Multi-GNSS Simulations for Satellite Point Positioning

Alexander Dolgansky

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE

YORK UNIVERSITY,

TORONTO, ONTARIO

February, 2010

Multi-GNSS Simulations for Satellite Point Positioning

by **Alexander Dolgansky**

a thesis submitted to the Faculty of Graduate Studies of York University in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

Multi-GNSS Simulations for Satellite Point Positioning

by **Alexander Dolgansky**

Examination Committee members:
1. Jian-Guo Wang
2. Sunil Bisnath
3. Jinjun Shan
4. Andrew Eckford

Abstract

Global positioning and navigation relies on Global Navigation Satellite Systems (GNSSs), the best known among these being the Global Positioning System (GPS), followed by GLObal NAvigation Satellite System (GLONASS). With ongoing development of the Galileo and Compass satellite navigation systems, it appears that multi-GNSS data processing will become the norm in the next 20 years. It is now an opportune moment to evaluate and develop the potential benefits that may accrue from processing multi-GNSS data. The desire to pursue such an opportunity has given rise to the Multi-GNSS Observables Simulator (MGOS), a simulation package developed during the course of this research to study multi-GNSS scenarios involving multiple receiver platforms. A primary objective of this research is to evaluate how well point positioning can be applied to Low Earth Orbit (LEO) and Highly Elliptical Orbit (HEO) satellites. MGOS has a modular design, and provides a graphical user interface. Fundamentally, it simulates GNSS satellite and receiver trajectories, GNSS measurement noise, and GNSS receiver measurements. MGOS' simulation capabilities have been tested with external processing software for Precise Point Positioning (PPP) and relative positioning. MGOS is shown to be capable of simulating realistic measurements that allow for centimetre-level positioning or better. and its built-in own processing module is demonstrated to operate on par with high-quality Precise Point Positioning (PPP) processors.

In terms of the LEO and HEO studies, multiple GNSSs were incrementally combined to show how an increased number of satellites affects positioning performance. The LEO and HEO orbit types were chosen to exploit the dramatic contrast in their orbital dynamics. In addition, a sidelobe tracking feature was introduced to HEO scenarios to allow tracking of more satellites. Simulations show that multi-GNSS LEO positioning is similar in performance to kinematic ground based positioning. In the HEO case, it is shown that a multi-GNSS receiver is a necessity, since satellite geometry is very poor. Sidelobe tracking is desirable, since it can improve positioning performance to almost that afforded by LEO satellites (~1 m or less).

Analysis of position errors, Geometric Dilution of Precision (GDOP), and filter performance were evaluated. Position Root Mean Squared Errors (RMSEs) (3D) for the LEO simulations are shown to range between 2.0 metres (GPS-only case) to 0.9 metres (GPS + GLONASS + Galileo + Compass case). For the HEO simulations it has been found that although it is generally beneficial to track more satellites, each new satellite increases measurement noise which can negatively affect positioning performance. The issue of diminishing returns is also investigated by systematically adding GNSSs. Positioning performance improvements diminish with each GNSS added for both LEO and HEO simulations.

Acknowledgements

I would like to thank my supervisors, professors Anthony Szeto and Sunil Bisnath. I feel that this journey would not have been possible if not for their generous support, encouragement, and all of the effort they have put to make this research a success. They have pushed me through roadblocks and allowed me to expand my knowledge on the field of satellite position and navigation, and mature my character for future life endeavours. I would also like to extend my gratitude to my family members who encouraged me to think positively and provided endless love and understanding I needed to keep going. Special thanks also go to GEOIDE and NSERC for providing financial support for this work. Without these sponsors, this work would not have been possible.

Contents

List of Tables

List of Figures

Acronyms

AltBOC	Alternative Binary Offset Carrier
ANTEX	ANTenna EXchange
BOC	Binary Offset Carrier
BPSK	Binary Phase Shift Keying
CBOC	Composite Binary Offset Signals
CDMA	Code Division Multiple Access
ECEF	Earth-Centred Earth-Fixed
EOS	Earth Orbiting Spacecraft
ESA	European Space Agency
FDMA	Frequency Division Multiple Access
GDOP	Geometric Dilution of Precision
GEO	Geostationary Earth Orbit
GLONASS	GLObal NAvigation Satellite System

GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSO	Geosynchronous Orbit
GUI	Graphical User Interface
HEO	Highly Elliptical Orbit
IGS	International GNSS Service
IONEX	IONosphere Map EXchange
ITRF	International Terrestrial Reference Frame
KML	Keyhole Markup Language
LEO	Low Earth Orbit
MBOC	Multiplexed Binary Offset Carrier
MEO	Medium Earth Orbit
MGOS	Multi-GNSS Observables Simulator
NGS	National Geodetic Survey
PPP	Precise Point Positioning
PRN	Pseudorandom Noise
QPSK	Quadrature Phase Shift Keying
RINEX	Receiver INdependent Exchange Format

RMSE	Root Mean Squared Error
RMS	Root Mean Square
RTK	Real Time Kinematic
SVN	Satellite Vehicle Number
TEC	Total Electron Content
TFSR	Tapped Feedback Shift Register
TLE	Two Line Element

Chapter 1

Introduction

Positioning on a global scale became practical with the introduction of satellite-based positioning. A constellation of satellites of known position equipped with transmitters is used to broadcast signals to a ground-based receiver, which then uses these signals to position itself. Exactly how broadcast signals are generated and how they are used depends on the specific design of the satellite positioning system. The TRANSIT system, the U.S. Navy's satellite positioning constellation of the 1970s and 80s, used Doppler shifts of several passing satellites to determine receiver position (Black, 1990). Such a method led to position errors in the order of kilometres, because satellite range was derived from Doppler information only, with satellite positions left entirely unknown and in need of estimation (Aksnes et al., 1988). A better method was developed for GPS that used specially modulated signals. These signals can be used to both obtain range to satellites and satellite positions without any knowledge of satellite positions and orbital behaviour on the part of the user.

GPS was introduced in the late 1980s as the follow-on to TRANSIT, and was originally stipulated as a constellation of 24 satellites in six orbital planes, inclined at 55 degrees to the equator, with orbital periods of approximately 12 hours. Today over 30 satellites are in use. Many satellites launched in the early 1990s are still operational today (U.S. Naval Observatory, 2009). GPS satellites are distinguished from one another by means of Code Division Multiple Access (CDMA). This technique uses a unique Pseudorandom Noise (PRN) code that identifies each satellite, which enables accurate ranging information to be collected. The broadcast signal structure includes a detailed navigation message, which is used to determine various corrections. This information is used to assist in computing satellite positions. GPS is self-contained in the sense that it does not require access to any information beyond that transmitted by satellites.

The discussion of other GNSSs is presented in the next section. Following that, the main thesis objectives are presented, along with a discussion of the novelty of this work. The summary of the thesis structure is used to conclude this chapter.

1.1 Introduction to Global Navigation Satellite Systems (GNSSs)

GPS is not the only GNSS available for positioning. The former Soviet Union began developing a system similar to GPS in the 1980s. This system is known as the GLObal NAvigation Satellite System (GLONASS) and, at the time of writing, has nearly reached full operational status. Unlike GPS, its satellites are not differentiated by PRN codes, but rather by means of Frequency Division Multiple Access (FDMA). Moreover, GLONASS uses a reference system that is different from that of GPS (Leick, 2004). Despite these differences, multi-GNSS positioning can be achieved. The capability of multi-GNSS positioning will likely improve dramatically in the next 20 or so years as other GNSSs come into service.

Another system is Galileo, being developed by the European Union and scheduled to be completed around 2018. While this system is modelled on GPS, it is predicated to be completely open to civilian use, and in fact will be operated by non-military agencies. Galileo will feature 30 satellites in three orbital planes, inclined at or beyond 56 degrees. Another system is Compass, a Chinese system featuring geostationary satellites as part of its constellation. Compass, or Beidou 2, is an extension to the Beidou 1, a project initially intended for regional positioning services. Exact details on Compass are yet to be released by relevant Chinese authorities. Galileo and Compass will feature radically different signal structures from those used by GPS. Binary Offset Carrier (BOC) or its more advance version Multiplexed Binary Offset Carrier (MBOC) signal modulation will be used in Galileo and Compass. BOC and MBOC signals provide better tracking and indoor capabilities than GPS signals, and can split a single data channel into two (Avila-Rodriguez et al., 2007). Chapter 2 contains more information on signal structures.

Raw range measurements derived from satellites in a particular GNSS must be processed in order to compute a receiver position. A basic technique utilized by most commercial receivers is point positioning. Point positioning uses least-squares or Kalman filtering to estimate position based on functional models of observations. The accuracy of these techniques depends on several factors, such as the quality of receiver signal tracking and the quality of error modelling. Satellite-receiver range is determined by signal travel time from a satellite to a receiver. By correlating a PRN sequence used by a satellite and a replica produced in a receiver, the lag time at which these two PRN sequences align provides the time delay. Multiplying the travel time by the speed of light will result in a range estimate.

1.2 Thesis Objectives

The primary objective of this research is to study benefits that may arise when processing data from more than one GNSS. It is generally expected that if data from multiple GNSSs can be successfully combined together, position accuracy will improve. This work will examine these expectations for receivers mounted on Earth Orbiting Spacecraft (EOS), rather than ground or near-ground platforms. Due to the availability of GLONASS data and the promise of more GNSSs, a great deal of research has been conducted on the benefits of using multi-GNSS data (Cai and Gao, 2007; Wanninger and Wallstab-Freitag, 2007; Zinoviev, 2005). Interoperability of GNSSs becomes important when combining data. Differences between systems must be taken into account if one wishes to process GPS and GLONASS data together. Encouragingly, it has been shown to be beneficial to process GPS data combined with GLONASS data (Zinoviev, 2005).

The secondary objective of this thesis is to explore applications involving multi-GNSS Earth Orbiting Spacecraft (EOS). Two types of orbits will be examined: Low Earth Orbit (LEO) and Highly Elliptical Orbit (HEO). LEOs are near-circular orbits with orbital altitudes ranging from several hundred kilometres to several thousand kilometres. HEOs are highly elliptical orbits with apogee radii that can extend beyond GPS orbits. A HEO called Geo-Transfer Orbit can be used to transfer satellites from LEO to Geostationary Earth Orbit (GEO) or Geosynchronous Orbit (GSO) (Larson and Wertz, 1999). HEO satellites can be used for telecommunication satellites as an alternative to GEOs. As GEO and GSO objects may collide due to crowding (Pardini and Anselmo, 2001; Wytrzyszczak et al., 2004), so accurate HEO positioning would be useful for GEO or GSO satellite placement or for debris removal. Moreover, accurate LEO positioning using a multi-GNSS receiver can reduce operational costs, since extensive ground-tracking of satellites would become unnecessary (Montenbruck et al., 2008). Unfortunately, no multi-GNSS-equipped LEO satellites or even GPS equipped HEO satellites are currently available to perform such studies. The fact that Galileo and Compass systems are not available altogether renders such investigations even more difficult. One way to perform such studies is by means of simulation. A multi-GNSS simulator capable of simulating LEO and HEO satellite trajectories, GNSS satellite trajectories, as well as relevant error sources, appears to be essential.

Three types of GNSS simulators are currently available: software simulators, hardware simulators, and hybrids. A software simulator is essentially a computer program which simulates raw observables. This program does not rely on any external devices to acquire or process signals, and may or may not use real-life data. A hardware simulator is a stand-alone unit that can simulate radio frequency signals and signal errors, and is primary designed to test receiver equipment. A hybrid receiver is a combination of software and hardware simulators. Hardware simulators are expensive, as they require a dedicated hardware base, and are often designed in a fashion not to facilitate modification. Software simulators tend to be much less costly, since they do not require dedicated hardware in order to operate. Software simulator and hybrid simulators can be modified or improved if source code is provided. A custom software simulator proves necessary for this research.

1.3 Novelty of Research

As stated, conducting multi-GNSS studies using real data is currently not possible. As a result, simulation appears to be the only feasible methodology for evaluating multi-GNSS positioning and navigation. A custom made simulator named the Multi-GNSS Observables Simulator (MGOS) will be introduced in Chapter 3. MGOS is a Graphical User

Interface (GUI) based program written in C++ designed to simulate realistic multi-GNSS measurements. Using MGOS, LEO and HEO studies will be conducted. The confidence in realism of these studies will be established using various tests conducted with external software packages designed to process real-life data files. Although LEO studies using a GPS-receiver have been conducted elsewhere (Bisnath and Langley, 2001; Bock et al., 2009; Švehla and Rothacher, 2003), HEO studies involving GPS (let alone GLONASS, Galileo or Compass) are difficult to find. This work is likely to be the first set of multi-GNSS LEO and HEO studies ever conducted.

MGOS is capable of exporting its simulated measurements to a Receiver INdependent Exchange Format (RINEX) file used by many processing agencies and widely supported by receiver manufacturers. Using RINEX output, it will be possible to ascertain how realistic MGOS can simulate GNSS observables; a feature not common to GNSS simulators examined in Chapter 3. Unfortunately, data processors employed in this study only deal with GPS data, so evaluation of the accuracy of processed MGOS simulated data will need to be gauged by GPS simulations alone. It will be shown in Chapter 3 that MGOS is indeed capable of simulating GPS measurements as accurately as those produced by high quality receivers. In fact, MGOS is capable of simulating data files that can be combined with real data files to achieve centimetre or better positioning. Such capabilities are limited to high-grade software GNSS receivers. Fortunately for MGOS users, this software is open-source, which means it can be used at no cost and it can redistributed. Furthermore, due to its modular structure MGOS capabilities can be readily modified or enhanced.

1.4 Thesis Outline

Chapter 2 presents descriptions of various GNSSs, error sources, positioning techniques, and LEO and HEO positioning. Chapter 3 is divided into three parts. The first part discusses existing multi-GNSS simulators and how the need to develop MGOS came about. The second part introduces MGOS, focusing on design, features, simulation capabilities, and error source modelling. The last part introduces three tests used to evaluate MGOS: static PPP, relative positioning, and kinematic PPP, and present results for the same. Chapter 4 introduces LEO and HEO studies conducted using MGOS, and present results. Explanations of various MGOS components necessary for LEO and HEO studies are included in Chapter 4. Chapter 5 presents conclusions and indicate suggestions for future work.

Chapter 2

Overview of GNSSs and Satellite Point Positioning

This chapter presents information on signal structures of GPS, GLONASS, Galileo and Compass. The structure of GNSS signals must be studied to facilitate understanding of signal tracking and processing of observables. The point positioning technique and how it can be used to position Earth orbiting satellites will be also presented.

2.1 Overview of GNSSs

In this section, background information on the four major GNSSs: GPS, GLONASS, Galileo and Compass is presented, including information on signals and their modulation.

2.1.1 Signal Tracking Fundamentals

Communication, telemetry and range signals are transmitted on a carrier wave. To transfer information, the carrier wave is modulated. Modulation entails multiplying the carrier wave by a modulating signal with a mixer. The output from the mixer is often passed through a filter to remove unnecessary harmonics generated by the mixer. The selection process for the frequency of a carrier wave is not trivial. Factors such as environmental effects (e.g., ionospheric and tropospheric delays), regulations regarding transmission, and receiver signal acquisition issues are all part of the decision process. Moreover, it is often required to transmit several carrier waves at different frequencies or channels. Each channel may be modulated with a distinct modulation signal, and may play a particular role in a specific service offered by a GNSS.

2.1.2 Binary Modulation Signals

The carrier is modulated by a binary signal consisting of a sequence of pseudo-random pulses of electromagnetic radiation. Such signals are generated by a Tapped Feedback Shift Register (TFSR) (GPS Joint Program Office, 1993; Coordination Scientific Information Center, 2002). An example TFSR circuit is shown in Figure 2.1, which will be used to discuss how a TFSR generates pseudorandom codes. Note that the TFSR shown in Figure 2.1 is one of many possible implementations of such a device and was drawn based on schematics from the GPS Joint Program Office (1993) and a description of a 4-stage TFSR presented in Ziff (1998).

Figure 2.1 is an example schematic diagram of a 4 stage TFSR with tapping sequence described by the following polynomial: $1+X+X^3$. The polynomial indicates that the outputs of the first and the third shift registers (the D flip-flops) are to be modulo-2 added together and fed into the first shift register. The signal entering the input terminal of one shift register is fed to the input of another shift register with every clock cycle (the box labelled “Clock”). The “Init” and “Reset” buttons can be used to perform initialization and

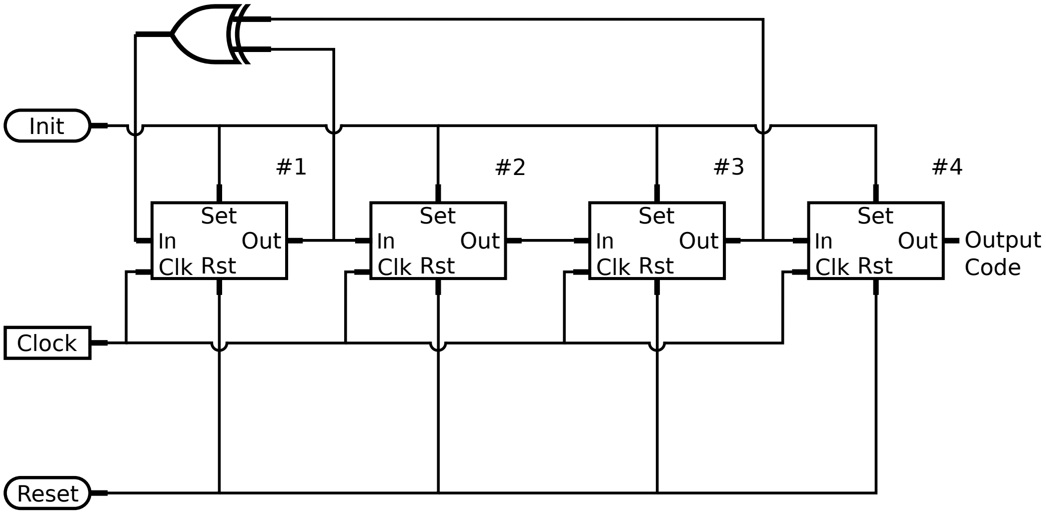


Figure 2.1: Circuit of a simple 4-stage Tapped Feedback Shift Register

reset operations, respectively. These operations are crucial parts of the TFSR operation, and are described in more detail in the next section.

2.1.3 Overview of GPS

GPS signals utilize spread-spectrum technology to increase their total radiated power in relation to their transmission flux density. A spread-spectrum signal is created by the mixing of a small bandwidth data signal with a large bandwidth "spreading" signal. To differentiate between signals sent by different satellites, GPS adopts Code Division Multiple Access (CDMA). GPS satellites transmit on two frequency channels, L1 (1.57542 GHz) and L2 (1.22760 GHz) (Hofmann-Wellenhof et al., 2001). L1 is intended for general use, and is modulated with the civilian accessible C/A-code. The C/A-code can be used to achieve few metre-level positioning using a single-frequency receiver. L1 is also modulated with the more complex P-code. The P-code offers improved positioning performance over the C/A-code, is only available to authorized parties, and is further be modulated by the encrypted W-code to form the Y-code or P(Y)-code (Hofmann-Wellenhof et al., 2001). The modernization of GPS has brought a new civilian code (known as L2C) to the L2 frequency. GPS satellites belonging to block IIF also modulate L2 with new civilian codes called L2CM and L2CL (Leick, 2004). As these codes were only introduced in 2005, they are not used in this thesis, as only those codes that are currently used extensively will be discussed. Future GPS satellites will also transmit a third frequency L5 (1.17645 GHz).

2.1.3.1 GPS Channels and Associated Signals

The GPS carrier can be modulated by four signals: C/A-code, P-code, W-code and the navigation message. The navigation message is not a pseudo-random sequence, but rather a coded binary message. The other codes are distinguished by their chipping rates, code lengths and assigned roles. The C/A-code or Coarse/Acquisition-code appears only on L1, has a chipping rate of 1.023 MHz, and code length of one millisecond. This code is designed primarily for quick signal acquisition. The P-code appears on both GPS frequencies, is 90° out of phase with the C/A-code, has a frequency of 10.23 MHz, and a code length spanning 37 weeks.

To improve acquisition time, the P-code is reset every week to a new epoch for each satellite, bringing the effective code length to just one week. The purpose of the P-code is to provide a more precise ranging signal to a restricted set of users. The W-code and the navigation message are not directly used to establish the position of the user's receiver. The W-code is used to convert the P-code into the Y-code (which is simply an encrypted version of the P-code) to prevent unauthorized use. The W-code has a frequency of 0.5115 MHz and a code length equal to that of the P-code. The navigation message is used to provide the user with satellite orbit information, satellite clock, satellite health status, various correction terms, and the ability to track the P-code using the so called Z-count. The navigation message has a frequency of 50 Hz, and takes 30 seconds to be acquired (Hofmann-Wellenhof et al., 2001).

2.1.3.2 GPS C/A-Code Modulation

The C/A-code is used to provide the standard positioning service to GPS users. Its is generated using two 10-state TFSRs connected together with an XOR (exclusive OR) gate. This process generates a Gold Code (GPS Joint Program Office, 1993). The code consists of 1023 chips, and is 1 millisecond long. The two 10 state TFSRs are initialized to all ones and are designated as G1 and G2. G1 is tapped at positions 3 and 10, while G2 is tapped at positions 2, 3, 6, 8, 9 and 10. Once 1023 chips have been generated, the registers are reset to a new epoch. Each satellite is assigned a unique tapping position on the G2 register through a "phase select logic" circuit, which effectively delays the C/A-code by increments of 1 chip. Each C/A-code epoch is used to generate a message log clock signal using a circuit known as "1023 DECODE" (which creates a positive pulse with every new C/A-code epoch). A frequency divider is used to reduce the 1 kpbs rate of the C/A-code to 50 bps (GPS Joint Program Office, 1993).

2.1.3.3 GPS P-Code Modulation

The P-code offers a precise positioning service to GPS users. The P-code is generated from four 12-stage TFSRs combined in the following fashion. The first two 12-stage TFSRs, labelled X1A and X1B, are combined via an XOR gate to form a Gold Code. X1A is tapped at positions 6, 8, 11, and 12, while X1B is tapped at positions 1, 2, 5, 8, 9, 10, 11, and 12. All tapped register values are fed through an XOR gate, and the output is returned to the first register of the next epoch. The other two 12 stage TFSRs are designated X2A and X2B. Their tap positions are as follows: X2A is tapped at positions 1, 3, 4, 5, 7, 8, 9, 10, 11, and 12, and X2B is tapped at positions 2, 3, 4, 8, 9, and 12.

The Gold Code generated by X1A and X1B is labelled X1, while the Gold Code generated by the X2A and X2B is X2. The output X2 is fed into a shift register whose length depends on the Satellite Vehicle Number (SVN). For example, SVN 1 will have a shift register of length 2, while SVN 2 will have a shift register of length 3. The role of this extra shift register is to generate a unique P-code for each satellite. P-code length is very long due to the number and size of the TFSRs used. Each 12-stage TFSR can generate 4095 chips of code before repeating. Short-cycling of the registers or deliberate reduction in the length of a generated code is used to create a delay between X1 and X2, which improves the random nature of the P-code (GPS Joint Program Office, 1993).

2.1.4 Overview of GLONASS

The GLONASS can be traced back to the 1980s when the U.S. was developing GPS. Like GPS, the original goal for GLONASS was to have 24 operational satellites. This goal was reached in 1998. However, due to financial difficulties and short satellite life, the number of available GLONASS satellites diminished to 5 in 2002 (Leick, 2004). Recent commitments by the Russian government have raised the number of active dual-frequency satellites to 18 at the time of this writing. GLONASS is also currently undergoing a significant modernization. However, for many reasons, the major one being the relatively low number of satellites (Gibbons, 2008b), GLONASS does not yet offer comparable performance to GPS. There are many reasons for this. Nevertheless, GLONASS can be used to strengthen GPS solutions (Leick, 2004).

GLONASS satellites currently transmit on two frequency channels labelled L1 and L2. Like GPS, two types of services are provided to users: a standard positioning service and a precise positioning service. These are equivalent to C/A-code and P-code services available from GPS. GLONASS PRN codes are not as complex as those employed in GPS, since satellites are distinguished by frequency. GLONASS and GPS also differ in the way satellite broadcast ephemeris are formatted. GPS provides necessary corrections and coefficients for a standard algorithm that can be used to determine few metre-level satellite positions. GLONASS provides instantaneous satellite positions at half-hour intervals. These positions need to be extrapolated to a desired epoch (Coordination Scientific Information Center, 2002). Depending on the extrapolation method, varying accuracies can be achieved (Zinoviev, 2005).

2.1.4.1 GLONASS Signal Modulation

Unlike GPS (or other GNSSs to be discussed), GLONASS uses Frequency Division Multiple Access (FDMA) to distinguish the signals transmitted from its satellites. FDMA reduces the need for complex PRN modulation schemes in CDMA systems. GLONASS uses a 9-stage shift register and the polynomial $1+X^5+X^9$ to generate a one millisecond code consisting of 511 chips. The shift register is driven by a 0.511 MHz clock signal derived from an on-board 5.0 MHz master clock (Coordination Scientific Information Center, 2002).

The navigation message transmitted by GLONASS satellites is composed of a sequence of repeating strings of 2 seconds duration. The first 1.7 seconds of each of the strings contains 85 bits of data transmitted at 50 bps, while the last 0.3 seconds contains the 30-bit time mark transmitted at 100 bps (Coordination Scientific Information Center, 2002).

The time mark is a short PRN sequence generated with a 9-state shift register and the polynomial $1+X^3+X^5$. The final message is a modulo-2 summation of the square-wave signal running at 100 bps, a coded message as well as the coded message delayed by 1-bit. The addition of the 1-bit delayed coded message signal with the original coded message is used to complete the time mark of the previous navigation message.

The navigation message itself transmits the following terms: enumeration of the satellite time marks, difference between on-board time and GLONASS time, relative difference between carrier frequency and its nominal value, ephemeris parameters, status of all satellites in the constellation, coarse correction terms for on-board clock drifts for all satellites in the constellation, coarse orbital parameters of all satellites, and correction to GLONASS time relative to UTC(SU) (Revnivykh, 2005).

2.1.4.2 GLONASS Channels and Associated Signals

GLONASS satellites transmit on L1 and L2 frequencies, which vary from satellite to satellite and are determined by Equation 2.1:

$$\begin{aligned} f_k^{L1} &= f_0^{L1} + k\Delta f^{L1} \\ f_k^{L2} &= f_0^{L2} + k\Delta f^{L2} \end{aligned} \tag{2.1}$$

In Equation 2.1, f_0^{L1} is the nominal GLONASS L1 frequency, set at 1602 MHz, f_0^{L2} is the nominal L2 at 1246 MHz, Δf^{L1} and Δf^{L2} are frequency steps for each satellite, and k is satellite frequency channel number. The value of k can be obtained from the official GLONASS website maintained by the Russian Space Agency (Information-Analytical Centre, 2006). The cross interference between adjacent frequency channels is guaranteed to not exceed -48 dB. The expected power level is guaranteed to be no less than -161 dBW for the L1 sub-band, and no less than -167 dBW for the L2 sub-band for any satellite visible above the horizon at elevation angles of 5° or greater (Revnivykh, 2005).

GLONASS offers two positioning services to its customers: Standard Accuracy Service (SAS) and High Accuracy Service (HAS). SAS is provided on both L1 and L2 channels, and has a clock rate of 0.511 MHz. HAS is also provided on both channels. It has a clock rate of 5.11 MHz. Usage of HAS requires permission from the Ministry of Defence of Russia. Binary Phase Shift Keying (BPSK) modulation is used for all available channels. Adoption of BPSK simplifies the code generation scheme. SAS can only be used by a receiver capable of dealing with FDMA signals. It is not possible for a CDMA-based receiver to take advantage of the GLONASS constellation.

The US has invited Russia to consider adopting CDMA for future GLONASS satellites. It is claimed that CDMA reduces the complexity of the receiver, increases the quality of the ranging signal, and improves the accuracy of carrier-phase based applications. In 2007, GLONASS designers undertook to consider switching to CDMA in the near future (Revnivykh, 2005). As of this writing, the Russian government has agreed to sponsor GLONASS modernization efforts where satellites would transmit using both CDMA and FDMA (Gibbons, 2008a).

2.1.5 Overview of Galileo

Galileo will be the European Union GNSS consisting of 30 satellites with an emphasis on civilian use. Galileo was borne out of a recognized need for Europe to have its own GNSS that will be under civilian control, and plans to offer services that are completely open to public use. Galileo will offer several levels of service, some of which require payment (ESA, 2007b). Galileo is also intended to stimulate Europe's economy. More information on Galileo's development plan can be found in Constantine (2008) or the European Space Agency's website (ESA, 2010).

Unlike GLONASS, the Galileo system will be directly inter-operable with GPS and GLONASS (ESA, 2007a). Galileo has adopted three E-band transmission channels with frequencies which are close to those of GPS. To avoid interference, Galileo has introduced a new type of signal known as Multiplexed Binary Offset Carrier (MBOC), which uses two multiplexed Composite Binary Offset Signals (CBOC), and is a more sophisticated version of Binary Phase Shift Keying (BPSK). CBOC consists of carrier and sub-carrier signals that allow a single-frequency channel to have two data sub-channels. Galileo will require at least a dual-frequency receiver, even for civilian applications. At the time of this writing, two Galileo test satellites (GIOVE A and B, respectively) have been successfully launched and are being tested (Tossaint et al., 2008; Gatti et al., 2008).

2.1.5.1 Galileo Channels and Associated Signals

Galileo will utilize three frequency channels in the E-band designated as E1 (1.57542 GHz), E5 (1.18668 GHz) and E6 (1.27875 GHz) for use in its five proposed services: open service (OS), commercial service (CS), safety of life (SoL), public regulated service (PRS) and search and rescue (SAR). The sub-carrier of the Galileo signal will allow each frequency channel to be split into two. The split signals in E5 have been designated with letters a and b. Signals E5a (1.17645 GHz) and E5b (1.20714 GHz) will be further subdivided into two more signals known as data and pilot signals. Data signal refers to a modulated navigation message similar to one transmitted by GPS satellites, and the pilot signal will act as a ranging signal (Hollreiser et al., 2007).

2.1.5.2 Galileo Binary Offset Carrier Signal

A spread-spectrum, pseudo-random noise signal distinct from the one used by GPS or GLONASS has been developed, which is called the Binary Offset Carrier (BOC). The BOC signal consists of four components: carrier wave, sub-carrier wave, pseudo-random noise spreading signal and data sequence. The sub-carrier and the PRN spreading signal are both binary signals. The role of the sub-carrier signal is to allow for multiple signals to be modulated on a single carrier wave. To generate the BOC signal, two variables m and n are needed. Thus, a unique BOC signal is specified by a combination of m and n terms (Avila-Rodriguez et al., 2007).

The two aforementioned parameters of the BOC signal are used to compute the frequencies of the sub-carrier signal and the PRN signal, respectively. These parameters are nominally integers; however, it is possible for them to be real numbers. The frequencies of the sub-carrier signal and the PRN spread-spectrum signal are defined as follows:

$$\begin{aligned} f_{SC} &= mf_0 \\ f_{PRN} &= nf_0 \end{aligned} \tag{2.2}$$

In Equation 2.2, f_0 is carrier frequency. The sub-carrier is a continuous rectangular waveform that alternates between 1 and -1. Each PRN chip lasts for one period of the sub-carrier signal, if the frequency of the PRN signal is exactly equal to the frequency of the sub-carrier. In other words, there are two sub-carrier half cycles for each PRN chip when $m = n$. The number of sub-carrier half cycles is known as the BOC order, described in Avila-Rodriguez et al. (2007) as follows:

$$N_{BOC} = 2\frac{m}{n} \tag{2.3}$$

The final BOC signal consists of the product of the carrier wave, the PRN signal, and the sub-carrier. Mathematically, the BOC signal as a function of m and n and can be represented as follows:

$$BOC(m,n,t) = SC\left(mf_0t\right) \cdot PRN\left(nf_0t\right) \cdot CW\left(f_0t\right) \tag{2.4}$$

In Equation 2.4, $PRN\left(nf_0t\right)$ is a time domain representation of the PRN signal at time t, $SC\left(mf_0t\right)$ is a time domain representation of the sub-carrier signal at time t, and $CW\left(f_0t\right)$ is a time domain representation of the carrier signal at time t. BOC signal components are illustrated in Figure 2.2, where numbers above the PRN signal indicate the sample code used (-1 1 1 -1 -1).

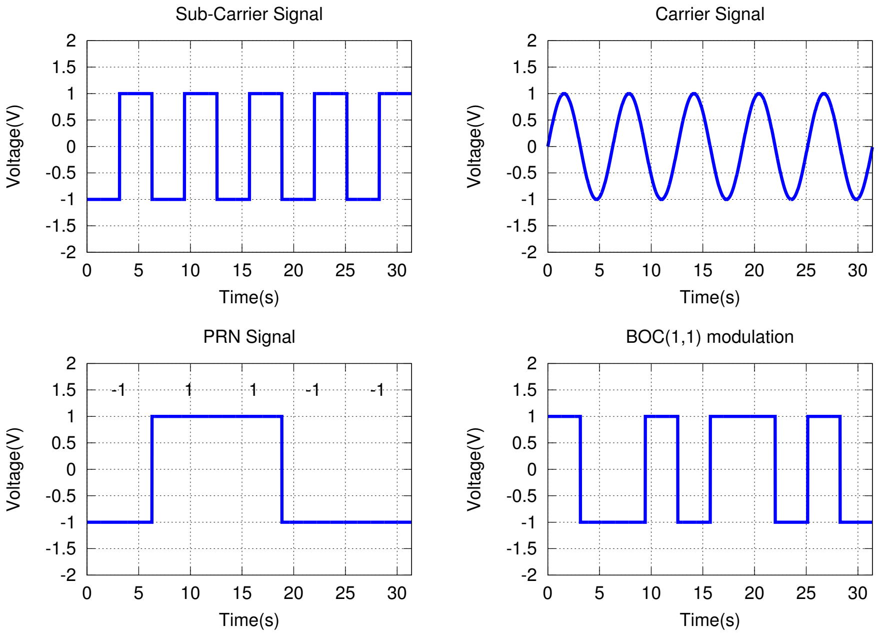


Figure 2.2: Structure of Binary Offset Carrier Signal

2.1.5.3 Galileo Multiplexed Binary Offset Carrier Signal

A more advanced form of BOC, MBOC, uses two CBOC signals summed together in a specific way. The CBOC signal is a weighted sum/difference of two BOC signals, and is represented by the previously defined coefficients *m* and *n* with an additional coefficient *k*. Mathematically, an MBOC signal can be represented as follows:

$$\begin{aligned} CBOC^{+}(m,n,k)\big|_t &= \sqrt{1-k}BOC(n,1)\big|_t + \sqrt{k}BOC(m,1) \\ CBOC^{-}(m,n,k)\big|_t &= \sqrt{1-k}BOC(n,1)\big|_t - \sqrt{k}BOC(m,1) \end{aligned} \tag{2.5}$$

In Equation 2.5, subscript t signifies that a function is evaluated at a discrete epoch t. An MBOC signal can be generated using two CBOC signals as follows:

$$MBOC(m,n,k)\big|_t = \frac{1}{\sqrt{2}}\left[CBOC^{+}(m,n,k)\big|_t - CBOC^{-}(m,n,k)\big|_t\right] \tag{2.6}$$

The choice of value k is arbitrary. For Galileo, $k = \frac{1}{11}$ (Avila-Rodriguez et al., 2007), which means that the overall contribution by the higher frequency components of the MBOC signal (namely CBOC(6,1)) are greatly reduced. Figure 2.3 illustrates the components of an MBOC signal.

2.1.5.4 Galileo Alternative Binary Offset Carrier Signal

The Alternative Binary Offset Carrier (AltBOC) is a modified version of the BOC in which a complex rectangular sub-carrier is used instead of the one used in BOC. Rather than multiplying the data and pilot signals by a rectangular sine wave, the signals are multiplied by a complex exponential signal as follows (Hein et al., 2002):

$$s_s(t) = s(t) \cdot sign\left(cos\left(2\pi f_s t\right) + j \cdot sign\left(sin\left(2\pi f_s t\right)\right)\right) \tag{2.7}$$

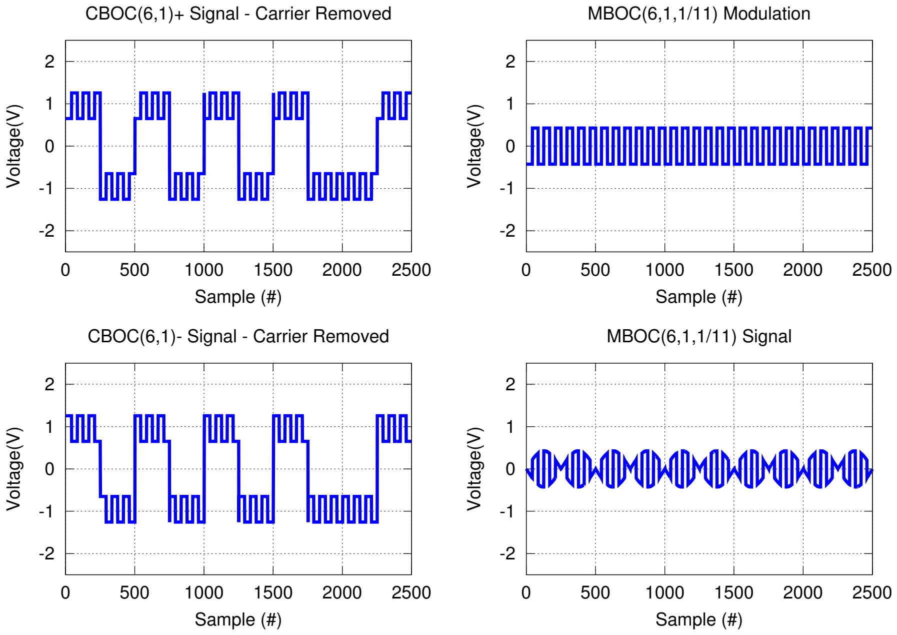


Figure 2.3: Structure of Multiplexed Binary Offset Carrier Signal

In Equation 2.7, $s(t)$ is the signal to be modulated, f_s is frequency offset of the signal relative to a reference signal, j is a complex number, and *sign* is a function returning a sign of its argument. In Galileo, s can be either E5a or E5b channel, while the reference signal is E5. Further information on this modulation can be found in Hein et al. (2002). The multiplication shown in Equation 2.7 allows for data modulation on both side lobes, while maintaining the simplicity of the BOC implementation. An AltBOC signal is described using the same coefficients m and n used to describe a BOC signal. The effective difference between BOC and AltBOC is that BOC is a BPSK signal (operating in real domain), while AltBOC is a Quadrature Phase Shift Keying (QPSK) signal. Advantages for the AltBOC signal are as follows (Hein et al., 2002):

1. Correlation losses are lower than those found with BOC signals,

2. Ability to transmit many side-lobes in a wide-band coherent signal gives rise to a

gain in precision,

3. Allows two services to exist on the same channel using the dual-band nature of the AltBOC signal (the primary service can use one band, while the secondary service can use both if necessary),

4. Payload baseband generator and amplifier/output multiplexer subsystems are simpler.

2.1.6 Overview of Compass

Compass is the Chinese satellite navigation system. According to plans, Compass will consist of 30 Medium Earth Orbit (MEO) and 5 geostationary satellites. Currently, two test MEO satellites have been launched (Grelier et al., 2007), with an additional test geostationary satellite launched on January 17, 2010 (Gibbons, 2010). The overall structure of the signals appears to resemble that of GPS and Galileo. Results confirm that early Compass signals employ CDMA and that they occupy the L-band (Grelier et al., 2007; Gao et al., 2007). China has reserved four frequency channels for Compass. They have the following frequencies and labels: 1.58974 GHz (E1), 1.56110 GHz (E2), 1.26852 GHz (E5b), and 1.20714 GHz (E6).

2.1.6.1 Compass Preliminary Signal Structure

Several techniques have been used to analyze the signal structure of Compass. Frequency domain representations of Compass signals have been used to establish the nature of the signals (Gao et al., 2007). These representations have identified the code modulation (such as BPSK, QPSK and so on), and help guide the demodulation process. Researchers were able to derive a PRN code generator for the E6 channel. It appears to be a Gold Code generated by two 11-stage shift registers with tapping polynomials $1+X+X^7+X^8+X^9+X^{10}+X^{11}$. The code is determined to be 2046 bits long, and has a period of 1 millisecond. Researchers also revealed that a secondary code used to modulate the E6 signal is a Neuman-Hoffman code with the following sequence: -1 -1 -1 -1 -1 1 -1 1 1 -1 1 -1 1 -1 -1 1 1 1 -1 (Gao et al., 2007). Signals appearing in other frequency channels are also under study. The E6 signal appears to be split into two parts, a head and a tail. Both parts are generated with 13-stage Gold Code generators, and having a chip length of 10,230 chips.

2.2 Overview of Point Positioning

Since positioning takes place in 3-dimensional space, at least 3 ranges are necessary to calculate a receiver's coordinates. Owing to the presence of numerous error sources, satellite ranges computed by a receiver are referred to as pseudoranges, since they are not true ranges. One of the most fundamental error sources (and the reason why only pseudoranges can be determined) has to do with the receiver clock. Satellite positioning depends on the synchronization of satellite and receiver clocks for accurate positioning. A mis-synchronization of a receiver's clock of a few microseconds can lead to range errors on the order of hundreds of metres. Satellite atomic clocks are also not perfectly accurate and tend to drift slowly. Such drifts and other types of deviations lead to errors in range estimates that must be corrected (Hofmann-Wellenhof et al., 2001). Satellite clock errors are modelled by various agencies, and corrections are also encoded into the satellite broadcast message. Receiver clock error is treated as unknown. Thus it is necessary to obtain pseudoranges from at least four satellites to compute one's 3D position. Besides receiver and satellite clock errors, there are other sources of error that must be taken into account.

A signal transmitted from a GNSS satellite passes through the atmosphere, causing it to be refracted. Two major layers of the atmosphere that cause measurable refraction are the ionosphere and troposphere. The ionosphere is a dispersive medium: the speed of signal propagation depends on frequency. The troposphere is a non-dispersive medium for frequencies below 30 GHz (which is the case for all GNSSs), so travel delay is frequency independent (Leick, 2004). Out of the two errors, ionospheric refraction typically makes a greater contribution to range error (typically between metres to decametres). If more than one frequency is available, the ionospheric effect can be eliminated or at least greatly reduced by taking advantaging of the so-called ionosphere-free combination (Kaplan and Hegarty, 2006). The tropospheric effect is most often modelled by two components: "dry" and "wet". The dry component (more accurately known as the hydrostatic component) accounts for about 90 percent of the effect, and is a result of neutral gaseous constituents of the tropospheric layer located at altitudes from zero to up to approximately 20 kilometres. The wet component is a result of water vapour in this region. It is possible to accurately model the dry component of the tropospheric delay; however, the wet component depends on irregular water vapour conditions, which are difficult to model accurately (Hofmann-Wellenhof et al., 2001).

Other error sources exist that further corrupt pseudorange measurements. The most common ones are multipath, solid Earth and ocean tides, relativistic effect, and hardware (receiver and satellite) biases. Some of these additional effects can be significant (such as relativistic effects and multipath), and some are small (e.g., hardware biases and tides). Depending on positioning accuracy requirements, these errors can either be ignored or mitigated in some fashion such as modelling. Not all error sources can be modelled effectively. For instance, multipath is dependent on the reflective environment in the vicinity of the receiving antenna. Regardless of how these errors are treated, it is not possible to convert pseudorange exactly into geometrical range, since there are always sources of error present.

GPS receivers also track the carrier phase signals being modulated by the pseudoranges. As with code derived pseudorange, it is possible to determine a pseudorange from carrier phase. This is known as phase pseudorange. Unlike the code pseudorange, the phase pseudorange does not directly correspond to the distance from satellite to receiver, since phase pseudorange depends on counting phase cycles. Although modern receivers can count phase cycles much more accurately than they can align PRN codes, there is an unknown number of cycles that have occurred prior to the start of signal tracking. This is the issue of integer ambiguity. Receivers occasionally reset their phase count in unpredictable ways (cycle slips), which adds to the problem. Carrier-phase processing is used to achieve centimetre or better positioning by resolving these cycle ambiguities. Part of the reason for this degree of success lies in the ability to reduce (by approximately a factor of 100) observable noise as compared to code pseudorange noise. Cycle ambiguities can be either estimated as real numbers or recovered as integers. This process is known as ambiguity resolution (Hofmann-Wellenhof et al., 2001). The general observation model for code and phase pseudoranges is given in Equation 2.8 (based on terminology presented in Hofmann-Wellenhof et al. (2001); Leick (2004); Kaplan and Hegarty (2006); Xu (2007)).

$$\begin{aligned} P &= \rho + c\,(\delta_r - \delta^s) + \Delta Iono + \Delta Tropo - \Delta Relv + \\ &\quad + \Delta M_C + \Delta H_C + \Delta D + \Delta O_C + \delta\varepsilon_C \\ L &= \frac{1}{\lambda_L}\Big[\rho + c\,(\delta_r - \delta^s) + \lambda_L N - \Delta Iono + \Delta Tropo - \Delta Relv + \\ &\quad + \Delta M_L + \Delta H_L + \Delta D + \Delta O_L + \delta\varepsilon_L\Big] \end{aligned} \tag{2.8}$$

Parameters in Equation 2.8 are defined as follows:

- P, L are code (in metres) and phase (in cycles) observables, respectively,

- c is speed of light (in metres per second),

- δ_r, δ^s are receiver and satellite clock biases, respectively (in seconds),

- N is integer carrier-phase ambiguity (unitless),

- $\Delta Iono$ is ionospheric error (in metres),

- $\Delta Tropo$ is tropospheric error (in metres),

- $\Delta Relv$ is relativistic error (in metres),

- $\Delta M_C, \Delta M_L$ are code and phase multipath errors, respectively (in metres),

- ΔD is tidal (solid Earth and ocean loading) errors (in metres),

- $\Delta O_C, \Delta O_L$ are other code and phase errors, respectively (in metres),

- $\delta\varepsilon_C, \delta\varepsilon_L$ are code and phase observable noise, respectively (in metres),

- λ_L is wavelength of a particular GNSS carrier wave from which phase pseudorange is formed (in metres).

Code and phase observables defined in Equation 2.8 are those extracted from a single receiver. In basic positioning, these observables are used to compute receiver position. More advanced positioning applications can use observables from other receivers either in real-time or post-processing to enhance position accuracy. For example, in Real Time Kinematic (RTK) positioning, data between two receivers is differenced to remove various error sources (such as hardware biases and atmospheric effects, if the two stations are close to each other) to achieve centimetre positioning in real-time (Lee and Ge, 2006). Other advanced techniques, such as Precise Point Positioning (PPP), use various corrections to original data (e.g., satellite orbits and clocks) to improve position accuracy. With PPP it is possible to achieve centimetre (static) positioning using raw or undifferenced data. PPP has gained popularity because it does not require reference stations. Moreover, it is possible to use PPP method in real-time even with a single frequency receiver (Gao et al., 2006). In some application such as real-time spacecraft positioning it may not be possible to utilize PPP effectively due to an inability of acquiring the necessary corrections. For such applications, the standard point positioning technique is used.

2.3 Satellite Positioning with GNSS Satellites

Traditionally, LEO tracking has been from the ground, and these measurements are combined with a dynamical model for positioning or orbit determination. The cost of operating tracking stations as well as their availability and accuracy make GPS-based positioning an attractive alternative. GPS-based LEO positioning accuracy has been demonstrated to be on-par with or even better than traditional LEO positioning (Bisnath and Langley, 2001). Various techniques exist for LEO positioning, with most incorporate models of spacecraft motion into normal equations to enhance position accuracy (Švehla and Rothacher, 2003). It is also possible to position without any knowledge of orbital dynamics or using a single-frequency receiver (Hwang and Born, 2005). Although LEO spaceborne receivers are relatively common, not many receivers are installed on Highly Elliptical Orbit (HEO) spacecraft. Balbach et al. (1998) showed that it is possible to track GPS satellites even on HEO spacecraft. To evaluate LEO and HEO positioning, the point positioning technique is used.

Chapter 3

Multi-GNSS Observables Simulator (MGOS)

This chapter will begin with a discussion of various GNSS simulators and how the evaluation of these simulators led to the development of the in-house simulator named Multi-GNSS Observables Simulator (MGOS). It was deemed necessary to develop a multi-GNSS simulator from scratch to not only control how the simulator will operate, but also what features the simulator will have. And it was not possible to know ahead of time how rich the feature set of MGOS must be in order to complete the main studies of this thesis. MGOS has gone through many changes and its feature set has been expanded to suit the needs of this work. The development of MGOS has also been influenced by a desire to build a simulator suitable for a variety of applications, where multi-GNSS data simulation is relevant. The chapter is divided into three parts: the first part introduces MGOS and its design and major features; the second part discusses how various error sources are simulated within MGOS; and the third and final part presents and discussed the results for the tests used to evaluate the simulation capabilities of MGOS.

3.1 Overview of GNSS Simulators

Signals or observables can be simulated using either hardware or software. A hardware approach requires specialized equipment that simulates signals, signal propagation errors, and hardware related losses. Such devices tend to be very expensive. Examples are Spirent's GSS6700 (Spirent Communications, 2010) and CAST's 1000 simulator (CAST Navigation, LLC, 2007). Hardware simulators operate in the signal domain, which means that they simulate GNSS signals rather than observables. Simulators which generate observables are said to operate in the observable domain. Such simulators are not required to simulate actual satellite signals, nor to produce observables that are prone to error sources. Software simulators are generally less expensive than hardware simulators, and can be upgraded in a more flexible way than hardware simulators. Some software simulators have a designated hardware component that allows them to perform operations of a hardware simulator. One example is a GPS signal simulator developed by NAVSYS that uses external acquisition cards to acquire real-world signals and process them using Matlab toolboxes developed by NAVSYS (Brown et al., 2000). Other software simulators such as GPSoft's SATNAV toolbox or Accord Software's GPS lab generate observables using compiled or interpreted computer codes (GPSoft LLC, 2006; Accord Software & Systems Inc., 2007).

This research deals with raw multi-GNSS observables. A hardware simulator can feed radio frequency signals to a receiver to generate limited observables, but as stated hardware simulators are very costly. Only modestly priced software packages are open to consideration in this research project. A candidate software package must be able to simulate observables from multiple GNSSs. After an initial search, SATNAV by GPSoft was acquired for study (GPSoft LLC, 2006). The SATNAV toolbox is a set of MATLAB© scripts used for generating GPS and GPS-like observables. These observables can be exploited to study potential improvements in positioning and navigation solutions arising from multi-constellation position systems as compared to a single-constellation position system. Using satellite ephemeris information, SATNAV can simulate an entire GPS/Galileo/GLONASS as well as Geostationary constellations for the purpose of generating GNSS observables at any user location. Using satellite geometries as well as user-defined positions, SATNAV can generate true satellite range measurements, and optionally simulate measurement noise sources present within real GNSS measurements (GPSoft LLC, 2003).

After a thorough evaluation it was concluded that although a comprehensive tool, SATNAV does not have all the features necessary to accomplish some of the goals set out for the research work reported here. The toolbox has adopted all of the standard atmospheric error models, such as the Klobuchar model for ionospheric delay, and modified Hopfield model for tropospheric delay, in order to generate code-based pseudorange observables. However, ionospheric and tropospheric error generation routines do not incorporate any stochastic and bias modelling. If such simulated measurements were to be processed by an external program, the computed ionospheric and tropospheric delay values would be completely cancelled, resulting in unrealistically accurate solutions. This drawback in the atmospheric error generation scheme adopted by SATNAV forces the user to rely solely on processing routines provided by SATNAV. This situation does not allow for independent evaluation or corroboration.

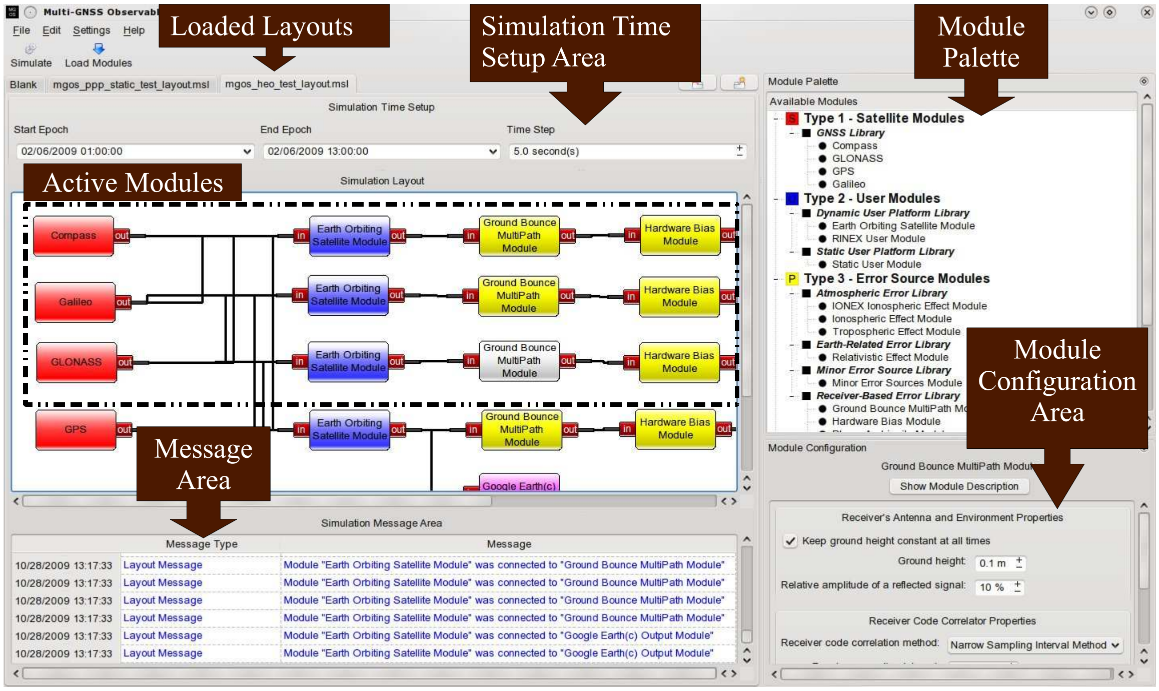
Multipath errors are modelled using an autoregressive model which does not take into account receiver environment. This may be acceptable for simple simulations; however, for more complex (and realistic) simulations, SATNAV capabilities are inadequate. Moreover, SATNAV does not model hardware biases and tidal errors. Finally, SATNAV lacks an ability to export data in RINEX format. These shortcomings have lead to the development of the MGOS.

MGOS is designed to be flexible and easy to use. Since its conception, MGOS has been envisioned to incorporate a Graphical User Interface (GUI) for several reasons. First, complex simulation scenarios will need to be constructed to obtain the necessary data for this research. To describe these complex simulations in an intuitive way, a GUI will be helpful. Second, simulation parameters can be more readily adjusted using a GUI than a text-based interface. Since a user can easily alter the behaviour of MGOS without dealing with computer code, many useful and interesting features have been incorporated into MGOS' GUI over its development. MGOS has a modular design. Its simulation components can be easily modified and extended, thus simulation schemes constructed using MGOS consist of dynamically loadable modules that can be directly manipulated and reconfigured by a user.

MGOS' capabilities and operation appear from second hand accounts to resemble those of the BaiCES software simulator designed by Institute of Geodesy and Navigation (2008). Similarities to BaiCES are purely coincidental, as MGOS has arisen from an evaluation of the SATNAV alone. Since it has not been possible to acquire BaiCES as an aid to this research project, it is not possible to verify to what extent MGOS resembles BaiCES. MGOS can be regarded as an inexpensive substitute for a costly multi-GNSS simulator, as it is completely open-source and free for everyone to use.

3.2 MGOS Overview

MGOS is a modular graphical application written in C++ using the Qt Graphical User Interface application framework. The Qt framework is chosen as it supports multiple computer platforms (including Linux, Mac OS X and Windows), and has an extensive library of C++ classes (Qt Development Frameworks, 2009). The MGOS GUI has been designed to allow the user to easily create simulation scenarios (called simulation layouts) within MGOS. A simulation layout is composed of interconnected graphical components, which will be referred to as MGOS modules. These modules, as well as MGOS itself, have been created over a period of one and half years and consist of approximately 56,000 lines of code. Figure 3.1 shows a typical MGOS scenario incorporating several modules and major MGOS GUI components.


Loaded Layouts
Simulation Time Setup Area
Module Palette
Active Modules
Message Area
Module Configuration Area
Multi-GNSS Observabl
File Edit Settings Help
Simulate Load Modules
Blank
mgos_ppp_static_test_layout.msl
mgos_heo_test_layout.msl
Simulation Time Setup
Start Epoch
End Epoch
Time Step
Simulation Layout
Compass
Galileo
GLONASS
GPS
Earth Orbiting Satellite Module
Ground Bounce MultiPath Module
Hardware Bias Module
Google Earth(c)
in
out
Module Palette
Available Modules
S Type 1 - Satellite Modules
GNSS Library
Compass
GLONASS
GPS
Galileo
Type 2 - User Modules
Dynamic User Platform Library
Earth Orbiting Satellite Module
RINEX User Module
Static User Platform Library
Static User Module
P Type 3 - Error Source Modules
Atmospheric Error Library
IONEX Ionospheric Effect Module
Ionospheric Effect Module
Tropospheric Effect Module
Earth-Related Error Library
Relativistic Effect Module
Minor Error Source Library
Minor Error Sources Module
Receiver-Based Error Library
Ground Bounce MultiPath Mo
Hardware Bias Module
Module Configuration
Ground Bounce MultiPath Modul
Show Module Description
Simulation Message Area

Start Epoch: 02/06/2009 01:00:00
End Epoch: 02/06/2009 13:00:00
Time Step: 5.0 second(s)

	Message Type	Message
10/28/2009 13:17:33	Layout Message	Module "Earth Orbiting Satellite Module" was connected to "Ground Bounce MultiPath Module"
10/28/2009 13:17:33	Layout Message	Module "Earth Orbiting Satellite Module" was connected to "Ground Bounce MultiPath Module"
10/28/2009 13:17:33	Layout Message	Module "Earth Orbiting Satellite Module" was connected to "Ground Bounce MultiPath Module"
10/28/2009 13:17:33	Layout Message	Module "Earth Orbiting Satellite Module" was connected to "Ground Bounce MultiPath Module"
10/28/2009 13:17:33	Layout Message	Module "Earth Orbiting Satellite Module" was connected to "Google Earth(c) Output Module"
10/28/2009 13:17:33	Layout Message	Module "Earth Orbiting Satellite Module" was connected to "Google Earth(c) Output Module"

Receiver's Antenna and Environment Properties

☑ Keep ground height constant at all times

Ground height: 0.1 m

Relative amplitude of a reflected signal: 10 %

Receiver Code Correlator Properties

Receiver code correlation method: Narrow Sampling Interval Method

Figure 3.1: MGOS Graphical User Interface

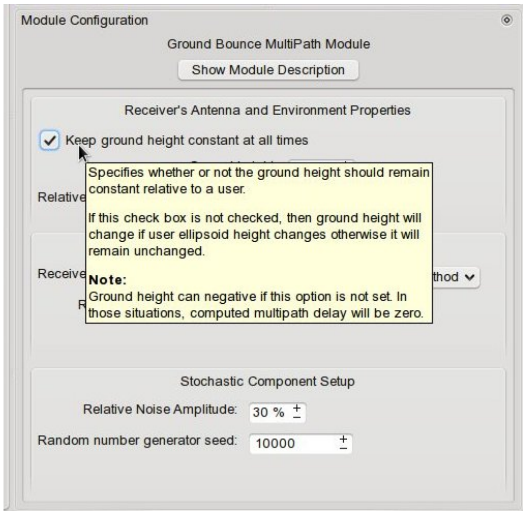
The modules can generate or modify GNSS observables. The MGOS module library consists of satellite, user, error source, and data visualization modules. Twenty modules have been developed so far. Each module belongs to a group (such as a user module group or satellite module group). Current modules are presented in Table 3.1.

Major Group	Minor Group	Module Name
Satellite Library	GNSS Library	GPS
		GLONASS
		Galileo
		Compass
User Library	Dynamic User Platform Library	Earth Orbiting Spacecraft Module
		RINEXUser Module
	Static User Platform Library	Static User Module
Error Source Library	Atmospheric Error Library	Tropospheric Module
		Ionospheric Module
		IONEX Ionospheric Module
	Earth-Related Error Library	Relativistic Effect Module
	Minor Error Source Library	Minor Error Sources Module
	Receiver-Based Error Library	Ground Bounce Multipath Module
		Hardware Bias Module
		Phase Ambiguity Module
		Simple Multipath Effect Module
Output Module Library	File Generating Module Library	Google Earth Output Module
		RINEX Output Module
	MGOS Evaluation Module Library	Multi-GNSS Processor Module
		Observables Noise Module

Table 3.1: Listing of all available MGOS modules by group and sub-groups

Above modules will be described in some detail later. A description of a module can be obtained via the GUI by clicking the “Show Module Description” button in the module configuration area (see Figure 3.1). Most modules also show messages (tooltips) in their configuration widgets when the cursor is placed over certain parameters, as shown in Figure 3.2.

MGOS modules operate in the observation domain rather than the signal domain, in order to provide flexible ways of adding more features to the application. Simulated observables can be used to assess future multi-constellation performance, and aid in the de-



Module Configuration

Ground Bounce MultiPath Module

Show Module Description

Receiver's Antenna and Environment Properties

☑ Keep ground height constant at all times

Specifies whether or not the ground height should remain constant relative to a user.

If this check box is not checked, then ground height will change if user ellipsoid height changes otherwise it will remain unchanged.

Note:
Ground height can negative if this option is not set. In those situations, computed multipath delay will be zero.

Stochastic Component Setup

Relative Noise Amplitude: 30 %

Random number generator seed: 10000

Figure 3.2: Example of a hint message shown by components of a module configuration widget

velopment of enhanced processing algorithms. The observation domain also simplifies the representation of the majority of error sources to be defined in this chapter. Simulated observables can be output in the RINEX format (using the RINEX Output Module), visualized (using Observables Noise Module), or processed (using the Multi-GNSS Processor Module). It is also possible to generate a Keyhole Markup Language (KML) file to be viewed by Google Earth©, and to generate animated paths for moving receiver platforms.

MGOS allows a user to save layouts to memory and re-load them later. All module parameters are saved with the layout. Module parameters can be copied from module to module via a keyboard shortcut. For convenience, several modules that require external data (for example GNSS modules that require precise ephemeris files) can be set to download the necessary data from the International GNSS Service (IGS) website. MGOS also features a messaging area which notifies a user when a simulation starts and ends, reports simulation errors, and reports other relevant information that occurs while the simulation is running (such as when an external file being downloaded). For ease of interpretation, the message area breaks various reported messages down into categories, colour codes each message, and time tags each message.

3.3 MGOS Design

The MGOS GUI is built from simple widgets such as check-boxes, combo-boxes, buttons, etc. The GUI consists of three major components: the simulation layout, module palette, and module configuration. The simulation layout is the largest component of the GUI. It is composed of a graphical schematic widget, a messaging area, and a simulation time setup area (see Figure 3.1). Every new simulation layout generates its own messaging area as well as simulation configuration area. The simulation time setup area has been designed to make sure that only logical simulation spans can be set. Parameters from the simulation time setup area are stored when a user performs saves a layout. The message area keeps track of current messages. Each entry in the message area is time stamped and categorized (colour coded). There are five message groups identified by the message area: normal layout messages, layout error messages, module messages, normal simulation messages (e.g., indication of simulation start or end epochs), and simulation error messages.

MGOS modules themselves are composed of three components: a layout component, a configuration widget, and an engine. A graphical component (shown in Figure 3.1) is composed of a rectangular body containing the module's name, and one or two ports. Satellite modules have only the output port available as they do not accept any information from other modules. Output modules do not generate any new information, hence they only have the input port available. Other module types have both the input and the output ports available. The engine of a module is defined through its interface. Each module type has a base interface, defined as a C++ class used by MGOS to communicate with the module.

To create a module, a developer needs to extend the base class of a particular module type. Some functions in the base classes have default behaviours, others need to be defined within a new module. Each base class also performs tasks that are common to its type (e.g., a user module base class will generate initial observables for all active satellites automatically). Moreover, base classes automatically call all necessary initialization routines needed for the proper operation of the derived modules.

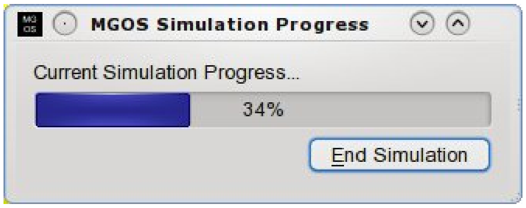
Each MGOS modules must provide a configuration widget. MGOS has a mechanism to export and import module configuration widget parameters. Unfortunately, it is not possible for MGOS to deduce a module's simulation parameters from its widget alone. This means that developers must take care of specifying parameters. Parameters can also be copied from one module to another provided these modules are compatible with each other. This is useful when creating module chains that have the same modules. Data flow from one module to the next is represented by a wire. A wire is a graphical object which knows which modules are connected to it. This information is used by a simulation layout to register and unregister data connections between modules. Through bending, a wire can join modules horizontally or vertically, and modules can be connected in chains. Some modules support multiple input or output connections. To form a proper simulation scheme, most modules require at least one input and output connection. Moreover, at least one satellite, user and output module must be involved in order to perform a simulation. Table 3.2 summarizes the minimum and maximum number of input/output connections supported by each module type.

Module Type	Module Requirement	Input Connections		Output Connections	
		Minimum	Maximum	Minimum	Maximum
Satellite	Yes	N/A	N/A	1	Unbound
User	Yes	1	Unbound	1	Unbound
Error Source	No	1	1	1	1
Output	Yes	1	1	N/A	N/A

Table 3.2: Connection scheme for Multi-GNSS Observables Simulator modules in a simulation scheme

With multiple input or output options for some modules, it is possible to form parallel chains. In order to protect data passed from one module to the next, MGOS never executes a simulation in parallel but in sequence, one chain at a time. This automates execution of MGOS module chains, so there is no need to manually execute each chain, and since each chain is executed in sequence regardless of how it is presented in the layout, modules cannot interfere with each other during a run. An MGOS simulation runs in a dedicated process or a thread. This prevents MGOS from becoming unresponsive during a run, and it allows several important features to function. The first feature is the ability of a simulation layout area to receive messages from modules. The second feature allows a user to stop (or forcefully terminate) a running simulation. The third and final feature that can only be implemented if simulation takes place in a dedicated process, is an ongoing report of simulation progress. Whenever a user initiates a simulation, an MGOS Simulation Progress widget appears. This widget (shown in Figure 3.3) can be used to both track the progress of a simulation run or stop it.

MGOS simulations are executed epoch by epoch. Epoch data consist of entries that allow MGOS modules to acquire all necessary information they require to operate. The entries reside in a data container labelled MGOSSimDataStructure, which is passed from module to module in either a read-only or modifiable state. There are in total 9 data en-



Current Simulation Progress...

Figure 3.3: MGOS simulation progress and control widget

tries which can be used to store and extract the following information: receiver spatial and clock information; receiver name, antenna type and other miscellaneous information; satellite spatial and clock information and clock bias values; satellite channel names and their frequencies; observables; GNSS coordinate frame; information on GNSS; names of data channels and their PRN codes; and chipping rates of data channel PRN codes.

MGOS modules can extract the above information as necessary without a tedious "unwrapping" of MGOSSimDataStructure, but only with a few function calls to MGOSSimDataStructure. For some modules (such as the error source modules or satellite modules), base classes perform all the necessary tasks of populating MGOSSimDataStructure, and only interrogate relevant modules when specific information is needed. For example, satellite modules are only asked to provide information for a single satellite rather than all satellites at the same time. Similarly, derived error source modules need only to modify one observable per function call rather than all of them at the same time, greatly simplifying error source module development. Unfortunately, simplifications made to satellite modules and error source modules mentioned above cannot be applied to output modules. Output modules may need to process the entire MGOSSimDataStructure, so it is handed directly without any prior unwrapping. All MGOS output modules store epoch data in memory until the end of a simulation run. Output modules are responsible to output data in some way, and data can be visualized in a form of a plot (or other graphical representation), output to a file, or both.

A simplified schematic depicting the interaction among the user, MGOS GUI, and the active modules, is shown in Figure 3.4. Arrows indicate direction of data flow or control. Solid lines represent user related interactions. Dashed lines represent communication among modules.

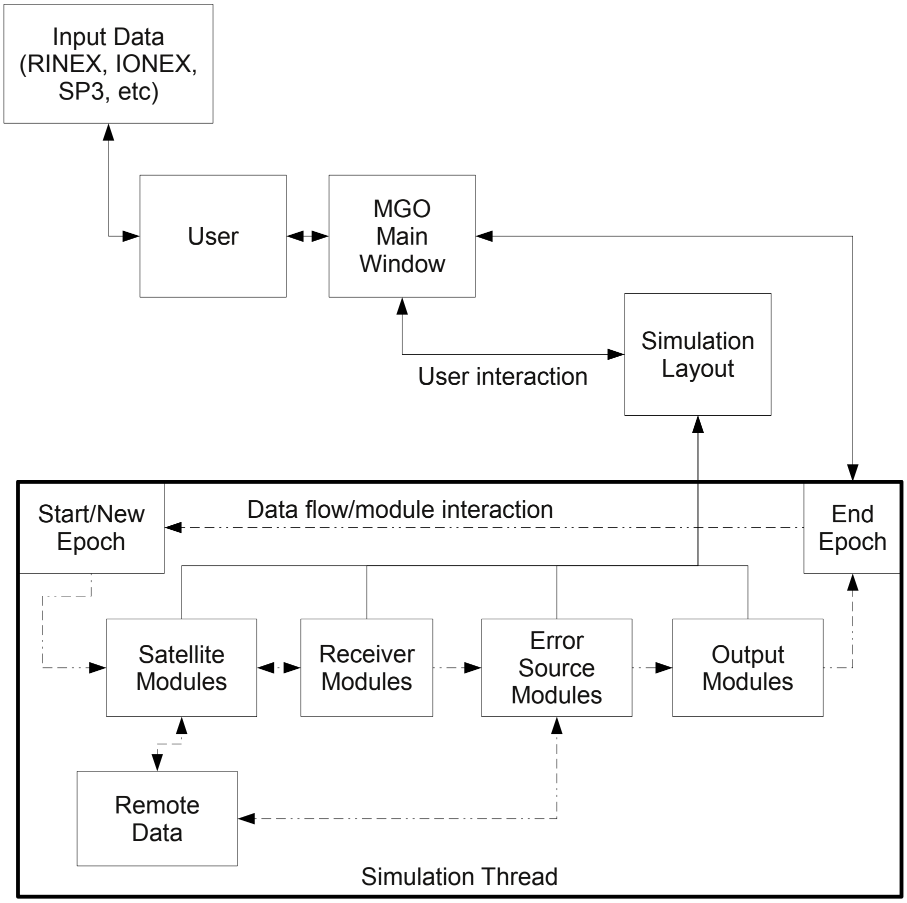


Figure 3.4: Depiction of primary MGOS components and how they interact with each other during a simulation run

3.4 GNSS Satellite Modules

In this section, a brief overview of GNSS modules will be given. Most GNSS modules are based on the GPS module and thus have many common components. This is especially true for Galileo and Compass GNSSs, since they utilize CDMA. As GLONASS uses FDMA, its module structure is slightly different.

3.4.1 GPS Module Implementation

Satellite coordinates are generated using ideal elliptical orbits described by the six Keplerian orbital elements: right ascension of ascending node, argument of perigee, inclination, eccentricity, semi-major axis, and mean anomaly. Coordinates are computed by evolving satellite motion from a reference epoch (i.e., time of ephemeris, or TOE) to the required simulation epoch. This module, by default, assumes that all satellites move in perfectly elliptical orbits. In reality, satellites do not travel in such ideal orbits due to various forces acting on them. In order to obtain an accurate estimate of satellite position, numerical integration might be required which can be very time consuming. This module provides a means to include 4th order gravitational perturbations using a Runge-Kutta-Fehlberg integrator where high accuracy is warranted.

The reference Keplerian orbital elements for all GPS satellites supported by this module are stored in a 2D array known as the internal ephemeris. All values stored in the internal ephemeris are hard coded. These values were obtained by external means and entered manually into the internal ephemeris. This file cannot be modified once created, and is shared by all instances of the module. Such a scheme also saves computer memory. The internal ephemeris also stores satellite clock bias coefficients used to estimate the satellite clock bias. The latter bias is estimated using the following equation:

$$\delta C = A0 + A1 \times (T - TOE) + A2 \times (T - TOE)^2 \tag{3.1}$$

In Equation 3.1, $A0$, $A1$, and $A2$ are satellite clock bias coefficients, T is the simulation epoch, and TOE is the reference epoch. Satellite clock bias calculated using the above equation becomes part of the output generated by this module. It is stored in the MGOSSpatialClockEntry data structure, which is part of MGOS' data structure library. The basic computation method described in the previous two paragraphs for satellite coordinates and clock bias is used whenever no external data sources are available (which is the default behaviour). In case external satellite coordinate and clock bias data are available, this method is not used.

This module is capable of extracting satellite clock bias information and coordinate information from SP3 and clock RINEX files versions 2.0 to 3.0 provided by various GNSS agencies (such as the International GNSS Service (IGS)). SP3 and clock RINEX files are parsed and processed by external widgets, which GPS integrates within its configuration widget. Since these widgets deal with real data, it is possible to run into issues using the parsed data. In particular, it may not be possible to perform simulations longer than the span of obtained data. In this case, a user can choose either to use a default method described previously or stop a simulation run.

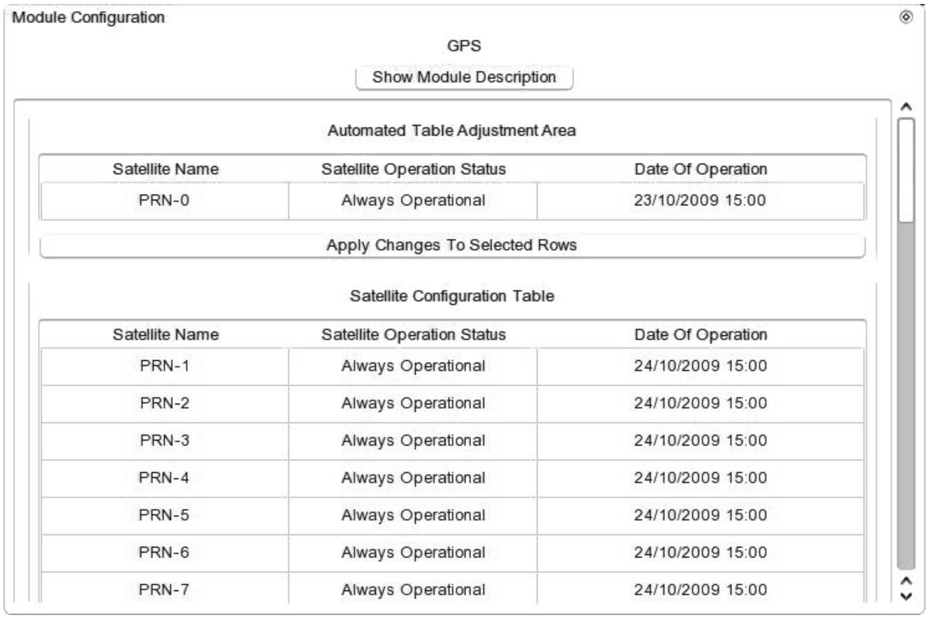
To minimize this problem, widgets can download all necessary files from the IGS automatically (provided that an Internet connection is available). The SP3 parsing widget also has a built-in component that allows a user to enter satellite phase centre offsets necessary for translating SP3 derived coordinates from satellite's centre to antenna phase centre. These corrections are provided with ANTEX files which can be downloaded automatically by the widget. Once all satellite coordinates and clock biases have been computed, the GPS module prepares information on relevant channels used by its satellites.

Each available GPS satellite can transmit up to three frequency channels. Each GPS signal (frequency channel) can have a number of modulation codes that are used to derived pseudorange and phase measurements. Each satellite will have a separate entry for its channel frequency in order to accommodate FDMA systems such as GLONASS. The number of channels that each satellite can transmit, frequencies of each channel and modulation codes each channel carries, are described in Table 3.3.

Channel	Frequency (MHz)	PRN codes
L1	1575.42	C/A and P
L2	1227.60	C/A, P and LC
L5	1176.45	I,Q and I+Q

Table 3.3: Summary of available GPS data channels and PRN codes

The LC code in Table 3.3 represents the L2C observable in the RINEX 3.00 file format. I, Q and I+Q codes are extracted from RINEX 3.00 file standard (Gurtner and Estey, 2006). Each modulation code has its own chipping rate. The chipping rates for each modulation code are as follows: C/A-code chipping rate is 1.023 MHz, and P, LC, I, Q and I+Q codes have a chipping rate of 10.23 MHz. Also, this module will not generate LC, I,Q and I+Q codes unless an appropriate number of frequency channels is selected. Along with an ability to select channel and code combinations, a user can also select which satellites will become part of a simulation run. By default, all available satellites are included. Satellite can either be removed from a simulation completely, or set to appear at certain times. This is useful in constructing complex simulations where certain satellites may not be immediately "healthy" to use. Figure 3.5 shows how satellites are configured within the GPS module's configuration widget.



Automated Table Adjustment Area

Satellite Name	Satellite Operation Status	Date Of Operation
PRN-0	Always Operational	23/10/2009 15:00

Satellite Configuration Table

Satellite Name	Satellite Operation Status	Date Of Operation
PRN-1	Always Operational	24/10/2009 15:00
PRN-2	Always Operational	24/10/2009 15:00
PRN-3	Always Operational	24/10/2009 15:00
PRN-4	Always Operational	24/10/2009 15:00
PRN-5	Always Operational	24/10/2009 15:00
PRN-6	Always Operational	24/10/2009 15:00
PRN-7	Always Operational	24/10/2009 15:00

Figure 3.5: GPS satellite selection table widget

Satellite selection is automatically updated when a local SP3 file has been set. This feature allows a user to dynamically extend the GPS module's definition of the GPS constellation without a need to obtain an updated copy of the module. In case a SP3 file is downloaded automatically, it will not be possible for a user to adjust the table during a simulation run. Nevertheless, it is possible to control satellite visibility even without adjusting the satellite configuration table. The GPS module also supports an ability to export coordinates and clock biases of all active satellite into a SP3 file. A file is generated in 24 hour increments in a SP3-C format.

3.4.2 GLONASS Module

The GLONASS module shares the same features available in the GPS module. Since GLONASS employs FDMA, each of its satellites broadcasts a distinct frequency on each channel. Frequencies are assigned using a look-up table compiled using information available on the Russian Space Agency's website (Information-Analytical Centre, 2006). This module only offers C/A- and P-codes on L1 and L2 channels. C/A-code has a chipping rate of 0.511 MHz while P-code has a chipping rate of 5.11 MHz. As with the GPS module, SP3 and clock RINEX input are available. Only GLONASS SP3 files are accepted by this module. In case of clock RINEX files, it is possible to use GPS files as well as GLONASS files.

GLONASS satellite coordinates are defined in a way very similar to that used in the GPS module. An internal ephemeris is generated from a reference SP3 file and hard-coded into the GLONASS module. The internal ephemeris is updated periodically in order to make sure it reflects the most recent changes to the GLONASS constellation. In most cases, it is better to use the SP3-C input feature to dynamically extend the GLONASS module constellation defined by this module, so as to have the most realistic representation of GLONASS as possible.

3.4.3 Galileo and Compass Modules

Galileo and Compass modules share the same features that are available in the GPS module. Since Galileo and Compass will use CDMA, they are more similar to GPS than GLONASS. That said, channel and code definitions in these two systems are different from that in GPS. Galileo's channel and code entries have been modelled after RINEX 3.00 format (Gurtner and Estey, 2006). Since little is known about Compass GNSS, it is assumed to be similar to Galileo GNSS. To simplify the implementation of the Compass module, Compass' geostationary satellites are not simulated. Galileo's and Compass' channel and code definitions are given in Table 3.4.

I, Q, and I+Q codes have chipping rates of 10.23 MHz, 2.5575 MHz and 1.23 MHz,

GNSS	Specification				
Galileo	Channel	E1	E5	E5a+E5b	E6
	Codes	A,B,C, B+C, A+B+C	I,Q, I+Q	I,Q, I+Q	A,B,C
	Frequency (MHz)	1575.42	1186.68	1176.45/1207.14	1278.75
Compass	Channel	E1	E2	E5B	E6
	Codes	A,B,C, B+C, A+B+C	I,Q,I+Q	I,Q,I+Q	A,B,C
	Frequency (MHz)	1589.74	1561.10	1268.52	1207.14

Table 3.4: Channel and code definitions for Galileo and Compass modules

respectively. A, B and C codes chipping rates are 10.23 MHz, 2.5575 MHz and 1.23 MHz, respectively. B+C and A+B+C codes chipping rates are assumed to be 2.5575 MHz and 10.23 MHz, respectively. The SP3 input features found in these two modules only support their respective GNSSs. The clock RINEX input feature is available only in the GPS and GLONASS modules. Coordinates and clock values generated for Galileo and Compass GNSSs are very approximate. All entries are based on the best knowledge of plans for these GNSSs at the time of this writing. In future releases of MGOS, these entries may need to be updated.

3.5 Simulation of Error Sources

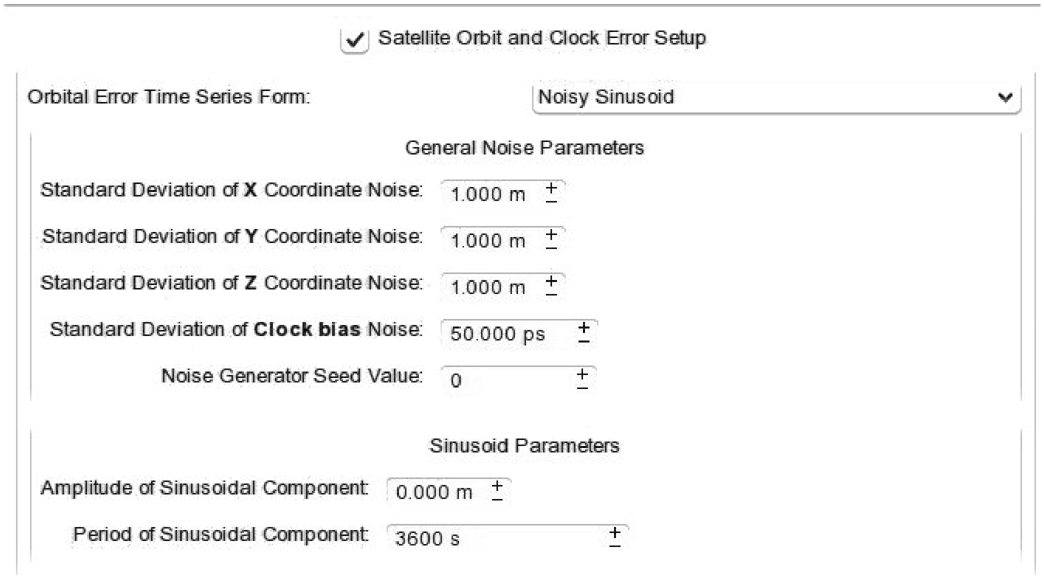
In this section, the MGOS module responsible for simulating various error sources such as satellite orbit and clock errors, multipath errors are discussed. Whenever appropriate, sample module output will be shown.

3.5.1 Satellite Orbital and Clock Errors

Since the broadcast ephemeris is used to determine satellite positions, orbital and clock errors must be simulated. Unfortunately, orbital errors are difficult to simulate. Various efforts have been made to evaluate GPS broadcast ephemeris quality. It appears that GPS orbital errors vary in quasi-sinusoidal fashion during the course of a day (Langley et al., 2000; Warren and Raquet, 2003). This is a crude approximation as actual orbital errors behave in a very complex manner. What further complicates orbital error simulation is the fact that GLONASS' broadcast ephemeris transmits satellite position and velocity values at ephemeris epoch, which require manual integration. Thus GLONASS orbital errors depend on the chosen integration algorithm and choice of time step (Stewart and Tsakiri, 1998).

It is beyond the scope of this research to attempt to very accurately simulate orbital errors, hence the following approximations are made. Orbit and clock errors are simulated as either a sinusoid with an added Gaussian noise component or just as Gaussian noise. A user can specify standard deviations for each satellite coordinate and clock error separately. For situations where a sinusoidal error structure is used, a user can set its amplitude and period. The configuration widget for orbit and clock errors is shown in Figure 3.6.

Orbital and clock errors are generated with a random number generator seeded according to the user's selection. In case a sinusoidal orbital error is specified, an additional randomly chosen phase offset is generated at the start of every simulation run. This phase offset is generated between 0 and 2π, making it different for each satellite. The complete



☑ Satellite Orbit and Clock Error Setup

Orbital Error Time Series Form: Noisy Sinusoid

General Noise Parameters

Standard Deviation of **X** Coordinate Noise: 1.000 m

Standard Deviation of **Y** Coordinate Noise: 1.000 m

Standard Deviation of **Z** Coordinate Noise: 1.000 m

Standard Deviation of **Clock bias** Noise: 50.000 ps

Noise Generator Seed Value: 0

Sinusoid Parameters

Amplitude of Sinusoidal Component: 0.000 m

Period of Sinusoidal Component: 3600 s

Figure 3.6: Satellite orbital and clock configuration widget

model of orbital and clock errors is shown in Equation 3.2.

$$
\begin{aligned}
\Delta X &= R_G(\phi,\sigma_X) + A\cdot sin\left(2\pi\cdot\tfrac{\Delta t}{T} + R_U(\phi,T)\right)\\
\Delta Y &= R_G(\phi,\sigma_Y) + A\cdot sin\left(2\pi\cdot\tfrac{\Delta t}{T} + R_U(\phi,T)\right)\\
\Delta Z &= R_G(\phi,\sigma_Z) + A\cdot sin\left(2\pi\cdot\tfrac{\Delta t}{T} + R_U(\phi,T)\right)\\
\Delta\delta &= R_G(\phi,\sigma_\delta)
\end{aligned}
\tag{3.2}
$$

ΔX, ΔY, and ΔZ are respective errors in X, Y and Z Earth-Centred Earth-Fixed (ECEF) satellite positions, R_G is a Gaussian random function, ϕ is a user defined seed value, σ_X,σ_Y and σ_Z are user defined standard deviations in X, Y and Z ECEF satellite position error, A is a user-defined sinusoid component amplitude, Δt is elapsed simulation time, T is user defined period, and R_U is a uniform random number function. In case a user wishes to simulate orbital errors as Gaussian noise, the sine component in Equation 3.2 can be dropped. The term $R_U(\phi,T)$ defines the phase offset mentioned earlier. This value is kept constant for the entire simulation run. It is stored in a look-up table with satellite IDs acting as keys. At the start of every simulation run, these values are generated based on new values of ϕ and T.

It is certainly possible to implement a more advanced model of orbit and clock errors than the one defined here. One example in GPS would be to use the broadcast ephemeris to emulate broadcast ephemeris orbital errors. Unfortunately this requires the user to provide additional files and this method only works in GPS simulations. For GLONASS, a numerical integrator must be implemented along with an ability to parse GLONASS navigation files. It would appear that defining very realistic orbital and clock errors is not a straightforward matter. So the model shown here is a compromise between simplicity and accuracy.

3.5.2 Troposphere

The neutrality of the troposphere causes GNSS signals to be delayed equally regardless of their frequency. Due to its close proximity to the Earth's surface, the troposphere contains water vapour, which affects GNSS observables in a way different from other atmospheric constituents. The tropospheric effect is usually modelled in two parts: "wet" (i.e., containing water vapour) and hydrostatic or "dry" (not containing water vapour). Since the tropospheric effect is frequency independent, simulation of tropospheric errors is relatively straightforward. One just needs a model of two (wet and dry) zenith tropospheric delays, and then apply a slant transformation. A commonly used tropospheric model is the Modified Hopfield model, which is incorporated in most receiver firmware. The Modified Hopfield model is adopted in the "Tropospheric Effect" module of MGOS, which incorporates three user defined site parameters: temperature, pressure and humidity (Goad and

Goodman, 1974).

The model used here is summarized by Leick (1990). It requires the height of a receiver above mean sea level, as well as heights at which temperature, pressure and humidity measurements are made. To simplify the implementation of this model, all site atmospheric parameters are assumed to be measured at receiver height. The tropospheric delay is assumed to depend only on receiver atmospheric conditions and satellite elevation angle. The receiver height is calculated from its true coordinates provided during a simulation. The overall tropospheric error is generated as follows:

$$dTropo(P(t),T(t),H(t),El(t)) = dTropo^{Hopfield}(P(t),T(t),H(t),El(t)) + \omega \; [m] \quad (3.3)$$

$dTropo^{Hopfield}$ is the slant tropospheric delay generated using the Goad and Goodman (1974) tropospheric model, $P(t)$ is pressure, $T(t)$ is temperature, $H(t)$ is humidity, $El(t)$ is elevation angle, and ω is Gaussian white noise. All parameters described in Equation 3.3 (except for $El(t)$) are user-controlled.

Typical tropospheric delay that can be generated using this model as a function of a satellite elevation angle is shown in Figure 3.7. The user can specify an expression for each parameter that can vary with simulation epoch. Phase tropospheric delay is computed by converting distance to an equivalent number of cycles using the appropriate frequency. In generating time series for pressure, temperature and humidity, MGOS allows for noise to enter. The purpose of the noise term is to ensure that the generated tropospheric delay cannot be completely removed by post-processing software. The Modified Hopfield model has been adapted for a ground based user. Therefore invalid values would be generated at elevations exceeding 40 kilometres (Goad and Goodman, 1974). According to the Modified

Hopfield model, generated tropospheric errors grow as the cosecant of elevation angles. This leads to unrealistically large delays at low elevation angles.

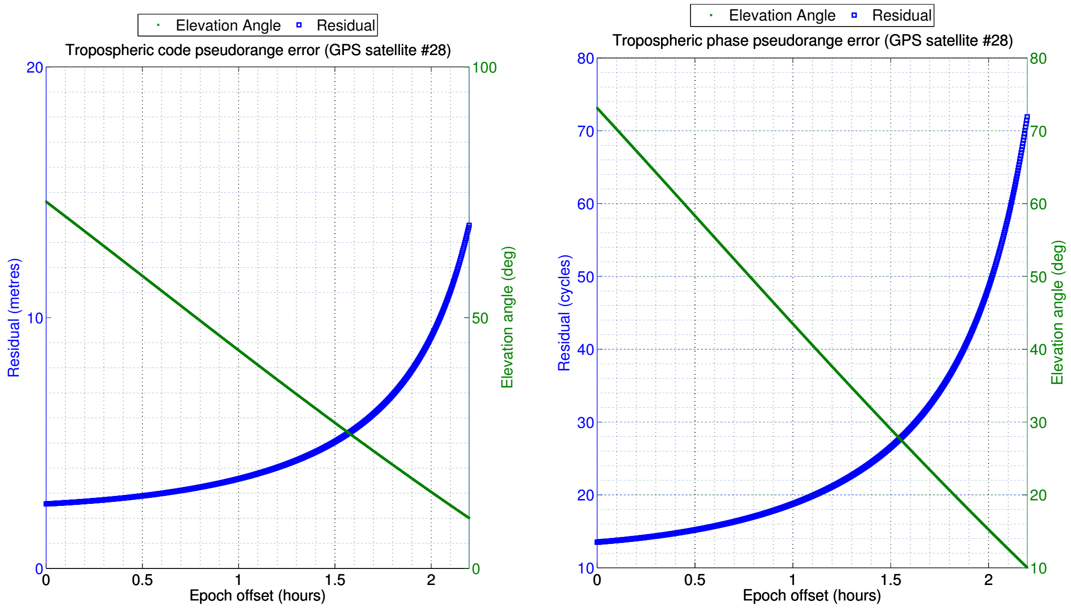


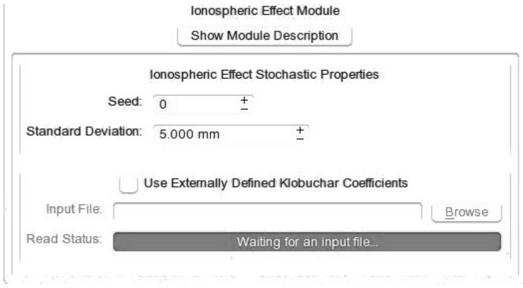
Figure 3.7: Example of generated tropospheric delay for the GPS L1 channel

3.5.3 Ionosphere

Ionospheric error is the largest magnitude GNSS error to be simulated. Since the ionosphere is a dispersive medium, and each GNSS has its own set of carrier frequencies, ionospheric error has to be generated for each GNSS data channel individually. A further complication is that for a system like GLONASS that employs Frequency Division Multiple Access (FDMA), ionospheric error must be generated for each satellite separately. Currently, MGOS has two modules that can generate ionospheric delay: the Klobuchar module, and the IONEX module. The first module adopts the Klobuchar model relevant to the GPS L1 channel and is described in the GPS Interface Control Document with alpha and beta coefficients set to some predefined values (GPS Joint Program Office, 1993). The algorithm adopted in the Klobuchar module is described by the following equation:

$$\begin{aligned}\Delta P(f_L,El) &= \left(\frac{f_{L1}}{f_L}\right)^2 dIono_{L1}(El)+\omega(\sigma)\ [m]\\ \Delta L(f_L,El) &= -\frac{c}{f_L}\left[\left(\frac{f_{L1}}{f_L}\right)^2 dIono_{L1}(El)+C\omega(\sigma)\right]\ [cycles]\end{aligned} \tag{3.4}$$

ΔP is code pseudorange error, ΔL is phase pseudorange error, El is elevation angle to a satellite, c is speed of light, $dIono_{L1}$ is ionospheric correction computed from the Klobuchar model for GPS L1 channel, f_L is the frequency of a signal for which the ionospheric effect is to be generated, f_{L1} is frequency of GPS L1 channel, ω is Gaussian white noise with user defined standard deviation σ, and C is code-to-phase ionospheric noise scale factor (0.001). As with the tropospheric delay model described previously, additional Gaussian noise is used in order to avoid cancellation of the generated ionospheric residuals with post processing software. Model coefficients can also be dynamically loaded using GPS RINEX navigation files that contain “ION ALPHA” and “ION BETA” parameters. This allows for more realistic ionospheric errors to be generated for a specific simulation day. Figure 3.8 shows available configuration parameters as they appear in MGOS.



Ionospheric Effect Module

Ionospheric Effect Stochastic Properties

Seed: 0

Standard Deviation: 5.000 mm

□ Use Externally Defined Klobuchar Coefficients

Input File:

Read Status: Waiting for an input file...

Figure 3.8: Klobuchar ionospheric module configuration widget

Since the Klobuchar model only accounts for approximately 60 percent of ionospheric error, MGOS provides an additional ionospheric module that can process IONEX implements the algorithm described by Schaer et al. (1998), which is used to calculate vertical Total Electron Content (TEC) for a specific receiver location. Using an inverse cosine mapping function, TEC values are converted into range delays and phase advances. Noise can also be added in this module, based either on Root Mean Square (RMS) values extracted from TEC values or user selected standard deviation. The mathematical model for code and phase pseudorange ionospheric delays using IONEX approach is defined in the following equation:

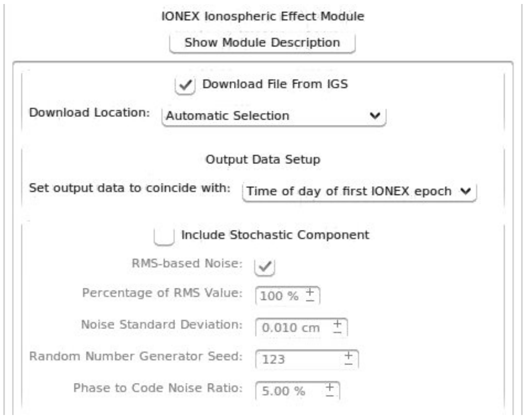
$$
\begin{aligned}
\Delta P(f_L, El) &= \left(\tfrac{40.3\times10^{16}}{f_L^2}\right) M(El)\cdot[TEC+\omega(\sigma)]\ [m] \\
\Delta L(f_L, El) &= -\left(\tfrac{c}{f_L}\right)\left(\tfrac{40.3\times10^{16}}{f_L^2}\right) M(El)\cdot[TEC+C\omega(\sigma)]\ [cycles]
\end{aligned}
\tag{3.5}
$$

In Equation 3.5, TEC is interpolated total electron content, $M(El)$ is a mapping function used to convert ionospheric to arbitrary line of sight inclined at El degrees. This mapping function is defined as follows:

$$
M(El) = \frac{R_{IONO}}{\sqrt{R_{IONO}^2 - a^2\cdot cos(El)^2}} \tag{3.6}
$$

In Equation 3.6, R_{IONO} is the ionospheric layer radius defined in an IONEX file and a is the reference radius of Earth (6371 kilometres). Like the Klobuchar model, the ionospheric noise in the phase observable is scaled down by a factor of C. This factor is user controlled to allow for more flexibility. However, a default value of 0.001 is most appropriate for nearly all simulations. To preserve the day/night variation of the ionosphere captured by an IONEX file, the module can be set to preserve the time of day of the file during a simulation run, otherwise the ionospheric effect is time-shifted to the start of the run. The IONEX module therefore generates ionospheric errors that are significantly more realistic than those that can be obtained with the Klobuchar module. IONEX maps are used to compare the performance of ionospheric activity monitoring software to the IGS (e.g., such processing is described by Leandro et al. (2007b); Le et al. (2008)).

All parameters can be user controlled via the IONEX module interface. The IONEX files can either be fed to the module via its configuration widget or downloaded automatically by the module. Its configuration widget is shown in Figure 3.9. Despite the fact that the IONEX module is more comprehensive than the Klobuchar module, there are occasions Klobuchar-like ionospheric errors are required, e.g., when testing GNSS processing software. For illustration purposes, Figure 3.10 compares the ionospheric errors generated by the Klobuchar module to those generated by the IONEX module during the 9 AM to 1 PM period on April 1st, 2009 for a station located at 43°N and 78°W. Klobuchar coefficients for this day have been obtained from Centre for Orbit Determination in Europe (CODE) website (CODE, 2009).



IONEX Ionospheric Effect Module

☑ Download File From IGS

Download Location: Automatic Selection

Output Data Setup

Set output data to coincide with: Time of day of first IONEX epoch

□ Include Stochastic Component

RMS-based Noise: ☑

Percentage of RMS Value: 100 %

Noise Standard Deviation: 0.010 cm

Random Number Generator Seed: 123

Phase to Code Noise Ratio: 5.00 %

Figure 3.9: IONEXionospheric module configuration widget

The IONEX file is acquired automatically by the IONEX module from IGS. Ionospheric conditions for the given site are very quiet at the given time and date (evidenced by downloaded TEC values), which is why the generated ionospheric errors are small (less than 5 metres at an elevation angle of 10 degrees). From Figure 3.10, the Klobuchar model has generated ionospheric errors approximately 75 percent larger than those generated by the IONEX module. This shows that for single-frequency data sets, the IONEX module tends to produce smaller biases than the Klobuchar model. For lower accuracy (say several metres or more) applications, the Klobuchar model may suffice.

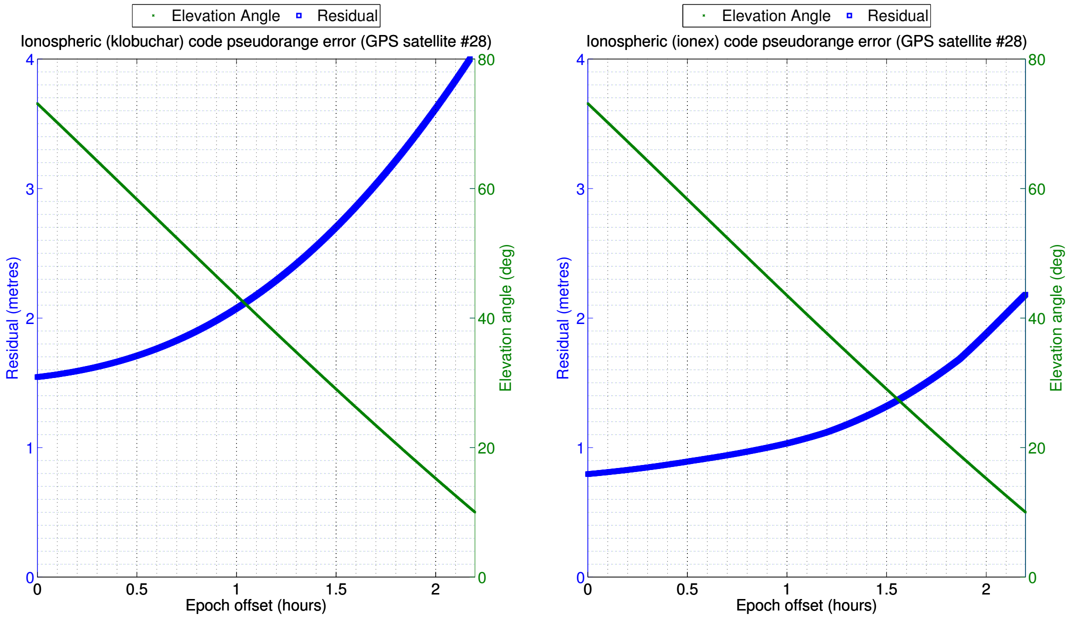


Figure 3.10: Example of generated ionospheric errors by Klobuchar (left) and IONEX (right) modules for the GPS L1 channel

3.5.4 Hardware Biases

Group delays are known to exist in GNSS measurements. Unfortunately these errors have not been extensively studied either because their impact on accuracy has previously been negligible or because they can be eliminated (e.g., in relative positioning). In recent years, analysis of hardware biases has become an important topic for time transfer, ionospheric mapping applications using GNSS receivers, and Precise Point Positioning (PPP). For example, it has been shown that satellite and receiver hardware biases vary from receiver to receiver and satellite to satellite, as well as between different frequency channels and modulation codes (Gao et al., 2001; Leandro et al., 2007a, b). It has been shown that these errors can generate range errors of half a metre or more, and can vary drastically in time.

As hardware bias estimation is an ongoing research area, it is not possible to accurately simulate hardware bias errors in MGOS. Nevertheless, preliminary knowledge of these errors has given rise to a simulation strategy suitable for adoption in MGOS modules. The strategy can be described mathematically as follows:

$$\begin{aligned} \Delta P &= \left(\delta_I^r + \delta_I^s + \delta_C^r + \delta_C^s + \delta_C^0 + s_C + \varepsilon_C\right) \text{ [metres]} \\ \Delta L &= f_L\left(\delta_I^r + \delta_I^s\right) \; [cycles] \end{aligned} \tag{3.7}$$

In Equation 3.7, c is speed of light, f_L is frequency of the L$^{\text{th}}$ frequency channel, δ_I^r and δ_I^s are group delays for r$^{\text{th}}$ receiver and s$^{\text{th}}$ satellite, respectively, in seconds, δ_C^r and δ_C^s are instrumental code biases for r$^{\text{th}}$ receiver and s$^{\text{th}}$ satellite, respectively, in seconds, δ_C^0 is a zero-mean user controlled code noise, s_C is a satellite-receiver hardware bias drift term in seconds, and ε_C is a Gaussian white noise for code drift term in seconds. Using the model described in Equation 3.7, pseudorange and phase hardware biases are generated for a specific receiver (identified by its ID) observing a satellite within any GNSS constellation that transmits at certain frequencies and uses certain modulation codes.

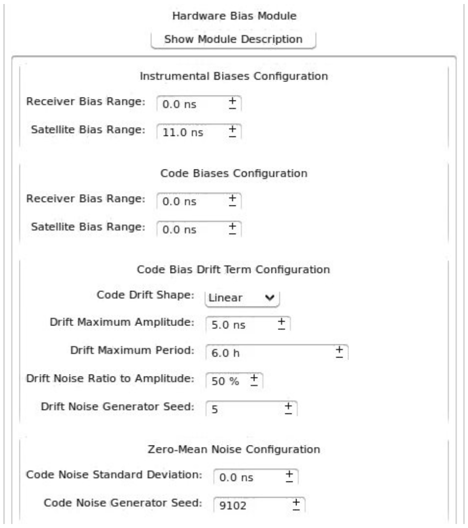
The biases are generated using a random number generator whose seed value is determined entirely by the designations assigned to each receiver, each satellite, each frequency channel, and each code. All generated biases are kept constant throughout the entire simulation run. Group delays are frequency dependent and can be eliminated using the ionosphere-free combination (Leick, 2004). Given this fact, the following simulation model has been adopted to simulate Group delays for any signal frequency f:

$$\delta_I^{r/s}(f) = \left(\frac{f_{L1}}{f}\right)^2 \left(\frac{1}{1-\frac{f_{L1}}{f_{L2}}}\right) \delta_{I_0}^{r/s} \tag{3.8}$$

In Equation 3.8, f is frequency of a particular data channel, f_{L1} and f_{L2} are GPS L1 and L2 frequencies, respectively, and $\delta_{I_0}^{r/s}$ is randomly generated group delays. This formalization guarantees that group delays cancel when using the ionosphere-free combination and gives the proper ratio between GPS L1 and L2 biases as well. Simulation of other hardware-related biases is less sophisticated than the one shown in Equation 3.8.

The code bias drift term shown in Equation 3.7 is added to simulate the temporal variations of hardware biases. This drift term can either be static (constant value) or linear (ramp). Drift noise is based entirely on the chosen amplitude of code bias drift term. A user needs to specify the relative amplitude of the code drift noise (in percent) if code drift noise is to be simulated. The value for the drift term is dependent on the chipping rate of a particular channel's code. In essence this dependence is achieved by scaling the term by the ratio between a signal's chipping rate and the C/A-code chipping rate. As a result, higher chipping rate PRN codes have less noise than lower chipping rate PRN codes. In case the drift term is either 0 or very small, a user can choose to add the zero-mean code noise. This is essentially a Gaussian white noise with a specified standard deviation. The zero-mean code noise parameter can increase code noise to make code noise more realistic. The zero-mean code noise can also be used with the code bias drift term if such a combination is required. A user can select the module's simulation parameters via its configuration widget in Figure 3.11.

Regardless how the drift term is represented (either static or linear), its mean value will change every time a new satellite is observed and will remain constant throughout the



Hardware Bias Module

Instrumental Biases Configuration

Receiver Bias Range: 0.0 ns

Satellite Bias Range: 11.0 ns

Code Biases Configuration

Receiver Bias Range: 0.0 ns

Satellite Bias Range: 0.0 ns

Code Bias Drift Term Configuration

Code Drift Shape: Linear

Drift Maximum Amplitude: 5.0 ns

Drift Maximum Period: 6.0 h

Drift Noise Ratio to Amplitude: 50 %

Drift Noise Generator Seed: 5

Zero-Mean Noise Configuration

Code Noise Standard Deviation: 0.0 ns

Code Noise Generator Seed: 9102

Figure 3.11: Hardware biases module configuration widget

entire observation period of that satellite. This is done by noting the time when a satellite is no longer observed, thus forcing regeneration of a mean value when that satellite is observed again. As a satellite remains in view, this mechanism is not used. The overall approach used in generating hardware biases attempts to mimic the behaviour described by Gao et al. (2001); Leandro et al. (2007a, b). An example of the simulated code and phase pseudorange residuals is shown in Figure 3.12. Unfortunately, the model can only be used to generate approximate hardware biases. Though the hardware biases generation quality of MGOS can be enhanced as more advanced models of hardware biases are developed.

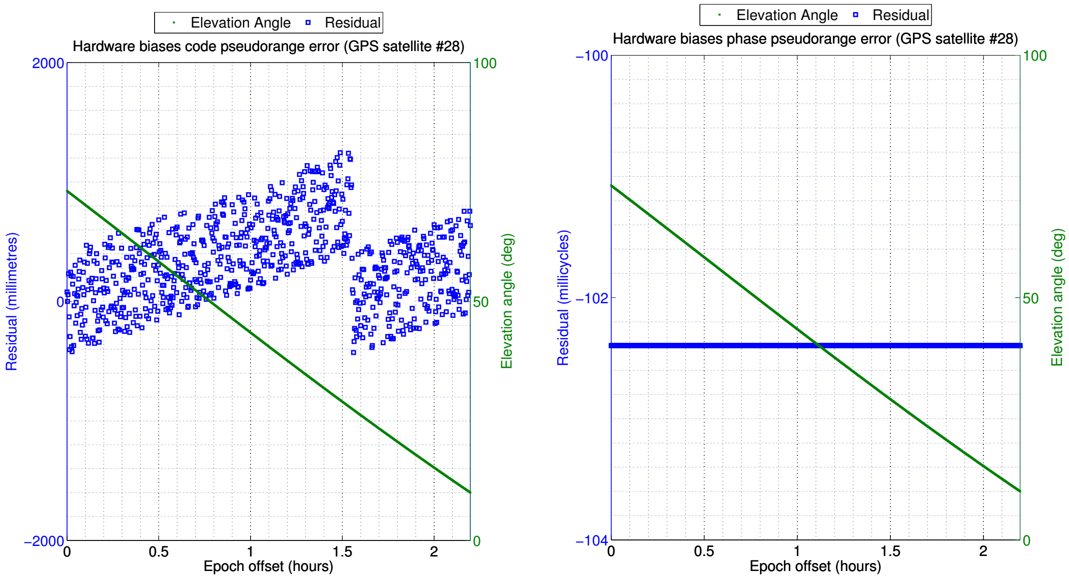


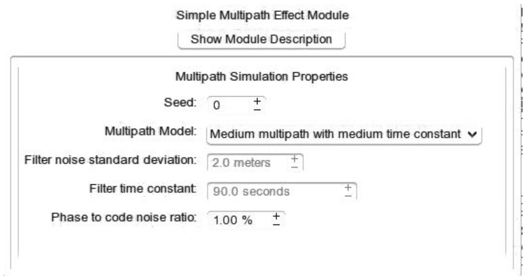
Figure 3.12: Example of generated hardware biases for the GPS L1 channel

3.5.5 Multipath

MGOS incorporates two models that deal with multipath delays. The first model is similar to that in SATNAV (GPSoft LLC, 2003). A Gaussian white noise sample is fed through a digital filter, creating an autoregressive noise sample, which is then scaled by elevation angle:

$$\begin{aligned} \Delta P(\phi, El) &= \frac{\phi_{C/A}}{\phi} filter(\omega, a, b) cos(El)\ [m] \\ \Delta L(\phi, El, f_L) &= \frac{c}{f_L} K \Delta P(\phi, El)\ [cycles] \end{aligned} \tag{3.9}$$

ϕ is chipping rate of a PRN code, $\phi_{C/A}$ is GPS C/A-code chipping rate, ω is a Gaussian white noise sample, a and b are filter parameters, El is elevation angle, c is speed of light, f_L is frequency of a channel L, and K is code-to-phase scaling factor (the default of which is 0.01). Filter parameters a and b are user selected, and are similar to those defined in SATNAV. The autoregressive model described in Equation 3.9 will only generate multipath error for GPS C/A-code. To generate multipath for other PRN codes (not necessarily GPS-related codes), the ratio is used which attempts to simulate the fact that lower chipping frequency PRN codes give rise to higher multipath. Phase multipath is generated in a similar manner, except scaled down by two orders of magnitude to show that phase multipath error is much smaller than code multipath error (Braasch and DiBenedetto, 2001). This is accomplished using the *K* parameter which can be set by a user to any desired value. The parameters for this multipath module are shown in Figure 3.13.



Simple Multipath Effect Module

Multipath Simulation Properties

Seed: 0

Multipath Model: Medium multipath with medium time constant

Filter noise standard deviation: 2.0 meters

Filter time constant: 90.0 seconds

Phase to code noise ratio: 1.00 %

Figure 3.13: Autoregressive multipath module configuration widget

One advantage of the above technique is that the multipath so generated does not require a complex list of parameters. Moreover, this method can represent multipath quite well in very basic situations (GPSoft LLC, 2003). A major disadvantage of this method, however, is that it completely ignores the receiver environment. As a result, all multipath generated samples will look the same whether the receiver is moving or stationary. It also does not demonstrate the dependence of the multipath error on receiver correlator properties. Examples of “medium” multipath error generated using the autoregressive module for C/A-code and phase observables are shown in Figure 3.14.

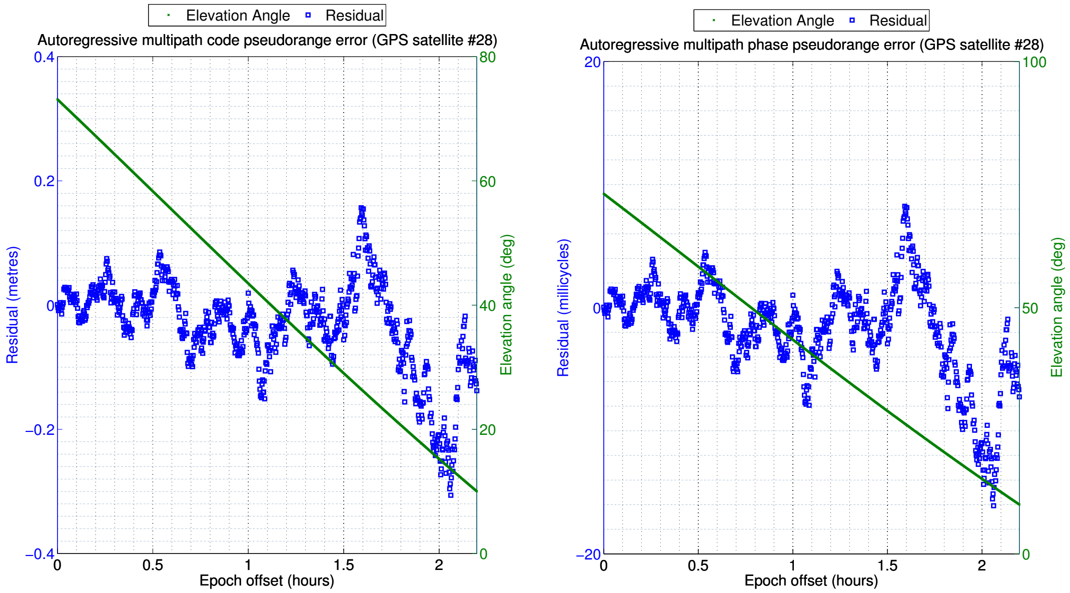
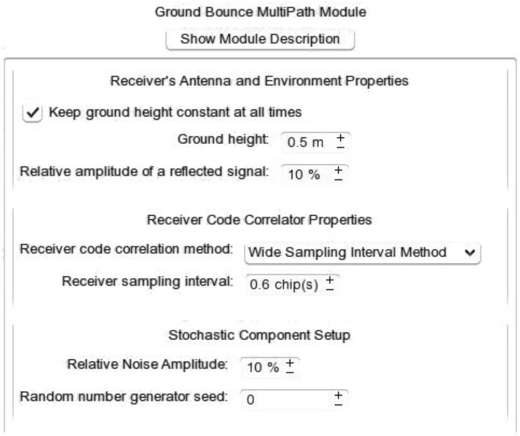


Figure 3.14: Example of generated multipath using the autoregressive multipath module for the GPS L1-channel

To represent multipath in a more realistic way, MGOS includes another multipath module that takes into account reflections due to a planar surface. The developed model has been influenced by the MUSTARD simulator presented by Byun et al. (2002), which can simulate not only ground-bounce multipath, but also multipath caused by refraction and objects surrounding the receiver. A multipath module based on a ray-tracing algorithm is planned in the future in order to allow for even more realistic multipath noise simulations. The ground multipath module requires a user to specify ground height, correlator type (narrow or wide), sampling interval, and how effectively a receiver can block reflected signals. The module encapsulating the ground-bounce multipath model has the configuration parameters shown in Figure 3.15.

The “relative amplitude of reflected signal” parameter shown in Figure 3.15 is the ratio of reflected signal strength and direct signal strength. The model then uses this parameter,



Ground Bounce MultiPath Module

Show Module Description

Receiver's Antenna and Environment Properties

☑ Keep ground height constant at all times

Ground height: 0.5 m

Relative amplitude of a reflected signal: 10 %

Receiver Code Correlator Properties

Receiver code correlation method: Wide Sampling Interval Method

Receiver sampling interval: 0.6 chip(s)

Stochastic Component Setup

Relative Noise Amplitude: 10 %

Random number generator seed: 0

Figure 3.15: Ground-bounce multipath module configuration widget

as well as channel frequency, and modulation code to calculate the expected multipath delay. Signal loss is also taken into account. Code multipath and phase multipath are computed using the equations described in Byun et al. (2002). Only the equation describing range delay caused by a single signal ray being reflected from a ground is presented here. This equation is derived from the geometrical definition of ground-bounce multipath delay shown in Figure 3.16.

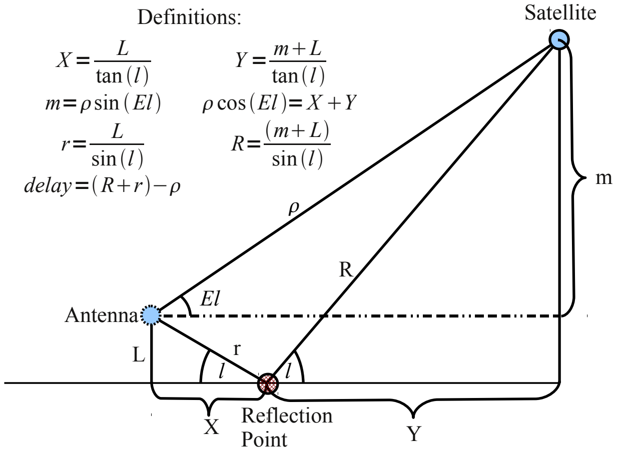


Figure 3.16: Ground-bounce multipath geometrical representation using a single signal ray

From Figure 3.16, range multipath delay can be obtained from Equation 3.10.

$$\Delta\rho(El) = \sqrt{4L^2 + 4L\rho sin(El) + \rho^2} - \rho \tag{3.10}$$

Elements of Equation 3.10 are defined in Figure 3.16. Code and phase delays are generated using user defined receiver correlator parameters and the computed$\Delta\rho(El)$ value. An example of generated multipath for the C/A-code on L1 for a randomly chosen GPS satellite is shown in Figure 3.17, where simulation parameters are the same as those used in Figure 3.15. MGOS tests showed that the resultant multipath error is consistent with the multipath error presented by Byun et al. (2002). It is interesting to note that ground-bounce multipath scale directly with elevation angle. Observed multipath tends to vary inversely to elevation angle, which is how the autoregressive model (presented earlier) behaves.

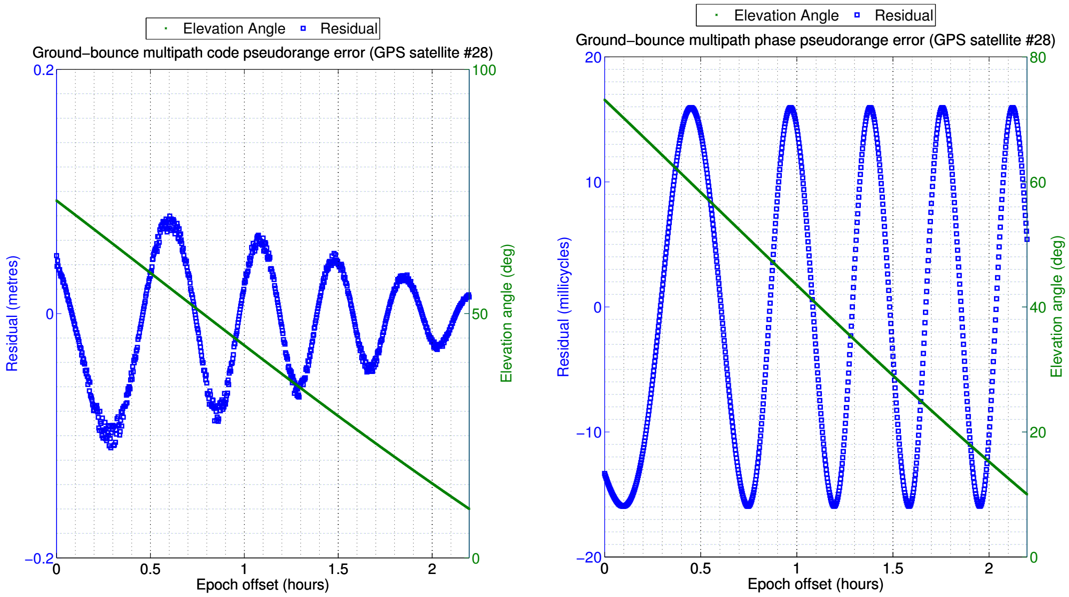


Figure 3.17: Example of generated multipath using the ground-bounce multipath module for the GPS C/A-code on L1

3.5.6 Phase Windup Effect

As a minor error source that can be generated by the “Minor Error Source Module,” (used mostly for undifferenced phase processing - such as PPP), phase windup is an effect caused by relative orientation change of a satellite’s antenna with respect to a receiver’s antenna. This is caused by the circular polarization utilized by GNSSs. For circularly polarized signals, relative orientation of transmission and reception antennas is important. Any orientation change causes biases to be formed in estimating the number of cycles of carrier wave being counted. GPS satellites rotate on a regular basis so as to orient their solar panels towards the Sun. Satellite orientation can also change rapidly during noon and midnight phases of the orbit. Noon phase is defined as a time when Sun-Earth line intersects spacecraft position before intersecting Earth centre, and midnight phase is defined as a time when Sun-Earth line intersects Earth centre before intersecting spacecraft position (Héroux and Kouba, 2001).

Phase windup is modelled as an angle between effective dipoles of receiver and satellite antennas. More details can be found in Héroux and Kouba (2001); García-Fernández et al. (2008). An important factor in properly determining phase windup is the knowledge of how each GNSS satellite is oriented. Invalid determination of the phase windup effect can lead to errors in post-processing involving phase observations. Nevertheless, successful models of satellite orientation (in particular satellite yaw angle) have been implemented. The model implemented for phase windup in MGOS is based almost entirely on that shown in Kouba (2009).

At the moment (as will be shown in section 3.7), the current implementation is a good approximation to models used by PPP post-processing software; however, receiver antenna rotation-based phase windup is not modelled in the module to simplify its implementation and use. In future releases of MGOS a more accurate phase windup model will be adopted to further enhance simulation realism. Future MGOS developments can also benefit from an improved knowledge of the phase windup effect for GLONASS, Galileo and Compass satellites. For illustration purposes Figure 3.18 shows a generated sample of phase windup for a GPS satellite.

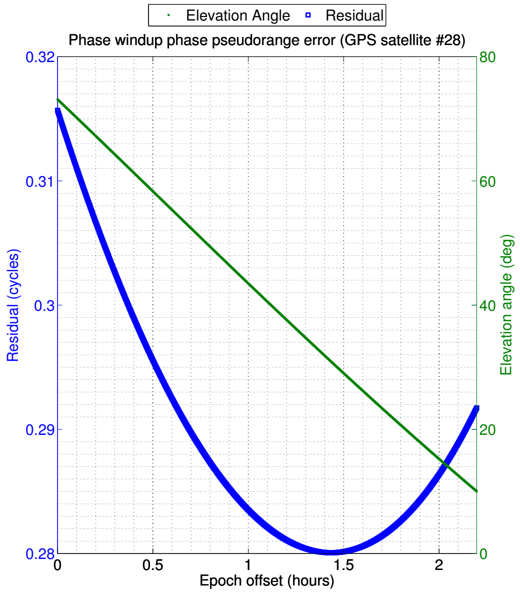


Figure 3.18: Example of generated phase windup error for the GPS L1 channel

3.5.7 Relativistic Range and Clock Errors

Relativistic effects are a result of relative motion between a satellite and a receiver. Receiver motion is due to rotation of Earth as well as its own motion, while the satellite's motion is due to its own orbit around Earth. According to relativity, clocks are affected by motion. Since a GNSS receiver acquires satellite ranges through timing, it is important to account for differences between the true satellite clock and perceived satellite clock, as well as effects caused by satellite orbital eccentricity (Hofmann-Wellenhof et al., 2001). MGOS'

Relativistic Effect" module simulates two types of relativistic effects: range effect, and satellite clock bias effect due to satellite orbital eccentricity.

The range effect is defined as follows (Hofmann-Wellenhof et al., 2001):

$$\Delta\rho_{rel} = \left(\frac{2\mu_{WGS84}}{c^2}\right) log\left(\frac{\rho^S + \rho_U + \rho_U^S}{\rho^S + \rho_U - \rho_U^S}\right) \; [m] \tag{3.11}$$

μ_{WGS84} is the product of Newton's gravitational constant and the mass of Earth as defined by WGS84, c is speed of light, ρ^S is geocentric satellite vector magnitude, ρ_U is geocentric receiver vector distance, and ρ_U^S is vector magnitude between the receiver and the satellite. Relativistic errors tend to reduce range estimates, so the above error needs to be subtracted from the simulated range estimate. The range satellite clock bias effect is computed as follows (Hofmann-Wellenhof et al., 2001):

$$\Delta\delta C_{rel} = \frac{2}{c}\left(\overrightarrow{R}_S \cdot \overrightarrow{V}_S\right) \; [m] \tag{3.12}$$

$\overrightarrow{R}_S$ is a geocentric satellite vector, and $\overrightarrow{V}_S$ is a satellite velocity vector. Generated clock bias is added to code and phase observables since the relativistic clock effect tends to increase range estimates. The phase observable relativistic error is computed by simply scaling range errors by the reciprocal of wavelength (as is done with other error sources mentioned). Both the clock error and range errors are computed at each epoch since satellite and receiver vectors change with time. For illustration purposes, Figure 3.19 shows generated code and phase relativistic effects for a satellite pass.

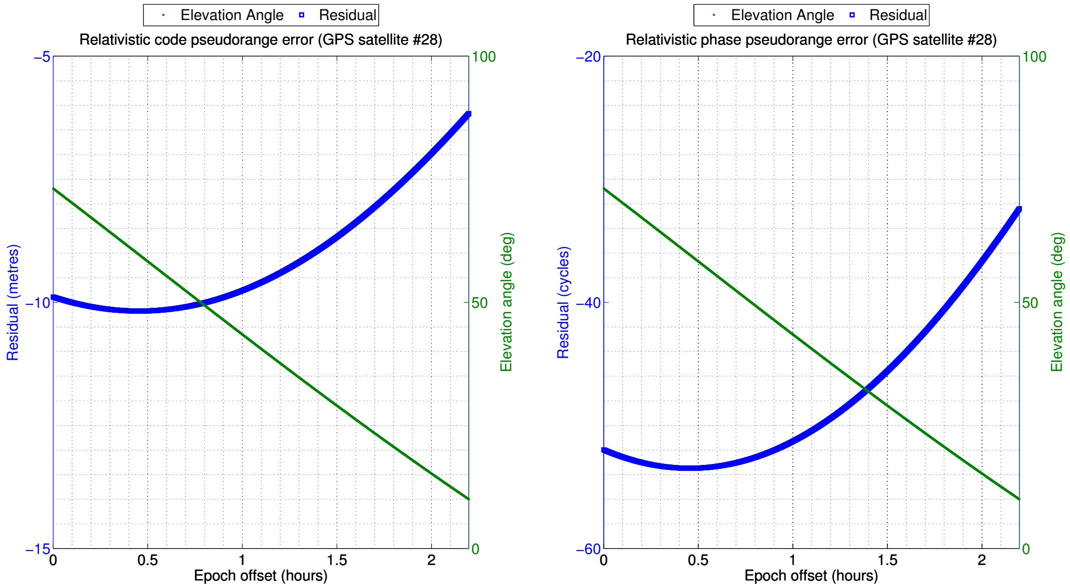


Figure 3.19: Example of generated code and phase relativistic observable errors for the GPS L1 channel

3.5.8 Tidal Effects

The “Minor Error Source Module” can also be used to generate two types of tidal related errors: solid Earth tides, and ocean loading. These effects result in minor (few centimetres) displacements in receiver position. These displacements need to be removed to obtain high accuracy positioning (Héroux and Kouba, 2001). In MGOS, these errors need to be incorporated if simulated data are to be processed by high-accuracy post-processing software that take these errors into account. For all other simulations where such errors are either negligible (for example decimetre or worst positioning) or completely irrelevant (like satellite positioning), it is possible to disable their generation in the Minor Error Source Module configuration widget. The following two sections describe these errors in detail, and offer samples for illustration purposes.

3.5.8.1 Solid Earth Tides Simulation

Solid Earth tides refer to the deformation of Earth's crust due to tidal forces (most notably luni-solar tides). These tides cause decimetre displacements in the radial component for ground-based receivers, and exhibit temporal and latitude variations. Typically they are modelled using spherical harmonics characterized by Love numbers (Héroux and Kouba, 2001). Tidal errors contain a secular term as well as diurnal and semi-diurnal components. It is possible to estimate these errors using GPS data in either static or kinematic stations as shown by King (2006) (where Earth-bound receivers require centimetre or better position accuracy). The model used to simulate solid Earth tidal displacements is described by Héroux and Kouba (2001).

Tide models depend on positions of the Moon and the Sun, Greenwich Mean Sidereal Time, as well as receiver coordinates. All of this information is computed directly by the Minor Error Source Module using receiver position at a particular simulation epoch. Coordinates of the Sun and the Moon computed using subroutines obtained from Natural Resources Canada. Since MGOS operates in the observation domain, the resultant displacements must be converted to ECEF coordinates, and then to range error, as follows:

$$\begin{aligned} \Delta P &= \delta \overrightarrow{T} \cdot \left(\overrightarrow{S} - \overrightarrow{R} \right) \; [metres] \\ \Delta L &= \left(\frac{c}{f} \right) \Delta P \; [cycles] \end{aligned} \tag{3.13}$$

ΔP is code observable error, ΔL is phase observable error, $\delta \overrightarrow{T}$ is the Earth tide displacement vector in ECEF, $\overrightarrow{S}$ is satellite geocentric vector in ECEF, $\overrightarrow{R}$ is receiver true geocentric vector in ECEF, c is speed of light, and f is frequency of associated data channel. Figure 3.20 shows a sample generated solid Earth tide observable error time series as a function of elevation angle.

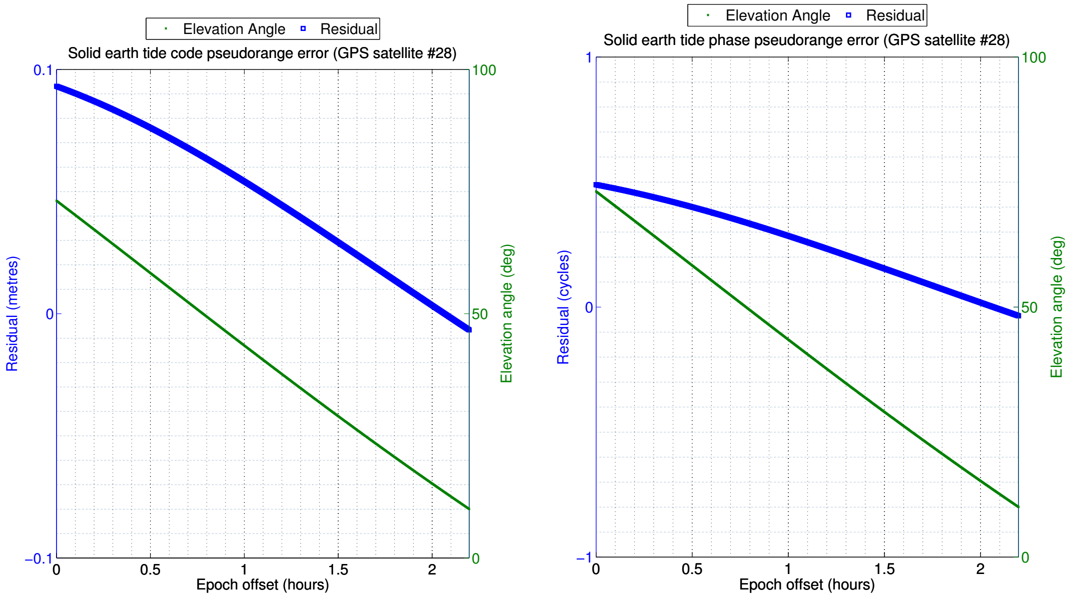


Figure 3.20: Example of generated solid Earth tidal error for code and phase observables for the GPS L1 channel

3.5.8.2 Ocean Loading Effects Simulation

Ocean loading tides are formed by pressures exerted by the ocean on Earth's crust. Ocean loading is in general approximately one order of magnitude smaller than solid Earth tides, and concentrated in coastal regions. For stations far from coastal regions (over 1000 kilometres), the ocean loading effect can be ignored. Ocean loading is modelled using spherical harmonics, and depends on receiver latitude and longitude. Formulation of the model is described briefly in Héroux and Kouba (2001) and in more detail in McCarthy and Petit (2004). In essence, to calculate site displacement (in North, East and Up coordinate frame), one requires the use of model coefficients (such as phase, frequency and amplitude for each component of the model). There are many algorithms available to obtain the necessary coefficients; the majority of which are described in Bos and Scherneck (2009a). To simulate ocean loading, a user must use the Minor Error Source module. The module can accept a file with ocean loading model coefficients used to compute the error. There exist two types of formats for this file: the older BLQ format and the newer HARPOS format. BLQ is a more common format; however, it only lists amplitudes for model components and does not list site location and harmonic coefficients. The HARPOS format is self contained, and can be used as is, without the need to look-up a site's location or to model harmonic coefficients.

With free on-line ocean loading available from Onsala Space Observatory, obtaining HARPOS or BLQ formatted ocean loading files is straightforward (Bos and Scherneck, 2009b). The user can provide HARPOS formatted files to Minor Error Source module in order to simulate ocean loading for a specific receiver location. Also, the Minor Error Source module automatically provides a built-in database for a large number of (namely 1096) stations around the world. Ocean loading error is computed using Algorithm A.1. The method used to locate ocean loading harmonic amplitude coefficients for a specific receiver location shown in Algorithm A.1 on line 7 is similar to that used in Natural Resources Canada PPP CSRS software (NRCan, 2004). Ocean loading error is converted to range error using Equation 3.13. Figure 3.21 shows the ocean loading code and phase errors generated for a station situated at 43° North and 78° West.

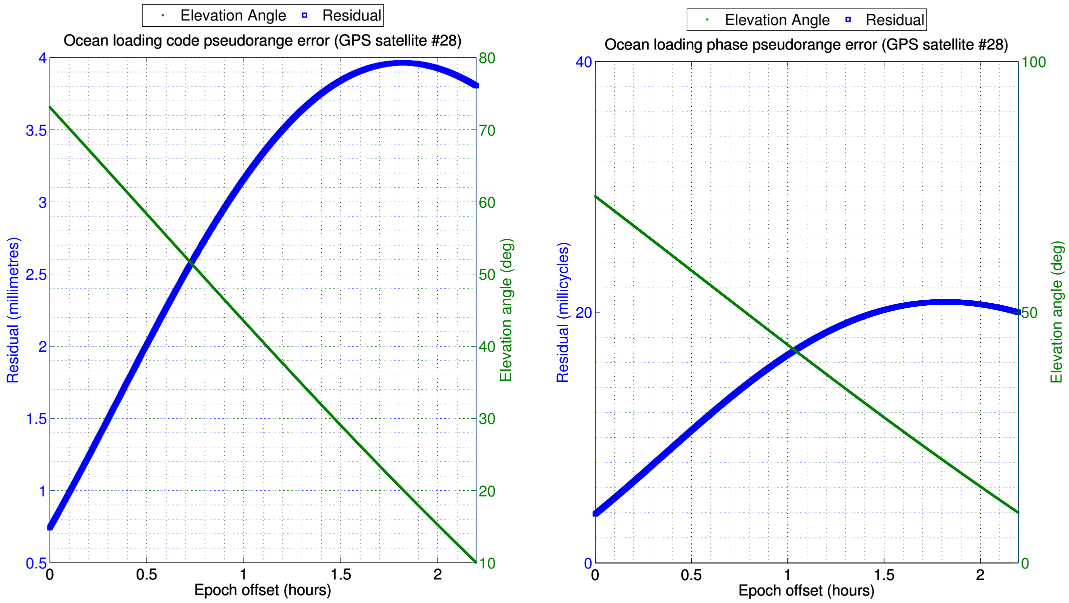
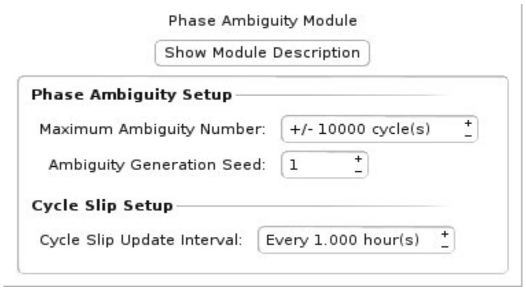


Figure 3.21: Example of generated ocean loading tidal error for the GPS L1 channel

3.5.9 Phase Ambiguities and Receiver Noise

Phase ambiguities are modelled as randomly chosen integers which vary from channel to channel, receiver to receiver, and satellite to satellite. Generated ambiguities can either be constant or set to slip at specified time intervals. It is also possible to generate a different set of ambiguities for each run by adjusting the random number generator seed. The maximum ambiguity value can also be controlled by the user. The available parameters for the phase ambiguity module are shown in Figure 3.22

The receiver noise is generated by the hardware biases, atmospheric, and user modules as Gaussian or normally distributed noise. MGOS does not feature a dedicated module for the receiver noise since error source modules can simulate code and phase observable noise by themselves. Nevertheless, a dedicated receiver noise module can be created to provide a more centralize control of this error source.



Phase Ambiguity Module

Show Module Description

Phase Ambiguity Setup

Maximum Ambiguity Number: +/- 10000 cycle(s)

Ambiguity Generation Seed: 1

Cycle Slip Setup

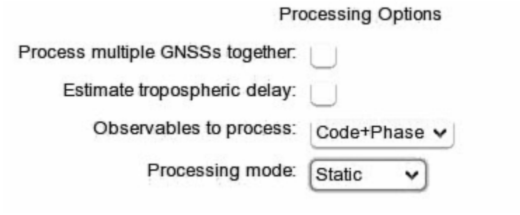
Cycle Slip Update Interval: Every 1.000 hour(s)

Figure 3.22: Phase ambiguities module configuration widget

3.6 Multi-GNSS Processor Module

MGOS would not be complete if it did not provide facilities to process measurements it generates. The Multi-GNSS Processor (MGP) module has been created to allow users to evaluate positioning performance of their simulations. The module calculates user position and compares it to true coordinates to determine true position errors. The module can also be used to estimate receiver clock bias, tropospheric delay, GDOP and phase ambiguities. Results are visualized using interactive plots which are displayed at the end of a simulation run. The MGP module also displays satellite tracks using a skyplot visualization. Results can be exported for plotting and analysis using external software.

MGP uses a sequential least-squares data processor (filter). This module is based on a PPP implementation described by Héroux and Kouba (2001). The elements of the filter state vector are completely defined by the user. Consequently, the filter's design matrix (matrix of measurement partials) and measurement vector are adjusted according to elements chosen for the state vector. For instance, if a user decides to process code and phase observables together, the state vector will include Cartesian coordinates, receiver clock bias, and phase ambiguities. If a user elects to process code observations alone, phase ambiguities will not be estimated. Elements for the state vector (and hence an observation model used by the filter itself) are determined via the MGP module's configuration widget in its "Processing Options" region shown in Figure 3.23.



Processing Options

Process multiple GNSSs together: □

Estimate tropospheric delay: □

Observables to process: Code+Phase

Processing mode: Static

Figure 3.23: Portion of Multi-GNSS Processor Module's configuration widget used to setup elements of the observation model

The "Process multiple GNSSs together" option shown in Figure 3.23 allows a user to combine data from several GNSSs together to form a single estimate of receiver position. The "Processing mode" option allows a user to define platform dynamics. Two modes are currently available: Static mode and Kinematic mode. Tropospheric delay is estimated using only the dry-component mapping function described by Niell (1996). As shown in section 3.5.2, the vertical tropospheric delay is computed as a linear sum of the dry and wet tropospheric delays. Separation of the two component errors is unnecessary, since the dry component error is not drastically larger than the wet component error (less than an order of magnitude). Hence only one parameter is used in estimating the total tropospheric delay. Along with the design matrix, the measurement vector and the weight matrix are also adjusted according to user needs. The complete description of each sequential least-squares filter component, is as follows:

$$\overline{L} = \begin{bmatrix} C_{S_1} & L_{S_1} & C_{S_2} & L_{S_2} & \cdots & C_{S_K} & L_{S_K} \end{bmatrix}^T \tag{3.14}$$

$$A = \begin{bmatrix} -\frac{(X_{s_1}-x)}{R_X^{S_1}} & -\frac{(Y_{s_1}-y)}{R_X^{S_1}} & -\frac{(Z_{s_1}-z)}{R_X^{S_1}} & 1 & M_X^{S_1} & 0 & 0 & \cdots & 0 \\ -\frac{(X_{s_1}-x)}{R_X^{S_1}} & -\frac{(Y_{s_1}-y)}{R_X^{S_1}} & -\frac{(Z_{s_1}-z)}{R_X^{S_1}} & 1 & M_X^{S_1} & 1 & 0 & \cdots & 0 \\ -\frac{(X_{s_2}-x)}{R_X^{S_2}} & -\frac{(Y_{s_2}-y)}{R_X^{S_2}} & -\frac{(Z_{s_2}-z)}{R_X^{S_2}} & 1 & M_X^{S_2} & 0 & 0 & \cdots & 0 \\ -\frac{(X_{s_2}-x)}{R_X^{S_2}} & -\frac{(Y_{s_2}-y)}{R_X^{S_2}} & -\frac{(Z_{s_2}-z)}{R_X^{S_2}} & 1 & M_X^{S_2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{(X_{s_K}-x)}{R_X^{S_K}} & -\frac{(Y_{s_K}-y)}{R_X^{S_K}} & -\frac{(Z_{s_K}-z)}{R_X^{S_K}} & 1 & M_X^{S_K} & 0 & 0 & \cdots & 0 \\ -\frac{(X_{s_K}-x)}{R_X^{S_K}} & -\frac{(Y_{s_K}-y)}{R_X^{S_K}} & -\frac{(Z_{s_K}-z)}{R_X^{S_K}} & 1 & M_X^{S_K} & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{3.15}$$

$$P = \begin{bmatrix} W_C^{S_1} & 0 & \cdots & 0 & 0 \\ 0 & W_L^{S_1} & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \vdots & 0 \\ 0 & 0 & \cdots & W_C^{S_n} & \vdots \\ 0 & 0 & \cdots & 0 & W_L^{S_n} \end{bmatrix} \tag{3.16}$$

Parameters in Equations 3.14, 3.15, and 3.16 are defined as follows:

- $\overline{L}$ is a measurement vector,
- A is a design matrix,
- P is a measurement weight matrix.
- S_i is i^{th} satellite,
- C_{S_i} is a code-derived pseudorange measurement for i^{th} satellite,
- L_{S_i} is phase-derived pseudorange measurement for i^{th} satellite,

- $M_X^{S_i}$ is Neill's dry-component mapping function value for current receiver's and i^{th} satellite's positions,
- $X_{s_i}, Y_{S_i}, Z_{S_i}$ are i^{th} satellite coordinates,
- x, y, z are receiver coordinates,
- $R_X^{S_i}$ is distance from the receiver to i^{th} satellite,
- $W_C^{S_i}$ is code-derived pseudorange weight for the i^{th} satellite,
- $W_L^{S_i}$ is phase-derived pseudorange weight for the i^{th} satellite,

The size of each filter component can be adjusted by simply dropping unnecessary rows or columns. The data passed to the filter for processing at every epoch consists of: 1) Satellite coordinates, 2) Code and phase observables, 3) Code and phase observable weights, and 4) Satellite ID and constellation name.

The chosen data format is flexible enough to be used with multiple GNSSs. When the "Process multiple GNSSs together" option shown in Figure 3.23 is checked, satellite observations from all available GNSSs are combined together in a single data list. If this option is unchecked, several filters are used to process GNSS data individually. Satellite coordinates, as well as code and phase observables must be in their final form prior to processing. For satellite coordinates, this means that they must be converted in the receiver's temporal and spatial frames. For observations, this means that all necessary corrections must be applied before hand, otherwise invalid results may be obtained. Observable weights are optional and are automatically set to 1 for code and 10000 for phase (representing 1 to 100 ratio in standard deviation between code and phase noise, respectively).

Since the sequential least-squares technique is used within the MGP module, covariance information is propagated from one epoch to the next. A co-factor matrix from a previous epoch is constructed and then updated to a current epoch. For a static receiver, position variance is assumed to be zero, so a zero update is made. For a kinematic receiver, a constant variance of a predefined value is used to update position variances. The user can adjust this value via the MGP module's configuration widget. Ambiguity values are assumed to be stationary, hence a zero update is used as well. Unfortunately, this method will fail during cycle slips where ambiguities may change. Cycle slip detection and repair mechanism have been left for future development. Fortunately, cycle slips can be disabled within MGOS, so the lack of cycle slip detection and repair does not pose an issue.

Clock bias and tropospheric error variance updating are done in the same fashion as for kinematic position variance updates. For clock bias variance, updated values can be controlled by the user. Tropospheric variance updates are not user controlled and are similar to what are done by Héroux and Kouba (2001) (approximately 25 mm squared per hour). For code-only processing, storing and recovering old covariance data is a straightforward matter, since matrix dimensions are consistent from one epoch to the next (the number of parameters does not change). For code and phase processing, storage and recovering of covariance data is more involved, since the number of satellites may change from one epoch to the next. This is especially problematic when processing data from several GNSSs together. To solve this problem, the data filter maps data from each satellite into a look-up table at the end of each epoch. If a old satellite is encountered in a new epoch, its covariance and phase ambiguity information is easily recovered. If a new satellite is encountered, then its ambiguity variance is set to a large value to make sure that its phase ambiguity (which is set to zero by default) is weighted down during the adjustment process. Once a new ambiguity value has been determined for this satellite, its information is stored in the same fashion as for the known satellites.

A sequential least-square filter needs to store information in memory for upcoming epochs. In cases where information for the current epoch cannot be obtained, the filter will reinitialize. During reinitialization, the state vector is reset if necessary and old covariance information is discarded. In the static case, previously estimated receiver coordinates can be used in the reinitialization process. In a kinematic case, receiver coordinates are recomputed using code-only processing regardless of the type of processing requested by the user. A complete description of the filter operation is described in Algorithm A.2.

3.7 Static Precise Point Positioning Data Simulation

To test the simulation capabilities of MGOS, data for a static station close to the IGS "ALGO" reference station in the Algonquin Park (Ontario, Canada) have been simulated for April 1st, 2009 and spans 24 hours. The station's true coordinates are shown in Table 3.5 relative to WGS84. ALGO's position is less than a metre from the test station position. Therefore ALGO and the test station will observe the same set of GNSS satellites. It is safe to assume that site conditions for the hypothetical station would be identical to those for ALGO. However MGOS cannot replicate ALGO's conditions exactly, as many error sources (such as tropospheric delay and multipath) are set approximately. It is not essential for MGOS to completely replicate data for stations, so long as MGOS is able to simulate realistic observables that allow for centimetre-level or better positioning. Centimetre or better positioning using a single receiver can be achieved using PPP.

This test was conducted using PPP processors supported by the Precise Point Positioning Software Centre at University of New Brunswick (Banville, 2008). At the time of writing, four on-line PPP processors were supported: CSRS-PPP developed by Natural Resources of Canada (NRCan), GPS Analysis and Positioning Software (GAPS) by the University

Coordinate	Value
Latitude	$+45^{\circ}57'20.8800$” N
Longitude	$-78^{\circ}04'16.9284$” E
Height above reference ellipsoid	200.925 m

Table 3.5: Coordinates for a static station used in static test scenario

of New Brunswick, Automatic Precise Positioning Service (APPS) by the Jet Propulsion Laboratory (NASA), and magicGNSS by GMV. To proceed, the user sends a data set to the Precise Point Positioning Software Centre, which returns results to the user in a compressed archive (Banville, 2008). One issue with this approach is that it is not clear what processing parameters are actually used to process the data, and it is not possible to specify processing parameters.

It is therefore assumed that parameters chosen by the PPP Software Centre for all supported data processors are suitable to properly process most data files. Since these processors expect actual data files, simulated data files need to be realistic. The quality of the simulated observables depends directly on the error source models used to generate them. Since most MGOS error source modules employ realistic or quasi-realistic error source models, most of them will be used here. To maximize realism, external data were used as much as possible where appropriate: e.g., the IONEX ionospheric effect module was used instead of the Klobuchar module while satellite coordinates and clock biases were derived using supplied data files (SP3-C file and 30 second clock RINEX file, respectively). Since most PPP processors deal exclusively with GPS data, the GPS module was used. Adopted GPS module parameters are given in Table 3.6.

Parameter Name	Model/ Data Used	Value
Satellite Coordinates	SP3-C file	N/A
Satellite Clock Bias	30 second clock RINEX file	N/A
Orbital errors	Gaussian Noise (Std. Dev.)	1.0 cm (each component)
Clock errors	Gaussian Noise	20 ps

Table 3.6: Global Positioning System (GPS) module parameters for static simulation case

Standard deviation values for orbital and clock errors shown in Table 3.6 are based on the quality of data provided by the IGS. The Gaussian noise model used for orbital and clock errors is a good approximation to true orbital and clock errors (Griffiths and Ray, 2009; Dow et al., 2005). However, this noise model is still only an approximation, as more realistic simulation of these errors is beyond the scope of this work. As with orbital and clock errors, atmospheric conditions at the test receiver location are approximated as well. As stated, ionospheric delay are modelled using the IONEX module. For the tropospheric delay, a degree 50 polynomial is fitted to temperature, pressure and humidity profiles for a station close to the simulation site. The high degree was necessary to make sure that the generated temperature profile closely matched the truth.

The data was obtained from Environment Canada's website for March 31st through April 2nd to make sure that adequate number of points were used to construct the polynomials. This method improves realism of the simulated tropospheric delay; however, there is no absolute need to accurately replicate atmospheric conditions at the time of the test, since PPP processors estimate this error (Héroux and Kouba, 2001). Tropospheric conditions are set as shown in Table 3.7.

Module	Parameter Name	Model/ Data Used	Value
IONEX Ionospheric Module	Stochastic Component	Gaussian Noise (Std. Dev.)	2.0 cm
	Initial Data Epoch Anchor	N/A	Data's time of day
Tropospheric Module	Pressure	Expression	50-degree polynomial
	Humidity	Expression	50-degree polynomial
	Temperature	Expression	50-degree polynomial
	Stochastic Component	Gaussian Noise (Std. Dev.)	1.0 mm

Table 3.7: Atmospheric module parameters for static simulation case

Other site conditions that need to be simulated include: multipath and site displacement due to tides. These effects are modelled using the appropriate MGOS modules whose parameters are shown in Table 3.8. These parameters have been chosen to represent typical conditions one might expect for a static receiver. It is also not possible to precisely set various hardware biases, therefore these biases are set in a quasi-realistic way. Along with various hardware biases, receiver clock bias is chosen arbitrarily as well. Parameters for the static receiver module and hardware bias module are described in Table 3.9.

Most PPP data processors compute estimated coordinates in curvilinear and Cartesian format, along with precision estimations (sigmas) obtained from the adjustment process. Results from the magicGNSS processor lack sigma values. Table 3.10 presents results from the static test for the simulated station and the real station. It can be seen that positioning performance obtained for the simulated station is similar to the performance associated with real data. For most processors, simulated MGOS data produce errors and sigma values are nearly identical to real-world results; however, position errors are somewhat different.

Module	Parameter Name	Model/ Data Used	Value
Ground Bounce Multipath Module	Ground Height	N/A	0.5 m
	Relative Amplitude of Reflected Signal	N/A	10 % of direct signal amplitude
	Receiver Correlator Sampling Interval	Narrow Sampling Interval Method	0.4 Chips
	Stochastic Component	Gaussian Noise (Std. Dev.)	20% of delay amplitude
Minor Error Source Module	Phase windup	N/A	Included
	Moon/Sun tidal offsets	N/A	Included
	Ocean loading	HARPOS file	N/A

Table 3.8: Station related error module parameters for static simulation case

Results show a 2 to 1 or higher ratio in error, which suggests that simulated MGOS data produces results are slightly more biased than when using actual ALGO data.

One possible explanation is that some of the simulated error sources create a vertical bias that could not be removed. One possible source of this bias is the behaviour of the troposphere model used. The Modified Hopfield model used cannot account for azimuthal variations in tropospheric delay, which can lead to mismodelling of the generated tropospheric delay by the PPP processors. In reality, this mismodelling is simply a result of the inherited weaknesses of the Modified Hopfield model. Despite this effect, MGOS results are nearly as good as those obtained with ALGO, which suggests that MGOS is capable of simulating data consistent with centimetre or better positioning.

Module	Parameter Name	Model/ Data Used	Value
Static User Module	Clock Bias	Ramp	Start value: 80 ns End Value: 81 ns Period: 24 h
	Clock Bias Noise	Gaussian (Std. Dev.)	1 ns
Hardware Bias Module	Instrumental Biases	Randomly chosen constants	Receiver bias range: ±0.0 ns Satellite bias range: ±12.0 ns
	Code Biases	Randomly chosen constants	Receiver bias range: ±0.5 ns Satellite bias range: ±0.5 ns
	Code Bias Drift	Linear	Amplitude: 0.5 ns
		Period	24.0 h
	Code Bias Drift Noise	Gaussian (Std. Dev.)	100% of drift amplitude
	Zero-Mean Code Noise	Gaussian (Std. Dev.)	0.5 ns

Table 3.9: Receiver and satellite hardware error parameters for static simulation case

	NRCan		magicGNSS		GAPS		APPS	
Station:	MGOS	ALGO	MGOS	ALGO	MGOS	ALGO	MGOS	ALGO
Latitude Error	-0.1	-0.2	-0.1	0.1	0.7	0.9	-0.2	0.1
Longitude Error	0.1	-0.2	0.1	0.5	-0.3	-0.1	0.2	0.3
Height Error	1.8	0.4	-3.1	-1.3	-2.6	-1.1	-4.0	-0.1
Latitude sigma	0.3	0.3	N/A	N/A	0.2	0.2	0.1	0.1
Longitude sigma	0.6	0.6	N/A	N/A	0.5	0.5	0.1	0.1
Height sigma	1.1	1.1	N/A	N/A	0.9	0.9	0.3	0.3

Table 3.10: Summary of static test results (in cm) from various PPP data processors

3.8 Static Relative Positioning Data Simulation

As another test to evaluate MGOS' realism, simulated data were processed using the relative positioning technique, where simulated data are combined with real data to yield centimetre or better positioning results. The on-line service OPUS (Online Positioning User Service), provided by the U.S. National Geodetic Survey of NOAA, was used (NGS, 2009). OPUS is an automated tool that finds the closest reference stations to the unknown station, and uses data from these stations to compute the coordinates of the unknown station. OPUS results are computed in the ITRF2000 coordinate frame rather than ITRF2005 used by PPP software packages in the previous section. To enable a consistent comparison between ALGO and the MGOS-generated station, MGOS-generated station coordinates are set to match those of ALGO. All other simulation parameters used in the previous section are kept the same for this test. Relevant results of this test are presented in Table 3.11.

Parameter	ALGO	MGOS
X offset (cm)	0.4	0.1
Y offset (cm)	0.5	-4.8
Z offset (cm)	0.6	5.3
X RMS (cm)	2.3	0.7
Y RMS (cm)	1.4	3.4
Z RMS (cm)	0.4	0.9
Number of ambiguities fixed (%)	90	83
Overall RMS (cm)	1.1	2.2
Reference Stations Used	PARY(167.8 km away)	PARY(167.8 km away)
	CAGS (180.8 km away)	CAGS (180.8 km away)
	NRC1 (198.5 km away)	NRC1 (198.5 km away)

Table 3.11: Results of relative positioning static test

Coordinate offsets are computed using ALGO's ITRF2000 reference coordinates obtained from ITRF (Institut Géographique National, 2009). Other statistics are obtained directly from the OPUS solution report. Table 3.11 shows that MGOS simulated files can achieve the expected centimetre position accuracy when processed in relative mode - that is having simulated data combined with real data. In fact, according to the OPUS website (National Geodetic Survey, 2009), the MGOS solution qualifies as a high-quality solution, and can be published as one of OPUS' reference stations. Indeed MGOS simulation capabilities can be further improved to narrow the gap between simulation and reality. For example, one should be able to reduce Y and Z coordinate biases seen in Table 3.11 by adopting a better tropospheric model or a more accurate model of hardware related biases.

It may even be possible to obtain better results by tuning certain simulation parameters. Regardless of the strategies that can be used to improve above results, it is clear that MGOS is capable of simulating realistic static station data. This test together with the previous static test confirm that MGOS can indeed generate realistic static RINEX files, thus fulfiling the goal set out for the static evaluation of MGOS.

3.9 Kinematic Data Simulation

In this test, a moving platform is simulated using a kinematic external RINEX observation file. The chosen kinematic data set corresponds to an airplane that flew with two on-board GPS receivers on May 23, 2002 in the Canadian Arctic. The flight path is shown in Figure 3.24. A kinematic test is conducted in two parts. In the first part, code-only results are compared to code and phase results to show that the filter can process phase observables correctly, and that MGOS phase observable noise has been appropriately simulated. In the second part, a flight trajectory estimated by the filter is compared to one obtained using NRCan PPP software.

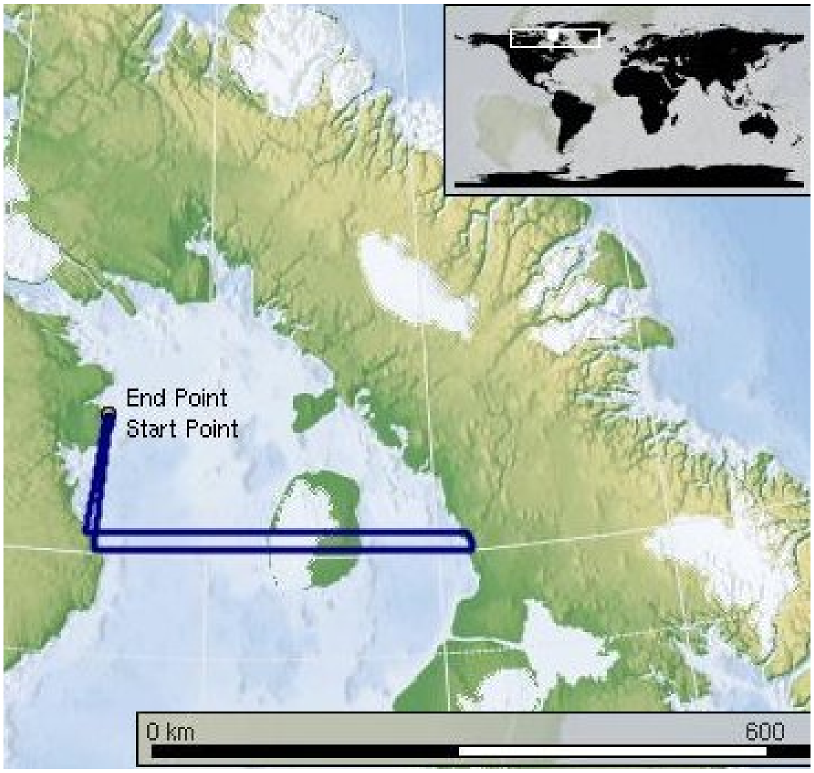


Figure 3.24: Flight path of air-plane in Canadian Arctic extracted from external kinematic data set

To make sure that the two data sets used are equivalent, simulated RINEX file with extracted trajectory was sent to NRCan PPP software for processing. The NRCan processor performs data smoothing when it operates in kinematic mode. Since MGOS' data processor does not smooth its results, this feature had to be disabled from NRCan software to obtained unsmoothed results. The goal of the second portion of the kinematic test is to show that the performance of the implemented data filter in MGOS is on par with the performance of data processors available today. Error source modules used in the static test is reused for the two portions of this test with minor changes. Phase windup and tidal errors were not simulated as they are assumed to be removed by the PPP processor. Receiver clock bias used in the static test is replaced with the one extracted from the original airplane RINEX file. All other error source module parameters are unchanged via-a-vis the static test. The simulated airplane receiver has dual-frequency capabilities for the code-only and the code-and-phase simulations. Therefore the ionosphere-free combination typically used by PPP processors (Héroux and Kouba, 2001) is used here as well.

Figure 3.25 shows position errors obtained with code-only and code-and-phase processing. It is clear that code and phase processing yields smoother and more accurate results. The position RMSEs are given in Table 3.12, with the initial convergence period (which lasts approximately 10 minutes) being excluded to show converged values. Figure 3.25 and Table 3.12 show that generated phase observables are much less noisy than code observables, which is what is observed with real data. Moreover, since the code-only solution is significantly worse than code-and-phase solution, it shows that MGOS' filter is able to realistically process phase observables. Thus it should be said with confidence that phase observable errors have been simulated in a realistic manner for kinematic simulations. To reinforce this point, estimated a priori variances obtained from the filter are plotted in Figure 3.26.

Case	Code-only	Code + Phase
Root Mean Squared Errors (cm)		
North	20	3
East	25	3
Up	58	6

Table 3.12: Position RMSEs for kinematic test

The sudden jump in code-and-phase Up component standard deviation around 2.5 hours seen in Figure 3.26 is a result of a jump in GDOP from around 2 to over 4. Code Up component standard deviation also experiences the same jump. To test the quality of MGOS data processing filter further, the results have been directly compared to those obtained

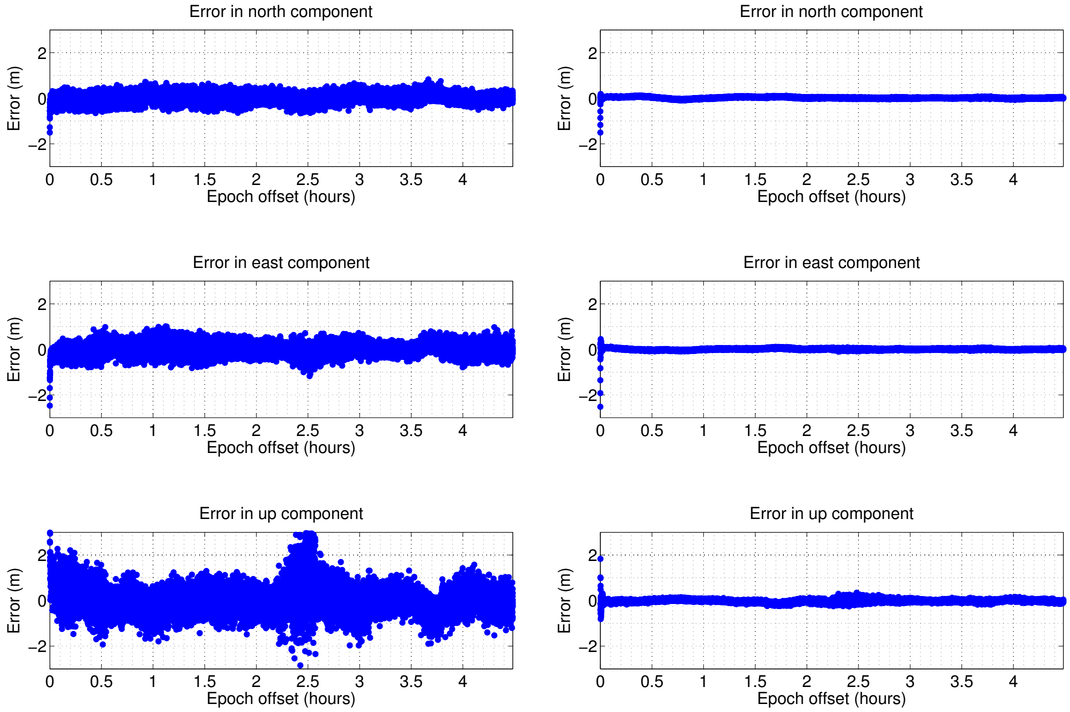


Figure 3.25: Position errors for kinematic test using code-only (left) and code and phase processing (right)

from NRCan's PPP software. Since the two filters have different implementation, it is not possible for either to agree exactly. A metric of performance consists of the structure and magnitude of deviations between MGOS' position errors and NRCan's. Difference between the two sets of position errors are shown in Figure 3.27, with statistics in Table 3.13 (again with initial convergence period excluded).

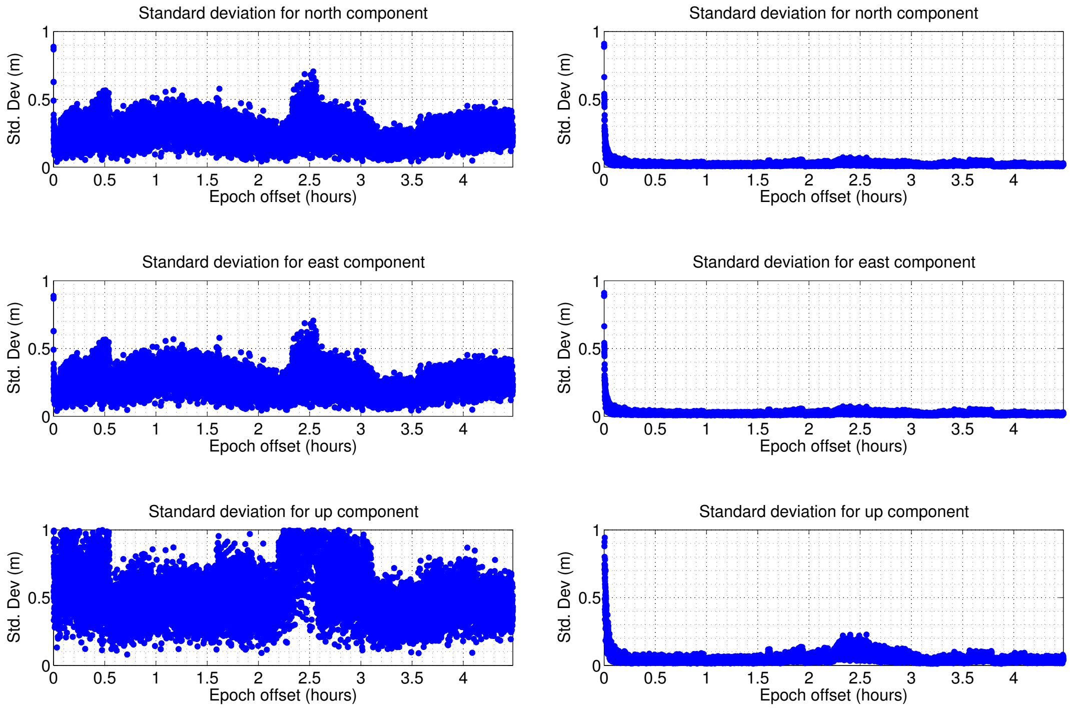


Figure 3.26: Estimated a priory variance for kinematic test using code-only (left) and code and phase processing (right)

It can be seen from Table 3.13 that MGOS' performance is on par with NRCan's. RMS of differences in North and East components are in centimetres, and only modestly larger in the Up component (likely due to the same vertical biases that affect the static results). The mean values from Table 3.12 confirm that MGOS' performance is very good (centimetre or better). Typically, kinematic PPP yields sub-decimetre accuracy in the horizontal, and approximately a decimetre in the vertical. The fact that the RMS difference between MGOS and NRCan solutions is approximately a decimetre in the vertical and sub-decimetre in the horizontal suggests that the two solutions are both within reasonable bounds. Moreover, maximum differences between NRCan's and MGOS' solutions are at most 5 decimetres, which suggests that in the worst case, the two solutions are only decimetres apart. Since

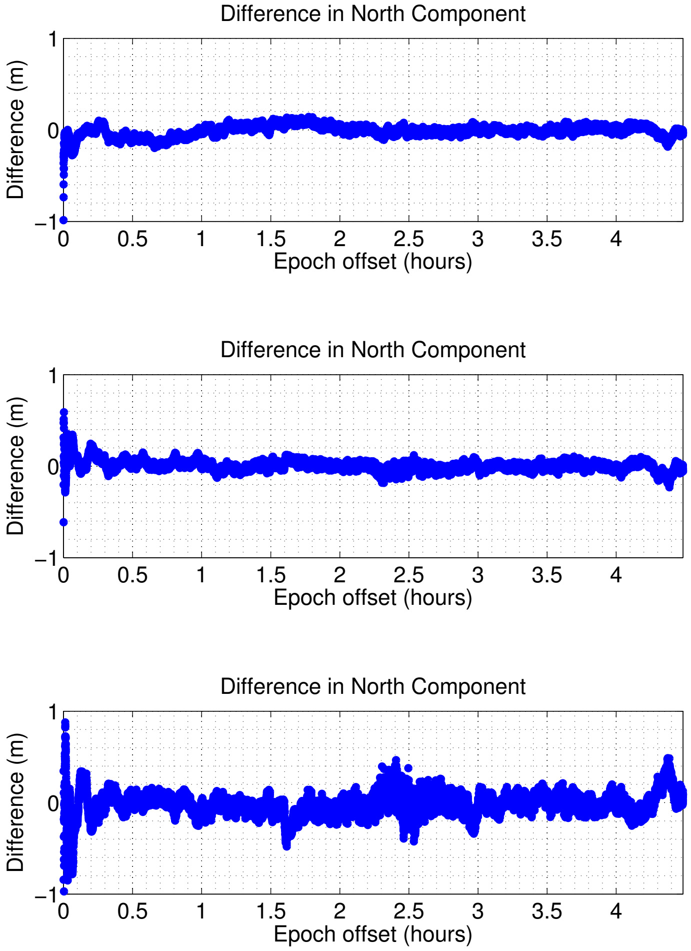


Figure 3.27: Difference between MGOS and NRCan PPP kinematic position error estimates

the NRCan solution is of high quality, the MGOS solution is of high quality as well. Since MGOS does not have as sophisticated a PPP data processing module, it is not surprising that its solution is not as precise as the NRCan solution. Nevertheless, the results of the kinematic tests show that the filter implemented in MGOS can deliver decimetre or better performance when processing kinematic data, which is nearly on par with that of a high-quality PPP software. And kinematic test results also show that simulated code-and-phase observables for a moving receiver are simulated as accurately as for a stationary one.

Parameter	RMS	Bias	Maximum
North Difference (cm)	5.4	0.6	19.8
East Difference (cm)	4.5	0.2	23.0
Up Difference (cm)	10.7	-1.6	48.6

Table 3.13: Position error differences between NRCan and MGOS data filter results

Chapter 4

Earth Orbiting Spacecraft (EOS) Simulations

MGOS capabilities discussed in Chapter 3 will now be used to study point positioning performance for Earth orbiting spacecraft. Many space missions require accurate determination of spacecraft position. The required precision can range from several kilometres to several centimetres depending on the mission. Spacecraft position can be determined with optical observations or laser/radar ranging which can be expensive and/or offer limited availability (e.g., due to sparse distribution of stations on Earth). By the mid 1990s, GPS was recognized as a potential competitor to ground-based spacecraft position determination (Bauer et al., 1998). However, GPS spaceborne equipment may not function well with spacecraft far away from Earth due to communication difficulties or stringent constraints on power and surface area (Lan et al., 2008). Efforts are underway to improve spaceborne GPS receiver performance Kondoh et al. (2009).

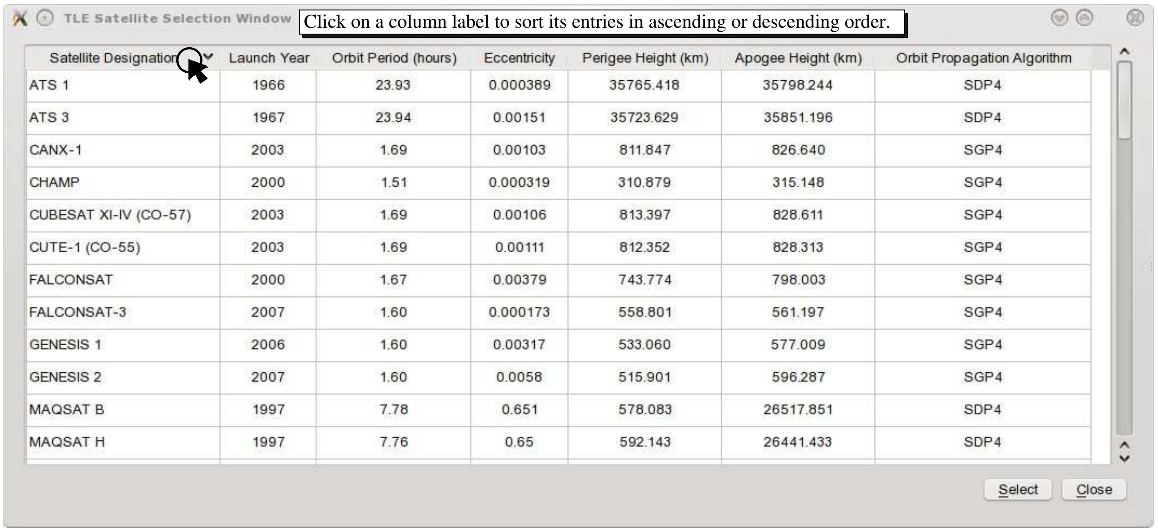
Although significant research has been dedicated to enhancing spaceborne GPS performance, little research has focused on multi-GNSS spacecraft positioning. The reason for this may simply be the lack of multi-GNSS receivers available for use with spacecraft. Nevertheless, with the firm initiative of the Russian government to bring GLONASS back to full operation, as well as efforts being made by the European Union and China, multi-GNSS receivers may well become widely available in the next 20 to 25 years. An important question arises: how will spacecraft positioning change if all pending GNSSs become available? MGOS can be used to address this question. A user module has been created for MGOS to simulate orbital motion of arbitrary satellites. In the next section, the EOS module will be described. In the subsequent section, the results for the spacecraft simulations will be presented and discussed.

4.1 Earth Orbiting Spacecraft Module and Simulations

In this section a detail overview of the Earth Orbiting Spacecraft module and spacecraft simulations are presented. All of the material presented here shall be referenced in the next section where simulation results are presented.

4.1.1 Earth Orbiting Spacecraft Module

The Earth Orbiting Spacecraft (EOS) module is designed to simulate orbital paths of real or fictitious spacecraft from user-supplied Two Line Element (TLE) files. The TLE file format specified by North America Air Defence (NORAD) and is used to store orbital parameters of any known tracked Earth-orbiting object (including large space debris) (T.S. Kelso, 2006). A detailed description of the format can be found in CelesTrack (2004). Values stored in the TLE file are designed to be processed with the so-called SGP4 (for near-Earth) and SDP4 (for deep space) algorithms (CelesTrack, 2004). In many implementations of these two algorithms, a single driver program is used to automatically select between these two algorithms, as described in Hoots and Roehrich (1980). The EOS module can also use these two algorithms interchangeably depending on the selected satellite orbital parameters. The satellite selection window of the EOS module contains all necessary information pertaining to satellites, and allows a user to sort column entries in an ascending or descending order as shown in Figure 4.1.



Click on a column label to sort its entries in ascending or descending order.

Satellite Designation	Launch Year	Orbit Period (hours)	Eccentricity	Perigee Height (km)	Apogee Height (km)	Orbit Propagation Algorithm
ATS 1	1966	23.93	0.000389	35765.418	35798.244	SDP4
ATS 3	1967	23.94	0.00151	35723.629	35851.196	SDP4
CANX-1	2003	1.69	0.00103	811.847	826.640	SGP4
CHAMP	2000	1.51	0.000319	310.879	315.148	SGP4
CUBESAT XI-IV (CO-57)	2003	1.69	0.00106	813.397	828.611	SGP4
CUTE-1 (CO-55)	2003	1.69	0.00111	812.352	828.313	SGP4
FALCONSAT	2000	1.67	0.00379	743.774	798.003	SGP4
FALCONSAT-3	2007	1.60	0.000173	558.801	561.197	SGP4
GENESIS 1	2006	1.60	0.00317	533.060	577.009	SGP4
GENESIS 2	2007	1.60	0.0058	515.901	596.287	SGP4
MAQSAT B	1997	7.78	0.651	578.083	26517.851	SDP4
MAQSAT H	1997	7.76	0.65	592.143	26441.433	SDP4

Figure 4.1: EOS module satellite selection window

Once a satellite has been selected, its estimated orbital path will be simulated. This means that all simulated measurements will be created relative to the estimated position of a selected satellite. Thus as far as the performance of a simulation is concerned, the accuracy of algorithms (or data) used to estimate the chosen satellite position is irrelevant; a great effort has been made to make sure that the implemented SPG4 and SDP4 algorithms adhere to specifications stated in Hoots and Roehrich (1980), resulting in realistic satellite tracks from the EOS module.

To further enhance spacecraft simulation capabilities of the EOS module, algorithms have been created to determine satellite visibility. A signal can be received from a GNSS satellite only when a receiver is facing the satellite. Also, the line-of-sight between satellite and receiver must be such that signal power is at an acceptable level. GNSS satellite antennas have a radiation pattern which limits the range of elevation angle within which a signal can be transmitted. Although MGOS satellite modules do not simulate GNSS satellite antenna radiation patterns, the EOS module has a mechanism to verify if a given satellite is facing a spaceborne receiver, and to determine if the line-of-sight angle is within a predefined limit. Given that all GNSS satellites transmit towards the centre of Earth, and assuming that the primary radiation beam is a right circular cone with half-angle of θ, the following measure is used by the EOS module to determine if a spaceborne receiver can receive signals from a GNSS antenna:

$$cos^{-1}\left(\frac{\overrightarrow{RS}\cdot\overrightarrow{S}}{|\overrightarrow{RS}||\overrightarrow{S}|}\right) < \theta \tag{4.1}$$

In Equation 4.1, $\overrightarrow{RS}$ is the vector from the GNSS satellite to spaceborne receiver, and $\overrightarrow{S}$ is the vector from the GNSS satellite to Earth centre. Equation 4.1 can be easily derived from Figure 4.2, which outlines the proposed method for determining the line-of-sight between a spaceborne receiver and a GNSS satellite.

Signals coming from any GNSS satellite that satisfy Equation 4.1 are considered part of a simulation run while others are rejected. The chosen θ angle is 22.5°, and it is based on the GPS primary lobe (Winternitz et al., 2009). It is beyond the scope of this thesis to include accurate models of various receiver and satellite antenna radiation patterns; however, a user can specify if sidelobes of GNSS signals are to be used. This change allows the user to extend satellite visibility by increasing θ to 30°, which can be useful for some spacecraft receivers (as will be shown later in this chapter). The simulator must also account

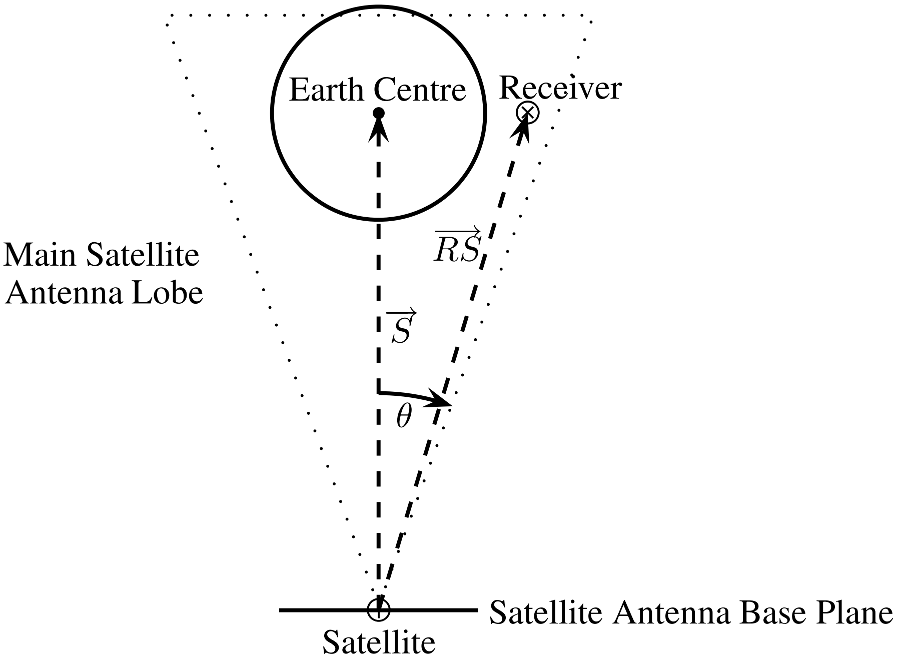


Figure 4.2: Depiction of EOS module method used to validate line-of-sight angles from GNSS satellites to spaceborne receivers

for signal occultation by Earth, when a signal undergoes severe refraction when traversing Earth's atmosphere before reaching the receiver. Current spaceborne GPS receivers filter out such occulted signals. The EOS module can detect both blocked and occulted signals when generating data. Signal categorization is illustrated in Figure 4.3, where R_E is Earth radius and R_{OR} is radius of the occultation zone.

As Both the occultation zone and Earth are assumed to be spherical, which for most EOS simulations, is an acceptable approximation: any signal passing close to Earth's equatorial bulge will be assumed to be in occultation, while signals passing beyond the occultation zone are assumed to undergo negligible refraction. R_{OR} is larger than R_E by 500 kilometres, which means that the chosen occultation zone covers the bulk of the ionosphere, and all of troposphere responsible for refraction. The algorithm used by EOS module to

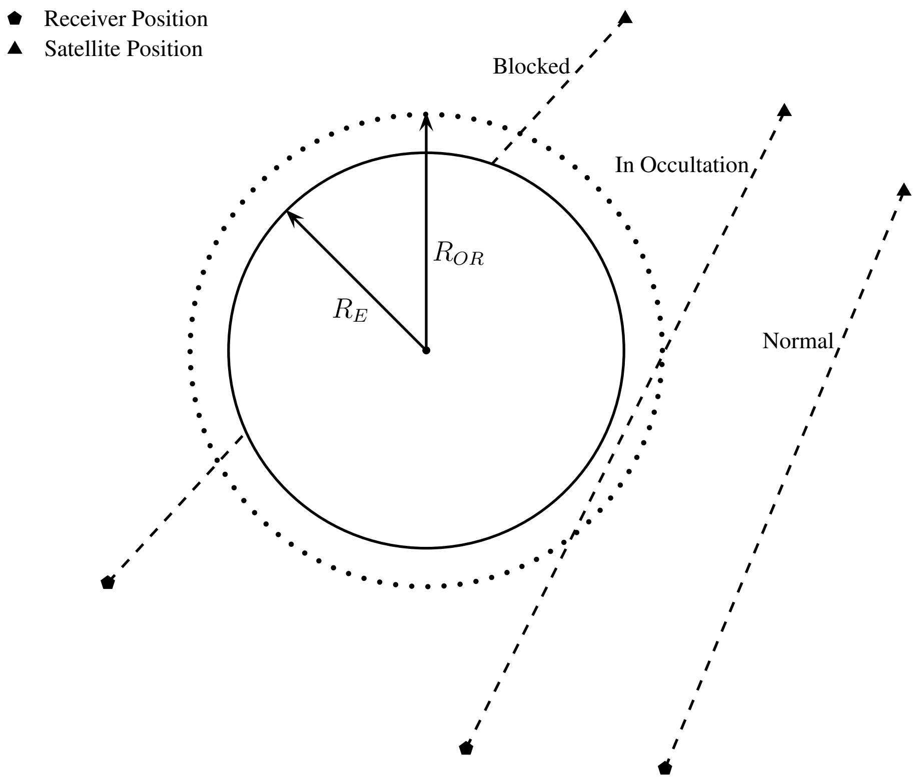


Figure 4.3: Signal categorization used by the EOS module.

categorize a signal is defined in Algorithm A.3.

The question arises as to where the GNSS receiving antenna should point in order to maximize signal tracking. For LEO spacecraft, antenna pointing is similar to that of ground-based receivers: the antenna should be pointed toward local “up”. Spacecraft with larger orbital radii may be required to point their antennas towards Earth to be able to see GNSS satellites (as would be the case with GEO satellites) (Winternitz et al., 2009). Most satellites today are 3-axis spin-stabilized. Some of these spacecraft maintain one side pointed towards the Earth by performing one rotation per orbital period (Fortescue et al., 2003). The EOS module simulates such a spacecraft, and allows a user to orient the receiver as necessary. Antenna orientation occurs in an along-track, cross-track, and radial coordinate system. Spaceborne antennas (and ground-based antennas) are simulated to be insensitive to signals arriving at low elevation angles in order to reduce multipath effects. The elevation angle for a spaceborne antenna is measured from the antenna base plane, which depends on the orientation of the spacecraft. By default, the EOS module orients its GNSS antenna towards zenith. The user can reorient the antenna by setting the along-track (θ_{AC}) and cross-track (θ_{CT}) angles as shown in Figure 4.4.

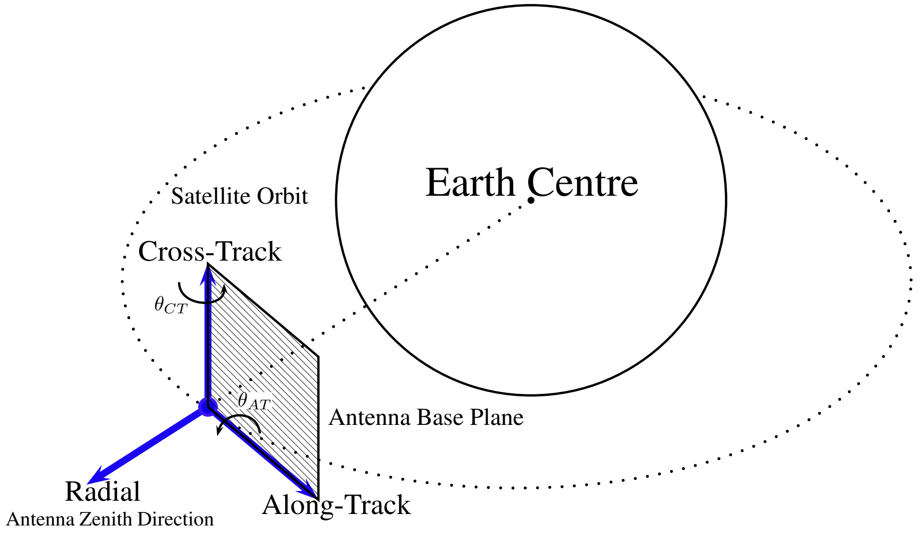


Figure 4.4: Depiction of antenna base plane in spacecraft orbital frame and antenna orientation angles

Since the EOS module can simulate satellites with various orbital radii, some satellites (in particular GEOs) may require a high sensitivity receiver to track GNSS satellites (Winternitz et al., 2009). The EOS module allows the user to specify the weakest signal level (in dBW) that can be processed by a spaceborne receiver. When a signal is transmitted from a GNSS satellite, its signal strength drops off as the square of distance due to free-space loss

(FSL) according to the following (in dB) (Roddy, 2001):

$$FSL = 20log_{10}\left(\frac{4\pi df}{c}\right) \tag{4.2}$$

In Equation 4.2, c is speed of light, d is distance travelled, and f is frequency. FSL is not the only loss that a signal experiences; however, it is one of the largest. Since MGOS does not primarily focus on signal propagation, the EOS module includes a basic model of signal propagation that is used to determine signal strength at a spaceborne antenna. To determine signal power at a receiver antenna, the EOS module uses the following equation (Roddy, 2001).

$$S_P = P_T + G_T - FSL\,[dBW] \tag{4.3}$$

In Equation 4.3, S_P is observed signal power, P_T is transmitted signal power, and G_T is transmission antenna gain. It is beyond the scope of the EOS module to accurately describe signals arriving from all possible GNSS satellites. The module is designed to offer only an approximate treatment of the signal propagation problem. For simplicity, signal frequency and anticipated signal power and antenna gains are chosen to be similar to those of GPS. Note that Equation 4.3 does not contain a receiver gain term. Such a term is not important for EOS module operations since the chosen S_P value defines the minimum signal strength needed to differentiate the signal from background noise. Since FSL needs to be computed quite often in Equation 4.3, a more efficient method has been developed for the EOS module. This method computes the maximum distance from a receiver to a GNSS satellite such that the signal power is still less than or equal to the chosen S_P value. The method is described in Algorithm A.4.

4.1.2 Low Earth Orbit Spacecraft Case

The first spacecraft to be simulated is PICOSat 9, a micro-satellite launched to conduct GPS ionospheric occultation experiments and electromagnetic radio tomography (NASA, 2009). It has a nearly circular orbit with apogee of 791 km and perigee of 789 km. At this altitude, atmospheric errors are negligible, which is the reason for selecting PICOSat 9 as a test subject for this research. Its to-scale orbital track is depicted in Figure 4.5. The satellite is inclined at 67 degrees and can be observed as far north as the Arctic Circle and as far south as Antarctica. It moves in a prograde fashion, completing a revolution in 7.8 hours. Its ground track is presented in Figure 4.6.

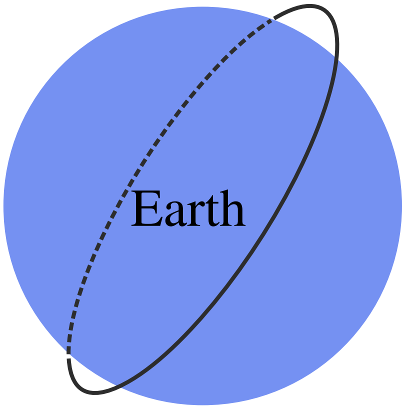


Figure 4.5: LEO PICOSat 9 satellite to-scale orbital track

Given PICOSat 9's orbit, its receiver parameters are similar to those of a ground-based receiver; the notable difference being in dynamics. As a LEO satellite, PICOSat 9's speed is approximately 7 km/s. Such a high speed makes it difficult to acquire GNSS satellite signals due to a large Doppler shift, thus requiring a specialized receiver for this and similar spaceborne missions (Winternitz et al., 2009). To track satellites, PICOSat 9's receiver antenna must point towards GNSS satellites. Since the antenna is attached to the spacecraft's body, it rotates along with the spacecraft, causing some satellites to go out of view.

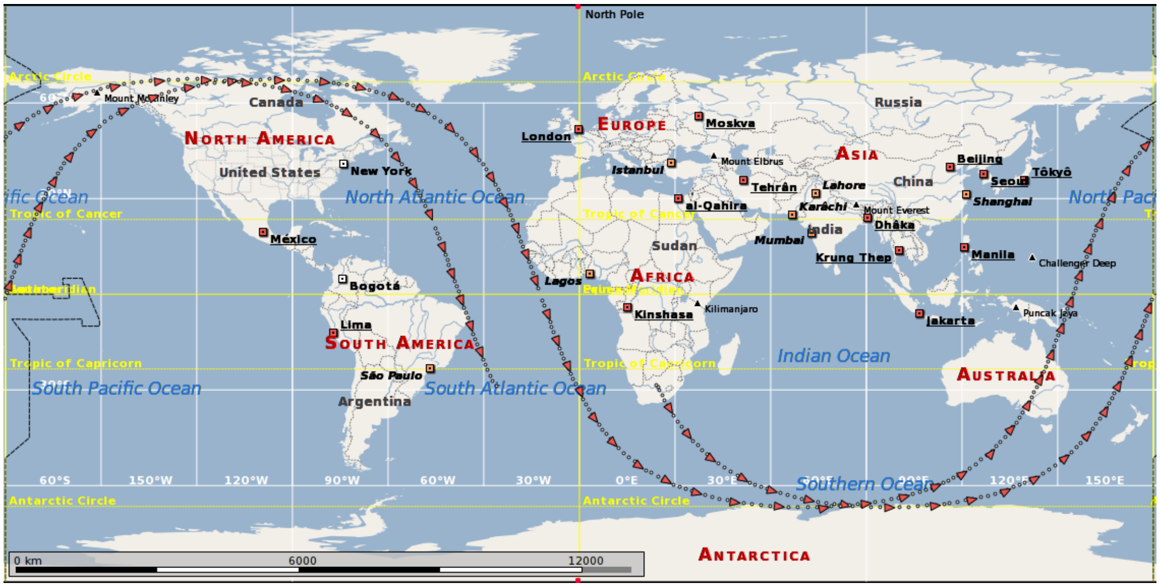


Figure 4.6: LEO PICOSat 9 satellite ground track

For the purpose of LEO simulations here, a hypothetical PICOSat 9 satellite is created. This satellite borrows PICOSat 9's orbital characteristics, and assumes the following spacecraft configuration (parameters for which are given in Table 4.1). The LEO has a synchronous rotation rate with the antenna pointing towards zenith (as shown in Figure 4.4). The receiver tracks all available GNSS satellites except those below a cutoff elevation angle (as viewed from spacecraft zenith direction). Signal power is also a factor in LEO simulations. A dual-frequency receiver is assumed, thus removing the need to accurately model ionospheric error, since the ionosphere-free data combination will eliminate almost all of the ionospheric error. As such, no ionospheric modules are used in any LEO simulation. The choice of minimum signal power is based on signal levels necessary for tracking by a ground-based receiver. Occulted signals are removed by receiver software in order to maximize positioning accuracy. LEO mission results will be presented in section 4.2.

Parameter	Value
Antenna Orientation	Zenith direction / Co-rotating with spacecraft
Elevation Mask	5 degrees
Minimum Signal Power	-185 dBW
Receiver Sidelobe Tracking	Not available
Number of Frequency Channels Tracked	2

Table 4.1: LEO mission parameters

4.1.3 Highly Elliptical Orbit Spacecraft Case

To understand how well a multi-GNSS receiver performs in spacecraft positioning, different types of spacecraft orbits must be examined. EOSs are described by various orbit types ranging from Low Earth Orbit (LEO) to Geostationary Earth Orbit (GEO). Geostationary Earth Orbits are similar to Low Earth Orbits, but with much larger orbital radii. In particular interest for this research is the Highly Elliptical Orbits which are used as transfer orbits from LEO to GEO or used for telecommunication satellites. Molniya or Tundra orbits developed by the former Soviet Union in the 1960s (Montenbruck and Gill, 2000) are instances of Highly Elliptical Orbits. Highly Elliptical Orbits features a high orbital eccentricity (as high as 0.7 for Molniya orbits). A HEO satellite can approach Earth as close as 500 km, yet travel as far as 40,000 km away. Given the fact that HEO spacecraft exhibit a large orbital speed range, it is instructive to evaluate a HEO mission in order to gain insight into multi-GNSS spacecraft positioning. Through a search of NORAD TLE data, the TEAMSat mission has been identified as a suitable mission for study.

TEAMSat was launched in 1997 (Bandecchi and Ockels, 1998) into a low inclination (7.5°), prograde orbit. The satellite is equipped with a multitude of instruments, one of which is a GPS receiver used primarily to determine the number of GPS satellites that can be tracked (Bandecchi and Ockels, 1998). The TEAMSat orbital period is 7.8 hours (about half of that of Molniya). Its orbital eccentricity is 0.65 (about 0.1 less than that of Molniya). Its altitude varies between about 600 km and 26,600 km. Figure 4.7 shows the ground track of two passes of TEAMSat.

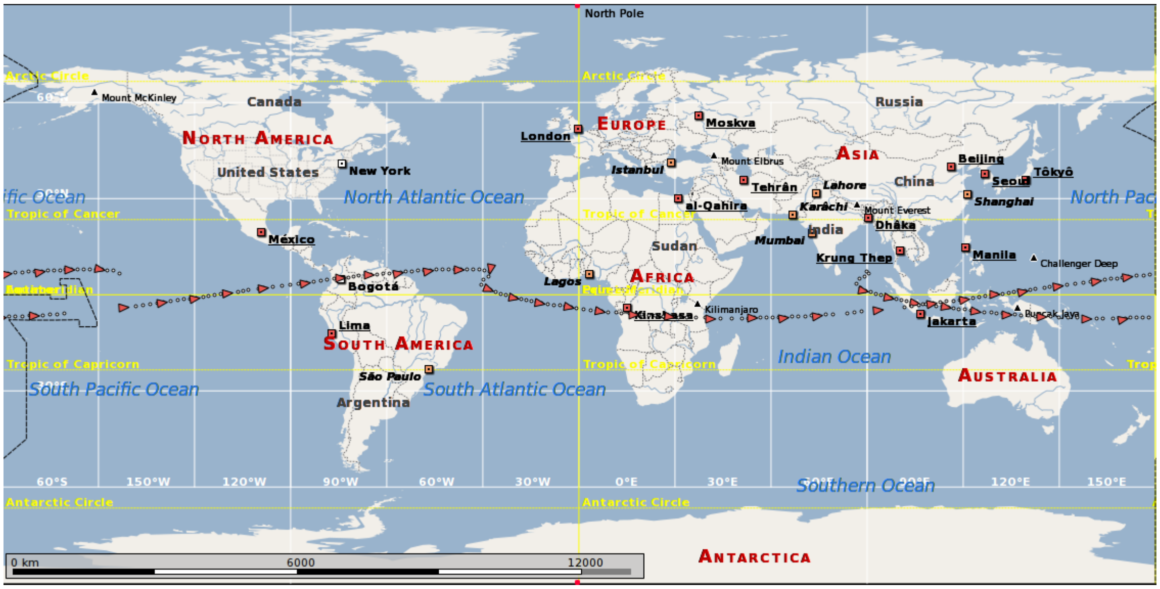


Figure 4.7: HEO TEAMSat and GPS satellite to-scale orbital tracks

Given that GPS satellites orbit Earth at an altitude of approximately 20,200 km, TEAMSat cannot "see" any GPS satellites for a portion of its orbit, if its antenna is pointed toward zenith. On the other hand, once it makes its closest approach to Earth, the antenna must be pointed toward zenith to maximize satellite visibility. This observation creates an interesting challenge for spacecraft designers wishing to install a GNSS receiver on a HEO spacecraft similar to TEAMSat. To resolve this issue, it is necessary to examine the entire TEAMSat's orbital path in relation to the Earth and GNSS constellation. Figure 4.8 provides a to-scale drawing for visualizing TEAMSat's orbital path around the Earth and the GPS orbit.

TEAMSat's antenna should be pointed towards nadir, since there are more satellites in the Earth's direction than away from it. Unfortunately, such antenna orientation present problems when TEAMSat makes its closest approach to Earth, where Earth would block most of GNSS satellites from view. For optimal positioning performance, TEAMSat's antenna should reorient itself automatically during each orbital pass to maximize satellite visibility. However, this approach may not be practical as other vital parts of the spacecraft (such as solar panels) may need to be pointed elsewhere.

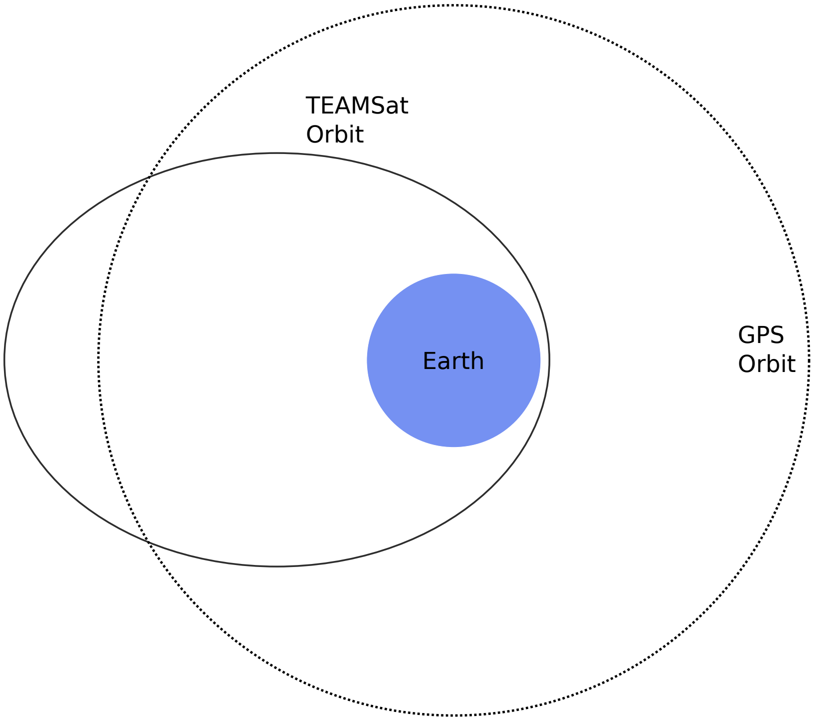


Figure 4.8: HEO satellite flight path

Besides pointing issues, TEAMSat's receiver must also be able to track very faint signals. Since TEAMSat's antenna is to be oriented towards nadir, then most of the satellites observed will appear from the other side of Earth, and those signals will therefore experience large free space loss due to the vast distances they must travel to reach the receiver. Free space loss for TEAMSat can be as large as 190 dB due to satellite's large orbital radius.

Fortunately, the received signal power is above the -185 dB level due to the transmitting and receiving antenna gains. As a result, a minimum signal level will be set to -185 dB.

Since the goal of the HEO simulations is to demonstrate point positioning performance for HEO satellites, it is important to be able to see as many GNSS satellites as possible. Thus TEAMSat's antenna needs to point towards nadir at all times. The overall mission parameters pertaining to TEAMSat are indicated in Table 4.2. Given the fact that GNSS satellites is far away from TEAMSat, the sidelobe tracking feature will be added to TEAMSat's receiver, and tested. Table 4.2 lists the sidelobe tracking feature as "varied" as two simulation runs will be conducted: one in which sidelobe tracking is turned off, and the other where this feature is turned on. A comparison of these runs will help evaluate the extent to which sidelobe tracking can enhance point positioning. Overall results from this mission will be compared to the results obtained from the LEO mission is to assess the overall benefits that multi-GNSS receivers bring to satellite positioning.

Parameter	Value
Antenna Orientation	Nadir direction / Co-rotating with spacecraft
Elevation Mask	5 degrees
Minimum Signal Power	-185 dBW
Receiver Sidelobe Tracking	Varied
Number of Frequency Channels Tracked	2

Table 4.2: HEO mission parameters

4.2 Simulations and Results

This section will introduce the LEO and HEO simulations, their parameters, and results. LEO and HEO simulations are grouped into four cases: GPS-only, GPS + GLONASS, GPS + GLONASS + Galileo, and GPS + GLONASS + Galileo + Compass. Each HEO case will be further subdivided into two sub-cases: with and without sidelobe tracking feature.

4.2.1 Simulation Setup and Parameters

To compare and contrast results obtained through the LEO and HEO simulations, a common simulation set is needed. A simulation set is defined as a group of simulation parameters used to generate simulation results. In the LEO and HEO simulations, relevant parameters include: 1) simulation time span and data sampling rate, 2) GNSS constellations used and their parameters (such as orbital and clock errors), 3) error source modules and their parameters, and 4) spaceborne receiver/antenna parameters (including receiver clock bias).

It is crucial, as well as challenging, to define a simulation set that applies to both LEO and HEO simulations. The goal of these simulations is to determine how orbital dynamics affect positioning results. LEO and HEO receivers are assumed identical, so that positioning performance can be compared in a straight forward manner. Simulation parameters for the LEO case are made identical to those for the HEO case except for time span. In each case one and a half orbital periods is chosen. This time span is long enough to determine if the data processing filter has been able to properly converge to receiver location, and allows for analysis of filter convergence at the start of a simulation run. In terms of time, HEO simulations span 12 hours, while LEO simulations span 2.5 hours.

LEO and HEO receivers operate in real-time mode acquiring and processing data within each epoch update. This implies that each receiver track satellites and compute the GNSS coordinates using their broadcast ephemeris, and no ground support of any kind is required. To conduct all necessary simulations, several assumptions are made about the receiver hardware used in each mission: 1) each receiver tracks as many satellites as possible, 2) the receiver data processor converts spatial and temporal satellite information to a common coordinate and time frame (in this case WGS84), and 3) the code and phase observables are processed.

As ionospheric, tropospheric and tidal errors are negligible for spacecraft, they are not simulated. Other error sources (such as phase windup or relativistic effects) are assumed to be handled by receiver software or neglected due to their magnitude. Only three error source modules available in MGOS are used in spacecraft simulations: 1) ground-bounce multipath, 2) phase ambiguities, and 3) hardware biases.Parameters associated with these modules are indicated in Table 4.3. To improve simulation realism, it is assumed that spaceborne receivers are significantly affected by ground-bounce multipath. Similarly, the hardware biases parameters are assumed to accurately reflect the behaviour of the spaceborne receivers' hardware biases. GNSS satellite code and instrumental biases are based on GPS satellite parameters used in the static MGOS tests presented in section 3.7. Phase ambiguity parameters are based on the phase ambiguity parameters used in static MGOS tests.

Orbital and clock errors (see section 3.5.1) for GNSS are described in Table 4.4. For Galileo and Compass, GNSS orbital errors are assumed to be equivalent to those of GPS' broadcast ephemeris, whose values are based on results shown by Warren and Raquet (2003). In the case of GLONASS, error shown are assumed to reflect internal capabilities of the receiver hardware (see section 3.5.1). Given current MGOS simulation capabilities, orbital errors for GPS, Galileo and Compass are simulated as noisy sinusoids. For GLONASS, Gaussian noise will be simulated. Clock errors will be assumed to be Gaussian white noise, since MGOS is not capable of setting it in any other way.

Module	Parameter	Value/Range
Ground-bounce multipath	Ground height	0.1 m
	Relative amplitude of reflected signal	10 %
	Receive correlator method	Narrow Sampling
	Receiver sampling interval	0.4 Chips
	Relative noise amplitude	30 %
Phase ambiguities	Ambiguity	± 5000 cycles
	Cycle slips	None
Hardware biases	Satellite instrument bias	± 12.0 ns
	Receiver code bias	± 1.0 ns
	Satellite code bias	± 0.5 ns
	Receiver code bias drift term mode	Constant
	Receiver code bias drift term amplitude	10.0 ns
	Receiver drift noise ratio to amplitude	50 %
	Zero-mean code noise standard deviation	5.0 ns

Table 4.3: Spacecraft simulations error source module parameters

Error	GPS, Galileo and Compass		GLONASS	
Orbital	Parameter:	Value	Parameter:	Value
	Shape:	Sinusoid	Shape:	None
	Amplitude:	1.5 m	Amplitude:	0 m
	Noise Std.Dev:	0.05 m	Noise Std.Dev:	1.00 m
Clock	Parameter:	Value	Parameter:	Value
	Shape:	None	Shape:	None
	Amplitude:	0 ps	Amplitude:	0 ps
	Noise Std.Dev:	50 ps	Noise Std.Dev:	50 ps

Table 4.4: Orbital and clock error parameters for all participating GNSSs

Finally, data sample rate depends completely on spaceborne receiver electronics. For LEO and HEO, all processing will be done on-board. Since PICOSat 9 and TEAMSat move at high speed, the data sampling rate must be selected with care. If the data sampling rate is too low (say lower than 0.033 Hz), the on-board data processor may not be able to quickly and correctly recover its position after a loss of lock to GNSS satellites. For both LEO and HEO missions, the data sampling rate will be set to 0.2 Hz. This value is chosen as a compromise between rapid data availability and hardware capabilities of receivers.

4.2.2 Satellite Visibility Results and Analysis

It is expected for PICOSat 9 and TEAMSat to observe different numbers of satellites due to their different orbits. As part of its output, MGOS plots satellite visibility for the entire simulation run. In Figure 4.9, satellite visibility for the LEO case and HEO case without sidelobe tracking are plotted.

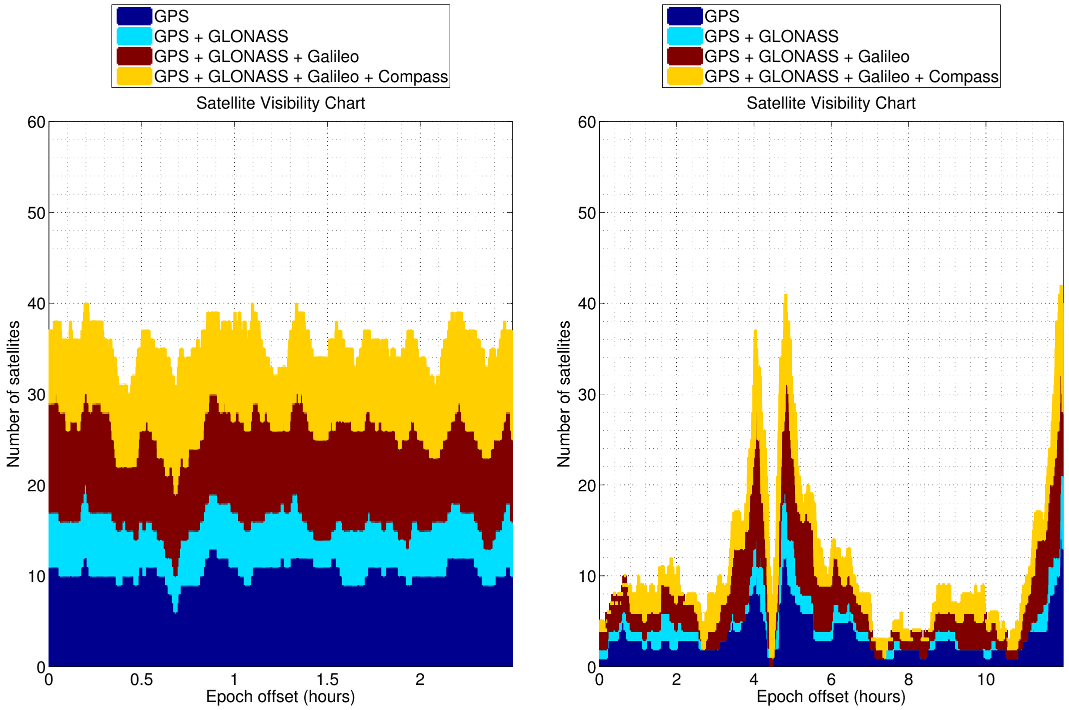


Figure 4.9: Satellite visibility plots for LEO case (left) and HEO case (right) without sidelobe tracking

PICOSat 9 tracks anywhere between 6 (GPS-only simulation) to 40 satellites (GPS + GLONASS + Galileo + Compass simulation). It is informative to note that for all PICOSat 9 simulations, satellite visibility remains well above the minimum (of 4 satellites) required to compute a position. PICOSat 9 hardly ever sees a constant number of satellites for more than 10 minutes, due to its speed. Evidently TEAMSat tracks satellites very differently from PICOSat 9. Although TEAMSat is able to track 39 satellites for brief intervals during the last portion of the fifth simulation hour, for the majority of its orbit it tracks fewer than 10 satellites. Moreover, TEAMSat's satellite visibility plot shows abrupt changes in the number of visible satellites during the fifth simulation hour. At one stage the number of satellites drops to zero for the GPS-only case. The significant variability in TEAMSat visibility can be explained using Figure 4.10.

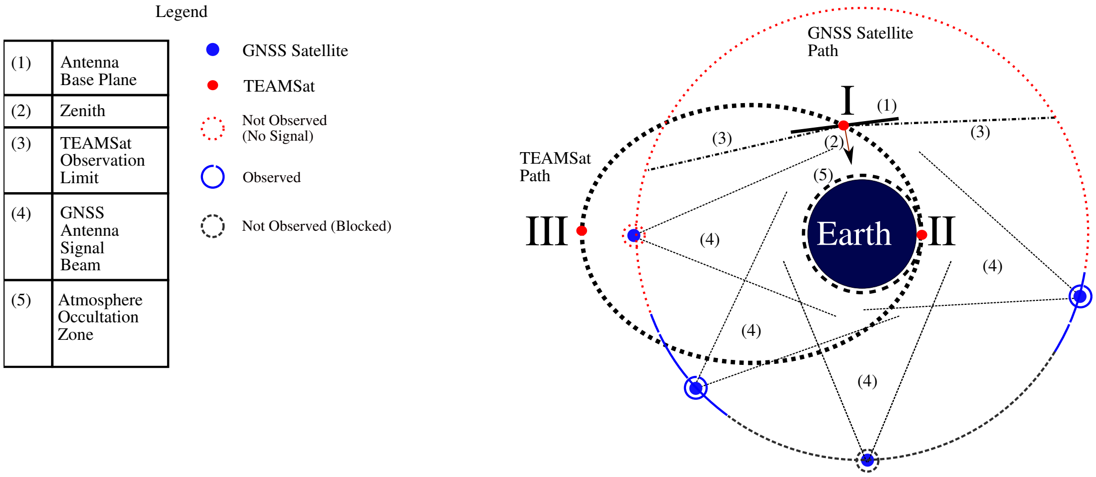


Figure 4.10: Depiction of distribution of visible satellites for the HEO case

TEAMSat is able to track a great many satellites when it is just about to pass its perigee point (point "I" in the figure), when it can "see" GNSS satellites on both of its sides, and GNSS satellite beams are oriented towards TEAMSat. This maximizes the number of visible satellites, leading to a rapid increase in satellite visibility. As TEAMSat approaches point "II", visibility drops to nearly zero, and is a result of TEAMSat's GNSS antenna orientation. At point II, the TEAMSat antenna is pointing towards nadir, effectively looking away from all GNSS satellites. This event lasts very briefly as TEAMSat's speed is at its maximum. At point "III," TEAMSat is oriented towards nadir; however, the majority of GNSS satellites are broadcasting towards the Earth, with only a few of them actually oriented towards TEAMSat. This occurs at the start of simulation runs, and explains why TEAMSat visibility of GNSS satellites is initially poor. As described in section 4.1.1, sidelobe tracking allows TEAMSat's receiver to track GNSS satellite signal sidelobes, effectively increasing the total number of visible satellites. Figure 4.11 shows the significant increase in the number of satellites that TEAMSat's receiver is able to track with sidelobe tracking turned on.

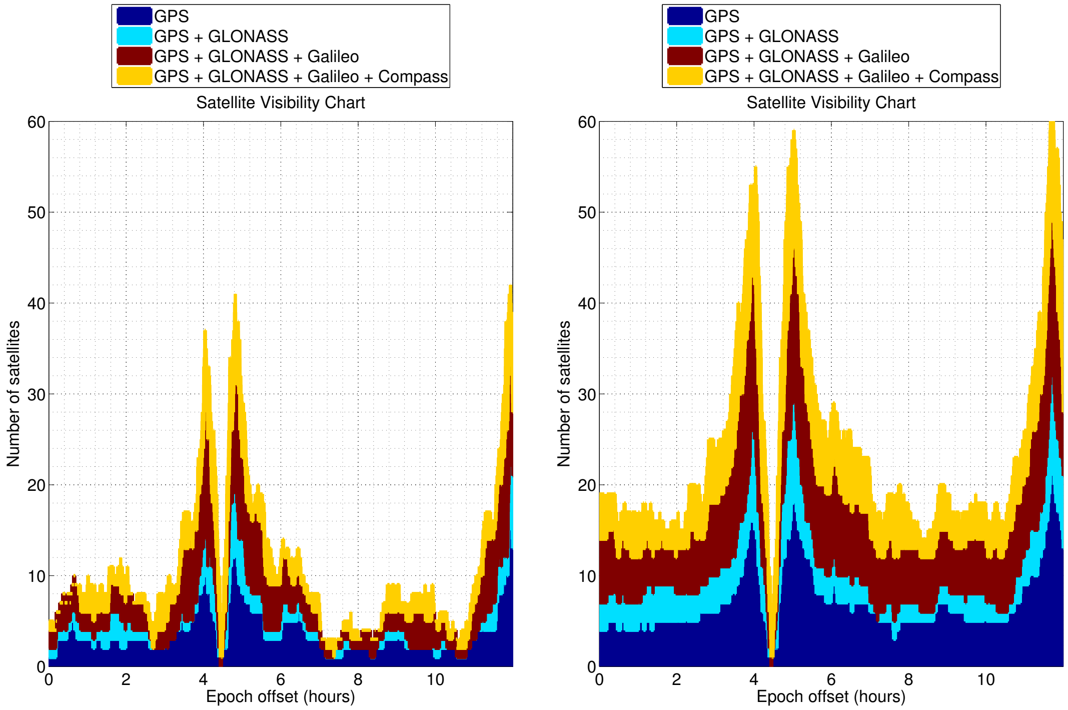


Figure 4.11: Satellite visibility for the HEO case without (left) and with (right) sidelobe tracking

4.2.3 GPS-only Results

The GPS-only results will serve as the baseline for analyzing further results. 31 GPS satellites are assumed - the constellation at the time of writing, and simulated data are processed with the multi-GNSS data processor module (described in section 3.6). To enhance realism, the GPS constellation is defined using a precise ephemeris and clock file. Position errors from simulated data are defined in the antenna coordinate frames (as described in section 4.1.1). Figure 4.12 shows position errors for the LEO case and the HEO without sidelobe tracking case and reveals three important differences between the two.

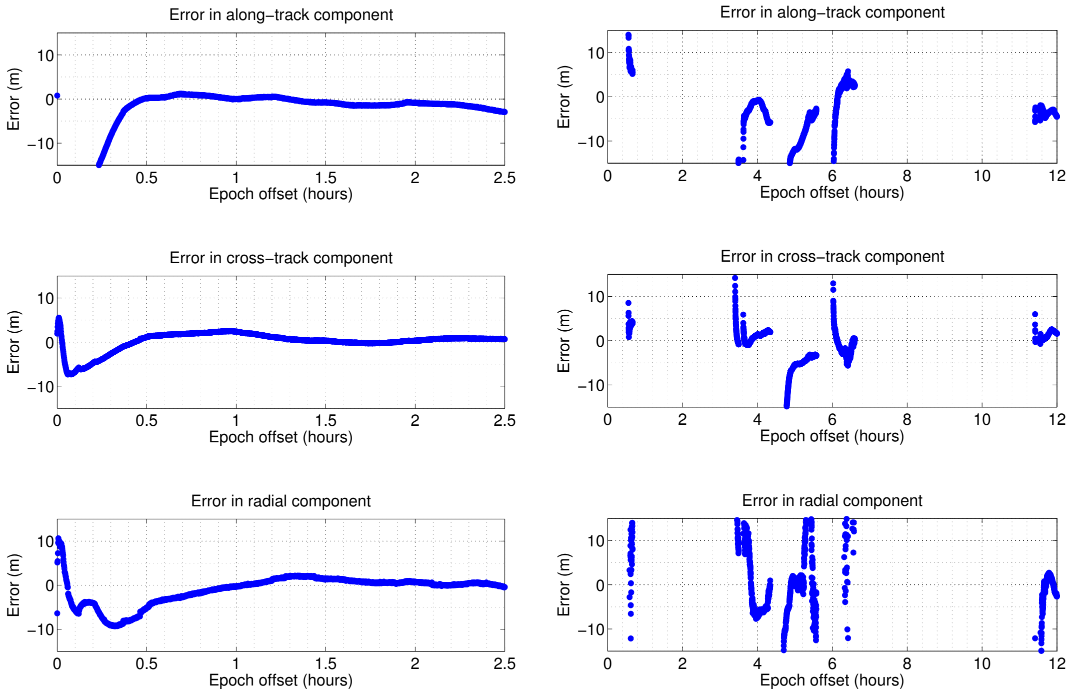


Figure 4.12: Position errors based on observing GPS satellites in LEO case (left) and HEO case (right) without sidelobe tracking

The first difference is that the simulated LEO satellite is able to continuously determine its position for the entire duration of its orbit, while the HEO satellite cannot effectively determine its position, as large gaps can be seen in the plotted position error. These gaps are a result of either an inadequate number of satellites being tracked (as can be seen Figure 4.9), or failure of the data processing filter to converge to a solution. The second difference is in the position errors themselves. In case of the LEO, position errors decrease in time, thought for along-track and cross-track components, position errors exhibit low frequency wave-like patterns after the initial convergence period.

The wave-like structures do diminish closer to the end of the simulation, suggesting that the filter has finally converged onto PICOSat 9's position. In the HEO case, the filter appears to converge at different rates for each position component. For the radial component, the filter has a very difficult time converging, since the temporal variation in the error resembles more of a static noise than exponential decay. The third difference lies in the variation of position errors. For the LEO, position errors tend to vary very slowly after the initial convergence period (which is about 30 minutes). HEO position errors behave more erratically, with larger deviations from zero.

As can be seen in Figure 4.13, the LEO's GDOP values are significantly smaller than those of the HEO's (hence the need for the semi-log scale). GDOP values for the LEO case indicate that the geometry is relatively good (2 or less) - on par with what is expected for ground receivers. Unfortunately, GDOP does not provide clues as to why position errors are drifting slowly during the later portions of LEO's simulation run. One possibility is that the data processing filter does not estimate state covariance adequately, resulting in position estimates that deviate significantly from the true values. This possibility will be evaluated later in this section. In the HEO case, GDOP values clearly show that it is not possible for the data filter to produce adequate position estimates, since the geometry is very poor. As a result, GPS alone is not adequate for reliable positioning of a TEAMSat-like spacecraft.

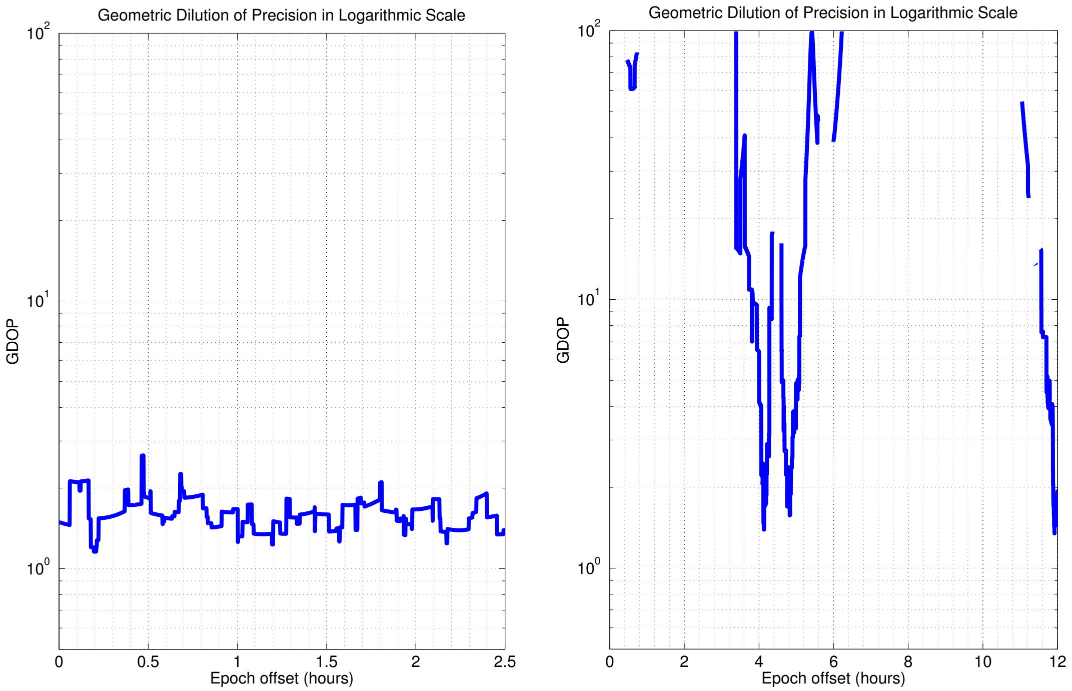
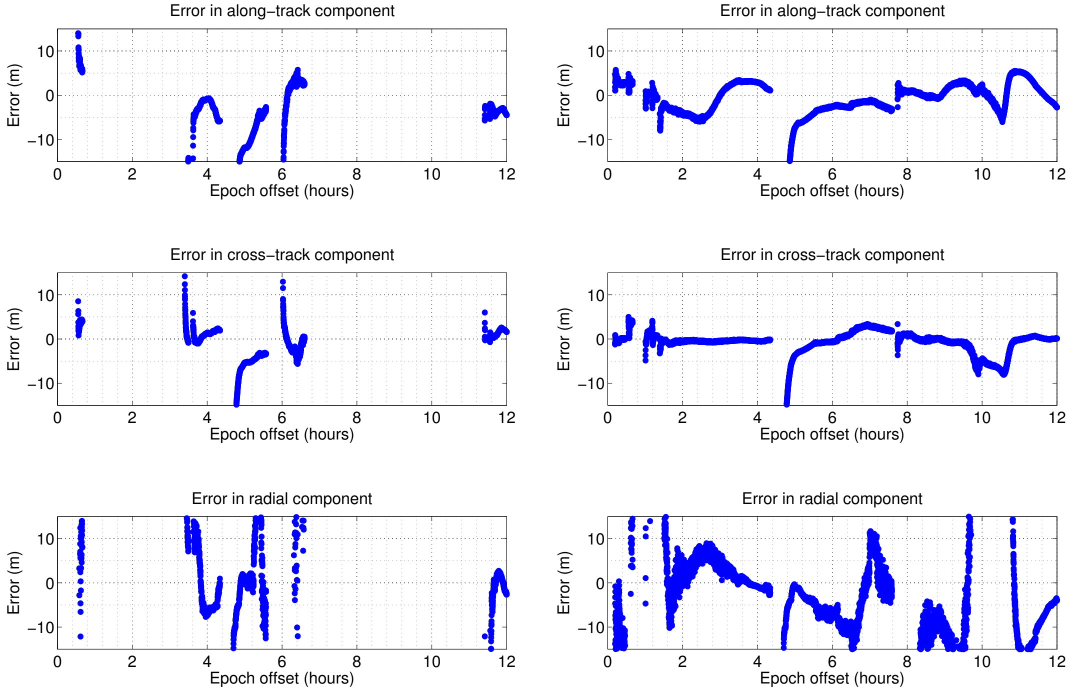


Figure 4.13: GDOP values based on observing GPS satellites in LEO case (left) and HEO case (right) without sidelobe tracking

The next question is to what extent sidelobe tracking can improve HEO results? As mentioned in section 4.1.1, sidelobe tracking widens the effective GNSS satellite antenna beam to 30°, based on the width of the primary sidelobes of GPS satellite antennas. Figure 4.14 compares HEO position errors with sidelobe tracking deactivated and activated. It is evident that sidelobe tracking is of great benefit: position estimates are available for a much longer period (approximately 420% longer); and, estimated positions are much more accurate, since position errors have been measurably reduced for all components.

Figure 4.15 shows a significant improvement in GDOP values if sidelobe tracking is available, which reflects itself in position accuracy. Nevertheless, the HEO's geometrical strength is still about an order of magnitude worst (even with the sidelobe tracking feature) than that of the LEO's, suggesting that a GPS-only receiver would require the assistance


Error in along−track component

Error in along−track component

Error in cross−track component

Error in cross−track component

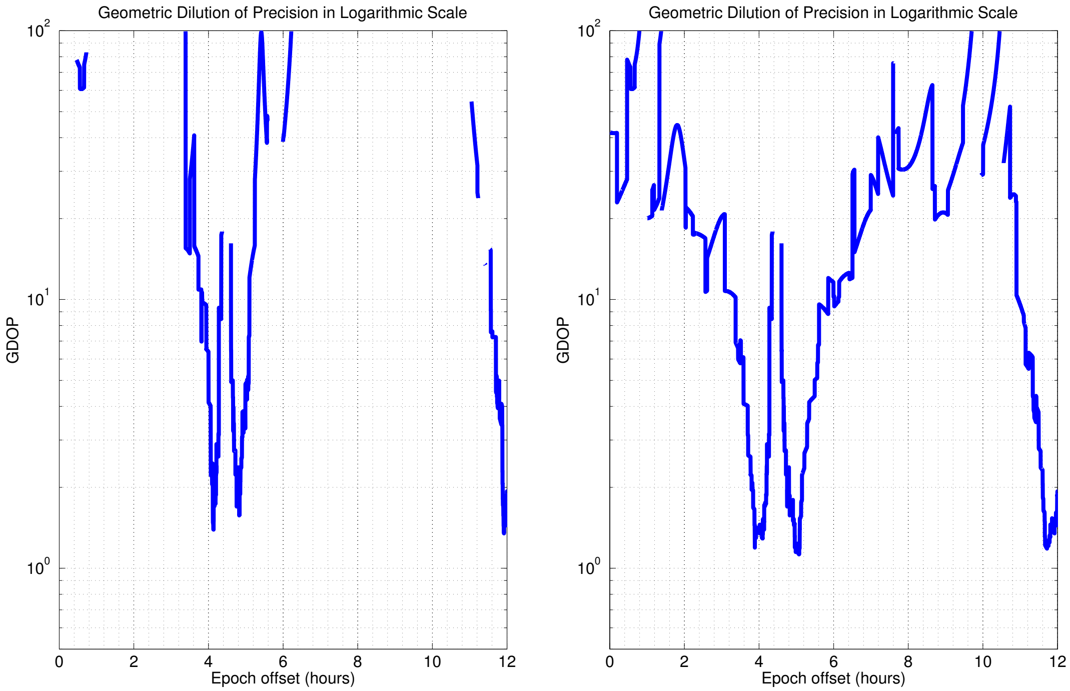
Error in radial component

Error in radial component

Figure 4.14: Position errors based on observing GPS satellites in the HEO case without sidelobe tracking (left) and with sidelobe tracking (right)

of external sensors (such as inertial sensors) to provide better positioning. Unlike the LEO case where continuous positioning can be made with GPS alone, the HEO requires more than one GNSS for adequate positioning.

To compare the results presented thus far, the position Root Mean Squared Errors (RMSEs) for carefully chosen time periods are computed and presented in Table 4.5. For the LEO, the chosen time period is from 0.7 to 2.5 hours, as the filter has reached (or is very close to reaching) a steady-state. For the HEO without and with sidelobe tracking, two time periods will be examined: the first from 5.0 to 5.5 hours, and second from 11.6 to 12.0 hours, as they are the only intervals with adequate positioning estimates. Table 4.5 shows that it is “easier” to position oneself in a LEO than an HEO environment. The position error in LEO simulations is at the metre-level 30 to 45 minutes following acquisition of GPS


Geometric Dilution of Precision in Logarithmic Scale

Geometric Dilution of Precision in Logarithmic Scale

Figure 4.15: GDOP values based on observing GPS satellites in the HEO case without sidelobe tracking (left) and with sidelobe tracking (right)

signals. Most of the error is due to satellite orbit and clocks, since they have the largest impact on positioning. On the other hand, HEO positioning greatly benefits from sidelobe tracking, with RMSE values being nearly comparable to the LEO RMSE over the majority of TEAMSat's orbit. Without sidelobe tracking, HEO positioning may still be acceptable (depending on the application); however, position cannot be established over much of the orbital track. The cross-track RMSE of the sidelobe tracking enabled HEO for the 11.6 - 12.0 hours period is inconsistent with all other RMSEs. A likely cause for this anomaly is a sudden change in geometry that favour the cross-track component during the relevant interval. Due to its very small (sub-metre) magnitude, the cross-track RMSE for sidelobe tracking enabled HEO is treated as an outlier, and will be regarded as anomalous when comparisons are made against upcoming simulation scenarios.

Case	LEO	HEO (Without Sidelobes)		HEO (With Sidelobes)	
Time Period	0.7-2.5 hrs	5.0-5.5 hrs	11.6-12.0 hrs	5.0-5.5 hrs	11.6-12.0 hrs
Root Mean Squared Errors (m)					
Along-track	1.18	8.42	3.57	5.67	1.35
Cross-track	1.18	4.51	1.57	2.42	0.12
Radial	1.09	17.74	6.38	3.14	6.16

Table 4.5: Position RMSEs for LEO and HEO cases involving GPS

Since it is not possible to obtain true position errors when estimating a receiver location with real data, one has to rely on a suitable data processor while estimating position uncertainty. A sequential least-squares filter can estimate state parameter covariance, which is the only information available to gauge solution precision. A relevant question is "How accurate are the estimated position errors versus the true errors?" MGOS can accurately answer this question since the true position errors are known. A comparison can be made as follows:

$$\overrightarrow{\Delta} = 3\overrightarrow{\sigma} - \overrightarrow{\delta P} \tag{4.4}$$

$\overrightarrow{\delta P}$ is the vector representing absolute true position errors, and $\overrightarrow{\sigma}$ is the standard deviation vector of coordinate components obtained from the sequential least-squares filter. Assuming that $\overrightarrow{\sigma}$ adheres to a Gaussian random distribution, $3\overrightarrow{\sigma}$ represents a confidence level exceeding 99.7 percent. Using Equation 4.4, Figure 4.16 compares filter performances of LEO and sidelobe-activated HEO simulations. Due to the large gaps present in sidelobe-deactivated HEO simulations, the latter case is not included in the comparison.

Figure 4.16 can be interpreted as follows: A positive value indicates that the sequential least-squares filter has overestimated the position error vis-a-vis the true error. A zero value means that the filter has estimated the position error perfectly. A negative value means that

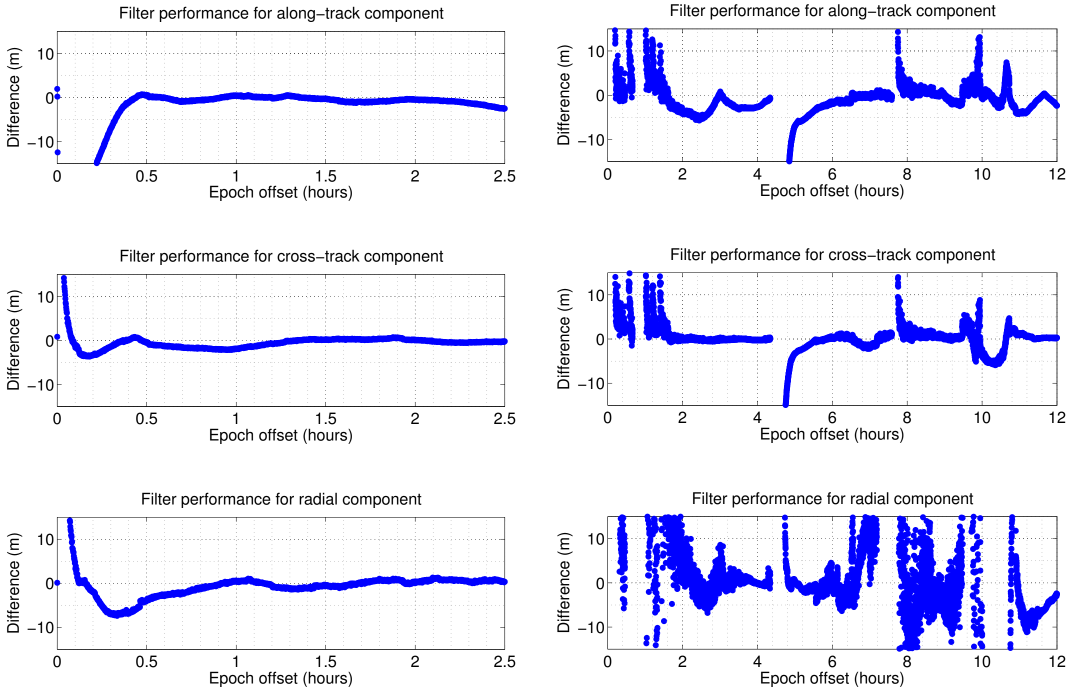


Figure 4.16: Filter performance based on observing GPS satellites in LEO case (left) and HEO case (right) with sidelobe tracking

filter has underestimated position error. In the LEO case, during filter convergence, the filter greatly underestimates along-track position error, while it overestimates the cross-track and radial position errors. These differences occur, because while converging, the filter's a priori covariance is inaccurate, which leads to poor position error estimates. Over time, filter covariance becomes more accurate thereby improving position error estimates. Figure 4.16 also shows that as with ground receivers, the radial component has the weakest geometric strength, since it takes slightly longer for that component solution to converge due to GNSS receiver/satellite geometry (i.e., no GNSS satellites below receiver). After convergence, the filter performance is very good, since the estimated position errors (at the 3σ level) are similar (within a metre) to true position errors.

The same cannot be said about the HEO case, where even with sidelobe tracking, filter performance is quite poor resulting in the HEO not being able to correctly estimate position uncertainty over a majority of the simulation. This is especially true for the radial component, where this poor performance is caused by an unstable number of usable satellites, which in turn causes the filter to reset and to re-estimate its position. Also, the geometrical strength is be much poorer in the HEO case than the LEO case, which further degrades filter performance. Since the filter relies on consistent data acquisition to accurately estimate position uncertainty, any data gaps will cause it to reinitialize. During this time, as with the LEO case, position uncertainty estimates are very poor.

4.2.4 GPS + GLONASS Results

To allow a direct comparison with results in the GPS-only case, all specifications of the receiver will be the same as before, and the same antenna frame will be used. The LEO results shown in Figure 4.17 are very similar to those presented in Figure 4.12. One immediate difference is that the GPS + GLONASS solution is noisier than the GPS-only solution, which is partly due to slightly larger orbital errors for GLONASS satellite and its relatively poorer geometry. On the other hand, antenna along-track and cross-track component errors are very similar, suggesting that GLONASS does not enhance geometrical strength for these two components. During the first half hour of the simulation, the radial error grows to a value of approximately -15 metres (~5 metres more than for the GPS-only case) and takes approximately 6 minutes longer to converge. Despite this behaviour, after the filter converges, position errors remain more closely bound to zero, and do not “wander off” as the case with GPS-only position errors.

The wave-like structures seen in the LEO position errors for simulation times greater than 0.5 hours seen in Figure 4.17, these are significantly reduced in comparison with similar fluctuations in Figure 4.12. The addition of GLONASS satellites appears to provide a more significant benefit to HEO positioning. As can be seen in Figure 4.17, position estimates are now available much longer than with GPS-only tracking and processing. For example, between 0.2 to 2.1 hours, it is not possible to estimate position using the GPS constellation alone. With GLONASS tracking added, it becomes possible to estimate TEAMSat's position. Nevertheless, the estimated position is still poor (as is the case throughout the entire simulation) and likely not adequate for high precision applications.

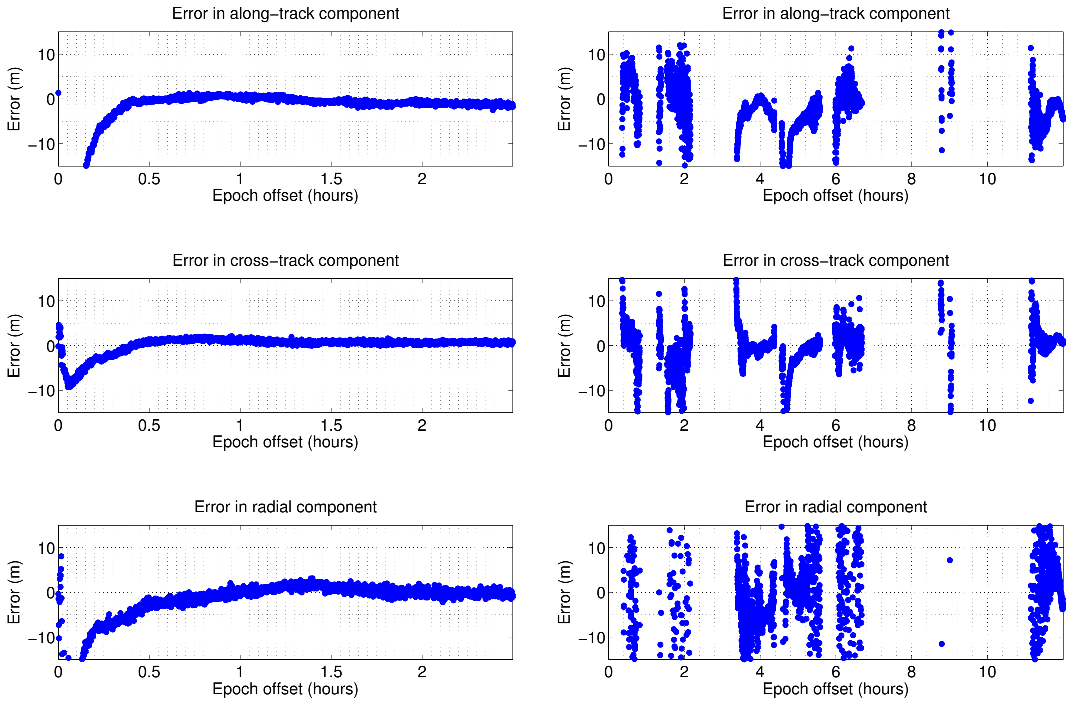


Figure 4.17: Position errors based on GPS and GLONASS satellite observations for LEO (left) and HEO without sidelobe tracking (right)

GLONASS should also improve the geometrical strength beyond that obtained by tracking GPS satellites alone. As can be seen in Figure 4.18, mean LEO GDOP values have been actually reduced from about 1.8 to 1.4. Although not a significant improvement, it shows that the slight reduction in convergence time noted earlier is not due to reduction in geometric strength. A more likely explanation is that measurement noise has increased when GLONASS was added (due to larger orbital errors), which negatively impacted the data filter. In the HEO case, GLONASS provides a more significant improvement in geometry. For example, near the end of the sixth hour, GDOP drops from approximately 100 to 40. Although the HEO results are very poor, they nevertheless shed light on how multi-GNSS tracking and data processing can assist in situations where satellite visibility might be poor.

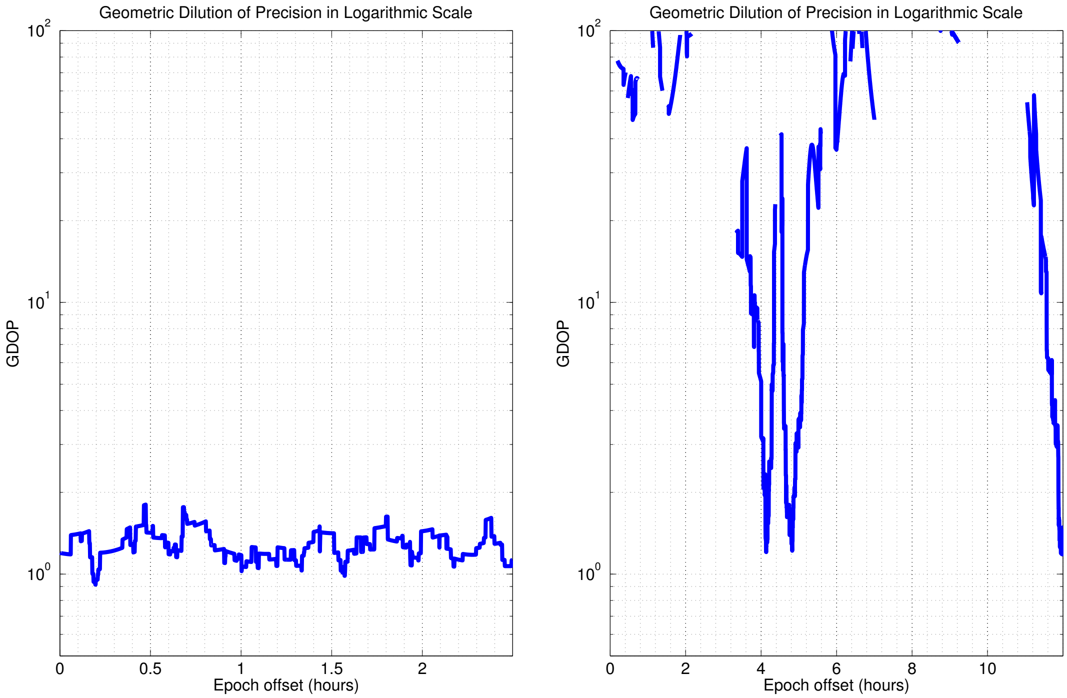


Figure 4.18: GDOP values for the LEO (left) and the HEO with sidelobe tracking turned off (right) observing GPS and GLONASS satellites

From Figure 4.19, it is immediately evident that sidelobe tracking drastically improves satellite visibility. Compared to the simulation where without sidelobe tracking only ~60% of positions were estimated, now 99% of the entire TEAMSat's orbital path is estimated. This significant improvement is a result of improved satellite visibility (as can be seen in

Figure 4.11). Solution precision has also benefited from the increased number of visible satellites, since for the majority of the simulation (~90%), position errors for along-track and cross-track components remain within ±10 metres. As with the GPS-only scenario, the radial component has a error which can exceed 15 or more metres and does not improve with addition of GLONASS satellites (in fact, in general, it is degrade by a factor of ~2 - likely due to higher orbital errors). Figure 4.20 illustrates that the geometrical strength has nearly doubled with the addition of sidelobe tracking. Most GDOP values stay bounded within a value of 50. This improvement reflects itself in the positioning performance reported above. Although positioning performance for the HEO is still relatively poor, it nevertheless would be adequate for less demanding missions.

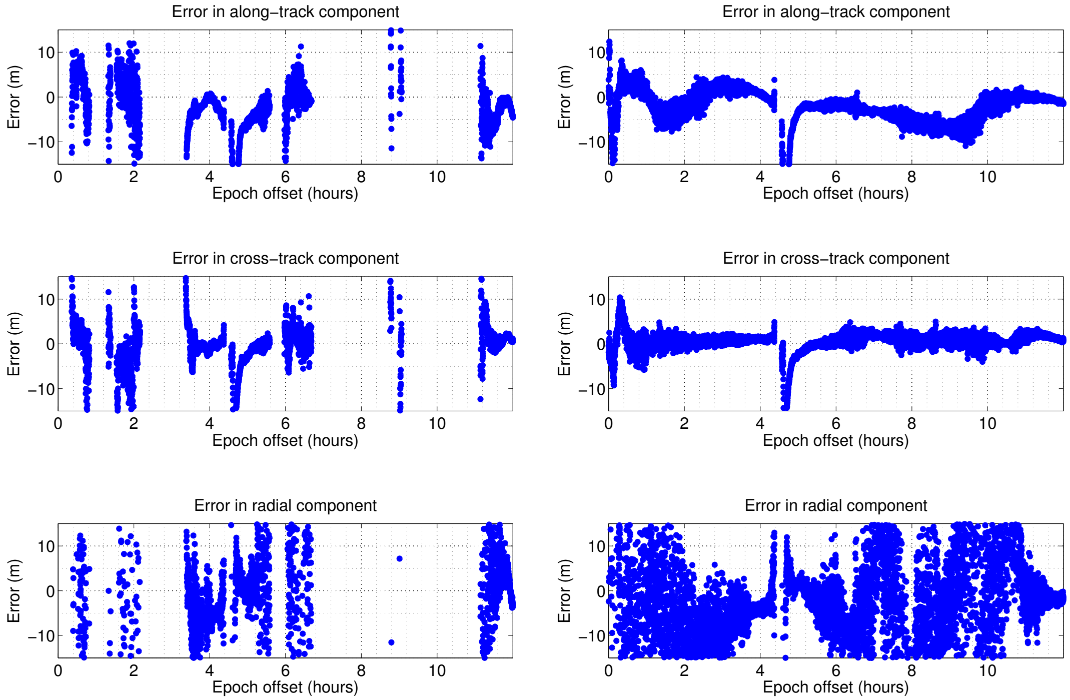


Figure 4.19: Position errors for HEO case without sidelobe tracking (left) and with sidelobe tracking (right) observing GPS and GLONASS satellites

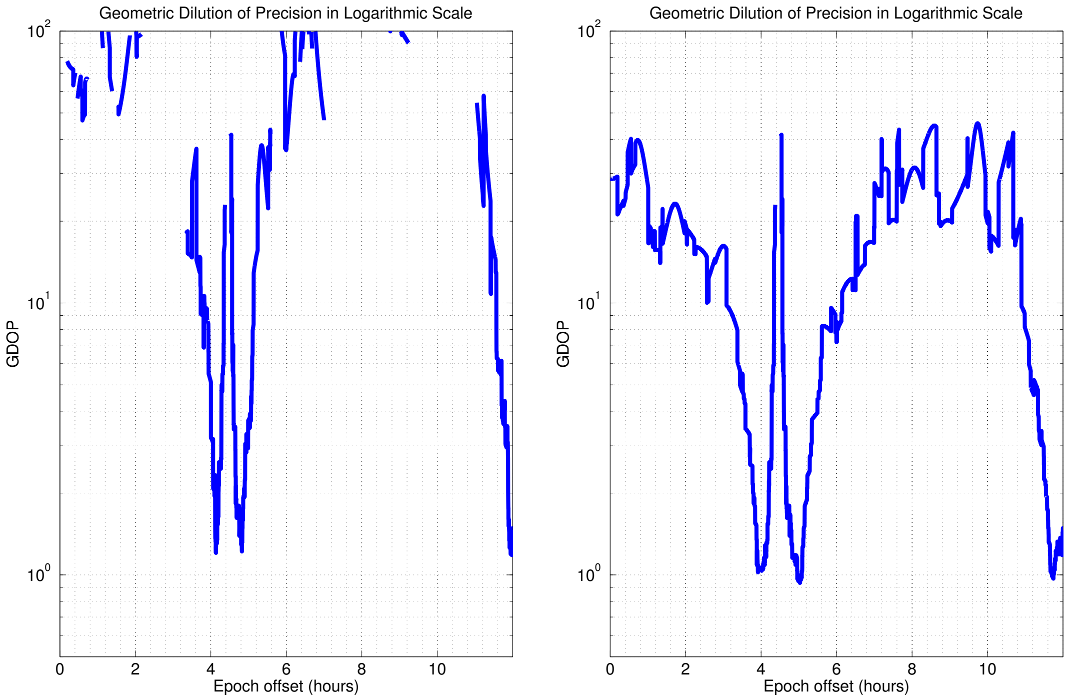


Figure 4.20: GDOP values based on GPS and GLONASS satellite observations for HEO case without sidelobe tracking (left) and with sidelobe tracking (right)

Table 4.6 (and all similar tables for upcoming simulations) shows the difference, in metres as $\Delta(m)$ and percentage as $\Delta(\%)$, between the current and all previous simulation results. Negative numbers for $\Delta(m)$ and positive numbers for $\Delta(\%)$ indicate improvement over previous results, and vice versa. As such, Table 4.6 shows that the LEO cross-track, along-track and radial errors were reduced compared to GPS-only errors. The along-track and cross-track errors improve by 3 dm while radial error improves by 1 dm - improvements of 27, 21 and 10%, respectively. In HEO simulations, mixed results were obtained depending on whether or not sidelobe tracking is enabled. Where the sidelobe tracking is deactivated, significant improvement (more than 52%) in along-track and cross-track RMSEs are obtained for period of 5.0 - 5.5 hours, and only 4% improvement for radial RMSE. For period of 11.6 - 12.0 hours, along-track, cross-track and radial RMSEs improved by approximately 23%, respectively.

Where the sidelobe tracking is activated, most of the improvement (over 31%) is made during the period of 5.0 - 5.5 hours. During the period of 11.6 - 12.0 hours, the along-track and radial RMSEs have improved significantly (over 39%). Surprisingly, the cross-track RMSE has apparently degraded by an incredible 953%. However, this apparent degradation is the result of an anomalously low corresponding GPS-only cross-track RMSE. So the apparent degradation will be disregarded. With more data available, the RMSE values obtained for GPS + GLONASS case shows a much more realistic overview of the positioning performance than in the GPS-only case, but with the penalty of more noisy and more biased data.

Case	LEO	HEO (Without Sidelobes)		HEO (With Sidelobes)	
Time Period	0.7-2.5 hrs	5.0-5.5 hrs	11.6-12.0 hrs	5.0-5.5 hrs	11.6-12.0 hrs
Root Mean Squared Errors (m)					
Along-track	0.87	4.01	2.74	2.26	0.82
Cross-track	0.93	1.22	1.20	1.25	1.31
Radial	0.98	17.00	5.00	2.16	1.87
Comparison with GPS-only case $[\Delta(m), \Delta(\%)]$					
Along-track	-0.31, +27	-4.42, +52	-0.82, +23	-3.41, +60	-0.52, +39
Cross-track	-0.25, +21	-3.29, +73	-0.37, +23	-1.17, +48	+1.18, -953
Radial	-0.11, +10	-0.73, +4	-1.38, +22	-0.98, +31	-4.29, +70

Table 4.6: Position RMSEs for the LEO and HEO cases involving GPS and GLONASS with comparison to the previous results

The final result to be presented (shown in Figure 4.21) is the overall filter performance (as defined by Equation 4.4). GLONASS has a very limited positive effect on LEO positioning and HEO position uncertainty estimation. It can be seen that the radial performance has in fact degraded for both situations when compared with GPS alone. The difference between estimated and true radial errors also appears noisier. Overall GLONASS does enhances satellite visibility for the HEO case during the time span over which position estimates can be made. GLONASS also improves overall geometric strength for both LEO and HEO cases; however, introduction of GLONASS entails a noisier environment, leading to a degraded filter performance.

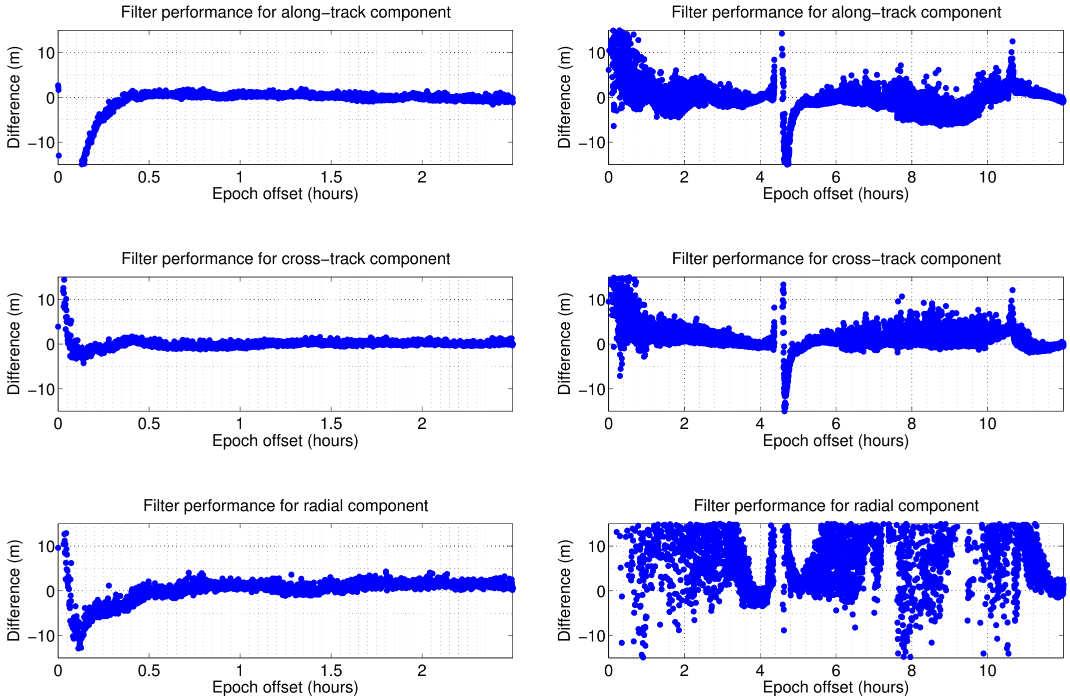


Figure 4.21: Filter performance based on GPS and GLONASS satellite observations for LEO (left) and HEO with sidelobe tracking (right)

4.2.5 GPS + GLONASS + Galileo Results

The previous simulation scenario is now expanded to include the Galileo constellation. A built-in Galileo ephemeris definition is used as no Galileo ephemeris files exist. By comparing results shown in Figure 4.22 to those in Figure 4.17, it is clear that for the LEO, convergence periods for cross-track and radial components have been greatly improved, and more or less unchanged for the along-track component.

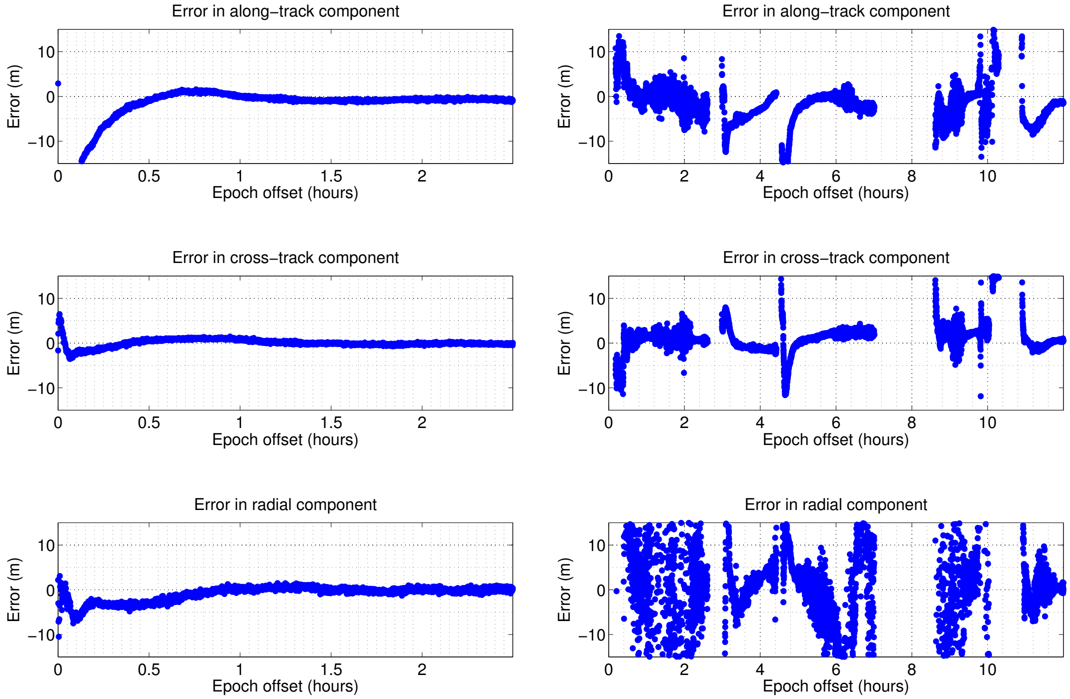


Figure 4.22: Position errors based on GPS, GLONASS and Galileo satellites for LEO (left) and HEO without sidelobe tracking (right)

For the HEO, a large portion of the orbital path is recovered without the need for sidelobe tracking. Moreover, along-track and cross-track errors have been noticeably reduced, and there is a modest improvement in the radial error. Figure 4.23 shows that GDOP values for the LEO case are now around unity, while HEO GDOP values are slightly better than those for GPS + GLONASS (improvement of ~10 units). This suggests better geometry results, with preference given to the HEO case. Improvements in geometry should lead to better filter performance.

From Figure 4.24, it is clear that sidelobe tracking dramatically improves positioning results for GPS + GLONASS + Galileo over GPS + GLONASS. Position estimates are available for nearly the entire simulation run, only disappearing shortly before and after the “blind-zone” described in section 4.2.2. Moreover, position errors have been re-

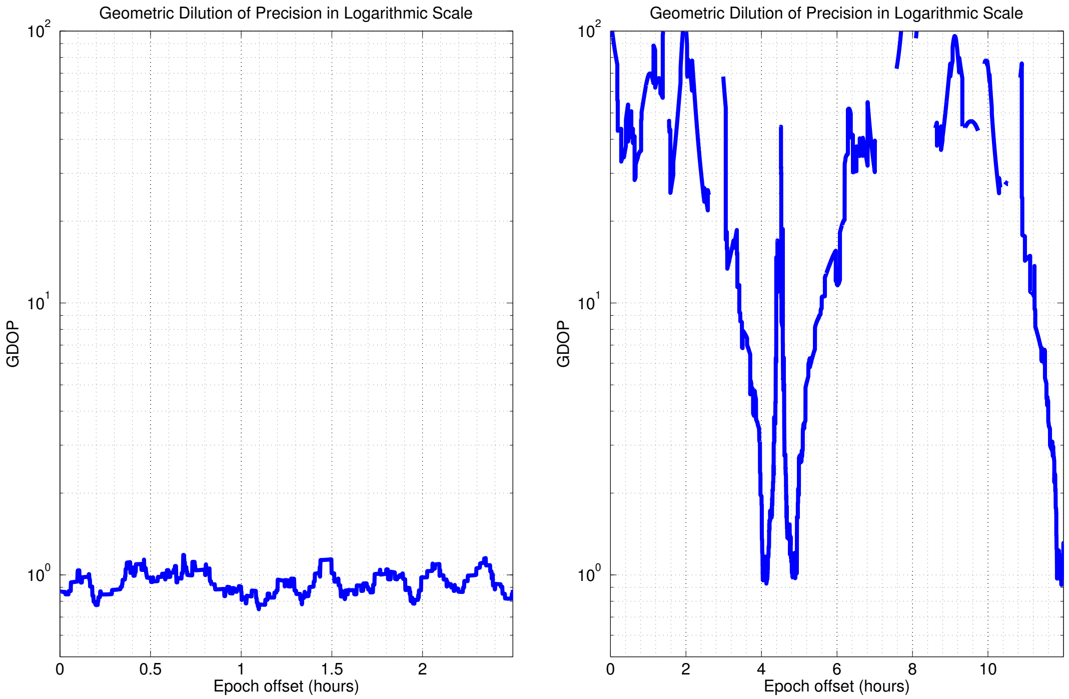


Figure 4.23: GDOP values based on GPS, GLONASS and Galileo satellites for LEO (left) and HEO without sidelobe tracking (right)

duced significantly (by several metres to 5 metres or more) when compared to the sidelobe-deactivated case, indicating superior filter performance. Even the radial component error has been reduced, which is not seen in the previous two scenarios. Due to relatively poor geometrical strength in the HEO, any improvement in satellite visibility can have an impact on dilution of precision. As seen in previous cases, the sidelobe tracking tends to improve GDOP values. A substantial improvement in GDOP values can be observed from Figure 4.25, with most values below 20. Without sidelobe tracking GDOP values can reach 100 or more.

Table 4.7 shows that for the LEO case, the improvements made between the GPS-only case and this case are approximately 14% less than improvements made between the GPS-only and the GPS + GLONASS case. The HEO results are mixed. Compared to the GPS-

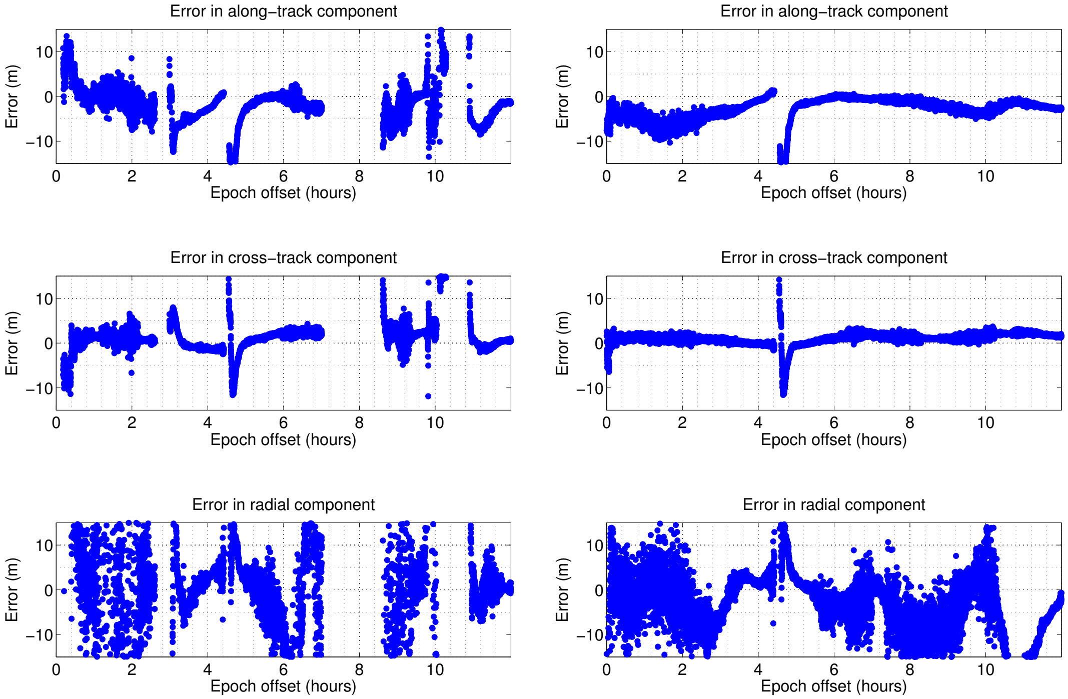


Figure 4.24: Position errors based on GPS, GLONASS and Galileo satellites for HEO without sidelobe tracking (left) and with sidelobe tracking (right)

only results, the HEO results for the 5.0 to 5.5 hours period show significant improvement in all components (ranging from 61% to 86%). Similarly, compared to GPS + GLONASS results for the 5.0 to 5.5 hours period, improvements can be noted for all components. Overall, the improvements made over the GPS + GLONASS case for 5.0 to 5.5 hours period is smaller than those made over the equivalent GPS-only case. This suggests that tracking a third constellation (Galileo) results in a diminishing return.

For the 11.6 to 12.0 hours period, sidelobe activated HEO simulations show significant degradation (93%) in RMSEs compared to previous simulation results. In fact, for sidelobe activated HEO simulation, performance during the 11.6 to 12.0 hours period is significantly (several metres) worse than the equivalent performance observed for the GPS-only and GPS + GLONASS cases. This result can be explained by noting that during this

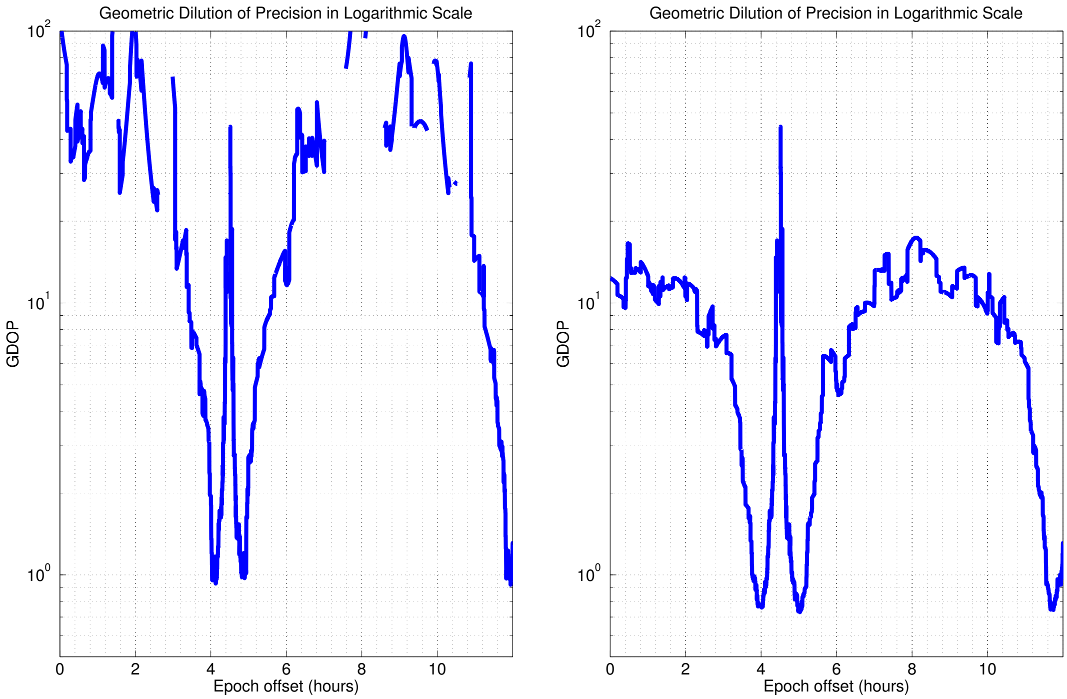


Figure 4.25: GDOP values based on GPS, GLONASS and Galileo satellites for HEO without sidelobe tracking (left) and with sidelobe tracking (right)

period, the total number of visible satellites increases dramatically. As each satellite adds more measurement errors (such as hardware biases, multipath, and orbital and clock errors) position errors increase, causing a jump in position RMSEs. A similar effect is noted in the GPS + GLONASS + Galileo LEO simulation; however, due to better geometry, the filter in the LEO case is able to average out additional measurement biases better than in the HEO cases. In turn, this leads to drastic differences between filter performance for these two cases.

It is interesting to note from Figure 4.26 that the LEO data processor still requires approximately 30 minutes for the along-track error to converge, while other components convergence period has been reduced by at least a factor of 2 or more. Although convergence periods have decreased for the cross-track and radial components, the filter (as was shown

Case	LEO	HEO (Without Sidelobes)		HEO (With Sidelobes)	
Time Period	0.7-2.5 hrs	5.0-5.5 hrs	11.6-12.0 hrs	5.0-5.5 hrs	11.6-12.0 hrs
Root Mean Squared Errors (m)					
Along-track	0.77	1.60	2.06	1.63	2.60
Cross-track	0.48	0.83	0.47	0.34	1.69
Radial	0.52	2.79	1.14	1.21	5.05
Comparison with GPS-only case $[\Delta(m), \Delta(\%)]$					
Along-track	-0.41, +35	-6.82, +81	-1.51, +42	-4.05, +71	+1.26, -93
Cross-track	-0.69, +59	-3.68, +82	-1.11, +70	-2.08, +86	+1.56, -1259
Radial	-0.58, +53	-14.95, +84	-5.23, +82	-1.93, +61	-1.11, +18
Comparison with GPS + GLONASS case $[\Delta(m), \Delta(\%)]$					
Along-track	-0.10, +11	-2.40, +60	-0.69, +25	-0.63, +28	+1.78, -217
Cross-track	-0.44, +48	-0.39, +32	-0.74, +61	-0.91, +73	+0.38, -29
Radial	-0.47, +47	-14.21, +84	-3.86, +77	-0.95, +44	+3.18, -170

Table 4.7: Position RMSEs for the LEO and HEO cases involving GPS, GLONASS and Galileo with comparison to the previous results

previously) had to average down additional measurement noise, which caused it to produce a solution that is slightly worse (by a few centimetres) than the one produced for the GPS + GLONASS case. The same observation can be made about the filter performance of the HEO simulations.

The addition of Galileo satellites has improved the geometry for both the LEO and HEO simulations with the additional benefit of increased satellite availability for the HEO simulations at the expense of filter performance in some situations. The situations have been identified as times when satellite visibility is relatively high (over 20 satellites visible), with new satellites being introduced. The results for the LEO case also show a less significant improvement in positioning is made by tracking GPS, GLONASS and Galileo than tracking GPS and GLONASS alone. This suggests that for the LEO case, performance is subjected to the law of diminishing returns as the number of tracked satellites increases.

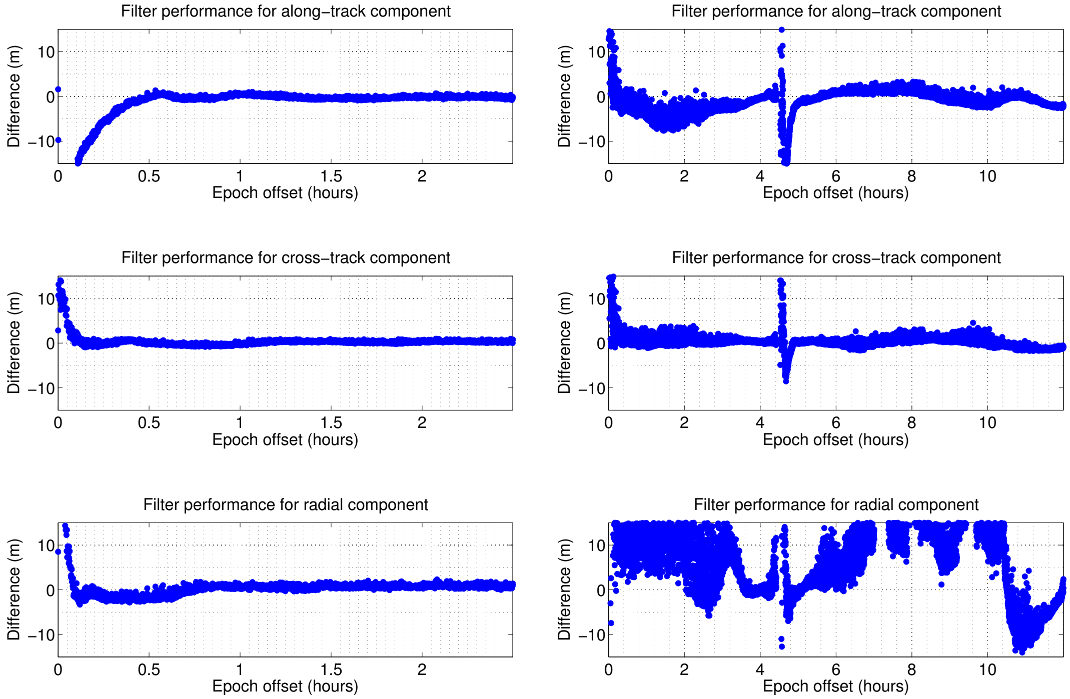


Figure 4.26: Filter performance for LEO (left) and HEO with sidelobe tracking (right) based on observations of GPS, GLONASS and Galileo satellites

4.2.6 GPS + GLONASS + Galileo + Compass Results

The final GNSS to be incorporated is Compass. Since the specifications of the Compass constellation are not definitively known at this time, parameters similar to Galileo's are adopted. As noted in section 3.4.3, Compass' geostationary satellites are not modelled by the Compass module, so only MEO satellites are modelled. It is not immediately evident from Figure 4.27 if significant improvement is made over previous scenarios. For LEO, position errors appear to be similar to those in GPS + GLONASS + Galileo scenario. Position errors during the convergence period are slightly larger (approximately a metre at most) than those in previous scenarios, likely due to a slightly less accurate initial covariance and position state estimation owing to an increased number of state parameters In the

HEO case, a similar conclusion can be made with the exception that position estimates are available for a longer period (nearly the entire simulation run). In all previous scenarios, the filter never reaches a perfectly steady state as position errors deviate from zero well after the initial convergence period. This phenomenon is most readily detected in the LEO case. The HEO case also experiences this phenomenon. However, due to a longer simulation span and a more erratic behaviour in position errors, departures from a steady state is not easily distinguished from a noisy behaviour.

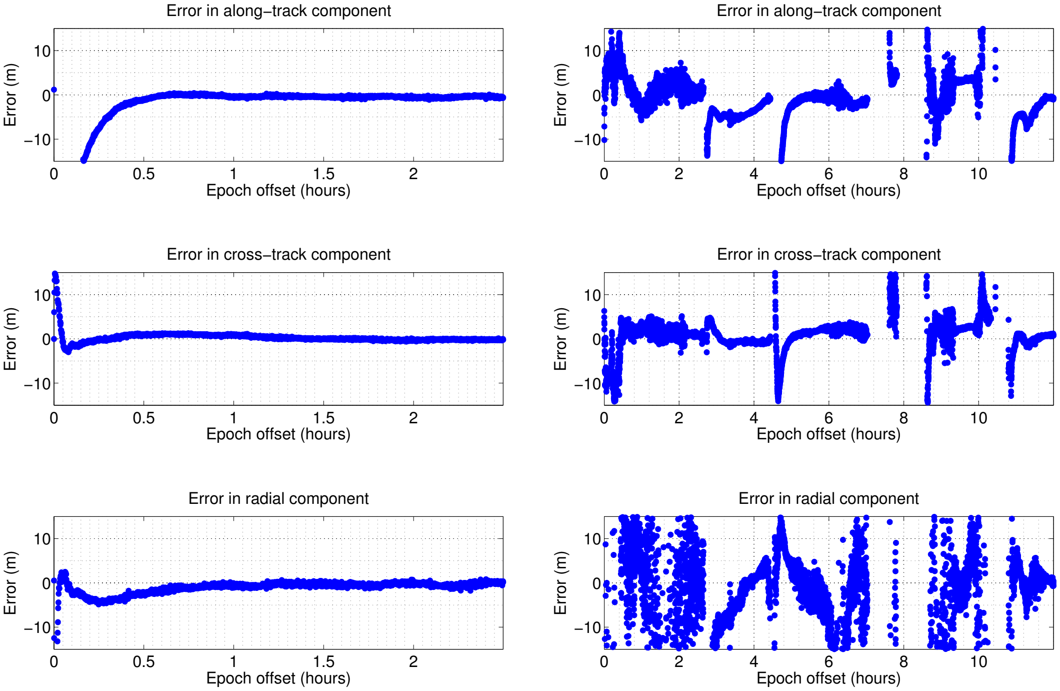


Figure 4.27: Position errors based on GPS, GLONASS, Galileo and Compass satellites for LEO (left) and HEO without sidelobe tracking (right)

The previous results show that GDOP values improve with an increased number of satellites. However GDOP values alone may not be sufficient in predicting or explaining position errors. Most likely position errors are caused by stochastic mismodelling in the data filter, which causes computed positions to deviate from the true ones. This proposition will be examined later. For the moment, it is important to confirm that geometry does not play a significant role in determining position error. As shown in Figure 4.28, GDOP values for the LEO have been improved to below 1. In the HEO case, a very small improvement is made (~10-20 units), suggesting that geometry for this scenario is still extremely poor.

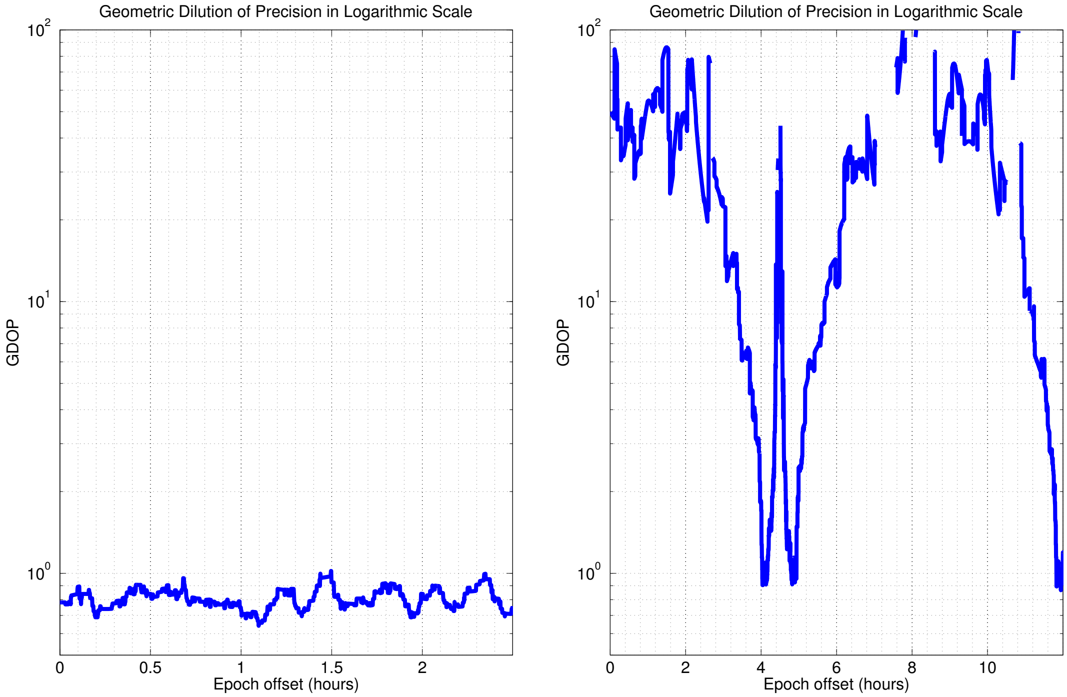


Figure 4.28: GDOP values based on GPS, GLONASS, Galileo and Compass satellites for LEO (left) and HEO without sidelobe tracking (right)

HEO position error with and without sidelobe tracking is presented in Figure 4.29. For the HEO with sidelobe tracking, an almost complete satellite orbit can be recovered with a "blind-spot" lasting only about 20 minutes. Despite this solution availability, very little improvement in position error is gained, suggesting that improvements made by adding Compass tracking are significantly lower than those made by adding Galileo. That is, the addition of a fourth constellation saturates the solution and invokes the law of diminishing returns. From Figure 4.28, it is argued that little improvement in GDOP values results from the introduction of Compass GNSS. It is interesting to see if this conclusion persists when sidelobe tracking is activated.

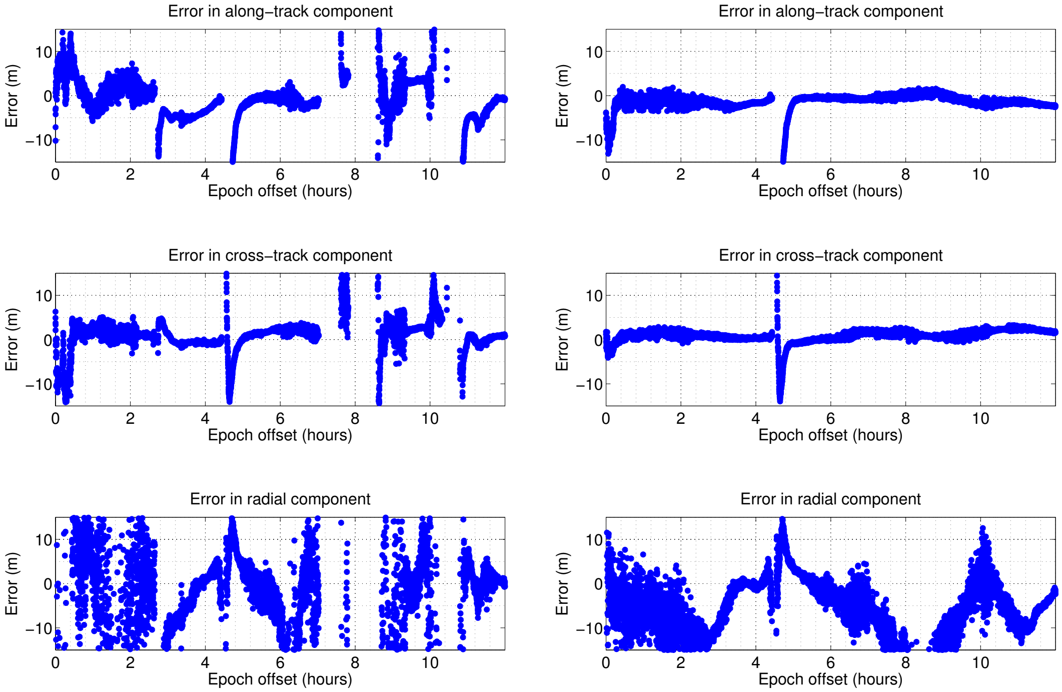


Figure 4.29: Position errors based on GPS, GLONASS, Galileo and Compass satellites for HEO without sidelobe tracking (left) and with sidelobe tracking (right)

Figure 4.30 shows that GDOP values improve dramatically when the sidelobe tracking feature is activated. When compared to GDOP values in (Figure 4.25), little improvement is seen. GDOP values are reduced very little, and the overall structure of the plot remains similar. This suggests that the addition of Compass leads to little overall improvement.

From Table 4.8, little improvements in the along-track and cross-track components (~4-43%) and some degradation in radial component (~15%) are found in RMSEs for the LEO simulations compared to GPS + GLONASS + Galileo RMSEs. Although more improvement was made over the GPS-only case than the GPS + GLONASS case, the magnitude of improvement appears to diminish. Similarly, the HEO results for the simulations in-

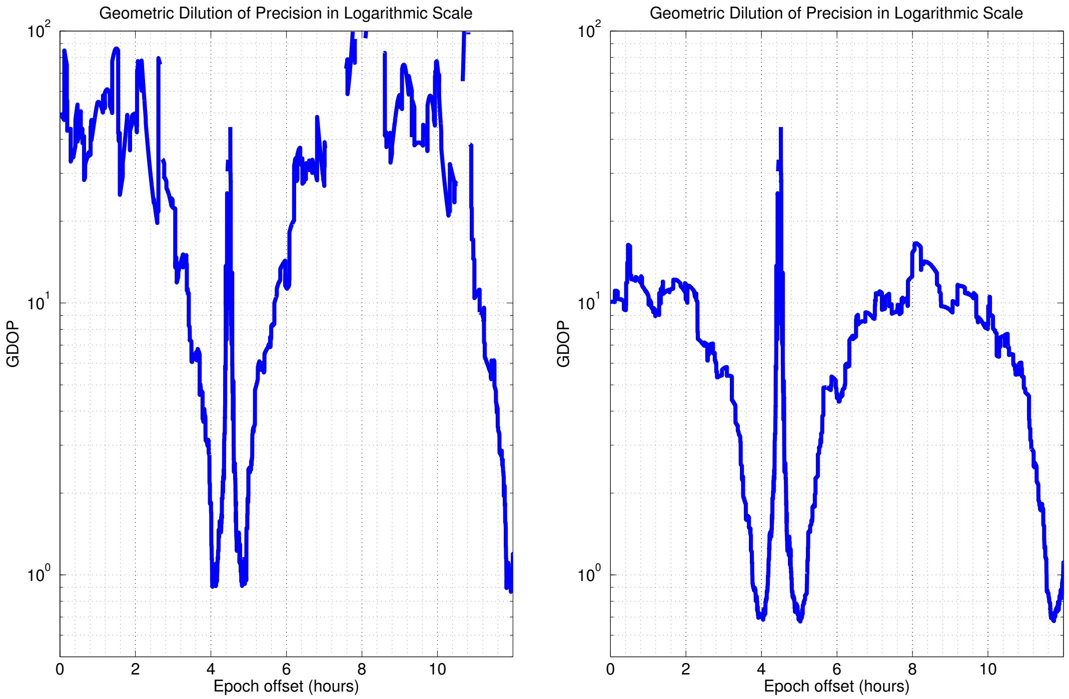


Figure 4.30: GDOP values based on GPS, GLONASS, Galileo and Compass satellites for HEO without sidelobe tracking (left) and with sidelobe tracking (right)

volving no sidelobe tracking for this case show diminishing improvements over equivalent results for the other cases. For the HEO simulations with sidelobe tracking, a different outcome can be observed. Almost all components saw either large (126-166%) or moderate (~0-62%) degradation.

As was mention for the GPS + GLONASS and GPS + GLONASS + Galileo cases, this result is likely due to increase in measurement noise (brought on by multipath and orbital and clock errors) and measurement biases (brought on by residual hardware biases) coupled with poor geometry. Although degradation are significant, they are not larger than a few metres. As a result, it is possible to position in HEO to within 5 metres by tracking all four GNSSs. Unfortunately, positioning performance is in fact subject to diminishing returns, as most improvements over the GPS + GLONASS + Galileo case are not significantly better

Case	LEO	HEO (Without Sidelobes)		HEO (With Sidelobes)	
Time Period	0.7-2.5 hrs	5.0-5.5 hrs	11.6-12.0 hrs	5.0-5.5 hrs	11.6-12.0 hrs
Root Mean Squared Errors (m)					
Along-track	0.44	1.36	1.51	0.81	2.19
Cross-track	0.46	1.08	0.69	0.64	1.88
Radial	0.60	2.53	1.15	2.13	4.23
Comparison with GPS-only case $[\Delta(m),\Delta(\%)]$					
Along-track	-0.74, +63	-7.06, +84	-2.06, +58	-4.86, +86	+0.84, -62
Cross-track	-0.71, +61	-3.43, +76	-0.88, +56	-1.78, +74	+1.75, -1413
Radial	-0.50, +46	-15.20, +86	-5.23, +82	-1.01, +32	-1.93, +31
Comparison with GPS + GLONASS case $[\Delta(m),\Delta(\%)]$					
Along-track	-0.43, +49	-2.65, +66	-1.23, +45	-1.45, +64	+1.36, -166
Cross-track	-0.46, +50	-0.15, +12	-0.52, +43	-0.61, +49	+0.57, -44
Radial	-0.39, +40	-14.47, +85	-3.85, +77	-0.03, +1	+2.36, -126
Comparison with GPS + GLONASS + Galileo case $[\Delta(m),\Delta(\%)]$					
Along-track	-0.33, +43	-0.24, +15	-0.55, +27	-0.81, +50	-0.42, +16
Cross-track	-0.02, +4	+0.24, -29	+0.22, -48	+0.30, -90	+0.19, -11
Radial	+0.08, -15	-0.26, +9	+0.01, -0	+0.91, -75	-0.83, +16

Table 4.8: Position RMSEs for the LEO and HEO cases involving GPS and GLONASS, Galileo and Compass with comparison to the previous results

than for the other cases. Therefore one can suggest that further expansion of GNSS satellite tracking will impact results even less.

Figure 4.31 shows that filter performance has not improved much over previous scenarios as difference curves are very similar to those shown in Figure 4.26. This suggests that no further significant positioning improvements can be expected from introducing additional satellites beyond a certain number. The best positioning that can be made in the LEO case, given the way the LEO simulation data were created and processed is approximately 0.6 metres in the along- and cross-track directions and 0.6 metres in the radial direction. For the HEO case without sidelobe tracking, position accuracy of approximately 1.7 metres in along- and cross-track directions and 1.2 metres in radial direction were obtained.

For the HEO case with sidelobe tracking, position accuracy can be on par with the sidelobe tracking deactivated case; however, satellite visibility is significantly better. Hence position can be estimated for a much longer duration (sometimes by a whole order of magnitude). These results suggest that with a multi-GNSS receiver, it may be possible to position a LEO to better than one metre and a HEO to better than 5 metres (3D).

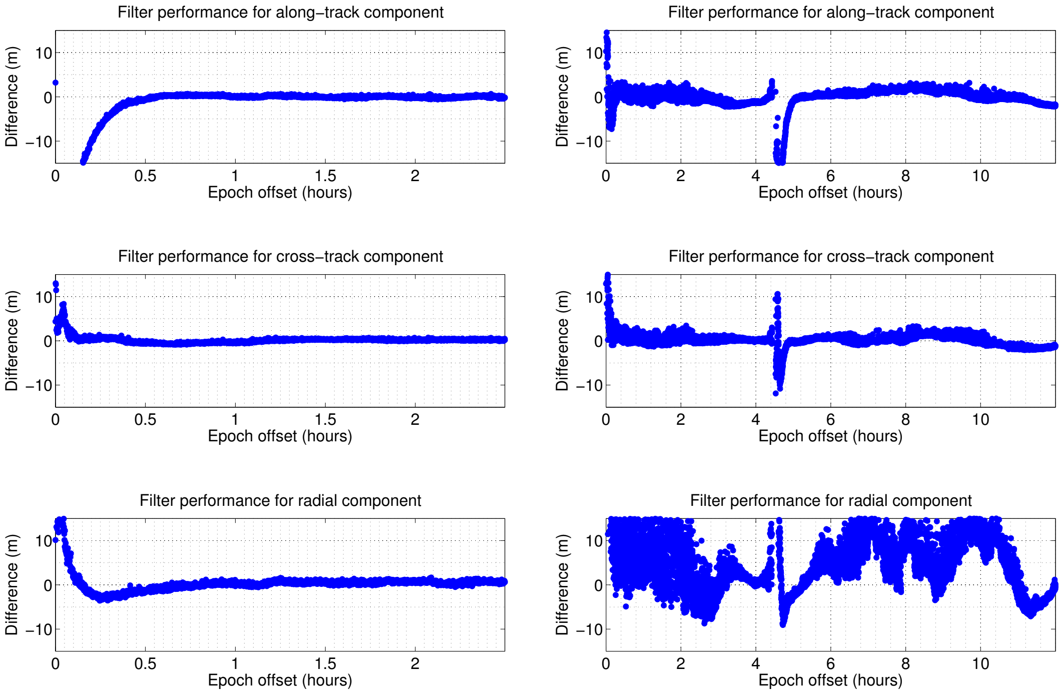


Figure 4.31: Filter performance for LEO (left) and HEO with sidelobe tracking (right) based on observations of GPS, GLONASS, Galileo and Compass satellites

4.2.7 Summary of Results

For the LEO simulation case, it is shown that GPS is adequate to provide metre-level positioning. It is also shown that it takes approximately 30 to 40 minutes for the LEO receiver's data filter to converge to a solution. Two HEO cases are presented, one with direct lobe tracking, and the other with additional sidelobe tracking. It is shown that the sidelobe tracking feature drastically enhanced satellite visibility and positioning performance. Results for HEO cases also show the need for multi-GNSS positioning as it is not possible to adequately recover the spacecraft's trajectory without tracking of three GNSS constellations. LEO satellite positioning performance improves steadily with every additional GNSS. With the HEO cases, improvements are made when additional satellites become visible; however, positioning performance is still very poor compared to the LEO case. The best positioning performance for the HEO case is 5 metres (3D RMSE); however, this is not sustained during the entire simulation run. HEO cases also showed that it is not possible to maintain continuous positioning, since there is a period of time (dubbed as a "blind-spot") where practical no satellites are visible. This "blind-spot" is caused by the restrictive antenna pointing requirements of the HEO satellite. GDOP values for HEO cases are highly unstable, changing from values below 1 (for portions of GPS + GLONASS + Galileo + Compass scenario with sidelobe tracking turned on) to well over 100 (with a single constellation). For the LEO case, GDOP values steadily improved from one scenario to the next, dropping below a value of 1 for the last scenario.

Notably, the tracking of more GNSSs does not necessarily lead to significant improvement in results. In some cases tracking more satellites can lead to degradation in performance. To examine this idea, RMSE and satellite count plots for LEO and HEO cases are incorporated into Figure 4.32. In the LEO case, it appears that positioning performance generally improves with an increasing number of satellites until Compass is added, after which slight degradations in the radial component can be observed. The degradation is possibly due to increased measurement noise (such as receiver hardware biases) associated with an overwhelming number of satellites being observed, preventing the data filter from effectively averaging this noise out. In the HEO simulations another intriguing observation can be made: positioning performance can either improve or degrade depending on the GNSS satellites being tracked. For example, tracking of the GPS, GLONASS and Galileo satellites appears to lead to a better positioning performance than tracking of the GPS, GLONASS, Galileo and Compass satellites for both HEO cases.

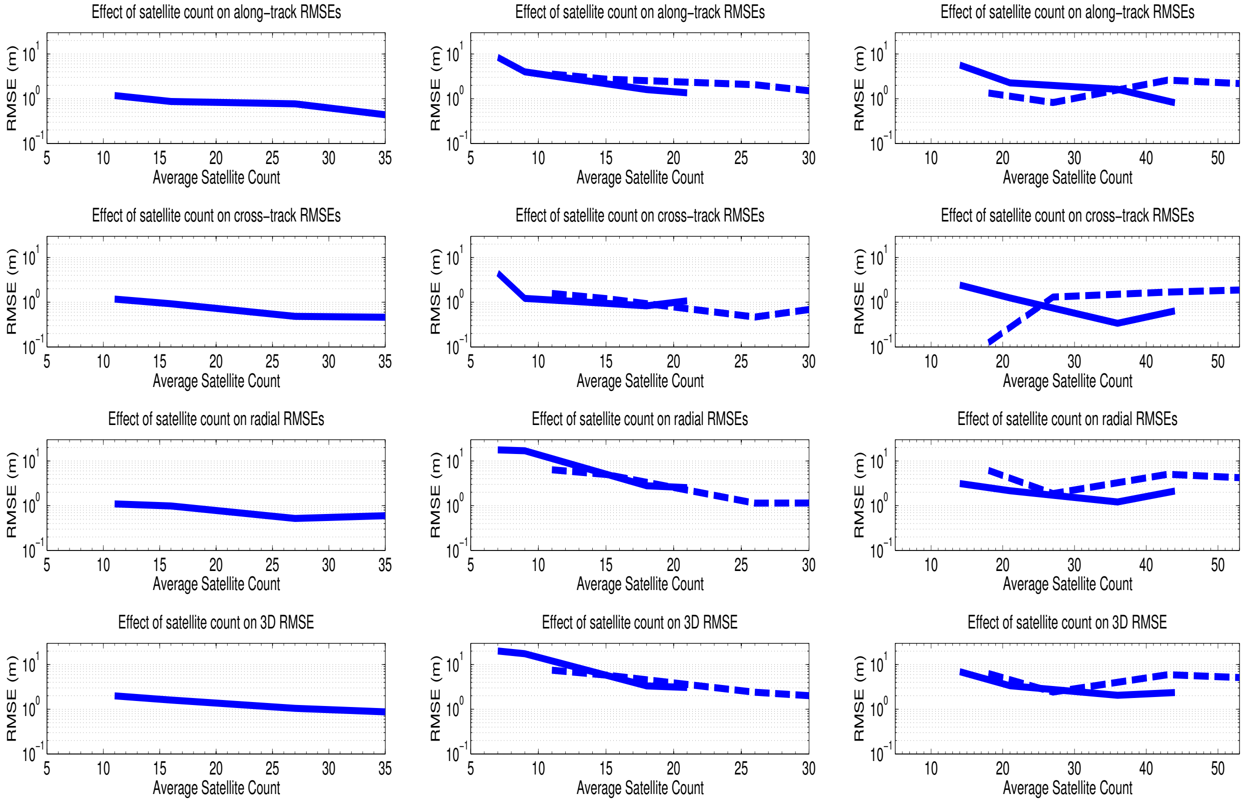


Figure 4.32: Variation of RMSEs with the average number of satellites for the LEO case (left), and the HEO cases with (middle) and without (right) sidelobe tracking for the 5.0 to 5.5 hours period (solid line) and the 11.6 to 12.0 hours period (dashed line)

Despite this HEO with sidelobe tracking tracking of GPS and GLONASS appears to lead to an overall better positioning performance than tracking of GPS, GLONASS and Galileo satellites. Overall, it appears that the benefit of tracking additional GNSS satellites does in fact diminish in both the LEO and HEO simulations (as 3D RMSEs appear to diminish with the number of satellites). Of course, the resolution of the plots in Figure 4.32 is not very high, since only four simulation scenarios are involved. In all simulation cases, multi-GNSS data processing brings improved position accuracy and satellite availability. Despite this, the improvements made will diminish with every new GNSS. Moreover, it is crucial to take into account the data quality derived from each GNSS, as it may not be favourable to process data from all GNSS at once, since that may lead to a degradation. Thus multi-GNSS data processing is not necessarily a beneficial option in LEO applications, since GPS alone is adequate for good positioning. In the HEO case, multi-GNSS data processing is necessary, since satellite visibility is typically poor, and spacecraft dynamics are high. If possible, sidelobe tracking should be employed to enhance satellite visibility and thus positioning accuracy.

Chapter 5

Conclusions and Future Work

This chapter will begin with summary of the salient features and capabilities of MGOS. This will be followed by a review of some of the more pertinent results from EOS studies (Chapter 4), and important conclusions that may be drawn from them. A discussion of how one may enhance MGOS, and in what other applications MGOS may prove useful will conclude this thesis.

5.1 Conclusions

The principal goal of this research is to study how multi-GNSS data processing can be used to enhance single-GNSS processing, particularly in satellite positioning. This research consists of two major components. The first is the development and evaluation of Multi-GNSS Observables Simulator (MGOS). MGOS is a GUI-based, modular program written in C++ with approximately 56,000 lines of code. It is designed to simulate sophisticated multi-GNSS scenarios in a realistic manner, is easy to use, and is flexible in providing control for users and developers alike. The second component is a multi-GNSS study involving GPS, GLONASS, Galileo and Compass GNSSs for LEO and HEO spacecraft equipped with multi-GNSS receivers. Evaluation of MGOS begins with a series of tests where GNSS observables are generated in RINEX 2.11 format, which are subsequently processed by various GPS processing software. Several software simulators considered prior to the development of MGOS (such as SATNAV toolbox by GPSoft (GPSoft LLC, 2006)) either offer limited RINEX output capabilities or none at all. MGOS' ability to generate RINEX files is a significant feature that separates it from other software GNSS simulators.

Three tests have been conducted: static PPP, static relative positioning in which simulated and real data are double-differenced, and kinematic PPP. The static PPP and relative positioning tests show that MGOS files can yield centimetre positioning accuracy, on par with what is expected from high-quality data sets regardless of the mode of processing, and demonstrated that MGOS is capable of simulating GPS measurements that are comparable in terms of realism with real data. These tests have validated the ability of MGOS in simulating realistic data. The kinematic test was conducted by first extracting the receiver path from an airplane RINEX observation file, and then using this trajectory to simulate a new RINEX observation file. The resultant RINEX file is processed by the NRCan PPP software as well as MGOS' own sequential least-squares data processor module, both producing state-of-the-art subdecimetre-level results. Thus MGOS users can use the available data processor with confidence rather than develop their own, which simplifies MGOS usage, and makes it an all-in-one package.

Having validated MGOS' simulation and data processing capabilities, one can confidently proceed with multi-GNSS studies for LEO and HEO applications. LEO and HEO orbital types are chosen in order to exploit the high contrast in orbital dynamics. LEO spacecraft are located relatively close to Earth's surface - their orbital velocities are practically constant from one orbit to the next. HEO spacecraft have dramatically varying orbital velocities - their distances from Earth change drastically. The goal of the LEO and HEO tests is to evaluate positioning performance of a multi-GNSS receiver with an increasing number of GNSS satellites being tracked. The test is divided into four cases: GPS-only, GPS + GLONASS, GPS + GLONASS + Galileo, and GPS + GLONASS + Galileo + Compass. HEO scenarios are further subdivided into two cases: with and without sidelobe tracking. Sidelobe tracking is a feature that allows a receiver to track more GNSS satellite signals than would normally be possible. In all cases, only broadcast ephemerides are used, so satellite orbital and clock errors are significantly larger than those expected of precise ephemerides. Several simplifying assumptions are made in the LEO and HEO tests.

Positioning in LEO is easier than in HEO due to several factors such as geometry and satellite visibility. However, in both cases, performance tends to stop improving beyond a certain number of satellites being tracked. It the LEO case, GPS alone is adequate for obtaining metre positioning accuracy relatively quickly (less than 30 minutes with a 5 second data sampling interval). Adding GLONASS tracking exhibits a slightly positive impact on positioning performance. However, due to lower quality ephemerides, position estimates are noisier than for the GPS-only case. The addition of Galileo and Compass shows diminishing improvement in positioning performance. In the HEO scenarios without sidelobe tracking, even combining data from GPS and GLONASS, it is not possible to continuously estimate receiver position. Part of the reason is the very poor geometry caused by the high altitude of the HEO (above the GNSS constellations). GDOP values of 100 or more are common for almost the entire duration of satellite's orbit. With sidelobe tracking, GPS and GLONASS, as well as GPS on its own are able to sustain position estimates for nearly the entire duration of the orbit with much improved position accuracy.

Unaccounted measurement residuals such as hardware biases and multipath tend to lead to degraded performance when additional satellites are being tracked. For example, in case of LEO, positioning performance for the radial component degrades by 15% when Compass satellites are tracked along with GPS, GLONASS and Galileo satellites. Similarly, in HEO simulations, degradation of as much as several metres is observed, perhaps due to an accumulation of unaccounted for errors. Overall, in all simulations, performance is affected by incorrect initial position and covariance values chosen by the data processor module. Due to consistent satellite tracking in the LEO case, accuracy of the estimated covariance values tends to improve over time. In the HEO cases, the covariance values must be re-estimated each time the number of tracked satellites drops below a certain value (which is a function of the total number of parameters being estimated). Finally, positioning performance for the LEO and HEO cases experiences diminishing returns as more and more satellites are tracked.

5.2 Proposed Future Research

This research has covered much ground with the successful development and evaluation of MGOS, as well as studies on LEO and HEO scenarios; however, there are improvements to be made to MGOS. A number of desirable features can be added to MGOS' GUI. For example, options to redo/undo certain operations, and to copy/paste modules from one simulation layout to another. (It is currently already possible to copy module parameters). Additional documentation would be beneficial. At the core of MGOS, several error source models have either been simplified or altogether ignored. Receiver antenna phase variations have not been taken into account due to fact that it would be a difficult task to treat these variations properly. The adopted troposphere error model is overly simplified in that it does not include azimuth dependence, nor does it incorporate dynamical models of temperature, pressure and humidity parameters. Improvements in the troposphere error can further improve the realism of simulation. Hardware biases need to be better estimated. Other features or improvements that can be made to MGOS' core include:

- Optimize the execution of simulations to yield faster MGOS performance,
- Allow auxiliary data (such as RINEX files, IONEX files, etc.) to be securely and compactly stored in a MGOS layout files so they can be shared more easily,
- Provide a means of recovery after either MGOS or one of its modules crashes,
- Allow a user to execute multiple simulation layouts at once (this feature was originally planned for MGOS, but due to time constraints has not been developed),
- Store simulation parameters on a per-simulation basis, so a user can quickly and easily recover simulation parameters for a specific simulation,
- Provide a simplified and more robust module programming interface with which to develop modules.

The modular and open source nature of MGOS should allow for improvements to be made with relative ease. MGOS was designed to be expandable and flexible from the start. Although this research project has been completed, it is hoped that MGOS will continue to evolve and expand beyond its current status. One worthy goal for MGOS would be to generate RINEX files that can be used in place of real data for RTK positioning. MGOS is not limited to applications presented in this thesis. Suitably enhanced, MGOS can be used to create a network of virtual reference stations that can be used in areas where real stations do not exist. MGOS can be used to test performance and accuracy of PPP or related software. MGOS can be used as a planning tool for any application dependent on GNSS services, as it can predict not only satellite visibility but also the type of performance that can be expected from a GNSS receiver on a particular day, in a particular environment and under certain atmospheric conditions.

References

Accord Software & Systems Inc. (2007). GPS Lab. URL http://www.gpslab.us/index1.htm. Accessed: Jan., 2010.

Aksnes, K., P. Andersen, and E. Haugen (1988). A Precise Multipass Method for Satellite Doppler Positioning. Celestial Mechanics and Dynamical Astronomy, 44(4): pp. 317–338.

Avila-Rodriguez, J., G. Hein, S. Wallner, J. Issler, L. Ries, L. Lestarquit, A. de Latour, J. Godet, F. Bastide, T. Pratt, and J. Owen (2007). The MBOC Modulation: The Final Touch to the Galileo Frequency and Signal Plan. Proceedings of the 20th International Technical Meeting of the Satellite Division. Fort Worth, Texas, U.S.A, pp. 1515–1529.

Balbach, O., B. Eissfeller, G. Hein, W. Enderle, M. Schmidhuber, and N. Lemke (1998). Tracking GPS Above GPS Satellite Altitude - First Results of the GPS Experiment on the HEO Mission Equator-S. IEEE 1998 Position Location and Navigation Symposium. Palm Springs, California, U.S.A, pp. 243–249.

Bandecchi, M. and W. Ockels (1998). The TEAMSAT Experience. ESA bulletin, 95.

Banville, S. (2008). The Precise Point Positioning Software Centre. URL http://gge.unb.ca/Resources/PPP/Purpose.html. Accessed: Oct., 2009.

Bauer, F., K. Hartman, and E. Lightsey (1998). Spaceborne GPS Current Status and Future Visions. Proceedings of IEEE 1998 Aerospace Conference, vol. 3. pp. 195–208.

Bisnath, S. and R. Langley (2001). Precise Orbit Determination of Low Earth Orbiters with GPS Point Positioning. Proceedings of the Institute of Navigation National Technical Meeting. Long Beach, California, U.S.A, pp. 22–24.

Black, H. (1990). Early Development of Transit, the Navy Navigation Satellite System. J. Guidance, Control, and Dynamics, 13(4): pp. 577–585.

Bock, H., A. Jäggi, R. Dach, S. Schaer, and G. Beutler (2009). GPS Single-Frequency Orbit Determination for Low Earth Orbiting Satellites. Advances in Space Research, 43(5): pp. 783–791.

Bos, M. and H.-G. Scherneck (2009a). A brief Description of the Available Ocean Tide Models. Onsala Space Observatory - Ocean tide loading provider, URL http://www.oso.chalmers.se/~loading/tidemodels.html. Accessed: Oct., 2009.

Bos, M. and H.-G. Scherneck (2009b). Welcome to the Free Ocean Tide Loading Provider. Onsala Space Observatory - Ocean tide loading provider, URL http://www.oso.chalmers.se/~loading/index.html. Accessed: Oct., 2009.

Braasch, M. and M. DiBenedetto (2001). Spread-spectrum Ranging Multipath Model Validation. IEEE Transactions on Aerospace and Electronic Systems, 37(1): pp. 298–304.

Brown, A., N. Gerein, and K. Taylor (2000). Modeling and Simulation of GPS Using Software Signal Generation and Digital Signal Reconstruction. Proceedings of the Institute of Navigation National Technical Meeting. Anaheim, California, U.S.A, pp. 646–652.

Byun, S., G. Hajj, and L. Young (2002). Development and Application of GPS Signal Multipath Simulator. Radio Science, 37(6): pp. 1–23.

Cai, C. and Y. Gao (2007). Precise Point Positioning Using Combined GPS and GLONASS Observations. Journal of Global Positioning Systems, 6(1): pp. 13–22.

CAST Navigation, LLC (2007). The CAST-1000 GPS Simulation System. URL http://www.castnav.com/products/cast_1000.html. Accessed: Jan., 2010.

CelesTrack (2004). NORAD Two-Line Element Set Format. URL http://celestrak.com/NORAD/documentation/tle-fmt.asp. Accessed: Sep., 2009.

CODE (2009). Klobuchar-Style Ionospheric Coefficients. Center for Orbit Determination in Europe, URL http://cmslive2.unibe.ch/unibe/philnat/aiub/content/e15/e59/e440/e447/e571/index_eng.html. Accessed: Oct., 2009.

Constantine, R. (2008). GPS & Galileo. Friendly Foes? Walker paper (Maxwell Air Force Base, Alaska), URL http://www.au.af.mil/au/aul/aupress/Walker_Papers/PDF_Bin/Constantine.pdf. Accessed: Dec., 2009.

Coordination Scientific Information Center (2002). GLONASS Interface Control Document. Tech. Rep. Version 5.0, Russian Space Agency.

Dow, J., R. Neilan, and G. Gendt (2005). The International GPS Service: Celebrating the 10th Anniversary and Looking to the Next Decade. Advances in Space Research, 36(3): pp. 320–326.

ESA (2007a). What Is Galileo? URL http://www.esa.int/esaNA/GGGMX650NDC_galileo_0.html. Accessed: Dec., 2009.

ESA (2007b). Why Europe Needs Galileo. URL http://www.esa.int/esaNA/GGG0H750NDC_galileo_0.html. Accessed: Dec., 2009.

ESA (2010). The Future - Galileo. URL http://www.esa.int/esaNA/galileo.html. Accessed: Jan., 2010.

Fortescue, P., J. Stark, and G. Swinerd (2003). Spacecraft Systems Engineering. Wiley, New Jersey, third edn., 678 pp.

Gao, G., A. Chen, S. Lo, D. De Lorenzo, and P. Enge (2007). GNSS Over China - The Compass MEO Satellite Codes. Inside GNSS, 2(5): pp. 36–43.

Gao, Y., F. Lahaye, P. Héroux, X. Liao, N. Beck, and M. Olynik (2001). Modeling and Estimation of C1–P1 Bias in GPS Receivers. Journal of Geodesy, 74(9): pp. 621–626.

Gao, Y., Y. Zhang, and K. Chen (2006). Development of a Real-Time Single-Frequency Precise Point Positioning System and Road Test Results. Proceedings of ION GNSS. Fort Worth, Texas, U.S.A, pp. 26–29.

García-Fernández, M., M. Markgraf, and O. Montenbruck (2008). Spin Rate Estimation of Sounding Rockets Using GPS Wind-Up. GPS Solutions, 12(3): pp. 155–161.

Gatti, G., M. Falcone, V. Alpe, M. Malik, T. Burger, and M. Rapisarda (2008). GIOVE-B Chilbolton In-Orbit Test: Initial Results from the Second Galileo Satellite. Inside GNSS: pp. 30–35.

Gibbons, G. (2008a). Russia Approves CDMA Signals for GLONASS, Discussing Common Signal Design. Inside GNSS.[Online] April, URL http://www.insidegnss.com/node/648. Accessed: Oct., 2009.

Gibbons, G. (2008b). Russia Dwells on GLONASS Future. Inside GNSS.[Online] March/April, URL http://www.insidegnss.com/node/591. Accessed: Nov., 2009.

Gibbons, G. (2010). China's GNSS Program, Compass Beidou 2, Launches New GEO Satellite. Inside GNSS.[Online] April, URL http://www.insidegnss.com/node/1848. Accessed: Feb., 2010.

Goad, C. and L. Goodman (1974). A Modified Hopfield Tropospheric Refraction Correction Model. Proceedings of the Fall Annual Meeting of the American Geophysical Union. San Francisco, California, pp. 12–17.

GPS Joint Program Office (1993). Navstar GPS Space Segment/Navigation User Interfaces. Tech. Rep. ICD-GPS-200C, ARINC Research Corporation.

GPSoft LLC (2003). Satellite Navigation TOOLBOX 3.0 User's Guide. Athens, Ohio, U.S.A, 146 pp.

GPSoft LLC (2006). Satellite Navigation (SatNav) ToolBox 3.0. URL http://www.gpsoftnav.com/satnav.html. Accessed: Jan., 2010.

Grelier, T., A. Ghion, J. Dantepal, L. Ries, A. DeLatour, J. Issler, J. Avila-Rodriguez, S. Wallner, and G. Hein (2007). Compass Signal Structure and First Measurements. Proceedings of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation. Fort Worth, Texas, U.S.A, pp. 3015–3024.

Griffiths, J. and J. Ray (2009). On the Precision and Accuracy of IGS Orbits. Journal of Geodesy, 83(3): pp. 277–287.

Gurtner, W. and L. Estey (2006). RINEX: The Receiver Independent Exchange Format Version 3.00. Astronomical Institute, University of Berne.

Hein, G., J. Godet, J. Issler, J. Martin, P. Erhard, R. Lucas-Rodriguez, and T. Pratt (2002). Status of Galileo Frequency and Signal Design. 15th International Technical Meeting of the Satellite Division of the Institute of Navigation. Portland, Orlando, U.S.A.

Héroux, P. and J. Kouba (2001). GPS Precise Point Positioning Using IGS Orbit Products. Physics and Chemistry of the Earth, Part A, 26(6-8): pp. 573–578.

Hofmann-Wellenhof, B., H. Lichtenegger, and J. Collins (2001). GPS Theory and Practice. Springer Wein, New York, fifth edn., 370 pp.

Hollreiser, M., M. Crisci, J. Sleewaegen, J. Giraud, A. Simsky, D. Mertens, T. Burger, and M. Falcone (2007). Galileo Signal Experimentation. GPS WORLD, 18(5): p. 44.

Hoots, F. and R. Roehrich (1980). Spacetrack Report No. 3–Models for Propagation of NORAD Element Sets. Aerospace Defense Command, United States Air Force.

Hwang, Y. and G. Born (2005). Orbit Determination Strategy Using Single-Frequency Global-Positioning-System Data. Journal of Spacecraft and Rockets, 42(5): pp. 896–901.

Information-Analytical Centre (2006). GLONASS Constellation Status. Russian Space Agency, URL http://www.glonass-ianc.rsa.ru/pls/htmldb/f?p=202:20:18152560162244826079::NO. Accessed: Jan., 2010.

Institut Géographique National (2009). Welcome to the International Terrestrial Reference Frame web site. URL http://itrf.ign.fr/. Accessed: Jan., 2010.

Institute of Geodesy and Navigation (2008). GALILEO & GNSS–2 – GNSS Software Simulation. University FAF Munich, Germany, URL http://www.ifen.unibw-muenchen.de/research/gnss_simulator.htm. Accessed: Oct., 2009.

Kaplan, E. and C. Hegarty (2006). Understanding GPS: Principles and Applications. Artech House, Boston, second edn.

King, M. (2006). Kinematic and Static GPS Techniques for Estimating Tidal Displacements with Application to Antarctica. Journal of Geodynamics, 41(1-3): pp. 77–86.

Kondoh, Y., Y. Ishijima, I. Kawano, T. Iwata, H. Suzuki, S. Kumagai, M. Kakinuma, T. Eda, and M. Kasahara (2009). Development of a New Generation Spaceborne GPS Receiver. IEEE 2009 Aerospace conference. pp. 1–8.

Kouba, J. (2009). A Simplified Yaw-Attitude Model for Eclipsing GPS Satellites. GPS Solutions, 13(1): pp. 1–12.

Lan, S.-C., R. Sun, J.-X. Zhang, J.-H. Wang, and G.-D. Xu (2008). Satellite Position Calibration in Distributed Spacecraft System by Multi-Dimensional Scaling. Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium on. pp. 1–4.

Langley, R., H. Jannasch, B. Peeters, and S. Bisnath (2000). The GPS Broadcast Orbits: An Accuracy Analysis. Viewgraphs of paper presented in Session B, 2: pp. 16–23.

Larson, W. and J. Wertz, editors (1999). Space Mission Analysis and Design. Microcosm Press, California, third edn., 969 pp.

Le, A., C. Tiberius, H. van der Marel, and N. Jakowski (2008). Use of Global and Regional Ionosphere Maps for Single-Frequency Precise Point Positioning. International Association of Geodesy Symposia, 133: pp. 759–769.

Leandro, R., R. Langley, and M. Santos (2007a). Estimation of P2-C2 Biases by Means of Precise Point Positioning. Proceedings of the Institute of Navigation 63rd Annual Meeting. Cambridge, Massachusetts, U.S.A, pp. 23–25.

Leandro, R. F., M. C. Santos, and R. B. Langley (2007b). PPP-based Ionospheric Activity Monitoring. Proceedings of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation. Fort Worth, Texas, U.S.A, pp. 2849–2853.

Lee, I. and L. Ge (2006). The Performance of RTK-GPS for Surveying Under Challenging Environmental Conditions. Earth, Planets, and Space, 58(5): pp. 515–522.

Leick, A. (1990). GPS Satellite Survey. Wiley, New York, first edn., 352 pp.

Leick, A. (2004). GPS Satellite Surveying. John Wiley, New Jersey, third edn., 435 pp.

McCarthy, D. and G. Petit (2004). IERS Conventions (2003). IERS Technical Note, 32: pp. 3–128.

Montenbruck, O. and E. Gill (2000). Satellite Orbits: Models, Methods, and Applications. Springer, New York, first edn., 309 pp.

Montenbruck, O., M. Markgraf, M. Garcia-Fernandez, and A. Helm (2008). GPS for Microsatellites–Status and Perspectives. Small Satellites for Earth Observation: pp. 165–174.

NASA (2009). Picosat 9. URL http://nssdc.gsfc.nasa.gov/nmc/masterCatalog.do?sc=2001-043B. Accessed: Jun., 2009.

National Geodetic Survey (2009). Publishing Requirements. URL http://www.ngs.noaa.gov/OPUS/view.jsp. Accessed: Jan., 2010.

NGS (2009). OPUS: Online Positioning User Service. URL http://www.ngs.noaa.gov/OPUS/. Accessed: Jan., 2010.

Niell, A. (1996). Global Mapping Functions for the Atmosphere Delay at Radio Wavelengths. Journal of Geophysical Research, 101(B2): pp. 3227–3246.

NRCan (2004). On-line Precise Point Positioning ’How To Use’ Document. PDF Document, URL http://www.geod.nrcan.gc.ca/userguide/pdf/howtouse.pdf. Accessed: Oct., 2009.

Pardini, C. and L. Anselmo (2001). Influence of the Spacecraft End-of-Life Re-Orbiting Altitude on the Long-Term Collision Risk in the Geostationary Ring. Advances in Space Research, 28(9): pp. 1403–1408.

Qt Development Frameworks (2009). Products - Qt A cross-platform Application and UI Framework. Nokia Coorporation, URL http://qt.nokia.com/products. Accessed: Oct., 2009.

Revnivykh, S. (2005). GLONASS: Status and Perspectives. Munich Satellite Navigation Summit.

Roddy, D. (2001). Satellite Communications. McGraw-Hill, New York, third edn., 569 pp.

Schaer, S., W. Gurtner, and J. Feltens (1998). IONEX: The IONosphere map eXchange Format Version 1. Proceedings of the IGS AC Workshop, vol. 9. Darmstadt, Germany.

Spirent Communications (2010). GSS6700 - Multi-GNSS Constellation Simulator System. URL http://www.spirent.com/Solutions-Directory/GSS6700.aspx. Accessed: Jan., 2010.

Stewart, M. and M. Tsakiri (1998). GLONASS Broadcast Orbit Computation. GPS Solutions, 2(2): pp. 16–27.

Tossaint, M., S. Binda, J. Hahn, and M. Falcone (2008). SYSTEM DESIGN & TEST-Galileo Validation - The Navigation Message-The GIOVE - A satellite Broadcast the First-Ever Galileo Signals on January 12, 2006. GIOVE-B Is Scheduled for Launch by April 2008. GPS World, 19(2): pp. 30–37.

T.S. Kelso (2006). Frequently Asked Questions: Two-Line Element Set Format. URL http://celestrak.com/columns/v04n03/. Accessed: Feb., 2010.

U.S. Naval Observatory (2009). Current GPS constellation. URL http://tycho.usno.navy.mil/gpscurr.html. Accessed: Nov., 2009.

Švehla, D. and M. Rothacher (2003). Kinematic and Reduced - Dynamic Precise Orbit Determination of Low Earth Orbiters. Advances in Geosciences, 1(1): pp. 47–56.

Wanninger, L. and S. Wallstab-Freitag (2007). Combined Processing of GPS, GLONASS, and SBAS Code Phase and Carrier Phase Measurements. Proceedings of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation. Fort Worth, Texas, U.S.A, pp. 866–875.

Warren, D. and J. Raquet (2003). Broadcast Vs. Precise GPS Ephemerides: A Historical Perspective. GPS Solutions, 7(3): pp. 151–156.

Winternitz, L., W. Bamford, and G. Heckler (2009). A GPS Receiver for High-Altitude Satellite Navigation. Selected Topics in Signal Processing, IEEE Journal of, 3(4): pp. 541–556.

Wytrzyszczak, I., E. Wnuk, and J. Kaczmarek (2004). Reducing the Collision Risk for High Inclination Constellations in the Geosynchronous Region. Advances in Space Research, 34(5): pp. 1193–1197.

Xu, G. (2007). GPS Theory, Algorithms, and Applications. Springer, Berlin, second edn., 340 pp.

Ziff, R. (1998). Four-tap Shift-Register-Sequence Random-Number Generators. Computers in Physics, 12: p. 385.

Zinoviev, A. (2005). Using GLONASS in Combined GNSS Receivers: Current Status. Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation. Long Beach, California, U.S.A, pp. 1046–1057.

Appendix A: MGOS Algorithms

This appendix contains details about various MGOS algorithms described in the previous chapters.

Algorithm A.1 Computation of ocean loading error for a specific location using a provided HARPOS file

```
Given current epoch $t$ and receiver location vector $\overrightarrow{R}_t$
if (no HARPOS data available) then
    return No error (0 value)
end if
Compute distance between current receiver location and previously stored receiver location: $R = \left\| \overrightarrow{R}_t - \overrightarrow{R}_{t-1} \right\|$
if ($R > R_{MAX}$) then
    Search through read points and find a point within $R_{MAX}$ distance (by default, it is 100 kilometres) from the receiver location $\overrightarrow{R}_t$
end if
if (point found) then
    Compute ocean loading offset in North, East and Up frame
    Set $\overrightarrow{R}_{t-1} = \overrightarrow{R}_t$
    return Compute error at time $t$ using ocean loading coefficients for model defined in a read HARPOS file and amplitude coefficients for the located point
else
    return No error (0 value)
end if
```

Algorithm A.2 Sequential least squares filter model

```
Given satellite current epoch $t$ and satellite data matrix $S_t$
**if** (filter reinitialization is required) **and** (state vector fails to reset ) **then**
    **return** No solution
**else**
    Recover $t-1$ state vector $\overline{X}_{t-1}$ and co-factor matrix $C_{\overline{X}_{t-1}}$, and compute $C_{\overline{X}_t^0} = C_{\overline{X}_{t-1}} + CofactorUpdateMatrix$
**end if**
Construct matrices $A$ (design), $C_{\overline{X}_t}$ (co-factor) and $P$ (weight), and vectors $\overline{W}$ (residual), and $\overline{L}$ (measurement) and $\overline{V}$ (measurement adjustment)
Set measurement vector and weight matrix using data from $\overline{S}_t$ matrix
Set total iterations made, $t_{iter}$ to 1 and convergence value, $\delta\varepsilon$ to $\varepsilon$ {$\varepsilon$ is set to $10^{-5}$ }
**while** ($t_{iter} \leq t_{iter}^{Max}$ **and** $\delta\varepsilon \geq \varepsilon$) **do** {$t_{iter}^{Max}$ is set to 100}
    Update design matrix and residual vector to $A_{t_{iter}}$ and $\overline{W}_{t_{iter}}$.
    **if** (filter reinitialization is required) **then**
        Calculate new co-factor matrix: $C_{\overline{X}_t} = \left(A_{t_{iter}}^T \cdot P \cdot A_{t_{iter}}\right)^{-1}$
        **if** (failed to compute the new covariance) **then**
            **return** No solution
        **end if**
    **else**
        Calculate new co-factor matrix: $C_{\overline{X}_t} = \left[\left(C_{\overline{X}_t^0}\right)^{-1} + A_{t_{iter}}^T \cdot P \cdot A_{t_{iter}}\right]^{-1}$
        **if** (failed to compute the new covariance) **then**
            **return** No solution
        **end if**
    **end if**
    Compute adjustment to $\overline{X}$ and $\overline{L}$: $d\overline{X}_t = C_{\overline{X}_t} \cdot A_{t_{iter}}^T \cdot P \cdot \overline{W}_{t_{iter}}$ and $d\overline{V} = A_{t_{iter}} \cdot d\overline{X}_t - \overline{W}_{t_{iter}}$
    Update vectors, $\overline{X}, \overline{V}$ and $\overline{L}$ vectors: $\overline{X} = \overline{X} + d\overline{X}_t$, $\overline{V} = \overline{V} + d\overline{V}$ and $\overline{L} = \overline{L} + d\overline{V}$
    Update $\delta\varepsilon$ value: $\delta\varepsilon = \|\overline{X}\|$
**end while**
**if** ($\delta\varepsilon < \varepsilon$ **and** $t_{iter} \leq t_{iter}^{MAX}$) **then**
    Compute a posteriori variance: $\sigma_0^2 = \frac{\overline{V}^T \cdot P \cdot \overline{V}}{TotalMeasurements - TotalParameters}$
    Store current state vector and co-factor matrix in memory for the next epoch
    **return** $\overline{X}_t, C_t$ and $\sigma_0^2$
**else**
    **return** No results {Filter will be reinitialization on the next epoch}
**end if**
```

Algorithm A.3 Method used by EOS module to categorize blocked, occultated, and normal signals

```
Given a receiver vector $\overrightarrow{R} = (X_R, Y_R, Z_R)$ and a GNSS satellite vector $\overrightarrow{S} = (X_S, Y_S, Z_S)$
$\overrightarrow{RS} = \overrightarrow{S} - \overrightarrow{R}$ {Compute vector between satellite and receiver}
$R = \sqrt{X_R^2 + Y_R^2 + Z_R^2}$ {Receiver vector magnitude}
if $R \leq R_{OR}$ then {Determine the radius of a region, $R_O$, which will obstruct the signal}
    $R_O = R_E$
else
    $R_O = R_{OR}$
end if
$U = \frac{-\overrightarrow{R} \cdot \left(\overrightarrow{RS}\right)}{|\overrightarrow{RS}|^2}$ {This parameter is used to determine if signal is directed towards the obstruction region}
if $U < 0$ or $U > 1$ then {No intersection with $R_O$ sphere is possible if this is true}
    return Normal
end if
Determine whether signal line intersects with $R_O$ sphere using a simple line-sphere intersection algorithm:
$A = \overrightarrow{RS} \cdot \overrightarrow{RS}$
$B = 2\left(\overrightarrow{R} \cdot \overrightarrow{RS}\right)$
$C = \overrightarrow{S} \cdot \overrightarrow{R} - R_O^2$
$ARG = B^2 - 4AC$
if $ARG > 0$ and $R_O \leq R_{OR}$ then
    return Blocked {Since receiver is located within the occultation zone.}
else if $ARG > 0$ then
    return In Occultation {Since receiver is located outside of the occultation zone.}
end if
return Normal {Case $ARG \leq 0$ means that signal has touched by did not penetrate the $R_O$sphere which is still the normal case.}
```

Algorithm A.4 Rejection of satellite signals based on their signal power using a simplified signal propagation model

```
Given a receiver vector $\overrightarrow{R} = (X_R, Y_R, Z_R)$ and a GNSS satellite vector $\overrightarrow{S} = (X_S, Y_S, Z_S)$
$R = |\overrightarrow{S} - \overrightarrow{R}|$ {Compute distance between receiver and satellite locations}
if initializing then {This occurs at the start of every simulation run}
    $R_{MAX} = \frac{c}{4\pi f_{GPS}} 10^{\frac{1}{20}\left(P_T^{GPS} + G_T^{GPS} - S_P^{MIN}\right)}$ {Symbols: $P_T^{GPS}$- Chosen transmission power (GPS-like), $G_T^{GPS}$- Chosen transmission antenna gain (GPS-like), $S_P^{MIN}$- User determined minimum signal power at reception, $f_{GPS}$- Signal frequency (GPS-like)}
end if
if $R \leq R_{MAX}$ then
    return Signal can be processed
else
    return Signal cannot be processed and therefore should be rejected
end if
```