POSITIONING PERFORMANCES OF SINGLE FREQUENCY GPS, GLONASS AND CARRIER BASED ALGORITHMS IN A SOFTWARE PLATFORM

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

BY

CANER SAVAŞ

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

AUGUST 2014

Approval of the thesis:

POSITIONING PERFORMANCES OF SINGLE FREQUENCY GPS, GLONASS AND CARRIER BASED ALGORITHMS IN A SOFTWARE PLATFORM

submitted by **CANER SAVAŞ** 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. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. T. Engin Tuncer
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. T. Engin Tuncer
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Dept., METU _____

Dr. Uğur Kayasal
Roketsan Missiles Industries Inc. _____

Date: 29.08.2014

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: Caner SAVAŞ

Signature:

ABSTRACT

POSITIONING PERFORMANCES OF SINGLE FREQUENCY GPS, GLONASS AND CARRIER BASED ALGORITHMS IN A SOFTWARE PLATFORM

Savaş, Caner

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. T. Engin Tuncer

August 2014, 137 pages

There exist various algorithms (signal acquisition, tracking, etc.), methods (RTK (Real Time Kinematic), DGPS (Differential Global Positioning System), etc.) or aid systems (SBAS (Satellite Based Augmentation Systems), INS (Inertial Navigation System) aid etc.) to improve the GNSS (Global Navigation Satellite System) positioning accuracy. The positioning accuracy of different systems is usually determined by using a test setup which includes costly hardware and software units. In order to compare positioning performances of GPS, GLONASS and carrier based algorithms, namely LAMBDA (Least-squares Ambiguity De-correlation Adjustment) method, in a cost effective manner, a software platform is developed.

GNSS SDR (Software Defined Radio) receiver approach provides ease in modification of algorithms to meet the expectations of different types of applications. In the light of this approach, single L1 frequency GPS and GLONASS SDR algorithms are implemented in MATLAB. IF (Intermediate Frequency) sampled data is taken from the commercial IFEN GNSS receiver to test the algorithms. In order to improve the positioning accuracy, GPS-GLONASS combined solution and two

receiver positioning algorithms, namely LAMBDA method, for GPS L1 single frequency approach are implemented.

It is observed that GLONASS solution is in general less accurate than GPS solution. Furthermore, GPS-GLONASS combined solution approaches to GPS solution which reveals the fact that combined solution is useful when some GPS satellites are not observable. Moreover, a tracking lock status indicator in signal tracking loops is implemented to improve the accuracy for dynamic scenario. Finally, it is shown that LAMBDA method can achieve centimeter level accuracy in two-receiver configuration.

Keywords: GNSS SDR receiver, GPS-GLONASS Combined Solution, LAMBDA Method

ÖZ

YAZILIM PLATFORMUNDA TEK FREKANSTA GPS, GLONASS VE TAŞIYICI SİNYAL TABANLI ALGORİTMA UYGULAMALARI VE KONUMLANDIRMA PERFORMANSLARI

Savaş, Caner

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. T. Engin Tuncer

Ağustos 2014, 137 sayfa

GNSS (Küresel Navigasyon Uydu Sistemleri) konumlama doğruluğunu geliştirmek için çeşitli algoritmalar (sinyal yakalama, takip algoritmaları vs.), yöntemler (RTK (Gerçek Zamanlı Kinematik), DGPS (Diferansiyel Küresel Konumlandırma Sistemi, vs.) veya yardım sistemleri (SBAS (Uydu Tabanlı İyileştirme Sistemi), INS (Ataletsel Navigasyon Sistemi) yardımı vs.) bulunmaktadır. Farklı sistemlerin konumlandırma hassasiyeti genellikle maliyetli donanım ve yazılım birimleri içeren bir test kurulumu ile belirlenir. GPS, GLONASS konumlandırma algoritmaları ve taşıyıcı sinyal tabanlı algoritma uygulamalarından LAMBDA (En Küçük Kareler Belirsizlik Dekorelasyon Ayarı) metodunun doğruluk değerlerini karşılaştırmak amacıyla yazılım tabanlı GNSS alıcısı platformu geliştirilmiştir.

Yazılım tabanlı GNSS alıcısı, farklı alanlardaki uygulamalara yönelik ihtiyaçların karşılanabilmesi için algoritmaların değiştirilebilmesi olanağını sağlamaktadır. MATLAB ortamında GPS ve GLONASS sistemleri L1 frekansı sinyallerine uyumlu yazılım tabanlı alıcı algoritmaları oluşturulmuştur. Ticari IFEN GNSS alıcısından alınan, ara frekansta örneklenmiş veriler algoritmaları test etmek için kullanılmıştır. Konumlandırma hassasiyetini arttırmak amacıyla, GPS-GLONASS ortak çözüm

oluşturma algoritmaları ve iki alıcı konfigürasyonu kullanılarak GPS L1 frekansı için LAMBDA yöntemi uygulanmıştır.

GLONASS konumlandırma doğruluğunun GPS çözümüne göre daha az hassasiyette olduğu gözlenmiştir. GPS-GLONASS ortak çözümün doğruluk değerlerinin GPS doğruluğuna yakınsadığı ve dolayısıyla bazı GPS uydularının gözlemlenebilir olmadığı durumlarda ortak çözümün yararlı olduğu ortaya çıkmıştır. Dinamik senaryolar altında doğruluğu arttırmak için sinyal izleme kilit durum göstergesi sinyal izleme algoritmalarına eklenmiştir. Son olarak, iki alıcı konfigürasyonunda, konumu bilinen bir pozisyona göre, LAMBDA yöntemi uygulanarak santimetre düzeyinde doğruluk seviyesine ulaşıldığı gösterilmiştir.

Anahtar Kelimeler: GNSS SDR alıcısı, GPS-GLONASS Ortak Çözüm, LAMBDA Yöntemi

To my family

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere thanks to my supervisor Prof. Dr. T. Engin Tuncer for his complete guidance, advice, criticism and encouragement throughout the development of this thesis study.

I would like to express my appreciation to Roketsan Missiles Industries Inc. for the resources and the facilities that I utilized for this thesis study.

I am grateful to my colleague Mehmet Akif Koca for his help, advices and tolerance during this study.

I appreciate my colleague Lisan Ozan Yaman for the discussions throughout the development of this thesis study and for reviewing this thesis.

I would like to extend my special appreciation to my love Ebru İzgi for her love, support, patience and understanding throughout this study.

Last but not the least; I would like to express my deepest gratitude to my parents, Hatice and Necdet Savaş and my sister Birnur for their love, trust, understanding and permanent support throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

A/D	Analog to Digital Converter
BIH	Bureau International de l'Heure
BPSK	Binary Phase Shift Keying
C/A	Coarse Acquisition
CDMA	Code Division Multiple Access
DOP	Dilution of Precision
DSSS	Direct Sequence Spread Spectrum
FDMA	Frequency Division Multiple Access
FLL	Frequency Lock Loop
GLONASS	Global'naya Navigatsionnaya Sputnikovaya Sistema
GMT	Greenwich Mean Time
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IERS	International Earth Rotation Service
IF	Intermediate Frequency
ML	Maximum Likelihood
NCO	Numerically Controlled Oscillator
PLL	Phase Lock Loop
PPS	Precise Positioning Service
PRN	Pseudo Random Noise
RF	Radio Frequency
RMS	Root Mean Square
SDR	Software Defined Radio
SPS	Standard Positioning Service
TAI	International Atomic Time
TDMA	Time Division Multiple Access
USNO	United States Naval Observatory
UTC	Universal Time Coordinated

CHAPTER 1

INTRODUCTION

This chapter starts with the general background of GPS/GLONASS satellite systems, SDR approach, and the positioning techniques. The chapter continues with literature review related to this research. In section 1.3, objective of thesis is defined. At the end of this chapter, in section 1.4, outline of thesis is presented.

1.1 Background

GPS and GLONASS are two mostly used systems to get navigation solution in different applications such as land, marine, aviation, science, personal navigation etc. over worldwide.

The Global Positioning System (GPS) is a U.S.-owned utility that provides users with positioning, navigation, and timing (PNT) services [1].GPS developed by U.S. Department of Defense (DOD) started operating by launching of first 11 satellites between 1978 and 1985. IOC (Initial Operational Capability) was on 8th December, 1983. Now, nearly 30 satellites are operational in the constellation.

The Russian GPS called GLONASS (Global'naya Navigatsionnaya Sputnikovaya Sistema), had first satellite launch in 1982, and initial test was carried out in 1984. The GLONASS system was brought into operational testing in 1993 [2]. To 1995 the whole orbit group of 24 satellites was formed [2]. Then, GLONASS constellation system retrogressed to 7 satellites in the constellation for some time. Now, GLONASS has 24 operational satellites in the constellation.

The satellite and signal properties of GPS system compared to GLONASS are shown in Table 1.

Table 1 - Comparison of GPS and GLONASS System Properties [1], [2], [3]

	GPS	GLONASS
Orbital Properties	- 24 satellites[1] in 6 orbital planes - 55º orbital inclination - 26560km semi major axis - 11h 57.96min revolution period	- 24 satellites in 3 orbital planes - 64.8º orbital inclination - 25508km semi major axis - 11h 15.73min revolution period
Signal Properties	- Spread Spectrum System - CDMA (Code Division Multiple Access) L1:1575.42 MHz	- Spread Spectrum System - FDMA (Frequency Division Multiple Access) - L1:1598-1605 MHz
Reference/Time Systems	- WGS84 geodetic reference system - GPS Time, UTC (USNO – US Naval Observatory)	- PZ-90 geodetic reference system - GLONASS Time, UTC(SU) (Soviet UTC)
Ephemerides Data Properties	Kepler parameters, correction coefficients for satellite position calculations	Position, velocity and acceleration vectors of the satellite positions

[1] Since 2004, 28–31 spacecraft have been typically operational in GPS constellation with modernization.

Software-defined radios (SDRs) have been around for more than a decade [4]. The first complete SDR Global Positioning System (GPS) implementation was described by Dennis Akos in 1997 [4].

SDR implementations on global navigation satellite systems' applications have opened an area which provides flexible and adaptable architectures to satisfy the user's needs in terms of accuracy, robustness, cost etc. SDR architectures have expanding area of usage at both commercial products and military solutions. SDR technology's purpose is building a reconfigurable system so as to design new architectures and signal processing algorithms that can easily be adapted to modernization of the satellite systems and their signals and also that can meet the expectations of different types of applications.

Continuously and accurately tracking of the incoming signal in a GNSS receiver is necessary for producing accurate pseudorange values and shaping a position fix. In the design of GNSS receiver tracking loops, the type and order of the loop filters is a major concern so as to shape a robust receiver that is insensitive to limited acceleration and jerk values. Therefore, SDR approach provides flexibility in tracking algorithms depending on type of applications.

In different applications, better accuracy than provided by standard positioning techniques is required. To achieve centimeter level accuracy in positioning, carrier phase based algorithms are implemented. However, due to fact that it is not possible to measure the entire number of carrier cycles between receiver and the satellites, phase measurements include an unknown number of cycles. To eliminate ambiguity, with the positioning technique which uses carrier phase based observations, position is solved considering its observations and observations from a near reference position. Carrier cycle integer ambiguity resolution is an active area in the field of DGPS research [5].

1.2 Literature Review

This thesis utilizes two GNSS receivers (IFEN and Magellan) and a GNSS simulator (Spirent GSS8000), to obtain IF data, to compare positioning accuracies and to test algorithms under different scenarios. All algorithms have been implemented as software-based in MATLAB.

In this thesis, single L1 frequency GPS and GLONASS SDR algorithms are considered as an important insight into the solution of GNSS positioning and implemented as software. In the literature, SDR implementations on GNSS system applications have been an active research area. With the implementation of SDR on GPS by Dennis Akos in 1997, software based GNSS receivers opened a new research area. Reference 4 and 9 provide a complete overview on software approach. These approaches and algorithms have been a starting point for most researchers.

All implemented algorithms have been shaped with the help of the reference 5, fundamental GNSS principles and applications book by D. Kaplan. This book can be thought as a first read to understand complete navigation operations in detail. Reference 5 and also another primary book by Misra and Enge in reference 8 are widely utilized in this thesis work.

Through implementation of algorithms, matrix operations are needed to be examined in detailed. In reference 11, combination of linear algebra, geodesy and GPS algorithms by Gilbert Strang and Kai Borre offers a detailed theory of GPS applications.

Essentially, all algorithms are created considering GPS and GLONASS navigation user interface documents in reference 6 and 7.

GPS and GLONASS systems differ from each other at some points. In reference 3, by Hoffman, all GNSS systems (GPS, GLONASS, Galileo, etc.) are defined in detail and compared to each other. It is a great source to cover GNSS systems for now and possibly future.

In this thesis, improving the positioning accuracy only by implementing algorithms for GPS, GLONASS and combined GPS-GLONASS systems is one of

the goals. Moreover, carrier-phase based positioning algorithms (RTK – Real Time Kinematics) are examined to have positioning accuracy in centimeter level and a method for single L1 frequency approach is created by considering reference methods. In reference 8, by Misra and Enge, carrier phase precise positioning methods and models are described in detail. Reference 15, by Teunissen, offers the LAMBDA method steps in detail. Reference 15 is the most important and comprehensive source for LAMBDA method and reference 12, 13 and 14 can be seen as helpful intermediate steps to understand whole process given in reference 15.

1.3 Objectives of the Thesis

In this thesis, in the light of implemented algorithms in MATLAB, presenting a comparison of positioning performances of GPS and GLONASS systems is aimed. Additionally, positioning accuracy of combined GPS-GLONASS solution is intended to be analyzed in terms of accuracy.

Although MATLAB algorithms are not able to provide a real-time solution, implemented single L1 frequency GPS and GLONASS SDR algorithms have been an important insight into the solution of GNSS positioning. Signal acquisition, tracking, navigation demodulation techniques have been performed in post processing using MATLAB. Thanks to flexibility of SDR approach, performance of signal processing algorithms with different parameters and architectures can be observed.

In order to improve the positioning accuracy in centimeter level, two receiver carrier-phase based positioning algorithms (RTK – Real Time Kinematics) are applied.

1.4 Outline of the Thesis

Chapter 1 provides brief information about this thesis study and literature review related to the research.

Chapter 2 presents GPS system background in terms of its L1 signal properties, transmitted satellite data and signal acquisition/tracking loop architectures. MATLAB algorithms' outputs in signal acquisition/tracking and navigation data extraction level are included. Under static and dynamic scenarios, positioning performance of algorithms compared to the commercial GNSS receiver and actual simulator data is shared.

Chapter 3 provides the fundamental characteristics of GLONASS system. With the knowledge on GLONASS L1 signal characteristics and transmitted satellite data, acquisition/tracking and navigation demodulation algorithms are presented. Under the same static and dynamic scenarios that the GPS algorithms are tested, comparison of GLONASS positioning performance and simulator data is given.

Chapter 4 presents the combined GPS-GLONASS algorithm and simulation results including the comparison of GPS-only and GLONASS-only positioning algorithms under same dynamic and static scenarios to compare positioning performances.

Chapter 5 gives information about carrier based algorithms, and theory of LAMBDA method for single frequency. Details of the implemented algorithms and field test results in two-receiver configuration are presented.

Chapter 6 presents summary and draws conclusions of the work on this thesis.

CHAPTER 2

GPS POSITIONING ALGORITHMS

This chapter starts with the general overview of a GNSS receiver and the software approach for implementation. In second section of this chapter, GPS signal structure and characteristics are shared. The question of how to find the user position is answered in the third section. In the subsections of section 2.3, signal acquisition, tracking, and processing of the extracted navigation data to compute user position are explained in detail. In the last section, simulation results of the implemented algorithms under static and dynamic scenario are presented.

2.1 GNSS Receiver Overview

The basic concept of a GNSS receiver is to compute the user position or provide some measurements that can be used for navigation.

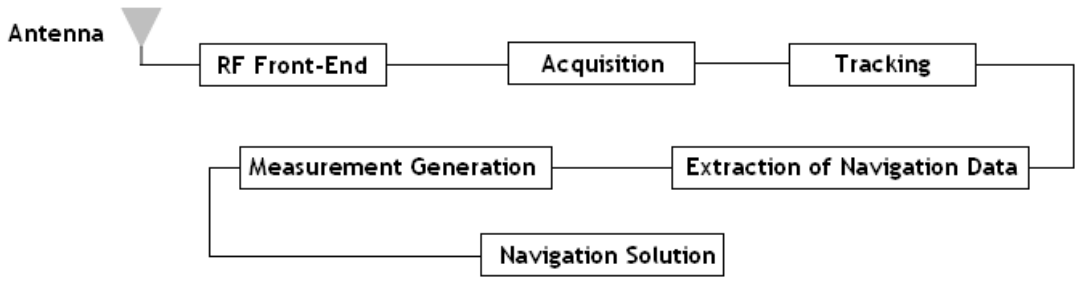

Figure 1 - General GNSS Receiver Overview

A general GNSS receiver overview in block diagram is shown in Figure 1. GNSS signal is received by an antenna. The received signal is pre-amplified, down converted and sampled by RF front end. Digitized signal is processed using hardware in a conventional GNSS receiver. Software radio concept gives a chance of processing the sampled IF data using a software. Performing signal acquisition and tracking using software will be helpful to introduce new methods without changing any hardware design. After acquisition and tracking, navigation data is extracted. Finally, user position is computed using navigation data and the implementation of measurement models.

2. 2 Signal Characteristics

A PSK (Phase shift keying) modulated digital signal $s_m(t)$ with M possible phases can be represented as following [17]:

$$s_m(t) = g(t)\cos\left[\frac{2\pi}{M}(m-1)\right]\cos[2\pi f_c t] - g(t)\sin[\frac{2\pi}{M}(m-1)]\sin[2\pi f_c t] \quad 2.1$$

$$\theta_m = \frac{2\pi(m-1)}{M}, m = 1,2\ldots,M$$

GPS signal is a BPSK (bi-phase shift keying) modulated signal with $\phi = \pm\pi$ and BPSK modulation on GPS signal is illustrated in Figure 2.

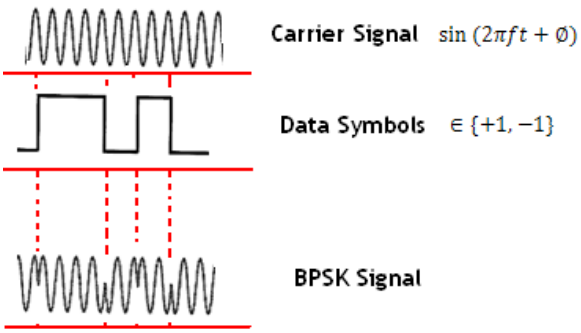

Figure 2 - BPSK Modulation on GPS Signal

A GPS signal transmitted from satellite "k" can be written as follows [4]:

$$s^k(t) = \sqrt{2P_C}\left(C^k(t) \oplus D^k(t)\right)\cos(2\pi(f_{L1} + \Delta f_{SV})t + \phi) + \sqrt{2P_P}\left(P^k(t) \oplus D^k(t)\right)\sin(2\pi(f_{L1} + \Delta f_{SV})t + \phi) \quad 2.2$$

$s^k(t)$: $Transmitted\ signal\ from\ satellite\ k$

P_C: $C/A\ code\ signal\ power$

C^k: $Coarse/\ Acquisition\ (C/A)\ code\ with\ chipping\ rate\ of\ 1.023\ Mcps$

D^k: $Navigation\ data\ message\ at\ 50\ bps$

f_{L1}: $L1\ carrier\ frequency\ (1575.42\ MHz)$

Δf_{SV}: $Satellite\ frequency\ offset$

ϕ: $Small\ phase\ noise\ and\ oscillator\ drift\ component\ (Carrier\ phase\ offset)$

P_P: $P\ code\ signal\ power$

P^k: $P\ code, \pm 1\ pseudorandom\ sequence\ with\ a\ clock\ rate\ of\ 10.23\ Mcps$

Figure 3 shows the vector phase diagram of P and C/A coded signal given in equation 2.2. As seen in Figure 3, there is 90° phase difference between C/A-coded data signal and P-coded data signal.

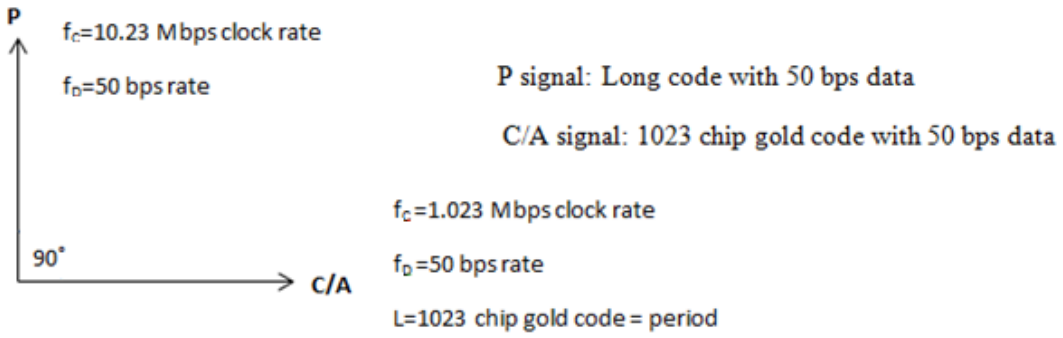

Figure 3 - GPS L1 Signal Structure

Precision P(Y) code is not accessible by civil users; it is only available to classified users, primarily military. Hence, algorithms in this thesis are focused on C/A (Coarse/Acquisition) code.

All GPS satellites' signals use same center frequency for L1 signal. They have different C/A codes. Main characteristics of CDMA are given in Table 2.

Chipping rate of C/A modulation code ($1.023\ Mcps$) is higher than the data rate ($50\ bps$). (DSSS- Direct Sequence Spread Spectrum)

Therefore, the GPS signal can be defined as CDMA, using DSSS (Direct Sequence Spread Spectrum), bi-phase modulated signal.

Table 2 - Main Characteristics of Code Division Multiple Access [Adapted from 16]

Code Division Multiple Access **(CDMA)**	Advantages	Disadvantages
Transmit with unique pseudorandom noise sequences	• Efficient use of spectrum • Full bandwidth for all users all the time • Inherent security (if PRN code is secret) • Inherent resistance to NB interference (due to spread spectrum) • Inherent time synchronization	• Complex baseband processing • Subject to 'self-interference'

2.3 Navigation Signal Processing

GNSS receiver is created as a multi-channel processing structure. Every acquired satellite initializes a channel until no available channel exists to process signals. After acquisition, code and carrier signals of the enabled channels are tracked and the navigation data is decoded. Using navigation data and implementing positioning algorithms, user position is computed as summarized in Figure 4 below.

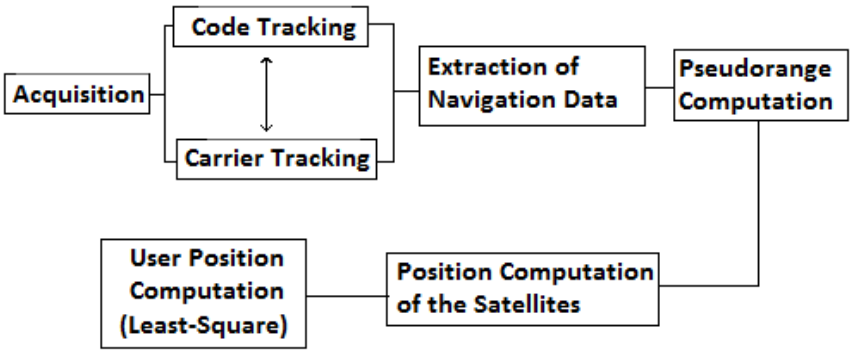

Figure 4 - Position Computation Block Diagram

2.3.1 Signal Acquisition

The aim of signal acquisition is to find the visible satellites, deviation of IF frequency from the actual value and code phase value of the related satellite signals.

The GPS C/A code belongs to the family of Pseudorandom Noise (PRN) codes known as Gold Codes [9]. For each satellite, there is a different C/A code sequence. Its phase changes between 0 and π. Locally generated PRN code is aligned with the incoming signal code and so that the signal is decoded as shown in Figure 5.

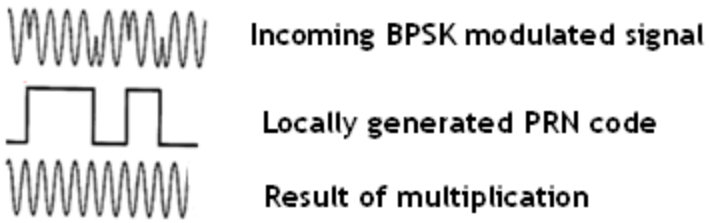

Figure 5 - Code Demodulation

The signal transmitted from satellite "k" and induced at the antenna can be modeled as follows:

$$s_{ant}^{k}(t) = A_{ant,I}^{k}(t)P_{ant}^{k}\left(t-\tau_{ant,P}^{k}\right)D_{ant}^{k}\left(t-\tau_{ant,P}^{k}\right) \qquad 2.3$$
$$\sin\left(2\pi\left(f_{L1}+\Delta f_{SV,ant}+\Delta f_{D,ant}\right)t+\phi_{ant}^{k}\left(t-\tau_{ant,L1}^{k}\right)\right)$$
$$+A_{ant,Q}^{k}(t)C_{ant}^{k}\left(t-\tau_{ant,C/A}^{k}\right)D_{ant}^{k}\left(t-\tau_{ant,C/A}^{k}\right)$$
$$\cos\left(2\pi\left(f_{L1}+\Delta f_{SV,ant}+\Delta f_{D,ant}\right)t+\phi_{ant}^{k}\left(t-\tau_{ant,L1}^{k}\right)\right)$$

$s_{ant}^{k}(t)$: $Received\ signal$

$\tau_{ant,C/A}^{k}$: $C/A\ \ code\ propagation\ delay$

$\tau_{ant,P}^{k}$: $P\ code\ propagation\ delay$

$\Delta f_{D,ant}$: $Doppler\ frequency\ offset$

$\tau_{ant,L1}^{k}$: $L1\ carrier\ propagation\ delay$

$\phi^{ant}(t)$: $Time\ dependent\ carrier\ phase$

$-\ models\ signal\ delays\ (e.g.ionospheric\ effects)$

Simplified IF signal after RF front-end (filtering and down-conversion) part is:

$$s_i^k(t) = A_{I,i}\, P_i^k(t-\tau) D_i^k(\mathrm{t}-\tau)\sin(2\pi(f_{IF}+\Delta f_i)+\phi_i) + A_{Q,i}\, C_i^k(t\tau) D_i^k(\mathrm{t}-\tau)\cos(2\pi(f_{IF}+\Delta f_i)+\phi_i) \quad 2.4$$

f_{IF}: $Intermediate\ carrier\ frequency\ downconverted\ by\ front-end$

A/D conversion and filtering is applied at the RF front-end and the signal can be modeled as on equation 2.5. It is assumed that the receiver is using C/A code and there exists a narrow band-pass filter around C/A to distort P code.

$$s_i^k(n) = C_i^k(nT_s-\tau) D_i^k(\mathrm{n}T_s-\tau)\cos(2\pi(f_{IF}+\Delta f_i)nT_s+\phi_i(nT_s-\tau)) + w(n) \quad 2.5$$

Simplifying the equation 2.5 as follows to express the acquisition procedure, we obtain:

$$s_i^k(n) = C_i^k(n) D_i^k(\mathrm{n})\cos(\omega_{IF} n) + w(n) \quad 2.6$$

When the input signal is multiplied with the locally generated carrier signal, the following components are obtained:

$$s_i^k(n)cos(\omega_{IF} n) = C_i^k(n) D_i^k(n) cos(\omega_{IF} n) cos(\omega_{IF} n) \quad 2.7$$

$$= \frac{1}{2} C_i^k(n) D_i^k(n) + \frac{1}{2} C_i^k(n) D_i^k(n) cos(2\omega_{IF} n)$$

$$s_i^k(n)sin(\omega_{IF} n) = C_i^k(n) D_i^k(n) sin(\omega_{IF} n) cos(\omega_{IF} n) \quad 2.8$$

$$= \frac{1}{2} C_i^k(n) D_i^k(n) sin(2\omega_{IF} n)$$

In order to remove the C/A code from the in-phase and quadrature signals above, locally generated C/A code is used.

C/A code is periodic and repeats constantly every millisecond as a fixed length of 1023 chips. If one millisecond of signal is gathered, the starting point of the C/A code is needed to be found, in other words, incoming signal code phase is needed to be searched through 1023 different code phases.

At the same time, assuming that the frequency deviation from actual IF frequency value is up to ±14 kHz and the error in frequency search is within 500 Hz, 57 possible frequencies should be searched for each satellite signal.

In this acquisition step, a circular correlation between locally generated code and the incoming signal is applied, instead of implementing multiplication with 1023 different code phases in linear search. Correlation result is compared to a defined threshold as in the case of ML estimation to decide whether signal is acquired or not.

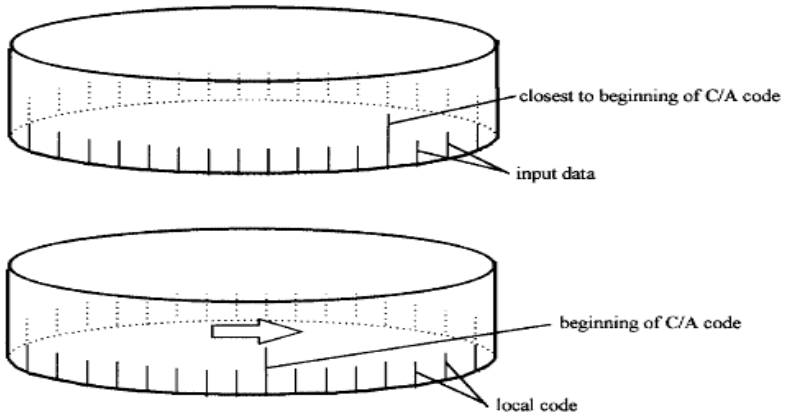

Figure 6 - Periodic correlation of incoming signal code with locally generated one [Retrieved from 9]

Figure 6 illustrates the circular correlation between locally generated C/A code and incoming coded signal in time domain.

$$R_{x,y}[m] = \sum_{n=0}^{L-1} x(n+m)y(n) \; - Correlation\ between\ x(n)\ and\ y(n) \tag{2.9}$$

$x(n) : Incoming\ signal$

$y(n) : Locally\ generated\ code$

Cross-correlation result is found in frequency domain by performing Fourier transform of the equation 2.9.

FFT-implemented correlation is circular due to periodicity of DFT. Fourier transform result will have a distinct peak at the IF frequency index of the carrier signal when the locally generated code is aligned with the incoming one.

Hence, by applying inverse FFT, time domain result is obtained.

$$R_{x,y}[m] = \mathcal{F}^{-1}(X(k)Y^*(k)) \qquad 2.10$$

To sum up, by implementing the acquisition block diagram shown in

Figure **7**, carrier frequency and code phase values of the satellite signals are obtained.

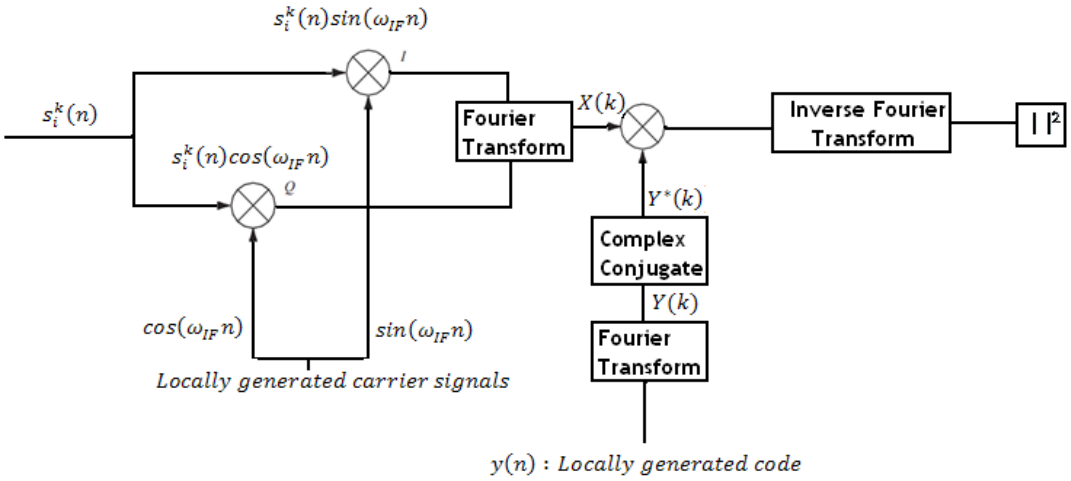

Figure 7 - Acquisition – Parallel Code Phase Search Algorithm Block Diagram [Adapted from Reference-4]

I and Q terms in Figure 7 are the signals given in the equation 2.8 which are obtained by multiplying the input signal with the locally generated carrier signal.

Searching 57 possible frequencies for each satellite is performed by applying the software of the acquisition block diagram 57 times.

In phase and quadrature signals obtained in equation 2.7 and 2.8 are combined as following:

$$x(n) = I(n) + jQ(n) \qquad 2.11$$
$$Q(n) = s_i^k(n)cos(\omega_{IF}n)$$
$$I(n) = s_i^k(n)sin(\omega_{IF}n)$$

$x(n)$ in equation 2.11 is the same with the $x(n)$ in equation 2.9. Finally code phase search is realized as circular cross-correlation in frequency domain for each step as explained above.

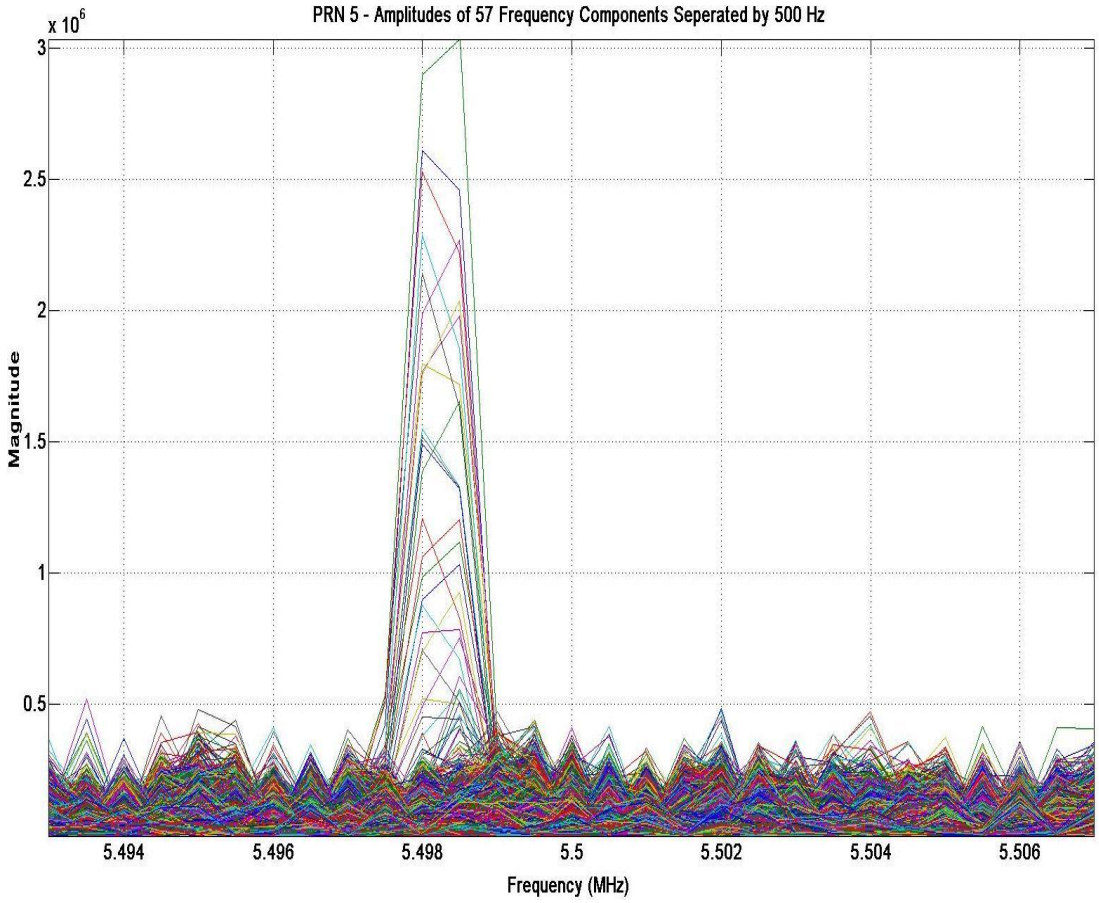

Figure 8 - GPS Acquisition – Magnitudes of 57 Frequency Components Separated by 500 Hz

In Figure 8, computed cross-correlation magnitudes belong to each carrier frequency value (±14 kHz in 500Hz steps) for PRN5 satellite signal are plotted on the same graph. Locally generated signals are created for 57 frequency components separated by 500 Hz to find the related IF value for the carrier signal.

The same manner with Figure 8, applying circular correlation, the beginning of C/A code for PRN 5 is found with the significant peak seen in Figure 9 after cross-correlation.

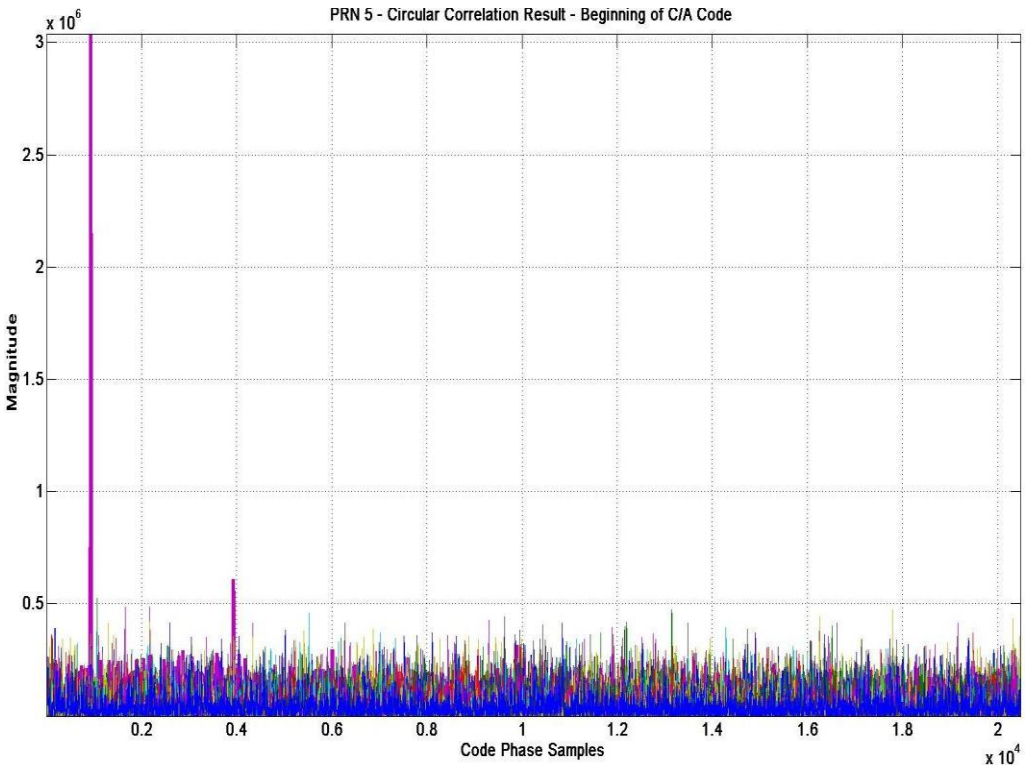

Figure 9 - GPS Acquisition – Magnitudes of Cross Correlation Results for C/A code Phase Computation

Carrier frequency and code phase values for PRN 5 in acquisition process can be obtained from Figure 8 and Figure 9.

Through acquisition process, IF carrier frequencies and code phase values of all acquired satellites are shared in Table 3.

Table 3 - Acquired Satellites with IF Carrier Frequency and Code Phase Values

Channel Number	Satellite Number	IF Carrier Frequency (Hz)	Code Phase (Samples)
1	15	5.502392578125000e+006	15344
2	8	5.500185546875000e+006	10053
3	5	5.500839843750000e+006	485
4	2	5.500898437500000e+006	11046
5	9	5.500566406250000e+006	17650
6	4	5.497578125000000e+006	18236
7	10	5.498710937500000e+006	15021
8	29	5.501601562500000e+006	19776

However, IF and sampling frequency values of the sampled IF data which is obtained from IFEN GNSS receiver and used through MATLAB algorithms are as follows:

- IF frequency: 5.499998474121090e+06 Hz
- Sampling frequency: 20480000 Hz
- IF sample format : 2 - bit samples

As seen in Table 3, satellite signals have nearly same IF center frequencies as expected due to CDMA.

2.3.2 Signal Tracking

In the acquisition process, carrier frequency and code phase values of the visible satellites' signals are obtained.

The idea behind signal tracking is that stripping off the carrier and C/A code by following signal to obtain whole navigation data. The values gathered from the acquisition process are the first rough estimates of the tracking process.

Both the carrier frequency and C/A code will be affected from the motion of user and satellites, in other words, *Doppler Effect*. Therefore, so as to obtain the satellite navigation data, the carrier frequency and C/A code are needed to be tracked in two different tracking loops as seen on the Figure 4.

2.3.2.1 Carrier Tracking

Figure 10 shows the block diagram of carrier tracking. The main goal of carrier tracking is wiping off the carrier generating a perfectly aligned carrier.

Locally generated C/A code should be aligned to the coming signal, and therefore; C/A code in the Figure 10 is get from the code tracking loop which will provide an aligned one.

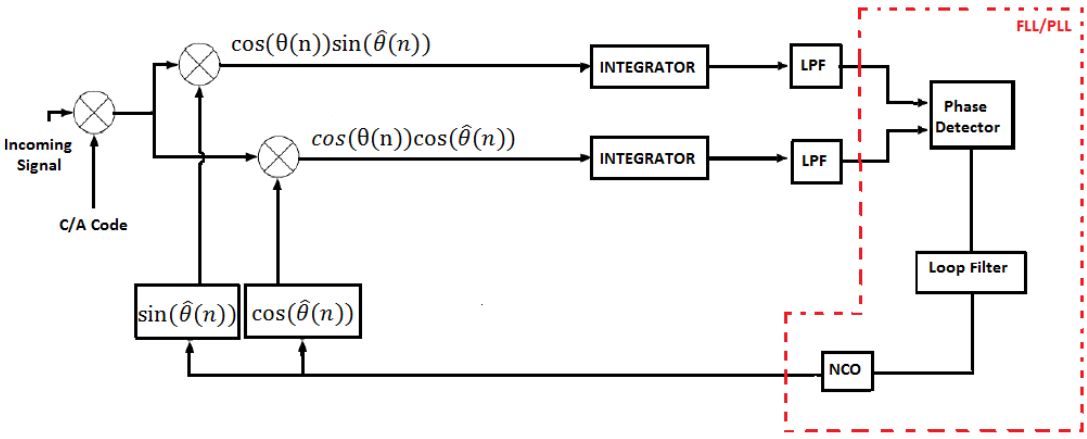

Figure 10 - Block Diagram of Carrier Signal Tracking

Consider the incoming signal as modeled in equation 2.6:

$$s_i^k(n) = C_i^k(n)D_i^k(\mathrm{n})\cos(\omega_{IF}n) + w(n)$$

After C/A code cross-correlation is applied, C/A code is being stripped off. Then the signal is multiplied with locally generated signals.

In phase component can be written as,

$$D_i^k(\mathrm{n})\cos(\omega_{IF}n)\cos(\omega_{IF}n+\theta) \qquad 2.12$$
$$= \frac{1}{2}D_i^k(\mathrm{n})\cos(2\omega_{IF}n+\theta) + \frac{1}{2}D_i^k(\mathrm{n})\cos(\theta)$$

Quadrature component is,

$$D_i^k(\mathrm{n})\cos(\omega_{IF}n)\sin(\omega_{\mathrm{IF}}n+\theta) \qquad 2.12$$
$$= \frac{1}{2}D_i^k(\mathrm{n})\sin(2\omega_{IF}n+\theta) - \frac{1}{2}D_i^k(\mathrm{n})\sin(-\theta)$$

1ms integration and low pass filter are applied to in-phase and quadrature signals. After low pass filter, the signals can be written as follows,

$$I_i^k(\mathrm{n}) = \frac{1}{2}D_i^k(\mathrm{n})\cos(\theta) \qquad 2.13$$

$$Q_i^k(n) = \frac{1}{2}D_i^k(\mathrm{n})\sin(\theta) \qquad 2.14$$

Then, in-phase and quadrature signals are fed into the phase detector.

There are different types of discriminators. In Table 4, chosen discriminators are shared for PLL and FLL. PLL discriminator computes the difference between the

phases of incoming signal and locally generated signal. FLL discriminator tracks the change in carrier phases. Table 4 gives the details of these units.

Table 4 - Discriminators for FLL and PLL [4], [5], [18]

Tracking Loop	Discriminator
PLL (Phase Locked Loop)	$D = ATAN(Q_i^k, I_i^k)/(2 \cdot pi)$
FLL (Frequency Locked Loop)	$D = ATAN2(Cross, Dot)/(t2 - t1) \cdot (2 \cdot pi)$ $Cross = I_{i,t1}^k \cdot Q_{i,t2}^k - I_{i,t2}^k \cdot Q_{i,t1}^k$ $Dot = I_{i,t1}^k \cdot I_{i,t2}^k + Q_{i,t1}^k \cdot Q_{i,t2}^k$ Samples taken above belong to t1 and t2 times. $i.e. t2 - t1 = 5\ msec$

Discriminator or phase detector outputs are fed into the FLL/PLL loop filter.

In GPS signal tracking, 3rd order PLL assisted with 2nd order FLL filter is implemented. Considering the motions which include high dynamic maneuvers PLL is selected as 3rd order which will provide insensitivity to acceleration. An assisted wideband FLL will offer reducing excess frequency error in pull-in states.

Figure 11 shows the digital version of the 3rd order PLL assisted with 2nd order FLL filter. It is converted from analog form using bilinear transformation. Frequency error and phase error values taken from phase detector are shown as input in Figure 11 below and the circles are used to indicate the newly named variables for simplicity.

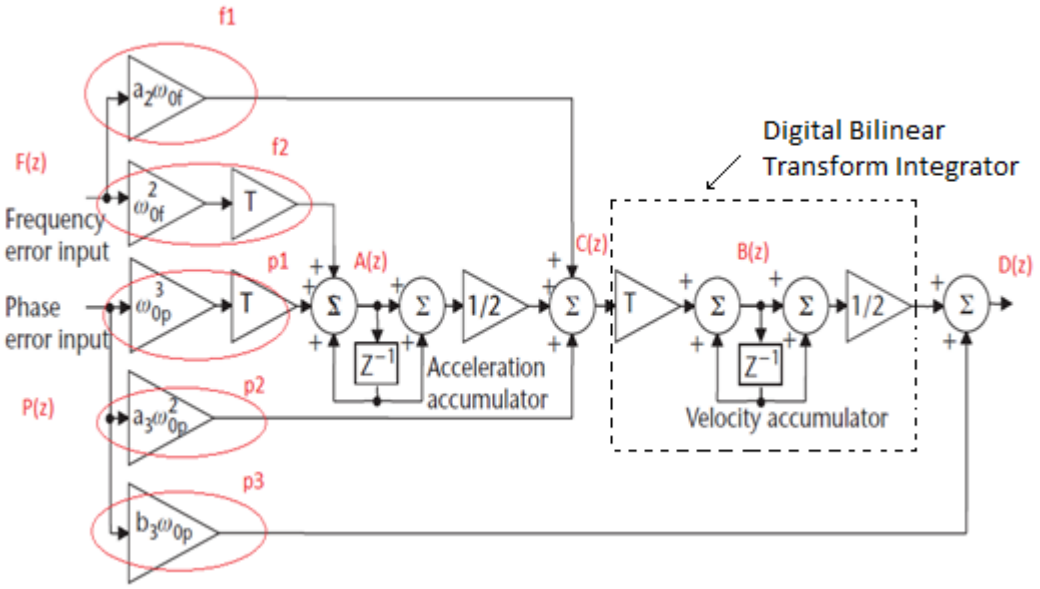

Figure 11 - Figure 3rd Order Digital PLL Filter Assisted with 2nd Order Digital FLL Filter [Adapted from Reference 5]

The filter is simplified to implement in MATLAB. Filter values and noise bandwidth conversions given in equations 2.15 and 2.16 are selected considering the reference [5] due to the fact that the limitations were determined through Monte Carlo simulations. The loop filter order and bandwidth values are chosen experimentally. Determined noise bandwidths of FLL and PLL filters are 25 Hz and 15 Hz, respectively.

Computation of the loop filter values of PLL and FLL shown on the figure above is through following equation sets:

$$B_{n_{FLL}} = \frac{\omega_{0f}(1 + a_2^2)}{4 \cdot a_2} = 0.5286 \cdot \omega_{0f} \qquad B_{n_{FLL}} : FLL\ Noise\ BW\ (Hz) \qquad 2.15$$

$$\omega_{0f} = \frac{8\zeta B_{n_{FLL}}}{4\zeta^2 + 1} = B_{n_{FLL}} \cdot 1.8919 \qquad \omega_{0f} : Natural\ frequency$$

$$\zeta : Damping\ ratio$$

$$f_2 = \omega_{0f}^2 \cdot T \qquad T : Time\ difference\ between\ each\ sample$$

$$f_1 = a_2 \cdot \omega_{0f} = 1.414 \cdot \omega_{0f}$$

$$B_{n\_PLL} = 0.7845 \cdot \omega_{0p} \qquad B_{n\_PLL} : PLL\ Noise\ BW\ (Hz) \qquad 2.16$$

$$p_1 = \omega_{0p}^3 \cdot T \qquad T : Time\ difference\ between\ each\ sample$$

$$p_2 = a_3 \cdot \omega_{op}^2 = 1.1\ \cdot \omega_{op}^2$$

$$p_3 = b_3 \cdot \omega_{0p} = 1.414\ \cdot \omega_{0p}$$

After loop coefficients are computed, considering the loop filter in Figure 11, we can write:

$$F(z) \cdot f_2 + P(z) \cdot p_1 + A(z) \cdot z^{-1} = A(z) \qquad 2.17$$

$$\frac{1}{2}(A(z) + A(z) \cdot z^{-1}) + F(z) \cdot f_1 + P(z) \cdot p_2 = C(z) \qquad 2.18$$

Multiplying both sides of the equation 2.17 by $\frac{1}{2}(1 + z^{-1})$ yields:

$$\frac{1}{2}A(z)(1 + z^{-1}) = \frac{1}{2}A(z)(z^{-1} + z^{-2}) + \frac{1}{2}F(z)f_2(1 + z^{-1}) + \frac{1}{2}P(z)p_1(1 + z^{-1}) \qquad 2.19$$

In equation 2.19, the term $\frac{1}{2}A(z)(z^{-1} + z^{-2})$ can be replaced using equation 2.18 multiplied by z^{-1}:

$$\frac{1}{2}A(z)(z^{-1} + z^{-2}) + F(z)f_1z^{-1} + P(z)p_2z^{-1} = C(z)\ z^{-1}$$

$$\frac{1}{2}A(z)(z^{-1} + z^{-2}) = C(z)z^{-1} - F(z)f_1z^{-1} - P(z)p_2z^{-1} \qquad 2.20$$

Using equation 2.19 and 2.20, equation 2.18 can be written as following:

$$C(z)z^{-1} - F(z)f_1z^{-1} - P(z)p_2z^{-1} + \frac{1}{2}F(z)f_2(1 + z^{-1}) + \frac{1}{2}P(z)p_1(1 + z^{-1}) + F(z) \cdot f_1 + P(z) \cdot p_2 = C(z) \qquad 2.19$$

Equation 2.19 can be put into the following form:

$$\left[\frac{1}{2}p_1 + p_2\right] \cdot P(z) + \left[\frac{1}{2}p_1 - p_2\right] \cdot P(z) \cdot z^{-1} + \left[\frac{1}{2}f_2 + f_1\right] \cdot F(z) + \left[\frac{1}{2}f_2 - f_1\right] \cdot F(z) \cdot z^{-1} + C(z) \cdot z^{-1} = C(z) \quad 2.20$$

In time domain, equation 2.20 can be expressed as:

$$c(n) = c(n-1) + \frac{1}{2}f_2 \cdot [f(n) + f(n-1)] + f_1 \cdot [f(n) - f(n-1)] + \frac{1}{2}p_1 \cdot [p(n) - p(n-1)] + p_2 \cdot [p(n) - p(n-1)] \quad 2.21$$

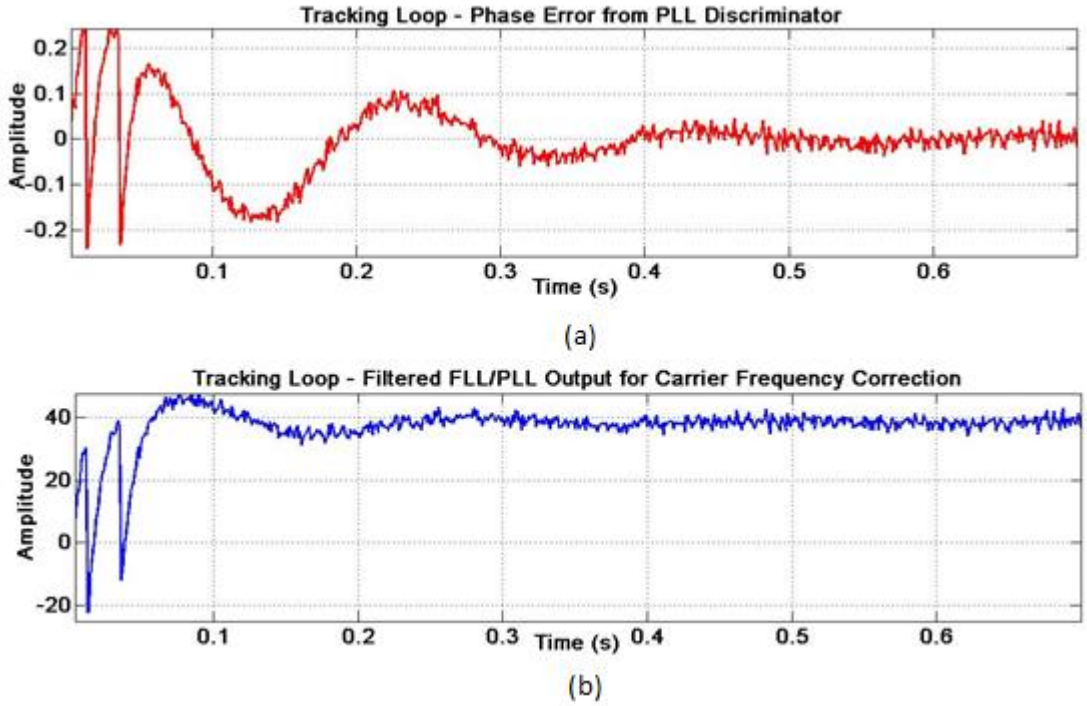

Figure 12 - Tracking Loop Outputs in Time for PRN-5 Satellite Signal: (a) Computed Phase Error Value from PLL Discriminator Output (b) Carrier Frequency Correction Obtained from Filtered FLL/PLL Output in Time

In Figure 12, computed phase errors from PLL discriminator in time and FLL/PLL loop filter output which will be fed for frequency correction are shown. The filter accumulates the computed difference over time and keeps it. As seen on the second subplot, there is nearly 40 Hz frequency differences between generated and incoming signal, initially.

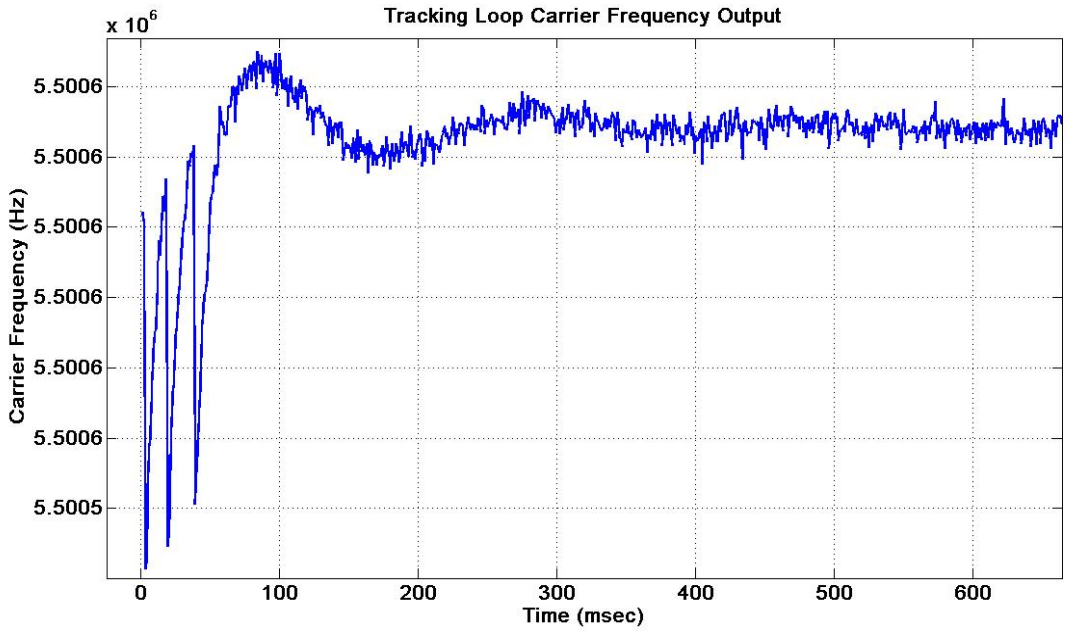

Figure 13 – Tracking Loop – Change in Carrier Frequency Output Over Time for PRN-5 Satellite Signal

In Figure 13, change in the carrier frequency of PRN-5 satellite signal in receiver channel 3 in tracking process is shown. The initial value is taken from acquisition output and then settling of carrier frequency occurred over time. FLL/PPL tracking loop output provides the correction on it over time by tracking the signal.

When the filter loses signal tracking at a point, it is needed to be detected and navigation bit decoding should be stopped. Especially, in high dynamic applications, the filter will encounter high Doppler frequency values and lose lock easily. In order to avoid this and provide robust signal tracking, two software indicators are created for PLL and FLL. Tracking lock status indicators are shared in Table 5.

Table 5 - Tracking Lock Status Indicators for FLL and PLL [18]

Tracking Loop	Lock Indicators
PLL (Phase Locked Loop)	$PLL_I = \frac{(I_i^k)^2 - (Q_i^k)^2}{(I_i^k)^2 + (Q_i^k)^2}$
FLL (Frequency Locked Loop)	$Cross = I_{i,t1}^k \cdot Q_{i,t2}^k - I_{i,t2}^k \cdot Q_{i,t1}^k$ $Dot = I_{i,t1}^k \cdot I_{i,t2}^k + Q_{i,t1}^k \cdot Q_{i,t2}^k$ $FLL_I = \frac{(Dot)^2 - (Cross)^2}{(Dot)^2 + (Cross)^2}$

While FLL aided PLL operates, at first FLL is initialized and it tries to achieve lock applying high pulling-in range in frequency. It provides preventing a false lock of PLL. After carrier tracking is operated as FLL alone, FLL assisted PLL operation starts. When FLL input error becomes zero, the loop filter behaves as if a pure PLL. Figure 14 shows PLL/FLL tracking indicators with decoded navigation bit from code tracking loop synchronized in time. Numbered intervals indicate the mentioned operation states above.

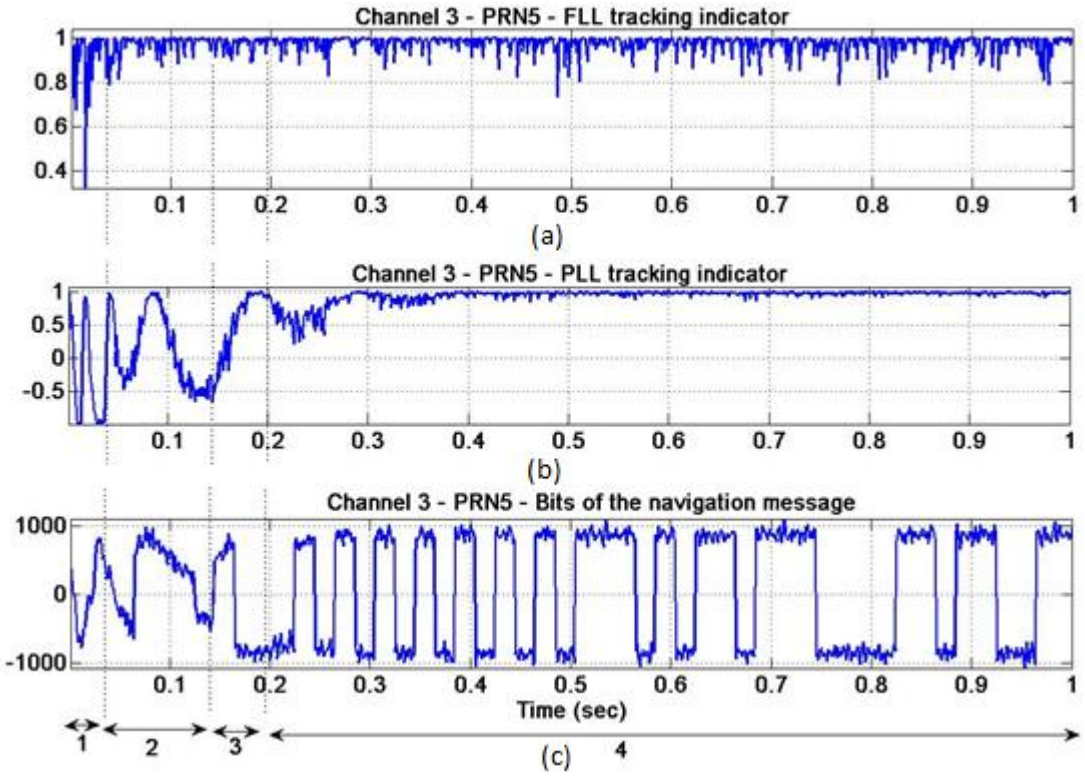

Figure 14 – Signal Tracking Loop Outputs Synchronized in Time: (a) FLL Tracking Indicator Value (b) PLL Tracking Indicator Value (c) Decoded Unnormalized Navigation Bits From DLL IP (In-phase Prompt) Correlator Output

2.3.2.2 Code Tracking

Figure 15 shows the block diagram of code tracking, namely DLL (Delay Lock Loop) with six correlators and carrier tracking loop together.

The idea behind the collaboration of carrier and code tracking loops is that

- using aligned carriers in C/A code stripping in code tracking loop and

- utilizing aligned C/A codes to remove carrier signal synchronously in carrier tracking loop.

Therefore, to reach accurate navigation data, collaboration of carrier and code tracking is required.

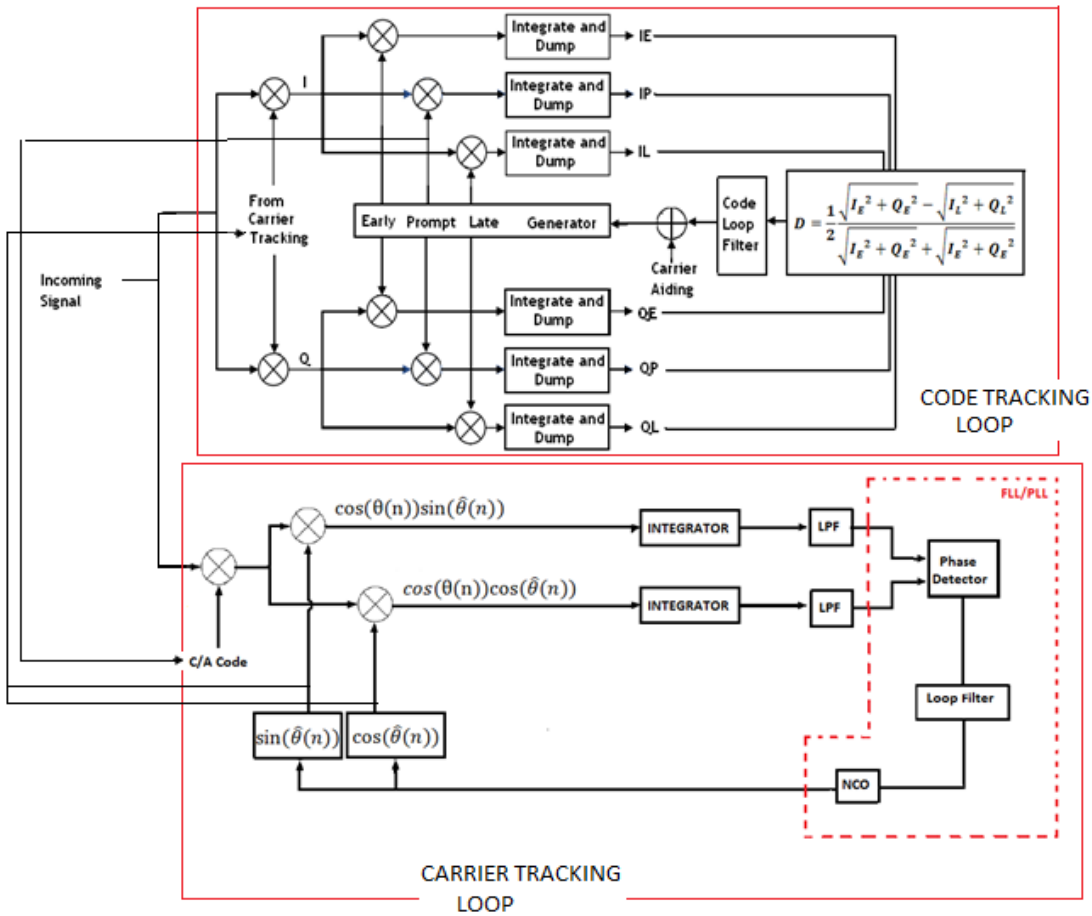

Figure 15 – Block Diagram of Signal Code Tracking

Incoming signal seen in Figure 15 is the same signal which is modeled in equation 2.6 $[s_i^k(n) = C_i^k(n)D_i^k(\mathrm{n})\cos(\omega_{IF}n) + w(n)]$.

In code tracking, firstly, the signal is multiplied with locally generated signals which come from carrier tracking loop as seen in Figure 15. Then using early-prompt-late generator, locally generated C/A codes are multiplied with the incoming signal.

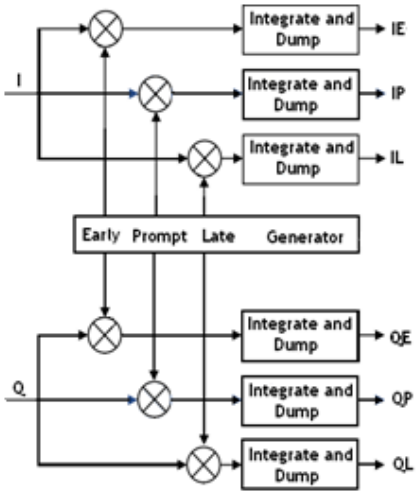

Figure 16 – Code Tracking Loop – Block Diagram of Early/Prompt/Late Generator and Correlator Outputs

Figure 16 shows the early-prompt-late generator and correlator outputs. In early-prompt-late generator, generated C/A codes have half chip spacing between each other. Considering the highest correlator output of three correlations for each I (IE,IP,IL) and Q (QE,QP,QL) arms, generated C/A code is tried to be aligned with the incoming C/A code over time.

As seen in Figure 17, when highest correlation value belongs to promptly generated code namely IP, code tracking is achieved. This result is consistent with the fact that if the locally generated carrier wave is in phase with the incoming signal, all the energy is in the in-phase arm. Figure 17 shows this fact. As expected and seen in Figure 18, energy in quadrature arm decreases with the achievement of signal tracking. As seen in Figure 17 and Figure 18, after some time code tracking is achieved and it also means that locally generated signal is in phase with the incoming signal.

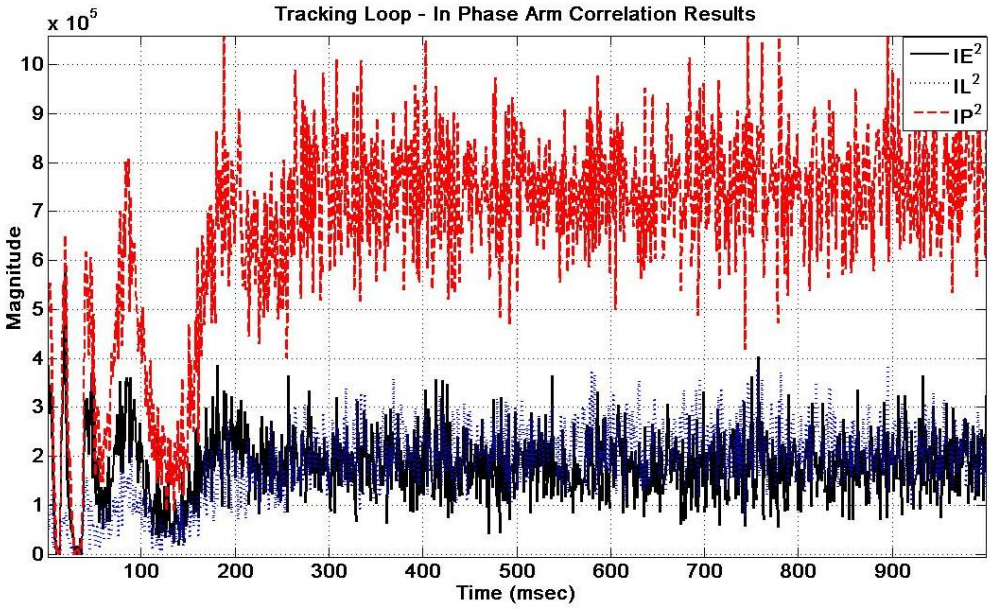

Figure 17 – Tracking Loop – Comparison of In-Phase Arm Correlation Results namely I-Early, I-Prompt and I-Late Correlator Outputs in Time

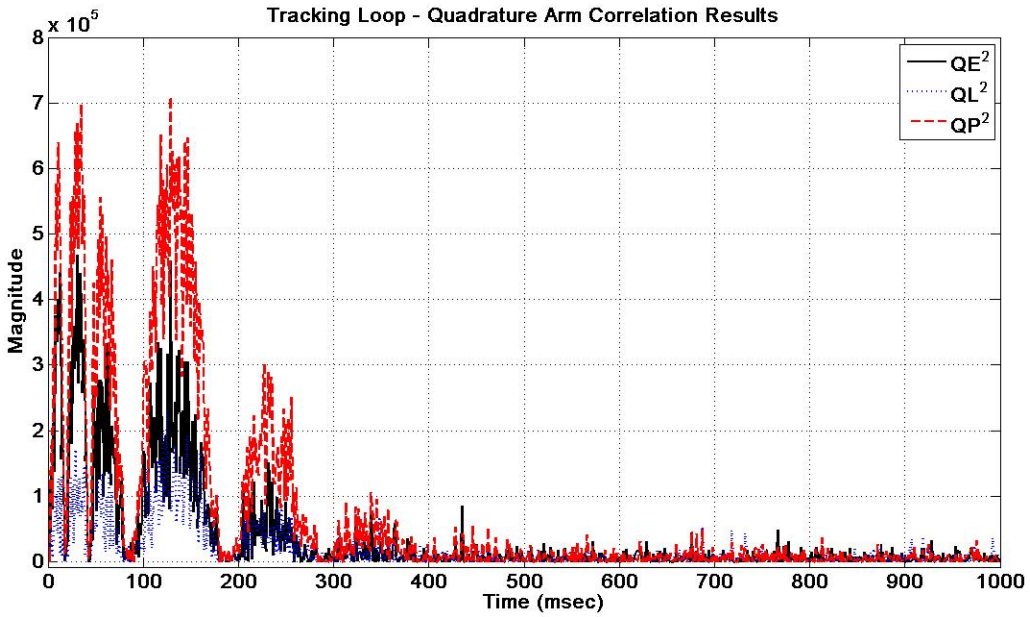

Figure 18 - Tracking Loop – Comparison of Quadrature Phase Arm Correlation Results namely Q-Early, Q-Prompt and Q-Late Correlator Outputs

Figure 19 which shows IQ phasor diagram makes clear the reason of signal energy switch between in-phase and quadrature arms in Figure 17 and Figure 18.

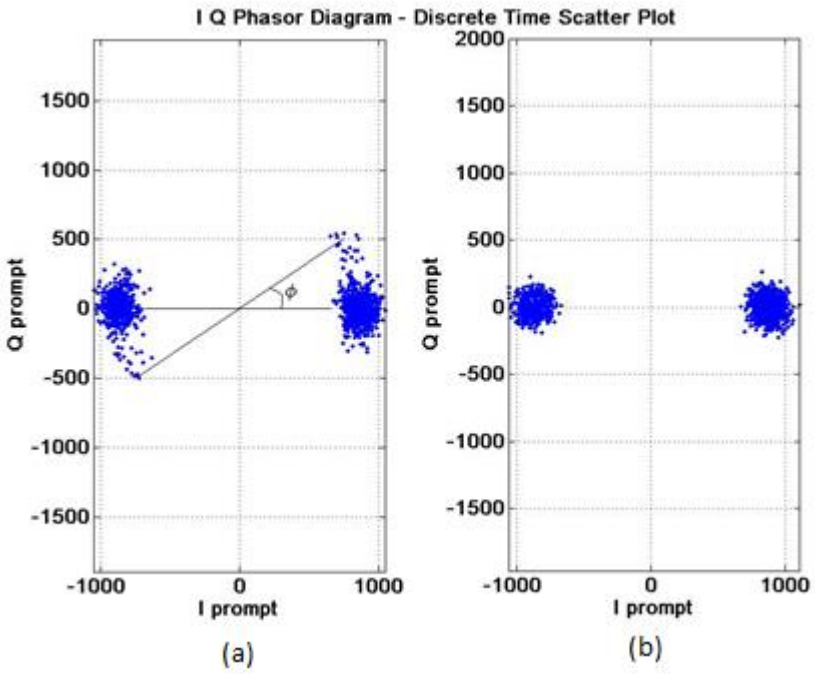

Figure 19 - Tracking Loop – IQ Phasor Diagrams (Discrete Time Scatter Plot): (a) PLL is not in phase lock (b) Locally generated carrier signal is in phase with the input signal

Two IQ phasor diagrams which belong to different time intervals of the scenario are plotted in Figure 19.

The phasor rotates at a directly proportional to the frequency error when the loop is in frequency lock [4]. For the first subplot (a), since PLL is not in phase-lock in the initialization state of the tracking process, there exists a phase error. For the second subplot (b), locally generated carrier signal is in phase with the input signal and there is no phase error.

Figure 20 shows the in-phase prompt (IP) and quadrature arm prompt (QP) correlation results in time. This code tracking results shows the unnormalized +1, -1 bit values in-phase arm. Therefore, for every millisecond bit values of navigation data can be obtained from tracking results output if tracking is achieved.

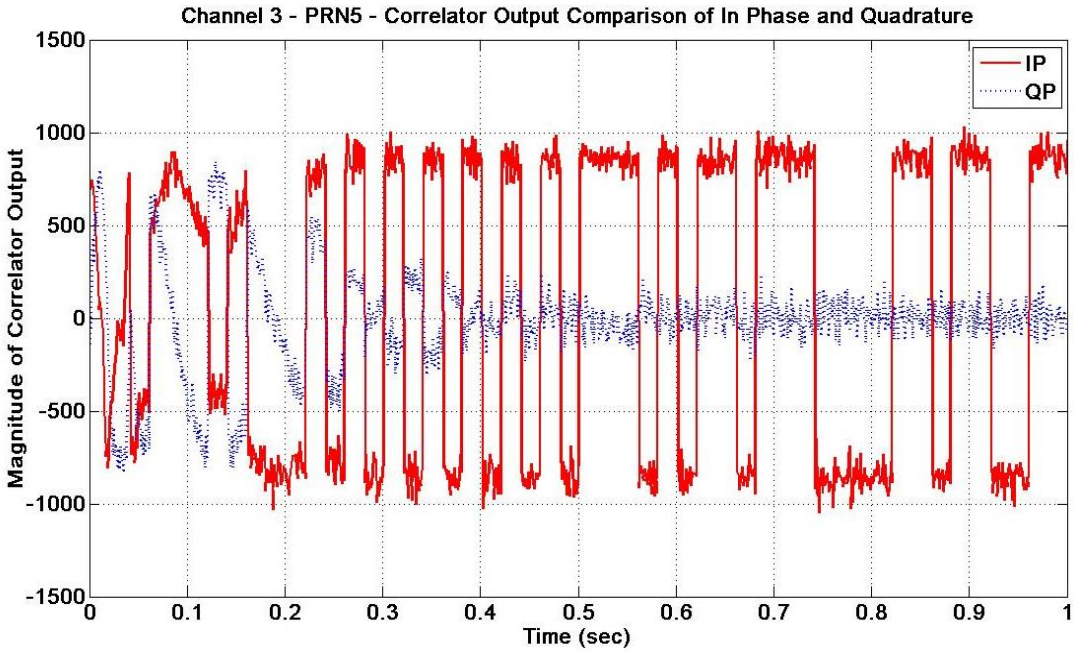

Figure 20 – Tracking Loop – Comparison of In-Phase Prompt (IP) and Quadrature Arm Prompt (QP) Correlator Outputs

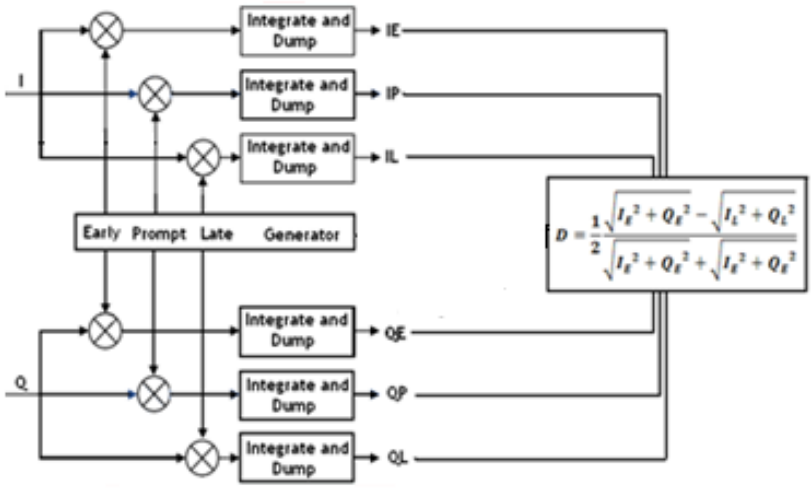

Figure 21 – Tracking Loop – Feeding Correlator Outputs to Discriminator Inputs

As seen in Figure 15 which is general tracking block diagram and in Figure 21 which is some cutted part of Figure 15, after correlator outputs are integrated and dumped, integrator outputs are fed into the discriminator.

Table 6 gives the details of the discriminator used in code tracking loop.

Table 6 - Discriminator for DLL [4]

$D = \frac{1}{2}\frac{\sqrt{{I_E}^2 + {Q_E}^2} - \sqrt{{I_L}^2 + {Q_L}^2}}{\sqrt{{I_E}^2 + {Q_E}^2} + \sqrt{{I_L}^2 + {Q_L}^2}}$	Since implemented this DLL discriminator uses both in-phase and quadrature correlator outputs, it is not directly dependent on PLL performance.

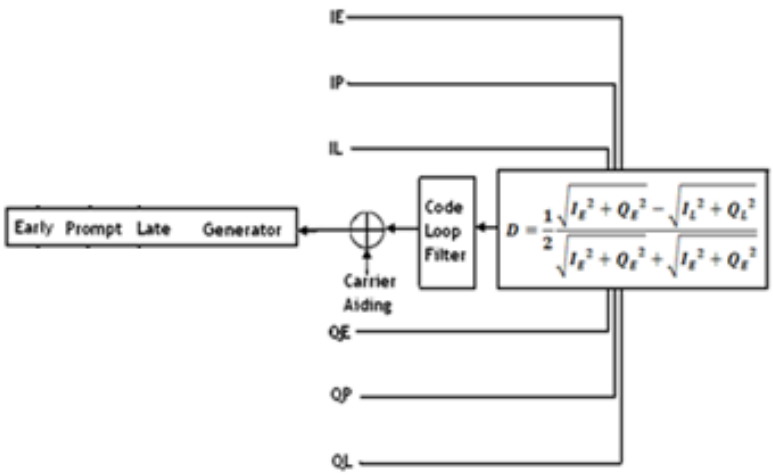

Figure 22 – Code Tracking Loop - DLL Discriminator, Code Loop Filter and Carrier Aiding

Code loop filter in Figure 22 is implemented as a 2nd order PLL with 2 Hz noise bandwidth. However, with carrier aiding to the output of code filter, DLL noise bandwidth is reduced to 1Hz bandwidth, because carrier loop aiding decreases the dynamic effect on code loop filter. *Doppler Effect* is proportional to emitted frequency and, therefore; it will have smaller effect on C/A code (1.023 MHz) than L1 carrier signal (1575.42 MHz). In this aiding, a scaling is applied to reflect this proportional effect.

Figure 23 shows the change in code frequency output provided by DLL code loop filter for C/A Code of PRN-5 satellite signal during code tracking. As expected, C/A code frequency has a frequency of 1.023 MHz and settles in time.

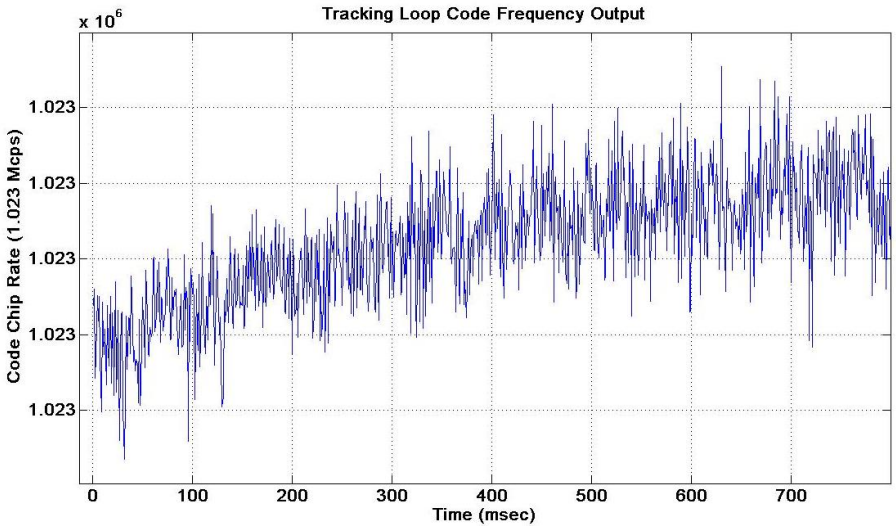

Figure 23 - Tracking Loop – Code Frequency Output for C/A Code of PRN-5 Satellite Signal from DLL Code Filter

2.3.3 Computation of User Position

In this section, computation steps for positioning are discussed after signal tracking is achieved.

As explained in previous section and observed in Figure 20, tracking results provides unnormalized +1, -1 bit values for every millisecond. Tracking results are fed into the software of message decoding block as seen in Figure 24.

Figure 24 shows the block diagram for position generation.

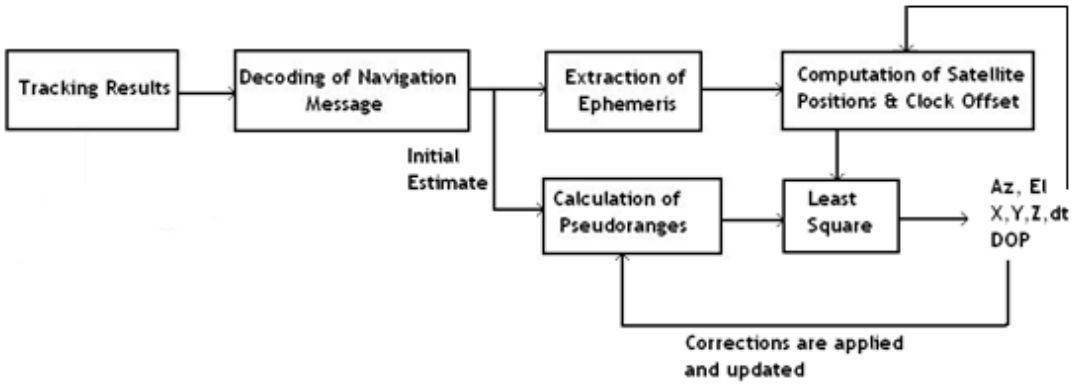

Figure 24 - Block Diagram for Position Computation

Functions of each block in the position computation diagram given in Figure 24 are explained in following sub-sections:

2.3.3.1 Decoding of Navigation Message

In decoding of navigation message block seen in Figure 24, firstly bit synchronization is performed due to the fact that rate of navigation data is 50 bps, but tracking results exist for every millisecond. By grouping every 20 bits, summing them up and normalizing the result, synchronization is performed.

After bit synchronization is performed, navigation data is needed to be extracted.

Figure 25 shows GPS navigation data structure.

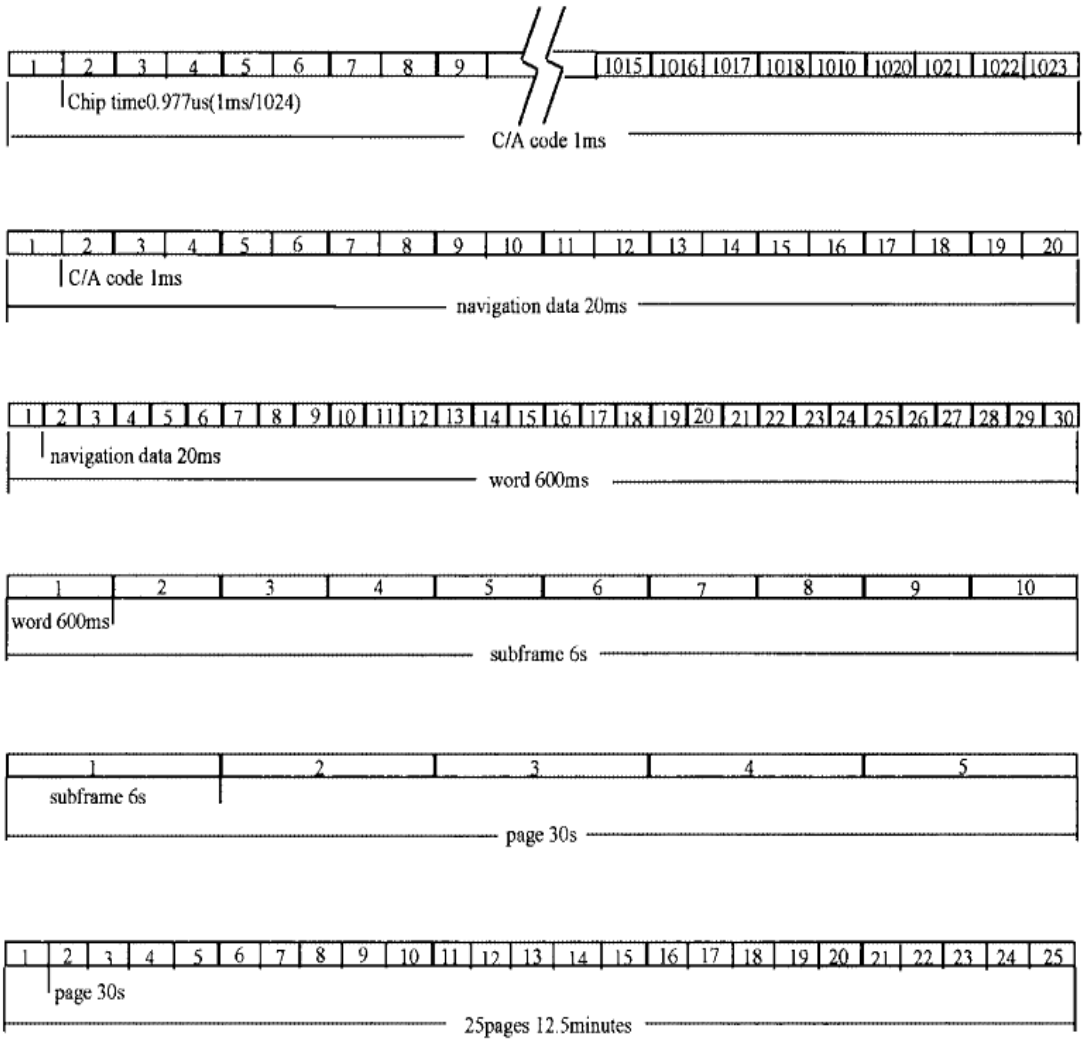

Figure 25 - GPS Data Format [9]

Every subframe in Figure 25 starts with a pre-defined 8-bit-long preamble. Preamble is searched through navigation data considering that this preamble will occur in every 6 sec for all subframes. It provides the starting points of the navigation data. Then considering number of bits, signs, and scale factors of all parameters placed in subframes, navigation data is decoded as given in ICD-GPS-200C document in reference 6.

2.3.3.2 Extraction of Ephemeris

In the extraction of ephemeris block seen in Figure 24, ephemeris parameters which will provide Keplerian orbit elements for each tracked satellite are extracted from decoded navigation data by considering ICD-GPS-200C document given in reference 6. Moreover, extracted ephemeris parameters are detailed in APPENDIX A.

2.3.3.3 Calculation of Pseudoranges

In the calculation of pseudoranges block given in Figure 24, by using decoded navigation message, pseudoranges of the tracked satellites are estimated comparing all channels. While preambles are searched, for each channel a starting point for preamble in data is found if tracking is successfully achieved for that channel.

Figure 26 shows an example for starting points of preambles for different channels.

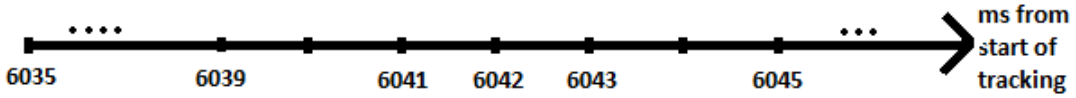

Figure 26 – Relative Pseudorange Estimation - Starting Points for Preamble of Different Channels

The closest satellite to the user will have smallest preamble starting point due to having small travel time. Travel times of the other channels are compared to the smallest one.

With the knowledge that maximum time difference between two satellites is about 19 milliseconds and the travel time from the satellites to Earth is between 65 and 83 milliseconds [4,9], the minimum travel time is set to a pre-decided value (i.e. 70ms).

Then, the other travel times of the satellite signals and so that pseudoranges values are computed initially.

2.3.3.4 Computation of Satellite Positions

Computation of satellite positions using ephemeris parameters are detailed in APPENDIX A. Comparison of computed position of satellite (PRN-5) by implemented MATLAB algorithm and the obtained true values from simulator is given in Table 7 below.

Table 7 – Comparison of GPS PRN5 Satellite Position Computed by Algorithm and Simulator

PRN-5 TOW=219878 sec	**MATLAB Algorithm Solution**	**Simulator Satellite Data**
Position X (m) (WGS84 ECEF)	7088731.35877928	7088730.857
Position Y (m) (WGS84 ECEF)	-14323709.0673348	-14323708.6924
Position Z (m) (WGS84 ECEF)	21152466.9436620	21152467.3186

2.3.3.5 Least Square Approach

Finally, as seen in Figure 24, satellite positions and pseudoranges are used to compute user navigation solution with Least Square Approach. Constructing the linearized observation model as detailed in APPENDIX C, position solution is obtained.

Using computed user position, satellites' positions, elevation and azimuth angles, sky-plot is obtained as in Figure 27. Sky-plot can be thought as a sky over the antenna which is connected to the receiver.

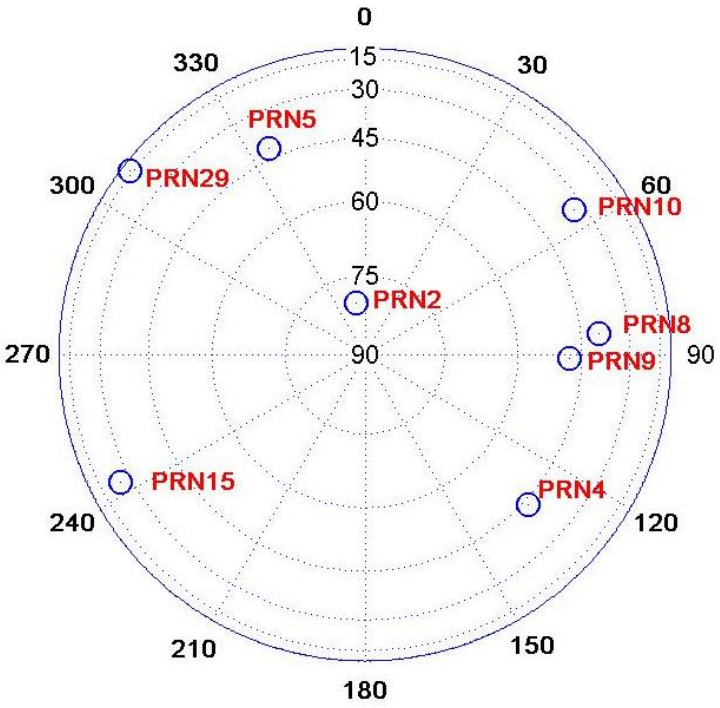

Figure 27 – Sky Plot of GPS Satellites Under Dynamic Scenario

Satellite geometry constructed by tracked satellites is also important besides number of satellites. Since GPS positioning principle relies on the concept of triangularization, coverage of GPS satellites, distances and seperation between them are important. There is a parameter named as Dilution of Precision which measures geometry. Its computation is given in APPENDIX C in detail.

Dilutions of Precision (DOP) values, which give an idea of satellite geometry, are calculated during computation of position from least-square approach. Computed values are presented in Table 8. All DOP parameters can be used to describe the accuracy of various components of the position/time solution [5].

Table 8 – Values of DOP Parameters Under GPS Dynamic Scenario

Geometric Dilution of Precision (GDOP)	2.64795460172953
Position Dilution of Precision (PDOP)	2.33274850089260
Horizontal Dilution of Precision (HDOP)	2.01168528474910
Vertical Dilution of Precision (VDOP)	1.18103254973798
Time Dilution of Precision (TDOP)	1.25297565994075

As shared in Table 8, DOP values do not have units. Since they are all computed from same measurement matrix as detailed in APPENDIX C, they are not independent from each other. DOP values describe the precision of many components of the time and position solution. They represent the effect of the measurement errors onto the solution.

For better coverage with maximized volume by tracked satellites, smaller DOP values are searched. Therefore, DOP values should be as small as possible to have best user position accuracy [9]. Therefore, these values are useful to classify the algorithms' ability for positioning.

2.4 Simulation Results

In sections 2.4.1 and 2.4.2, positioning accuracy performance of implemented GPS algorithms is given with graphs of comparison. The test setup used is shown in Figure 28 below.

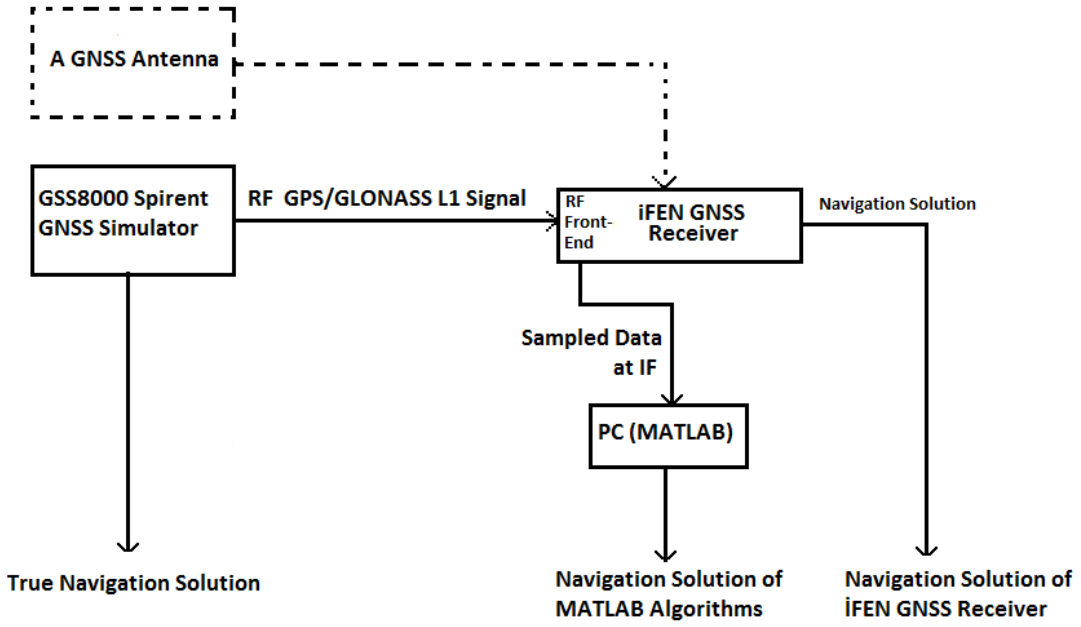

Figure 28 - Laboratory Test Setup

GNSS Simulator provides an environment where signal conditions, error values, types/locations/motions of vehicles, and date/time of the scenario are defined. In simulator, considering motion of defined vehicles which are assumed as GNSS receivers, dynamic and static scenarios can be created. Therefore, it offers a test environment as if receiver navigates according to defined parameters of the scenario. Receiver performances can be seen under different conditions by changing scenario conditions. Moreover, it provides plotting and logging data to compare with the receiver's navigation solution.

As seen in Figure 28 laboratory setup, created RF L1 GPS/GLONASS signals from simulator output are fed into a commercial GNSS receiver which will provide sampled data at IF. In this setup, IFEN GNSS receiver front-end provides sampled IF data in a file. Then, the data can be read in MATLAB and algorithms can be tested. Also, IFEN GNSS receiver computes its navigation solution using same IF data.

In some tests, especially related to the tests of the algorithms implemented for position computation of GPS/GLONASS satellites, RF signal is also fed from an antenna to the receiver as seen in Figure 28 with dashed lines. The solutions can be compared with other commercial receivers which logs data synchronously in time.

With this setup, we have three separate navigation solutions, namely true solution obtained from simulator, implemented MATLAB algorithms' solution and IFEN GNSS receiver solution. Performances of implemented GPS algorithms at MATLAB are compared with the other two systems mentioned.

2.4.1 Static Scenario

GNSS Simulator gives a chance of creating scenario files using its GUI. GNSS simulator models GPS/GLONASS signals, motion of vehicles, the satellites, and other effects according to defined scenario files. Then, an environment where the vehicle or the receiver navigates according to pre-defined scenario parameters is created.

Using Spirent GSS8000 simulator, a static scenario is created. Actual coordinate values of the chosen position over "*Red Sea"* are shown in Table 9 below:

Table 9 - GNSS Simulator Defined Position Coordinate Values (Static Scenario)

In ECEF	In LLH
X : 4692885.16m	Latitude: 20˚38'(N)
Y : 3692936.47 m	Longitude: 38˚12'(E)
Z : 2233520.16 m	Height: 200 m

The navigation solutions of the implemented algorithms at MATLAB are compared with the true position solution obtained from the simulator and the position solution of IFEN GNSS receiver which IF data is acquired as seen in Figure 28, laboratory test setup.

In Figure 29, obtained navigation solutions in ECEF X-axis from simulator (true value), IFEN GNSS receiver and implemented MATLAB algorithm are plotted on the same graph.

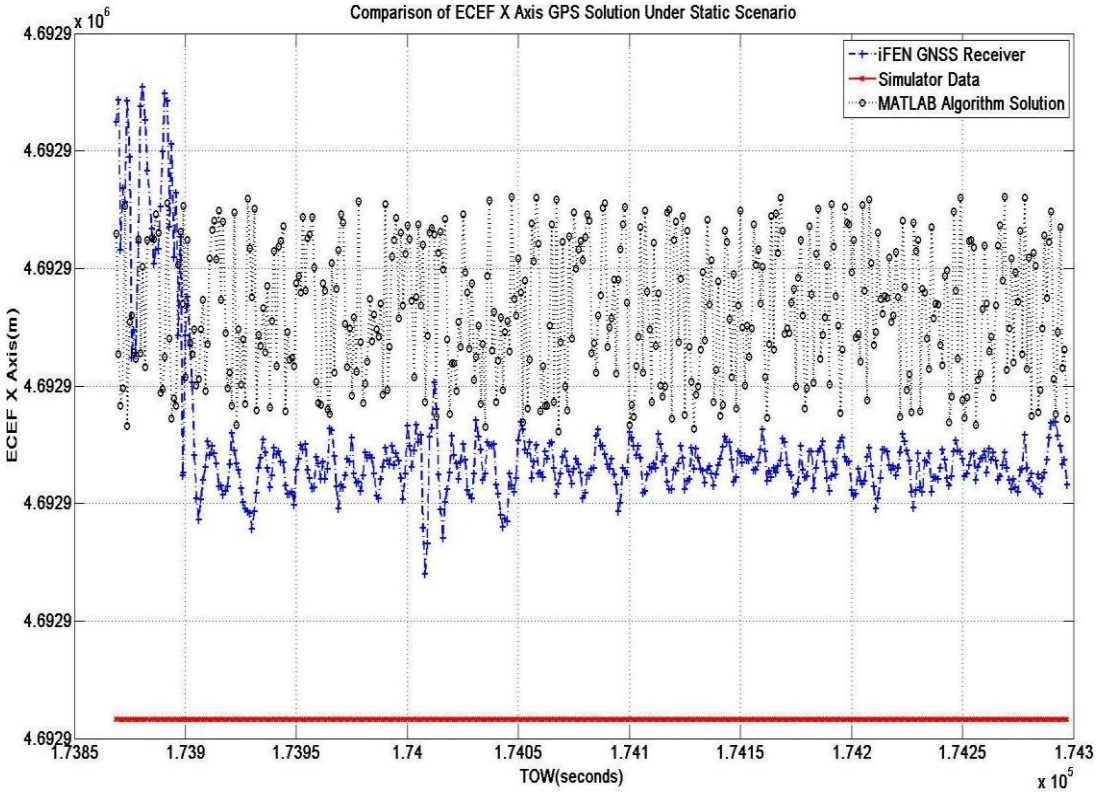

Figure 29 - Comparison of GPS Solutions in ECEF X Axis (Static Scenario)

In Figure 30, the error values of the navigation solutions in ECEF X-axis of IFEN GNSS receiver and implemented MATLAB algorithms are plotted on the same graph. They are calculated by subtracting computed values from actual true value provided by simulator. RMS position errors in ECEF X-axis of IFEN GNSS receiver and MATLAB algorithms are 2.3892 meters and 3.5262 meters, respectively.

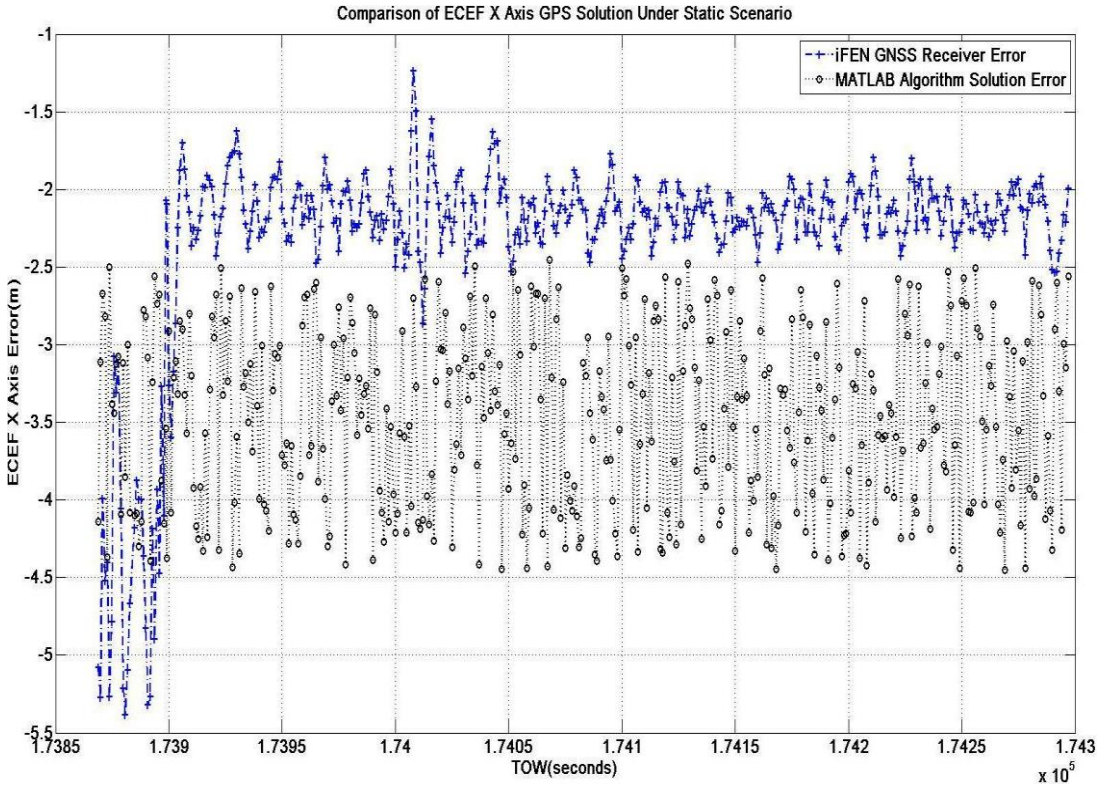

Figure 30 - Comparison of GPS Solutions in ECEF X Axis - Error Values (Static Scenario)

In Figure 31, navigation solutions of IFEN GNSS receiver, implemented MATLAB algorithms and the true value obtained from the simulator in ECEF Y-axis are plotted on the same graph.

Figure 32 shows the error values of the navigation solutions of IFEN GNSS receiver and implemented MATLAB algorithms in ECEF Y-axis. Each represents the difference which is computed by subtracting it from the true value provided by simulator. RMS position errors in ECEF Y-axis of IFEN GNSS receiver and MATLAB algorithms are 1.7365 meters and 2.5815 meters, respectively.

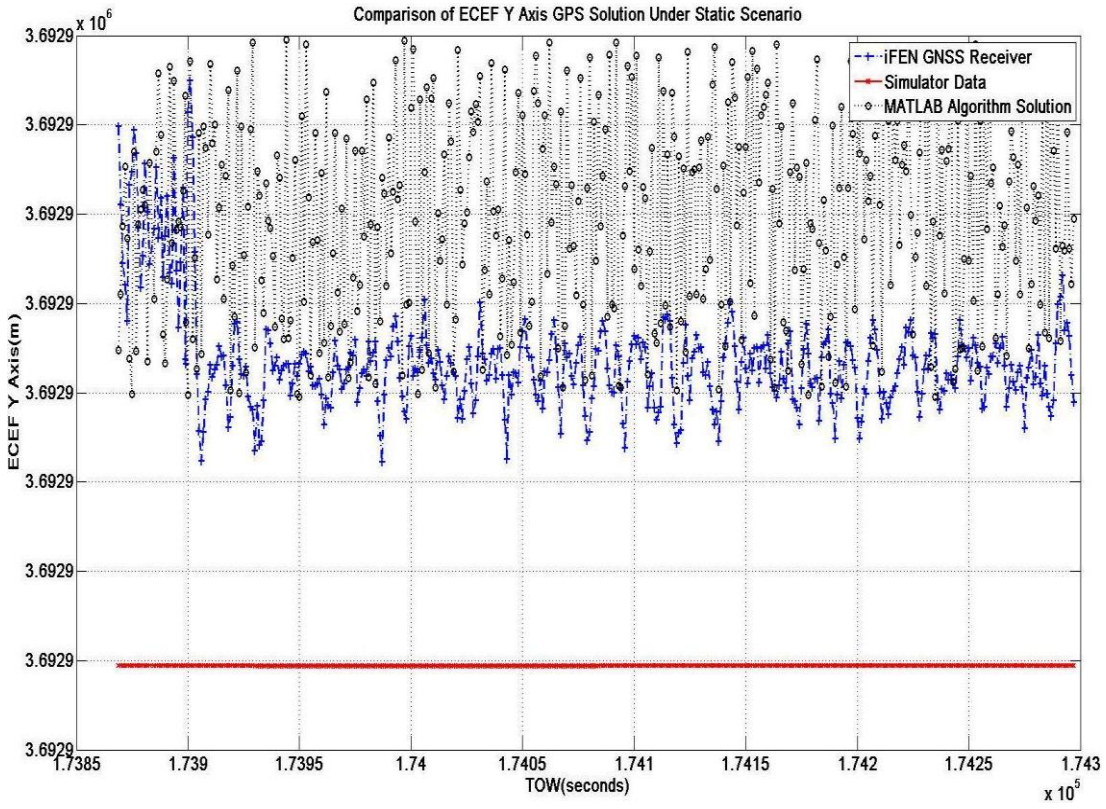

Figure 31 - Comparison of GPS Solutions in ECEF Y Axis (Static Scenario)

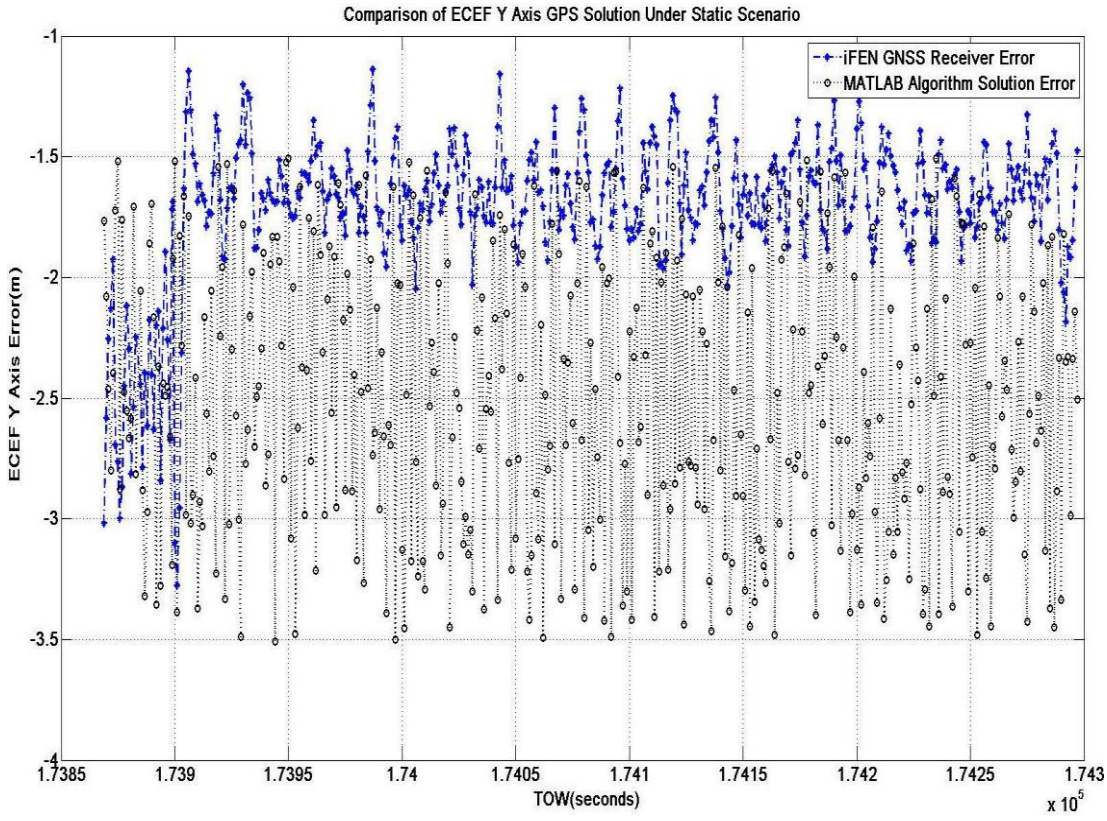

Figure 32 - Comparison of GPS Solutions in ECEF Y Axis - Error Values (Static Scenario)

In Figure 33, navigation solutions of IFEN GNSS receiver, implemented MATLAB algorithms and the true value obtained from the simulator in ECEF Z-axis are plotted on the same graph.

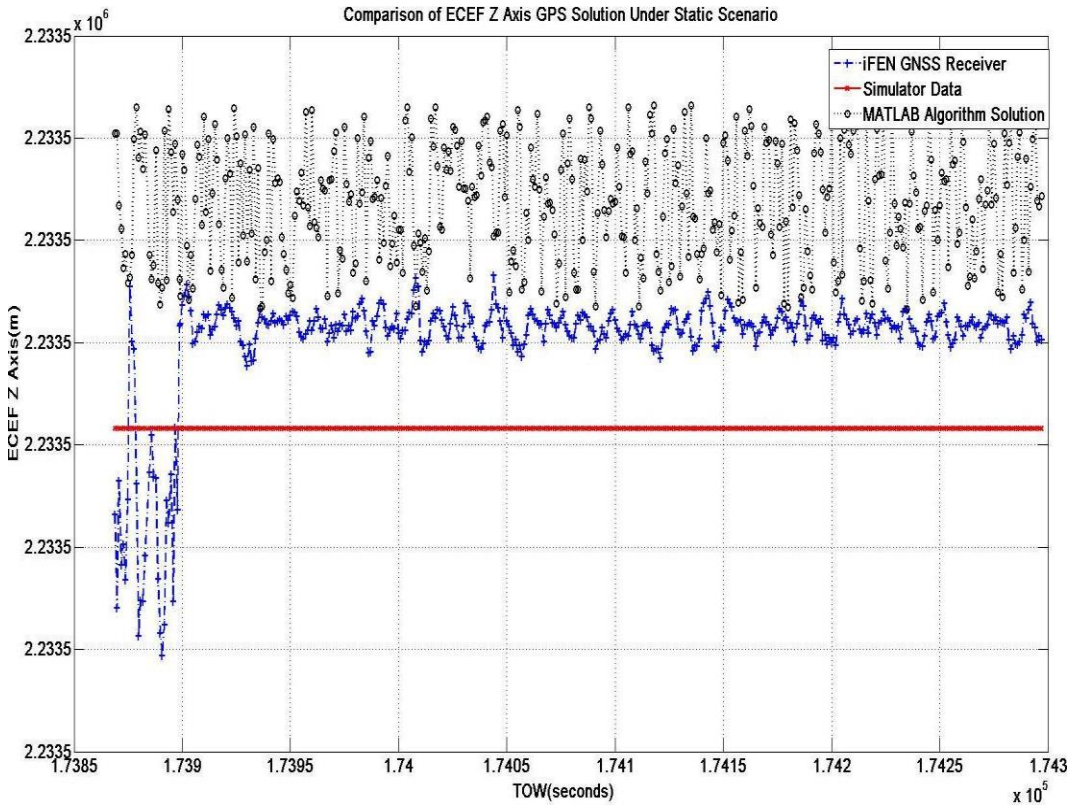

Figure 33 - Comparison of GPS Solutions in ECEF Z Axis (Static Scenario)

In Figure 34, error values of the navigation solutions of IFEN GNSS receiver and implemented MATLAB algorithms in ECEF Z-axis are plotted on the same graph. They are obtained by subtracting them from the actual value which the simulator provides. RMS position errors in ECEF Z-axis of IFEN GNSS receiver and MATLAB algorithms are 1.0341 meters and 2.2762 meters, respectively.

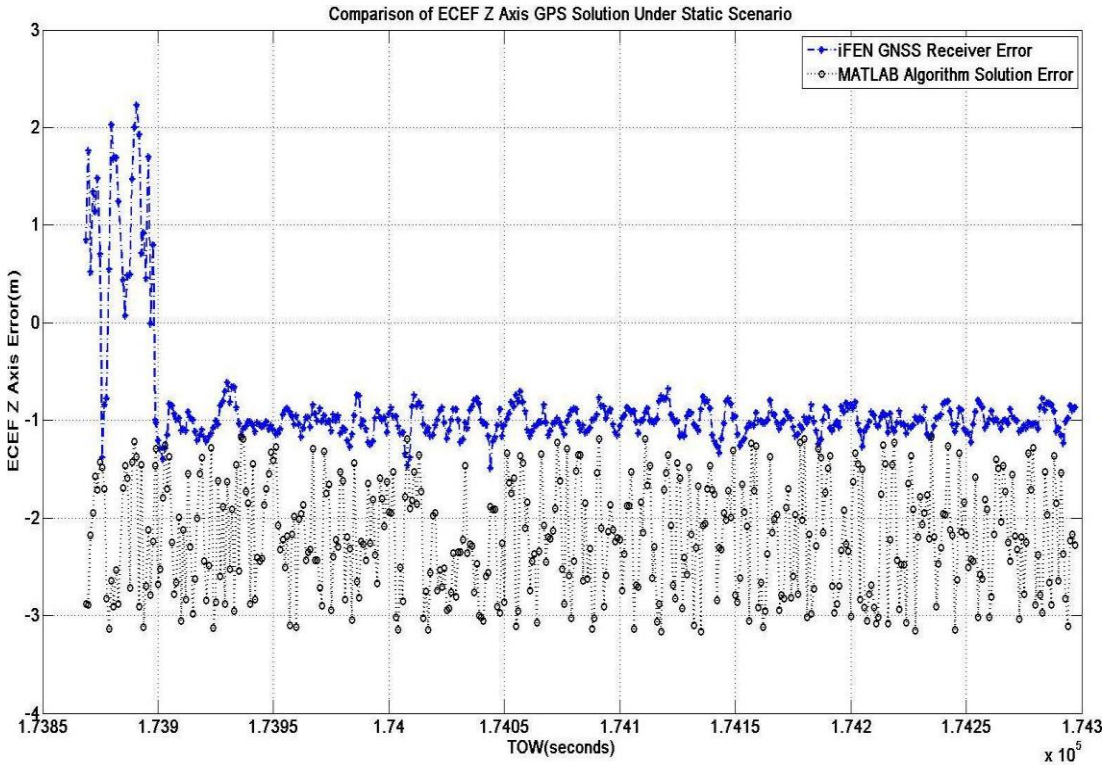

Figure 34 - Comparison of GPS Solutions in ECEF Z Axis - Error Values (Static Scenario)

To sum up, RMS position solution errors in each axis are shown on Table 10 below.

Table 10 – Root Mean Square Position Errors of GPS-only Algorithms

	X Error (m) (ECEF)	Y Error (m) (ECEF)	Z Error (m) (ECEF)
MATLAB Algorithm	3.5262	2.5815	2.2762
IFEN GNSS Receiver	2.3892	1.7365	1.0341

As seen in Table 10, it is inferred that the implemented MATLAB algorithms for GPS-only positioning work well in navigation solution computation when compared with a commercial GNSS receiver.

2.4.2 Dynamic Scenario

As mentioned before, considering motion of defined vehicles which are assumed as GNSS receivers, dynamic and static scenarios can be created in the simulator. In previous sub-section 2.4.1, a static point is defined for receiver without choosing any vehicle in the simulator and performance of algorithms and IFEN GNSS receiver for a stationary scenario is observed.

In this section, in order to analyze the positioning and signal tracking performance of the algorithms under dynamic conditions, a dynamic scenario is created. After an aircraft is defined as vehicle, motion of the aircraft, in other words motion of the receiver is defined.

An aircraft flight scenario is created using aircraft motion command file GUI of the simulator and the details of the scenario are given in Figure 35.

Aircraft flight trajectory is plotted in Figure 36, using gathered motion data from the simulator.

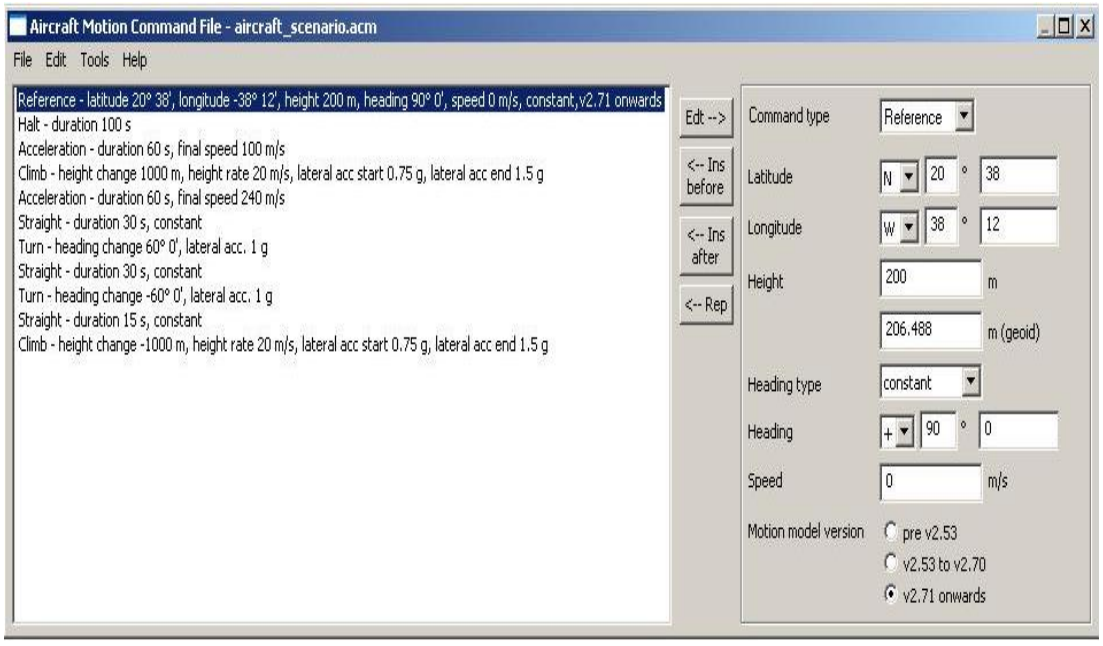

Figure 35 - Simulator Aircraft Motion File (Dynamic Scenario)

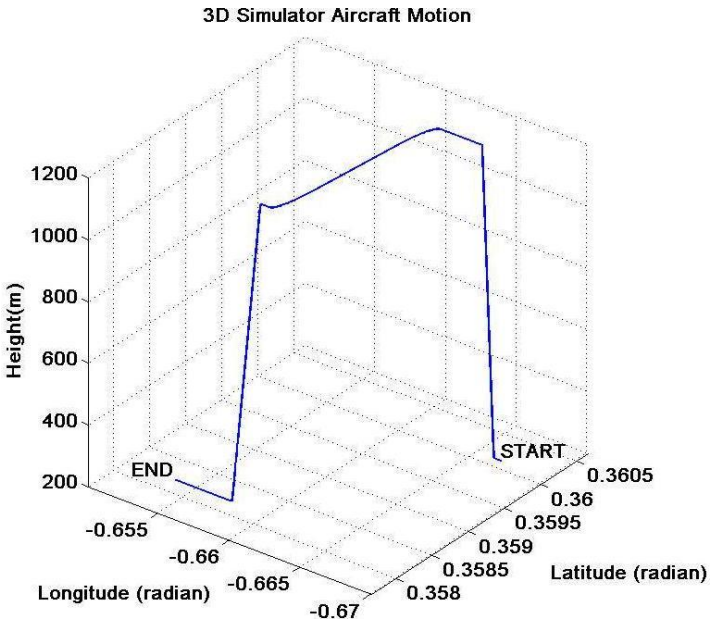

Figure 36 - 3D Motion Graph of Simulator Aircraft Motion File (Dynamic Scenario)

Figure 36 shows the plot of the defined trajectory of the aircraft in geodetic coordinate system (LLH). To compare navigation solutions of IFEN GNSS receiver and MATLAB algorithms with true solution of the simulator, LLH to ECEF conversion is implemented.

The navigation solutions of the algorithms under dynamic scenario are plotted in each ECEF axis in Figure 37 below, including the true position obtained from simulator and the commercial IFEN GNSS receiver's navigation solution on the same graph.

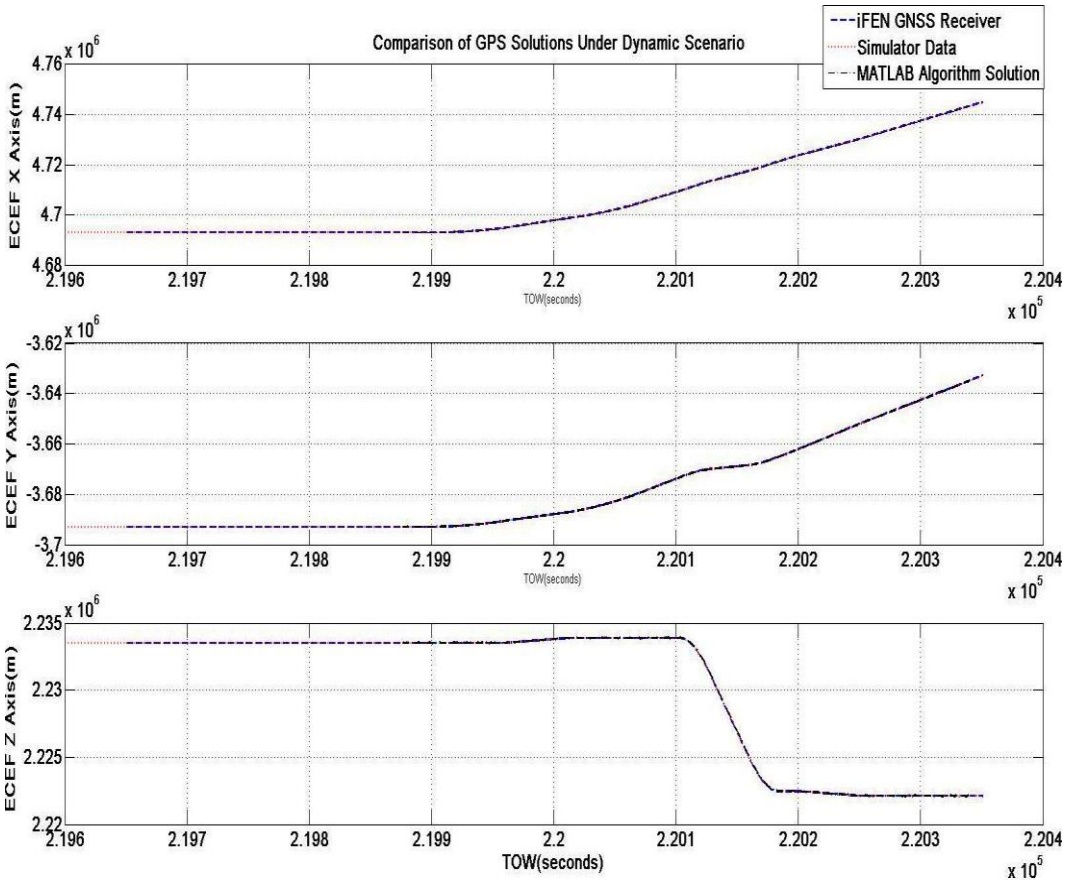

Figure 37 - Comparison of GPS Solutions (Dynamic Aircraft Scenario)

In implemented tests it is observed that addition of tracking lock status indicator-check in signal tracking loops is improved accuracy under dynamic scenario compared with the obtained accuracies when tracking lock status indicator-check is not applied. With indicator check, when the tracking is lost, the position solution is not computed which prevents false computation of navigation solution. When loss and recovery points of the navigation solution are not taken into account, algorithm results are nearly same with the static scenario.

CHAPTER 3

GLONASS POSITIONING ALGORITHMS

This chapter starts with the GLONASS L1 band signal structure and characteristics. In second section of this chapter, GLONASS navigation signal processing algorithms to compute navigation solution are provided. In the subsections of section 3.2, GLONASS signal acquisition and tracking algorithms with graphs of results are presented in detail. Some similar parts with GPS are referred since they are presented in section 2. Procedure of user position computation is provided in the last subsection of section 3.2. In the last section of this chapter, simulation results of the implemented GLONASS navigation signal processing algorithms under static and dynamic scenario are presented and the positioning performances are evaluated.

3.1 Signal Characteristics

As stated in second chapter, all GPS satellites' signals use the same center frequency for L1 signal. Therefore, GPS uses different C/A code sequences for each satellite. Main characteristics of this CDMA structure were given in the second chapter. However, GLONASS system uses FDMA and a same C/A code sequences for all satellites. GLONASS L1 Signal Characteristics are presented in Table 11 and FDMA principle is summarized in Table 12.

Table 11 - GLONASS SPS L1 Signal Characteristics [7]

Detail	**GLONASS**
Fundamental clock frequency	5.0 MHz
Signal separation technique	FDMA
L1 Carrier frequencies (MHz)	1598.0625-1605.375
C/A Code Clock Rate (MHz)	0.511
C/A Code Length (chips)	511

Table 12 - Main Characteristics of Frequency Division Multiple Access [16]

Frequency Division Multiple Access **(FDMA)**	Advantages	Disadvantages
Transmit on unique frequencies –	• Simple transmitters/ receivers • Simple frequency control (tuned XO)	• Reduced bandwidth per user • Wasted bandwidth when not in use by some users • No security

Figure 38 below shows GPS and GLONASS L1 frequencies.

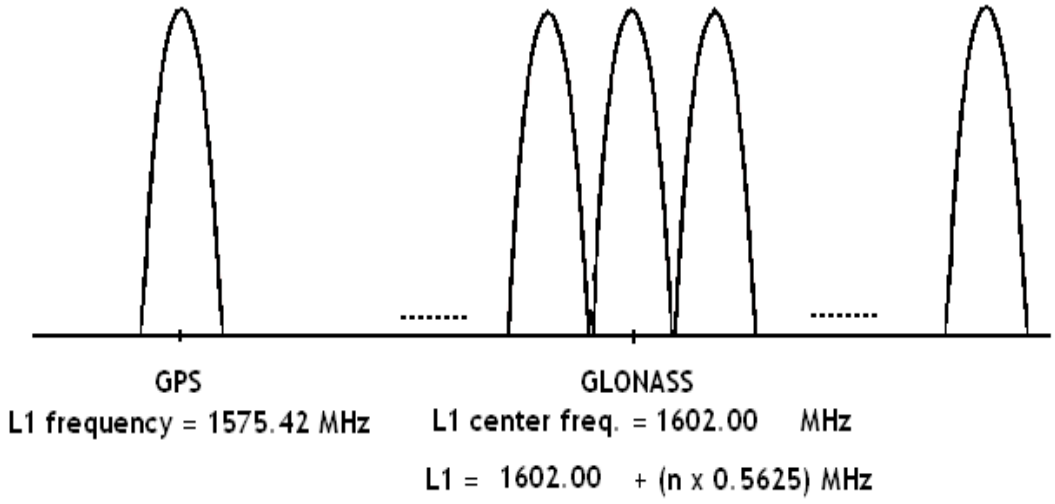

Figure 38 – Placement of GPS and GLONASS L1 Frequencies

Center frequency of GLONASS L1 (1602 MHz) and allocated frequencies for GLONASS satellites are shown in the figure above. Each GLONASS satellites transmit its signals on a different frequency channel and it is modelled in the following equation 3.1. As indicated, there is 0.5625 MHz spacing between L1 band frequencies of GLONASS satellites.

$$f_{L1,n} = f_{L1} + \Delta f_{L1} \cdot n = 1602.0000 + 0.5625 \cdot n \text{ MHz} \quad (3.1)$$

$f_{L1} : GLONASS\ L1\ band -\ center\ frequency$

$n : Frequency\ channel\ number$

$n = -7, -6, -5, \ldots, 0, \ldots\ldots, 6$

Since end of 2009, there have been 24 satellites in GLONASS constellation. Although, it seems that the channels are not enough for accommodation of all satellites, the channel limitation indicated above is not a problem; because, two satellites are assigned on the same FDMA channel number. And also it is guaranteed that any user on Earth will never reach those both satellites' signals which have same FDMA channel number at the same time due to the fact that those satellites are placed in antipodal position.

Figure 39 below shows the GLONASS L1 signal structure. GLONASS L1 signal is a BPSK modulated signal same as GPS. However, C/A code chipping rates of GLONASS ($0.511\ Mchips/s$) is different than the C/A code in GPS ($1.023\ Mcps$). GLONASS C/A code has a length of $511\ chips$ with a rate of $0.511\ Mchips/s$. It repeats itself every 1 millisecond.

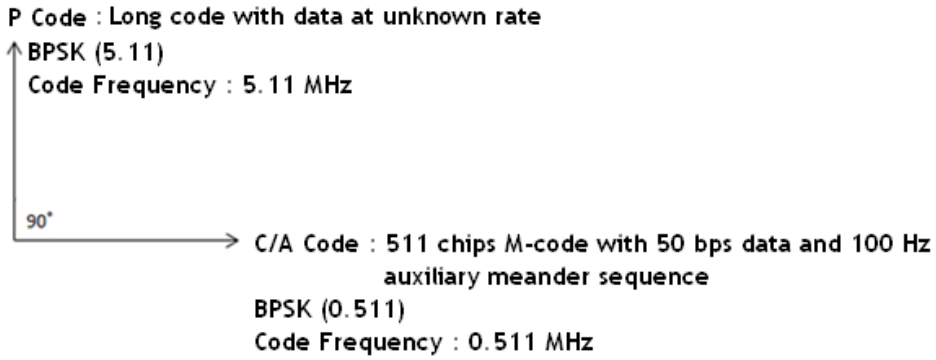

Figure 39 - GLONASS L1 Signal Structure

GLONASS has the same signal structure with GPS. In GLONASS case, still HP (High Precision) P code is not accessible by civil users. Hence, algorithms in this thesis are focused on C/A (Coarse/Acquisition) code as expected.

Although, navigation data rates for both GPS and GLONASS satellite signals are same ($50\ bps$), received GLONASS message is the modulo-2 addition of meander sequence (100 Hz) and navigation data (20ms) [24]. The received navigation message data rate is 100 bps since meander code is phase locked to the navigation data [20]. This situation does not affect the acquisition and tracking process, only in navigation data demodulation part it should be taken into account.

3.2 Navigation Signal Processing

Implemented GLONASS navigation signal processing algorithm consists of 8 channels in parallel, and each acquired satellite initializes a new channel for tracking and measurement constructing to compute navigation solution. The block diagram in Figure 4 is also valid for GLONASS navigation signal processing and related modifications to GPS algorithms, outputs and results are shared in the following subsections.

3.2.1 Signal Acquisition

After A/D conversion and filtering are applied to the signal at RF front-end, the sampled GLONASS signal from satellite "k" is modeled simply as following:

$$s^k(n) = A^k C(nT_s - \tau) M^k(nT_s - \tau) D^k(nT_s - \tau) \times \cos\left(\omega_{IF} nT_s + \omega_{Dopp} nT_s + \phi(nT_s - \tau_{L1})\right) + w(n) \quad (3.2)$$

$A^k : Amplitude$

$C : Periodic\ C/A\ code$

$M^k : Meander\ code$

$D^k : Navigation\ data$

$\omega_{IF} : IF\ (Intermediate\ Frequency)$

$\omega_{Dopp} : Doppler\ frequency$

$\phi : Carrier\ phase\ perturbation$

$\tau : Code\ propagation\ delay$

$\tau_{L1} : Carrier\ propagation\ delay$

$w : Noise$

GLONASS acquisition method is same as the method used in GPS acquisition shown in Figure 7, in principle. However, two differences between GPS and GLONASS should be taken into account while configuring parallel code phase search algorithm, namely, GLONASS C/A code and FDMA structure:

- Only one ranging code is needed to be created for all GLONASS satellites' signals.
- Since all satellites have different center carrier frequencies, they will have different IF values when the signals are down-converted. Then, Doppler-search frequency values should be created around the related satellite's down-converted IF value.

Figure 40 shows the magnitude of cross-correlation result for the satellite signal of FDMA channel-1 with IF value and the beginning point of C/A as detailed in section 2.3.1.

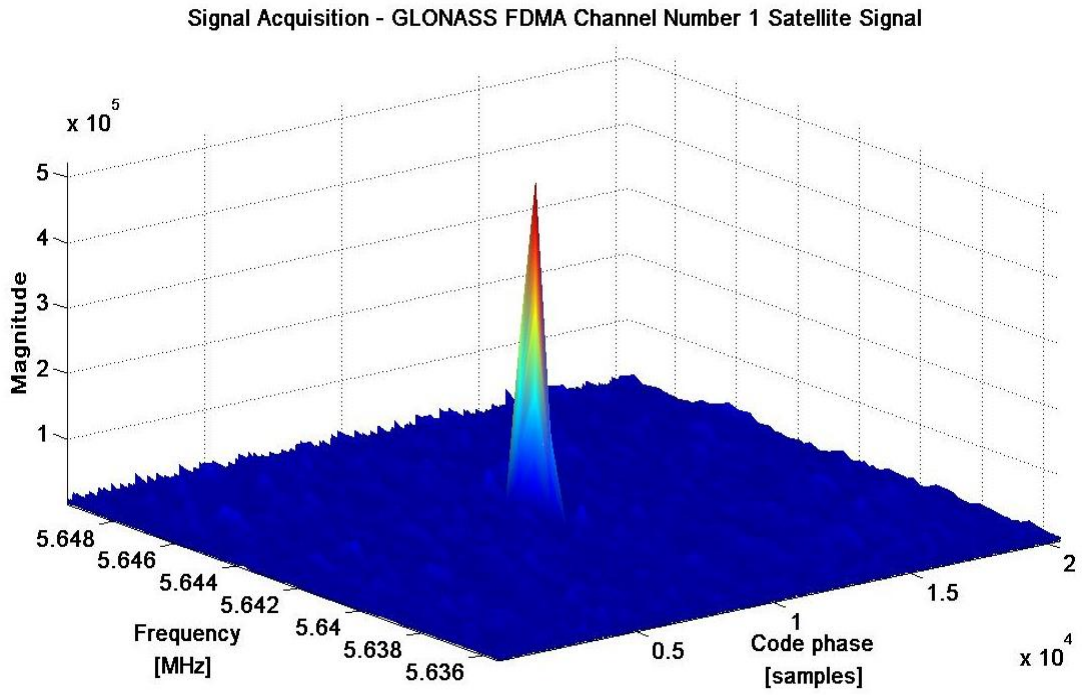

Figure 40 – GLONASS Signal Acquisition – FDMA Channel 1 Acquisition Result

Comparison of cross-correlation results of all acquired GLONASS satellites and pre-defined threshold value is plotted in Figure 41.

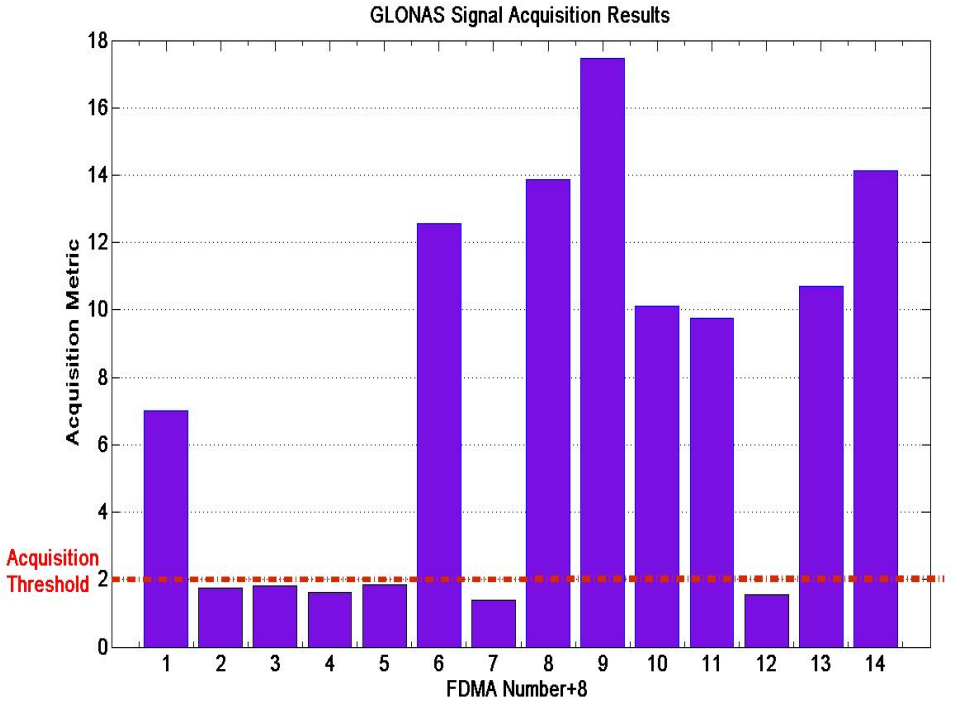

Figure 41 – GLONASS Signal Acquisition Results

As shared in Figure 40, the carrier frequency and code phase values of the GLONASS satellite (FDMA Channel 1) are obtained in acquisition process. Through acquisition process, IF carrier frequencies and code phase values of all acquired GLONASS satellites are given in Table 13 below.

Table 13 – GLONASS Acquired Satellites with IF Carrier Frequency and Code Phase Values

Channel Number	**FDMA Number**	**Carrier IF Frequency (Hz)**	**Code Phase (Samples)**
8	-7	1143498.47412109	12812
4	-2	3952998.47412109	12931
3	0	5083498.47412109	14535
1	1	5642998.47412109	9772
6	2	6207498.47412109	14198
7	3	6766498.47412109	14900
5	5	7890998.47412109	814
2	6	8453998.47412109	14276

However, characteristics of used sampled IF data which is obtained from IFEN GNSS receiver are as follows:

- IF frequency : 5079998.47412109 Hz
- Sampling frequency: 20480000 Hz
- IF sample format : 2 - bit samples

As seen in Table 13, satellite signals have different IF carrier frequencies as expected due to FDMA nature.

3.2.2 Signal Tracking

In this section, carrier and code tracking loops of GLONASS positioning algorithms are described and results of tracking loops are presented.

Although some modifications to the GPS tracking loops are needed to perform GLONASS signal tracking due to nature of FDMA, tracking loop architecture is same with GPS case in principle.

3.2.2.1 Carrier Tracking

Same carrier tracking algorithm in principle with GPS is implemented as seen in Figure 10. In signal tracking, obtained coarse carrier frequency and code phase values of visible satellites in acquisition process are the initial inputs to the tracking algorithm.

Same types of discriminators with GPS are applied to the signal after integration and low pass filtering. Discriminators are shared in Table 4. Discriminator outputs are fed into the frequency and phase error inputs of FLL/PLL loop filter in Figure 42.

In GLONASS carrier signal tracking, 2^{nd} order PLL assisted with 1^{st} order FLL filter is implemented. The loop filter order and bandwidth values are chosen experimentally. Determined noise bandwidths of PLL and FLL are 15 Hz and150 Hz respectively.

Figure 42 shows the digital version of the 2^{nd} order PLL assisted with 1^{st} order FLL filter. It is converted from analog form using bilinear transformation.

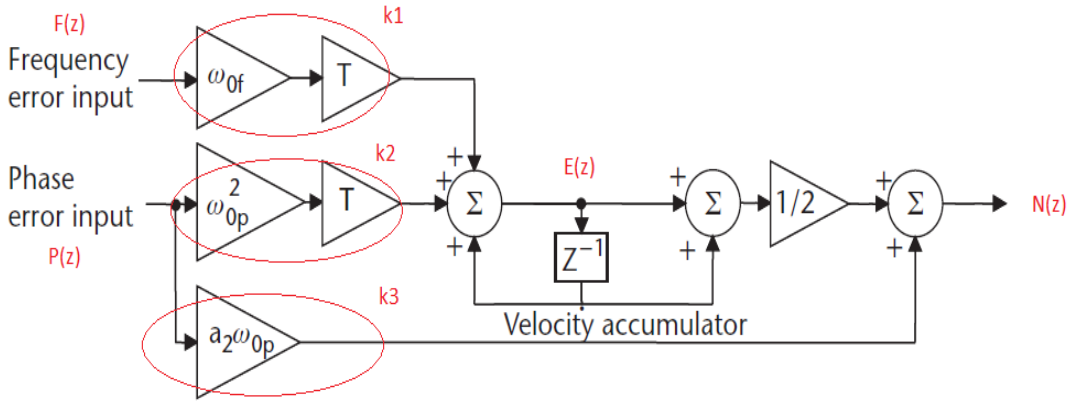

Figure 42 - 2nd Order PLL Filter Assisted with 1st Order FLL Filter [Adapted from Reference 5]

Loop filter values of PLL and FLL are computed using equation sets 3.3:

$$B_{n\_PLL} = \frac{\omega_{0p}(1+a_2^2)}{4 \cdot a_2} = 0.5286 \cdot \omega_0 \qquad B_{n_{PLL}}: PLL\ Noise\ BW\ (Hz) \qquad 3.3$$

$$\omega_{0p} = \frac{8\zeta B_{n_{PLL}}}{4\zeta^2 + 1} = B_{n_{PLL}} \cdot 1.8919 \qquad \omega_{0p}: PLL\ natural\ frequency$$

$$\zeta: Damping\ ratio$$

$$k_3 = a_2 \cdot \omega_{0p} = 1.414 \cdot \omega_{0p}$$

$$k_2 = \omega_{0p}^2 \cdot T = \left(\frac{\omega_{0p}(1+a_2^2)}{4 \cdot a_2}\right) \qquad T: Time\ difference\ between\ each\ sample$$

$$k_1 = \omega_{0f} \cdot T = 4 \cdot B_{n\_FLL} \cdot T \qquad B_{n_{PLL}}: PLL\ Noise\ BW\ (Hz)$$

$$\omega_{0f}: FLL\ natural\ frequency$$

With the determination of the BW values of the loop filters at first, natural frequencies and the coefficients of 2nd order PLL (ω_{0p}^2,$a_2\omega_{0p}$) and 1st order FLL filters (ω_{0f}) are computed step by step using equation sets in 3.3 above.

After loop coefficients are computed, considering the loop filter in Figure 42, we can write:

$$F(z)\cdot k_1 + P(z)\cdot k_2 + E(z)\cdot z^{-1} = E(z) \tag{3.4}$$

$$\xrightarrow{yields} E(z) = \frac{F(z)\cdot k_1 + P(z)\cdot k_2}{1 - z^{-1}} \tag{3.5}$$

$$\frac{1}{2}E(z)(1 + z^{-1}) + P(z)\cdot k_3 = \widetilde{N}(z) \tag{3.6}$$

$$\xrightarrow{yields} \frac{1}{2}F(z)\cdot k_1 \cdot \frac{1 + z^{-1}}{1 - z^{-1}} + \frac{1}{2}P(z)\cdot k_2 \cdot \frac{1 + z^{-1}}{1 - z^{-1}} + P(z)\cdot k_3 = \widetilde{N}(z) \tag{3.7}$$

$$\xrightarrow{yields} \frac{1}{2}F(z)\cdot k_1 \cdot (1 + z^{-1}) + \frac{1}{2}P(z)\cdot k_2 \cdot (1 + z^{-1}) + P(z)\cdot k_3 \cdot (1 - z^{-1}) = \widetilde{N}(z)\cdot(1 - z^{-1}) \tag{3.8}$$

In sampled time domain, equation 3.8 can be expressed as following:

$$\frac{1}{2}k_1 \cdot [f(n) + f(n-1)] + \left[\frac{1}{2}k_2 + k_3\right]\cdot p(n) + \left[\frac{1}{2}k_2 - k_3\right]\cdot p(n) + \tilde{n}(n-1) = \tilde{n}(n) \tag{3.9}$$

As seen in Figure 42, discriminator outputs, namely frequency and phase errors are fed into the loop filter. As mentioned in section 2, PLL discriminator computes the difference between the phases of incoming signal and locally generated signal and FLL discriminator tracks the change in carrier phases. In Figure 43, computed phase error from PLL discriminator in subplot (a) and the FLL assisted PLL loop filter output in subplot (b) are given.

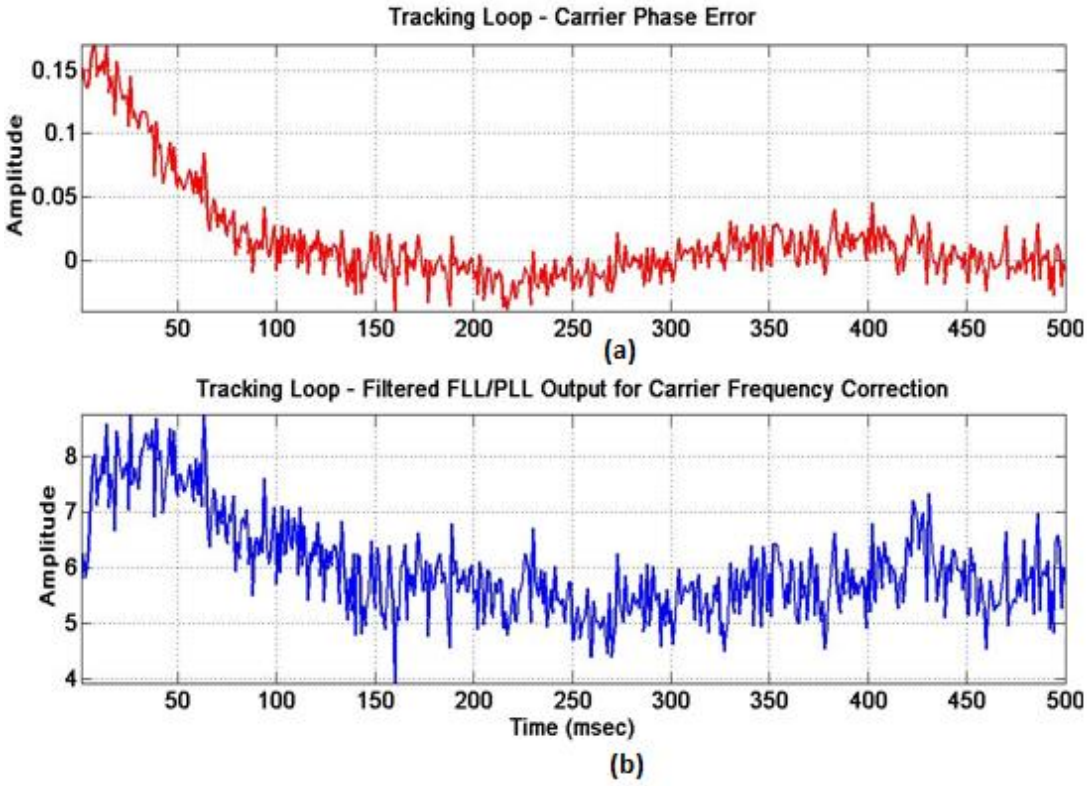

Figure 43 – GLONASS Tracking Loop Outputs in Time for Channel-1 (FDMA Channel Number-1) Satellite Signal: (a) Computed Carrier Phase Error from PLL Discriminator Output (b) Carrier Frequency Correction for Channel-1 (FDMA Channel Number-1) Satellite Signal Obtained from Filtered FLL/PLL Output in Time

Figure 44 shows the change in carrier frequency of FDMA channel number-1 satellite signal in receiver channel 1 during signal tracking. The initial coarse value is taken from acquisition output and then settling of carrier frequency occurred over time. FLL assisted PPL tracking loop output provides the correction on carrier frequency over time by tracking the signal.

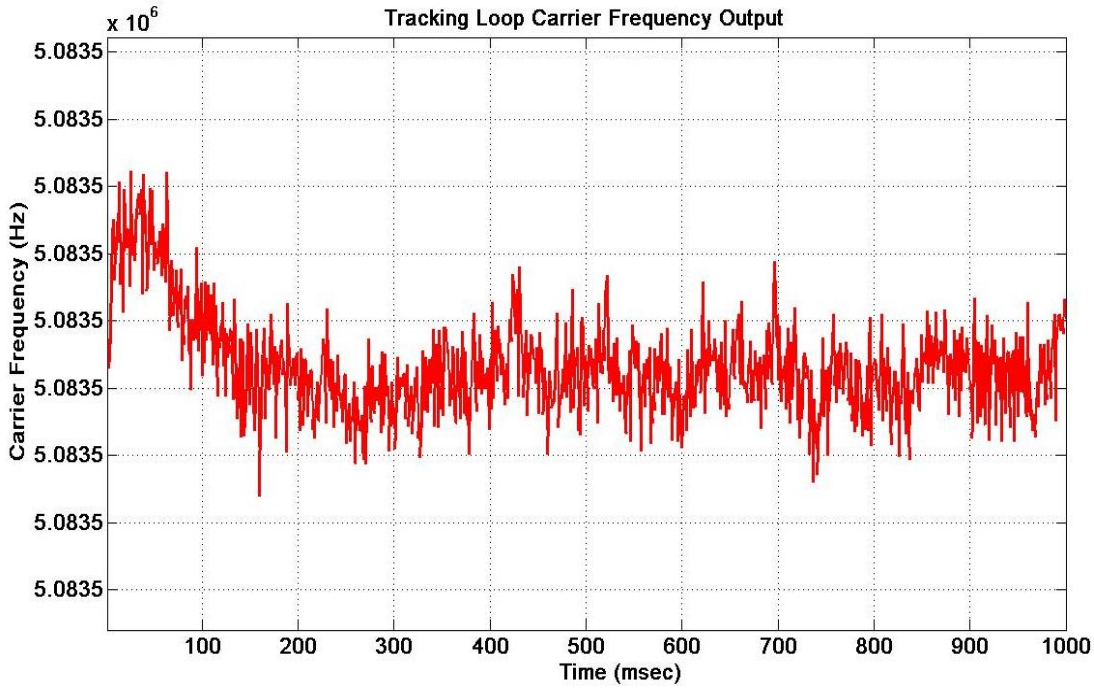

Figure 44 – GLONASS Tracking Loop – Carrier Frequency Output for Channel-1 (FDMA Channel Number-1) Satellite Signal in Time

Figure 45 shows PLL/FLL tracking indicators with decoded navigation bits from code tracking loop synchronized in time. Same lock indicators with GPS are implemented as given in Table 5. As seen in Figure 45, before signal tracking is achieved, FLL/PLL tracking values are not equal or close to 1 and with the achievement of signal tracking, decoded navigation data makes sense as seen in Figure 45 – c.

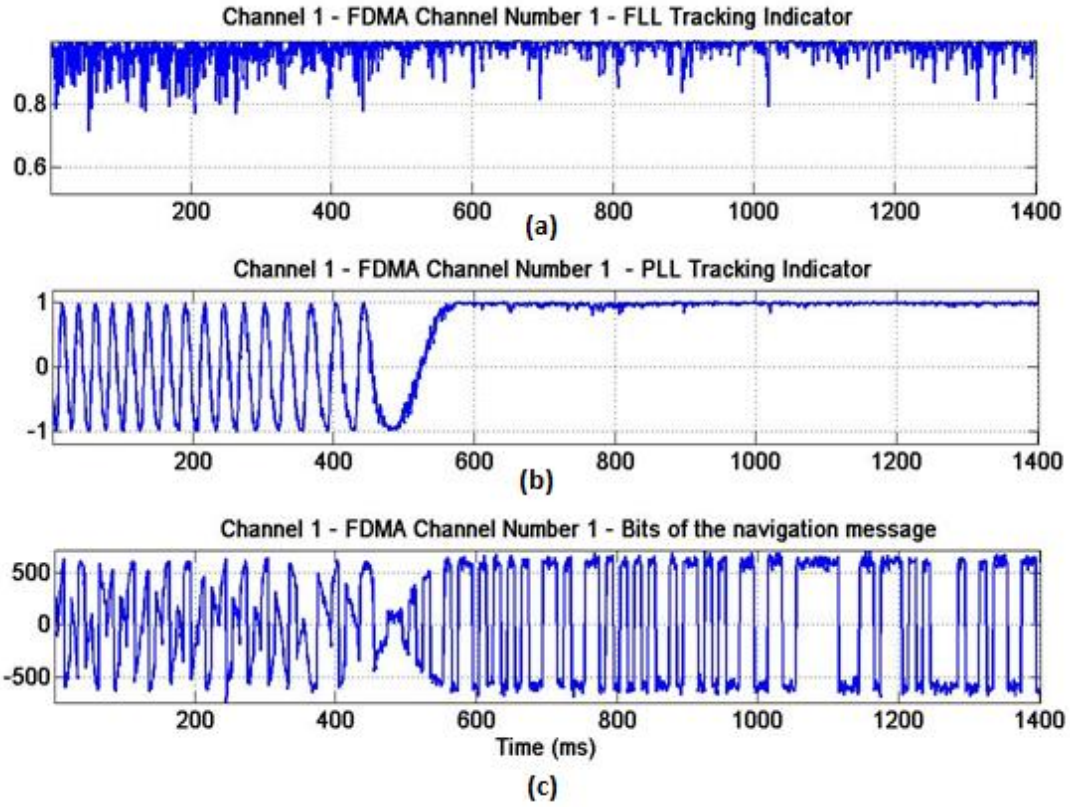

Figure 45 – GLONASS Signal Tracking Loop Outputs Synchronized in Time: (a) Change in FLL Tracking Indicator Value in Time (b) Change in PLL Tracking Indicator Value in Time (c) Decoded Unnormalized Navigation Bits (Receiver Channel-1/FDMA Channel Number-1) DLL IP (In-phase Prompt) Correlator Output

3.2.2.2 Code Tracking

The code tracking block diagram in Figure 15 is also implemented for GLONASS code tracking to get estimate of code phase of the GLONASS ranging code while it is being tracked. As seen in Figure 15, carrier and code tracking loops are coupled so that while DLL gets an accurate estimate of carrier frequency from FLL/PLL and carrier tracking loop uses a precise estimate of code phase from DLL.

Since locally generated signal is aligned to the in-phase of the incoming signal, all the signal energy is expected to be located in the in-phase arm with the achievement of tracking. This consistency is illustrated in Figure 46 and Figure 47.

Figure 46 shows that code tracking is achieved after some time since highest correlation value belongs to promptly generated code namely IP (In-phase Prompt) correlator output.

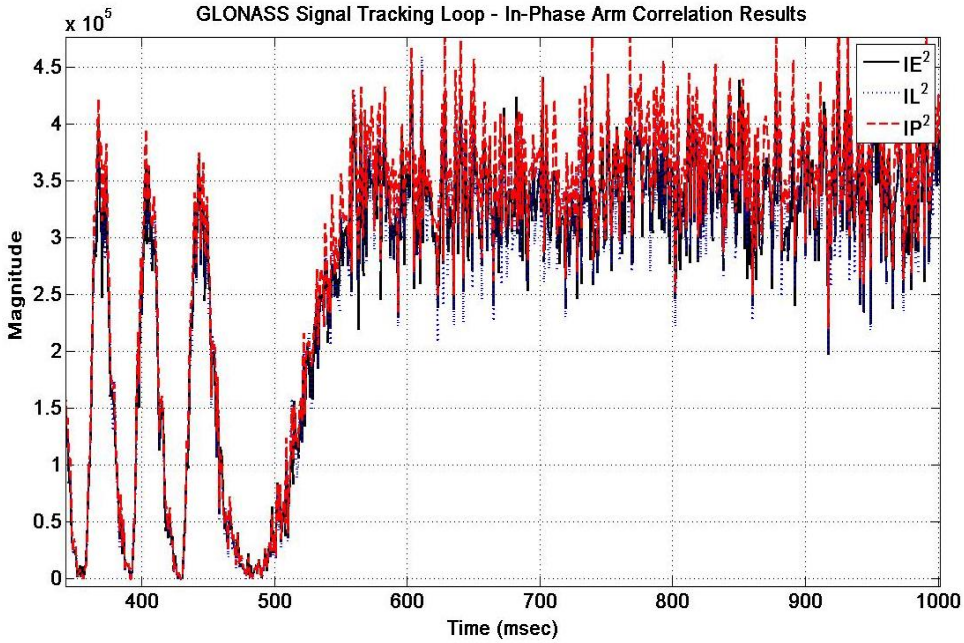

Figure 46 – GLONASS Tracking Loop of Receiver Channel-1/FDMA Channel Number-1 – Comparison of In-Phase Arm Correlation Results namely I-Early, I-Prompt and I-Late Correlator Outputs in Time

From Figure 47, it is inferred that the energy in quadrature arm decreases with the achievement of signal tracking by considering three correlation outputs in quadrature-arm, namely, QE (Quadrature-arm Early), QP (Quadrature-arm Prompt), and QL (Quadrature-arm Late). It means that locally generated signal is in phase with the incoming signal with the achievement of signal tracking.

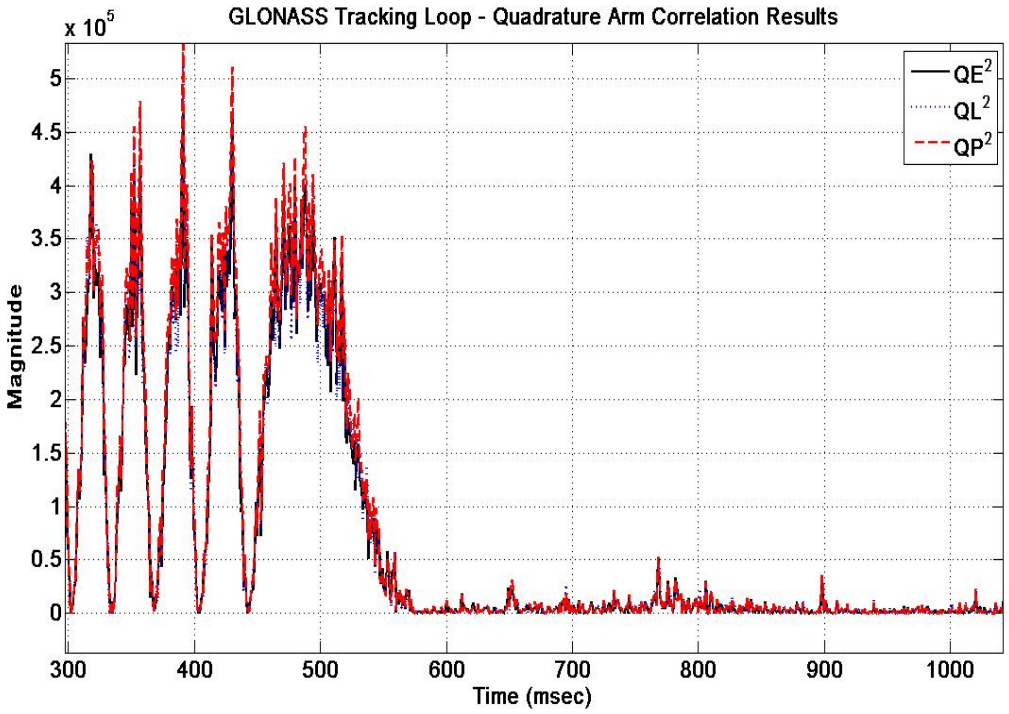

Figure 47 – GLONASS Signal Tracking Loop Outputs of Receiver Channel-1/FDMA Channel Number-1 – Comparison of Quadrature Phase Arm Correlation Results namely Q-Early, Q-Prompt and Q-Late Correlator Outputs

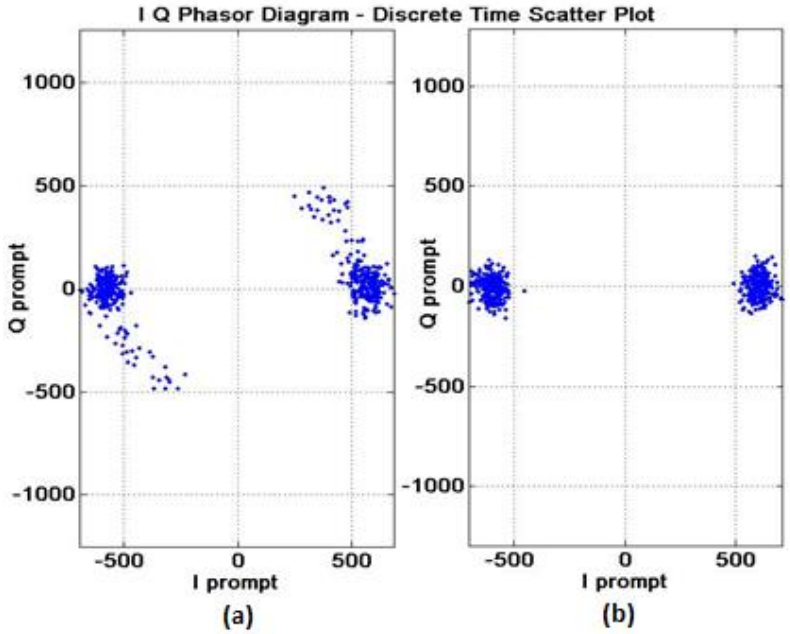

Figure 48 – GLONASS Tracking Loop Outputs of Receiver Channel-1/FDMA Channel Number-1 – IQ Phasor Diagrams (Discrete Time Scatter Plots): (a) PLL is not in phase lock (b) Locally generated carrier signal is in phase with the input signal

Two IQ phasor diagrams which belong to different time intervals of the scenario are plotted in Figure 48. Figure 48 shows the signal energy switch between in-phase and quadrature arms during carrier/code tracking as indicated in Figure 46 and Figure 47.

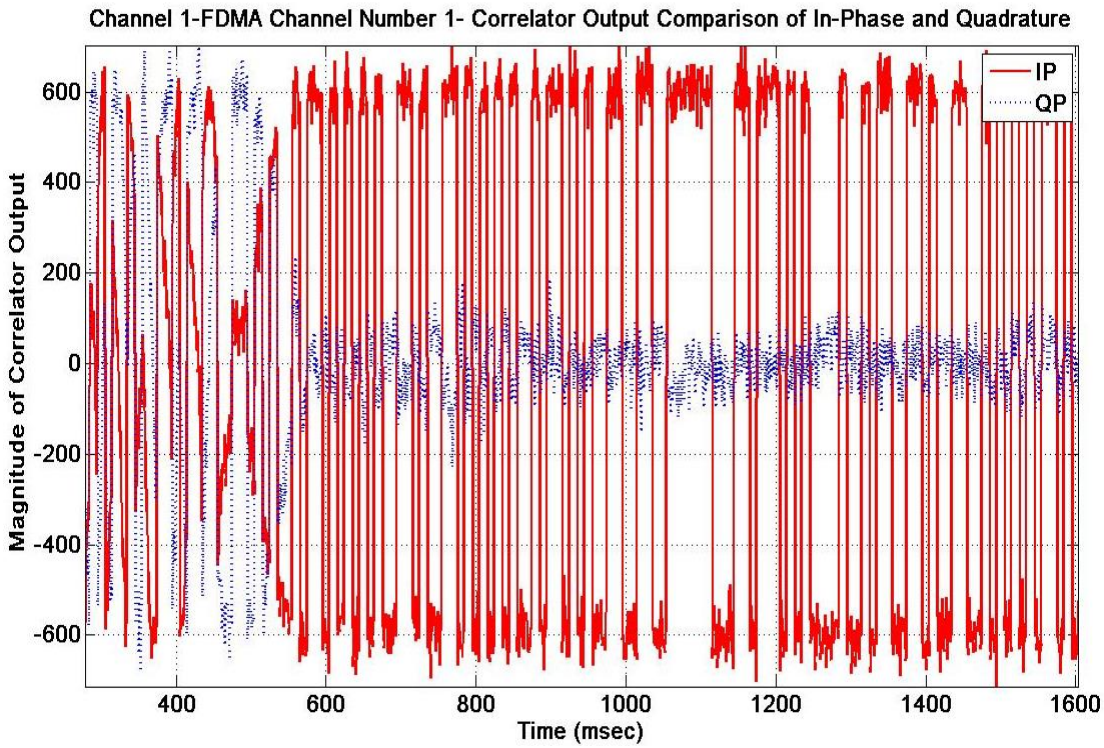

Figure 49 – GLONASS Tracking Loop – Comparison of In-Phase and Quadrature Arm Correlator Outputs (Receiver Channel-1/FDMA Channel Number-1)

Figure 49 shows the in-phase prompt (IP) and quadrature arm prompt (QP) correlation results in time. This code tracking results shows the unnormalized +1, -1 bit values in-phase arm. Therefore, for every millisecond bit values of navigation data can be obtained from tracking results output if tracking is achieved successfully.

Early/late/prompt correlator outputs are integrated, dumped and the integrator outputs are fed into the DLL discriminator as implemented in the block diagram in Figure 21. Same discriminator with GPS DLL is implemented for GLONASS code tracking. Code loop is 2^{nd} order PLL loop filter with 1 Hz noise bandwidth. However, carrier aiding to the output of code filter is different than applied in GPS. C/A code frequency and carrier frequencies are different than GPS and also due to FDMA nature, an adaptive scaling is applied for each carrier frequency.

Figure 50 shows the change in code frequency output provided by DLL for C/A Code of FDMA channel number-1 in receiver channel-1 satellite signal during code tracking. As expected, GLONASS C/A code frequency has a frequency of 0.511 MHz and settles in time.

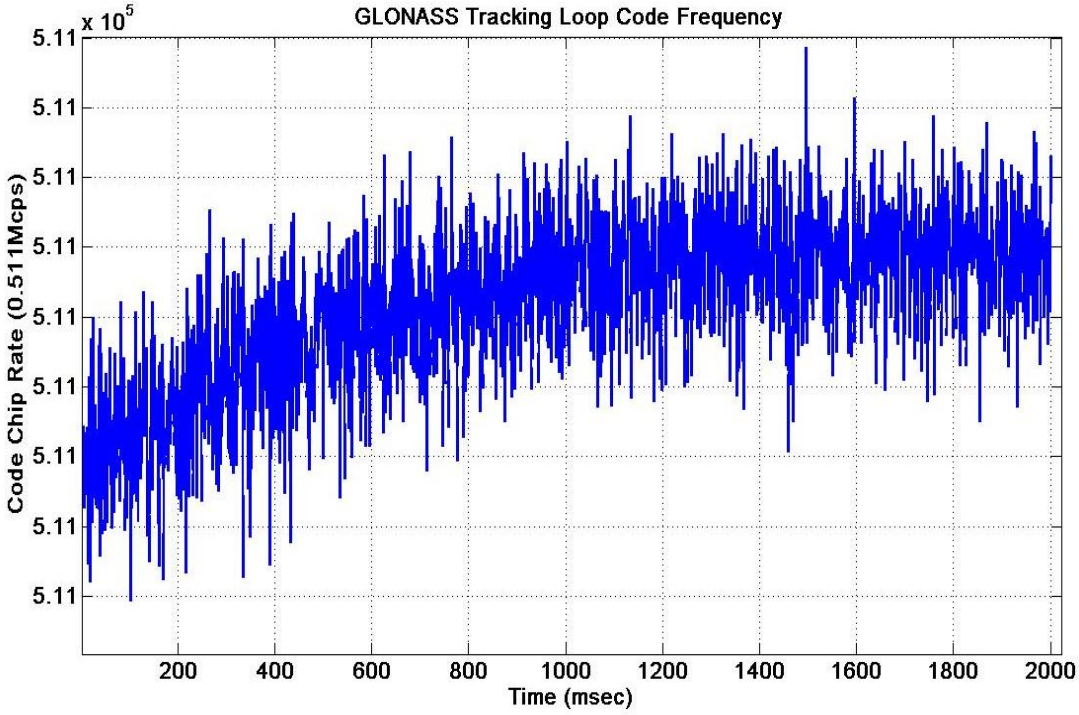

Figure 50 – GLONASS Tracking Loop – Code Frequency Output from GLONASS DLL Code Filter for C/A Code of FDMA Channel Number-1 Satellite Signal

3.2.3 Computation of User Position

Block diagram for position computation given in Figure 24 is also valid for GLONASS position computation. Same computation steps with some modifications for GLONASS system are implemented.

Figure 51 shows the basic transmitted navigation data structure from GLONASS satellites. Each frame under superframe has duration of 30 seconds and 15 strings. Each string has duration of 2 seconds as seen in Figure 52. During 1.7 seconds of a string, 85 data bits which are modulo-2 addition of 50 bps navigation data and 100 Hz auxiliary meander code are transmitted [7]. In last 0.3 seconds of each string, a time code is transmitted which can be thought as preamble in GPS case. Time mark is searched through navigation data with the check that whether time mark occurs in every 2 sec or not.

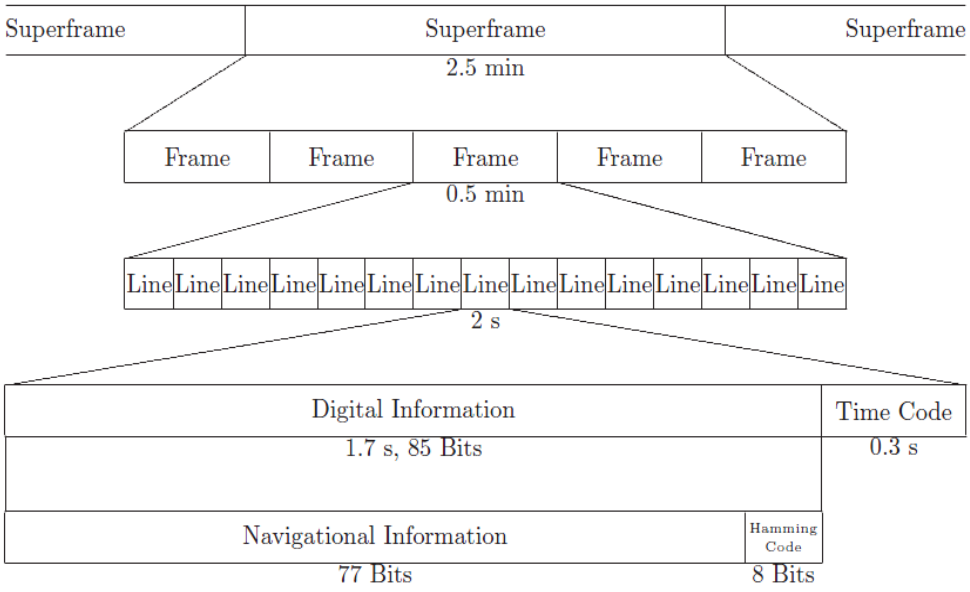

Figure 51 - Structure of the GLONASS C/A-Code Data Sequence [7]

After time marks are founded, bit synchronization is applied on the tracking results, because, tracking results provide unnormalized navigation data bits for every millisecond as plotted in Figure 49.

As being different from GPS case, there exists a meander code. During 1.7 seconds, 85 data bits which are modulo-2 addition of 50 bps navigation data and 100 Hz auxiliary meander code are transmitted as mentioned above. Therefore, conversion of 10 milliseconds data bits to 20 milliseconds data bits is required so that meander code is removed to reach navigation data.

Superframe structure which is shown in Figure 51 is detailed in Figure 52. When navigation data is obtained, considering number of bits, signs, and scale factors of all

parameters placed in strings, navigation data is decoded as given in GLONASS ICD document in reference [7].

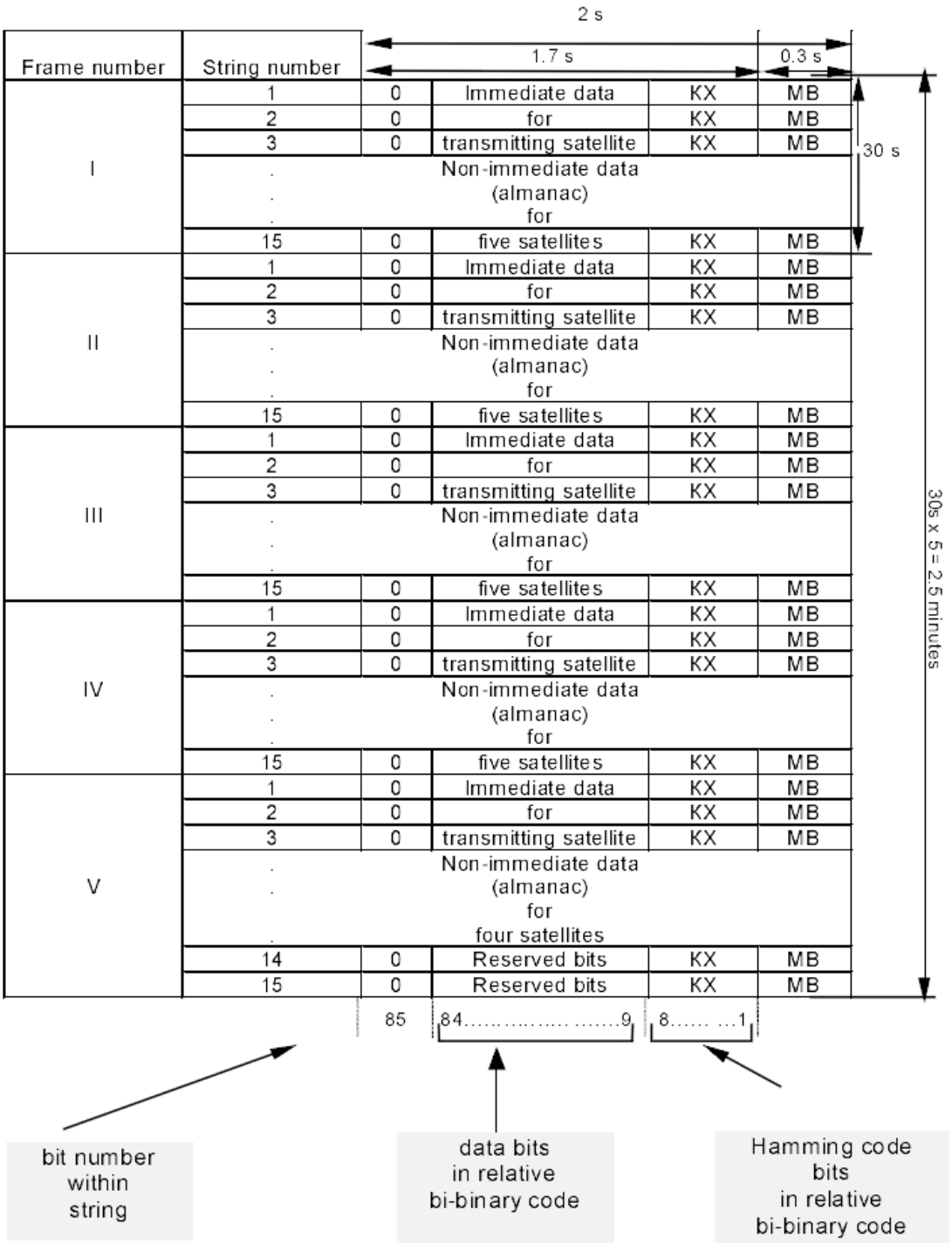

Figure 52 – GLONASS Superframe Structure [7]

From decoded navigation data, ephemeris parameters which will provide positions, velocity and acceleration vectors of the GLONASS satellites at an indicated reference time are needed to be extracted. Implemented ephemeris decoding algorithm and satellite position computation algorithm are tested. Figure 53 shows the test setup in block diagrams.

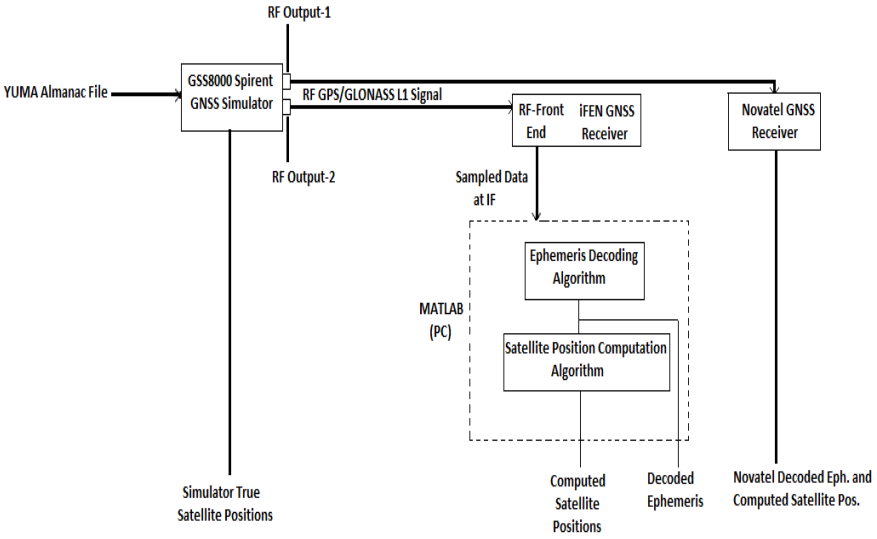

Figure 53 - Laboratory Test Setup for the Tests of Implemented Ephemeris Decoding and Satellite Position Computation Algorithms

YUMA Almanac files which provide Keplerian orbital parameters of all satellites for a specified time can be obtained on the Internet for both GPS and GLONASS satellites. All satellites broadcast same Almanac data containing all satellites' orbital parameters and clock information. However, each satellite broadcast its ephemeris data which includes more precise and detailed orbital information than almanac.

As seen in Figure 53, obtained Yuma Almanac file is fed into the simulator. Then for defined scenarios, an environment where those satellites in almanac are simulated in the correct locations at chosen date and time is created by simulator. Correct satellite positions in ECEF coordinate system are obtained from the simulator. Using two RF outputs simulated GNSS signals are fed into IFEN GNSS receiver and Novatel GNSS receiver at the same time. Sampled data at IF obtained from the RF front-end of IFEN GNSS receiver is used in the implemented algorithms at MATLAB. Novatel GNSS receiver provides both obtained ephemerides and computed satellite positions.

Table 14 – Comparison of Decoded Ephemeris Parameters' Values by Ephemeris Decoding Algorithm and Commercial GNSS Receiver

Ephemeris Parameters	**MATLAB Algorithm Result** [IF data obtained from IFEN GNSS Receiver] FDMA Channel Number-1		**Novatel GNSS Receiver**
t_k (hours)	3	[(3x3600)	10970 sec
t_k (minutes)	21	+ (21x60)	
t_k (seconds)	30	+30 = 12090]	[10800<11700<12600]
t_b	195 minutes [195*60 = 11700 seconds]		13 intervals with 15 minutes long (=195mins) [13*15*60 = 11700 seconds]
X Axis Position (PZ-90.02)	19406877.92968750 m		19406877.9296875 m
X Axis Velocity (PZ-90.02)	-1191.158294677734 m/s		-1191.15829467773 m/s
X Axis Acceleration (PZ-90.02)	0 m/s^2		0 m/s^2
Y Axis Position (PZ-90.02)	-12303303.71093750 m		-12303303.7109375 m
Y Axis Velocity (PZ-90.02)	970.4513549804688 m/s		970.451354980469 m/s
Y Axis Acceleration (PZ-90.02)	0 m/s^2		0 m/s^2
Z Axis Position (PZ-90.02)	11174048.82812500 m		11174048.8281250 m
Z Axis Velocity (PZ-90.02)	3129.803657531738 m/s		3129.80365753174 m/s
Z Axis Acceleration (PZ-90.02)	0 m/s^2		0 m/s^2
τ_n	4.95910644531e-005 sec		4.95910644531e-05 sec
$\Delta\tau_n$	0 sec		0 sec
γ_n	0		0

In Table 14, decoded ephemeris parameters' values by MATLAB algorithm and commercial Novatel GNSS receiver are presented together. It is inferred that ephemeris parameters are obtained correctly. GLONASS ephemerides provide position, velocity and acceleration vectors of the satellites at a given reference time, t_b .GLONASS Ephemeris parameters and computation of satellite positions are presented in APPENDIX B.

Comparison of computed positions of the satellite (FDMA channel number-1) by implemented MATLAB algorithm, Novatel GNSS receiver and the simulator data is given in Table 15. By considering results, it is concluded that the implemented MATLAB algorithm for computation of GLONASS satellites' positions is verified.

Table 15 – Comparison of Computed Positions of Satellite FDMA Ch.Number-1

GPS TIME 174087.000 (secs)	**MATLAB Alg.**	**Novatel GNSS Receiver**	**Correct Satellite Data from GNSS Simulator**
X Position (m) (PZ90.02)	18949335.1409917	18949336.2928000	18949339.9967000
Y Position (m) (PZ90.02)	- 11914545.6388394	-11914546.9160000	-11914551.2978000
Z Position (m) (PZ90.02)	12318994.1542105	12318995.5666000	12318981.6166000
X Velocity (m/s) (PZ90.02)	-1266.99920839467	- N/A	-1266.98000000000
Y Velocity (m/s) (PZ90.02)	1120.90790243311	- N/A	1120.88300000000
Z Velocity (m/s) (PZ90.02)	3025.33265903642	- N/A	3025.34800000000

In principle, position computation is same with the process in GPS such that satellite positions and pseudorange values are used to compute user navigation solution as shown in Figure 24. Applying the linearized observation model as described in APPENDIX C section, position solution is obtained.

Using computed user position, satellites' positions, elevation and azimuth angles, sky-plot is plotted in Figure 54.

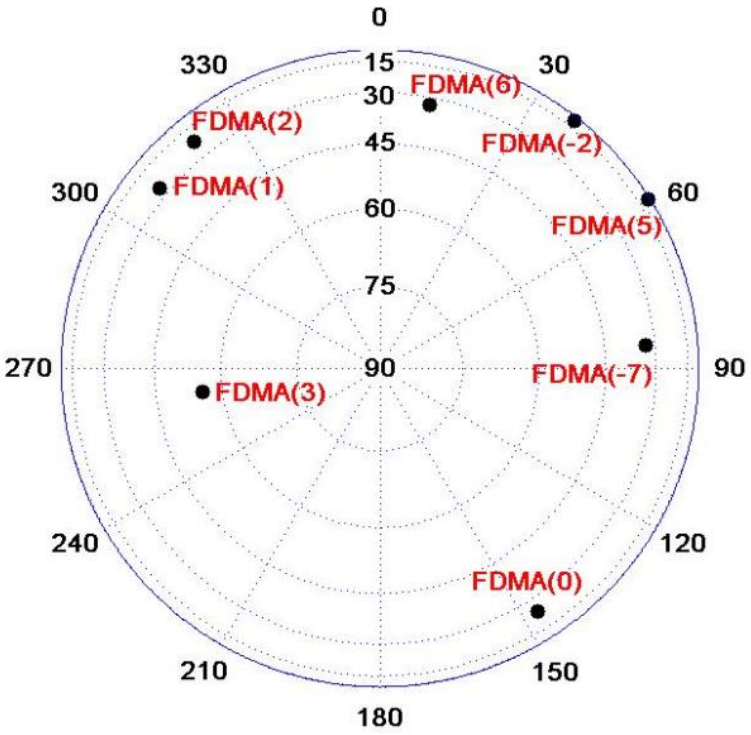

Figure 54 – Sky Plot of GLONASS Satellites Under Dynamic Scenario

Dilutions of Precision parameters (DOPs), which give idea of satellite geometry, are calculated during computation of position from least-square approach and presented in Table 16. Properties and characteristics of the DOP parameters are given in section 2.3.3 and the computation of those parameters are detailed in APPENDIX C.

Table 16 – Values of DOP Parameters Under GLONASS Dynamic Scenario

GDOP	2.94305624696432
PDOP	2.65272209403851
HDOP	2.23011384826591
VDOP	1.43649807934875
TDOP	1.27461584981346

3.3 Simulation Results

Positioning accuracy performances of implemented GLONASS algorithms is analyzed and the simulation results are shared in sections 3.3.1 and 3.3.2.

3.3.1 Static Scenario

Constructed laboratory setup as seen in Figure 28 is also used to test and verify the implemented GLONASS positioning algorithms.

As a remainder, in the test setup, simulated RF L1 GPS/GLONASS signals from simulator output are fed into IFEN GNSS receiver. IFEN GNSS receiver front-end provides sampled IF data in a file to the implemented MATLAB algorithms and also IFEN GNSS receiver computes its navigation solution.

Same static scenario which is used in the tests of implemented GPS algorithms is applied to the GLONASS positioning algorithms. Scenario is detailed as shared in Table 9.

Considering the navigation data outputs seen in Figure 28, GLONASS positioning solutions at MATLAB are compared with actual position obtained from the simulator and the navigation solution of the commercial GNSS receiver which IF data is obtained on the following graphs.

In Figure 55, obtained navigation solutions from GNSS simulator (true value), IFEN GNSS receiver and implemented MATLAB algorithm for GLONASS system are plotted in ECEF X-axis on the same graph.

In Figure 56, the error values of the navigation solutions of IFEN GNSS receiver and implemented MATLAB algorithms for GLONASS systems are plotted in ECEF X-axis on the same graph. They are calculated by subtracting computed values from actual true value provided by simulator. RMS position errors of IFEN GNSS receiver and GLONASS positioning MATLAB algorithms are 5.1786 meters and 4.4027 meters, respectively in ECEF X-axis.

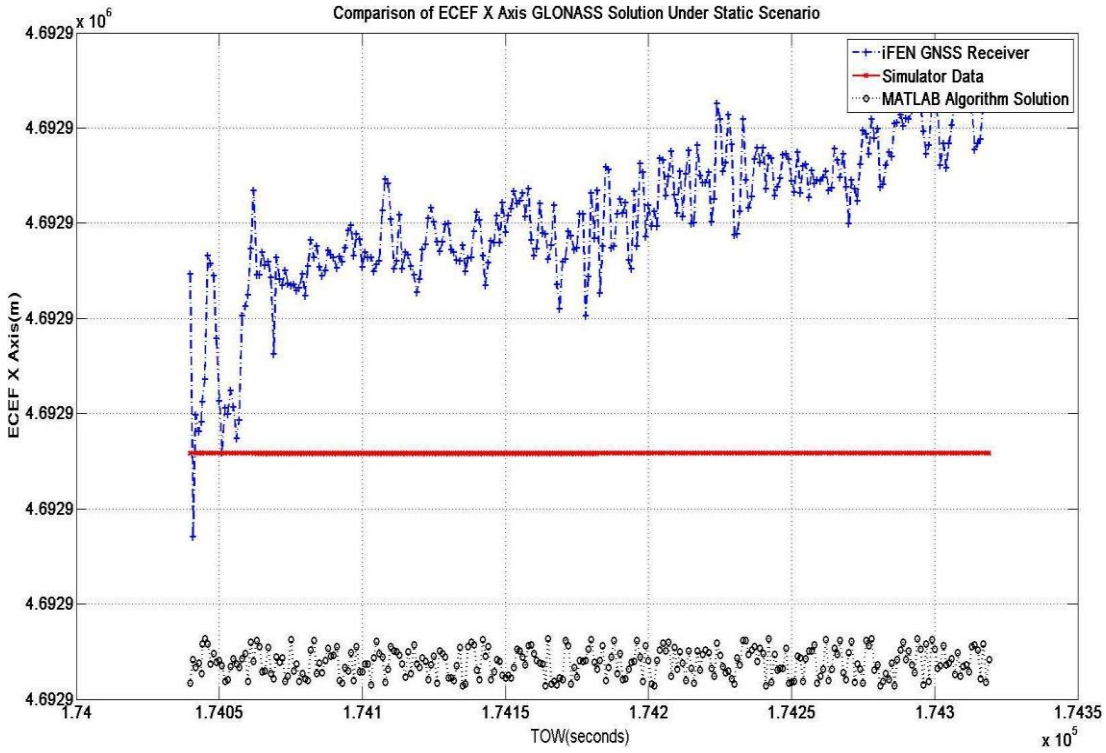

Figure 55 - Comparison of GLONASS Solutions in ECEF X Axis (Static Scenario)

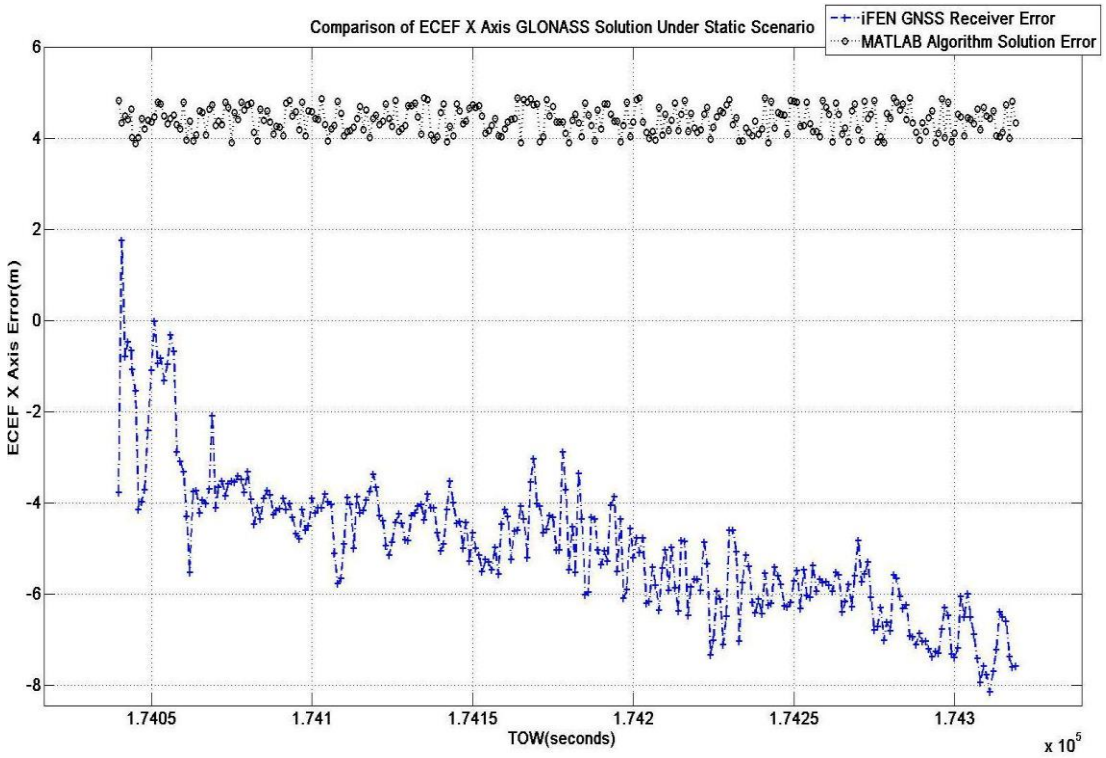

Figure 56 - Comparison of GLONASS Solutions in ECEF X Axis - Error Values (Static Scenario)

In Figure 57, navigation solutions of IFEN GNSS receiver, implemented MATLAB algorithms and the true value obtained from the simulator in ECEF Y-axis are plotted on the same graph.

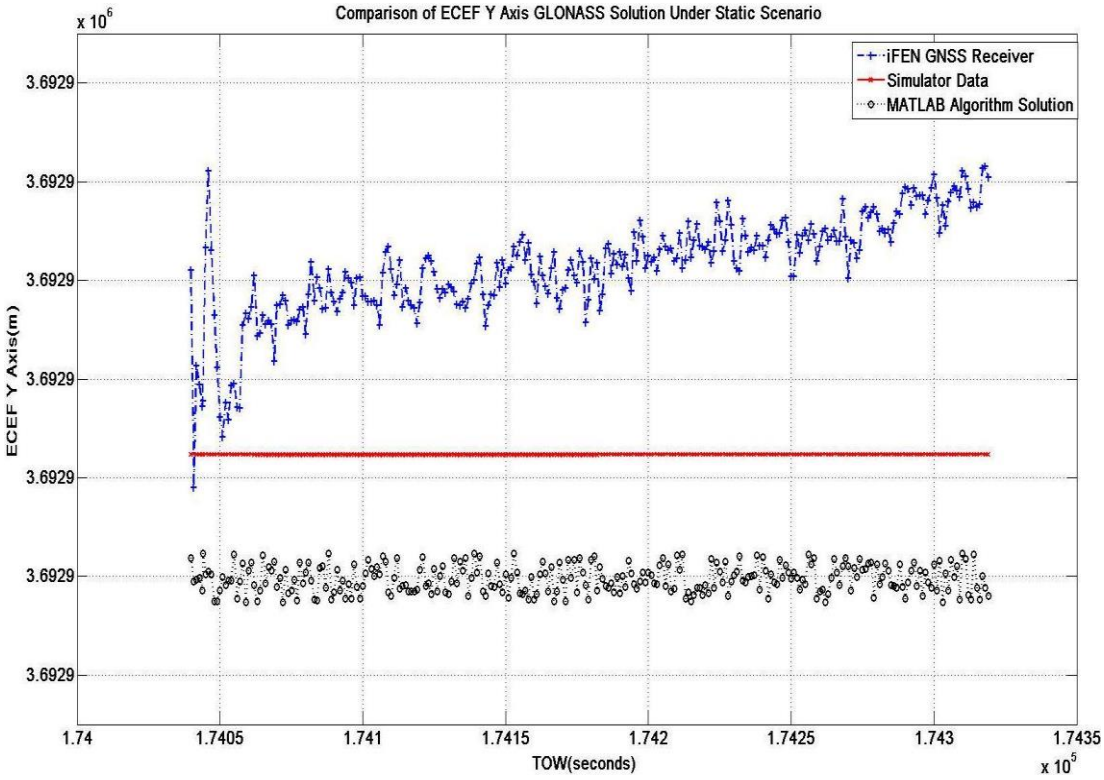

Figure 57 - Comparison of GLONASS Solutions in ECEF Y Axis (Static Scenario)

Figure 58 shows the error values of the navigation solutions of IFEN GNSS receiver and implemented MATLAB algorithms for GLONASS positioning in ECEF Y-axis. Each represents the difference which is computed by subtracting it from the true value provided by simulator. RMS position errors of IFEN GNSS receiver and MATLAB algorithms are 2.5349 meters and 3.9330 meters, respectively in ECEF Y-axis.

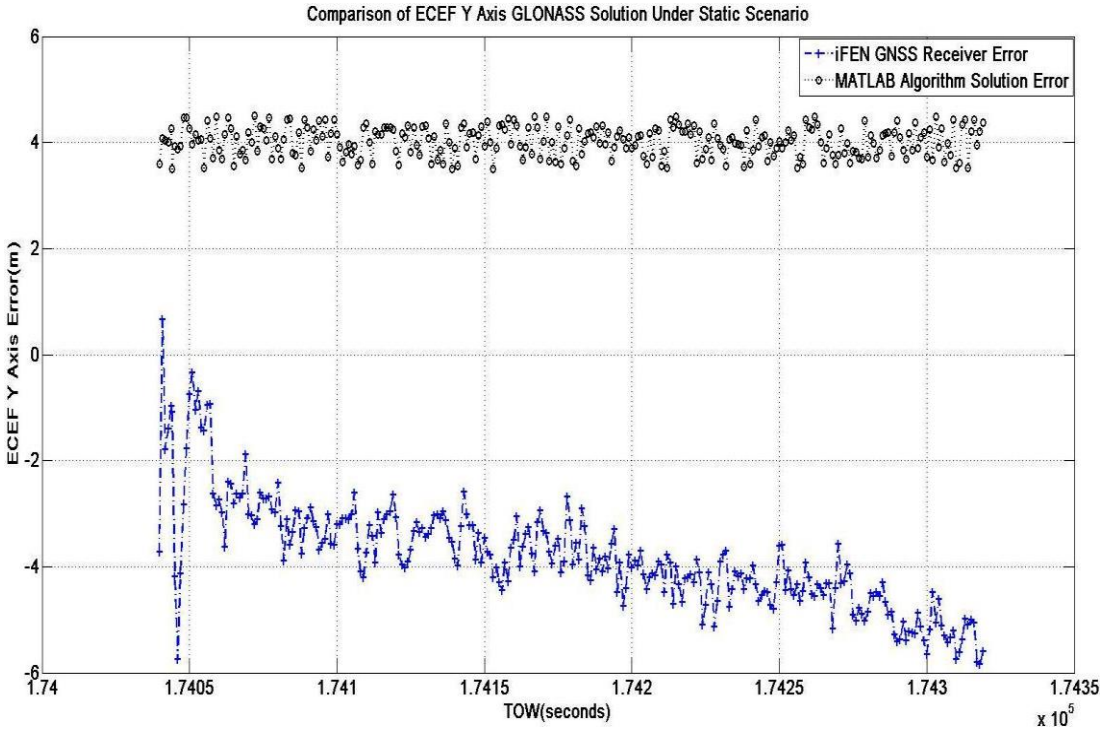

Figure 58 - Comparison of GLONASS Solutions in ECEF Y Axis - Error Values (Static Scenario)

In Figure 59, navigation solutions of IFEN GNSS receiver, implemented MATLAB algorithms and the true value obtained from the simulator are plotted in ECEF Z-axis on the same graph.

In Figure 60, error values of the navigation solutions of IFEN GNSS receiver and implemented MATLAB GLONASS positioning algorithms in ECEF Z-axis are plotted on the same graph. They are obtained by subtracting them from the actual value which the simulator provides. RMS position errors of IFEN GNSS receiver and MATLAB algorithms are 0.8272 meters and 3.4831 meters, respectively in ECEF Z-axis.

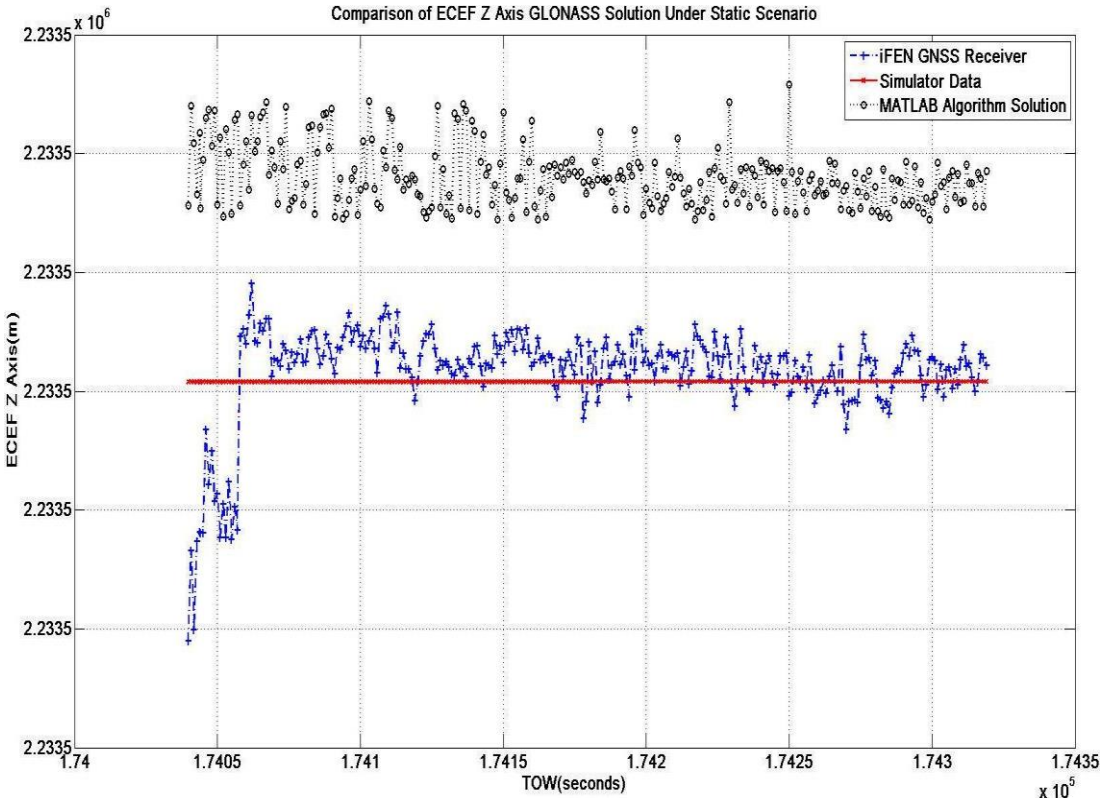

Figure 59 - Comparison of GLONASS Solutions in ECEF Z Axis (Static Scenario)

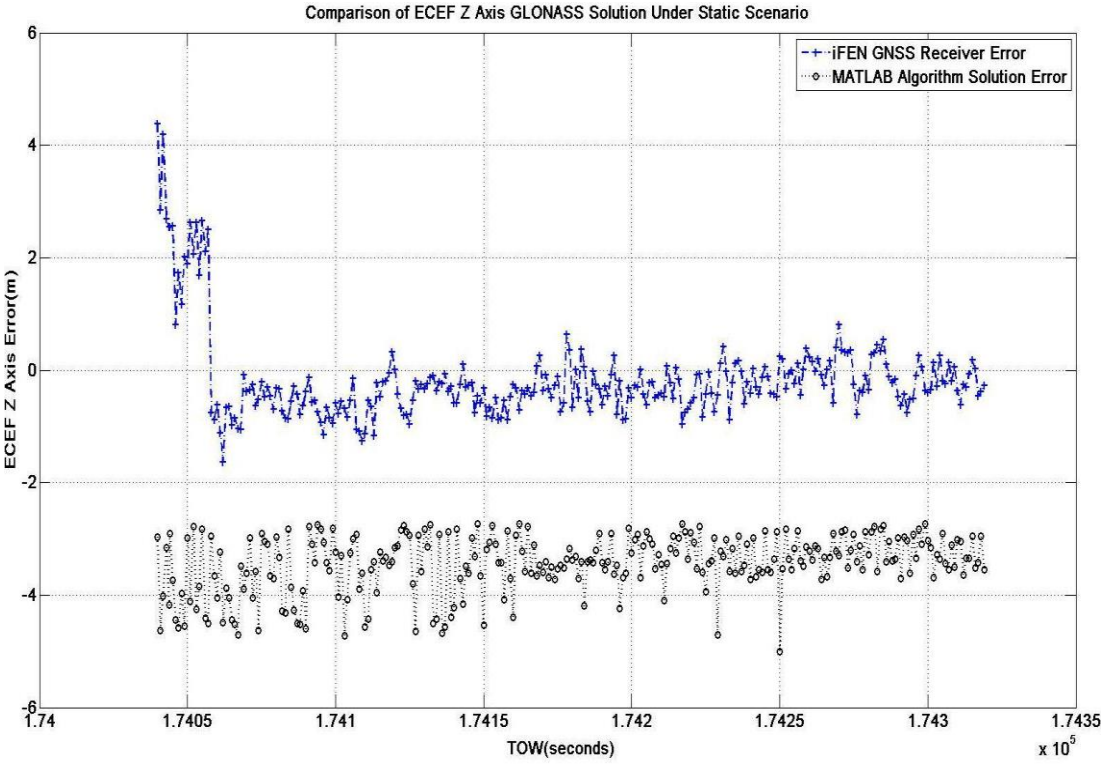

Figure 60 - Comparison of GLONASS Solutions in ECEF Z Axis - Error Values (Static Scenario)

To sum up, RMS position solution errors of implemented GLONASS positioning algorithms at MATLAB and IFEN GNSS receiver are presented for each axis in Table 17 below.

Table 17 – Root Mean Square Position Errors of GLONASS-only Algorithms

	X Error (m) (ECEF)	Y Error (m) (ECEF)	Z Error (m) (ECEF)
MATLAB Algorithm	4.4027	3.9330	3.4831
IFEN GNSS Receiver	5.1786	2.5349	0.8272

As seen in Table 17, it is deduced that the implemented MATLAB algorithms for GLONASS-only positioning work well in computation of navigation solution compared with a commercial GNSS receiver.

3.3.2 Dynamic Scenario

As mentioned in section 2.4.2, an aircraft flight scenario is created using aircraft motion command file GUI of GNSS simulator seen in Figure 35 in order to analyze the signal tracking performance of the algorithms under dynamic conditions. Aircraft flight motion is obtained in a file from the simulator after scenario is ended and obtained flight trajectory is plotted in Figure 36. Then implemented MATLAB algorithms' navigation solution can be compared with the navigation solution of IFEN GNSS receiver and true motion data obtained from the simulator.

The navigation solutions of the algorithms under dynamic scenario are plotted in each ECEF axis in Figure 61 below, including the true position obtained from the simulator and the commercial IFEN GNSS receiver's navigation solution on the same graph. As a remainder on how tests are performed, laboratory test setup in Figure 28 makes clear the plot below.

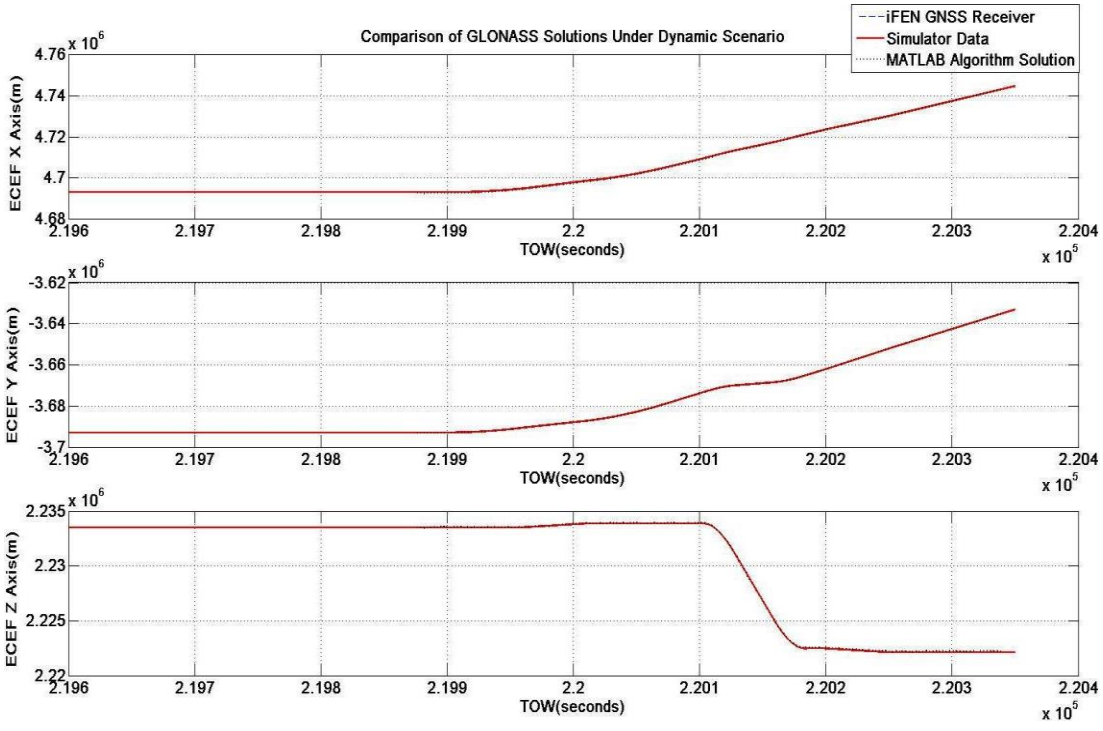

Figure 61 - Comparison of GLONASS Solutions (Dynamic Aircraft Scenario)

In implemented tests under dynamic scenarios it is observed that signal tracking algorithms still work well and addition of tracking lock status indicator-check in signal tracking loops is improved accuracy. All tracking loop outputs shared in

previous sections verify that the signal is being tracked successfully with indicator check.

Under dynamic scenario, with the addition of the indicator check, when the tracking is lost, the position solution is not computed which prevents false computation of navigation solution. Actually, when tracking is lost in a short time, errors in pseudorange values can grow faster which affects navigation solution adversely.

Loss and recovery points of the signal tracking and also navigation solution are decided by the values of tracking lock status indicators. When those points of the navigation solution are not taken into account, positioning performance of implemented algorithms are observed as nearly same as with the obtained performance under static scenario.

CHAPTER 4

GPS-GLONASS COMBINED POSITIONING ALGORITHMS

Positioning performances of implemented GPS and GLONASS algorithms are discussed in section-2 and section-3 of this thesis. In this section, outline of combining GPS/ GLONASS positioning algorithms is presented. Moreover, positioning performance of implemented GPS/GLONASS combined algorithm is analyzed.

In combining GPS and GLONASS measurements, there exist two problems that are needed to be solved, namely, differences in the time and the coordinate frames. Then, with having synchronized time and common coordinate frame for all satellites the algorithm in APPENDIX C is applied to have position solution. The requirements to reach combined solution and the solution methods are given in the first section.

In second section of this chapter, simulation results of combined solution compared with GPS-only and GLONASS-only solutions are provided.

4.1 Requirements for Combined Positioning

Different time frames and coordinate systems are employed by GPS and GLONASS systems. In order to combine navigation solution of GPS and GLONASS, it is essential to have common time frame and coordinate frame. In the following subsections, methods to combine them are defined.

4.1.1 Time Frames

GPS and GLONASS use different time scales that are related to UTC (Universal Time Coordinated) time. Time must be synchronized between both systems in order to have accurate combined navigation solution.

International Atomic Clock Time (TAI) was established as a reference time by BIH (Bureau International de l'Heure) [30].

GPS Time (GPST) is controlled by GPS control segment which monitors the atomic clocks on both satellites and ground stations.

There are 19 seconds time difference between TAI and GPST [31]:

$$TAI = GPST + 19.00sec \quad (4.1)$$

UTC is used to synchronize all used time signals to synchronize each other. By IERS (International Earth Rotation Service), computed numbers of seconds are added to TAI to tolerate the error due to the change in earth rotation speed considering the time when the Sun crosses Greenwich at noon [33]. UTC, unlike TAI, has real-time approximations maintained at the national laboratories [32].

GPS time is continuous time since 6th January 1980, and now it is ahead of UTC by 16 seconds:

$$GPST = UTC + t_{Leap} \quad (4.2)$$

$$TAI = UTC + 35$$

$$t_{Leap} : Leap\ second, currently\ 16\ secs$$

From equations 4.1 and 4.2, it can be said that currently TAI is ahead of UTC by 35 seconds.

Time used by all nations in world have a difference between UTC by an integer number or integer number and half an hour. This issue results in the definition of time zones. For example Moscow time and GMT (Greenwich Mean Time) has 3 hours shift.

While GPST is synchronized with UTC (USNO), GLONASS Time (GLONASST) is synchronized to UTC (SU) with an error less than 1 msec.

$$GLONASST = UTC(SU) + 3.00hour - \tau \quad 4.3$$
$$\tau < 1msec$$

Therefore, we can write the relation between GLONASST and GPSTIME using equations 4.3 and 4.4 as following:

$$t_{GLO} = t_{GPS} + 03h00\,\text{min} - t_{Leap} \quad 4.4$$
$$t_{GLO} : GLONASST\ (GLONASS\ Time)$$
$$t_{GPS} : GPST\ (GPS\ Time)\ ()$$
$$t_{Leap} : Leap\ second$$

Finally, time is synchronized between both systems using the relationship given in equation 4.4.

4.1.2 Coordinate Frame

GLONASS uses PZ-90.02 coordinate system as being different from WGS-84 used in GPS as given in Table 14. Although they are in both Earth-fixed coordinate frames, a transformation is needed between two coordinate systems.

Geodetic parameters of WGS-84 and PZ-90.02 are provided in Table 18 below:

Table 18 – Geodetic Parameters of WGS-84 and PZ-90.02 [29]

Parameter	WGS-84	PZ-90.02
Earth's rotation rate	$7.2921151467 \cdot 10^{-5}$	$7.292115 \cdot 10^{-5}$
Earth's gravitational constant	$3.986005 \cdot 10^{14}$	$3.986004 \cdot 10^{14}$
Semi-major axis	6378137	6378136
Flattening	1/298.256223563	1/298.25784
2nd Zonal Harmonic	$-484.16685 \cdot 10^{-6}$	$1082625.75 \cdot 10^{-9}$

As mentioned above, to obtain combined solution, satellite positions are needed to be transformed into a common frame. Since WGS-84 is more widely used, GLONASS solutions in PZ-90.02 are transformed into WGS-84. Up to 2007, GLONASS satellites positions were in PZ-90 coordinate system.

Conversion diagram from PZ-90 to WGS-84 and notation of coordinate systems are given in Figure 62 and in equation 4.5 below.

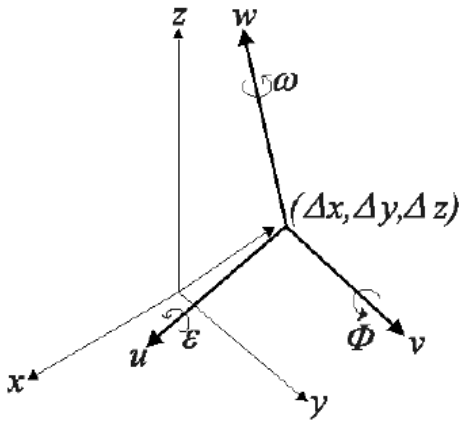

Figure 62 – Conversion from PZ-90 to WGS-84

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{WGS-84} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + (1+\delta s) \begin{bmatrix} 1 & \delta\omega & -\delta\phi \\ -\delta\omega & 1 & \delta\varepsilon \\ \delta\phi & \delta\varepsilon & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{PZ-90} \quad 4.5$$

$\delta\omega, \delta\varepsilon, \delta\phi : rotations\ between\ coordinate\ systems$

$\delta s : scale\ factor\ between\ coordinate\ systems$

$\Delta x, \Delta y, \Delta z : origin\ difference\ between\ coordinate\ systems$

Conversion from PZ-90.02 to WGS-84 is much simpler than PZ-90 conversion as following in equation 4.6. As being different from PZ-90 conversion, it only contains translations along each axis without rotations.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{WGS-84} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{PZ-90.02} + \begin{bmatrix} -0.36 \\ 0.08 \\ 0.18 \end{bmatrix} \quad 4.6$$

In sections 2 and 3, positioning algorithms for GPS and GLONASS systems are implemented independently from each other. Therefore, to have combined solution, WGS84 as a common coordinate system is used for both systems using the relationship given in equation 4.6.

4.2 Simulation Results

Same static and dynamic scenarios under which GPS and GLONASS algorithms are tested are applied to the combined GPS/GLONASS algorithms and performances are compared in following subsections.

4.2.1 Static Scenario

The positioning performance of combined algorithm is presented in following graphs under same static scenario test which is applied to the implemented GPS-only and GLONASS-only navigation algorithms.

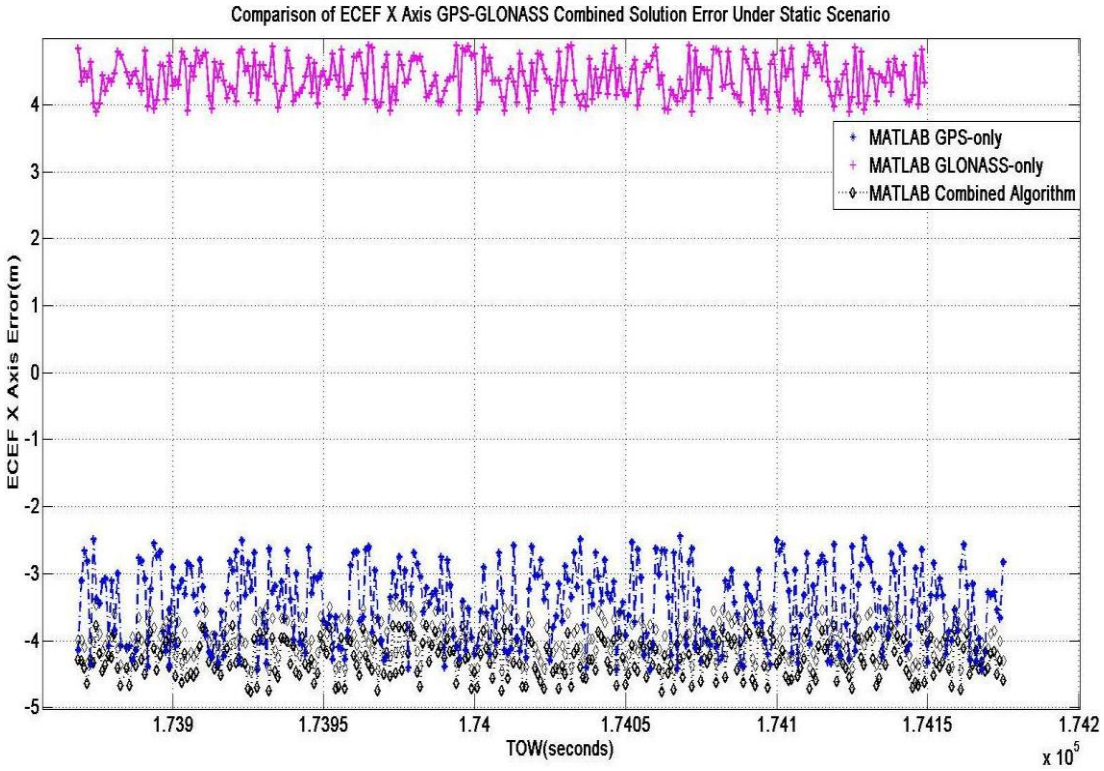

Figure 63 - Comparison of GPS-GLONASS Combined Solution and Separate Solutions in ECEF X Axis - Error Values (Static Scenario)

Figure 63 shows the error values (in ECEF X-axis) of the navigation solutions obtained by implemented MATLAB algorithms on the same graph. RMS position errors in ECEF X-axis of GPS-only, GLONASS-only and GPS-GLONASS combined algorithms are 3.5262 meters, 4.4027 meters and 3.9739 meters respectively.

In Figure 64, error values of the navigation solutions of implemented GPS-only, GLONASS-only and GPS-GLONASS combined MATLAB algorithms in ECEF Y-axis are plotted on the same graph. RMS position errors in ECEF Y-axis of GPS-only, GLONASS-only and GPS-GLONASS combined algorithms are 2.5815 meters, 3.9330 meters and 3.4069 meters respectively.

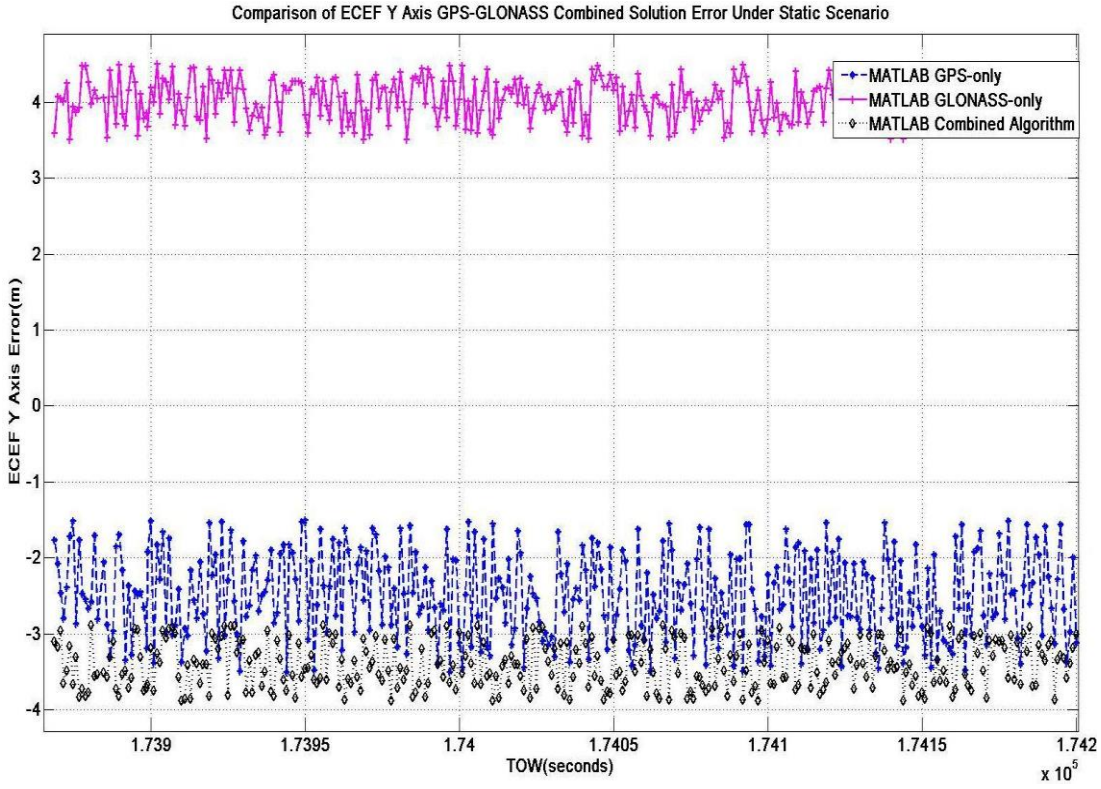

Figure 64 - Comparison of GPS-GLONASS Combined Solution and Separate Solutions in ECEF Y Axis - Error Values (Static Scenario)

In Figure 65, error values of the navigation solutions of implemented GPS-only, GLONASS-only and GPS-GLONASS combined MATLAB algorithms are plotted in ECEF Z-axis on the same graph. They have 2.2762 meters, 3.4831 meters and 2.6901 meters RMS errors in ECEF Y-axis, respectively.

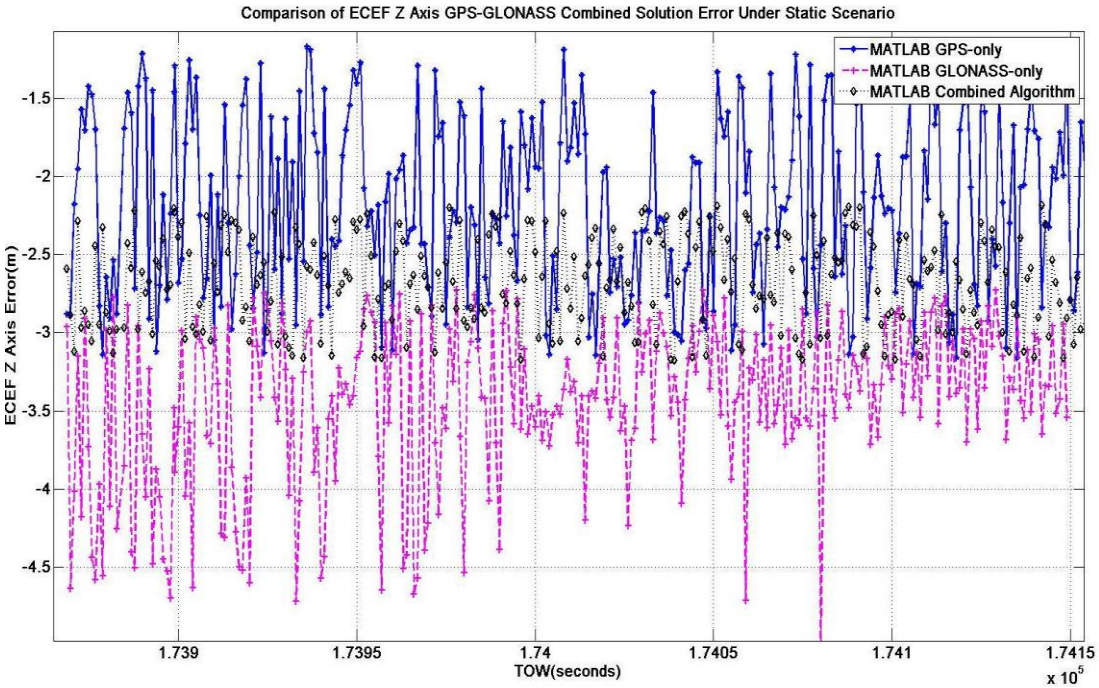

Figure 65 - Comparison of GPS-GLONASS Combined Solution and Separate Solutions in ECEF Z Axis - Error Values (Static Scenario)

To sum up, RMS position solution errors of all implemented algorithms in each axis are shown in Table 19 below.

Table 19 – Comparison of Root Mean Square Position Errors of GPS-GLONASS Combined, GPS-only and GLONASS-only Algorithms

Implemented Algorithm	X Error (m) (ECEF)	Y Error (m) (ECEF)	Z Error (m) (ECEF)
GPS-only Algorithm	3.5262	2.5815	2.2762
GLONASS-only Algorithm	4.4027	3.9330	3.4831
GPS-GLONASS Combined Algorithm	3.9739	3.4069	2.6901

As can be seen in Table 19, GPS-only solutions are better than GLONASS-only solutions. Errors in the navigation solution of GPS/GLONASS combined algorithm have approximate values to the errors in the solution of GPS-only algorithm. However, errors in the navigation solution of GLONASS-only algorithm are a bit more than the values encountered in other two algorithms' results.

4.2.2 Dynamic Scenario

Using same aircraft flight trajectory which is plotted in Figure 36 dynamic scenario is created and also applied to the combined algorithm. No significant difference in performance when compared with the performance of static case is observed.

Sky-plot is given in Figure 66, below.

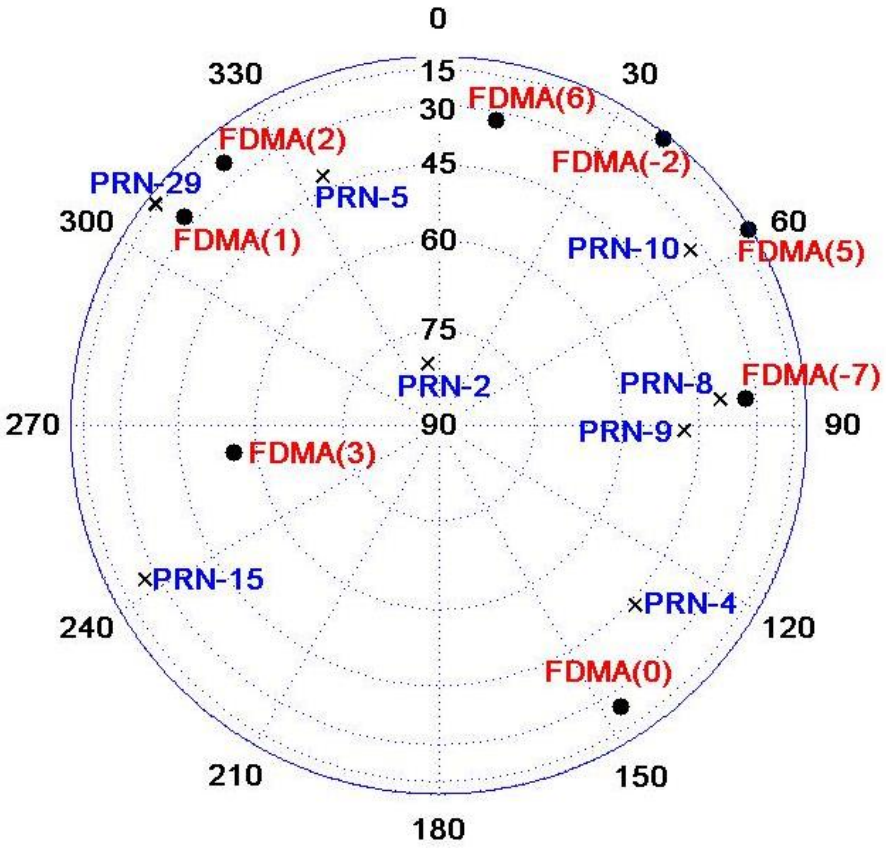

Figure 66 - Sky Plot for GPS/GLONASS satellites Under Dynamic Scenario

As mentioned in section 2.3, DOP values describe the precision of many components of the time and position solution and they represent the effect of the measurement errors onto the solution.

In Table 20, incremental benefit of the GLONASS signals to the combined solution can be seen on DOP values.

Table 20 – Values of DOP Parameters (GPS/GLONASS Combined) Under Dynamic Scenario

	GPS-only	GLONASS-only	GPS-GLONASS Combined
GDOP	2.6479	2.9430	1.4417
PDOP	2.3327	2.6527	1.2928
HDOP	2.0116	2.2301	1.1141
VDOP	1.1810	1.4364	0.6557
TDOP	1.2529	1.2746	0.6382

It is deduced from Table 20, combined case has better satellite coverage with maximized volume by tracked satellites and so that it has smaller DOP values than other two cases. Due to the fact that these values are useful to classify the algorithms' ability for positioning, it is concluded combined GPS-GLONASS case has better ability for obtaining an accurate user position.

CHAPTER 5

TWO RECEIVER POSITIONING ALGORITHMS

Chapters 3, 4 and 5 have revealed the positioning performances of code phase-based GPS, GLONASS and combined GPS-GLONASS algorithms. In the literature, there are studies where carrier based DGPS (Differential GPS) algorithms are applied in order to improve the positioning accuracy.

In this chapter, to achieve positioning accuracy in centimeter level with two-receiver configuration, a DPGS augmentation LAMBDA (Least-squares Ambiguity De-correlation Adjustment) method for GPS L1 single frequency approach is provided.

In the first section, definitions of DGPS methods and background information are provided.

Comparison of carrier phase and pseudorange measurement models and the theoretical background of LAMBDA method are given in section 5.2.

In section 5.3, positioning performances of code phase-based algorithms and LAMBDA method for two receiver configurations are compared in terms of accuracy of computed baseline distances.

The field test results of LAMBDA method for different baselines and performance comparison with Magellan GNSS Receiver sets are involved in section 5.4.

5.1 Differential Global Positioning System (DGPS) and Carrier Based Methods

If a certain level of accuracy is expected in an application, single frequency GPS/GLONASS or combined solution will not be adequate to attain that accuracy. For those applications that require high levels of accuracy, augmentation is almost obligatory. There are two general types of augmentation, namely, DGPS (Differential GPS) and external sensors/systems [5].

5.1.1 Differential Global Positioning System (DGPS)

Figure 67 shows the schematic of differential GPS. If the position of a user or a GPS receiver is known, for visible satellites related satellite clock, atmospheric delay, and ephemeris errors can be estimated accurately.

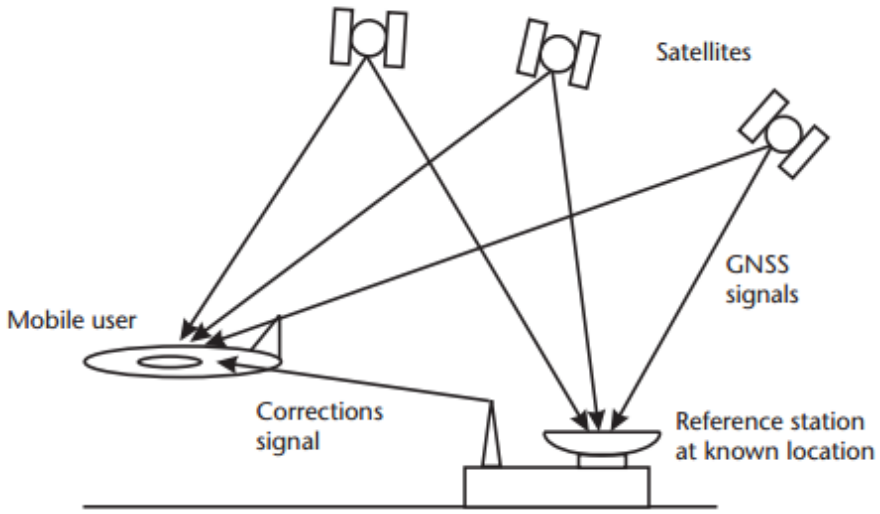

Figure 67 - Schematic of Differential GPS (Figure is adapted from [10])

Satellite related errors such as on satellite clock, ephemeris, and atmospheric delay have similar values for the close by users. Corrections on those errors can be sent to the users for position improvement as seen in Figure 67.

DGPS systems can use code or carrier measurements. As expected from the names, code based DGPS systems are based on code pseudorange measurements, carrier phase DGPS systems use phase measurements.

5.1.2 RTK (Real Time Kinematic)

RTK is a positioning technique which solves position using observations from both a reference station and a platform nearby reference station. Having the same idea with DGPS, it utilizes from both reference station and a platform of which navigation solution is computed according to the reference point.

A link is needed between base (reference) and rover (platform) stations to transmit and receive observations in real time. Additionally, measurements are needed to be synchronized in time. Exchanging raw data between base and rover stations will provide analyzing the same errors which are accepted as nearly same within short distances with respect to each other.

5.1.3 PPK (Post Processed Kinematic)

PPK resembles the method RTK. However, in PPK, all algorithms are applied to the obtained raw data in post processing not in real-time. PPK still requires one reference point.

In this thesis, one of PPK methods is implemented by creating a post-processing solution using acquired data from both reference receiver and a user receiver.

The algorithm implementations are detailed in section 5.2.

5.2 The LAMBDA Method

There are some precautions that we need to take to improve the accuracy and precision of the GNSS navigation solution.

In order to obtain a more accurate solution, ionospheric/tropospheric errors, position errors of the satellites and receiver/satellite clock errors are needed to be reduced. Since precise carrier phase measurement provides much more accurate measurements than code phase measurement, user position accuracy in centimeter level can be reached with carrier based algorithms.

Therefore, carrier phase measurements that will be provided in high precision can be combined with code phase measurements in two receiver configuration to minimize all errors.

In GPS receiver positioning techniques, there are two observables, namely pseudorange and carrier phase as illustrated in Figure 68.

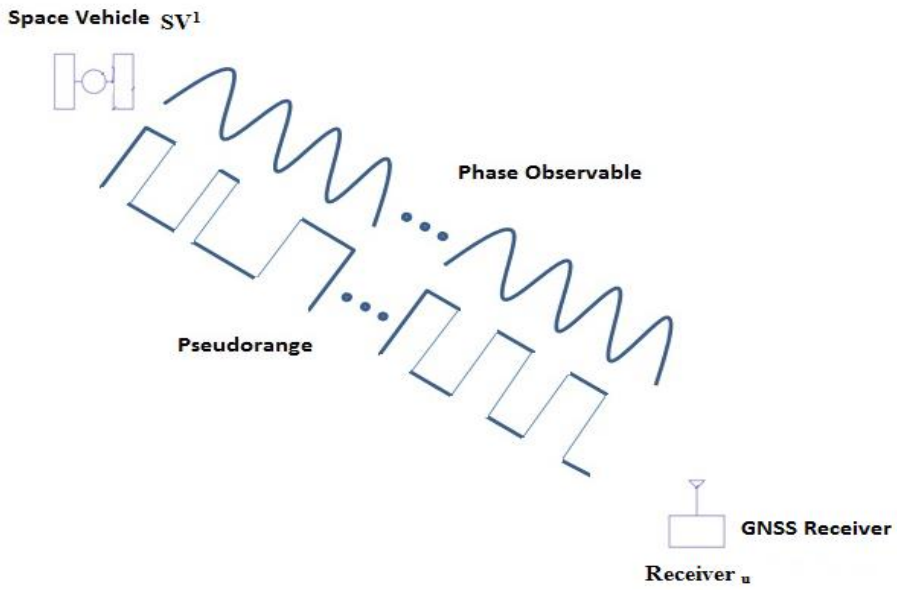

Figure 68 - Code Pseudorange and Carrier Phase Observables

Code and carrier phase measurements differ from each other as modelled in following equations 5.1 and 5.2:

- ✓ Code Pseudorange Measurement Model:

$$\rho_u^1(t) = r_u^1 + I_u^1 + T_u^1 + c\left(dt_u(t) - dt^1(t-\tau_u^1)\right) + \varepsilon_{\rho_u}^1 \qquad 5.1$$

r_u^1 : *Actual distance between satellite-1 and user receiver*

I_u^1 :*Ionospheric delay*

T_u^1 : *Tropospheric delay*

c : *Speed of light*

dt^1 : *Satellite clock offset*

τ_u^1 : *Signal travel time from the satellite-1 to the user receiver*

dt_u :*User receiver clock offset*

ε : *Measurement noise*

- ✓ Carrier Phase Measurement Model:

$$\lambda\phi_u^1(t) = r_u^1 - I_u^1 + T_u^1 + c\left(dt_u(t) - dt^1(t-\tau_u^1)\right) + \lambda N_u^1 + \varepsilon_{\phi_u}^1 \qquad 5.2$$

λ : *Wavelength*

N :*Carrier phase bias or unknown integer ambiguity*

$\phi_u^1(t)$: *Integrated carrier phase*

In above equations, ionospheric errors are same in magnitude but different in sign. Tropospheric errors, satellite/receiver clock errors and actual distances are all same in equations 5.1 and 5.2. Although multipath effect is disregarded in both equations, they are not same in carrier and code measurements. Furthermore, noise terms are not same and phase data noise is much smaller.

Unknown ambiguity N does not appear in code pseudorange measurement in equation 5.1. If integer ambiguity term N is known, carrier phase measurement can be used as pseudorange.

5.2.1 Double-Differenced Phase Solution

Centimeter level accuracy to an accurately known position is intended to be reached in two receiver configuration with the implementation of LAMBDA method. Post-processing operation on acquired data from both receivers (base and rover) is used through implementation of the method.

With the idea that many pseudorange and carrier phase errors are same at two nearby locations, double differenced phase solution with LAMBDA method is implemented. Double difference phase solution is created with two satellites and base/rover receivers as seen in Figure 69 below.

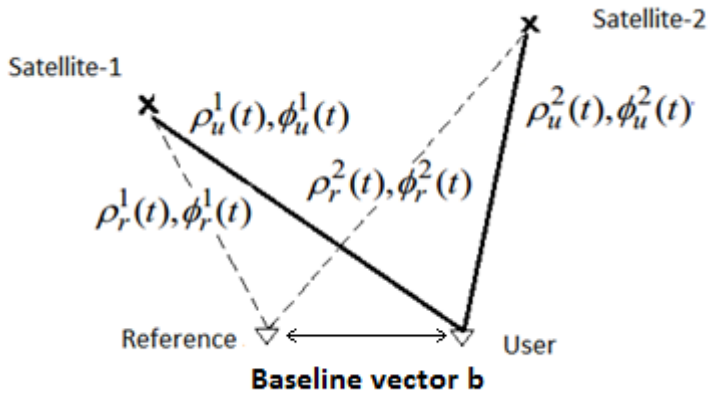

Figure 69 – Formed Double Differenced Phase Solution with Two Satellites and Base/Rover Receivers

Carrier phase measurements of reference and user receivers are written for SV1 in Figure 69 as follows:

$$\lambda\phi_r^1(t) = r_r^1 - I_r^1 + T_r^1 + c\left(dt_r(t) - dt^1(t-\tau_r^1)\right) + \lambda N_r^1 + \varepsilon_{\phi_r}^1 \quad 5.3$$

$$\lambda\phi_u^1(t) = r_u^1 - I_u^1 + T_u^1 + c\left(dt_u(t) - dt^1(t-\tau_u^1)\right) + \lambda N_u^1 + \varepsilon_{\phi_u}^1 \quad 5.4$$

$$\lambda\Delta\phi_{ru}^1(t) = r_r^1 - r_u^1 + c\left(dt_r(t) - dt_u(t)\right) + (I_u^1 - I_r^1) + (T_r^1 - T_u^1) + \lambda(N_r^1 - N_u^1) + (\varepsilon_{\phi_r}^1 - \varepsilon_{\phi_u}^1) \quad 5.5$$

$$\lambda\Delta\phi_{ru}^1(t) = \Delta r_{ru}^1 + c\left(dt_{ru}(t)\right) + \lambda\Delta N_{ru}^1 + \Delta\varepsilon_{\phi_{ru}}^1 \quad 5.6$$

Signal travel times from the satellite to base and rover receivers, τ_r^1 and τ_u^1, are not same due to the fact that they represent different ranges. However, for 300km distance between base and rover, this difference will be maximum 1ms and for short distances it can be neglected.

Single difference equation can be written as following from equation 5.6:

$$SD_{ru}^1 = \Delta r_{ru}^1 + c dt_{ru}(t) + \lambda \Delta N_{ru}^1 + \Delta \varepsilon_{\phi_{ru}}^1 \quad 5.7$$

Double difference is the difference of single differences for satellites 1 and 2:

$$DD_{ru}^{12} = SD_{ru}^1 - SD_{ru}^2 = \Delta r_{ru}^{12} + \lambda \Delta N_{ru}^{12} + \Delta \varepsilon_{\phi_{ru}}^{12} \quad 5.8$$

As seen in equation 5.8, receiver and satellite clock errors are cancelled. Ionospheric and tropospheric delays are assumed as same due to short distance between them. To use carrier phase data, we must estimate the ambiguity term.

Using combination of carrier phase and pseudorange data with ephemeris to determine position of satellites, ambiguity resolution method can be implemented. LAMBDA (Least-squares Ambiguity De-correlation Adjustment) method is one of frequently used techniques for ambiguity resolution.

At least 4 double-difference equations are needed for baseline computation. With five satellites, four DDs are constituted and the configuration is as follows:

$$\begin{bmatrix} DD_1 \\ DD_2 \\ DD_3 \\ DD_4 \end{bmatrix} = \underbrace{\begin{bmatrix} e_{12_x} & e_{12_x} & e_{12_x} \\ e_{13_x} & e_{13_y} & e_{13_z} \\ e_{14_x} & e_{14_y} & e_{14_x} \\ e_{15_x} & e_{15_x} & e_{15_x} \end{bmatrix}}_{H} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} \lambda + e \quad 5.9$$

$DD : carrier - phase\ double\ differences$

$b : Baseline\ vector\ (in\ ECEF\ frame)$

$H : Data\ matrix - differenced\ unit\ vectors\ between\ the\ satellites$

$N : Unknown\ double\ difference\ ambiguities$

$\lambda : Wavelength$

$e : Measurement\ errors$

The measurement errors term is disregarded for simplicity through the procedure

Linearized DD equations can be formed into following forms: [12]

$$y = [B \quad A]\begin{bmatrix} b \\ a \end{bmatrix} = B_{m\times 3} \cdot b_{3x1} + A_{m\times n} \cdot a_{n\times 1} \tag{5.10}$$

$n : Number\ of\ satellites$

m: $Number\ of\ DD\ measurements\ which\ is\ equal\ to\ n-1$

$y : Double\ differenced\ observations\ minus\ computed\ ones$

$B : Design\ matrix\ holding\ LOS\ components$

$A : Design\ matrix\ to\ relate\ ambiguity\ terms\ to\ DD\ carrier\ phase\ meas.$

$a : n\ double\ difference\ ambiguities$

The normal equation to least square problem is founded.

$$\begin{bmatrix} B^T \\ A^T \end{bmatrix}[B \quad A]\begin{bmatrix} \hat{b} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} B^T \\ A^T \end{bmatrix} y \xrightarrow{yields} \begin{bmatrix} B^T B & B^T A \\ A^T B & A^T A \end{bmatrix}\begin{bmatrix} \hat{b} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} B^T \\ A^T \end{bmatrix} y \tag{5.11}$$

Double difference observations are correlated and $Q_y$covariance matrix is included to turn the solution into weighted least squares as in equation 5.12 below.

$$\underbrace{\begin{bmatrix} B^T Q_y^{-1} B & B^T Q_y^{-1} A \\ A^T Q_y^{-1} B & A^T Q_y^{-1} A \end{bmatrix}}_{N}\underbrace{\begin{bmatrix} \hat{b} \\ \hat{a} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} B^T Q_y^{-1} y \\ A^T Q_y^{-1} y \end{bmatrix}}_{r} \tag{5.12}$$

As Cholesky Decomposition states that the matrix A which is real symmetric with its all eigenvalues greater than zero can be factored as $A = LL^T$ [13] and we can write matrix N with the assumption above as following:

$$N = LU\ which\ L\ is\ lower, U\ is\ upper\ triangular\ matrices \tag{5.13}$$

By Gaussian elimination forward and backward substitution, the system is solved. [14]

$$L(U * x) = r \tag{5.14}$$

$$U * x = L^{-1} * r \tag{5.15}$$

Then we get the estimates and the covariance or variance-covariance matrix as follows: [12]

$$\begin{bmatrix}\hat{b}\\ \hat{a}\end{bmatrix}, \begin{bmatrix}Q_{\hat{b}} & Q_{\widehat{ba}}\\ Q_{\widehat{ab}} & Q_{\hat{a}}\end{bmatrix} \quad 5.16$$

Now, the solutions are real values; however, we need integer solution.

Integer ambiguity estimation step starts with equation 5.17.

$$\min_{a} \left|\left|\hat{a}-a\right|\right|^2_{Q_{\hat{a}}^{-1}} \quad , a \in Z^n \quad 5.17$$

We need to restrict the solution space into the space of integers by defining an ellipsoidal region in solution space of real values. Region will be centered at $\hat{a}$ and the orientation will be defined by $Q_{\hat{a}}$ ambiguity variance-covariance matrix and positive chosen value χ^2[15].

The minimization problem will lead to a weighted least square problem as follows:

$$(\hat{a}-a)^T Q_{\hat{a}}^{-1}(\hat{a}-a) \quad 5.18$$

If matrix $Q_{\hat{a}}^{-1}$ is diagonal, each component easily can be rounded to nearest integer values. However, due to the fact that the individual ambiguities are correlated we can try having a matrix which its off-diagonal entries are numerically smaller by de-correlating.

$$Q_{\hat{a}} = LDL^T \ (LDL^T \ decomposition) \quad 5.19$$

$L : Lower\ triangular$

$D : Diagonal\ matrix$

From matrix L, a new integer transformation matrix Z is defined so that variance-covariance matrix and float ambiguities are transformed.

$$\hat{z} = Z^T \hat{a} \quad 5.20$$

$$Q_{\hat{z}} = Z^T Q_{\hat{a}} Z \quad 5.21$$

Reaching an integer approximation of the matrix L is aimed by constructing the matrix Z. Moreover, having less correlated ambiguities with transformation will be attained.

The search defined in equation 5.18 can be changed by equation 5.22 and 5.23.

$$\min_{z} \left|\left|\hat{z} - z\right|\right|^2_{Q_{\hat{z}}^{-1}} \quad , z \in Z^n \tag{5.22}$$

$(\hat{z} - z)^T Q_{\hat{z}}^{-1} (\hat{z} - z) \leq \chi^2$ for z integer, χ^2 is chosen constant (5.23)

Finally, with minimizing the equation 5.18 over integer vector a through equations 5.19 to 5.23, integer solution $\check{a}$ is found. Then we substitute $\hat{a}$ in equation 5.11 to find $\hat{b}$.

$$\begin{bmatrix} B^T B & B^T A \\ A^T B & A^T A \end{bmatrix} \begin{bmatrix} \hat{b} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} B^T \\ A^T \end{bmatrix} y \xrightarrow{yields} \begin{bmatrix} \hat{b} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} B^T B & B^T A \\ A^T B & A^T A \end{bmatrix}^{-1} \begin{bmatrix} B^T \\ A^T \end{bmatrix} y \tag{5.24}$$

$$\begin{bmatrix} \hat{b} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} Q_{\hat{b}} & Q_{\widehat{ba}} \\ Q_{\widehat{ba}}{}^T & Q_{\hat{a}} \end{bmatrix} \begin{bmatrix} B^T \\ A^T \end{bmatrix} y \tag{5.25}$$

Applying Gauss Elimination method in equation 5.25 – pre-multiplying second row with $Q_{\widehat{ba}} Q_{\hat{a}}{}^{-1}$ and subtracting it from first row – we get:

$$\begin{bmatrix} \hat{b} - Q_{\widehat{ba}} Q_{\hat{a}}{}^{-1} \hat{a} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} Q_{\hat{b}} - Q_{\widehat{ba}} Q_{\hat{a}}{}^{-1} Q_{\widehat{ba}}{}^T & 0 \\ Q_{\widehat{ba}}{}^T & Q_{\hat{a}} \end{bmatrix} \begin{bmatrix} B^T \\ A^T \end{bmatrix} y \tag{5.26}$$

$$\hat{b} - Q_{\widehat{ba}} Q_{\hat{a}}{}^{-1} \hat{a} = (Q_{\hat{b}} - Q_{\widehat{ba}} Q_{\hat{a}}{}^{-1} Q_{\widehat{ba}}{}^T) B^T y \tag{5.27}$$

The right side of the equation 5.27 is constant and known from the results of the equations 5.10 and 5.16. Therefore, we know the value of left side of the equation.

Changing $\hat{a} \rightarrow \check{a}$ and $\hat{b} \rightarrow \check{b}$:

$$\hat{b} - Q_{\widehat{ba}} Q_{\hat{a}}{}^{-1} \hat{a} = \check{b} - Q_{\widehat{ba}} Q_{\hat{a}}{}^{-1} \check{a} \tag{5.29}$$

$$\check{b} = \hat{b} - Q_{\widehat{ba}} Q_{\hat{a}}{}^{-1} (\hat{a} - \check{a}) \tag{5.30}$$

$\hat{b}$: $baseline\ solution\ (float\ ambiguites)$

$\check{b}$: $baseline\ solution\ (fixed\ integer\ ambiguity)$

5.2.2 Algorithmic Steps

The LAMBDA method detailed in section 5.2.1, consists of four main steps, namely, constructing observation equations, defining estimation criteria, obtaining float solution, and estimating integer ambiguity and baseline solution. The algorithm implementation procedure is summarized as following: [12]

- Constructing Observation Equations:
  For each same epoch, construct double-difference equations using equations 5.3 to 5.9 and put into the linearized equations form in equation 5.10.
- Defining Estimation Criteria:
  Set normal equations and scale by variance-covariance weight matrix in equations 5.11 and 5.12
- Obtaining Float Solution in Ambiguity
  Apply Cholesky Decomposition to the normal matrix and by forward/ backward substitution obtain float ambiguities using equations 5.13 to 5.15.
- Estimation of Integer Ambiguity and Baseline Solution
  Through the procedure which is de-correlating the ambiguities and searching integers defined in equations 5.18 to 5.30, find integer ambiguity values and obtain distance between base and rover.

5.3 Comparison of Code and Carrier Based Baseline Computations

Precise positioning is achievable with combination of carrier phase data and code based pseudorange data as explained theoretically in detail in section 5.2.

Figure 70 shows the test setup with base and rover receivers having a distance of 20 cm between them.

Figure 71 shows the field test setup and implemented algorithms in block diagram.

Figure 70 - Test Setup (20cm Baseline Distance)

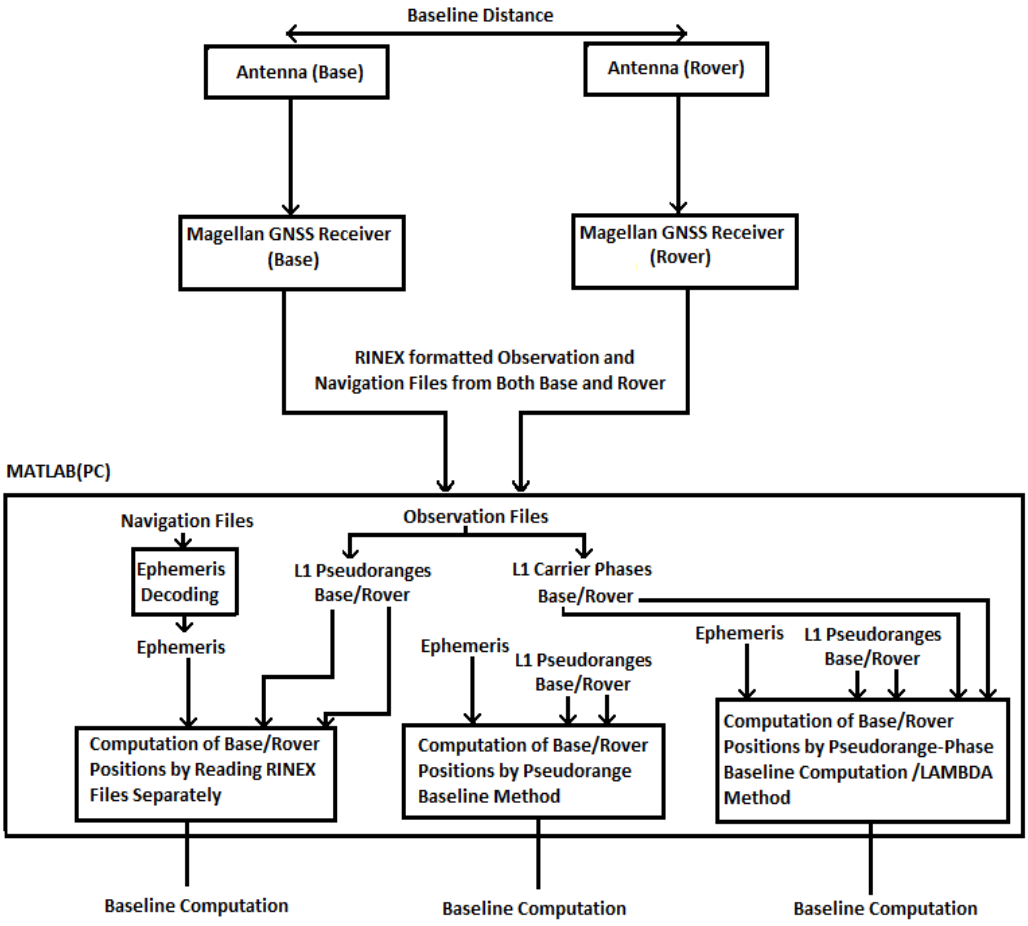

Figure 71 – Field Test Setup and Implemented Algorithms in Block Diagram

In this section, as can be seen in Figure 71, comparison of baseline distance accuracies resulting from three different algorithms, namely, computation from base and rover receivers separately, pseudorange baseline method and pseudoranges-carrier phase baseline method is made.

As seen in Figure 71, firstly, the code and carrier based data using RINEX Observation and Navigation files, both receivers' positions in ECEF frame are computed separately. Then, for each epoch, C/A pseudorange observations of base and rover receivers are used and baseline is computed. Finally, carrier phase observations are included and combined with pseudorange data to compute baseline. Performance comparison of code and carrier based positioning accuracies is shared throughout this section.

RINEX (Receiver Independent Exchange) formatted data files obtained from Magellan receivers are fed into Magellan post processing utility analysis tool. With 0.2 cm error, the distance between base and rover receivers is computed. However, as being different from Magellan system, using only GPS L1 signal ambiguity resolution method is implemented in MATLAB. Throughout this section, the comparison between code based and carrier based solutions are also included.

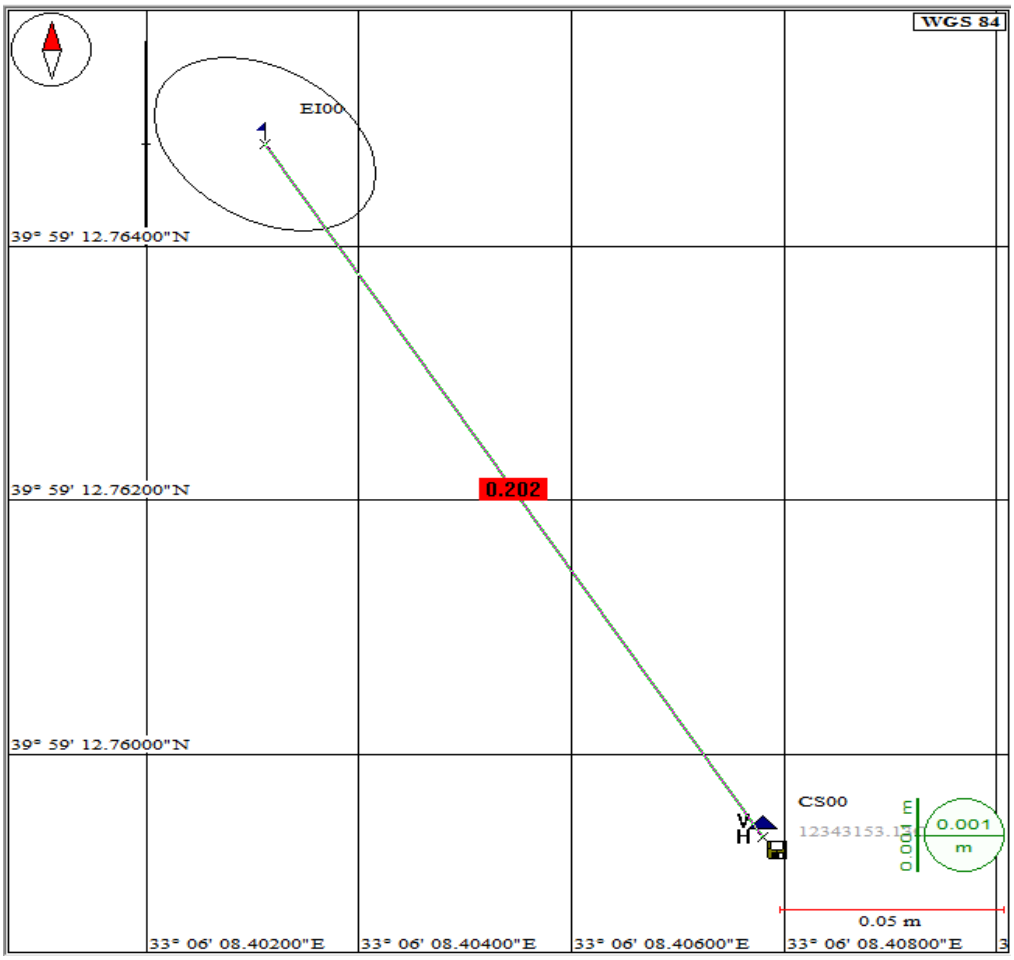

Figure 72 - Magellan GNSS Receivers Position Solution - Post Processing Utility Analysis – 20cm Baseline Distance

In MATLAB, using RINEX Observation and Navigation files, both receivers' positions in ECEF frame are computed separately. In computations, only pseudoranges and obtained ephemeris are used. The computation is repeated over 150 epochs. Each position is the result of an iterative least-squares procedure.

Figure 73 shows the separately computed position solutions of base and rover receivers in each ECEF axis on the same graphs.

Figure 74 shows the difference in coordinate values between base and rover receivers in each ECEF axis.

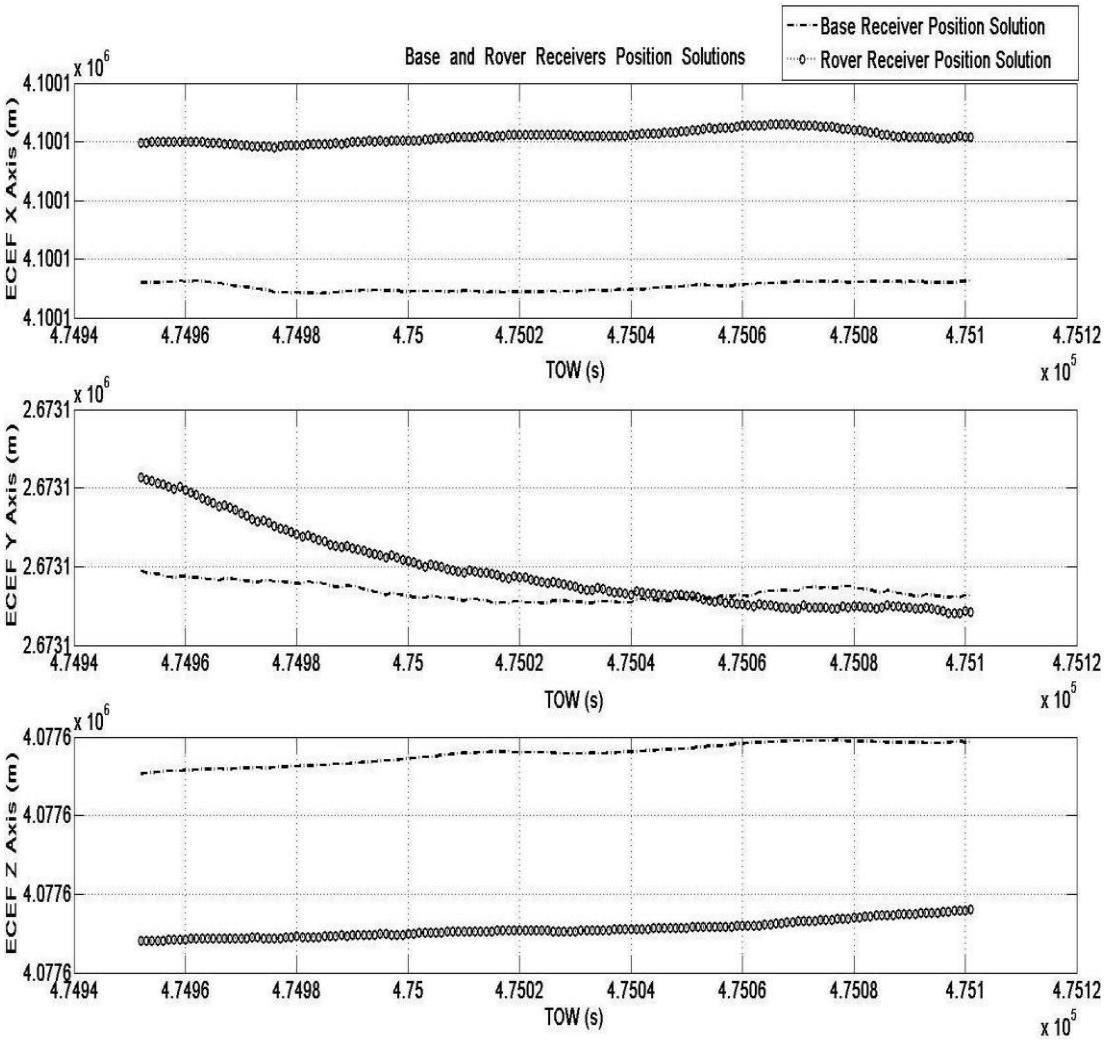

Figure 73 - Base and Rover Receiver Coordinate Values – (Reading Separate RINEX Obs. and Nav. Files

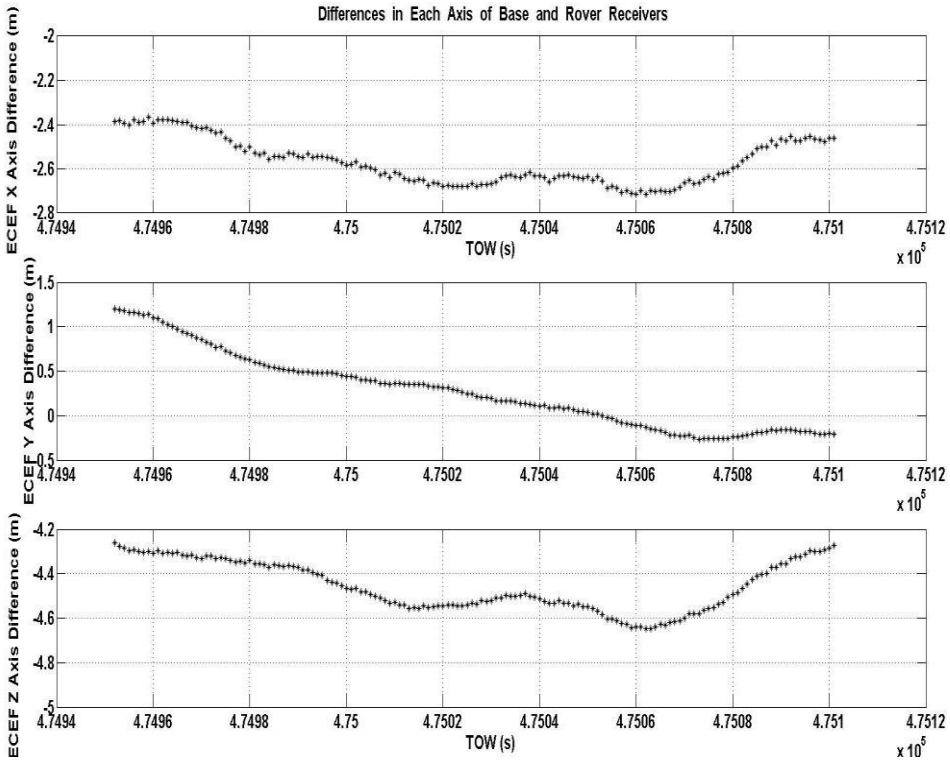

Figure 74 - Difference in Coordinate Values Between Base and Rover Receivers - (Reading Separate RINEX Obs. And Nav. Files)

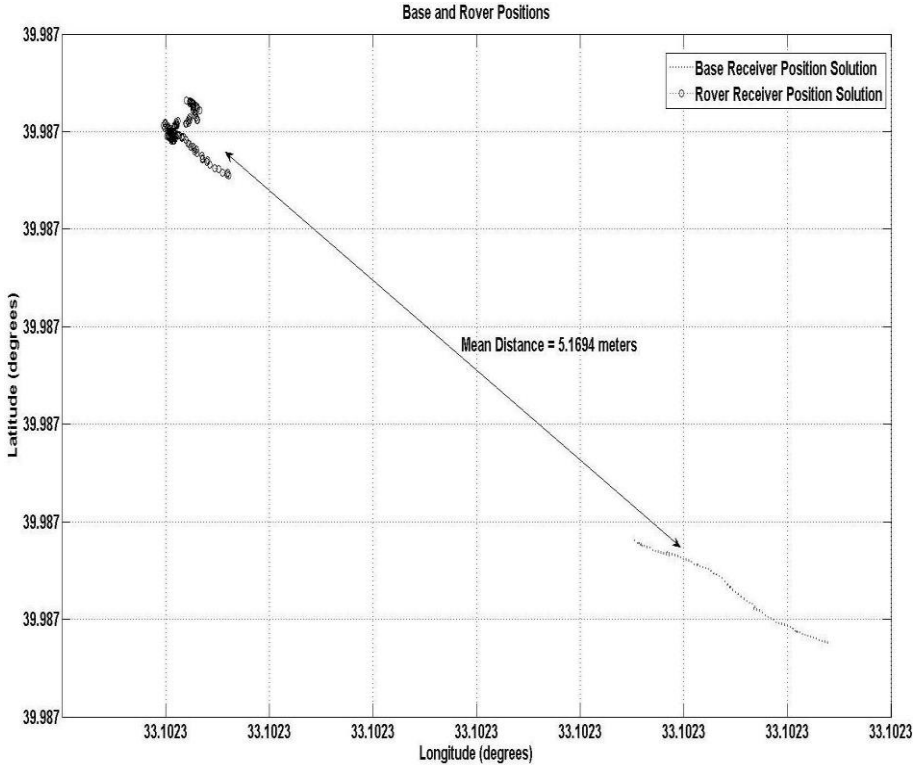

Figure 75 - Distance Between Base and Rover – (Reading Separate RINEX Obs. And Nav. Files)

As seen in Figure 75, although the distance between base and rover is 20 cm, we reached a value that shows the distance between base and rover is as 5.1694 meters. As expected, this result is consistent with the obtained positioning performances of implemented code phase-based GPS, GLONASS, and combined GPS/GLONASS algorithms.

Secondly, from both base and rover receivers, simultaneous pseudoranges are used. Baseline is estimated from pseudoranges alone. Carrier phase data is not used. Difference in each axis is reduced compared to separate position solution computation above.

Figure 76 shows the computed position solutions by processing simultaneous pseudoranges of base and rover receivers in each ECEF axis on the same graphs.

Figure 77 shows the difference in coordinate values between base and rover receivers in each ECEF axis in pseudorange baseline computation method.

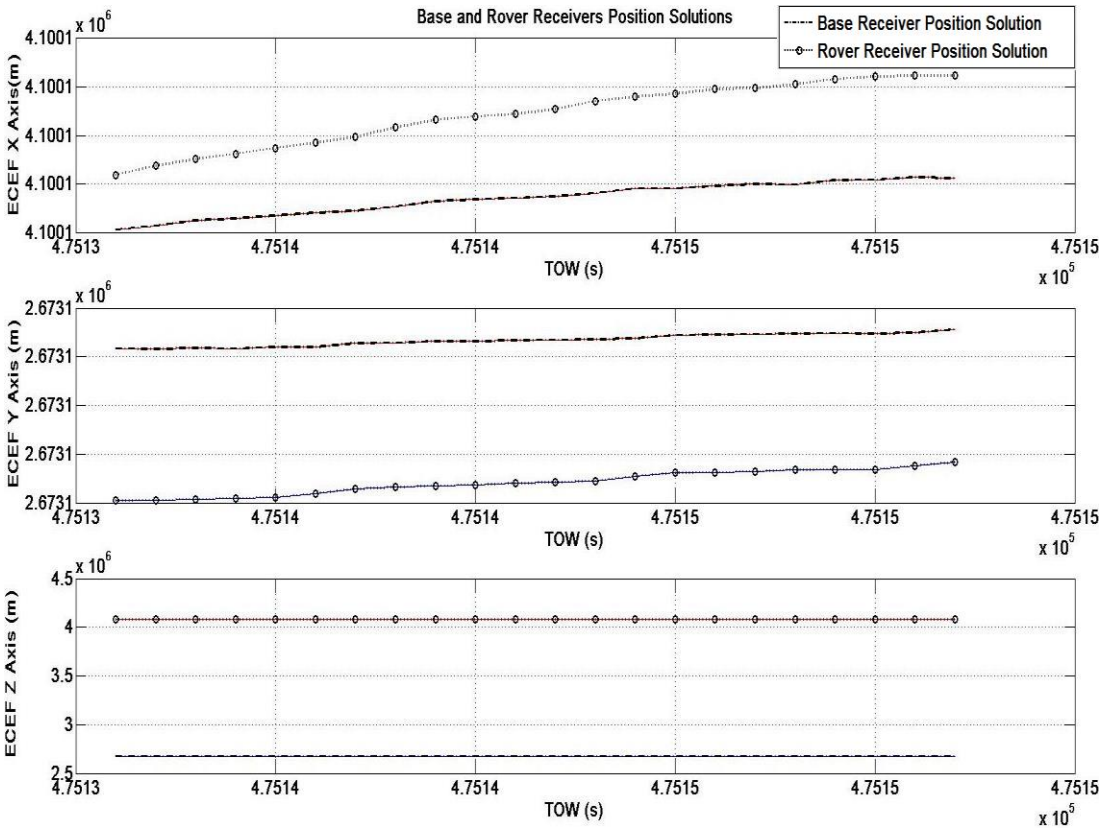

Figure 76 - Base and Rover Receiver Coordinate Values – (Pseudorange Baseline Computation)

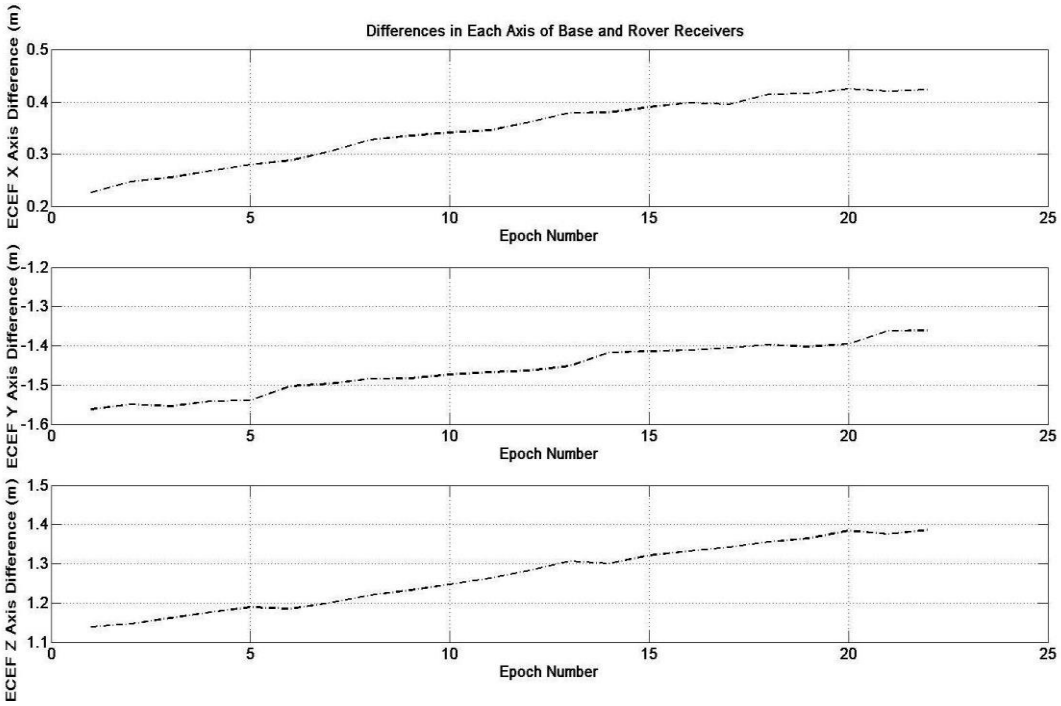

Figure 77 - Difference in Coordinate Values Between Base and Rover Receivers - (Pseudorange Baseline Computation)

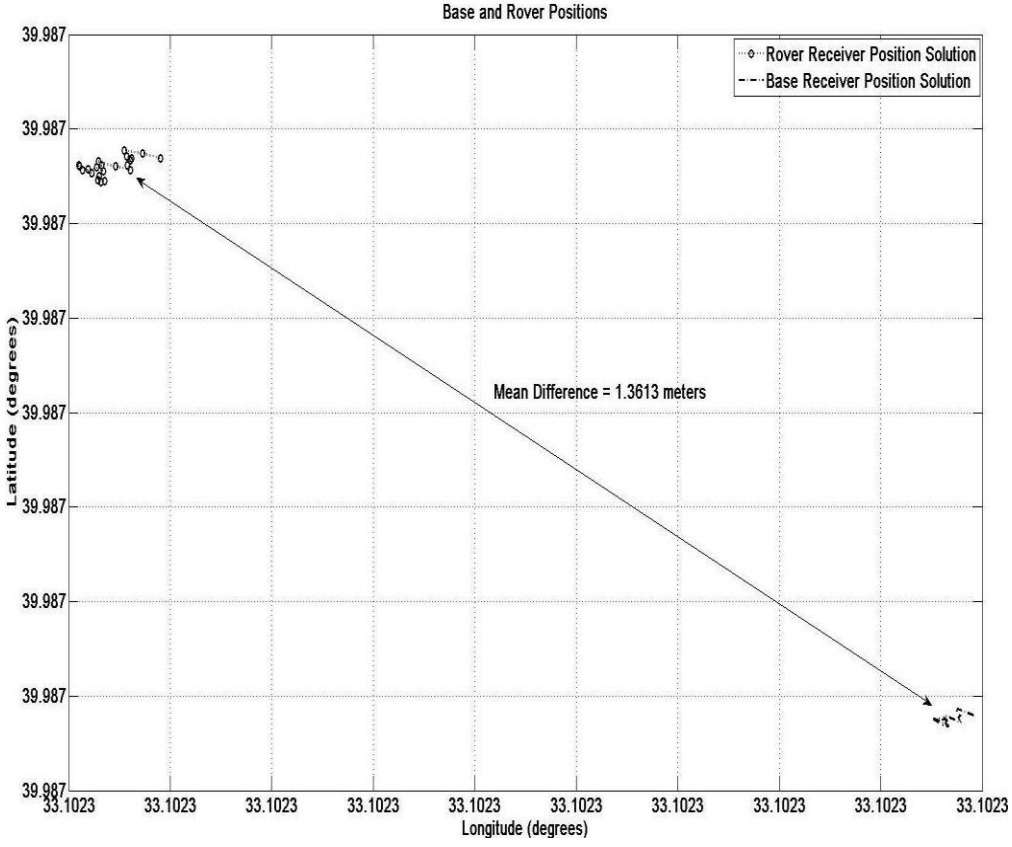

Figure 78 - Distance Between Base and Rover – (Pseudorange Baseline Computation)

As seen in Figure 78, using simultaneous pseudoranges in implemented baseline estimation method, we find the distance between base and rover receivers as 1.3613 meters, although the distance between base and rover is still 20 cm. We have still errors on the order of meters.

Finally, using pseudorange-phase based LAMBDA method explained in section 5.2; baseline between base and rover receivers is computed.

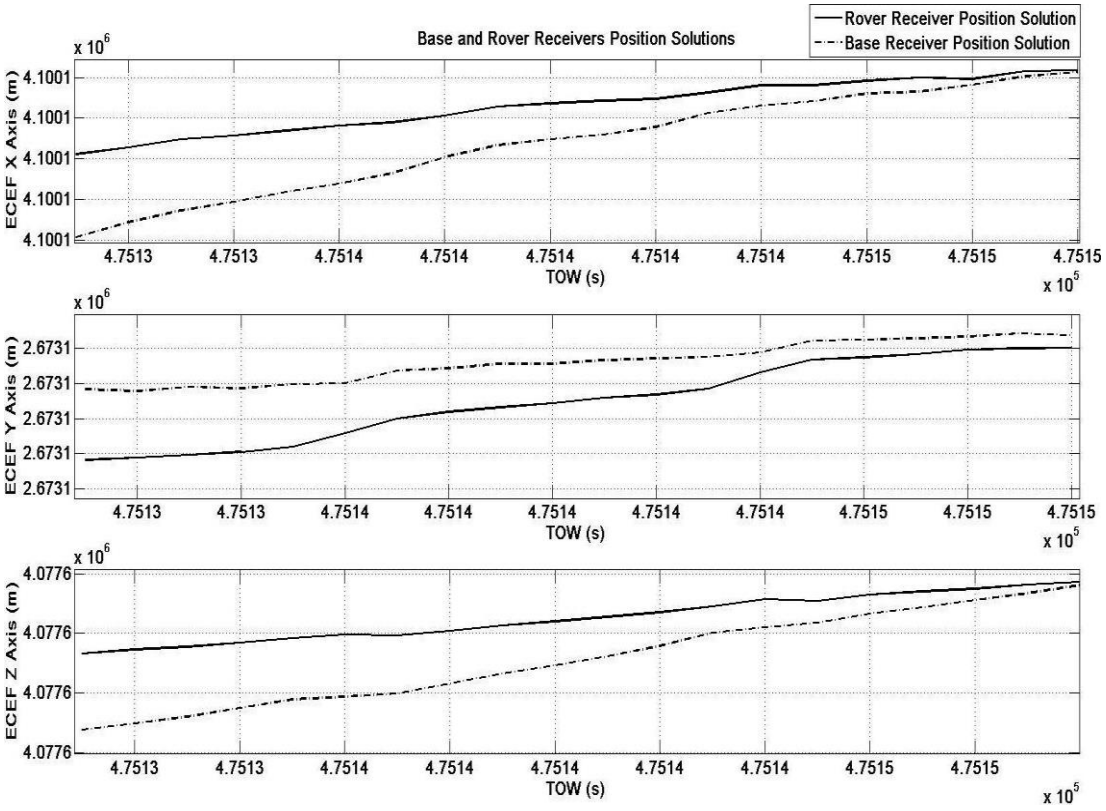

Figure 79 - Base and Rover Receiver Coordinate Values - (Pseudorange-Phase Baseline Computation / LAMBDA Method)

Figure 79 shows the computed position solutions of base and rover receivers by LAMBDA method in each ECEF axis on the same graphs.

In Figure 80, the difference in coordinate values between base and rover receivers in each ECEF axis in LAMBDA method is shown.

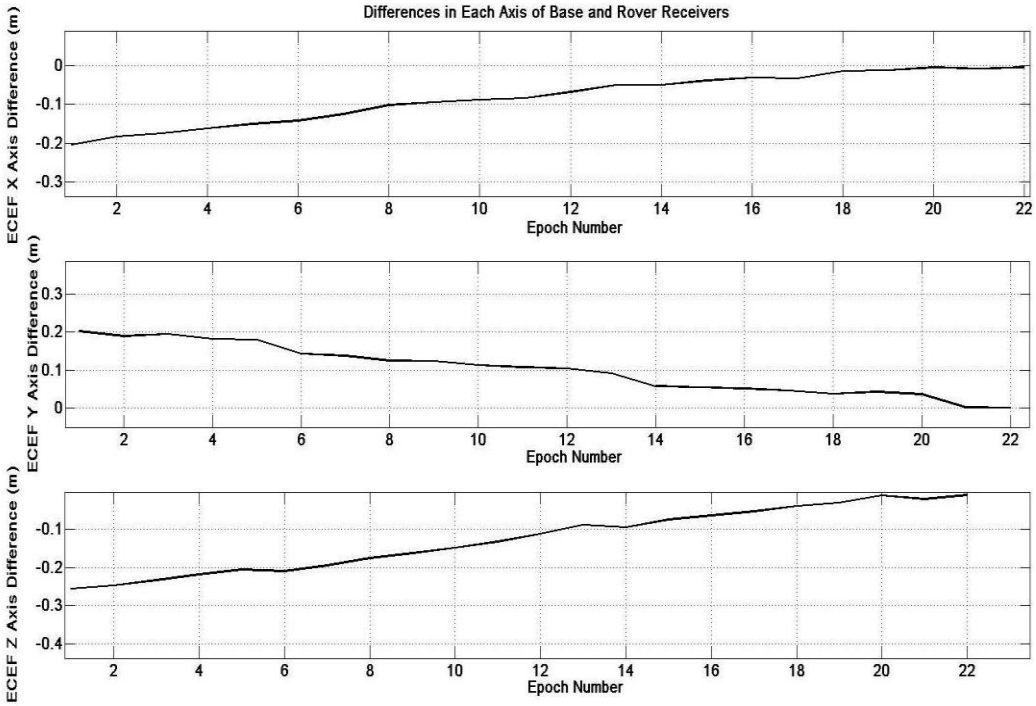

Figure 80 - Difference in Coordinate Values Between Base and Rover Receivers- (Pseudorange-Phase Baseline Computation / LAMBDA Method)

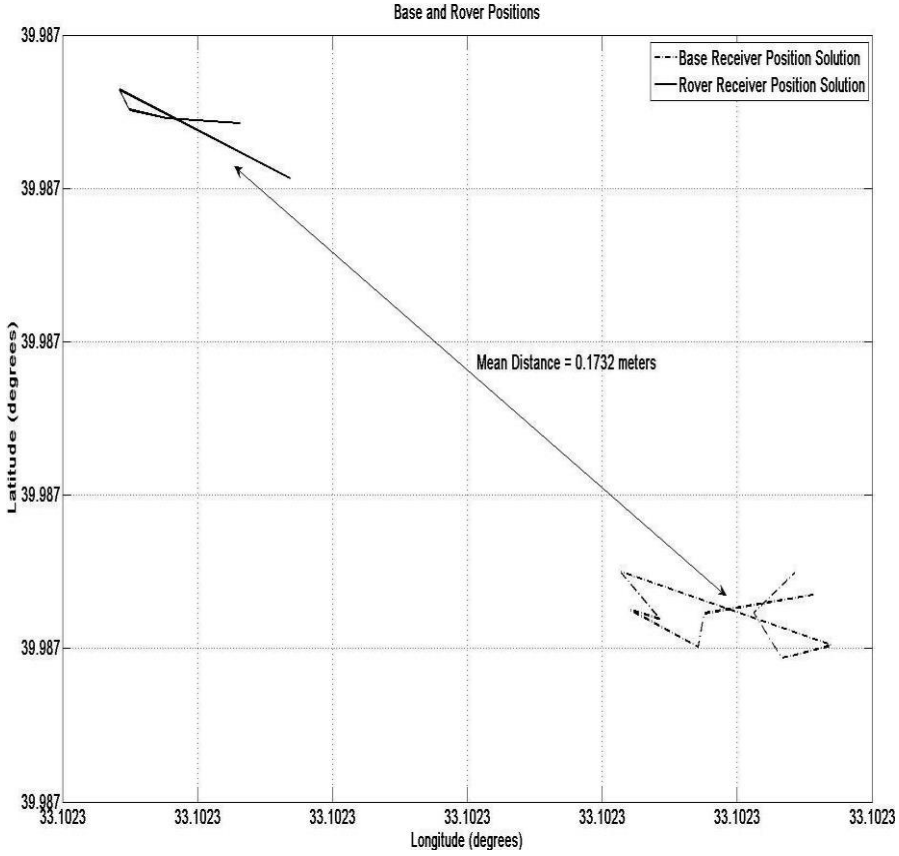

Figure 81 - Distance Between Base and Rover – (Pseudorange-Phase Baseline Computation / LAMBDA Method)

As seen in Figure 81, the difference of the variation of the coordinate values is 0.1732 meters with 2.68 cm error in distance. Therefore, it is concluded that implemented LAMBDA method works well in baseline distance computation

5.4 Field Test Results

Field tests are conducted for different distances between base and rover receivers in order to perceive the performance of positioning algorithm for short distances and the effects of the variation of errors with distance. Distance between base and rover receivers which is shown in Figure 70 and Figure 71 is adjusted to the values of 1m, 2m, 5m, 10m, 15m, 20m, 25m, 30m, 35m, 40m, 45m, and 50m to observe the performance of the implemented LAMBDA method.

Graphs of baselines between base and rover receivers and the comparison of computed error values for each distance are shared. Therefore, the idea which claims that many pseudorange and carrier phase errors are same at two nearby locations is verified with conducted tests.

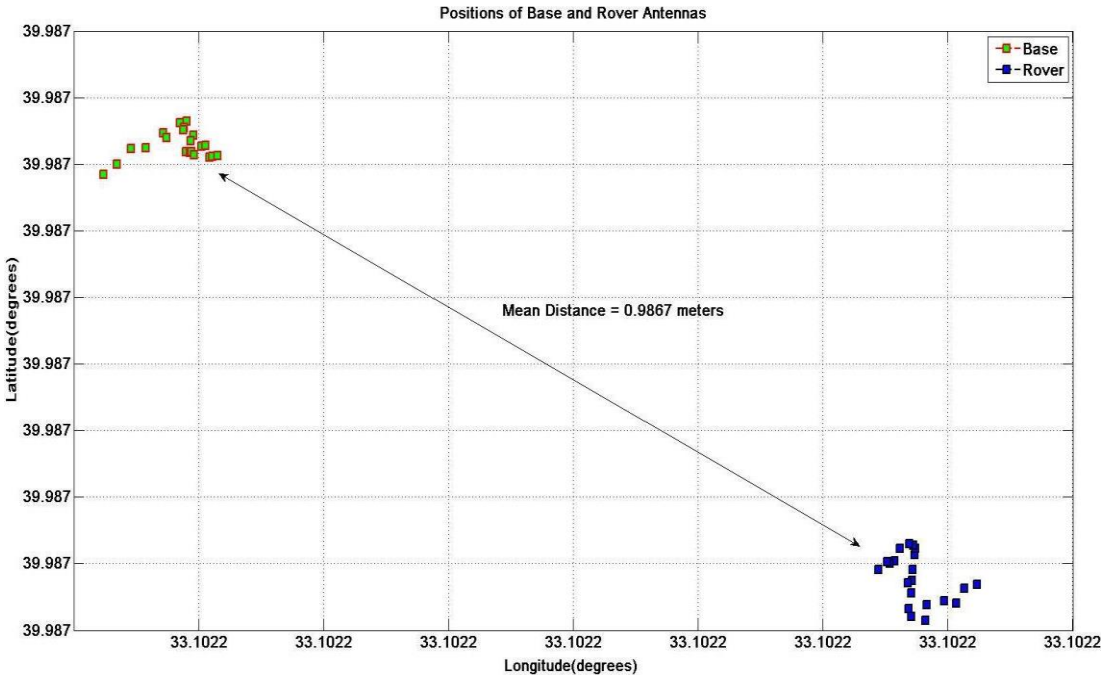

Figure 82 - Positions of Base and Rover Antennas - 1 meter Distance Test

As seen in Figure 82, the mean distance between base and rover is 0.9867 m with 1.33 cm error in 1 meter distance test.

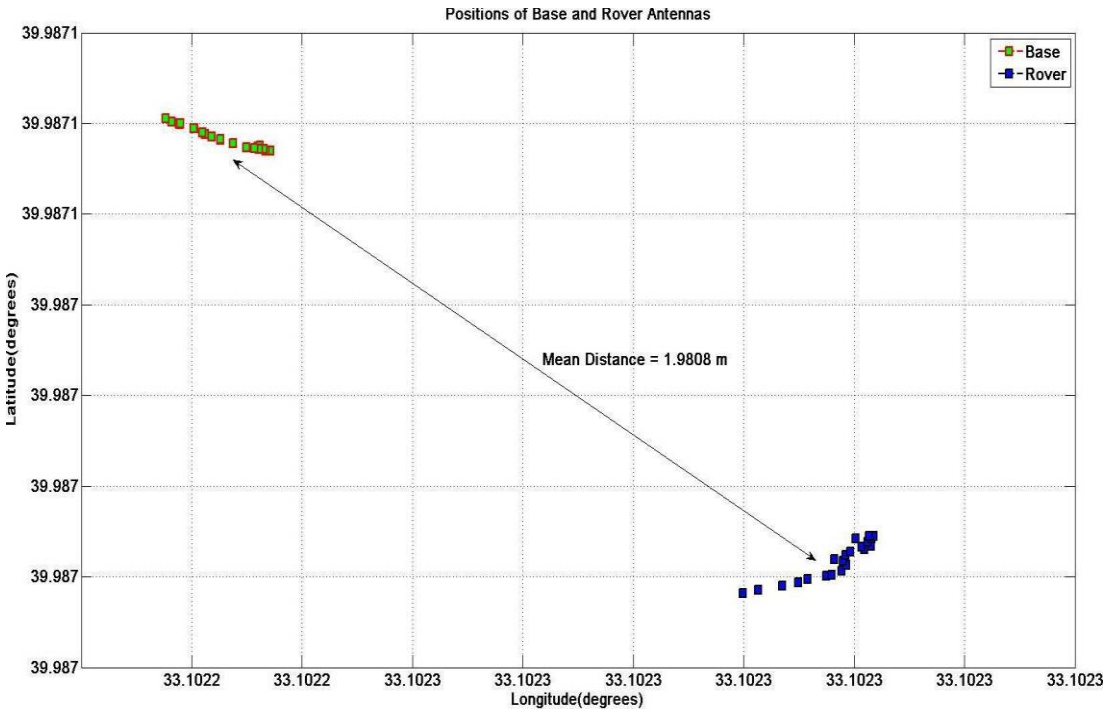

Figure 83 - Positions of Base and Rover Antennas - 2 meters Distance Test

The mean distance between base and rover is 1.9808 m with 1.92 cm error in 2 meters distance test, as can be seen in Figure 83 above.

Figure 84 – 5meters Distance Test Setup

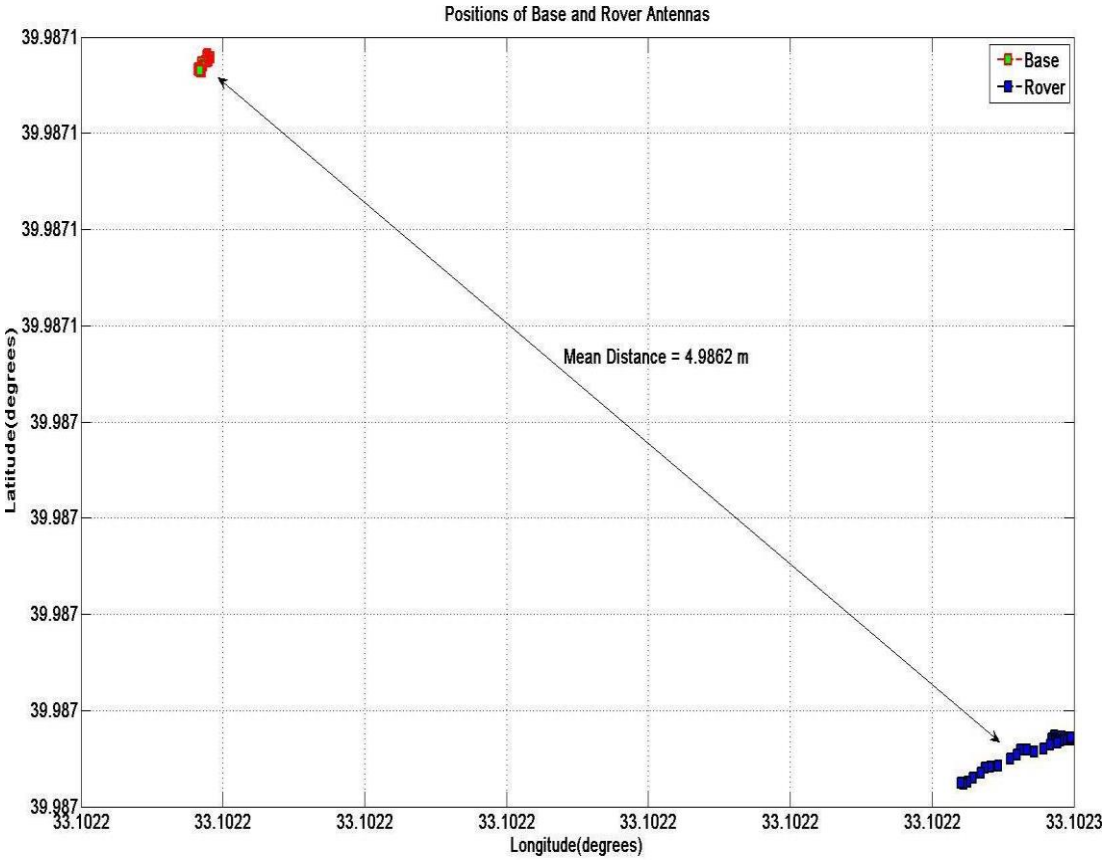

Figure 85 - Positions of Base and Rover Antennas - 5 meters Distance Test

In Figure 85, the mean distance between base and rover is 4.9862 m with 1.38 cm error.

Figure 86 - 10 meters Distance Test Setup

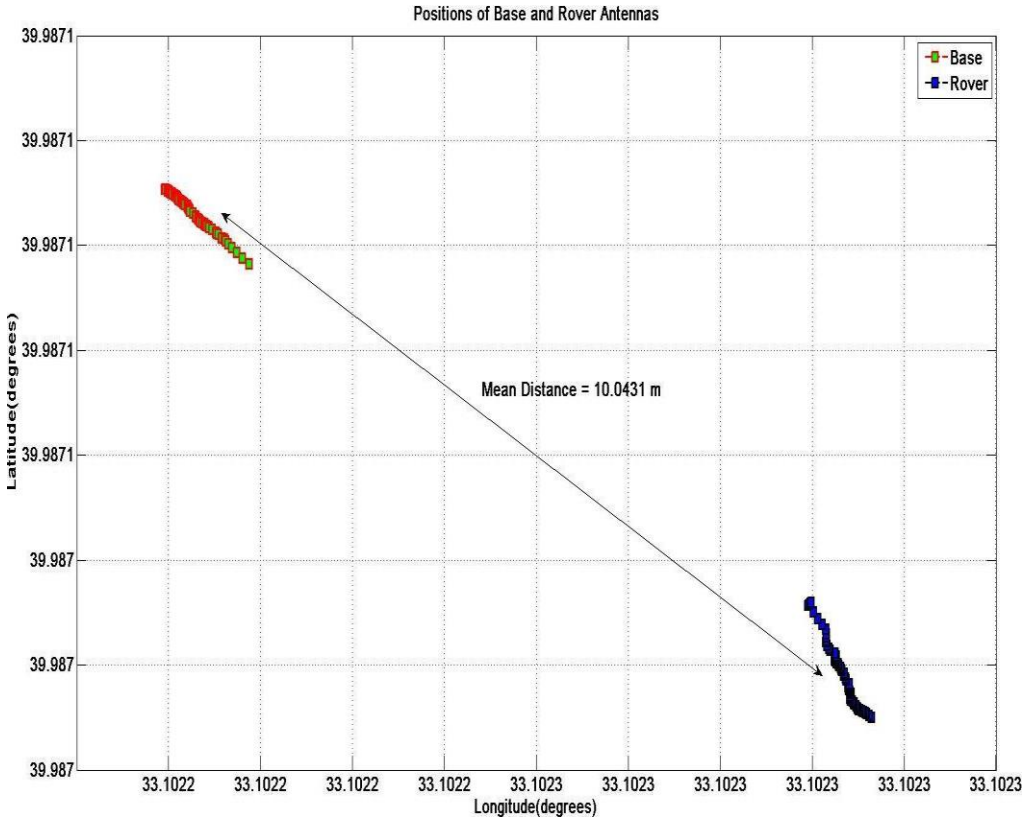

Figure 87 - Positions of Base and Rover Antennas - 10 meters Distance Test

Figure 86 shows the test setup for 10 meters distance between base and rover. The mean distance between base and rover is 10.0431 m with 4.31 cm error, as seen in Figure 87.

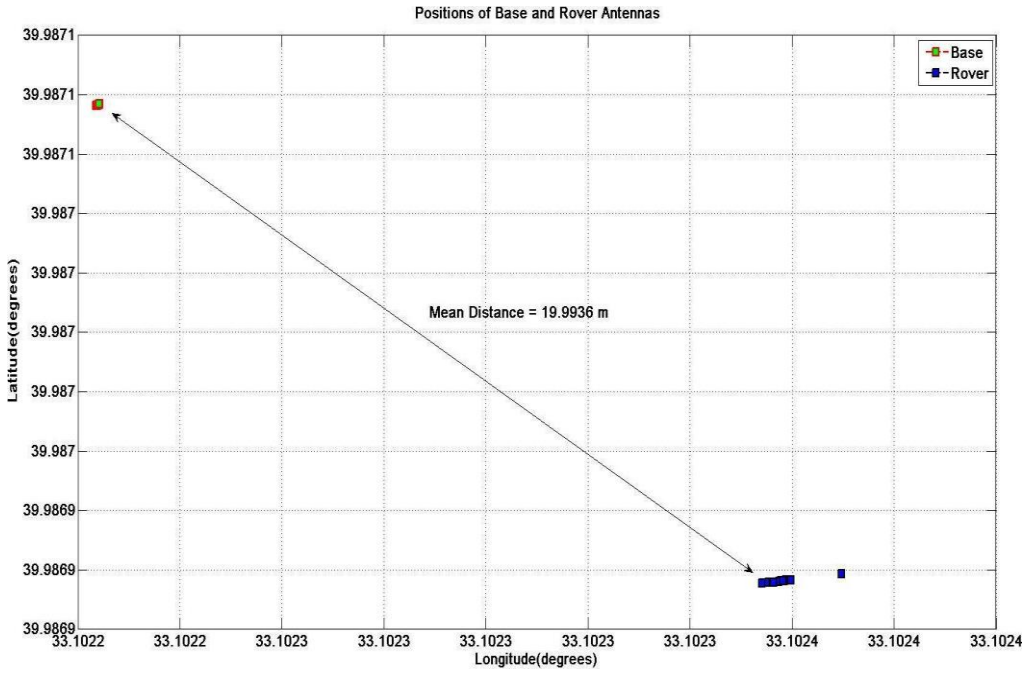

Figure 88 - Positions of Base and Rover Antennas - 20 meters Distance Test

The mean distance between base and rover is 19.9936 m with 0.64 cm error, as can be seen in Figure 88.

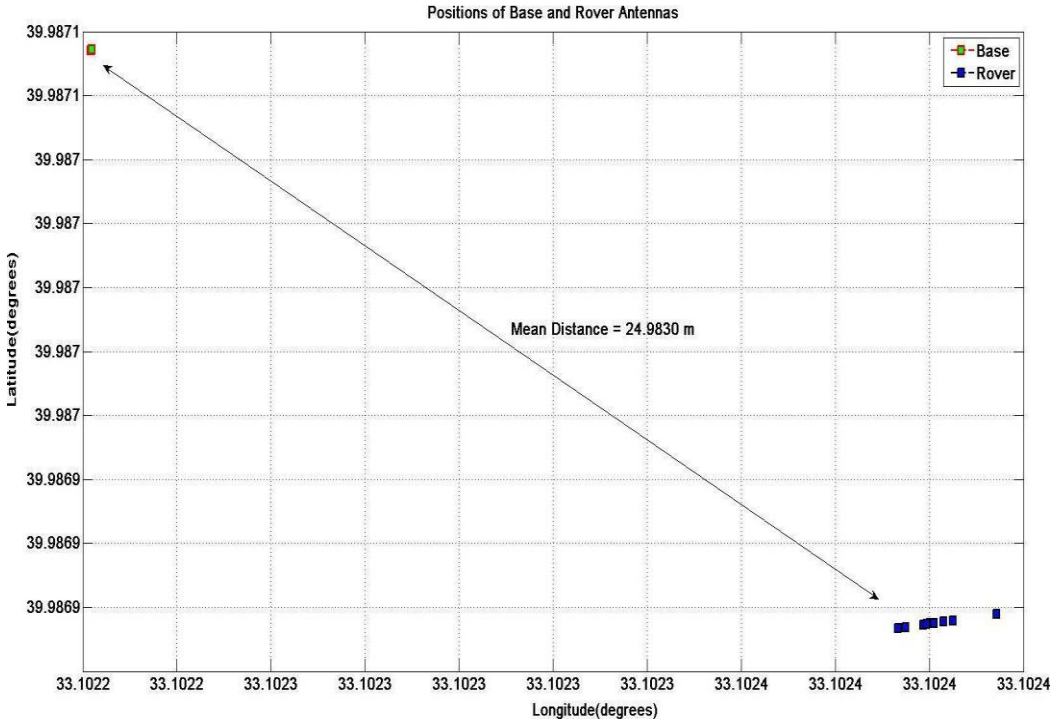

Figure 89 - Positions of Base and Rover Antennas - 25 meters Distance Test

From Figure 89, it is observed that the mean distance between base and rover is 24.9830 m with 1.70 cm error in 25 meters distance test.

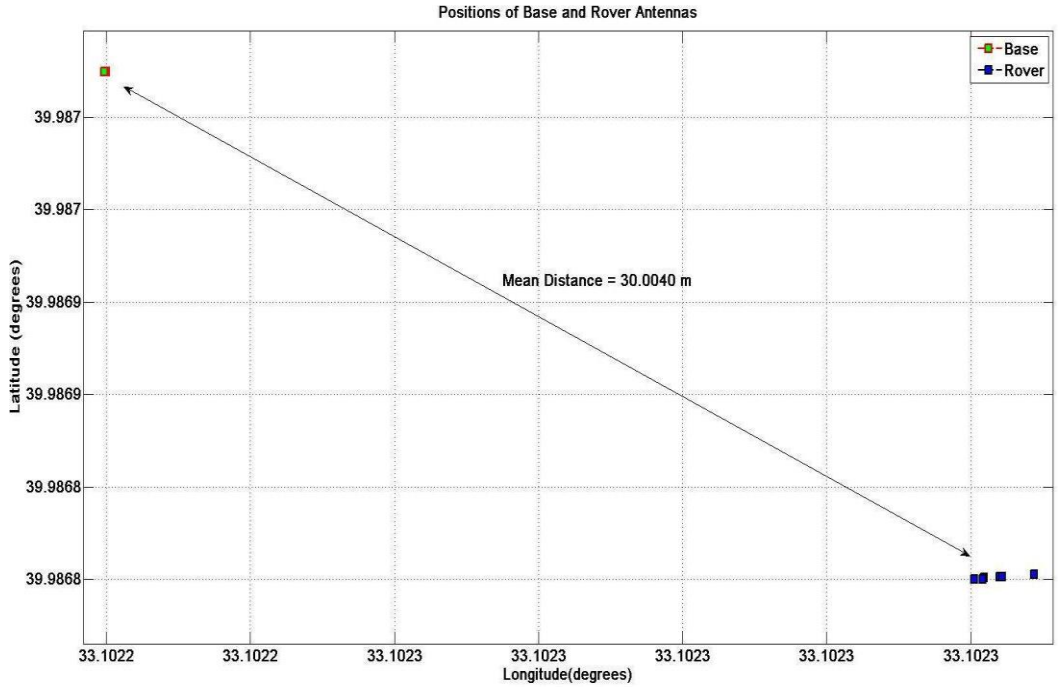

Figure 90 - Positions of Base and Rover Antennas - 30 meters Distance Test

Figure 90 shows the test results of 30 meters distance test between base and rover. The mean distance between base and rover is computed as 30.0040 m with 0.40 cm error.

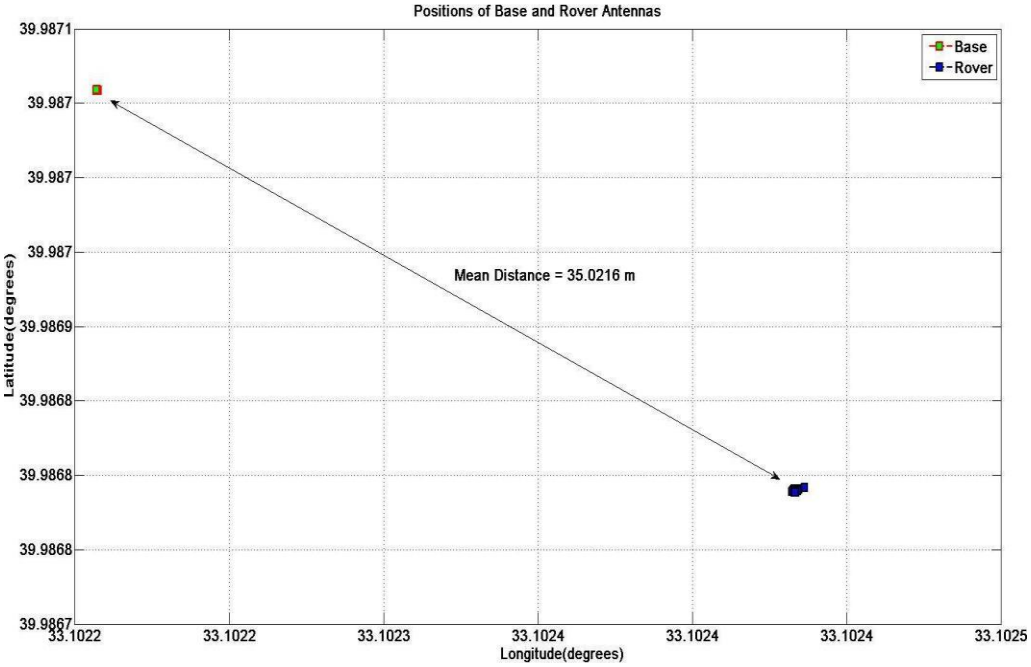

Figure 91 - Positions of Base and Rover Antennas - 35 meters Distance Test

The mean distance between base and rover is 35.0216 m with 2.16 cm error as seen in Figure 91.

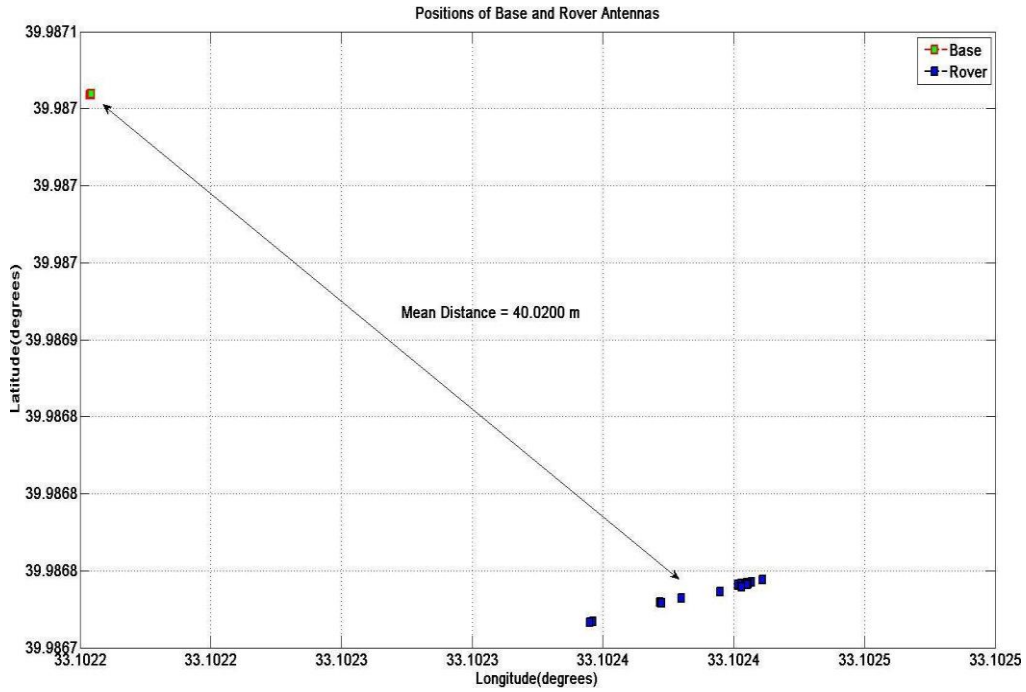

Figure 92 - Positions of Base and Rover Antennas - 40 meters Distance Test

In Figure 92, it is shown that the mean distance between base and rover is 40.0200 m with 2.00 cm error.

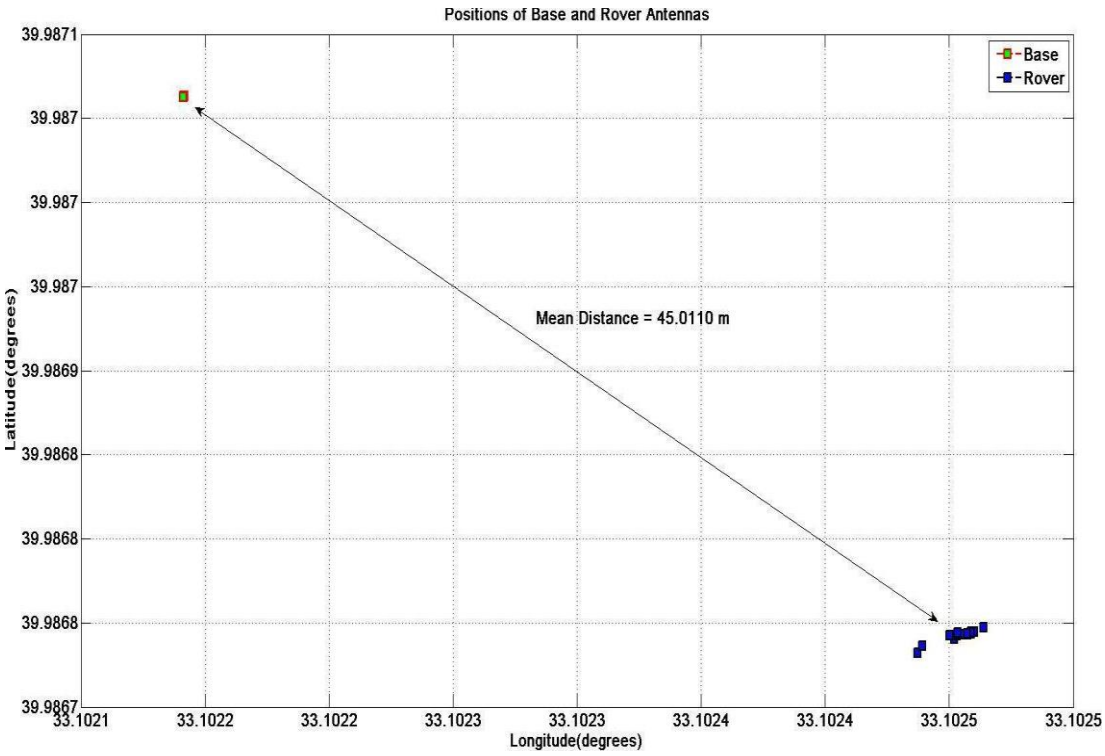

Figure 93 - Positions of Base and Rover Antennas - 45 meters Distance Test

The mean distance between base and rover is 45.0110 m with 1.10 cm error as can be seen in Figure 93.

Figure 94 - 50 meters Distance Test Setup

To sum up, computed errors for each field tests given above are presented in Figure 95 below.

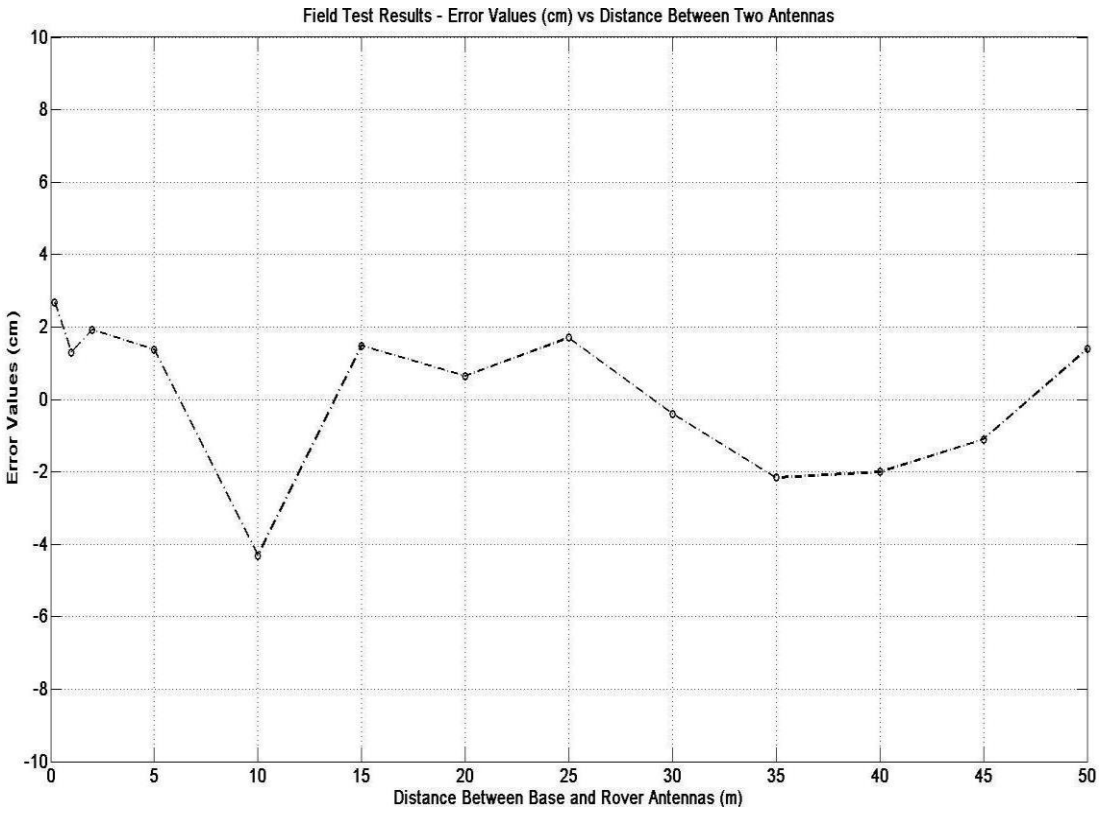

Figure 95 - PPK Computed Error Values for Each Distances

Change in error values with distances between base and rover is plotted in Figure 95. From the conducted field test results, it is concluded that for distances up to 50 meters, the pseudorange and carrier phase errors are same at two nearby locations.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this chapter, conclusions of this thesis are presented and also possible future works are mentioned.

Conclusion

In this thesis, positioning performances of GPS, GLONASS and combined GPS/GLONASS systems are analyzed in the light of implemented algorithms at MATLAB. In order to improve the positioning performance further, two receiver carrier-phase based positioning algorithms are implemented. All implemented algorithms are presented in detail and simulation results are provided. In addition to simulations, field tests are also conducted on carrier based algorithms' tests.

Initially, the fundamental information about GPS and GLONASS systems are presented in terms of their L1 band signal characteristics, transmitted satellite data and signal acquisition/tracking loop architectures. MATLAB algorithms' outputs in signal acquisition/tracking and navigation data extraction level are included.

Some improvements on implemented algorithms are needed when navigation outputs of implemented algorithms are considered. One of them is addition of tracking lock status indicator to the algorithms. Exposed to high dynamic maneuvers, GPS tracking loops can lose lock to signals and thus wrong or no solution gaps in flights can occur. These no solution gaps during flight causes the GPS solution to drift from true values. It is observed that addition of tracking lock status indicator-check in signal

tracking loops is improved accuracy under dynamic scenario by preventing drifting pseudorange values from expected true values.

After verifying GPS and GLONASS algorithms separately, GPS/GLONASS combined solution is created and positioning performances are compared in terms of RMSE of the navigation outputs.

Finally in order to improve the positioning accuracy of implemented code-phase algorithms, carrier-phase based positioning algorithms are applied. Then, satisfactory navigation accuracy is obtained to an accurately known position is reached in two receiver configuration.

Future Work

Expanding the software receiver to support new signals of GPS and GLONASS and also other satellite systems' signals (Galileo, Beidoo, etc.) can be achievable.

Further analysis can be done on signal tracking performance by making adaptive tracking algorithms with different PLL/FLL/DLL bandwidths, orders and structures.

Also adding GLONASS carrier based algorithms as being compatible to working with GPS PPK algorithms, precise positioning methods can be improved further.

In high dynamic applications, GNSS receivers encounter much more Doppler frequency shift and it will not be able to track signal anymore. Although, FLL assisted PLL and carrier aided DLL loop filters are implemented so as to make the tracking algorithms insensitive to acceleration and jerk stress under high dynamic applications, with unaided carrier and code tracking algorithms it is really difficult to keep on tracking continuously. Therefore, an approach which consists of an INS/IMU aid to code and carrier tracking loops directly to inform GPS about dynamics of the platform can be improved.

REFERENCES

[1] Official U.S. Government information about the Global Positioning System (GPS) and related topics, http://www.gps.gov/, Last Access Date: 04.05.2014

[2] Federal Space Agency Information – Analytical Centre, http://glonass-iac.ru/en/, Last Access Date: 04.05.2014

[3] Hofmann-Wellenhof, B., Lichtenegger, H., &Wasle, E.,GNSS-Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more. Wien: Springer., (2008)

[4] Borre, K., Software-defined GPS and Galileo receiver: A single-frequency approach. Boston: Birkhhäuser. (2007).

[5] Kaplan, Elliott D., and C Hegarty. , Understanding GPS: Principles and Applications. 2nd ed. Boston: Artech House(2006)

[6] GPS Space Segment/Navigation User Interfaces, ICD-GPS-200C, GPS Joint Program Office (1997)

[7] Global Navigation Satellite System GLONASS Interface Control Document, Navigational Radio-signal in Bands L1, L2 (Edition 5.1, 2008 Moscow)

[8]Misra, P and Enge, P. Global Positioning System: Signals, Measurements, and Performance, 2nd Edition, Lincoln, MA: Ganga-Jamuna Press (2006)

[9] Tsui, J-B. - Y. Basic GPS Concept, in Fundamentals of Global Positioning System Receivers: A Software Approach, John Wiley & Sons Inc., New York USA (2000)

[10] Groves, Paul D., Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Artech House (2008)

[11] Strang, G. and Borre, K., Linear Algebra, Geodesy and GPS, Wellesley Cambridge Press USA (1997)

[12] Jonge, P., and Tiberius, C., The LAMBDA method for integer ambiguity estimation: implementation aspects, Delft Geodetic Computing Centre, August 1996.

[13] Xing, H., Solving Least Square Problem Using LU and Cholesky Decomposition, MATH547 Course Notes (Spring 2013), The University of North Carolina

[14] Golub, G.H. and C.F. Van Loan, Matrix Computations, North Oxford Academic, The Johns Hopkins University Press, Baltimore, Maryland, USA 1983

[15] P.J.G., Teunissen, The Least-Squares Ambiguity Decorrelation Adjustment: A Method for Fast GPS Ambiguity Estimation, Delft Geodetic Computing Centre, Netherlands 1994

[16] Gunawardena,S., GNSS Receiver Design Tutorial, ION GNSS 2011 Conference

[17] Proakis, J. G., Digital Communications. Boston: McGraw-Hill (2001)

[18] Abbasiannik, S. Multichannel Dual Frequency GLONAS Software Receiver in Combination with GPS L1 C/A, Department of Geomatics Engineering University of Calgary (April, 2009)

[19] Zhang, W., Ghogho, M. and Aguado L.E., Extension of GPS Broadcast Ephemeris to Determine Satellite Velocity and Acceleration,

[20] Peng, S. and Ledvina, M., A Real Time Software Receiver for the GLONASS L1 Signal, ION GNSS 21st Technical Meeting of the Satellite Division (September, 2008)

[21] Goral, W. and Skorupa B., Determination of Intermediate Orbit and Position of GLONASS Satellites Based on the Generalized Problem of Two Fixed Centers, ActaGeodynGeomater., Vol. 9, No.3 (167), (p.283-290, June 2012)

[22] Gaglione, S, Angrisano, A., Pugliano, G., Robustelli, U. Vultaggio, M., A Stochastic Sigma Model for GLONASS Satellite Pseudorange, Parthenope University of Naples, Italy

[23] '*GLONASS Overview*', Paper Retrieved from http://webone.novatel.ca/assets/Documents/Papers/GLONASSOverview.pdf on April 30, 2012

[24] Nik, S. Abbasian and Petovello, M.G., Multichannel Dual Frequency GLONASS Software Receiver, ION GNSS 2008, Session B4, Savannah, GA, (September, 2008)

[25] Kline, P.A, "Atomic Clock Augmentation For Receiver Using the Global Positioning System" Electrical Engineering Department of Virginia Polytechnic Institute and State University, Blacksburg, Virginia, (February, 1997)

[26] Rao, M., Presti, L.L., Fantino, M. and Garbo, G., A Software Receiver Phase Lock Loop Analysis and Design to Implement Adaptive Phase Tracking Using a Finite Impulse Response Filter, 13th IAIN World Congress, Sweden, (October,2009)

[27] Roncaglio, P.A. and Garcia, J.G., High Dynamics and False Lock Resistant GNSS Carrier Tracking Loops, 20th ION GNSS Conference, (September, 2007)

[28] Chung, B.Y., Chien,C., Samueli, H., Jain, R., Performance Analysis of an All-Digital BPSK Direct-Sequence Spread-Spectrum IF Receiver Architecture, IEEE Journal on Selected Areas in Communications, Vol.11, No.7, (September, 1993)

[29] Ferrão, P.F.F.N., Positioning with Combined GPS and GLONASS Observations, Master Thesis in Aerospace Engineering Tecnico Lisboa, (May, 2013)

[30] European Space Agency (ESA) – Navipedia - Atomic Time, http://www.navipedia.net/index.php/Atomic_Time, Last Access Date: 19.07.2014

[31] European Space Agency (ESA) – Navipedia – Transformations between Time Systems, http://www.navipedia.net/index.php?title=Transformations_between_Time_Systems&oldid=12208 , Last Access Date: 19.07.2014

[32] Lewandowski, W. and Arias, E.F. GNSS Times and UTC, Bureau International des Poids et Mesures, Sevres, France, (July, 2011)

[33] BIPM (Bureau International des Poids et Mesures) International Atomic Time, http://www.bipm.org/en/scientific/tai/tai.html , Last Access Date: 19.07.2014

[34] Cheng, Chao-heh, Calculations For Postioning With the Global Navigation Satellite System, A Thesis Presented to College of Engineering and Technology Ohio University, (August, 1998)

[35] Hall, T., Burke, B., Pratt, M., and Misra, P. Comparison of GPS and GPS+GLONASS Positioning Performance, Lincoln Laboratory, Massachusetts Institute of Technology, (1997)

[36] Hein, W.G., Rossbach, U., and Eissfeller, B. Advances in GPS/GLONASS Combined Solutions, Institute of Geodesy and Navigation, University FAF Munich Neubiberg, Germany

[37] Hojo, H., Kawashima, S., Nakamura, M., Yui, K., Okuyama, A., Washizu, K. and Ogawa, K. An Integrated GPS/GLONASS Receiver, Japan Radio Company Ltd.

[38] Kaya, F.A and Sarıtaş M, A Computer Simulation of Dilution of Precision in the Global Positioning System Using MATLAB, Gazi University Faculty of Engineering and Architecture Department of Electrical and Electronics Engineering (December, 2005)

APPENDIX A

POSITION COMPUTATION OF GPS SATELLITES

In this section, mathematical model for position computation of a GPS satellite is given. GPS ICD-200C and a paper as an expanded version of ICD given in reference [19] are widely utilized to present algorithm steps.

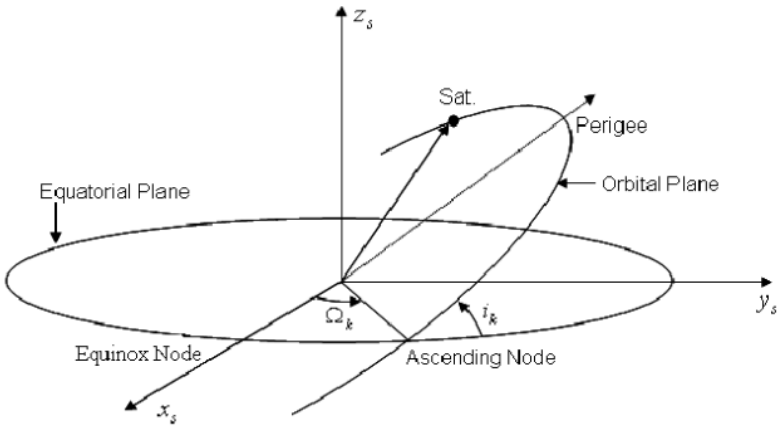

Figure 96 - The Keplerian Orbital Elements Defining the Orientation of the Orbit [Retrieved from Reference 19]

x_s axis points toward intersection between equator and the Greenwich meridian.z_s axis coincides with the spin axis of the Earth.. y_s forms right-handed coordinate system.

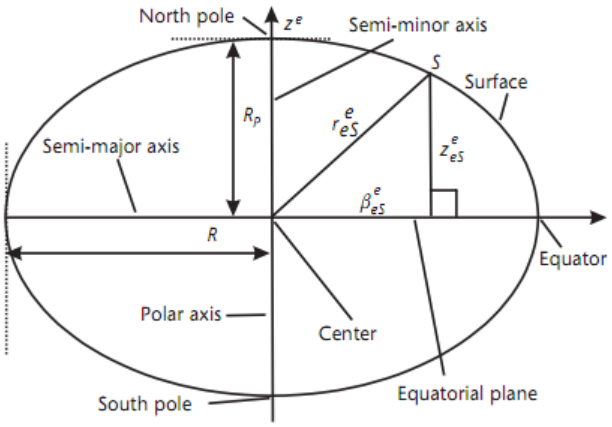

Figure 97 - Cross-section of the ellipsoid representing the Earth's surface [Retrieved from Reference 10]

Firstly, GPS ephemeris and satellite orbit parameters which used to compute satellite are defined:

Parameter	Definition
t_{oe}	Time of *Ephemeris*
$\sqrt{a}$	Square root of semi-major axis of orbit
e	Eccentricity
i_0	Inclination angle at reference time
Ω_0	Longitude of ascending node of orbital plane at weekly epoch
w	Argument of Perigee
M_0	Mean anomaly at t_{oe}
i_{dot}	Rate of change of inclination angle
$\dot{\Omega}$	Rate of change of longitude of ascending node
Δn	Mean motion difference from computed value
C_{us}	Amplitude of sine correction to argument of latitude
C_{uc}	Amplitude of cosine correction to argument of latitude
C_{rs}	Amplitude of sine correction to argument of radius
C_{rc}	Amplitude of cosine correction to argument of radius
C_{is}	Amplitude of sine correction to argument of inclination
C_{ic}	Amplitude of cosine correction to argument of inclination
$\dot{\Omega}_e$, w_e	WGS-84 Earth rotation rate
μ	WGS-84 Universal gravitational parameter

Algorithm steps are given as following:

$t_k = mod(t_k, 302400\ saniye)$	
$a = (\sqrt{a})^2$	Semi major axis
$n_0 = \sqrt{\frac{\mu}{A^3}}$	Corrected mean motion $\mu = 3.986005e14\ m^3/s^2$
$n = n_0 + \Delta n$	
$M_k = M_0 + nt_k$	Mean anomaly
$M_k = E_k - esin(E_k)$	Eccentric anomaly, E_k
$v_k = atan2(\frac{\sqrt{1-e^2}\sin(E_k)}{1-ecos(E_k)}, \frac{\cos(E_k)-e}{1-ecos(E_k)})$	True anomaly, v_k
$\emptyset_k = v_k + w$	Argument of latitude
$\partial\emptyset_k = C_{us}\sin(2\emptyset_k) + C_{uc}\cos(2\emptyset_k)$	Argument of latitude correction
$\partial r_k = C_{rs}\sin(2\emptyset_k) + C_{rc}\cos(2\emptyset_k)$	Radius correction
$\partial i_k = C_{is}\sin(2\emptyset_k) + C_{ic}\cos(2\emptyset_k)$	Inclination correction
$u_k = \emptyset_{k+}\partial\emptyset_k$	Corrected argument of latitude
$r_k = A\left(\frac{1-e^2}{1+ecos(v_k)}\right) + \partial r_k$	Corrected radius
$i_k = i_0 + i_{dot}t_k + \partial i_k$	Inclination correction
$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega_e})t_k - \dot{\Omega}_e t_{oe}$	Corrected longitude of node
$x_p = r_k\cos(u_k)$	In-plane x position
$y_p = r_k\sin(u_k)$	In-plane y position
$x_e = x_p\cos(\Omega_k) - y_p\cos(i_k)\sin(\Omega_k)$	ECEF x- coordinate
$y_e = x_p\sin(\Omega_k) + y_p\cos(i_k)\cos(\Omega_k)$	ECEF y-coordinate
$z_e = y_p\sin(i_k)$	ECEF z-coordinate

APPENDIX B

POSITION COMPUTATION OF GLONASS SATELLITES

References 20, 21, 22, and GLONASS ICD are used to achieve the computation of GLONASS position algorithms and computation steps with required parameters are given in this section.

PZ-90 system in ECEF reference frame [22]:

- X axis is directed to the point of intersection of the Earth's equatorial plane and zero meridian established by Bureau International de l'Heure (BIH),
- Z axis is directed to the Conventional Terrestrial Pole as recommended by the International Earth Rotation Service (IERS)
- Y axis completes the right-handed coordinate system

Used GLONASS ephemeris parameters and definition are shared as following:

Parameter	**Definition**
t_b	Reference Time
γ	Relativistic correction
$x(t_b), y(t_b), z(t_b)$	Satellite Position in PZ-90 (km)
$\dot{x}(t_b), \dot{y}(t_b), \dot{z}(t_b)$	Satellite Velocity in PZ-90 (km/s)
$\ddot{x}(t_b), \ddot{y}(t_b), \ddot{z}(t_b)$	Satellite lunar and solar perturbations (km/s^2)

Following 6 first order differential equations are needed to compute the satellite positions in any time [7] :

$$\frac{dx}{dt} = v_x(t) \quad \text{B.1}$$

$$\frac{dy}{dt} = v_y(t) \quad \text{B.2}$$

$$\frac{dz}{dt} = v_z(t) \quad \text{B.3}$$

$$\frac{dv_x}{dt} = -\frac{\mu x}{r^3} + \frac{3}{2} C_{20} \frac{\mu a_e^2 x}{r^5}\left(1 - 5\frac{z^2}{r^2}\right) + w_e^2 x + 2w_e \frac{dy}{dt} + \ddot{x} \quad \text{B.4}$$

$$\frac{dv_y}{dt} = -\frac{\mu y}{r^3} + \frac{3}{2} C_{20} \frac{\mu a_e^2 y}{r^5}\left(1 - 5\frac{z^2}{r^2}\right) + w_e^2 y - 2w_e \frac{dx}{dt} + \ddot{y} \quad \text{B.5}$$

$$\frac{dv_z}{dt} = -\frac{\mu z}{r^3} + \frac{3}{2} C_{20} \frac{\mu a_e^2 z}{r^5}\left(3 - 5\frac{z^2}{r^2}\right) + \ddot{z} \quad \text{B.6}$$

Runge Kutta-4 formula is set:

$$Y'(t) = F(t, Y(t)) \quad \text{B.7}$$

$$K_1 = F(t_n, Y_n) \quad \text{B.8}$$

$$K_2 = F\left(t_n + \frac{h}{2}, Y_n + \frac{hK_1}{2}\right) \quad \text{B.9}$$

$$K_3 = F\left(t_n + \frac{h}{2}, Y_n + \frac{hK_2}{2}\right) \quad \text{B.10}$$

$$K_4 = F(t_n + h, Y_n + hK_3) \quad \text{B.11}$$

$$Y_{n+1} = Y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad \text{B.12}$$

Detailed algorithm implementation steps in MATLAB are given below. The first values ($w_{10}, w_{20}, w_{30}, w_{40}, w_{50}, w_{60}$) in the algorithm are obtained from ephemeris considering Earth rotation during the time of signal travel from satellite to the user. For each time interval, that will be determined (i.e. 1ms, 10ms, 1 sec, etc.), between desired time and t_b, integration is applied:

$$t_o = t_b \quad \text{B.13}$$

$$w_{10} = x \quad \text{B.14}$$

$$w_{20} = y \quad \text{B.15}$$

$$w_{30} = z \quad \text{B.16}$$

$$w_{40} = \dot{x} \quad \text{B.17}$$

$$w_{50} = \dot{y} \quad \text{B.18}$$

$$w_{60} = \dot{z} \tag{B.19}$$

$$r = \sqrt{w_{10}^2 + w_{20}^2 + w_{30}^2} \tag{B.20}$$

$$k_{11} = hw_{40} \tag{B.21}$$

$$k_{12} = hw_{50} \tag{B.22}$$

$$k_{13} = hw_{60} \tag{B.23}$$

$$k_{14} = h(-\frac{\mu w_{10}}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2 w_{10}}{r^5}\left(1 - 5\frac{w_{30}^2}{r^2}\right) + w_e^2 w_{10} + 2w_e w_{50} + \ddot{x}) \tag{B.24}$$

$$k_{15} = h(-\frac{\mu w_{20}}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2 w_{20}}{r^5}\left(1 - 5\frac{w_{30}^2}{r^2}\right) + w_e^2 w_{20} - 2w_e w_{40} + \ddot{y}) \tag{B.25}$$

$$k_{16} = h(-\frac{\mu w_{30}}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2 w_{30}}{r^5}\left(3 - 5\frac{w_{30}^2}{r^2}\right) + \ddot{z}) \tag{B.26}$$

$$k_{21} = h(w_{40} + \frac{1}{2}k_{14}) \tag{B.27}$$

$$k_{22} = h(w_{50} + \frac{1}{2}k_{15}) \tag{B.28}$$

$$k_{23} = h(w_{60} + \frac{1}{2}k_{16}) \tag{B.29}$$

$$k_{24} = h(-\frac{\mu(w_{10} + \frac{1}{2}k_{11})}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2(w_{10} + \frac{1}{2}k_{11})}{r^5}\left(1 - 5\frac{(w_{30} + \frac{1}{2}k_{13})^2}{r^2}\right) + w_e^2(w_{10} + \frac{1}{2}k_{11}) + 2w_e(w_{50} + \frac{1}{2}k_{15}) + \ddot{x}) \tag{B.30}$$

$$k_{25} = h(-\frac{\mu\left(w_{20} + \frac{1}{2}k_{12}\right)}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2\left(w_{20} + \frac{1}{2}k_{12}\right)}{r^5}\left(1 - 5\frac{(w_{30} + \frac{1}{2}k_{13})^2}{r^2}\right) + w_e^2(w_{20} + \frac{1}{2}k_{12}) - 2w_e(w_{40} + \frac{1}{2}k_{14}) + \ddot{y}) \tag{B.31}$$

$$k_{26} = h(-\frac{\mu\left(w_{30} + \frac{1}{2}k_{13}\right)}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2\left(w_{30} + \frac{1}{2}k_{13}\right)}{r^5}\left(3 - 5\frac{(w_{30} + \frac{1}{2}k_{13})^{\,2}}{r^2}\right) + \ddot{z}) \quad \text{B.32}$$

$$k_{31} = h(w_{40} + \frac{1}{2}k_{24}) \quad \text{B.33}$$

$$k_{32} = h(w_{50} + \frac{1}{2}k_{25}) \quad \text{B.34}$$

$$k_{33} = h(w_{60} + \frac{1}{2}k_{26}) \quad \text{B.35}$$

$$k_{34} = h(-\frac{\mu(w_{10} + \frac{1}{2}k_{21})}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2(w_{10} + \frac{1}{2}k_{21})}{r^5}\left(1 - 5\frac{(w_{30} + \frac{1}{2}k_{23})^{\,2}}{r^2}\right) + w_e^2(w_{10} + \frac{1}{2}k_{21}) + 2w_e(w_{50} + \frac{1}{2}k_{25}) + \ddot{x}) \quad \text{B.36}$$

$$k_{35} = h(-\frac{\mu\left(w_{20} + \frac{1}{2}k_{22}\right)}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2\left(w_{20} + \frac{1}{2}k_{22}\right)}{r^5}\left(1 - 5\frac{(w_{30} + \frac{1}{2}k_{23})^{\,2}}{r^2}\right) + w_e^2(w_{20} + \frac{1}{2}k_{22}) - 2w_e(w_{40} + \frac{1}{2}k_{24}) + \ddot{y}) \quad \text{B.37}$$

$$k_{36} = h(-\frac{\mu\left(w_{30} + \frac{1}{2}k_{23}\right)}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2\left(w_{30} + \frac{1}{2}k_{23}\right)}{r^5}\left(3 - 5\frac{(w_{30} + \frac{1}{2}k_{23})^{\,2}}{r^2}\right) + \ddot{z}) \quad \text{B.38}$$

$$k_{41} = h(w_{40} + k_{34}) \quad \text{B.39}$$

$$k_{42} = h(w_{50} + k_{35}) \quad \text{B.40}$$

$$k_{43} = h(w_{60} + k_{36}) \quad \text{B.41}$$

$$k_{44} = h(-\frac{\mu(w_{10}+k_{31})}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2(w_{10}+k_{31})}{r^5}\left(1-5\frac{(w_{30}+k_{33})^2}{r^2}\right)+ +w_e^2(w_{10}+k_{31})+2w_e(w_{50}+k_{35})+\ddot{x}) \quad \text{B.42}$$

$$k_{45} = h(-\frac{\mu(w_{20}+k_{32})}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2(w_{20}+k_{32})}{r^5}\left(1-5\frac{(w_{30}+k_{33})^2}{r^2}\right)+ +w_e^2(w_{20}+k_{32})-2w_e(w_{40}+k_{34})+\ddot{y}) \quad \text{B.43}$$

$$k_{46} = h(-\frac{\mu(w_{30}+k_{33})}{r^3} + \frac{3}{2}C_{20}\frac{\mu a_e^2(w_{30}+k_{33})}{r^5}\left(3-5\frac{(w_{30}+k_{33})^2}{r^2}\right)+\ddot{z}) \quad \text{B.44}$$

$$w_{11} = w_{10} + \frac{1}{6}(k_{11}+2k_{21}+2k_{31}+k_{41}) \quad \text{B.45}$$

$$w_{21} = w_{20} + \frac{1}{6}(k_{12}+2k_{22}+2k_{32}+k_{42}) \quad \text{B.46}$$

$$w_{31} = w_{30} + \frac{1}{6}(k_{13}+2k_{23}+2k_{33}+k_{43}) \quad \text{B.47}$$

$$w_{41} = w_{40} + \frac{1}{6}(k_{14}+2k_{24}+2k_{34}+k_{44}) \quad \text{B.48}$$

$$w_{51} = w_{50} + \frac{1}{6}(k_{15}+2k_{25}+2k_{35}+k_{45}) \quad \text{B.49}$$

$$w_{61} = w_{60} + \frac{1}{6}(k_{16}+2k_{26}+2k_{36}+k_{46}) \quad \text{B.50}$$

$$x_s = w_{11} \quad \text{B.51}$$

$$y_s = w_{21} \quad \text{B.52}$$

$$z_s = w_{31} \quad \text{B.53}$$

$$\dot{x}_s = w_{41} \quad \text{B.54}$$

$$\dot{y}_s = w_{51} \quad \text{B.55}$$

$$\dot{z}_s = w_{61} \quad \text{B.56}$$

APPENDIX C

CONCEPT OF MULTI-LATERATION USING PSEUDORANGE MEASUREMENTS

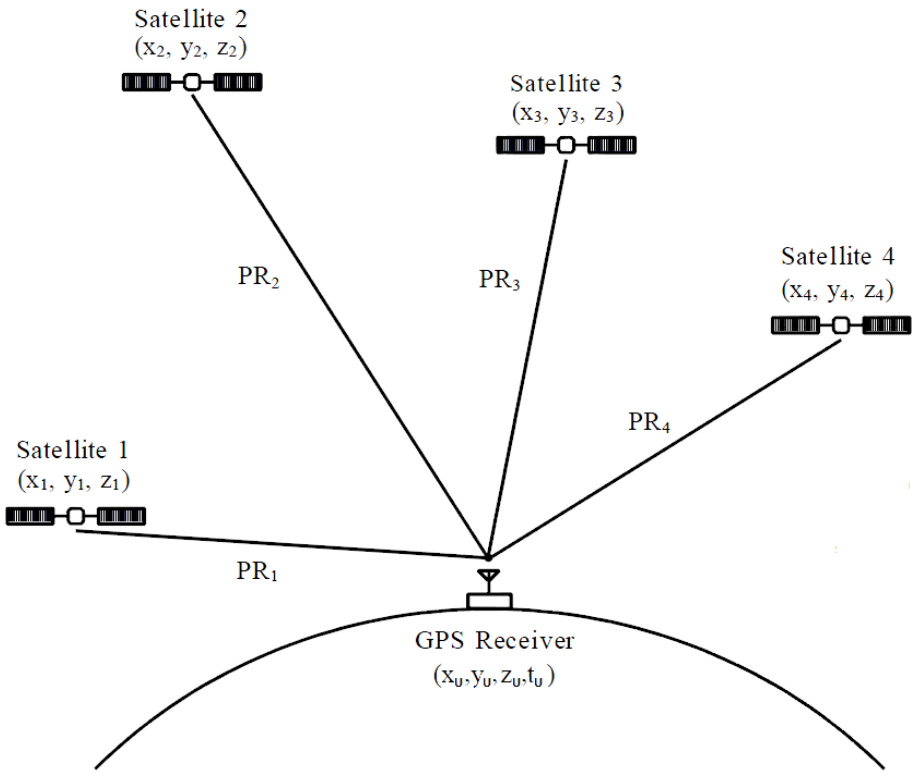

Figure 98 - Calculating Position Using GPS Pseudoranges [25]

Figure 98 shows the concept of multi-lateration using pseudorange (ρ) measurements from at least 4 satellites to solve for 3D position and time.

In order to determine user position in three dimensions (x_u, y_u, z_u) and time offset t_u, pseudorange measurements are made to four satellites.

$$\rho_1 = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + ct_u \quad \text{C.1}$$

$$\rho_2 = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + ct_u \quad \text{C.2}$$

$$\rho_3 = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + ct_u \quad \text{C.3}$$

$$\rho_4 = \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + ct_u \quad \text{C.4}$$

By performing iterative techniques, the equations C.1-C.4 are solved based on linearization.

If we know approximately where the receiver is, then we can denote the offset of the true position (x_u, y_u, z_u) from the approximate position $(\hat{x}_u, \hat{y}_u, \hat{z}_u)$ by a displacement $(\Delta x_u, \Delta y_u, \Delta z_u)$.

$$\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + ct_u = f(x_u, y_u, z_u, t_u) \quad \text{C.5}$$

$$\hat{\rho}_i = \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2 + (z_i - \hat{z}_u)^2} + ct_u = f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u) \quad \text{C.6}$$

The unknown user position and receiver clock offset is considered to consist of an approximate component and an incremental component:

$$x_u = \hat{x}_u + \Delta x_u \quad \text{C.7}$$

$$y_u = \hat{y}_u + \Delta y_u \quad \text{C.8}$$

$$z_u = \hat{z}_u + \Delta z_u \quad \text{C.9}$$

$$t_u = \hat{t}_u + \Delta t_u \quad \text{C.10}$$

Therefore, we can write,

$$f(x_u, y_u, z_u, t_u) = f(\hat{x}_u + \Delta x_u, \hat{y}_u + \Delta y_u, \hat{z}_u + \Delta z_u, \hat{t}_u + \Delta t_u) \quad \text{C.11}$$

$$\begin{aligned} f(\hat{x}_u + \Delta x_u, \hat{y}_u + \Delta y_u, \hat{z}_u + \Delta z_u, \hat{t}_u + \Delta t_u) &= f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u) \\ &+ \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{x}_u} \Delta x_u + \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{y}_u} \Delta y_u \\ &+ \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{z}_u} \Delta z_u + \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{t}_u} \Delta t_u \end{aligned} \quad \text{C.12}$$

Taylor series expansion has been truncated after the first-order partial derivatives to eliminate non-linear terms. The partial derivatives evaluate as follows:

$$\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{x}_u} = -\frac{x_i - \hat{x}_u}{\hat{r}_i} \quad \text{C.13}$$

$$\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{y}_u} = -\frac{y_i - \hat{y}_u}{\hat{r}_i} \quad \text{C.14}$$

$$\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{z}_u} = -\frac{z_i - \hat{z}_u}{\hat{r}_i} \quad \text{C.15}$$

$$\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)}{\partial \hat{t}_u} = c \quad \text{C.16}$$

$$\hat{r}_i = \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2 + (z_i - \hat{z}_u)^2} \quad \text{C.17}$$

$$\rho_i = \hat{\rho}_i - \frac{x_i - \hat{x}_u}{\hat{r}_i} \Delta x_u - \frac{y_i - \hat{y}_u}{\hat{r}_i} \Delta y_u - \frac{z_i - \hat{z}_u}{\hat{r}_i} \Delta z_u + c t_u \quad \text{C.18}$$

$$\hat{\rho}_i - \rho_i = \frac{x_i - \hat{x}_u}{\hat{r}_i} \Delta x_u + \frac{y_i - \hat{y}_u}{\hat{r}_i} \Delta y_u + \frac{z_i - \hat{z}_u}{\hat{r}_i} \Delta z_u - c t_u \quad \text{C.19}$$

$$\Delta\rho_i = \hat{\rho}_i - \rho_i \quad \text{C.20}$$

$$a_{xi} = \frac{x_i - \hat{x}_u}{\hat{r}_i} \quad \text{C.21}$$

$$a_{yi} = \frac{y_i - \hat{y}_u}{\hat{r}_i} \quad \text{C.22}$$

$$a_{zi} = \frac{z_i - \hat{z}_u}{\hat{r}_i} \quad \text{C.23}$$

$$\Delta\rho_i = a_{xi}\Delta x_u + a_{yi}\Delta y_u + a_{zi}\Delta z_u - ct_u \quad \text{C.24}$$

$$\Delta\rho_1 = a_{x1}\Delta x_u + a_{y1}\Delta y_u + a_{z1}\Delta z_u - ct_u \quad \text{C.25}$$

$$\Delta\rho_2 = a_{x2}\Delta x_u + a_{y2}\Delta y_u + a_{z2}\Delta z_u - ct_u \quad \text{C.26}$$

$$\Delta\rho_3 = a_{x3}\Delta x_u + a_{y3}\Delta y_u + a_{z3}\Delta z_u - ct_u \quad \text{C.27}$$

$$\Delta\rho_4 = a_{x4}\Delta x_u + a_{y4}\Delta y_u + a_{z4}\Delta z_u - ct_u \quad \text{C.28}$$

These equations can be put in matrix form by making the definitions:

$$\Delta\rho = \begin{bmatrix}\Delta\rho_1\\ \Delta\rho_2\\ \Delta\rho_3\\ \Delta\rho_4\end{bmatrix}, \quad H = \begin{bmatrix}a_{x1} & a_{y1} & a_{z1} & 1\\ a_{x2} & a_{y2} & a_{z2} & 1\\ a_{x3} & a_{y3} & a_{z3} & 1\\ a_{x4} & a_{y4} & a_{z4} & 1\end{bmatrix}, \quad \Delta x = \begin{bmatrix}\Delta x_u\\ \Delta y_u\\ \Delta z_u\\ -ct_u\end{bmatrix} \quad \text{C.29}$$

$$\Delta\rho = H\Delta x \;\longrightarrow\; \Delta x = H^{-1}\Delta\rho \quad \text{C.30}$$

The basic algorithm as follows: [25]

i. Calculate satellite positions
ii. Apply satellite clock corrections to the pseudorange measurements
iii. Form initial position estimate
iv. Iterate until convergence,
    - Calculate approximate pseudoranges based on position estimate and SV positions
    - Form the geometry matrix H
    - Subtract measured pseudoranges from estimated pseudoranges
    - Update the user states by solving $\Delta\rho = H\Delta x$

End Loop

This linearization scheme will work well as long as the displacement $(\Delta x_u, \Delta y_u \Delta z_u)$ is within close proximity of the linearization point. The acceptable displacement is dictated by the user's accuracy requirements. If the displacement does exceed the acceptable value, this process is reiterated with $\hat{\rho}$ being replaced by a new estimate of pseudorange based on the calculated point coordinates (x_u, y_u, z_u). [5]

The dilutions of precision (DOP) parameters' computations are as follows [5]:

$$(H^T H)^{-1} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix} \quad \text{C.31}$$

$$GDOP = \sqrt{D_{11} + D_{22} + D_{33} + D_{44}} \quad \text{C.32}$$

$$PDOP = \sqrt{D_{11} + D_{22} + D_{33}} \quad \text{C.33}$$

$$HDOP = \sqrt{D_{11} + D_{22}} \quad \text{C.34}$$

$$TDOP = \frac{\sqrt{D_{44}}}{c} \qquad c : speed\ of\ light \quad \text{C.35}$$