M

	VDFLL	**VDFLL-M**	**Difference**
Time spent for tracking 4 satellites (sec)	147.738	132.629	%10.37
Time spent for tracking 5 satellites (sec)	174.513	154.632	%11.39
Time spent for tracking 10 satellites (sec)	321.742	286.931	%10.82

The results prove that VDFLL-M runs faster than VDFLL as expected. The difference is about %10; but, as mentioned before, comparison should be done on a system with real time operation in the sense of computational load.

6.5 Summary

In this chapter, performance comparison of SLL, VDFLL and VDFLL-M algorithms are presented over three different simulation scenarios. As can be clearly seen from the results, VDFLL gives the best performance in all scenarios and VDFLL-M follows it slightly behind. Both of them give significantly better performances than scalar tracking algorithm.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this thesis, receiver architectures based on the vector tracking loops were presented. The tracking operation and navigation state estimation are combined in a single algorithm in these receiver architectures. The implemented vector tracking algorithm is a Vector Delay/Frequency Lock Loop (VDFLL) algorithm, which tracks both carrier frequencies and code phases in aggregate manner by a single Extended Kalman Filter (EKF). On the other hand, modified version of the VDFLL algorithm operates with an EKF having less number states. It only tracks carrier frequency and calculates the code phases indirectly using the information coming from carrier tracking loop. In this study, it is called VDFLL-M, since it is the modified version VDFLL. Performance comparisons of these vector tracking algorithms and scalar tracking algorithm were presented and various advantages of vector tracking algorithms over scalar tracking algorithms were illustrated and discussed. The results showed that vector tracking algorithms are able to operate significantly lower C/No ratio level and tolerate momentarily signal blockage. The results of the first dataset, where all satellites have decreasing signal powers, show that VDFLL can provide 3dB-Hz and VDFLL-M can provide 2dB-Hz better performance than SLL and the accuracy of the navigation solutions are improved. In the second dataset, in which combination of four satellites with two of them have decreasing power are used, the results show that VDFLL and VDFLL-M provide over 6dB-Hz better performances

over SLL and navigation solutions with smaller errors than SLL. The last dataset were also used to show the ability of VDFLL and VDFLL-M to tolerate momentarily signal blockages. Both of the vector tracking algorithms outperforms the scalar tracking algorithm in both accuracy in navigation solutions and ability to tolerate the signal power outage. Despite the fact that the VDFLL-M algorithm is a new proposed algorithm, the results are very promising and almost similar to VDFLL algorithm. In addition to good performances, it also provides lower computational load than VDFLL algorithm.

There is great potential of future work in the vector tracking algorithms. First of all, VDFLL and VDFLL-M algorithms can be compared in computational load sense in details. Also, they can be improved by using the outputs of implemented C/No ratio estimation methods to adjust the measurement noise covariance matrix. Moreover, comparing the performance results of SLL, VDFLL and VDFLL-M in real time operation may give more convincing results. Finally, integrating the vector tracking loops with an Inertial Measurement Unit would provide an Ultra Tightly Coupled system that performs better in high dynamics and has the ability to reacquire the signals after long period of signal blockage.

REFERENCES

[1] B. W. Parkinson, J. J. Spilker Jr., "Global Positioning System: Theory and Applications", American Institute of Aeronautics and Astronautics, Washington DC, 1996.

[2] Basic Guide to Advanced Navigation, NATO, RTO Publication, SET-054/RTG-30, 2004.

[3] C. Rizos: "Trends in Geopositioning for LBS, Navigation and Mapping" in 4th International Symposium And Exhibition on Geoinformation 2005, Pengang, Malaysia, 27-29 September, Invited Paper.

[4] M. Lashley and D. M. Bevly, "Analysis of Discriminator Based Vector Tracking Algorithms" in Proc. National Tech. Meeting Institute of Navigation., San Diego, CA, January 2007.

[5] S. Zhao, D. Akos, "An Open Source GPS/GNSS Vector Tracking Loop Implementation, Filter Tuning, and Results", Institute of Navigation, International Technical Meeting, January, 2011

[6] D. Benson, "Interference Benefits of a Vector Delay Lock Loop (VDLL) GPS Receiver", in Proc. 63rd Annu. Meeting Institute of Navigation, Cambridge, MA, April 2007

[7] Z. Zhenzhen, C. Zhu, T. Guangfu, L. Shufeng, H. Fukan, "EKF Based Vector Delay Lock Loop Algorithm for GPS Signal Tracking", International Conference On Computer Design And Applications, 2010.

[8] G. Liu, R. Zhang, M. Guo, X. Cui, "Accuracy Comparison of GNSS of Vector and Scalar Tracking Loop", in Proceedings of IEEE Chinese Guidance, Navigation and Control Conference, August 2010.

[9] M. Lashley, D. M. Bevly and J. Y. Hung, "A valid Comparison of Vector and Scalar Tracking Loops", in Proceedings of the 21st International Technical Meeting

of the Satellite Division of The Institute of Navigation, Svannah, GA, September 2008.

[10] J. J. Spilker, Fundamentals of Signal Tracking Theory, In: Global Positioning System: Theory and Applications, Vol. I. Progress in Astronautics and Aeronautics, Volume 163, AIAA, Washington, DC, 1996. St. Petersburg, Russia.

[11] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, S. H. Jensen, A Software-Defined GPS and Galileo Receiver, A single Frequency Approach, Boston, Birkhauser, 2007.

[12] U.S. Coast Guard Navigation Centre: http://www.navcen.uscg.gov/

[13] E. D. Kaplan, C. J. Hegarty, Understanding GPS Principles and Applications, Second Edition, Boston – London, Artech House, 2006

[14] R. Gold, Optimal Binary Sequence for Spread Spectrum Multiplexing, IEE Transactions on Information Theory, 13(4): 619-621.

[15] G. Hill, The Benefits of Undersampling, Electronic Design, 69-79 1994.

[16] Y. Tsui, J. Bao, Fundamentals of Global Positioning System Receivers, A Software Approach, USA, Wiley-Interscience 2005.

[17] F. D. Nunes, J. M. N. Leitao, "A New Fast Code/Frequency Acquisition Algorithm for GPS C/A Signals", Instituto de Telecomunicacoes – IST, Paris, IEEE, 766-770, 2003

[18] "RF Front-End IC Simplifies Software GPS Receiver", Maxim Integrated Application Note 4009, March 2007.

[19] G. Guojiang, INS-Assisted High Sensitivity GPS Receivers for Degraded Signal Navigation, PhD Thesis, Department of Geomatics Engineering, The University of Calgary, Canada.

[20] "SE4120L GNSS Receiver IC", Datasheet. Rev3p5, 2009

[21] Spirent. GSS7700 Multi-Channel GPS/SBAS Simulators with M-Noise Product Specification; Spirent Communications Limited: Sunnyvale, CA, October 2007.

[22] Bhattacharyya, S. Performance and Integrity Analysis of the Vector Tracking Architecture of GNSS Receivers, PhD Thesis, The University of Minnesota, US, 2012.

[23] Lashley, M. Modeling and Performance Analysis of GPS Vector Tracking Algorithms, PhD Thesis, Department of Electrical and Computer Engineering, Auburn University, Alabama, US, December 2009.

[24] M. S. Sharawi, D. N. Aloi, "C/No Estimation in a GPS Software Receiver in the Presence of RF Interference Mitigation via Null Steering for the Multipath Limiting Antenna", IEEE, January, 2007

[25] M. S. Sharawi, D. Akos, D. N. Aloi, "GPS C/N_o Estimation in the Presence of Interference and Limited Quantization Levels", IEEE Transactions on Aerospace and Electronic Systems, January, 2007

[26] Brown, G. R., Hwang Y. C. Patrick, "Introduction to Random Signals and Applied Kalman Filtering", John Wiley and Sons, 3rd Edition, 1997.

[27] Gelb A., "Applied Optimal Estimation", 1974, MIT Press.

[28] Welch, G., Bishop, G., An ıntroduction to the Kalman Filter, Technical Report 95-041, University of North Carolina, Chapel Hill, 2006.

[29] Kahraman, E., Navigation Algorithms and Autopilot Application for an Unmanned Airvehicle, MSc Thesis, Middle East Technical University, Ankara, TR, December 2010.

[30] Zubaroğlu, T., Yazılım Tabanlı GPS Almaçlarının İncelenmesi, MSc Thesis, Hacettepe University, Ankara, TR, July 2013.

[31] Control Segment. Retrieved [Last accessed on November 12, 2015], from http://www.gps.gov/systems/gps/control

[32] The Global Positioning System. Retrieved [Last accessed on November 12, 2015], from http://www.colorado.edu/systems/geography/gcraft/notes/gps/gps_f.html

[33] http://images.syriusamis.org/other/Miscelleneous [Last accessed on November 12, 2015]

[34] http://infohost.nmt.edu/~mreece/gps/whatisgps.html [Last accessed on November 12, 2015]

APPENDIX A

KALMAN FILTERING

A.1. Discrete Kalman Filter

For the linear dynamic systems with disturbances modeled by Gaussian random processes, the Kalman filtering is an optimal method to estimate the states of the system. The cost function defines the optimality of the filter. It is generally used to estimate the states of a system driven with Gaussian noise which corrupts the system's states measurements. The Kalman filter produces unbiased and minimum variance estimates, which are always the conditional mean of the state vector, of the system's states. The Kalman filter can be modeled for both continuous and discrete time cases. In this thesis, discrete time Kalman filter is implemented. Therefore, only discrete time Kalman filter will be discussed in this section. Detailed information can be found in [26], [27] and [28] for both continuous and discrete time cases.

Consider the system modeled in the form as shown in (A.1) to be estimated.

$$x_{k+1} = \Phi_k x_k + w_k \qquad \textbf{(A.1)}$$

$$w_k \sim N(0, Q_k)$$

The noisy measurement of the process is assumed to occur at discrete points in time as a linear combination of the current states of the system plus noise as in (A.2).

$$z_{k+1} = H_k x_k + v_k \qquad \textbf{(A.2)}$$

$$v_k \sim N(0, R_k)$$

where,

x_k = the (nx1) process state vector at time t_k.

Φ_k = the (nxn) matrix relating x_k to x_{k+1} or state transition matrix

w_k = the (nx1) noise vector assumed to be Gaussian with zero mean and covariance Q_k

z_k = (mx1) measurement vector at time t_k

H_k = the (mxn) matrix giving the ideal connection between the measurement and the state vector at time t_k

v_k = the (mx1) measurement error assumed to be Gaussian with zero mean and covariance R_k.

The processes and measurement noises are measured to be uncorrelated for all values and the covariance matrices are given as follows:

$$E\left[w_k w_i^T\right] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \quad \textbf{(A.3)}$$

$$E\left[v_k v_i^T\right] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \quad \textbf{(A.4)}$$

$$E\left[w_k v_i^T\right] = 0, \qquad \text{for all k and i} \quad \textbf{(A.5)}$$

The equations (A.6) to (A.10) are the recursive Kalman filter equations.

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - H_k \hat{x}_k^-\right) \quad \textbf{(A.6)}$$

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \quad \textbf{(A.7)}$$

$$P_k = (I - K_k H_k) P_k^- \quad \textbf{(A.8)}$$

$$\hat{x}_{k+1}^{-} = \Phi_k \hat{x}_k \quad \textbf{(A.9)}$$

$$P_{k+1}^{-} = \Phi_k P_k \Phi_k^T + Q_k \quad \textbf{(A.10)}$$

where,

$\hat{x}_k$ = the Kalman state vector. Topping “hat” means estimation.

P_k = the error covariance matrix.

z_k = the measurement vector.

R_k = the measurement noise covariance matrix.

Q_k = the noise covariance matrix.

H_k = the measurement matrix.

K_k = the Kalman gain matrix.

The derivations of the given recursive equations can be found on [26], [27] and [29].

Graphical interpretation of the Kalman filter algorithm is given in Figure A.1. It is assumed that there is an initial estimate of the process which is based on all the prior knowledge at some point in time t_k. As shown on the Figure A.1, the Kalman filter is initialized by using the prior estimate $\hat{x}_0^-$ and its error covariance P_0^-. The filter then calculates the Kalman gain matrix. After that, the differences between the measured and predicted states are found and used to update the filter estimations. The error covariance matrix is then calculated for the updated estimates to reflect the covariance of the corrected state vector estimate. Then the Kalman filter projects ahead the states of the system and the error covariance matrix to the next sampling time. The outputs of the last step are then used as initial conditions to the recursive algorithm.

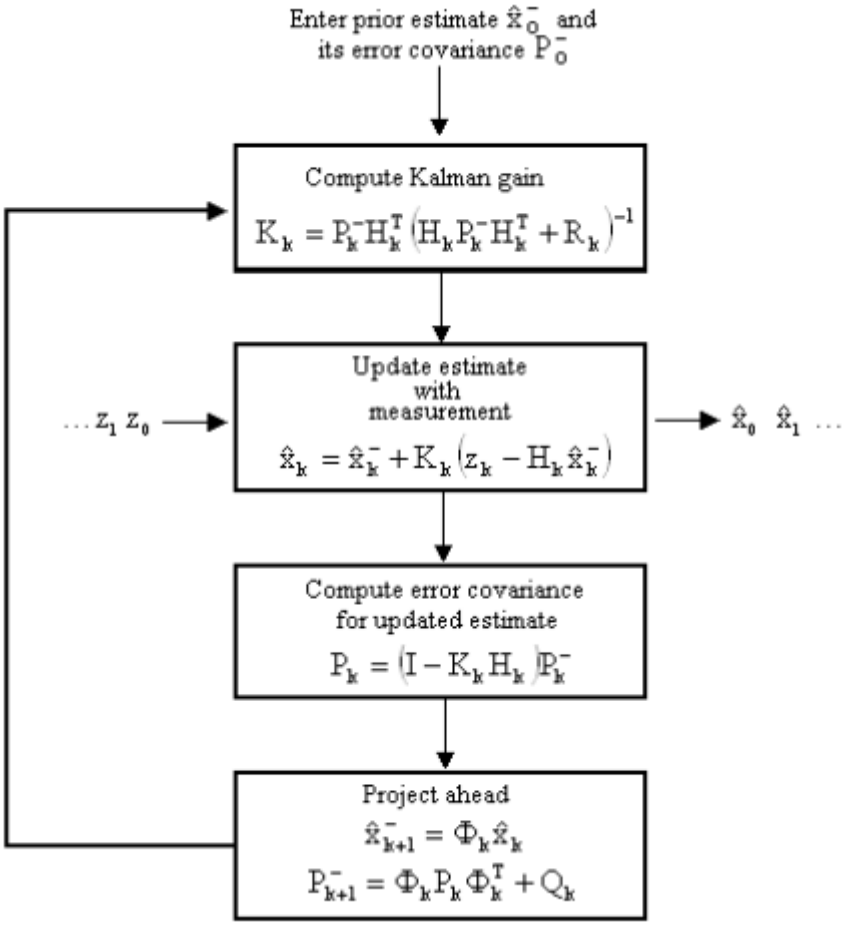

Figure A.1 Kalman Filter Algorithm [29]

A.2. Extended Kalman Filter

The discrete Kalman filter discussed above is optimal only for the linear systems. However, most of the applications require optimal estimators for nonlinear systems. The Extended Kalman Filter (EKF) is a specialized version of the Kalman filter that can be used for nonlinear systems. It linearizes about the current mean and covariance. Therefore, it is not optimal in the least squares sense but approximates the optimal solution. The system model presented in equation (A.1) is changed to the equation (A.11).

$$x_{k+1} = f(x_k, k) + w_k \tag{A.11}$$

$$w_k \sim N(0, Q_k)$$

As can be seen on equation (A.11), the state vector is a nonlinear function of the previous state time index *k,* plus the process noise which is assumed to be Gaussian random process with zero mean and covariance matrix Q_k .

The measurement equation given in (A.2) also changes to (A.12) since the measurements of the system are nonlinear functions of current states of the system and time index *k,* plus the measurement noise which is assumed to be Gaussian random process with zero mean and covariance matrix R_k .

$$z_{k+1} = h(x_k, k) + v_k \tag{A.12}$$

$$v_k \sim N(0, R_k)$$

As mentioned before, the principle of EKF is to linearize the system around the current best estimates of the states. The linearization of the nonlinear function $f(x_k, w_k)$ and the nonlinear measurement function $h(x_k, k)$ are shown on the equations (A.13) and (A.14).

$$\Phi_{k+1} \approx \left.\frac{\partial f(x_k, k)}{\partial x_k}\right|_{x=\hat{x}_k^-} \tag{A.13}$$

$$\Phi_{k+1} \approx \left.\frac{\partial f(x_k, k)}{\partial x_k}\right|_{x=\hat{x}_k^-} \tag{A.14}$$

Using these approximations, the Kalman filter algorithm can be used for the given nonlinear system. Only the equations (A.6) and (A.9) are rewritten as follows:

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - h\left(\hat{x}_k^-, k\right)\right) \tag{A.15}$$

$$\hat{x}_{k+1}^- = f\left(\hat{x}_k^-, k\right) \tag{A.16}$$

The graphical interpretation of the Extended Kalman Filter is given in Figure A.2 . The algorithm is the same as given in section A.1.

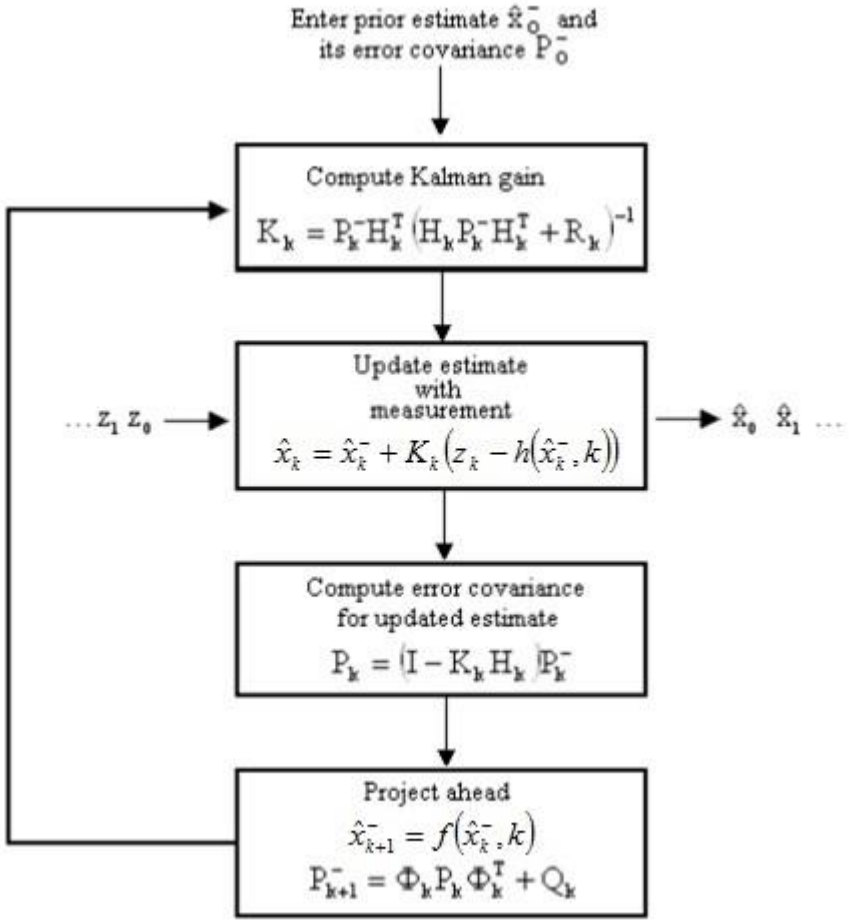

Figure A.2 Extended Kalman Filter Algorithm

APPENDIX B

C/N_0 RATIO ESTIMATION

B.1. Introduction

In this appendix, some C/N_o ratio estimation methods of received signals are explained. The given information is originally presented in [24] and [25]. C/No ratio estimation is a very important parameter for the performance description of GPS receivers. C/No ratio estimations of the received signals are used intensively while presenting the simulation results in this thesis. Two commonly used methods to estimate C/No ratios are used in this study and explained in this appendix.

B.2. C/No Estimation Methods

The power ratio method (PRM) and variance summing method (VSM) are presented in this appendix. VSM is based on computation of variance and mean of the in-phase (I) and quadrature (Q) components of tracked signal. The second method, PRM, is based on the calculation of the narrowband to wideband powers ratio over a time interval. Also, it depends on the mean of the corresponding ratio and sample size taken in the estimation of power.

B.2.1. Variance Summing Method (VSM)

This method is described in [24] and [25] in details. Z_k as the time series sample is calculated for every I and Q accumulations that are generated during tracking of the signal as in equation (B.1).

$$Z_k = I_k^2 + Q_k^2 \quad \textbf{(B.1)}$$

The mean and variance of the resulting time series can be calculated as follows:

$$Z_{mean} = \frac{1}{k}\sum_{1}^{k} Z_k \quad \textbf{(B.2)}$$

$$\sigma_Z^2 = \frac{1}{k-1}\sum_{1}^{k}\left(Z_k - Z_{mean}\right) \quad \textbf{(B.3)}$$

The average power can be written as:

$$\left(\frac{NA}{2}\right)^2 = \sqrt{Z_{mean}^2 - \sigma_Z^2} \quad \textbf{(B.4)}$$

where,

N = the number of samples (of the time series *Z)*

A = amplitude of the signal seen at the RF-front end output

The noise accumulation term variance for I and Q components is written as:

$$\sigma_{IQ}^2 = \frac{1}{2}\left(Z_{mean} - \sqrt{Z_{mean}^2 - \sigma_Z^2}\right) \quad \textbf{(B.5)}$$

Using the equation (B.5), C/No ratio can be calculated as follows.

$$\frac{C}{No} = 10\log_{10}\left[\frac{\left(NA/2\right)^2}{2T_{accu}\sigma_{IQ}^2}\right] \quad \textbf{(B.6)}$$

where,

T_{accu} = 0.001 for the 1 ms accumulation intervals as used in this thesis.

B.2.2. Power Ratio Method (PRM)

For the detailed explanation of this method please refer to [24] and [25]. Basically, it tries to estimate the signal to noise ratio (SNR) by comparing the powers in two different bandwidths, and relate it to *C/No*. The bandwidth of wideband power, called WBP_k, is $1/T_{accu}$, where T_{accu} is the time for the accumulation of the power measurement as mentioned above. In this study, the accumulation time is selected as 1 ms which causes a bandwidth of 1 KHz for the *WBP*. On the other hand, the bandwidth of the narrowband power, called as NBP_k, is $1/(MT_{accu})$, where *M* is the number of samples that is used to calculate the wideband power. Synchronization of navigation data bits is very important to estimate the *C/No* with the help of this method, since the summation during a navigation bit transition may result with incorrect estimations.

Powers of the narrowband and wideband are calculated by summing the I and Q components that are the correlator outputs.

The wideband and narrowband powers can be calculated as follows:

$$WBK_k = \sum_{i=1}^{M} \left(I_i^2 + Q_i^2 \right) \quad \textbf{(B.7)}$$

$$NBP_k = \left(\sum_{i=1}^{M} I_i \right)^2 + \left(\sum_{i=1}^{M} Q_i \right)^2 \quad \textbf{(B.8)}$$

where

M = the number of samples that is used the calculation of *WBP* and *NBP*.

The wideband and narrowband powers are calculated within the same navigation bit. The possible values for M are [1, 2, 4, 5, 10, 20] in order to prevent summation during a navigation bit transition, since the accumulation time is selected as 1 ms and the navigation data has 50Hz bitting rate. The noise power estimate at kth time epoch can be written as follows:

$$NP_k = \frac{NBP_k}{WBP_k} \quad \textbf{(B.9)}$$

The noise power calculated in equation (B.9) is related to C/No by averaging as follows:

$$\mu_{NP} = \frac{1}{h}\sum_{k=1}^{h} NP_k \qquad \textbf{(B.10)}$$

where

h = K/M

K = the number of samples used in the averaging interval.

By using the average noise power given in equation (B.10), C/No can be calculated as follows:

$$\frac{C}{No} = 10\log_{10}\left[\frac{1}{T_{accu}}\frac{\mu_{NP}-1}{M-\mu_{NP}}\right] \qquad \textbf{(B.11)}$$

As mentioned above, the averaging done in equation (B.10) should not happen during a navigation data bit transition. In such a case, the energy estimation may be erroneous which causes a wrong noise power and C/No ratio estimation for that interval. This problem can be solved by adjusting the begining of I and Q samples with the navigation data transition instant.