Improving the Position Accuracy with DGPS and EGNOS

- Group 1048 -
Dragos Catalin
Andrei Marinescu

April 22, 2010

UNIVERSITY OF AALBORG
GPS MASTER PROGRAMME

MASTER THESIS

Defended by
Dragos Catalin and Andrei Marinescu

Improving the Position Accuracy with DGPS and EGNOS

Supervisor: Kai Borre

Project Period: 1st September 2009 - 22 April 2010
Number of reports printed: 3
Number of pages in report: 58
Number of pages in appendix: 23

Acknowledgements

We would like to express our gratitude to our supervisor Prof. Dr. Kai Borre for the help and understanding provided during the months for developing this project, and also along the entire GPS Master Programme. Furthermore, we would like to thank Dr. Darius Plausinaitis for his assistance in several issues encountered in the project work.

Special thanks go to Kostas Dragunas for providing valuable help and guidance related to the EGNOS system and more.

Abstract

This thesis presents a study upon obtaining higher accuracy positions than those of average GPS. The research has been accomplished through means of EGNOS, DGPS and ionosphere models. A fusion of the available methods is employed towards obtaining best results. All data is processed in Matlab through real-time functions, and saved results over long time intervals of experiments are discussed.

Dragos Catalin ____________________

Andrei Marinescu ____________________

Contents

1 Introduction **1**

2 Background **3**

2.1 Error Sources In GPS 3

2.2 Ionosphere Influence on GPS Signals 4

2.3 The EGNOS System 5

2.3.1 The EGNOS Message 8

2.3.2 EGNOS Fast Corrections 9

2.3.3 EGNOS Slow Corrections 10

2.3.4 EGNOS Ionospheric Corrections 12

3 Problem Definition **17**

4 System Design **21**

4.1 EGNOS System Architecture 21

4.2 EGNOS Implementation 22

4.3 Signal in Space Through the Internet 25

4.4 Dual-frequency Data Processing 27

4.4.1 SW Approach 30

4.4.2 HW Approach 32

4.5 DGPS Implementation 36

5 Measurements and Results **39**

5.1 EGNOS Aided Measurements 39

5.1.1 Aspects that Influence the Results 47

5.2 Ionospheric Estimation Procedure 51

5.3 Results Of EGNOS Combined With Dual Frequency Data . . 54

6 Conclusions and Future Work **57**

Appendix A **59**

A.1 The Ionospheric Delay 59

A.1.1 Pierce Point Coordinates Determination 59

A.2 Interpreting Messages 18 and 26 62

A.2.1 Selecting the IGPs Involved in the Computation 66

A.2.2 Interpolating the GIVD Values 68

A.2.3 The Final Step 71

Appendix B **73**

Appendix C **75**

C.1 The Main Function 75

C.2 Ionospheric Correction Computation (EGNOS) 80

C.3 Real-time Data Acquisition 80

Bibliography **83**

List of Figures

2.1 Atmosphere layers affecting GPS signals 4
2.2 Augmentation Systems Worldwide 6
2.3 EGNOS Coverage Area (ECAC Area) 6
2.4 EGNOS Deployment Network 7
2.5 EGNOS Data Block Format 8
2.6 The IGP Grid 13
2.7 The Ionospheric Grid Mask 14

4.1 Ionospheric Pierce Point Locations 24
4.2 Ionospheric delay using pseudoranges 30
4.3 Double Differencing 31
4.4 Sky view reference satellite 33
4.5 Ambiguity on GPS signal 34
4.6 ZXtreme Pseudorange Ionosphere Estimation 35
4.7 Raw, Ionospheric, and DGPS corrected positions 38

5.1 Position Accuracy for 2005 Data Set 41
5.2 Position Accuracy with Respect to Time for 2005 Data Set . . 41
5.3 Iono Correction PRN 1 42
5.4 Fast Correction PRN 1 42
5.5 Iono Correction PRN 5 42

5.6 Fast Correction PRN 5 42
5.7 Iono Correction PRN 9 42
5.8 Fast Correction PRN 9 42
5.9 Iono Correction PRN 21 42
5.10 Fast Correction PRN 21 42
5.11 Position Accuracy for 2010 Data Set 44
5.12 Position Accuracy with Respect to Time for 2010 Data Set . . 44
5.13 Plot zoomed on the 6000-7000 seconds interval 10 degrees . . . 45
5.14 Plot zoomed on the 6000-7000 seconds interval 15 degrees . . . 45
5.15 Ionospheric Corrections over 24 Hours for Satellite PNR 2 . . 46
5.16 Real Time Results 48
5.17 Real Time Results on Time Scale 48
5.18 Jump is Position Accuracy 49
5.19 PRN 15 ionospheric delay using averaged pseudoranges 51
5.20 PRN 15 smoothed ionospheric delay after averaging interval . 52
5.21 Raw, Iono, DGPS and EGNOS distance from origin 54
5.22 Distance from origin 12 hours 55
5.23 Distance from origin, 20 hours 56
5.24 Variation from origin (m) in N-E, 20 hours 56

A.1 The IGP Locations defined in the IGP mask 63
A.2 The IGP Locations with valid GIVEI 65
A.3 The Ionospheric Delay of the Defined IGP Locations 66
A.4 The Contour of the IGP Ionospheric Delays 67
A.5 Four Point Interpolation Algorithm Definition 70
A.6 Three Point Interpolation Algorithm Definition 70
A.7 One degree Interpolation Delays 71
8 Work Flow 74

List of Tables

2.1 Sources of error 3

2.2 EGNOS Message Types 9

2.3 Half of Message Type 25 Format (Velocity Code = 0) 11

2.4 Half of Message Type 25 Format (Velocity Code = 1) 11

2.5 Message 18 parameters 14

2.6 Message 26 parameters 15

2.7 Evaluation of $GIVE_i$ 15

4.1 MBN Message Contents 28

5.1 Mean and Range for the 2005 Data Set 40

5.2 Mean and Range for the 2010 Data Set. 10 ° Elevation Mask . 43

5.3 Statistics for the 2010 Data Set 43

5.4 Statistics for EGNOS in Real-time 47

5.5 Mean and Range for Real-time Data 47

5.6 Mean and Range for the 2010 Data Set. 15 ° Elevation Mask . 50

5.7 Mean and Range for the 2010 Data Set. 5 ° Elevation Mask . 50

6.1 Evaluation of Precision 58

A.2 $GPGSV Message Format 60

A.3 $GPGSV Message Format 61

Nomenclature

CCF	Central Control Facility
CEP	Circular Error Probability
CMR	Compact Measurement Record
CPD	Carrier Phase Differential
CPF	Central Processing Facility
DGPS	Differential GPS
ECAC	European Civil Aviation Conference
ECEF	Earth Centered Earth Fixed
EGNOS	European Geostationary Navigation Overlay Service
ESA	European Space Agency
GEO	Geostationary Earth Orbit Satellite
GIVD	IGP Vertical Delay
GIVEI	Grid Ionospheric Vertical Error Indicator
GLONASS	GLObal NAvigation Satellite System
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
IGP	Ionospheric Grid Point
IPP	Ionospheric Pierce Point
MCC	Mission Control Center

MP	Multipath
NLES	Navigation Land Earth Stations
PRN	Pseudorange Number
RIMS	Ranging and Integrity Monitoring Stations
RTCA	Radio Technical Committee for Avionics
RTCM	Radio Technical Commission for Maritime Services
RTK	Real Time Kinematic
SBAS	Satellite Based Augmentation System
SISNeT	Signal in Space through the Internet
SNR	Signal-to-Noise Ratio
SV	Space Vehicle
TEC	Total Electron Content
TTFF	Time To First Fix
UDREI	User Defined Range Error Indicator
WAAS	Wide Area Augmentation System

Chapter 1

Introduction

The recent evolution in satellite based navigation systems and satellite based augmentation systems (SBAS) have brought new possibilities for research and development in terms of improving the accuracy of positioning without having to cut down on reliability. Navigation has seen along the years a continuous interest and improvements have been made in position accuracy, reliability, and cost as well.

Contributing to the progress of GPS solutions, this study merges two of the most popular ways in order to obtain better accuracy, namely the European Geostationary Navigation Overlay Service (EGNOS) and Differential GPS (DGPS). This comes in the context of an increasing demand in applications that can offer a very accurate and reliable position without asking for high prices. Some of the most common examples that require precise or reliable positioning are: safety of flight, with an accent on reliability, land surveying, traffic management, road pricing, and so on.

GPS accuracy is affected by a number of factors like: noise in the radio signal, atmospheric conditions, and natural elements that obstruct the signal. Noise can create an error of up to 1 meter [Farrell & Barth, 1999]. Objects such as mountains or buildings between the satellite and the receiver can also produce errors, sometimes up to 10 meters [Parkinson & Jr., 1996]. The most accurate determination of position occurs when the receiver has a clear view of the satellites, with no other objects interfering.

To overcome, and also to get around these factors, other technologies, such as AGPS, DGPS, and WAAS have been developed to aid in determining an accurate location.

There are several solutions to answer these demands. EGNOS and DGPS are just part of them. In this paper we test the fusion of these two systems and evaluate the results, in order to decide how to provide the best solution without raising the overall price too high.

The rest of the document is organized as follows:

- Chapter 2 summarizes the theoretical background, comprising error sources and some of the available ways to alleviate them
- Chapter 3 discusses in greater detail the goal of the project, including the reasons that led to it
- Chapter 4 describes the EGNOS system implementation, from the point of data build-up to the point of algorithm design; dual-frequency measurements and DGPS implementation follow
- Chapter 5 presents and discusses the outcome of our work, with analysis of each Matlab plot. EGNOS, DGPS and ionospheric delay estimations are taken into account.
- Chapter 6 evaluates the methods used and brings a final conclusion of the work conducted. Possibilities for improving and future work are considered in the end of the chapter

Chapter 2

Background

2.1 Error Sources In GPS

Ideally GPS requires for user positioning the distance between satellite and user and the satellite's position at the moment of the forementioned measurement. Both time and distance measurements need to be as accurate as possible, but there are different error sources that influence them. Part of them are shared by all users, others are common only for users in a particular area, and some are unique for each user. An approximation of the most important error sources is presented in Table 2.1 [Farrell & Barth, 1999]. As

Type	Errors	Magnitude (m)
Common	Clock and Ephemeris	3.6
Partially common	Ionosphere	7.0
	Troposphere	0.7
Uncommon	Receiver noise	0.1-0.7
	Multipath	0.1-3.0

Table 2.1: Sources of error

we can easily notice the ionospheric delay is the most influential among all possible error sources. For a single frequency receiver the position accuracy is degraded significantly, especially in case we have satellites rising/setting, where high ionospheric delays occur.

2.2 Ionosphere Influence on GPS Signals

The atmosphere affects radio waves in different ways. While in some cases (depending on the frequency used) radio waves bounce back from the atmosphere towards Earth when sent from a ground station, in the GPS case they pass through it, making the communication between satellites and ground stations possible.

On the path towards a destination the signal travels through different layers of the atmosphere, which alter the radio waves - the signal is eventually delayed, changing its speed of travel.

There are two layers which influence significantly the travel time of a wave, the troposphere and the ionosphere. The troposphere is an area of approximately 10 km starting from ground level. The effect caused by it can be easily computed, after recent studies, with a model that makes use of available meteorological data (temperature, air pressure, humidity). After taking into account also the elevation angle of a satellite, the tropospheric delay is determined and subtracted from the total travel time of the signal.

Unfortunately not the same procedure can be repeated with the ionospheric delay. Its computation is more sophisticated, since a model cannot be developed to accurately estimate the time delay caused by it. The previ-

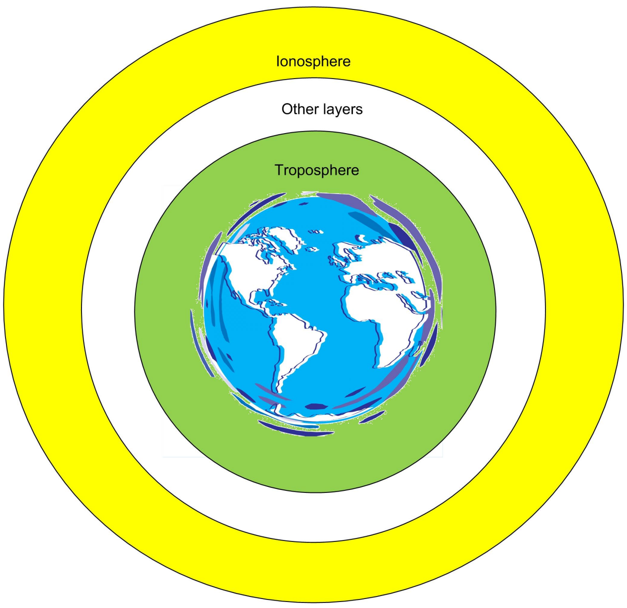

Figure 2.1: Atmosphere layers affecting GPS signals

ous remark is based on the ionosphere's property of unpredictable variation. This is due to Sun's activity, which directly influences the total amount of electrons in the ionosphere. The total electron content (TEC) is the parameter needed to be known in order to deduct the effect upon the GPS signal. According to [Parkinson & Jr., 1996] we have:

$$\delta t_{ion}(f) = \frac{40.3}{f^2} TEC \tag{2.1}$$

where t_{ion} is the ionospheric delay and f is the frequency of the traveling signal. TEC variation follows a daily cycle of high peaks during midday and decreased values along the night. The sun itself goes through a periodical process causing TEC variations, with cycles that last around 11 years. As we can well see the ionosphere is influenced by several factors, making it difficult to forecast.

The key point in estimating the ionospheric delay relays in the fact that the ionosphere influences signals differently based on their frequency. This was one of the main reasons for implementing dual frequency transmission of messages in the satellite's communication system. Using the two signals from the GPS satellite, L1 and L2, and subtracting the travel time of data from each other leads us to the total delay caused by the ionosphere. The whole procedure of estimating the delay is detailed in Chapter 4.

2.3 The EGNOS System

One aspect of our work deals with computing corrections based on the information provided by the EGNOS system. More precisely, *the aim is to improve the accuracy of the position using the slow varying, fast varying and ionospheric corrections* provided by the EGNOS system. The precision obtained will be further on compared with the accuracy obtained through other means (DGPS and dual-frequency receiver), in order to decide which method provides the best solution.

EGNOS, the European equivalent of the American WAAS system, or of the Japanese Multi-transport Satellite Based Augmentation System (MSAS), is Europe's contribution to the Global Navigation Satellite System (GNSS). Figure 2.2 shows the coverage area of the main augmentation and navigation systems designed until this moment.

The coverage area of the EGNOS system is the European Civil Aviation

Conference (ECAC) Area (most European countries, Turkey, the North Sea and the eastern part of the Atlantic), as it can be seen in Figure 2.3.

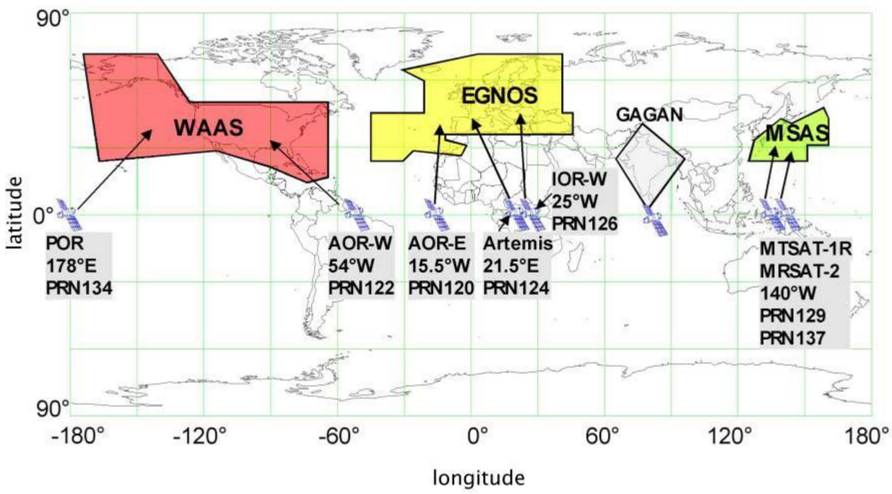

Figure 2.2: Augmentation Systems Worldwide [www.sxbluegps.com]

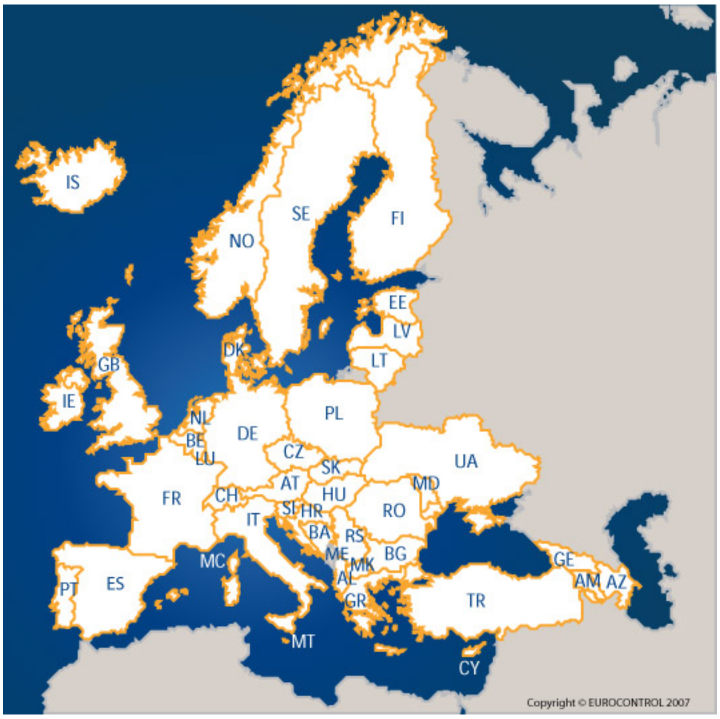

Figure 2.3: EGNOS Coverage Area (ECAC Area) [www.eurocontrol.int]

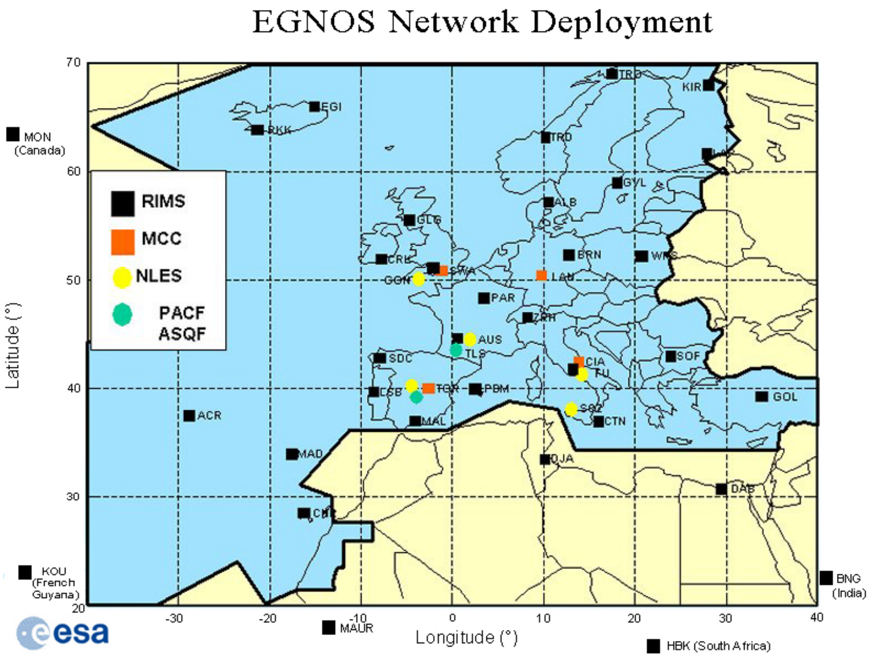

Figure 2.4: EGNOS Deployment Network [www.esa.int]

The SBAS system developed by European Space Agency (ESA) provides navigation signals for land and aeronautical applications along with integrity values for each of the broadcasted signals. The signal broadcasted via Geostationary Earth Orbit (GEO) satellites is generated by the ground segment of the system.

There are 34 Ranging and Integrity Monitoring Stations (RIMS) distributed all over Europe that track the GPS satellites and send the data acquired to the Mission Control Centers (MCC), which computes the range and ionospheric corrections and the integrity parameters for each satellite tracked and generates the GEO satellite ephemeris. The data are then forwarded to the Navigation Land Earth Stations (NLES) which then upload it to the EGNOS transponders that broadcast it to the users on the GPS L1 frequency. Figure 2.4 shows the location of the 34 RIMS, the 4 MCCs and the 4 NLES.

The data are broadcasted in messages of 250 bits with a rate of 250 bps using a GPS like signal on the frequency of GPS L1 (1575.42 MHz). The data will be rate 1/2 convolutional encoded with a Forward Error Correction (FEC) Code to achieve a low bit error probability. So, the receiver has to process 500 symbols per second (sps).

2.3.1 The EGNOS Message

The data blocks are 250 bits long (1 second), and consist of a 8 bit preamble, a 6 bit field indicating the message type, followed by the 212 bits message and a 24 bits Cyclic Redundancy Check (CRC). Figure 2.5 shows the data block format of the EGNOS message.

The preamble is part of a 24 bit unique word distributed over three successive blocks. The three preambles are `01010011`, `10011010`, and `11000110`. The start of the first 8 bit preamble of every other 24 bit distributed preamble will be synchronous with the 6-second GPS subframe epoch.

Table 2.2 shows the message types used by EGNOS. There are three different categories of messages: messages related to satellite information, messages related to ionospheric information and other additional messages.

Message type 1, the PRN mask, indicates to which satellites the data from messages 2 7, and 24 applies to. Messages 2 5 contain fast corrections that apply to the pseudorange measurements. Message 6 offers integrity information for the information contained in messages type 2 5. Message 7 holds the degradation factors in time for the fast corrections. Message 9 contains GEO satellites ephemeris data, while from Message 17 the almanac for the GEO satellites can be decoded. Message 18 provides the ionospheric mask, that indicates the locations which correspond to the delay values extracted from message 26. Message 25 provides slow varying corrections which alleviate the ephemeris and satellite clock errors, while message 24 is a mixed fast and slow corrections message.

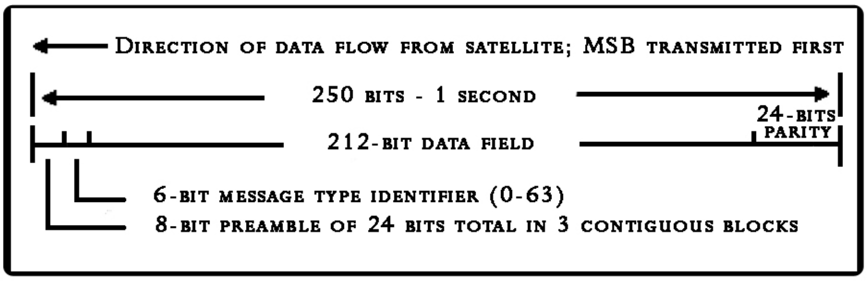

Figure 2.5: EGNOS Data Block Format

Type	Contents
0	WAAS testing
1	PRN mask assignment
2 to 5	Fast corrections
6	Integrity information
7	Fast correction degradation factor
8	Reserved for futrure messages
9	GEO navigation messages
10	Degradation Parameters
11	Reserved for future messages
12	EGNOS Network Time
13 to 16	Reserved for future messages
17	GEO satellite almanacs
18	Ionospheric grid point masks
19 to 23	Reserved for future messages
24	Mixed fast corrections/long term satellite error corrections
25	Long term satellite error corrections
26	Ionospheric delay corrections
27	WAAS Service Message
28	Clock-Ephemeris Covariance Matrix Message
29 to 61	Reserved for future messages
62	Internal Test Message
63	Null Message

Table 2.2: EGNOS Message Types

The attention will further on be focused on messages 1 5, 18, 24, 25 and 26 which contain the corrections needed to obtain a better position accuracy.

2.3.2 EGNOS Fast Corrections

Message Type 1

Message 1 provides the PRN mask, which consists of 210 ordered slots, indicating if data is provided to a particular satellite or not. For example if on the 10th position the value is set to 1, this indicates that information will be provided in messages 2 7, 24 and 25 for that satellite.

Before decoding the PRN mask, it's impossible to know to which satellite each corrections applies to.

The mask transitions to a new one each time a satellite fails or a new one is launched, which will be a very rare event. A 2 bit IODP value will indicate the change. If the IODP of the mask does not agree to the IODP in messages 2 7, 24 and 25, the corrections will not be used until a mask with a matching IODP is decoded.

Messages Type 2 to 5

Each fast corrections message contains a data set for 13 satellites. Message 2 contains data for the first 13 satellites defined in the PRN mask, message 3 for the next 13 satellites and so on. A fast correction message will be sent only in case the number of satellites defined in the PRN mask requires it. For example if only 30 satellites appear in the PRN mask, there is no need for message 5.

Also, if in one of the messages the number of satellites for which data is provided is less or equal to 6 then this data will be broadcasted in the first half of message 24.

For each satellite there is a 12 bit fast correction which has a resolution of 0.125 m with a range from -256 m to $+255.875$ m. Following the correction values are the UDREI values, which indicate the accuracy of the correction or if the correction should even be used.

2.3.3 EGNOS Slow Corrections

Message Type 25

Message 25 provides corrections for satellite ephemeris and clock errors. The data block is divided into two parts. The first bit of each part is the velocity code, which if it's set to 1 indicates that in that part of the message clock drift and velocity corrections are included, otherwise only the clock offset and position corrections are provided, but for two satellites instead of 1. Therefore the message can consist of corrections for 1, 2, 3 or 4 satellites, depending on the values of the velocity bits and how many satellites are being corrected.

The parameters and the corresponding number of bits assigned to each of the parameters is shown in Table 2.3 when the velocity bit is set 0 and in Table 2.3 if the velocity bit is set to 1.

As opposed to the data in Messages 2 5, the data in Message 25 doesn't have to appear in sequence, corrections will be repeated at a higher rate for satellites with faster changing slow term corrections.

Parameter	No. of Bits	Scale Factor	Effective Range	Units
Velocity Code = 0	1	1	–	unitless
PRN Mask No.	6	1	0 to 51	–
Issue Of Data	8	1	0 to 255	unitless
δx(ECEF)	9	0.125	±32	m
δy(ECEF)	9	0.125	±32	m
δz(ECEF)	9	0.125	±32	m
δa_{f0}	10	2^{-31}	$\pm 2^{-22}$	sec
PRN Mask No.	6	1	0 to 51	–
Issue Of Data	8	1	0 to 255	unitless
δx(ECEF)	9	0.125	±32	m
δy(ECEF)	9	0.125	±32	m
δz(ECEF)	9	0.125	±32	m
δa_{f0}	10	2^{-31}	$\pm 2^{-22}$	sec
IODP	2	1	0 to 3	unitless
Spare	1	–	–	–

Table 2.3: Half of Message Type 25 Format (Velocity Code = 0)

Parameter	No. of Bits	Scale Factor	Effective Range	Units
Velocity Code = 1	1	1	–	unitless
PRN Mask No.	6	1	0 to 51	–
Issue Of Data	8	1	0 to 255	unitless
δx(ECEF)	11	0.125	±128	m
δy(ECEF)	11	0.125	±128	m
δz(ECEF)	11	0.125	±128	m
δa_{f0}	11	2^{-31}	$\pm 2^{-21}$	sec
δx rate-of-change (ECEF)	8	2^{-11}	±0.0625	m/sec
δy rate-of-change (ECEF)	8	2^{-11}	±0.0625	m/sec
δz rate-of-change (ECEF)	8	2^{-11}	±0.0625	m/sec
δa_{f1}	8	2^{-39}	$\pm 2^{-32}$	sec
Time-Of-Day Applicability t_0	13	16	0 to 86384	sec
IODP	2	1	0 to 3	unitless

Table 2.4: Half of Message Type 25 Format (Velocity Code = 1)

Message Type 24

Type 24 message is a Mixed Fast Correction/Long Term Satellite Error Correction Message, which is broadcasted only in the situation when there are 6 or fewer satellites in a block message type 2 5 or 25.

The first half of the message is reserved for fast corrections and consists of 6 data sets as described in 2.3.2, followed by the IODP and a 2-bit Block ID indicating which block is replaced by this one. The Block ID take values in the range 0 to 3, 0 in case Message Type 2 is replaced, 1 for Message Type 3 and so on.

The second part of the Mixed Corrections Message holds the slow correction parameters, with a total of 106 bits and a structure as the one presented in Tables 2.3 and 2.3.

2.3.4 EGNOS Ionospheric Corrections

Messages type 18 and 26 contain the information necessary for computing the ionospheric delays.

The corrections are broadcast as vertical delay estimates at specified ionospheric grid points (IGPs), applicable to the signal on L1. [RTCA, 2001].

The IGP locations are defined in 11 bands, from 0 to 10. Bands 0 to 8 are vertical and bands 9 and 10 are horizontal. There are 1808 IGP locations defined in bands 0 to 8, with locations denser at lower latitudes because the distance covered by a degree of longitude becomes smaller at higher latitudes, and 384 IGP locations in bands 9 and 10. Figure 2.6 shows bands 0 to 8 of the IGP grid mapped to the Earth map.

The IGP grid has a 5° spacing at the equator, and increases to 10° north of N55° and south of S55°, and finally becomes spaced 90° at N85° and S85° around the poles.

Message Type 18

Due to the fact that it would be impractical to broadcast IGP delays for all possible locations, a mask is broadcast, in message type 18, to define the IGP locations that can be used further on in computations.

Message type 18 broadcasts the IGPs within a band. The IGPs are de-

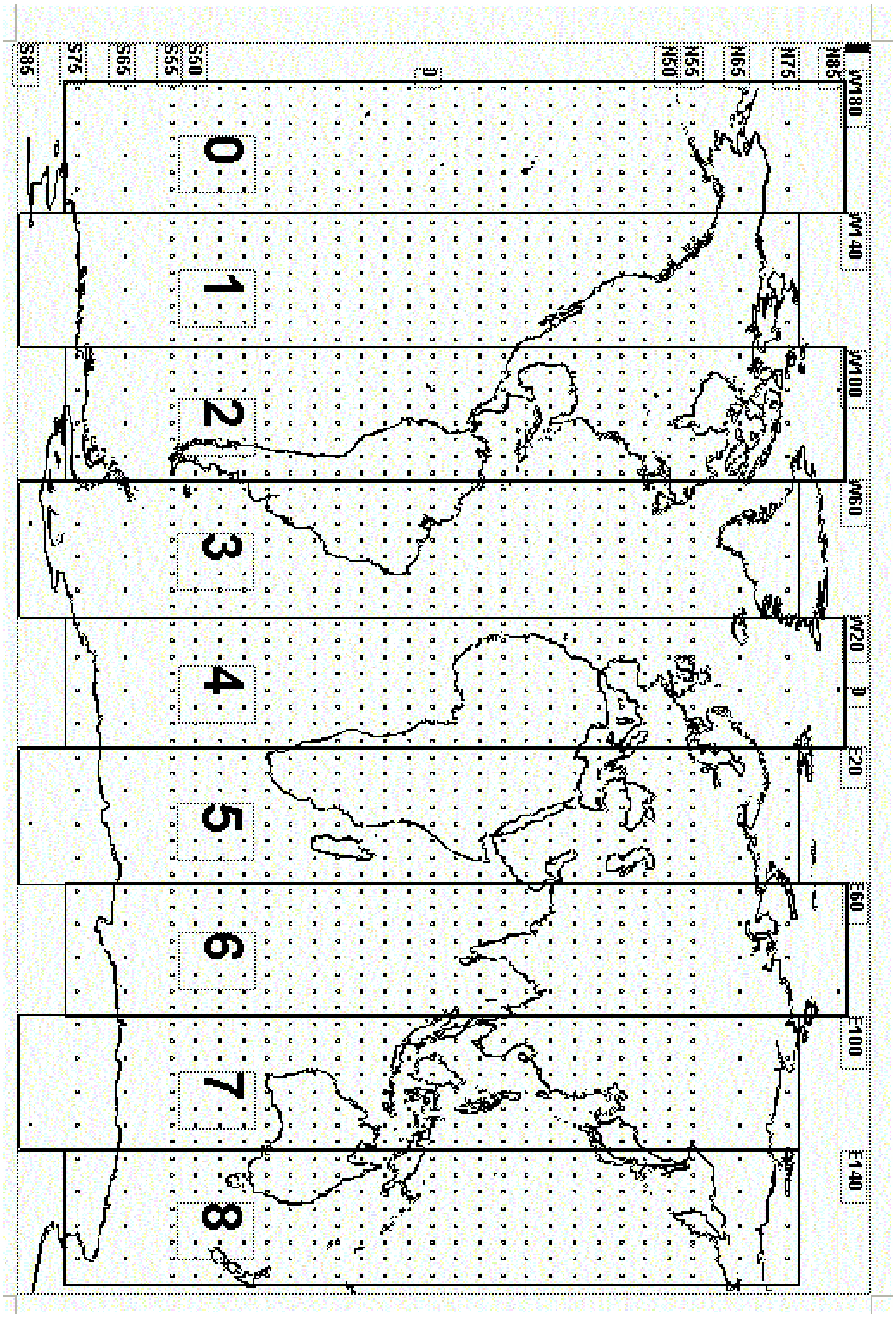

Figure 2.6: The IGP Grid
[www.gpsinformation.net]

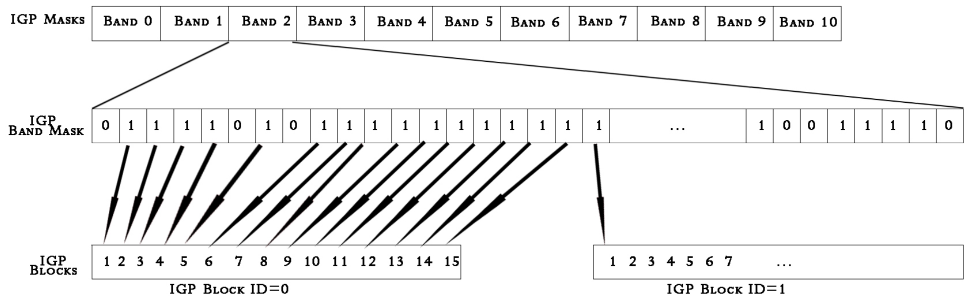

Figure 2.7: The Ionospheric Grid Mask

fined, by using a mask that works similar to the PRN mask from message type 1. If the slot on position 'n' is set to 1 this means that ionospheric correction information is being provided in message type 26 for the 'n'-th IGP location in the band.

An example of an ionospheric grid mask is shown in Figure 2.7. The IGP mask information is provided for band 2, and every slot that is set to 1 indicates there is ionospheric information available. The mask is further on organized in blocks. Each block addresses 15 IGPs from the IGP mask that are set to 1. So block 1 (with block id = 0) will address the first 15 IGPs from the IGP mask that are set to 1.

Table 2.5 shows the parameters in message 18. The first two pairs of 4 bits indicate the number of bands being broadcasted and the number of the band for which the mask contained in the message applies. The IODI parameter provides a means of ensuring that messages 18 and 26 are properly correlated. Both types of messages have the IODI parameter which has to have the same value in both cases. The IODI will change each time the IGP mask changes which is expected to happen rarely. In case it's not the same, another message 26 has to be decoded, until the two values are the sane.

Parameter	No. of bits	Scale Factor	Effective Range	Units
Number of Bands being Broadcast	4	1	0 to 11	unitless
Band Number	4	1	0 to 10	unitless
Issue of Data - Ionopshere (IODI)	2	1	0 to 3	unitless
IGP Mask	201	-	-	unitless
Spare	1	-	-	-

Table 2.5: Message 18 parameters

Message Type 26

Message type 26 provides the vertical ionospheric delays and their accuracies for the IGPs identified by the band number and the IGP number.

Parameter	No. of bits	Scale Factor	Effective Range	Units
Band Number	4	1	0 to 10	unitless
Block ID	4	1	0 to 13	unitless
For Each 15 Grid Points	13	-	-	
IGP Vestical Delay Estimate	9	0.125	0 to 63.875	unitless
Grid Ionopheric Vertical Error Indicator (GIVEI)	4	1	0 to 15	meters
IODI	2	1	0 to 3	unitless
Spare	7	-	-	

Table 2.6: Message 26 parameters

$\mathrm{GIVEI_i}$	$\mathrm{GIVE_i}$ Meters	$\sigma^2_{\mathrm{i,GIVE}}$ Meters
0	0.3	0.0084
1	0.6	0.0333
2	0.9	0.0749
3	1.2	0.1331
4	1.5	0.2079
5	1.8	0.2994
6	2.1	0.4075
7	2.4	0.5322
8	2.7	0.6735
9	3.0	0.8315
10	3.6	0.1974
11	4.5	1.8709
12	6.0	3.3260
13	15.0	20.7870
14	45.0	187.0826
15	Not Monitored	Not Monitored

Table 2.7: Evaluation of $\mathrm{GIVE_i}$

Due to the large number of IGPs defined in every ionospheric mask, each message 26 will only provide information for one block of IGP locations, which consists of 15 defined IGPs. Block 1 (block ID 0) contains the IGP corrections for the first 15 IGPs in the band mask, block 1 (block ID 1) for the next 15 and so on. Therefore, if the IGP mask contains 40 IGPs with values set to 1, there will be 3 messages type 26 (3 blocks $\times$ 15 IGPs) sent for that band, with the message for block number 3 having only 4 pairs of (ionospheric delay, accuracy) values.

Each message contains a band number and a block ID which indicates the location of the IGPs in the band. Table 2.6 shows the parameters and the bit allocation in message 26.

Note that the 9-bit IGP vertical delay has a 0.125 m resolution (111111111 = 63.875 m), for a 0 − 63.750 m valid range. If the range is exceeded, a 'don't use' flag will be set.

Another aspect that needs mentioning is related to the GIVE indicator. The GIVEI is extracted from the message in form of integer values in the range 0 to 15, so in order for it to supply useful information each of the indicators has to be converted to its equivalent in meters. The values of GIVEI and their equivalents are given in Table 2.7.

Chapter 3

Problem Definition

GPS applications have always strived for more and more performance when acquiring a position. Different GPS enhancement methods have been developed along the years, like SBAS, DGPS, Inertial Navigation Systems, and Assisted GPS. In the quest of accuracy SBAS systems were developed in several areas of the globe, EGNOS being one of them. EGNOS takes care particularly of the European continent, but extends more than the geopolitical borders of Europe.

DGPS is another way of improving the positioning the accuracy of the users, but it doesn't provide the integrity of EGNOS. It can be said that EGNOS is a wide area DGPS that addresses each error source it can correct in particular, while DGPS offers just a general correction of the pseudoranges.

In the preliminary instance of the study the target was to make a comparison of EGNOS and DGPS, and therefore to observe the advantages brought by each of the systems. Afterwards, in theory, by combining the two, it is possible to obtain the accuracy of DGPS and reliability of EGNOS for users. Moreover, the accuracy of a dual-frequency DGPS station is brought to a single frequency user, which decreases the overall price of accurate positioning significantly. This means a step forward in the cost effective implementation of fleet management, road pricing, and driving alert systems, among others.

The ionospheric delay is the error source that affects measurements the most, so estimating this bias improves significantly the positioning results. EGNOS provides a ionospheric delay correction system based on the interpolation of several measured grid points in the ionosphere, but because these points are spaced 5 or more degrees apart, variation values can occur for the delays estimated using the interpolation.

In order to deal with this variation we made appeal to the benefits of a DGPS base station. The DGPS station mentioned here is based on a dual frequency receiver and provides the possibility of deducting the ionospheric delay from the two frequencies pseudoranges.

In our approach all the measurements are downloaded from the receiver, here including the SBAS messages. The aim we set for our project is to analyze these methods individually and to evaluate their performance. We show results obtained using EGNOS, DGPS, dual frequency receiver measurements, and we also present a hybrid method of improving the accuracy by substituting the ionospheric correction used in the EGNOS system with one obtained from a dual frequency receiver.

As mentioned in the specifications of the DG14 receiver, [Thales, 2008] , the EGNOS enabled receiver used in our project, the precision offered is of 1.8 m CEP and 3.8 m in R95 in the EGNOS assisted case, whereas in the autonomous case the precision is 3.0 m CEP and 5.0 m R95. Therefore, theoretically, by combining EGNOS with the dual-frequency measurements, we will take advantage of the precision offered by the dual-frequency measurements (1.5 m CEP), and of the reliability provided by EGNOS. What led us to think this could be a feasible way to get precise measurements is the fact that the accuracy of the ionospheric correction supplied by EGNOS is not as good as the one a dual-frequency receiver can offer, considering the fact that EGNOS acts as a Wide Area DGPS.

Raw data measurements are taken from the receivers, comprising pseudoranges from available satellites, receiver time, ephemeris and:

- SBAS messages, from the EGNOS receiver
- pseudoranges on L2 frequency, from the dual frequency receiver

First of all data will be downloaded and saved in files which are to be processed later on, in order to verify the implementation on the same case. After checking that the corrections are applied properly, a real-time version will follow.

The medium for testing and developing the project is Matlab, while additional software from the receivers and ESA will be used for examination and double-checking of the obtained data.

Although our initial goal was to improve the position obtained through GPS computations to submeter level in order to be able to use GPS measurements for precise positioning application, and try to establish the basis a system that would take advantage of this service, due to the time limitations and the problems encountered along the way, we have limited our work to implementing an improved EGNOS system and analyzing if the accuracy provided is suitable for this kind of applications.

Chapter 4

System Design

4.1 EGNOS System Architecture

The system architecture is not very complex. EGNOS data, which consists of SBAS messages, and additional GPS information (pseudoranges and ephemeris data) are extracted from a receiver. The GPS data are processed in Matlab in order to obtain the receiver position and the EGNOS information helps improve the accuracy.

The receiver used is a DG-14 Ashtech sensor. We consider worth mentioning some of its characteristics. It is a single frequency, EGNOS enabled receiver, with 14 parallel channels that can all be used for tracking GPS satellites, or 2 of them can be used for EGNOS satellites and the remaining ones for GPS. DG14 receiver is able to provide raw data output for code and carrier, with rates up to 20 Hz.

In terms of performance, according to [Thales, 2008], the DG14 sensor offers 3.0 m accuracy in Horizontal CEP and for Horizontal R95(95%) it provides an accuracy of 5.0 m. Things improve slightly with SBAS aiding to 1.8 m CEP and 3.8 m in Horizontal R95(95%). The receiver also offers a precision of 200 ns in stand alone operation and 50 ns in differential mode.

The time required for a Time To First Fix (TTFF) when performing a Cold Start is 90 s, 35 s for a warm start, 11 s when performing a hot start and s are require for re-acquisition.

The fact that the receiver is able to provide raw data will prove to be very helpful in establishing a study case related to degree of accuracy improvement offered by the corrections obtained. The position of the receiver will be computed in four different ways in order to test the magnitude of the EGNOS ionospheric.

In the first case, the position will be computed using just raw pseudoranges, while in the other two cases the raw pseudoranges will be corrected using an ionospheric model and the EGNOS corrections. In the last case the position will be computed using the corrected pseudoranges obtained from the receiver. In this last case the pseudoranges are corrected for atmospheric effects and relativistic errors.

The ionospheric model that is going to be used is the one established by Klobuchar, which is said to reduce only by 50% the errors caused by the ionospheric effects.

The first attempt in correcting the position with EGNOS correction involved only ionospheric correction, due to the fact that they provide the biggest level of correction and that they are difficult to model, but this proved to be wrong because the EGNOS corrections show good results in accuracy improvement only when all corrections are used together. So in the second attempt, slow and fast varying corrections were added.

Another issue that was raised during the development of the program involved using real-time data or post-processed one. This is because EGNOS messages are broadcasted every second, and in real-time processing at each computation along with EGNOS messages, pseudoranges and occasionally (when the ephemeris change) ephemeris data need to be extracted, which leads to computing a far smaller number of positions. Due to the fact that both ways present their advantages and disadvantages, the Measurement and Results section of our project will show the results in both cases.

4.2 EGNOS Implementation

The aim of the program is to acquire in real-time data from the EGNOS system, decode it and use together with GPS information in order to obtain a precise position of the receiver. In order to get the real-time application, the first attempt was done on post-processed data, which simplifies things a little

The program extracts data from the receiver and processes it into fast correction, slow correction and ionospheric information, which will be used along with raw pseudoranges and ephemeris data. SBAS messages are sent every second, but the program takes approximately 3 minutes to acquire all necessary data and compute the correction.

The fast and slow correction don't require any special interpretation, they just have to be decoded following the format presented in Sections 2.3.2 and 2.3.3 and then be added to the corresponding pseudorange measurements or to the satellite position vector computed from the ephemeris data respectively.

The next part of this section will focus mainly on obtaining the ionopheric corrections. Greater attention is paid to this aspect due to the fact that it requires more complex computations and the corrections have a greater weight in obtaining a more accurate position.

In order to obtain the ionospheric delay values, the messages extracted from the receiver have to be decoded and interpreted and computations have to be performed on the decoded data. This is done in the program `ionospheric_computation.m`. The program follows the steps and the algorithm presented in [RTCA, 2001] in order to obtain the ionopheric delay for each satellite tracked by the receiver.

The ionospheric delay is proportional to the number of free electrons along the GPS signal path, called the total electron content (TEC). TEC, however depends on the time of day, the time of year, solar activity and the geographic location (electron density levels are minimum in midlands regions and highly irregular in polar, auroral and equatorial regions as it will be observed in the tests done in the project).

Therefore, the ionospheric delay varies from one moment to another and from day to night, proportionally to the Total Electron Content (TEC) values. This means that every satellite signal will be affected by a different amount of ionospheric delay, which also depends on the distance the signal travels through the ionosphere. Signals coming from satellites with low elevation angles will travel a longer distance through the ionosphere then signals coming from satellites with high elevation angles, so they will be delayed more then the others.

To obtain the ionospheric corrections there are four major steps that have to be completed. The first thing is to determine the elevation angle of each tracked satellites in order to determine the ionospheric grid signal pierce point as shown in A.1.1.

After having found the pierce points, the coordinates are rounded to the closest IGP point with a lower latitude and longitude, and the other 3 points

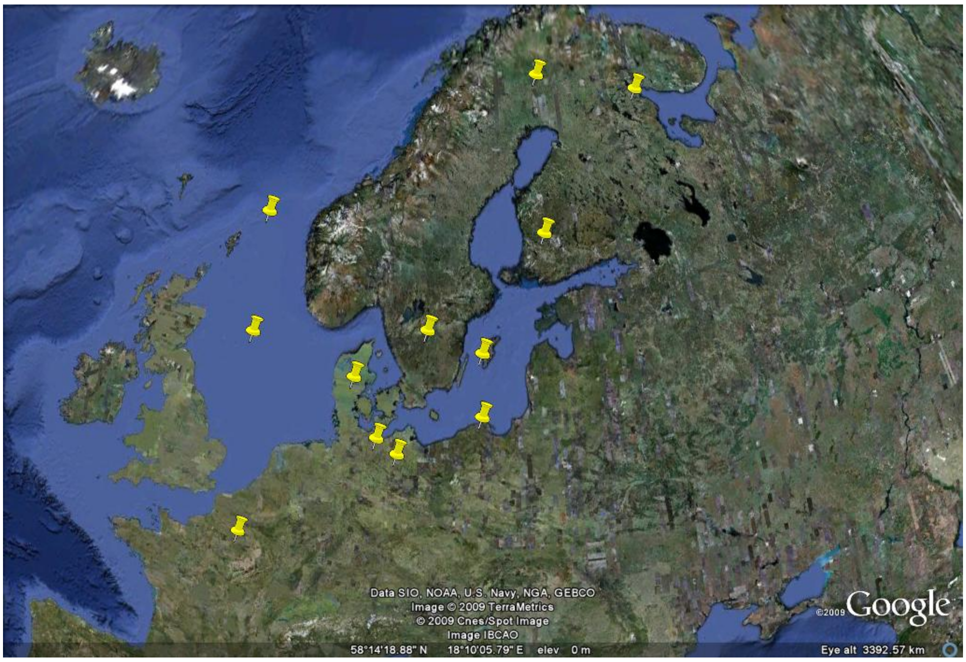

Figure 4.1: Ionospheric Pierce Point Locations

that surround the pierce point are found, in order to form a square surrounding the pierce point.

Considering the fact that pierce points are 350 km above Earth and a 10° elevation mask is used, the pierce points cannot be more than 2000 km away from the receiver position. Figure 4.1 shows the position of the pierce points at the moment one of the tests was done.

In order to determine the vertical delay of the pierce point, the GIVD values have to be obtained. This is done for the entire area covered by EGNOS, or in the case of this program, only for bands that cover Europe from −60° to 90° longitude, which is coincident to the only area for which EGNOS provides information, at least at the time the paper was written.

This process starts by creating a matrix from the IGP masks broadcasted in message 18, and then according to the data in this mask, the GIVD values are put in a separate matrix only in the locations where there is a 1 in the IGP matrix. The process is explained in greater detail in Appendix A, section A.2.

The next step is to determine the delay of each of the IGP locations previously determined. This is done by extracting from the GIVD matrix the proper values, that correspond to the locations that surround the pierce point and determine through interpolation the ionospheric vertical delay of the pierce point.

The values obtained represent the vertical ionopheric delay, and in order to obtain the slant ionospheric delay, the value that corrects the pseudorange measurement, the vertical delay has to be multiplied by an obliquity factor. A more detailed description of the program is presented in Appendix A. The same procedures are followed also in the post-processing case, with the difference that the number of positions plotted will be much greater, due to the fact that in this way no SBAS messages are skipped.

4.3 Signal in Space Through the Internet

SISNeT, project developed by ESA and released for use in 2002, is a new technology that combines the capabilities of satellite navigation and the Internet. EGNOS messages are now available over the Internet and in real time via SISNeT.

Even though satellite broadcasting through geostationary satellites (GEOs) has proved to be an efficient strategy for avionic applications for some applications GEO broadcasting may have some limitations because building obstacles in cities or rural canyons can interfere with the GEO reception.

Any user with access to the Internet (usually through wireless networks - GSM or GPRS) can access EGNOS through SISNeT, irrespective of the GEO visibility conditions. No EGNOS receiver is needed.

Therefore, through the use of SISNet the same messages that would be downloaded from the receiver can now be downloaded from the Internet in real-time, which could lead to improvements in accuracy for a receiver that is not EGNOS enabled.

According to [Toran et al., 2004], SISNeT data server now gets data directly from the Central Processing Facility, hence receiving messages in advance of GEO users, making it a feasible solution for acquiring the necessary EGNOS messages without the need to have an EGNOS enabled receiver, thus maybe saving some money. The only disadvantages this solution presents is related to the fact that the Internet delay is unpredictable, as it depends on traffic and packet routing. In the worst case, some messages may arrive after their timeout intervals or in some cases some messages may get lost. The effect of message loss differs according to its type. For example, missing a message type 1 (EGNOS mask) is worse than missing the GEO ephemeris message (message type 9). The logical impact of these effects is degradation of the navigation solution.

ESA has also made available free software for interpreting EGNOS data that proves to be very helpful in understanding and implementing a system that uses EGNOS. The following will present only the most important applications used.

SISNeT User Application Software

The SISNeT User Application Software was developed as a tool that allows to analyze the contents of the EGNOS messages obtained via the Internet, as well as to calculate other parameters derived from the EGNOS signal. SISNeT UAS connects the user to the EGNOS GEO message broadcasts that are available via the ESA SISNeT service. The EGNOS messages are received by the UAS and it decodes the contents of these messages in real time and displays them, making it a very good tool for checking the correct functionality of our application with real-time data. The tools that are presented in the next section helped test the application with post-processed data, making it possible to test if the application decodes and interprets the data correctly, being able to test one message at a time, through SISNeT Teacher, or all the messages in a time period through Mentor.

SISNeT Teacher and SISNeT Mentor

SBAS MeNTOR (SBAS Message Generator) is a software which logically follows the idea of SBAS TeACHER. Where the TeACHER enables users to modify contents of SBAS messages in visual way, the SBAS MeNTOR continues with enabling users to define the desired values and states of the whole SBAS (Space Based Augmentation System, e.g. EGNOS, WAAS) and let the software generate all the specific messages of the selected period. The same task could be achieved by TeACHER software, but it would take more time (in order of magnitude) and user would be responsible of linking the data together flawlessly, which is next to impossible.

The MeNTOR works with ESAs EMS (EGNOS Message System) file format and it is possible to load real EGNOS (SBAS) data for a selected period. Loaded EMS files are then transformed to the time dependent set of variables representing the original content. User can then modify some of them in any way and then save the result back to EMS file format.

4.4 Dual-frequency Data Processing

In our project the ionospheric delay estimation has two approaches. One is by handling the data provided by EGNOS and the second one employs a professional dual-frequency receiver. Here we detail the principle for the second method.

It is known that the ionospheric delay is frequency dependent so by using data received on both the civilian L_1 frequency f_1 (1575.42 MHz) and the military L_2 frequency f_2 (1227.60 MHz) we can obtain a good estimate of the delay caused by the ionosphere. The formula that applies in computing the delay is based on the pseudoranges measured by the receiver and on a coefficient composed out of the L_1 and L_2 frequencies [Strang & Borre, 1997].

$$d\,P_{ion} = \frac{f_2^2}{f_2^2 - f_1^2}(P_1 - P_2) \tag{4.1}$$

The receiver used in the experiments is an Ashtech Z-Xtreme^{TM}. In order to operate real-time we have used console commands for the communication with the receiver from Matlab. For requesting the pseudorange, phase measurements, elevation angle, and Doppler effect (and many others) the command that is sent towards the receiver is *'$PASHQ,MBN'*. The response of this command is presented in the following example from an experiment made on the 3rd of November 2009 at 11:00 o'clock.

```
$PASHQ,MBN
$PASHR,MPC,00608,06,24,30,151,01,000,22,5,049,08,-007252952.448,75.3202357,-03060.03206,
-00.13,200,032,22,5,046,-04,-007252952.452,75.3202350,-03060.03290,-00.08,200,032,22,
5,042,00,-005645838.636,75.3202171,-02384.43967,-01.08,200,061
$PASHR,MPC,00608,05,02,32,056,02,002,24,5,050,03,027874538.331,75.1660424,01476.66464,
-00.10,200,032,22,5,048,02,027874538.327,75.1660437,01476.66555,00.16,200,032,22,
5,043,00,021730357.965,75.1660128,01150.64028,-00.39,200,054
$PASHR,MPC,00608,04,31,51,285,03,002,24,5,053,02,-015054917.140,71.7644835,-01055.54847,
00.12,200,032,22,5,051,-01,-015054917.143,71.7644838,-01055.54908,00.04,200,032,22,
5,047,00,-011726582.606,71.7644581,-00822.52066,00.36,200,018
$PASHR,MPC,00608,03,30,61,119,07,002,24,5,055,05,032620463.447,68.3499821,01923.78892,
00.37,200,032,22,5,053,01,032620463.443,68.3499819,01923.78892,-00.06,200,032,22,5,
048,00,025418534.230,68.3499594,01499.05658,00.31,200,027
$PASHR,MPC,00608,02,29,72,201,08,002,24,5,054,02,015326000.480,68.2836621,-01029.23426,
00.07,200,032,22,5,054,-00,015326000.475,68.2836624,-01029.23433,-00.32,200,032,22,
5,049,00,011944988.544,68.2836389,-00801.99555,-00.10,200,051
$PASHR,MPC,00608,01,01,68,118,10,000,22,5,056,08,000476972.618,68.4422311,01414.49216,
00.22,200,032,22,5,054,03,000476972.615,68.4422337,01414.49193,00.28,200,032,22,5,
050,00,000356060.157,68.4422030,01102.20741,-00.22,200,024
```

```
$PASHR,MPC,00608,00,12,27,108,12,002,24,5,049,01,045778541.645,77.3775770,03180.04542,
-00.21,200,032,22,5,047,-00,045778541.636,77.3775808,03180.04596,00.55,200,032,22,
5,042,00,035661338.869,77.3775563,02477.93102,00.50,200,050
```

The receiver outputs a separate message for each satellite, each starting with the *'$PASHR,MPC'* string.

The useful parameters for our project from the outputted messages are the space vehicle (SV) PRN number, azimuth, elevation,carrier phase,code transmit time and Doppler measurement, which we obtain according to the information from Table 4.1. We extract these elements for the C/A code, PL1 code and PL2 code for further computations. For the tropospheric delay we need just the elevation and azimuth provided by the satellite, but for the ionospheric delay we need to have precise pseudorange measurement. Pseudoranges can be obtained either from the *code transmit time* (defined here as P) or from the *carrier phase* data in number of wavelengths (defined as Φ*)*. The Doppler measurement will show either if the satellite is rising or descending. The P pseudorange formula from the code transmit time (t) :

Type	**Size**	**Contents**
unsigned short	2	sequence tag (unit: 50 ms) modulo 30 minutes
unsigned char	1	number of remaining struct to be sent for current epoch
unsigned char	1	satellite PRN number
unsigned char	1	satellite elevation angle (degree)
unsigned char	1	satellite azimuth angle (two degree increments)
unsigned char	1	channel ID (1 - 12)
C/A code data block 29 bytes		
unsigned char	1	Warning flag
unsigned char	1	Indicates quality of the position measurement (good/bad)
char	1	(set to 5 for backward compatibility)
unsigned char	1	Signal to noise of satellite observation (db.Hz)
unsigned char	1	Spare
double	8	Full carrier phase measurements in cycles
double	8	Raw range to SV (in seconds) i.e., receive time - raw range = transmit time
long	4	Doppler (10^{-4} Hz)
long	4	bits: 0 - 23 Smooth correction (bit 0-22 = magnitude of correction in cms, bit 23 = sign) bits:24-31 Smooth count, unsigned. as follows: 0 = unsmoothed, 1=least smoothed, 200 = most smoothed
P code on L1 block, same format as C/A code data block		
P code on L1 block, same format as C/A code data block		
unsigned char	1	Checksum, a bytewise exclusive OR (XOR)
total bytes	95	

Table 4.1: MBN Message Contents

$$P = t \times v_{light} \times 10^{-3} \tag{4.2}$$

The 10^{-3} factor is due to the fact that t is measured in milliseconds (ms).

The pseudoranges obtained from the carrier phase are the most precise ones as they are measured till fractions of wavelengths, so that means they are accurate up to centimeter level. The problem which appears is that we will have an unknown number of wavelengths called ambiguity (N) that must be subtracted from the carrier phase value in order to obtain the whole number of wave cycles that are in between the satellite and the user. The ambiguity can be estimated by using the formula from [Strang & Borre, 1997]:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & \lambda_1 & 0 \\ 1 & \alpha & 0 & 0 \\ 1 & -\alpha & 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \rho^* \\ I \\ N_1 \\ N_2 \end{bmatrix} = \begin{bmatrix} P_1 \\ \Phi_1 \\ P_2 \\ \Phi_2 \end{bmatrix} \tag{4.3}$$

, where $\alpha = (f_1/f_2)^2$, λ_1 and λ_2 are the wavelengths, ρ^* is the true pseudorange without errors, and I is the ionospheric delay. Φ_1 and Φ_2 are here the phase measurements in meters, obtained from the carrier phase multiplied by the wavelength of each signal frequency (0.190 m and 0.128 m respectively).

As a result, we will obtain:

$$I = (-\lambda_1 N_1 + \lambda_2 N_2 + \Phi_1 - \Phi_2)/(\lambda - 1) \tag{4.4}$$

All would seem nice and smooth up to this point but things are not that easy. The code transmit time is affected by many elements: clock errors, ephemeris errors, ionospheric and tropospheric delays, and multipath. All of the effects besides multipath can be removed by differencing the pseudoranges obtained on both frequencies. The problem is that multipath can affect the pseudoranges with several meters, so when trying to obtain the ionospheric delay through differencing the pseudoranges multipath influences the result.

As the ambiguity N is solved also using code pseudoranges the phase pseudorange will be correlated to the code errors due to multipath.

In order to mitigate the multipath issue we use carrier phase measurements with the ambiguity number deducted through Real Time Kinematic (RTK) positioning. This time a master receiver is sending data to the rover (both Z-Xtreme), and the carrier phase is extracted from the last one. There are two possible ways of obtaining the ambiguities:

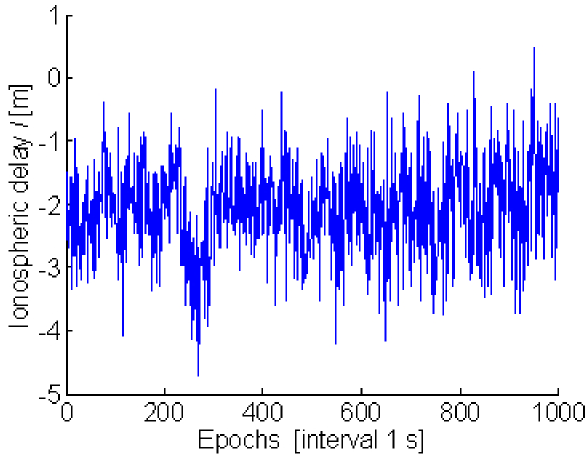

Figure 4.2: Ionospheric delay using pseudoranges

- downloading observation data in real-time from the two receivers and do the computations in Matlab (SW Approach)
- setting up the preset RTK mode for the rover and the base station and extract directly the ambiguities from the rover (HW approach)

The last way uses Ashtech Z-Xtreme Instant RTK function for computing the data, which can go as low as one epoch of data needed for obtaining the pseudoranges, but more epochs are used for increased reliability.

4.4.1 SW Approach

Both receivers in our case have knowledge of pseudoranges and carrier phases on the L1 and L2 frequencies.

RTK implies choosing the common satellites for our receivers in order to obtain the needed data. The common satellite with the highest elevation will be chosen as the reference satellite since it is the most reliable one. The purpose for choosing a reference satellite is double differencing. This is done using pseudoranges and carrier phases between two satellites and two receivers. We apply the formula from (4.3) with all the data double differenced now.

The equation for double difference with X as a general parameter is as follows:

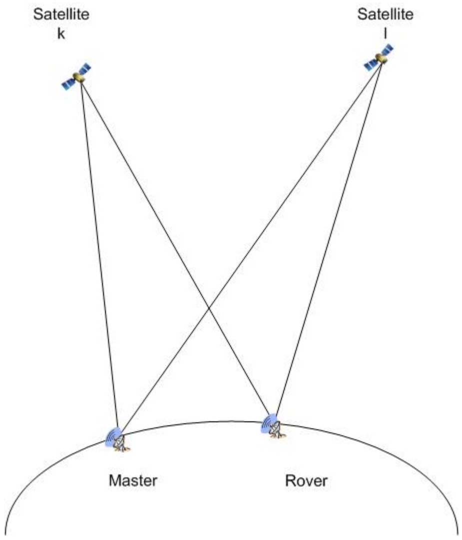

Figure 4.3: Double Differencing

$$X = (X_M^k - X_M^l) - (X_R^k - X_R^l) \tag{4.5}$$

, where X_M^k is observation data from the master receiver and satellite k and X_R^k is the same type of information from the rover receiver and satellite k (same procedure repeated for satellite l respectively). X will sequentially be the pseudorange and the carrier phase on both frequencies. Several epochs are needed for filtering in order to obtain accurate ambiguities. The overall epochs are averaged and the mean is used for further computations.

In the SW approach both master and rover receiver communicate on serial ports with Matlab, which directly sends commands and reads data from the receivers. For testing and redundancy purposes, Ashtech Evaluate software was used to verify the information from the command lines.

The source code is based on the Goad method for deducting the ambiguities. Following the procedure detailed in [Strang & Borre, 1997] we build a weight matrix for equation (4.3) since the error in pseudoranges can be as high as 2-3 meters due to slow chipping rate and the standard deviation for carrier phase is just a few millimeters.

$$C = \begin{bmatrix} 1/0.3^2 & 0 & 0 & 0 \\ 0 & 1/0.005^2 & 0 & 0 \\ 0 & 0 & 1/0.3^2 & 0 \\ 0 & 0 & 0 & 1/0.005^2 \end{bmatrix}$$

The Goad method uses the obtained ambiguities N_1 and N_2 as reals numbers. In order to get the closest integer number for them a special procedure is required. The two real ambiguities N_1 and N_2 develop the next set of equations:

$$K_1 = [N_1 - N_2] \tag{4.6}$$

$$K_2 = [60N_1 - 77N_2] \tag{4.7}$$

The coefficients for N_1 and N_2 in equation (4.7) refer to the ratio of the two broadcast frequencies for the satellite signal (L1=154*10.23 MHz and L2=120*10.23 MHz), where $\frac{L1}{L2} = \frac{154}{120} = \frac{77}{60}$. These two equations lead to the integer ambiguities:

$$\hat{N}_2 = (60K_1 - K_2)/17 \tag{4.8}$$

$$\hat{N}_1 = \hat{N}_2 + K_1 \tag{4.9}$$

Finally, the complete carrier phase pseudoranges are:

$$P1 = \Phi_1 - \lambda_1 N_1 \tag{4.10}$$

$$P2 = \Phi_2 - \lambda_2 N_2 \tag{4.11}$$

We follow the formula for computing the ionospheric delay from (4.1), but this time we use the carrier phase pseudoranges instead of the code delay based pseudoranges. The obtained result should be more precise due to the smaller influence the carrier phase multipath has on the final result compared to the multipath from the code measurements.

4.4.2 HW Approach

For the HW Approach a special setup was made following the instructions in [Navigation, 2002]. The two receivers are connected to two dual-frequency antennas on the top of the roof of building C1, and have a baseline of approximately 1 m. We have attached a Satel Radio Modem (SATELLINE-3AS) to each receiver in order to exchange data. For the base station we inserted an

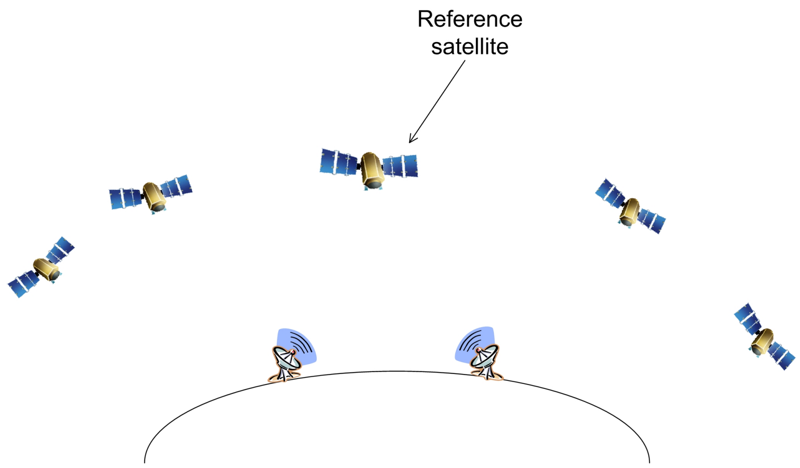

Figure 4.4: Sky view reference satellite

accurate measurement of antenna's position (cm level), required by the RTK Base mode. The rover was configured to Static RTK mode.

Three different message types are available for RTK initialisation. First one is The Radio Technical Commission for Maritime Services (RTCM) standard which is common to all receivers, second is The Compact Measurement Record (CMR) format which has a reduced bandwidth compared to the first, and third is Ashtech standard DBN message, which is the most compact type available. Therefore we have used DBN messages in order to obtain measurements as fast as possible in order to optimize the size of the handled message.

While in RTK mode, the rover is able to compute the ambiguities with the data received from the base station. At least five common satellites are required in order to solve the ambiguities. The more satellites are used the less epochs are needed for processing to ensure the accuracy of the result.

The satellite with the highest elevation is chosen as the reference satellite for double differencing purposes (fig. 4.4), because it's the one guaranteed to be in lock more time when compared to the other satellites in sight. A Kalman filter is employed for identifying the solution. A set of possible solutions are saved and after further epochs processed in the same way, the common solution is selected. A new message becomes available for requesting, which contains valuable Carrier Phase Differential (CPD) data.

Up to this point the project objectives were facile. Problems appeared in decoding the proper message from the receiver. The procedure was difficult since the data provided was in binary and the message documentation [ZX-TechRef] was incomplete. The authors had to decode bit by bit the OBN

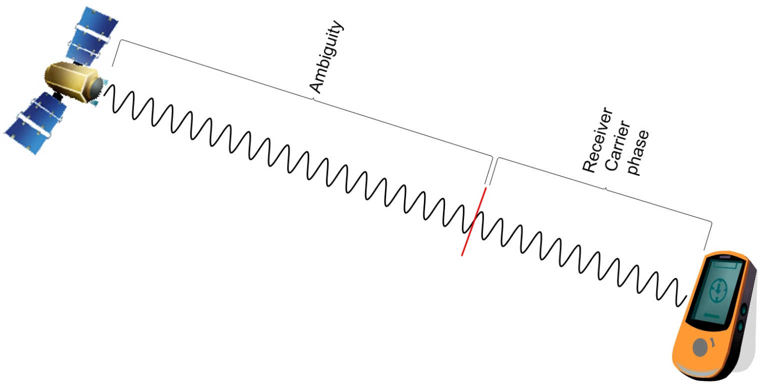

Figure 4.5: Ambiguity on GPS signal

message and to process the additional bits and obtain appropriate data for them. In the end it was discovered that the whole baseline vector field was missing from the description of the message. After composing the baseline data eventually the ambiguities were decoded properly.

Both methods provided experimental results that were different from each other. When adding the ambiguities to the measured carrier-phase the obtained results should have been the complete number of cycles in between the user and the satellite. Our results unfortunately were more than elusive, the composed result being far away (by order of thousands of kilometers) from the code pseudoranges. Thus, data computed on the carrier-phase didn't produce our expected results in ionospheric measurements.

We tried to find out the cause by going backwards to our previous assumptions. Taking the formula from (4.1) particularly on the Z-Xtreme was in the end the key to our problems. Knowing that the ionospheric delay is frequency dependent, L2 is more affected than the L1 frequency. The delay over L2 is greater than the one on L1 by the previously mentioned α coefficient from (4.3). Furthermore, the ionospheric delay on code measurements is a positive amount. As a result we have the following simple system (considering that the errors and biases are approximately the same on L1 and L2 when compared to the magnitude of the ionospheric delay):

$$\begin{cases} I_1 < I_2 (= \alpha I_1) \\ \rho_1 + \epsilon_1 = \rho_2 + \epsilon_2 \end{cases} \tag{4.12}$$

, which leads us to:

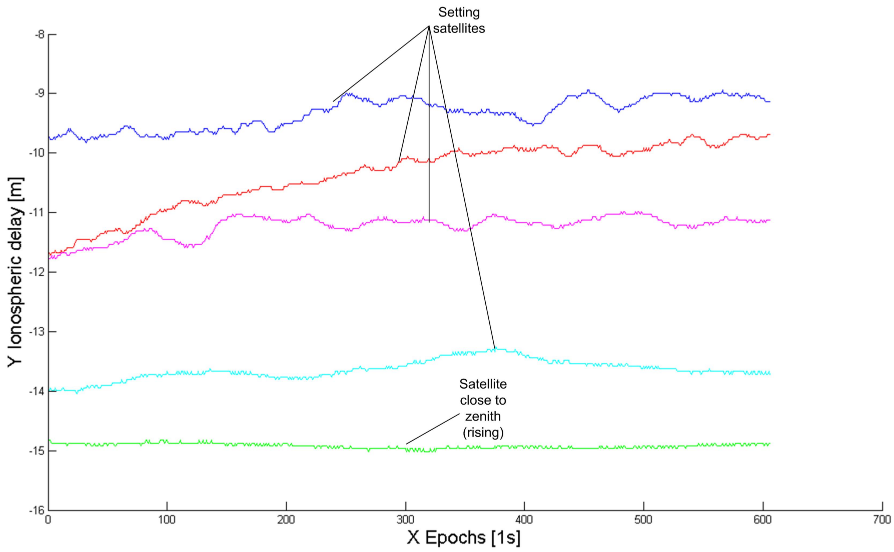

Figure 4.6: ZXtreme Pseudorange Ionosphere Estimation

$$\rho_1 + \epsilon_1 + I_1 = P1 < P2 = \rho_2 + \epsilon_2 + I_2 \tag{4.13}$$

In conclusion we should have the following relation between the raw ranges from the receiver:

$$P2 - P1 > 0$$

In both the real-time case and the downloaded rinex this inequation does not verify, in almost all the situations P1 being greater than P2. We have arrived to the conclusion that the receiver smooths the pseudorange before setting it to the presumed raw data, so our code measurements are corrupted by these operated corrections. Thus we were unable to obtain the true raw pseudoranges, fact that lead us to choose another set of receivers for the project, more precisely the Topcon Legacy-E.

We adapted all our code to the GRILL messages used by Topcon, in order to process all the data real-time on the new devices. Results in our code fit the previous conditions from (4.12) and (4.13). Although, in the Topcon case, due to lack of time, we appealed to the pseudorange computed ionospheric delay instead of using the carrier data.

4.5 DGPS Implementation

The sources of error in Table 2.1 can be dealt with in different ways. One approach is by estimating each of them individualy and at the end substract them from the pseudorange. This is what basically EGNOS does in the transmitted messages, forwarding corrections for all the error sources besides multipath and receiver noise. This is accomplished through several well spaced stations with precisely known positions that continously gather and observe GNSS data [ESA, 2006]. All these observation stations manage to cover corrections for an area that expands even beyond Europe's borders.

On a smaller scale, one station can provide corrections for surrounding users on a radius of tens of kilometers. In this case forwarded data considers error sources as a whole and the aided information can be either added to the end user position (position-space DGPS) or to the pseudoranges of each satellite (range-space DGPS) before computing the final position. Position-space DGPS sends a correction for each of the ECEF axis and requires the same set of satellite to be used for both the base station and the roving user. This demand many times fails to be fulfilled since satellite view can easily be obstructed for either the main station or the rover. Thus, we have employed for tests the range-space option and corrected only pseudoranges from common satellites while using the others unchanged, according to [Farrell & Barth, 1999].

For a better understanding we have provided the equations leading to the corrections. Considering the pseudorange measurements at the base station and rover we have:

$$\begin{aligned}
\rho_B &= \sqrt{[(X - x_B)^2 + (Y - y_B)^2 + (Z - z_B)^2]} + c\Delta t_{r_B} + \\
&\quad + c\Delta t_{SV} + T + I + E + MP_B + \mu_0 \\
\rho_R &= \sqrt{[(X - x_R)^2 + (Y - y_R)^2 + (Z - z_R)^2]} + c\Delta t_{r_R} + \\
&\quad + c\Delta t_{SV} + T + I + E + MP_R + \mu_0
\end{aligned} \tag{4.14}$$

where X,Y,Z are the ECEF coordinates of the satellite with subscripts R and B denoting the rover and base station terms, $c\Delta t_r$ is receiver error, $c\Delta t_{SV}$ is satellite clock error, *T, I* are tropospheric and ionospheric delays, E is ephemeris error, *MP* is multipath and μ is receiver thermal noise.

Further on, by using the known position of the base station, we can deduct the real pseudorange $\hat{\rho}$. Substracting it from ρ_B leaves us with the correction data, which is still corrupted by the overall uncommon error ξ (comprises *MP*

, μ, and $c\Delta t_r$), but the general performance is improved even so. Eventually we will have the factor Δ:

$$\Delta = -[c\Delta t_{SV} + E + T + I] + \xi \tag{4.15}$$

which is to be added to the rover's pseudoranges.

Range-space DGPS results are compared with the outcome of EGNOS corrected positions, ionospheric corrected positions, and finally ionosphere aided EGNOS, which are broadly presented in Chapter 5.

The last system mentioned, ionosphere aided EGNOS, employs changes in the ionospheric measurements generated by EGNOS by replacing them with the ones of the base station's dual frequency estimation. This way ξ from 4.15 is eliminated, providing an improved DGPS assistance from the point of view of additional errors if we look from the DGPS angle, while from the EGNOS angle we get an improved ionospheric delay estimation. We can see the comparison of the results in Figure 4.7. In this situation the following data was downloaded from the Topcon receiver: raw pseudoranges, time of measurement, and ephemeris. We have computed the raw positions, which are noted with white dots on the roof of DGC. Adding ionosphere estimation translates the results towards the real positions (yellow), while by using the DGPS correction we center the results around the actual antenna (green).

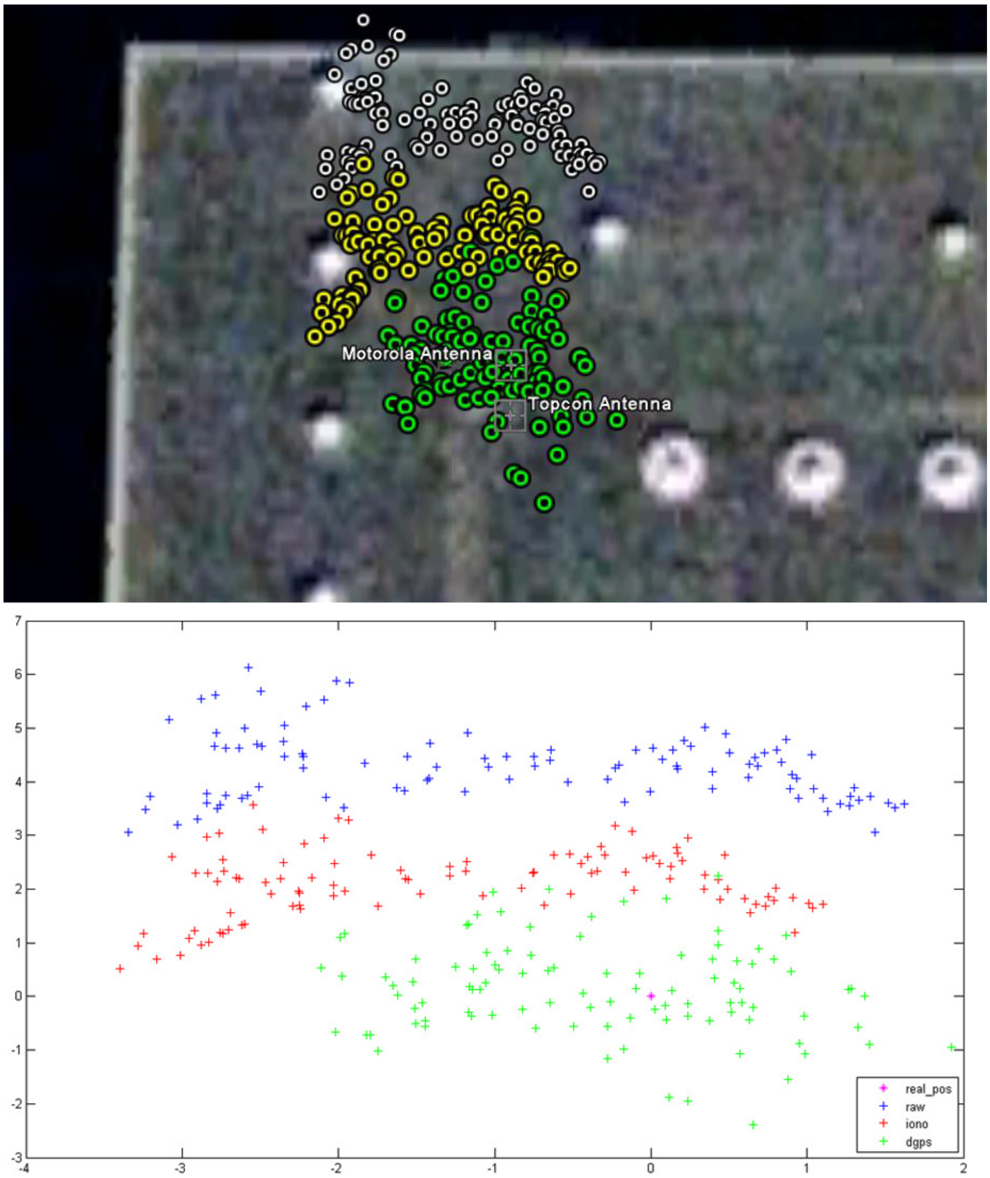

Figure 4.7: Raw, Ionospheric, and DGPS corrected positions

Chapter 5

Measurements and Results

5.1 EGNOS Aided Measurements

The following section will present the measurements and results obtained using the EGNOS enabled, DG14 receiver. There are several data sets that will be presented. The tests were done in the AAU campus, on the roof of the Danish GPS Center.

The computations have been done by post processing data acquired in 2005, and in 2010, in several days, over a time interval from 5 up to 24 hours. We chose a wide time interval in order to see how good EGNOS works in all conditions (day/night) and also to check the increase in ionospheric delay from night to day. The fact that the two moments when data was acquired are 5 years apart can also act as an evaluation of the way the GPS and EGNOS systems have evolved since then in terms of accuracy, especially for the latter one.

Later in the section we will also show results obtained in real-time on a time span of two and a half hours. The data was acquired using the same equipment, with the antenna from the roof of the Danish GPS Center, similar to the post processing scenario.

For all the measurements done, the results were plotted as variations on North and East, and also as deviations on North, East and Up from the known antenna position, with respect to time, for a better understanding of the errors involved in the computations.

The 2005 Data Set

The data set from 2005 has been acquired by Kostas Dragunas. Figures 5.1 and 5.2 show the results obtained in the case of the data acquired in 2005, over 5 hours. As it can be seen in 5.2, the smallest variation of the measurements is on the East axis, with most of the epochs having an error under 2 m. The biggest improvement is on the Up axis, from errors of almost 20 m to no more than 5 m.

As it can be seen from both Figures, the EGNOS corrections without the slow correction term bring very little improvement in accuracy, if any, and seem to act as a translation of the raw positions down on the North axis with a couple of meters, without improving the actual accuracy.

The results obtained are shown in the Table 5.1 below:

Position	Mean East [m]	Mean North [m]	Range East [m]	Range North [m]
Raw	0.069	1.775	5.365	9.099
EGNOS_{NS}[1]	-0.59	-1.74	5.31	11.1
EGNOS	0.083	-0.043	3.53	4.31

Table 5.1: Mean and Range for the 2005 Data Set

The results obtained with EGNOS on this data sample seem to be compliant to the standards, 1.8 m CEP and 3.8 m 95%, as stated in [Thales, 2008]. The errors are under 2 m, with noticeable improvements on the Up axis.

The following plots, Figures 5.3 to 5.10 show the fast and ionospheric corrections for some of the satellites involved in the computations for the data set from 2005, therefore the values show a 5 hour interval.

The ionopheric plots show also the elevation angles of the satellites. As expected the corrections decrease with the increase of the elevation. The range of the ionospheric corrections is between 2 to 8 meters, while the fast corrections vary from about -3 to 3 meters.

[1] EGNOS_{NS} refers to EGNOS corrections that exclude the slow correction term

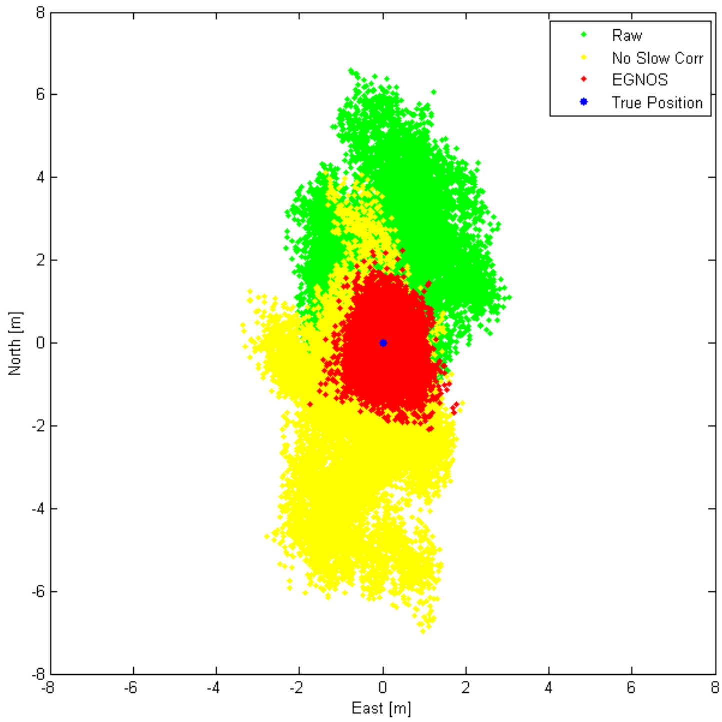

Figure 5.1: Position Accuracy for 2005 Data Set

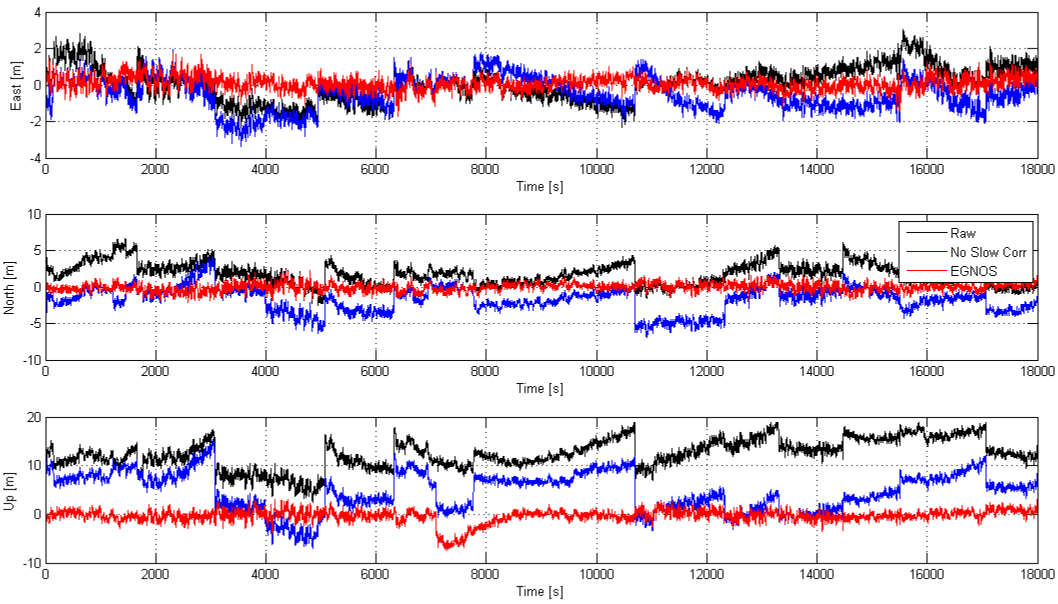

Figure 5.2: Position Accuracy with Respect to Time for 2005 Data Set

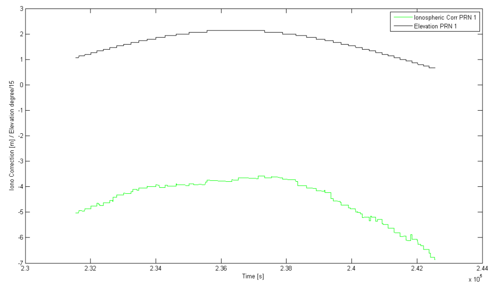

Figure 5.3: Iono Correction PRN 1

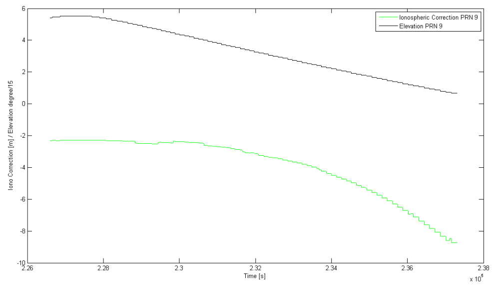

Figure 5.7: Iono Correction PRN 9

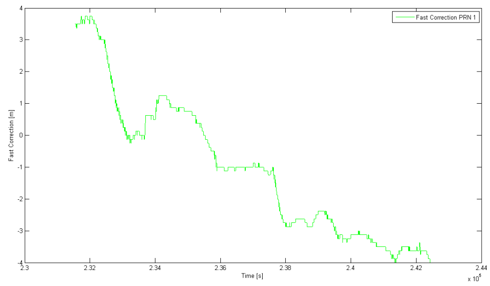

Figure 5.4: Fast Correction PRN 1

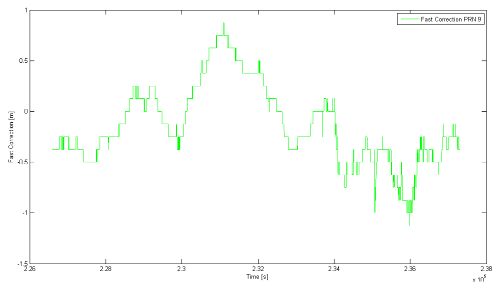

Figure 5.8: Fast Correction PRN 9

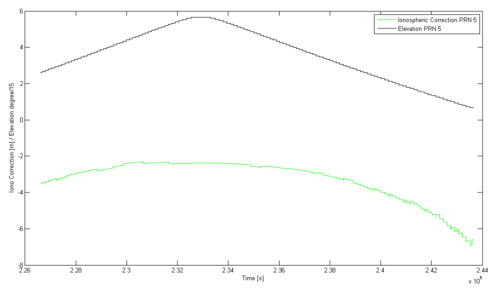

Figure 5.5: Iono Correction PRN 5

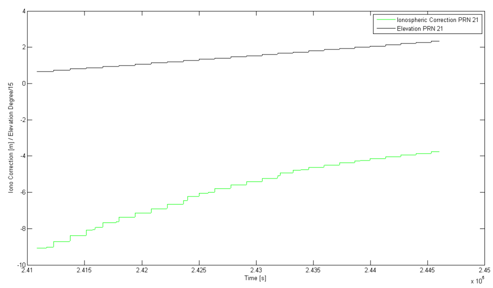

Figure 5.9: Iono Correction PRN 21

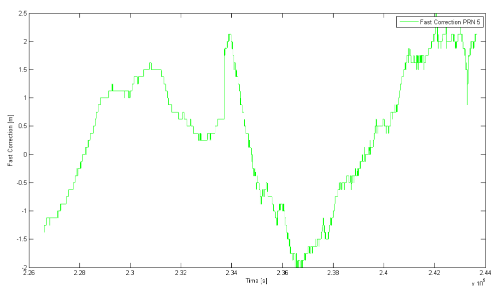

Figure 5.6: Fast Correction PRN 5

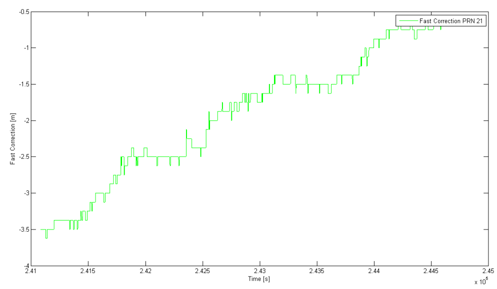

Figure 5.10: Fast Correction PRN 21

The 2010 Data Set

Samples were taken along several days, but the report will show only one of the data sets due to the fact that all days reveal similar results with minor variations from each other.

The particular sample that is analyzed here was taken in April 2010, over 24 hours. The results are plotted in Figures 5.11 and 5.12.

The results obtained in Table 5.2 show that even though EGNOS doesn't provide a better mean than the raw data, the improvement relies in lowering the absolute limits in North and East of the position range.

Position	Mean East [m]	Mean North [m]	Range East [m]	Range North [m]
Raw	0.043	0.54	6.57	14.56
EGNOS	-0.16	0.56	4.97	8.98

Table 5.2: Mean and Range for the 2010 Data Set. 10 ° Elevation Mask

Position Statistics	CEP(50%) [m]	R95(95%) [m]	CEP98(98%) [m]
Raw	1.407	3.447	4.226
EGNOS	0.815	1.810	2.154

Table 5.3: Statistics for the 2010 Data Set

Table 5.3 shows the error probability statistics obtained in the case of EGNOS aided measurements. If compared to the specifications of the DG14 receiver, one can see that the results obtained are within the limits specified by the producer.

The accuracy obtained in this case in not as pleasing as in the sample taken in 2005. The plots show more spikes, especially on the North axis, which seems to be more affected by errors than in the 2005 case. The first spike appears at the 6620th epoch, and lasts for about 15 epoch, increasing the error of the EGNOS corrected measurements from 1 m to 5 m, and going back to 1 m afterwards. Figure 5.13 shows the plot zoomed on that particular time interval.

Investigations led us to the conclusion that at least one of the reasons for the increase in error is a rising satellite. At epoch 6621, satellite 22 appears, and that is where things start to get worse. One way of eliminating this problem would be by using a higher elevation mask (the current one is 10°). For the same scenario, using a 15° mask we get the results shown in Figure

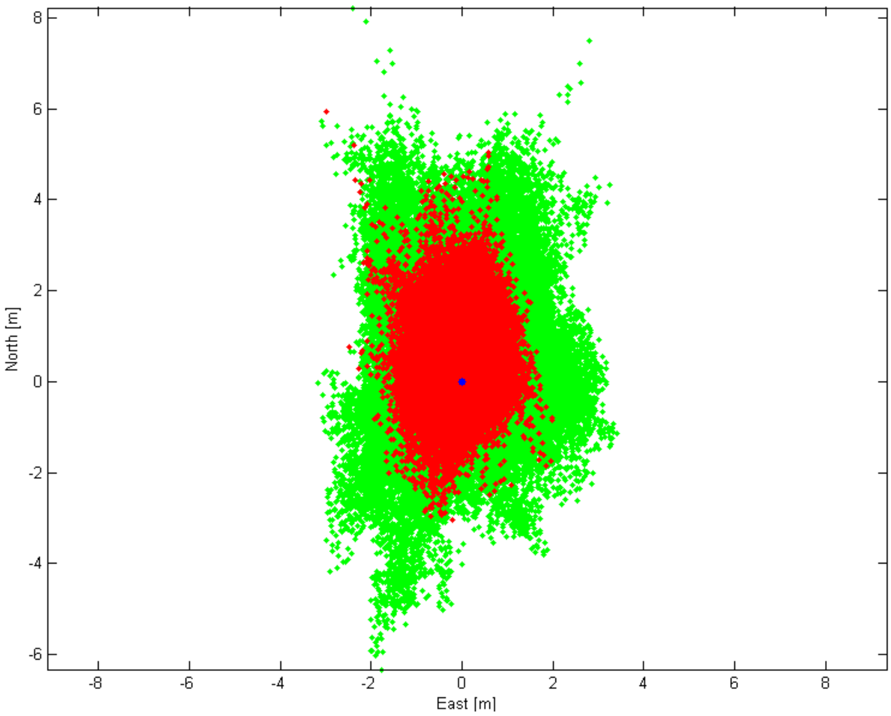

Figure 5.11: Position Accuracy for 2010 Data Set

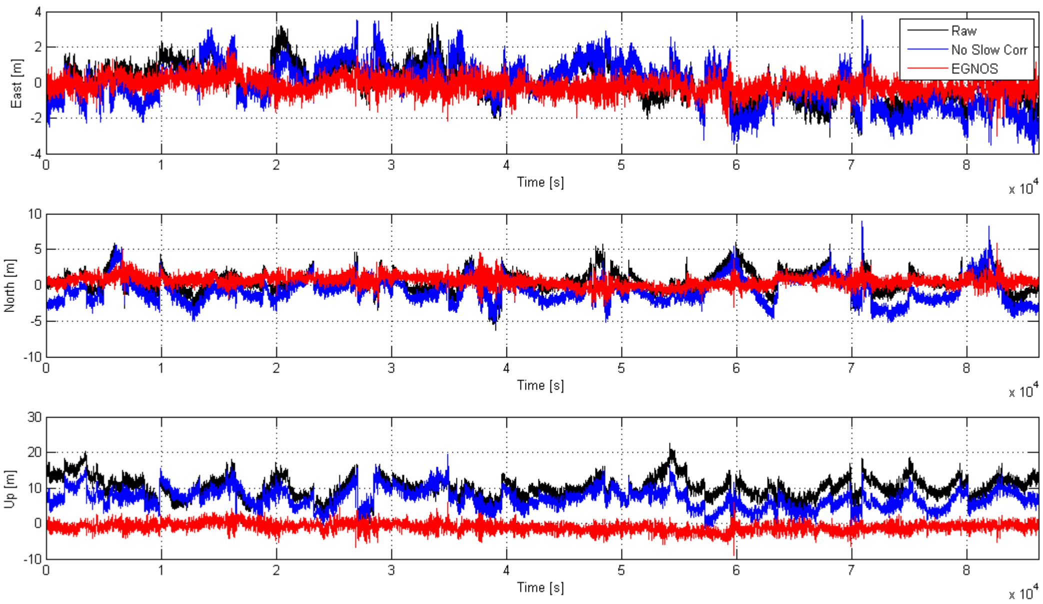

Figure 5.12: Position Accuracy with Respect to Time for 2010 Data Set

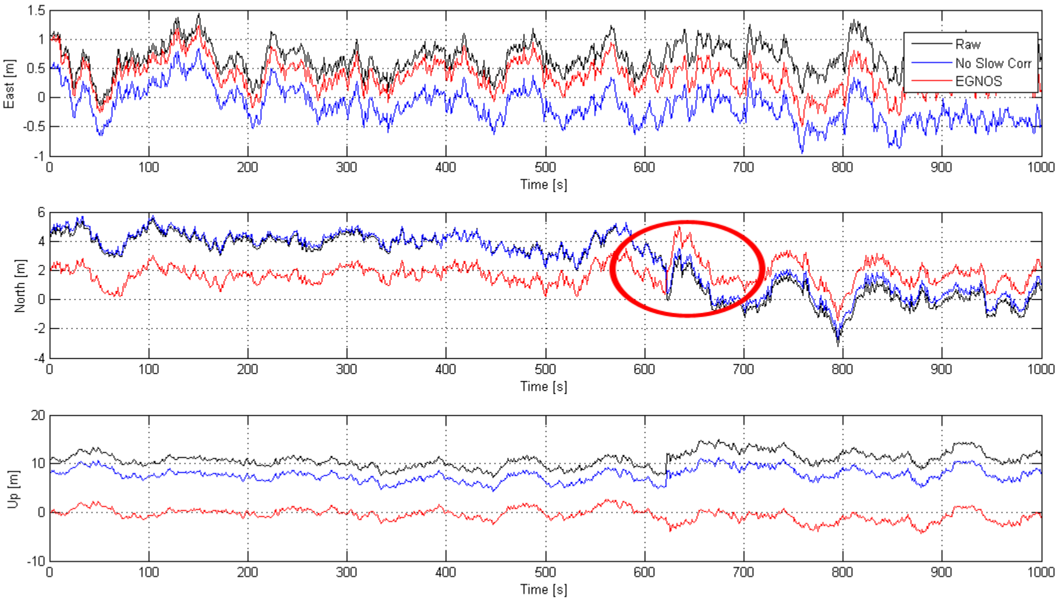

Figure 5.13: Plot zoomed on the 6000-7000 seconds interval 10 degrees

5.14. As it can be seen the error drops below 3 meters. So by using a higher elevation mask we obtained an improvement by at least 2 meters.

This modification can be detrimental to the accuracy obtained on a longer

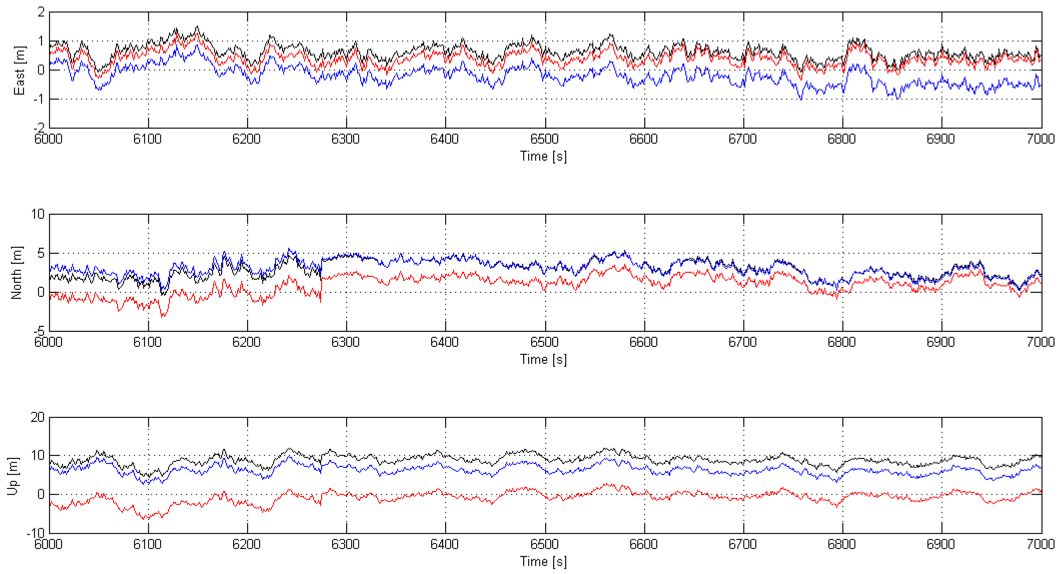

Figure 5.14: Plot zoomed on the 6000-7000 seconds interval 15 degrees

time interval, due to the fact that it reduces the number of satellites used in the computation, and possibly leading to situation when there will be less then 4 satellites available for computations. The effects of using a different elevation mask will be discussed in greater detail in 5.1.1

Figure 5.15 shows the evolution of the ionospheric corrections over 24 hours with respect to the elevation angle of the satellite PRN 2 for the 2010 data set. The data started to be acquired at 17:18 (Local Time) and the satellite first rises about 5 hours later between the time interval [22:31 − 02:23], while the second time it appears between [12:23 − 16:22].

This plot intends to show the comparison between the variation of ionospheric corrections during day and night sessions. Both times the elevation of the satellite reaches about 45°, but what makes the two cases different is the magnitude of the ionospheric correction. If in the first case the ionospheric correction reaches 1.5 m, in the second case, which is during the day, the value will reach 6 m, which proves the fact that measurements are more susceptible to errors during day time.

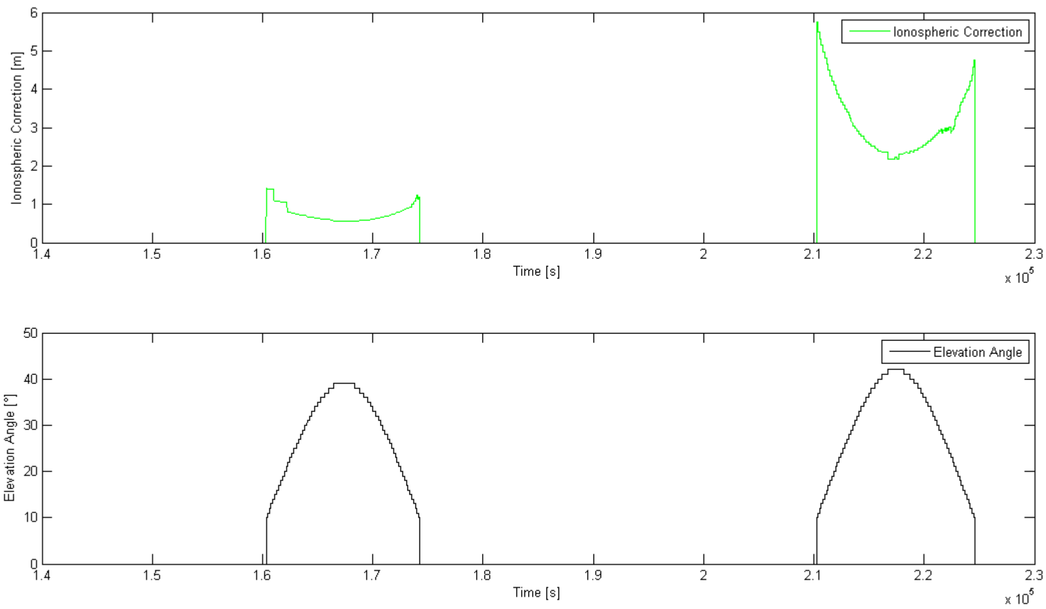

Figure 5.15: Ionospheric Corrections over 24 Hours for Satellite PNR 2

Real-Time Results

Figures 5.16 and 5.17 show the results obtained using the data acquired in real-time in 2.5 hours. Figure 5.16 also includes the positions obtained with Klobuchar model correction. Even though it's an obsolete method we wanted to see a comparison with the EGNOS method.

As expected we can see that using the Klobuchar method, the accuracy improves in comparison to the raw positions, but EGNOS provides better, more reliable results.

Tables 5.4 and 5.5 show the statistical and range results, respectively.

Position Statistics	CEP(50%) [m]	R95(95%) [m]	CEP98(98%) [m]
Raw	1.794	2.750	2.984
Klobuchar	1.246	2.126	2.345
EGNOS	0.935	1.739	2.007

Table 5.4: Statistics for EGNOS in Real-time

Position	Mean East [m]	Mean North [m]	Range East [m]	Range North [m]
Raw	0.043	0.54	6.57	14.56
EGNOS	-0.16	0.56	4.97	8.98

Table 5.5: Mean and Range for Real-time Data

5.1.1 Aspects that Influence the Results

Due to the complexity of the algorithm and of the computations involved, there were errors that infiltrated in the algorithm. The errors were eliminated, and in the following section we present the effect of leaving these errors unaddressed and some of the factors that could influence the accuracy of the results.

By having a closer look at the plots, one can see that the variation of the positions computed with EGNOS follows closely the pattern of the raw positions, but at a closer distance of the true position. In the plot from Figure 5.18, which is a preliminary version of the plot shown in Figure 5.2 with some errors unsolved yet, we can see that in most of the cases the EGNOS position doesn't show any anomalies, like spikes, or jumps, with just some minor exceptions. The right side of Figure 5.18 shows a zoomed in view on the spikes that appear at about 11000 seconds.

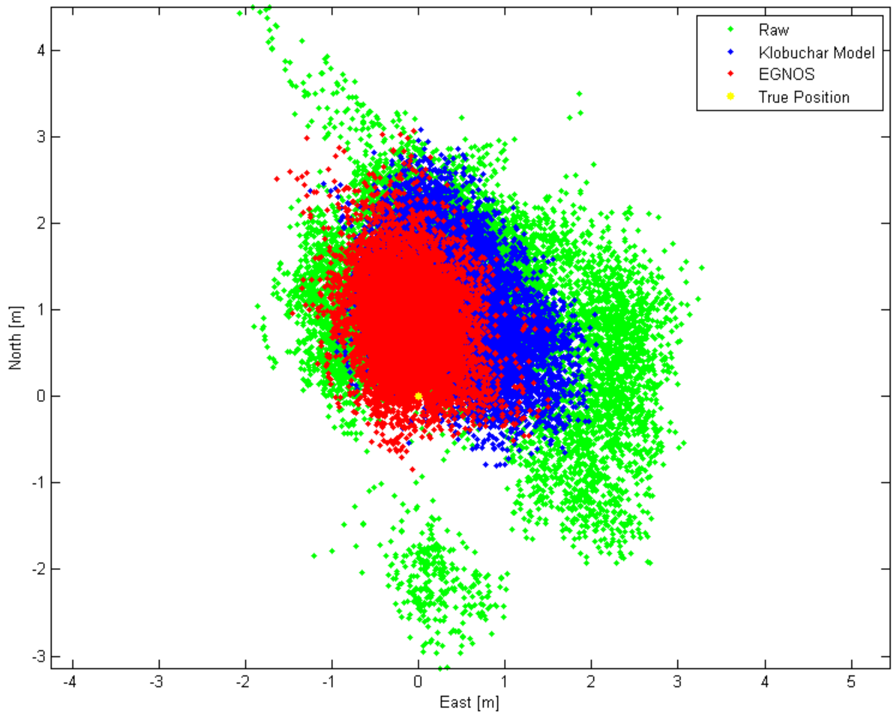

Figure 5.16: Real Time Results

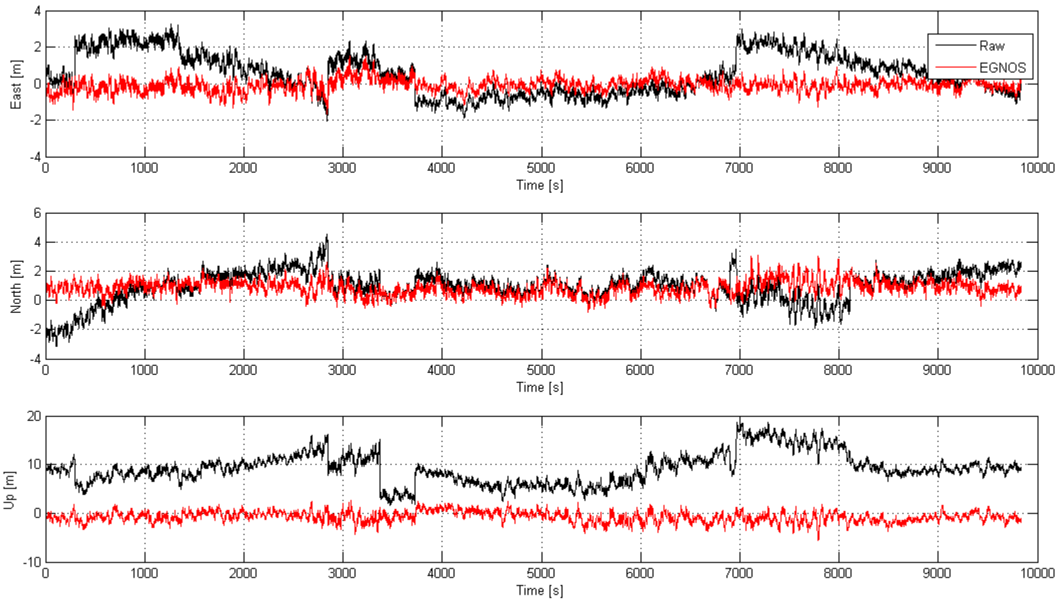

Figure 5.17: Real Time Results on Time Scale

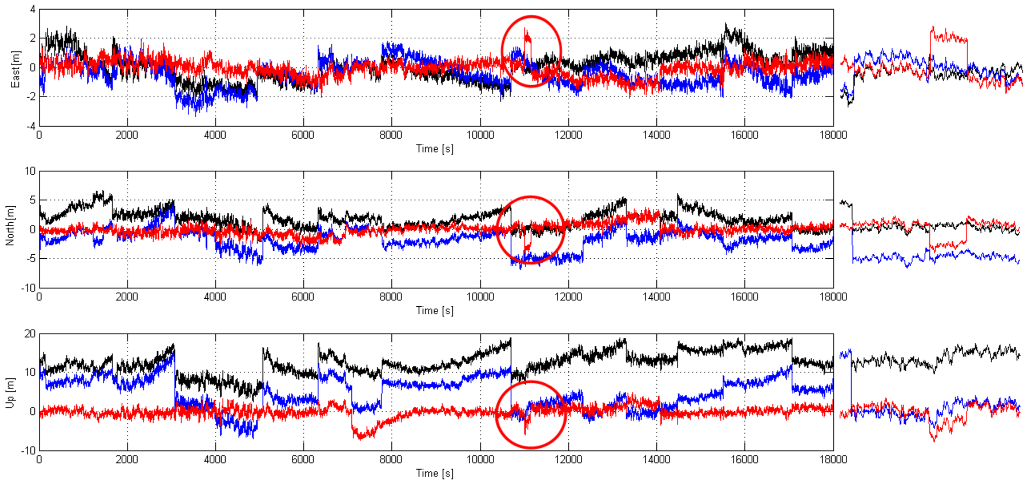

Figure 5.18: Jump is Position Accuracy

One of these spikes appears at around 11000 seconds. This can be caused by several things, like the rising or setting of a satellite or a bad correspondence between satellite ephemeris data and the slow corrections, a bad correspondence between the EGNOS corrections and the GPS data, or it could be the result of some problems that are not the topic of this report, as multipath or internal receiver noise.

By investigating in more detail that particular time we excluded the possibily of multipath due to the fact that the raw possition was not affected by a simmilar amount, and came to the conclusion that there were two factors that influenced this error.

First of all, as stated in [GPS, 1991], the ephemeris data are valid for 2 hours before and after the t_{oe}, or as in [Borre et al., 2007], the epoch t_{oe} is nominally at the center of the interval over which the ephemeris is useful, and selecting the ephemeris differently can lead to errors. Therefore using an ephemeris selection filter like ($t_{receiver} > (t_{oe} - 7200)$ & $t_{receiver} <= t_{oe}$), as used for the case shown in 5.18 leads to errors. This filter doesn't always lead to errors because most of the times the moment a new ephemeris set becomes valid is coincident to the t_{oe} of the last set, but there are those cases when there is a difference of a few seconds up to a few hundred seconds between t_{oe} and t_{ow} of the next epoch, which leads to the use of a wrong ephemeris set, and ultimately slow corrections will not be applied for those epochs.

The second factor relates to the specification of the EGNOS standard

[RTCA, 2001] which states that at least 3 previous ephemeris data sets need to be keeped in history, otherwise there can be mismatches between the slow corrections and the ephemeris data. In case the IODE of the current slow correction doesn't match the IODE of the current ephemeris data, a search is done in the prior ephemeris sets to find the one that has the same IODE as the slow correction.

After taking care of these problems, the plots present better results and we obtain the positions as shown in Figure 5.1 and 5.2.

Another aspect that raises questions appears in the plot from Figure 5.12, where we can see on the North axis a high jump for EGNOS from about 1 m to 5 m. A factor that influences the results is, as mentioned in 5.1, the appearance of new satellites on the sky. This affects the position only on a short term, and for the lowest elevations. One possible solution to this would be to use a higher elevation mask, but this doesn't necessarily solve the problem. Using a higher elevation mask eliminates a much higher number of satellite, decreasing the accuracy of the position, or in some case could also make it impossible to compute (if there are less then 4 satellite being used in computations).

As stated previously in 5.1, in particular cases, using a higher elevation mask proves to be successful, but tests on longer periods of time, up to 24 hours, demonstrate that the overall accuracy worsens.

Table 5.6 shows the results obtained using a 15° mask, in comparison to Table 5.2 for a 10° mask, and Table 5.7 for a 5° mask.

Position	Mean East [m]	Mean North [m]	Range East [m]	Range North [m]
Raw	0.059	0.61	6.6	13
EGNOS	-0.08	0.52	6.03	9.135

Table 5.6: Mean and Range for the 2010 Data Set. 15 ° Elevation Mask

Position	Mean East [m]	Mean North [m]	Range East [m]	Range North [m]
Raw	0.04	0.544	6.57	14.56
EGNOS	-0.16	0.56	4.97	8.98

Table 5.7: Mean and Range for the 2010 Data Set. 5 ° Elevation Mask

5.2 Ionospheric Estimation Procedure

As mentioned in Chapter 4 pseudoranges on L1 and L2 frequencies will be used for ionospheric delay estimation. Figure 4.2 shows that the delay estimation is highly affected by multipath and noise. Through experimental results we deducted that one way to deal with this problem is by averaging the results over a number of epochs from the moment of tracking and then use the phase measurements. The optimal number of epochs was from our experiments proved to be at around 600 epochs gathered at an interval of 1 s from each other. The decision for choosing an averaging over 600 seconds was taken considering the following reasons:

- a shorter interval may imply a greater variation of the ionospheric delay estimation of the epochs due to the fact that point intervals with unusual high/low peaks of multipath occur. These peak intervals are averaged after around 600 epochs, although the more epochs used the better
- averaging over a bigger number of epochs disconsiders the fast changes of the ionospheric delays that appear at the setting/rising of a satellite, when the amount of the GPS signal that travels through the satellite is much bigger than usually. Partly this rapid change is taken care of when setting the elevation mask for satellites at 10°. Still, the faster a ionospheric delay estimation is provided, the better.

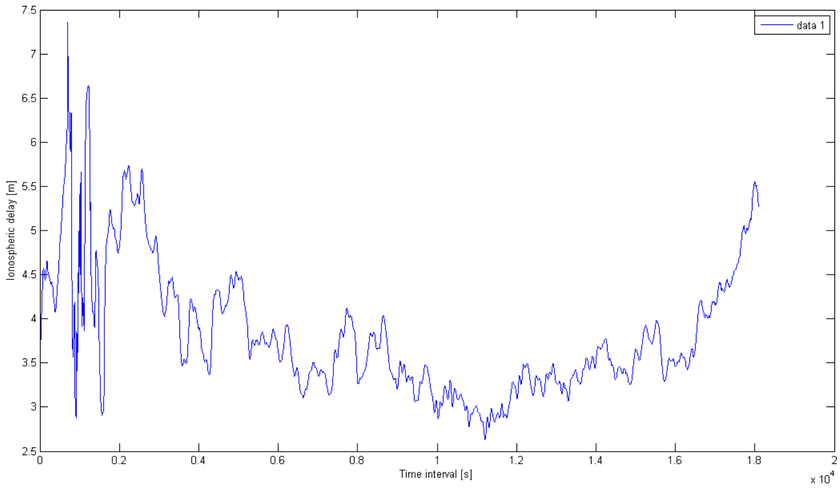

Figure 5.19: PRN 15 ionospheric delay using averaged pseudoranges

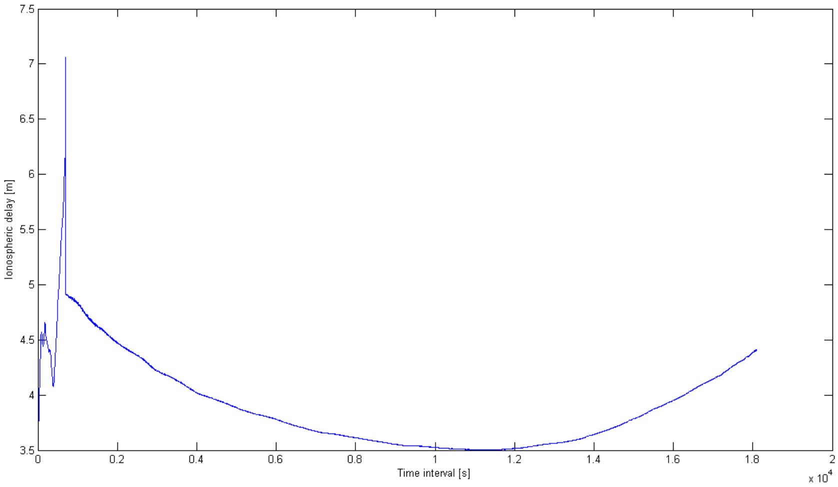

Figure 5.20: PRN 15 smoothed ionospheric delay after averaging interval

However, the real ionopsheric delay value changes little by little every epoch. To account for this change we weighted the values of the epochs when we averaged by using the following formula:

$$I_{Mean} = \frac{\sum_{i=1}^{600} I_i \sqrt{i}}{\sum_{i=1}^{600} \sqrt{i}} \tag{5.1}$$

As a result the last epochs will have more weight than the ones from the beginning.

After the averaging procedure the mean ionospheric delay over the optimum interval, 600 s, is used together with carrier phase data. We compute the offset at the last epoch between the phase ranges on L1 and L2 and afterwards add the offset to the mean value of the ionospheric delay. Subsequently, for each epoch, the difference between the phase ranges is added to this final sum. The offset has to be subtracted from the mean value in order to have a starting point for computing the variation of the ionosphere through carrier phases. The carrier smoothed ionospheric delay can be observed in figure 5.20, over a sample of 18000 epochs. The satellite used is PRN 15 starting from the moment of tracking, at 10°.

Figure 5.19 presents a version of ionospheric delay where averaging over short interval of 10 s was used. It can be easily noted in both Figure 5.19 and 5.20 that at the moment when the satellite is tracked multipath highly affects the result, so in the unsmoothed case we have after 600 epochs variations of

± 2 m from the real value. In comparison, the smoothing algorithm takes care of these variations and no peaks appear after operating with the carrier phases.

Additionally, cycle slips have to be taken into consideration when using carrier phase data. If the absolute value of the difference between two sequentially ionospheric values is greater than 0.19 m a cycle slip has occurred. The judgement behind this statement relies on the fact that the ionospheric variation and carrier phase multipath over 1 s are much lower than 19 cm, even when summed up. Since the wavelength of the shortest signal, L1, is of approximately 0.1903 m an absolute difference of a level greater than 0.19 m reveals a cycle slip on either L1 or L2(0.244 m). If such a situation materializes the offset is recomputed with the new values of carrier phases on L1 and L2 and the new value is used further on in the following computations of the ionospheric delay.

5.3 Results Of EGNOS Combined With Dual Frequency Data

The data processing porting into real-time allowed us to experiment and compare raw, DGPS, ionospheric aided EGNOS, and EGNOS samples taken in the same time. The ionospheric aided EGNOS has been computed following the procedure described in the previous section.

At the moment of experiments the raw position had a high accuracy as can be noted from Figure 5.21. Variations of the position on East-North axes were selected and the total distance from the real position was computed for each position on an approximately 2100 s interval at around 14 o'clock, with epochs taken every second.

The distances are sorted in ascending order for each of the measurements. We can note that the most accurate curve is the one of EGNOS, closely followed by DGPS. Strangely, the worst position is the one of EGNOS combined with ionospheric measurements from the dual frequency receiver. However, all the positions are 95% under 1.7 m, which is overall a pretty good position estimate. Further experiments need to be done over a longer time span to

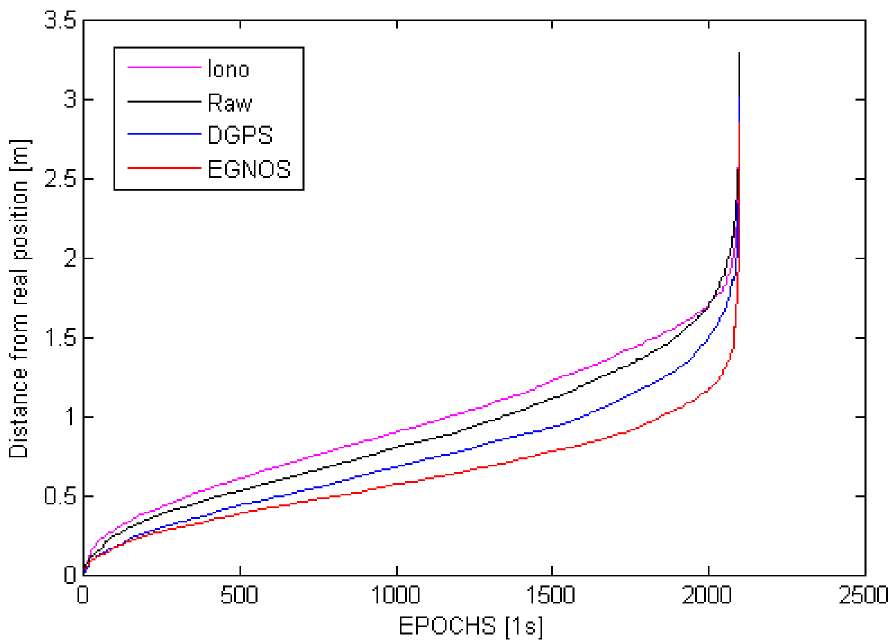

Figure 5.21: Raw, Iono, DGPS and EGNOS distance from origin

observe the behaviour over a 24 hour interval. At this moment the Topcon receiver generally has missing bits in the sent message (independent of the baud rate), which causes the program to crash.

In another sample taken about 22 hours later (12 hours), presented in Figure 5.22, EGNOS measurements combined with local ionospheric data show even worse positions, with only 3 m accuracy 95% of the time. The raw position still maintains a pretty good accuracy, this time at under 2.3 m 95% of the time, as for DGPS and EGNOS they now have reversed roles and with DGPS taking now the first place in means of accuracy, 1.15 m compared to 1.72 m of EGNOS, with the same 95% margin. Variations of the measurments are now greater, as the raw measurements are not so accurate as before.

For data redundancy we have made another experiment at 20 hours, same day with the previous sample. DGPS data maintains the same advantage over EGNOS, while Ionospheric aided EGNOS closes down on raw, although the hierarchy is overall the same as in the previous sample. Also this time we plot the data in East-North coordinates for another view upon the results. It is easy to notice that the DGPS positions are centered around the origin, and that the Ionospheric aided EGNOS has the most scatered positions.

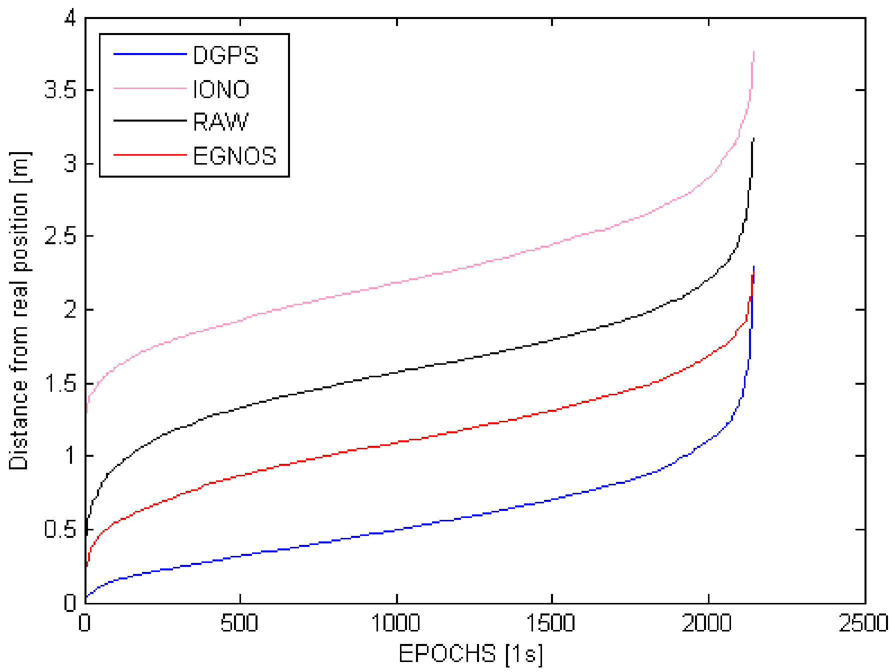

Figure 5.22: Distance from origin 12 hours

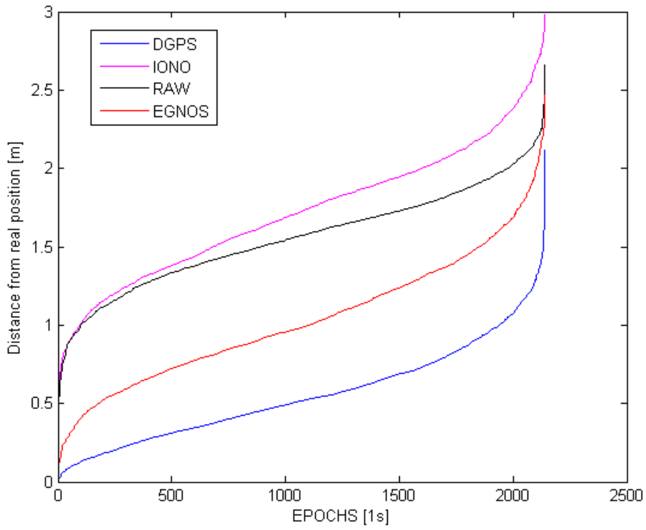

Figure 5.23: Distance from origin, 20 hours

As the ionospheric delay is decreased along the night due to low solar activitiy, the combined version obtains better results, but still not satisfactory when compared to DGPS and EGNOS alone.

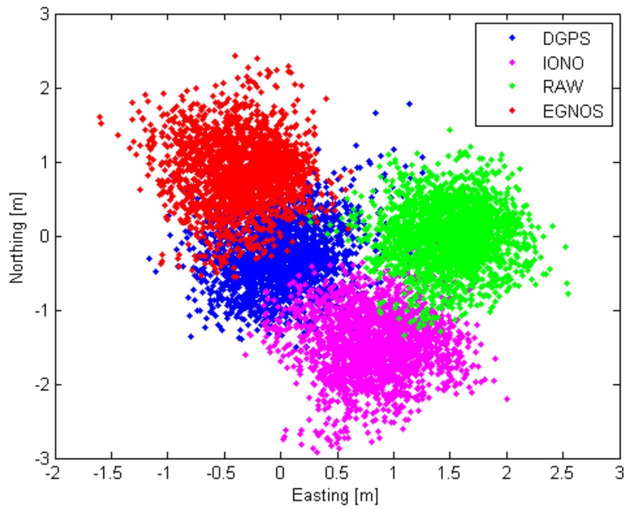

Figure 5.24: Variation from origin (m) in N-E, 20 hours

Chapter 6

Conclusions and Future Work

The measurements presented in this report are based on data acquired in several years which allows us to evaluate the evolution of the navigation system.

There are aspects that still need to be addressed that we may have overlooked due to the lack of experience or time, but the fact that all our experiments show similar results, within the limits specified by the producers or standards, allows us to conclude that the functionality of our implementation is correct.

Our initial target was to put the basis of a system that can accurately determine the position of cars (under 0.5 m), with a moderately priced system. Even though the lack of time made it impossible for us to reach this goal, the experiments done help us determine whether the systems we have been working on could lead, with further work, to reliable solutions for high accuracy car positioning applications.

We have implemented and tested 4 methods of improving the accuracy: EGNOS, DGPS, a combination between dual-frequency and EGNOS (replacing the ionospheric correction provided by EGNOS with the one computed by a dual-frequency receiver), and dual-frequency ionospheric aiding, but we have focused our attention on the first three. These have been chosen in the attempt of finding the best compromise between cost of implementation and precision.

From the cost of implementation point of view, the EGNOS solution is the most suitable one, due to the fact that EGNOS enabled receivers are not very expensive. The cost can be dropped even more by using a simple single frequency receiver combined with acquiring the EGNOS messages from the Internet via SISNeT. The other methods in subject require an additional dual-frequency receiver which increases the overall cost of the system.

In terms of position precision, Table 6.1 shows the results obtained with all systems. It can be easily noticed that DGPS holds the highest level in terms of accuracy, with EGNOS following at 0.5 m distance.

System	EGNOS	DGPS	Hybrid
R95[m]	1.7	1.15	2.5

Table 6.1: Evaluation of Precision

The Hybrid version (ionosphere aided EGNOS) doesn't provide satisfactory results, with a distance of over 1 m from the DGPS results. Our experiments, based on the study of the EGNOS system, lead us to the conclusion that the EGNOS corrections are interdependent, and the best results are obtained by placing together all EGNOS corrections as a whole.

In order to conclude, based on the results obtained so far, only the DGPS solution comes close to the standard we set for high accuracy car positioning applications. But this comes with the high cost of implementing a network of reference stations that would broadcast corrections to the user.

With further improvements, EGNOS could become a candidate for our requirements, having the advantage of a low price, the ease of use, and a wide coverage.

Future Work and Improvements

- improve the dual-frequency estimation of the ionospheric delay and optimize the hybrid algorithm
- manage to decode integer ambiguity resolution in order to obtain highly accurate ionospheric estimates from carrier phases
- uses better equipment/algorithm for multipath mitigation
- filter results for high peak elimination
- improve EGNOS corrections computation

Appendix A: Detailed Approach of the EGNOS Computations

The following chapter will provide more insight into the algorithm presented in Chapter 4 used in obtaining the ionospheric corrections. The attention is focused only on the way the ionospheric corrections were obtained, due to the complexity of the procedures and of the computations.

A.1 The Ionospheric Delay

A.1.1 Pierce Point Coordinates Determination

In order to determine the coordinates of the pierce point, which is defined to be the intersection of the line segment from the receiver to the satellite and an ellipsoid with constant height of 350 km above the WGS-84 ellipsoid [RTCA, 2001], we first need to determine the azimuth and elevation of the satellite and the coordinates of the user.

The azimuth and elevation are determined in `azmelev.m` by sending the command `$PASHQ,GSV,A` to the receiver, where A is the serial port on which the communication between the receiver and the computer is done. The command queries the receiver for the position of the satellites, their PRNs and the SNRs.

The response has the format:

```
$GPGSV,d1,d2,d3,d4,d5,d6,f7,d8,d9,d10,f11,d12,d13,d14,f15,d16,
d17,d18,f19*hh
```

with their meanings explained in Table A.3. One response line contains

Parameter	Description	Range
d1	Total number of GSV messages to be output	1 to 9
d2	Message number	1 to 9
d3	Total number of satellites in view	1 to 14
d4, d8, d12, d16	Satellite PRN number	1 to 32 for GPS 33 to 64 for SBAS
d5, d9 d13, d17	Elevation (degrees)	0° to 90°
d6, d10 d14, d18	Azimuth (degrees)	0° to 90°
f7, f11 f15, f19	SNR (dbHz)	30 to 60
*hh	Checksum	2-character hex

Table A.2: $GPGSV Message Format

data for up to four tracked satellites. The first three messages contain GPS satellite information and the second three messages contain GLONASS satellite information. So in the case of this program we are interested only in the first 3 lines, the next ones being ignored. All necessary information is saved in the structure sat and then the program proceeds to extract the user coordinates.

A similar procedure is followed for determining the coordinates of the user, where the command sent to the receiver is `$PASHQ,GLL,A`. The reply to this instruction is in the format:

```
$GPGLL,m1,c2,m3,c4,m5,c6,d7*hh
```

The values of the latitude and longitude are expressed as ddmm.mmmmm (d-degree, m-minute) so in order to get the correct values we have to divide by 100 both values, and in case the latitude sector is S or the longitude sector is 'W' the values have to be multiplied by '-1'.

Having all necessary information acquired, the next step is to compute the latitude and longitude of the Ionospheric Pierce Point (IPP).

First, the latitude is computed as:

$$\Phi_{pp} = \sin^{-1}\left(\sin\Phi_u \cos\Psi_{pp} + \cos\Phi_u \sin\Psi_{pp} \cos A\right)_{\text{radians}} \tag{A.1}$$

where Ψ_{pp} is the earth's central angle between the user position and the earth projection of the pierce point computed as:

$$\Psi_{pp} = \frac{\pi}{2} - E - \sin^{-1}\left(\frac{R_e}{R_e + h_I}\cos E\right)_{\text{radians}} \tag{A.2}$$

Parameter	Description	Range
m1	Latitude of the position fix	0° to 90°
c2	Latitude sector	N-North S-South
m3	Longitude of the position fix	0° to 180°
c4	Longitude sector	E-East W-West
m5	UTC time	0 to 235959.9
c6	Status of the position fix A-Valid V-Invalid	A,V
d7	Position system mode indicator A - Autonomous mode D - Differential Mode E - Estimated Mode M - Manual input mode S - Simulator Mode N - Data Not Valid	A, D, E, M, S or N
hh	Checksum	2-character hex

Table A.3: $GPGSV Message Format

where A is the azimuth angle of the satellite from the user's location (Φ_u,λ_u) computed previously and E is the elevation angle of the satellite from the user's location. R_e is the approximate radius of earth's ellipsoid (6378.1363 km) and h_I is the height of the maximum electron density (350 km).

Next the longitude is determined as follows:

If $\Phi_u > 70^\circ$, and $\tan\Psi_{pp}\cos A > \tan(\pi/2 - \Phi_u)$

or if $\Phi_u > -70^\circ$, and $-\tan\Psi_{pp}\cos A > \tan(\pi/2 - \Phi_u)$

$$\lambda_{pp} = \lambda_u + \pi - \sin^{-1}\left(\frac{\sin\Psi_{pp}\sin A}{\cos\Phi_{pp}}\right)_{\text{radians}} \tag{A.3}$$

Otherwise,

$$\lambda_{pp} = \lambda_u + \sin^{-1}\left(\frac{\sin\Psi_{pp}\sin A}{\cos\Phi_{pp}}\right)_{\text{radians}} \tag{A.4}$$

These values will be used later in the selection of the IGPs needed in the computation of the ionopheric delay.

For a better perspective upon the flow of the program, Figure 4.1 shows the position of the pierce points at the moment one of the tests was done, and what can be observed is that the pierce point locations are in the vicinity on the user position, with the furthest points corresponding to the the satellites that have a lower elevation angle, meaning that the signal will have a longer distance to cover from the satellite to the user. For an elevation angle of 10 °, the distance from the user to the signal piercing point cannot be larger than 2000 km, which is approximately 17 ° of latitude.

A.2 Interpreting Messages 18 and 26

One of the steps necessary in computing the ionospheric delay is to decode and interpret message type 18 along with message 26 to obtain the IGP mask that defines the IGP locations with available ionospheric information.

For this application only bands 3, 4, 5, 6, and 9 are required, which correspond to the area between longitude $-60°$ to $95°$ and latitude $0°$ to $85°$ covering Europe.

One of the first approaches to acquiring the necessary data was to investigate each message that comes from the receiver and in case it was a message type 18 the information was saved, otherwise the search continued, having all the other messages neglected until they were requested in the program. It proved much more efficient to investigate all messages that came from the receiver and save them if they were needed later in the program. This way a lot of time is saved (from about 400 seconds to approximately 180) due to the fact that the messages for needed for ionospheric computations are numerous and take a long time to be acquired, acting like a bottleneck in the flow of program.

After investigating [RTCA, 2001] and Figure 2.6 we can see that the IGP grid is organized as a matrix with 17 lines between 0 ° and N85 ° latitude and 32 columns between W60 ° and E95 ° longitude, therefore one way of saving the values found in the masks without too much effort is to have a 17×32 elements matrix , one cell for each IGP.

Figure A.1 maps the defined IGP points, extracted from the IGP masks of band 3, 4, 5, 6, and 9, to the map of the Earth. As it can be observed, the IGP locations cover a vast area, which will be significantly reduced by the lack of 'Monitored' IGPs as we will see further on, when the GIVD and GIVEI information will be taken into account.

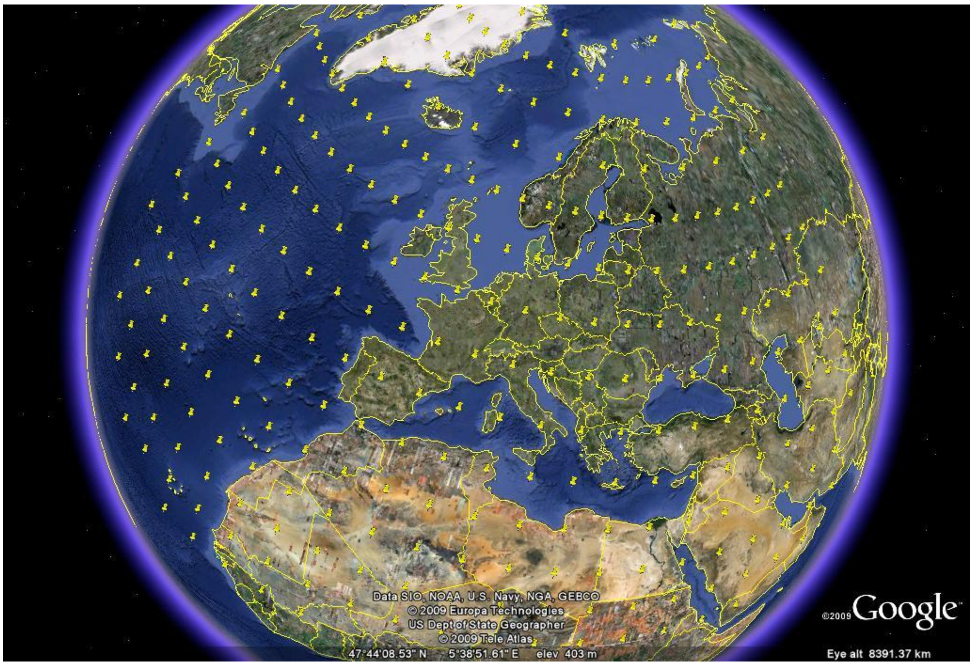

Figure A.1: The IGP Locations defined in the IGP mask

One problem arises from the fact that the IGPs are not evenly distributed worldwide. More precisely, at high latitudes the IGP locations are spaced more widely, from N60° to N80° the IGPs are spaced 10° longitude apart, and at 85° latitude they are spaced 30° longitude apart. As a result `igp_matrix` will have the elements set to 1 for any defined IGP location and set to 0 for undefined or inexistent IGP locations, being spaced 5° line and column wise.

Special attention has to be paid to the way `igp_matrix` was constructed. The area from N60° to N85° is covered by both band 9 and bands 3, 4, 5, and 6. The difference is that band 9 has a higher density of IGP points that are defined and that have available ionospheric information, but information for this band is not always available. Therefore, in case the mask for band 9 is also sent then the lines in the matrix corresponding to the area from N60° to N85° will be covered by the band 9 mask and the data from the other masks will be ignored.

Based on the content of the IGP mask matrix, two more matrices need to be created; one containing the GIVD values and one the GIVEI information. A similar procedure will be followed for extracting and formatting the data from messages type 26.

Each message 26 data block contains informations for a maximum of 15 IGPs. For each defined IGP there are two values associated. The GIVD, which is the vertical ionospheric delay of that particular IGP and the GIVEI which is an indicator of the accuracy provided by the correction.

At the moment the tests were done the information for band 3 was broadcasted in four blocks (49 IGPs), for band 4 in 5 blocks (70 IGPs), for band 5 in 6 blocks (76 IGPs), for 6 in 1 block (12 IGPs), and for band 9 in 6 blocks (80 IGPs). Due to the fact that messages 26 doesn't provide the user with the number of messages he should expect, the number of blocks being broadcasted has to be determined from `matrix_igp.m`, by dividing the number of 1s found in the masks by 15 (the maximum number of IGPs contained in one block), so if there are for example 76 values of 1 in the mask (as for band 5), there will be 6 blocks broadcasted (5 blocks $\times$ 15 IGP locations + 1 block with 1 location = 76).

Message 26 provides information for all IGPs that are defined in the IGP mask, keeping in mind that block id 0 corresponds to the first 15 IGPs defined in the masks, block id 1 provides information for the next 15 IGPs, and so on, as shows in Figure 2.7.

The last block that is broadcasted is usually shorter in size, due to the fact that the number of IGPs defined in the mask is, most of the times, not a multiple of 15.

The other matrix being constructed is the GIVEI matrix, which holds the information related to the integrity of the correction. Represented on 4 bit in the message, an indicator shows the accuracy of the correction sent, or if i is monitored or not. The values range from 0 to 15, with 15 being the indicator of a 'Not Monitored' IGP, and that IGP will further on be neglected from the computation of the ionospheric delay. In case four IGPs are selected for the interpolation and one of them is identified as 'Not Monitored' (GIVEI = 15), then three-point interpolation is used.

Figures A.1 and A.2 relate the IGP locations to their position on the Earth for a better visualization of the problem. In Figure A.1 all IGP locations extracted from the IGP masks that have the values set to 1 are plotted. It can be observed that the points cover the entire Europe and a great area of the Atlantic Ocean, but the situation changes in Figure A.2, where all IGPs that have the GIVEI set to 15 are excluded. To better see the difference between the defined and undefined IGPs the points in green show the IGPs defined in the masks and the orange points are the IGPs with valid information. This time they cover almost all Europe and very little of the Atlantic Ocean.

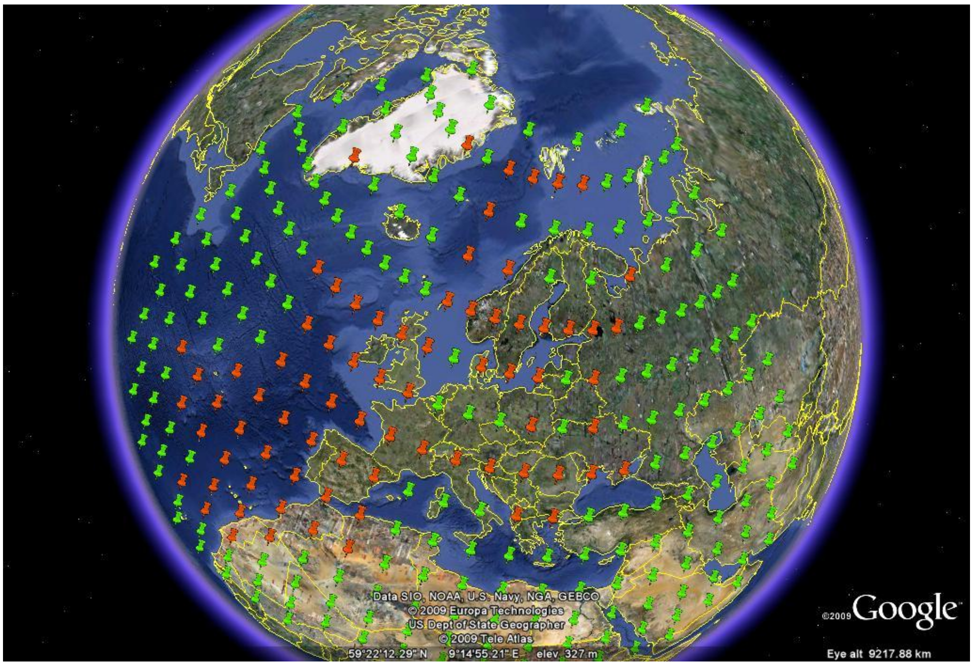

Figure A.2: The IGP Locations with valid GIVEI

The pictures were made using the data from Matlab in Google Earth with the help of Google Earth Toolbox for Matlab.

Each of these points that are monitored have a corresponding ionospheric delay that is transmitted to the user by means of Message Type 26. The values are used further on to determine the delay of the satellite signal by interpolating these known values, that form squares or triangles around the signal pierce point.

Figure A.3 shows a 3D plot of the ionospheric delays corresponding to the IGP locations, as they were when one of the experiments was made. The values are the delays of the grid points, at 350 km above the WGS-84 ellipsoid. The majority of the locations have a delay of up to two meters, extending to a maximum of approximately 8 meters in the area of 15° to 35° latitude and -40° to 0° longitude.

For a better view of things, Figure A.4 shows the contours of Figure A.3, and we observe the lack of 'Monitored' IGPs in the 9th band, more precisely in the area higher then 70° latitude. Also it can be observed, like in the previous figure, that the peak of the ionospheric delay is in the south of the Iberic Peninsula.

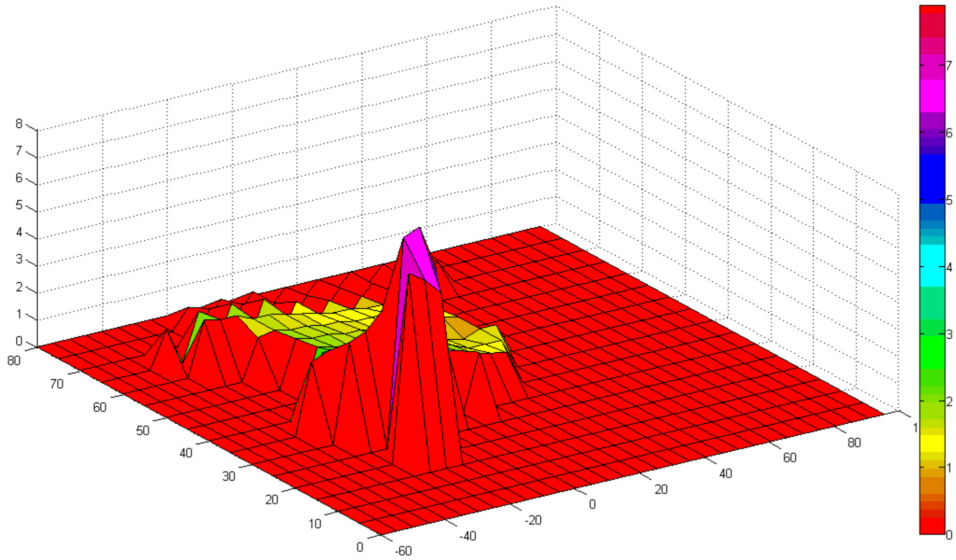

Figure A.3: The Ionospheric Delay of the Defined IGP Locations

A.2.1 Selecting the IGPs Involved in the Computation

After the coordinates of the pierce points are determined and the matrices containing the Message Type 18 and 26 data are constructed, the next step is to select the IGPs that will be used in the interpolation of the ionospheric correction.

All the following steps are presented in [RTCA, 2001] Appendix A with a graphic description in Appendix P.

1. For an IPP between N60° and S60°:

    a. if 4 IGPs that define a 5×5 degree cell around the IPP are set to 1 in the IGP mask, they are selected; else,

    b. if 3 IGPs that define a 5×5 degree triangle around the IPP are set to 1 in the IGP mask, they are selected; else,

    c. if 4 IGPs that define a 10×10 degree cell around the IPP are set to 1 in the IGP mask, they are selected; else,

    d. if 3 IGPs that define a 10×10 degree triangle around the IPP are set to 1 in the IGP mask, they are selected; else,

    e. no ionospheric correction available.

2. For an IPP between N60° and N75° or between S60° and S75°:

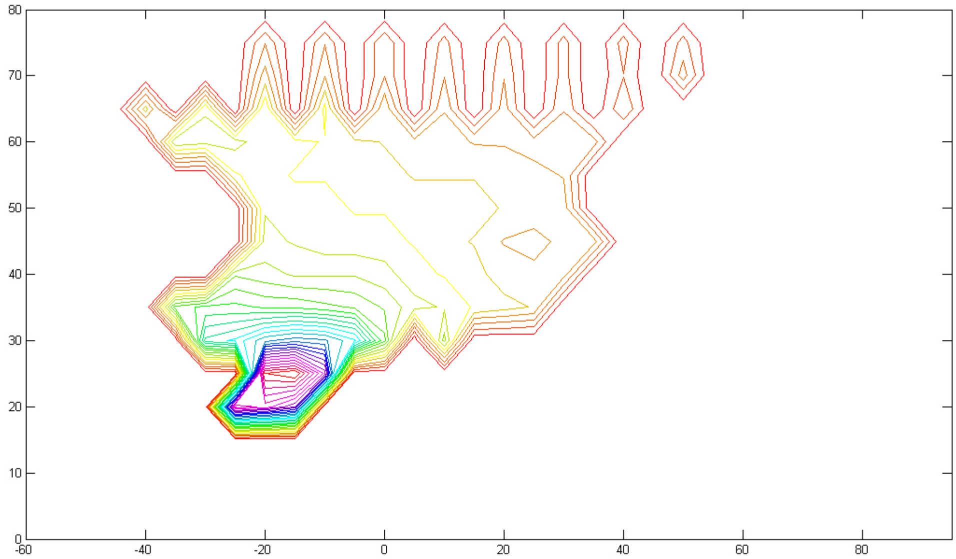

Figure A.4: The Contour of the IGP Ionospheric Delays

a. if 4 IGPs that define a 5° latitude by 10° longitude cell around the IPP are set to 1 in the IGP mask, they are selected; else,

b. if 3 IGPs that define a 5° latitude by 10° longitude triangle around the IPP are set to 1 in the IGP mask, they are selected; else,

c. if 4 IGPs that define a 10×10 degree cell around the IPP are set to 1 in the IGP mask, they are selected; else,

d. if 3 IGPs that define a 10×10 degree triangle around the IPP are set to 1 in the IGP mask, they are selected; else,

e. no ionospheric correction available.

3. For an IPP between N75° and N85° or between S75° and S85°:

a. if the 2 nearest IGPs at 75° and the 2 nearest IGPs at 85° are set to 1 in the IGP mask, a 10×10 degree cell is created by linearly interpolating between the IGPs at 85° to obtain virtual IGPs at longitudes equal to the longitudes of the IGPs at 75°; else,

b. no ionospheric correction available.

4. For an IPP north of N85°:

a. if the 4 IGPs at N85° latitude and longitudes of W180°, W90°, W0° and E90° are set to 1 in the IGP mask, they are selected; else,

b. no ionospheric correction available.

5. For an IPP south of S85°:

 a. if the 4 IGPs at S85° latitude and longitudes of W140°, W50°, E40° and E130° are set to 1 in the IGP mask, they are selected; else,

 b. no ionospheric correction available.

If any IGP of the 4 selected IGPs is identified as 'Do Not Use', the entire square must not be used, and an ionospheric correction is not available. If 4 IGPs are selected, and one of the four is identified as 'Not Monitored', then 3 point interpolation is used if the IPP is within the triangular region covered by three corrections that are provided.

A.2.2 Interpolating the GIVD Values

The final step in the flow of the algorithm after finding the coordinates of the IGP points is to read the data from the ionospheric delay values, and use it in an interpolation to determine the vertical delay of the IPP.

The following algorithm is extracted from [RTCA, 2001]. For the 4-point interpolation the vertical IPP delay $\tau_{vpp}(\Phi_{pp},\lambda_{pp})$ as a function of IPP latitude Φ_{pp} and longitude λ_{pp} is:

$$\tau_{vpp}(\Phi_{pp}, \lambda_{pp}) = \sum_{i=1}^{4}(x_{pp}, y_{pp})\tau_{vi} \tag{A.5}$$

where τ_{vi} are the broadcast grid point vertical delay values of the four IGPs and τ_{vpp} is the output value at the desired pierce point whose geographical coordinates are Φ_{pp}, λ_{pp}.

$$W_1 = x_{pp}y_{pp} \tag{A.6}$$
$$W_2 = (1 - x_{pp})y_{pp} \tag{A.7}$$
$$W_3 = (1 - x_{pp})(1 - y_{pp}) \tag{A.8}$$
$$W_4 = x_{pp}(1 - y_{pp}) \tag{A.9}$$
$$\Delta\lambda_{pp} = \lambda_{pp} - \lambda_1 \tag{A.10}$$
$$\Delta\Phi_{pp} = \Phi_{pp} - \Phi_1 \tag{A.11}$$

For IPP's betweem N85 ° and S85 °,

$$x_{pp} = \frac{\Delta\lambda_{pp}}{\lambda_2 - \lambda_1} \tag{A.12}$$

$$y_{pp} = \frac{\Delta\Phi_{pp}}{\Phi_2 - \Phi_1} \tag{A.13}$$

$$\begin{aligned} \text{where } \lambda_1 = &\ \text{longitude of IGPs west of the IPP} \\ \lambda_2 = &\ \text{longitude of IGPs east of the IPP} \\ \Phi_1 = &\ \text{latitude of IGPs south of IPP} \\ \Phi_2 = &\ \text{latitude of IGPs north of IPP} \end{aligned}$$

For IPP's north of N85 ° and south of S85 °,

$$y_{pp} = \frac{|\Phi_{pp}| - 85\,^\circ}{10\,^\circ} \tag{A.14}$$

$$x_{pp} = \frac{\lambda_{pp} - \lambda_3}{90^\circ}(1 - 2y_{pp}) + y_{pp} \tag{A.15}$$

$$\begin{aligned} \text{where } \lambda_1 = &\ \text{longitude of the second IGP to the east of the IPP} \\ \lambda_2 = &\ \text{longitude of the second IGP to the west of the IPP} \\ \lambda_3 = &\ \text{longitude of the closest IGP to the west of the IPP} \\ \lambda_4 = &\ \text{longitude of the closest IGP to the east of the IPP} \end{aligned}$$

Figure A.6 describes the way to select the variables for the 4 point interpolation. In the case of using a three point algorithm between 75 °S and 75 °N, the steps are:

$$\tau_{vpp}(\Phi_{pp}, \lambda_{pp}) = \sum_{i=1}^{3}(x_{pp}, y_{pp})\tau_{vi} \tag{A.16}$$

$$W_1 = y_{pp} \tag{A.17}$$

$$W_2 = 1 - x_{pp} - y_{pp} \tag{A.18}$$

$$W_3 = x_{pp} \tag{A.19}$$

At this point in the algorithm, preliminary results start to be obtained, which are indicators of how good the functionality of the program is. One of the test is done by replacing the coordinates of the pierce points with a grid of 1 ° latitude by 1 ° longitude and interpolate for all these values. Besides

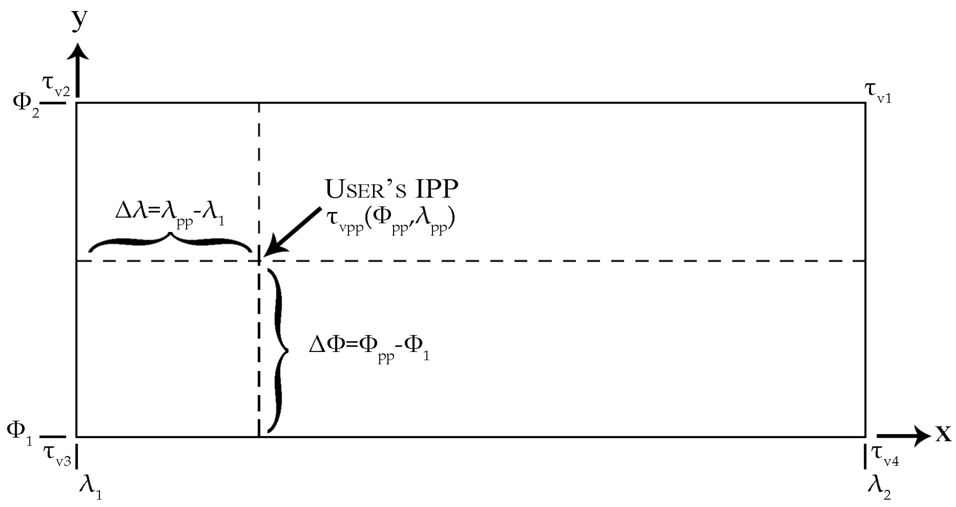

Figure A.5: Four Point Interpolation Algorithm Definition

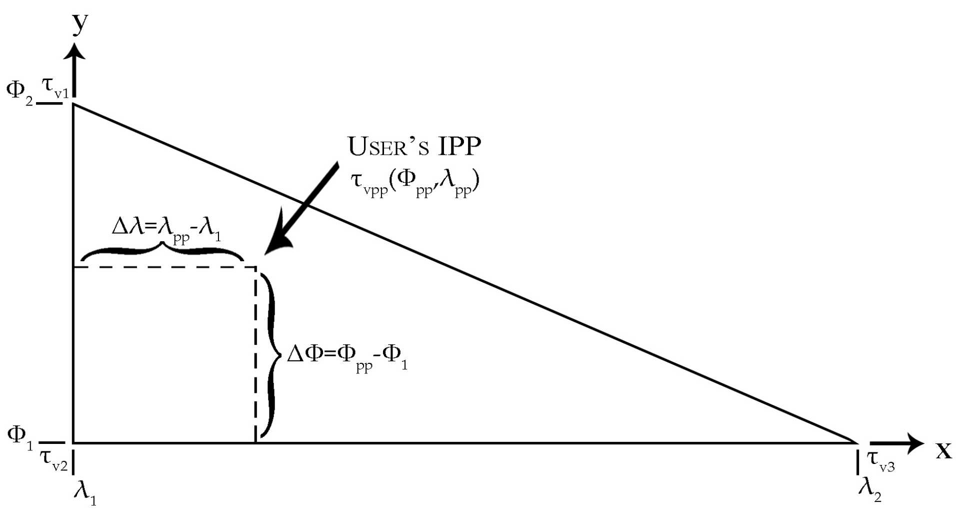

Figure A.6: Three Point Interpolation Algorithm Definition the fact that the program will be tested for a great amount of values, the result should not show any great leaps from one grid square to another.

The result of one test like this is shown in Figure A.7 obtained by interpolating on a grid spaced 1° apart.

One of the things that can be noticed is that there is a big peak of ionospheric delay at about 20° latitude with -15° longitude, but this is explained by checking Figure A.3, where the delay values of the IGP locations are plotted. It can be noticed that delay values reach their peak in the same area, which can explain things.

The second thing that can be noticed from this 3D plot is that besides the peak previously mentioned, the other values vary smoothly, from one area to another with no big leaps, which is an indicator that the corrections are realistic.

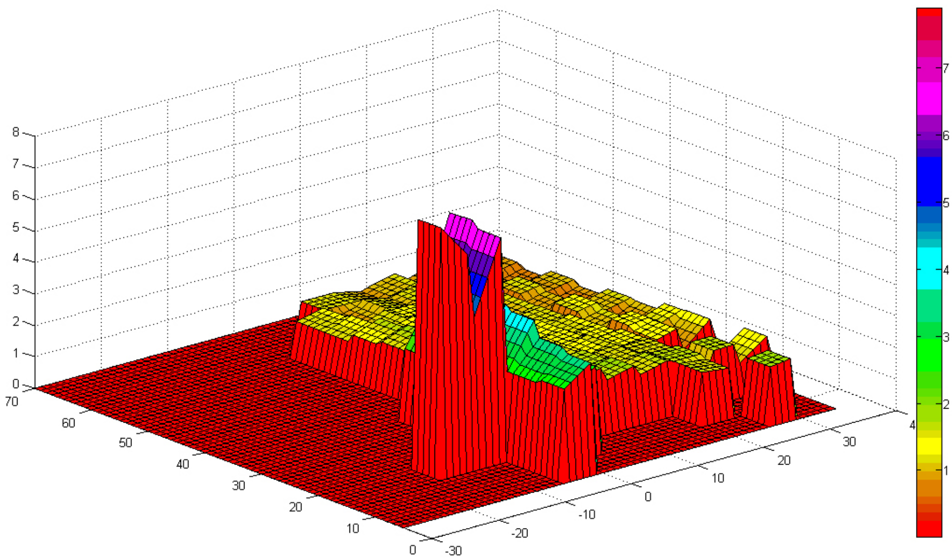

Figure A.7: One degree Interpolation Delays

A.2.3 The Final Step

The values obtained from the interpolation represent the vertical delays of the pierce point locations, which means that in order to obtain the real delay of the satellite signal, the slant ionospheric delay, the values have to be multiplied by an obliquity factor, and the results will be added to the pseudorange measurements.

$$IC_i = -F_{pp}\tau_{vpp}(\lambda_{pp}, \Phi_{pp}) \tag{A.20}$$

where τ_{vpp} is the interpolated vertical delay at the user-satellite IPP obtained at the previous step, and F_{pp} is obtained with the formula:

$$F_{pp} = \left[1 - \left(\frac{R_e cosE}{R_e + h_I}\right)^2\right]^{-\frac{1}{2}} \tag{A.21}$$

Appendix B: Work Flow

This chapter describes the flow of work of the project and the milestones that were accomplished along the work. Normally the project should have taken around 5 months, during the interval 01.09.2009 – 31.01.2010. The problems encountered along and the incorrect estimation of EGNOS corrections implementation have forced us to take another three months, with the project interval now 01.09.2009 – 07.05.2010.

The project started initially with a different base station receiver (Ashtech's ZXtreme) than the one used in the end (Topcon's Legacy E), due to which documentation had to be done in both receivers communication processes. The work flow is presented in Figure 8.

Working in a team of two persons allows flexibility in time management, so no major problems were encountered in the collaboration.

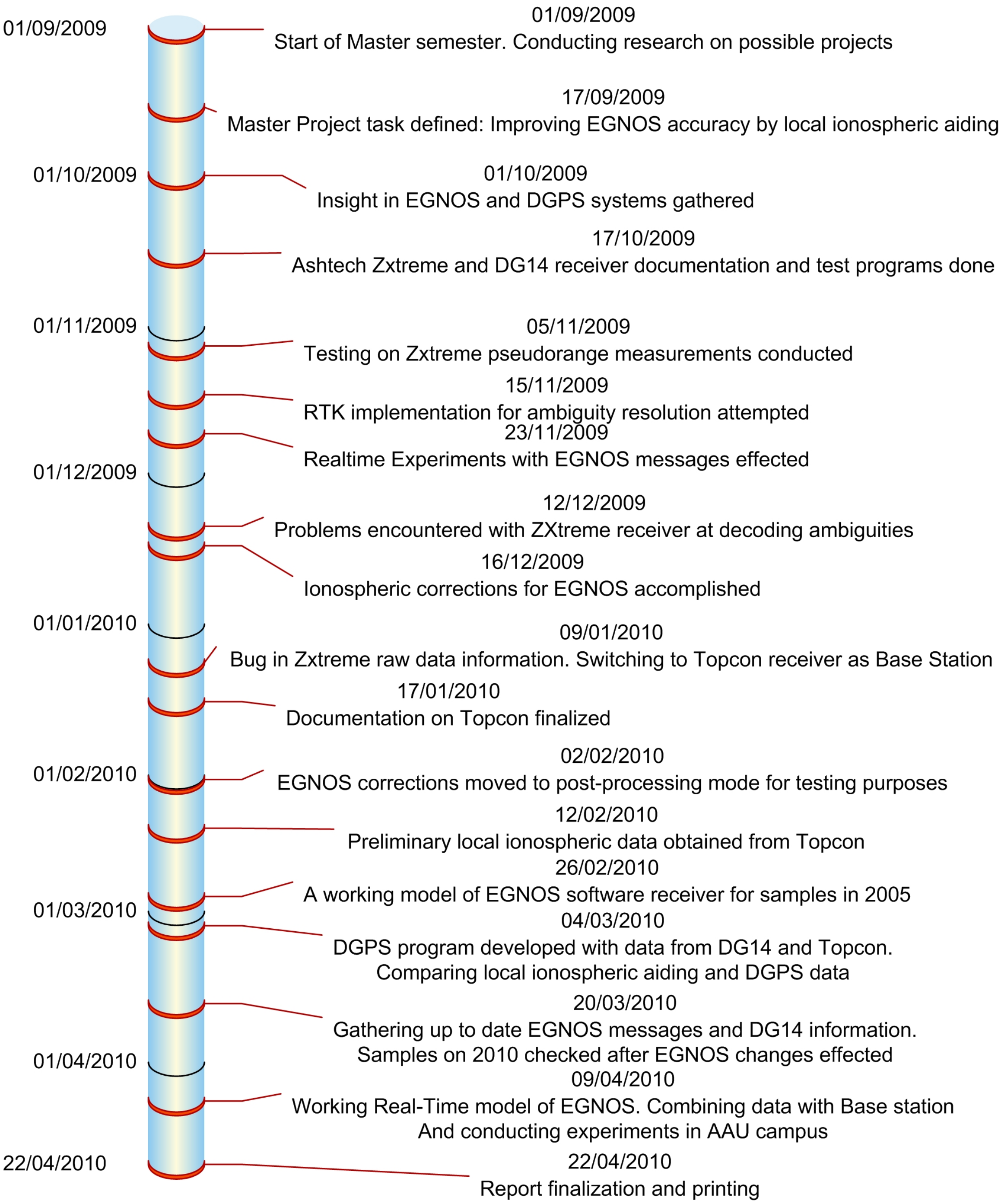

Figure 8: Work Flow

Appendix C: The Source Code

C.1 The Main Function

```
function compute_position()

port='COM5';
comm = serial(port);
set(comm,'BaudRate',115200,'DataBits',8);
set(comm,{'InputBufferSize','Terminator'},{50000,'CR/LF'});
set(comm,'FlowControl','hardware');
fopen(comm);

fprintf(comm,'$PASHS,RAW,MCA,A,OFF');
fprintf(comm,'$PASHS,SBA,DAT,A,OFF');
fprintf(comm,'$PASHS,RAW,SNV,A,OFF');
fprintf(comm,'$PASHS,RAW,XYZ,A,OFF');
fprintf(comm,'$PASHQ,ION');
[alpha,beta]=read_ionoparam(comm);
fprintf(comm,'$PASHS,RAW,MCA,A,ON,1');
fprintf(comm,'$PASHS,RAW,XYZ,A,ON,1');
fprintf(comm,'$PASHS,SBA,DAT,A,ON');
fprintf(comm,'$PASHS,RAW,SNV,A,ON,1');

while 1
    while 1
        [eph,epoch,msg,total_no_sats,t_gps,corrected]=feed_all(comm,1,eph,total_no_sats,t_gps);
        index_sat=[];
        for i=1:size(total_no_sats,2)
            if eph(1,total_no_sats(i)).aode(1)==-1
                index_sat=[index_sat,i];
            end
        end
        total_no_sats(index_sat)=[];
        if sum(sum(epoch.sat))==0 | isempty(sba)
            continue
        end
        update_msg=0;
        index=index+1;
        bin_value=hex2bin(msg);
        bin_message=bin_value(1:250);
        msg_type=bin2dec(num2str(bin_message(9:14)));

        switch msg_type
            case 1
                [corrected_sats,IODP_mask]=get_PRNmask(msg);
```

```
        status(msg_type).matrix=1;
        update_msg=1;
    case 0
        [data_fastcorr,IODP_msg2]=get_fastcorr(msg,data_fastcorr);
        status(msg_type).matrix=1;
        update_msg=1;
    case 2
        [data_fastcorr,IODP_msg2]=get_fastcorr(msg,data_fastcorr);
        status(msg_type).matrix=1;
        update_msg=1;
    case 3
        [data_fastcorr,IODP_msg3]=get_fastcorr(msg,data_fastcorr);
        status(msg_type).matrix=1;
        update_msg=1;
    case 4
        [data_fastcorr,IODP_msg4]=get_fastcorr(msg,data_fastcorr);
        status(msg_type).matrix=1;
        status(msg_type).matrix=1;
        update_msg=1;
    case 25
        [info_slowcorr,PRN_mask_no]=get_slowcorr(msg,info_slowcorr,PRN_mask_no);
        status(msg_type).matrix=1;
        update_msg=1;
    case 24
        [data_fastcorr,info_slowcorr,PRN_mask_no,IODP_msg24,status_closed24]=...
                get_mixedcorr(msg,data_fastcorr,info_slowcorr,PRN_mask_no);
            status(msg_type).matrix=1;
    case 18
        update_msg=1;
        [data_msg18,band_vec_18,IODI_mask,no_of_bands]=get_IGPmask(bands,msg,band_vec_18,data_msg18);
        band_vec_18=sort(band_vec_18);
        if isequal(band_vec_18,bands) | size(band_vec_18,2)==no_of_bands
            status(msg_type).matrix=1;
        end
    case 26
        update_msg=1;
        [data_msg26,vec,IODI_msg26]=get_delay_values(data_msg26,bands,msg,vec);
        band_vec_26=sort(band_vec_26);
        if status(18).matrix
            for i=band_vec_18
                if find(i==band_vec_18)
                    size1s(i)=data_msg18(i).size_ones;
                    no_blocks(i)=(size1s(i)-rem(size1s(i),15))/15+1;
                end
            end
            contor=0;
            for i=band_vec_18
                if no_blocks(i)~=0
                    for j=1:no_blocks(i)
                        if vec(i,j)==0
                            contor=contor+1;
                        end
                    end
                else
                    contor=1;
                end
                if contor==0
                    pass(i,1)=1;
                end
            end
            if pass(band_vec_18)==1
                status(msg_type).matrix=1;
```

```
                end
            end
        otherwise
            not_msg=1;
    end
    sats=intersect(find(epoch.sat(1,:)~=0), find(epoch.sat(3,:)>=10));
    sats=intersect(sats,total_no_sats);
    cont=0;
    for i=1:size(sats,2)
        if find(sats(i)==mask)
            cont=cont+1;
        end
    end
    if cont==size(sats,2) | status_closed24
        status(24).matrix=1;
                [common_sats,pos1,pos2]=intersect(sats,prn_mask);
    end
    if status(1).matrix && status(2).matrix && status(3).matrix && ...
            status(24).matrix && status(18).matrix && status(26).matrix
        first_computation=1;
        break
    end
    if (first_computation & update_msg) | (not_msg & first_computation)
        not_msg=0;
        break
    end
end

if first_computation
    vec_fast(1,1:13)=data_fastcorr(2).corr(:);
    vec_fast(1,14:26)=data_fastcorr(3).corr(:);
    vec_fast(1,27:32)=data_fastcorr(5).corr(:);
    vec_udrei(1,1:13)=data_fastcorr(2).udrei(:);
    vec_udrei(1,14:26)=data_fastcorr(3).udrei(:);
    vec_udrei(1,27:32)=data_fastcorr(5).udrei(:);

    fast_corrections=[];
    pp=1;
    for i=1:size(corrected_sats,2)
        if ~isempty(find(corrected_sats(i)==common_sats))
            k=find(corrected_sats(i)==common_sats);
            if vec_udrei(1,i)<14
                fast_corrections(pp,1)=vec_fast(1,i);
                pp=pp+1;
            else
                common_sats=setdiff(common_sats,common_sats(k));
            end

        end
    end

    data.sats=common_sats;
    data.azm=epoch.sat(2,common_sats);
    data.elv=epoch.sat(3,common_sats);
    data.time=epoch.time;
    data.bands=band_vec_18;
    iono_error=ionospheric_computation(epoch,data_msg18,data_msg26,data);

    g=g+1;
    for i=1:size(prn_mask,2)
        save_data(g).slow(1,corrected_sats(1,PRN_mask_no(i)))=info_slowcorr(PRN_mask_no(1,i)).delta_x;
        save_data(g).slow(2,corrected_sats(1,PRN_mask_no(i)))=info_slowcorr(PRN_mask_no(1,i)).delta_y;
```

```
    save_data(g).slow(3,corrected_sats(1,PRN_mask_no(i)))=info_slowcorr(PRN_mask_no(1,i)).delta_z;
    save_data(g).slow(4,corrected_sats(1,PRN_mask_no(i)))=info_slowcorr(PRN_mask_no(1,i)).delta_a;
    save_data(g).slow(5,corrected_sats(1,PRN_mask_no(i)))=info_slowcorr(PRN_mask_no(1,i)).IODE;
    save_data(g).slow(6,corrected_sats(1,PRN_mask_no(i)))=corrected_sats(1,PRN_mask_no(i));
end

prns=common_sats;
iono_corr=iono_error;
fast_corr=fast_corrections;
slow_corr=save_data(index_iteration).slow;
P=epoch.sat(4,prns)*299792458;
time=epoch.time;
previous_Eph=Eph_tot;
[efem,mask]=select_ephk(epoch,eph);

Eph_tot=set_eph(efem);
rr=1;
for kk=1:size(Eph_tot,2)
    if find(Eph_tot(1,kk)==prns)
        pos_sel=find(Eph_tot(1,kk)==prns);
        Eph(:,rr)=Eph_tot(:,kk);
        rr=rr+1;
    end
end
if ~isequal(previous_Eph,Eph_tot) & ~isempty(previous_Eph)
    Eph_last(:,previous_Eph(1,:),index_eph)=previous_Eph(:,:);
    index_eph=index_eph+1;
end

while 1
    contor_slow=0;
    for ii=1:size(prns,2)
        if ~isempty(find(slow_corr(6,ii)==Eph(1,:)))
            position=find(slow_corr(6,ii)==Eph(1,:));
            if slow_corr(5,ii)~=Eph(22,position)
                if ~isempty(find(slow_corr(6,ii)==Eph_last(1,:,:)))
                    position_last=find(slow_corr(6,ii)==Eph_last(1,:,:));
                    for jjj=1:size(position_last,1)
                        matr_no=(position_last(end+1-jjj)-1-rem(position_last(end+1-jjj)-1,32))/32+1;
                        column=slow_corr(6,ii);
                        if slow_corr(5,ii)==Eph_last(22,column,matr_no)
                            Eph(:,position)=Eph_last(:,column,matr_no);
                            position_contor=1;
                            break;
                        end
                    end
                    if position_contor==0
                        [xxx,pos_0]=intersect(prns,prns(ii));
                        slow_corr(:,pos_0)=0;
                        slow_corr(5,ii)=Eph(22,position);
                        slow_corr(6,ii)=Eph(1,position);
                        break
                    end
                else
                    [xxx,pos_0]=intersect(prns,prns(ii));
                    slow_corr(:,pos_0)=0;
                    slow_corr(5,ii)=Eph(22,position);
                    slow_corr(6,ii)=Eph(1,position);
                    break
                end
            else
                contor_slow=contor_slow+1;
```

```
                    end
                elseif isempty(find(slow_corr(6,ii)==Eph(1,:))) & size(slow_corr,2)>size(Eph,2)
                    slow_corr(:,ii)=[];
                end
            end
            if contor_slow==size(prns,2)
                break
            end
         end
    %alternative correction for ionosphere aided EGNOS
     for j=1:no_sats
      ind=find(dual.sats==sats(j));
      if ~isempty(ind)
         aux(j)=dual.sats(ind);
         iono(j)=dual.iono(ind);

         if abs(dual.iono(ind))<50
                  aver(aux(j),1)=aver(aux(j),1)+1;
         end
         if aver(aux(j),1)>600
           if aver(aux(j),1)==601
               aver(aux(j),3)=dual.phase(ind)/v_light*lambda1-dual.phase2(ind)/v_light*lambda2;
               aver(aux(j),2)=aver(aux(j),2)/sum(sqrt(1:600));
           end
           aver(aux(j),4)=aver(aux(j),2)+dual.phase(ind)/v_light*lambda1-dual.phase2(ind)/v_light*...
           lambda2-aver(aux(j),3);
           iono(j)=aver(aux(j),4);

           else

                if abs(dual.iono(ind))<50
                     aver(aux(j),2)=dual.iono(ind)*sqrt(aver(aux(j),1))+aver(aux(j),2);
                end
           end
          end
        end

        Pr=P'+iono_corr+fast_corr;
        P=P';
        pos=recpo_raw(P,prns,time,Eph);
        [N E U]          = car2utm(pos(1,1),pos(2,1),pos(3,1),32);
        dif(1:3,1)=[E-559948.21472328;N-6319663.50197794;U-25.6865675831214];

        [pos,number_errors]=recpo_slow(Pr,prns,time,Eph,slow_corr,number_errors);
        [N E U]          = car2utm(pos(1,1),pos(2,1),pos(3,1),32);
        dif_eg(1:3,1)=[E-559948.21472328;N-6319663.50197794;U-25.6865675831214];
    end
    if index==10000
        break
    end
end

fprintf(comm,'$PASHS,RAW,MCA,A,OFF');
fprintf(comm,'$PASHS,SBA,DAT,A,OFF');
fprintf(comm,'$PASHS,RAW,SNV,A,OFF');
fprintf(comm,'$PASHS,RAW,XYZ,A,OFF');
fclose(INSTRFIND)
```

C.2 Ionospheric Correction Computation (EGNOS)

```
function iono_error=ionospheric_computation(epoch,data_msg18,data_msg26,data)

% The program follows the following steps:
%       - determine the azimuth and elevation of each tracked GPS satellite
%       in azmelev.m
%       - determine the coordinates of the current user position in
%       get_coord.m
%       - determine the ionospheric grid signal pierce point knowing the
%       users position, the azimuth and elevation of the satellite in
%       determine_pp.m
%       - determine the neighbouring IGPs of the pierce point and extract
%       the GIVD of each IGP from message type 26 in IGP_computation.m
%       - interpolate the GIVD values of the neighbouring IGPs and find the
%       vertical delay of the pierce point
%       -  multiply the vertical delay of the pp to the obliquity factor
%       and we get the ionospheric delay that has to be added to the
%       pseudorange values


[lat_pp,long_pp,F_pp]=determine_pp(data);

[matrix]=matrix_igp(data_msg18,data);

[matr_givd,matr_give]=matrix_givd(data_msg18,data_msg26,data,matrix);

[delay,iono_int]=ipp_interpolation(matrix,matr_givd,matr_give,lat_pp,long_pp,data);

for i=1: size(F_pp,2)
    iono_error(i,1)=-F_pp(i)*delay(i);
    iono_integrity(i,1)=-F_pp(i)*iono_int(i);
end
```

C.3 Real-time Data Acquisition

```
function [sat_eph,epoch,msg,sats,t_gps]=feed_all(comm,iter,sat_eph,sats,t_gps)

while 1
    x=char(fread(comm,11));
    head=x';
%% Ephemeris Part
    if strcmp(head,'$PASHR,SNV,')
        eph(1,k)=SNV_decode(comm);
        PRN=eph(1,k).PRN;

        if eph(1,k).health==0 & find(sats==eph(1,k).PRN)
            no_eph=sat_eph(PRN).no_eph;
            if sat_eph(PRN).aode(no_eph)~=-1 & sat_eph(PRN).aode(no_eph)~=eph(1,k).iode
                sat_eph(PRN).no_eph=no_eph+1;
            end
            no_eph=sat_eph(PRN).no_eph;
            sat_eph(PRN).wn(no_eph)=eph(1,k).week_no;
            sat_eph(PRN).tow(no_eph)=eph(1,k).tow;
```

```
            sat_eph(PRN).tgd(no_eph)=eph(1,k).tgd;
            sat_eph(PRN).aodc(no_eph)=eph(1,k).iodc;
            sat_eph(PRN).toc(no_eph)=eph(1,k).toc;
            sat_eph(PRN).af2(no_eph)=eph(1,k).af2;
            sat_eph(PRN).af1(no_eph)=eph(1,k).af1;
            sat_eph(PRN).af0(no_eph)=eph(1,k).af0;
            sat_eph(PRN).aode(no_eph)=eph(1,k).iode;
            sat_eph(PRN).deltan(no_eph)=eph(1,k).deltan;
            sat_eph(PRN).m0(no_eph)=eph(1,k).m0;
            sat_eph(PRN).ecc(no_eph)=eph(1,k).e;
            sat_eph(PRN).roota(no_eph)=eph(1,k).roota;
            sat_eph(PRN).toe(no_eph)=eph(1,k).toe;
            sat_eph(PRN).cic(no_eph)=eph(1,k).cic;
            sat_eph(PRN).crc(no_eph)=eph(1,k).crc;
            sat_eph(PRN).cis(no_eph)=eph(1,k).cis;
            sat_eph(PRN).crs(no_eph)=eph(1,k).crs;
            sat_eph(PRN).cuc(no_eph)=eph(1,k).cuc;
            sat_eph(PRN).cus(no_eph)=eph(1,k).cus;
            sat_eph(PRN).omega0(no_eph)=eph(1,k).omega0;
            sat_eph(PRN).omega(no_eph)=eph(1,k).omega;
            sat_eph(PRN).i0(no_eph)=eph(1,k).i0;
            sat_eph(PRN).omegadot(no_eph)=eph(1,k).omegadot;
            sat_eph(PRN).idot(no_eph)=eph(1,k).idot;
            sat_eph(PRN).accuracy(no_eph)=eph(1,k).accuracy;
            sat_eph(PRN).fit(no_eph)=eph(1,k).fit;
            sat_eph(PRN).prnnum(no_eph)=eph(1,k).PRN;
            time_tow=sat_eph(PRN).tow(no_eph);
        end
    end

%% Pseudorange part
    if strcmp(head,'$PASHR,MCA,')
        msg=dec2bin(fread(comm,39));
        sir='';
        for i=1:size(msg,1)
            sir=strcat(sir, msg(i,:));
        end
        sequence_ID     = bin2dec(sir(1:16))/20;
        messages_left   = bin2dec(sir(17:24));
        PRN             = bin2dec(sir(25:32));
        if PRN<33
            mca_sats(num)=PRN;
            epoch(count).sat(1,PRN)   = PRN;
            epoch(count).sat(3,PRN)   = bin2dec(sir(33:40));
            epoch(count).sat(2,PRN)   = bin2dec(sir(41:48));
            epoch(count).sat(4,PRN)   = bin2double(sir(161:224));
            num=num+1;
        end

        if messages_left==0
            sats=mca_sats;
            mca_sats=[];
            frac_time=sequence_ID;
            num=1;
            if frac_time==0
                t_gps=t_gps+1800;  %verificat ideal pt sambata/duminica la rollover
            end
            time=t_gps-mod(t_gps,1800)+frac_time; %adding 30 mins
            epoch(count).time=time;
        end
    end
```

```
%% SBAS Part
    if strcmp(head,'$PASHR,SBA,')
        sba_msg=fgetl(comm);
        [dat,sat_no,msg_no,health,msg]=strread(sba_msg,'%s%d%d%d%s',1,'delimiter',',*');
        sba(count).msg=msg;
        sba_msg=char(msg);
        sba(count).time=time;
        count=count+1;
    end

    if strcmp(head,'$PASHR,XYZ,')
        msg=dec2bin(fread(comm,6),8);
        sir='';
        for i=1:size(msg,1)
            sir=strcat(sir, msg(i,:));
        end
        time=bin2dec(sir(1:32))/1000;
        no_sats=bin2dec(sir(33:48));
        bytes2read=no_sats*34+2+2;
        msg=dec2bin(fread(comm,bytes2read),8);
        sir1='';
        for i=1:size(msg,1)
            sir1=strcat(sir1, msg(i,:));
        end
        for k=1:no_sats
            prn=bin2dec(sir1(1+272*(k-1):16+272*(k-1)));
            prns(k)=prn;
            satx(k)=bin2double(sir1(17+272*(k-1):80+272*(k-1)));
            saty(k)=bin2double(sir1(81+272*(k-1):144+272*(k-1)));
            satz(k)=bin2double(sir1(145+272*(k-1):208+272*(k-1)));
            range(k)=bin2double(sir1(209+272*(k-1):272+272*(k-1)))*v_light;
        end
        for i=1:size(prns,2)
            for j=1:size(prns,2)
                if prns(i)<prns(j)
                    aux=prns(j);
                    prns(j)=prns(i);
                    prns(i)=aux;
                    aux=range(j);
                    range(j)=range(i);
                    range(i)=aux;
                end
            end
        end
        corrected.range=range;
        corrected.time=time;
        corrected.sats=prns;
    end
    if count>iter
        break;
    end
end
```

Bibliography

[Borre, 2003] Borre, K. (2003). http://kom.aau.dk/ borre/masters/gps/easy/.

[Borre et al., 2007] Borre, K., Akos, D. M., Bertelsen, N., Rinder, P. & Jensen, S. H. (2007). A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach. Birkhäuser Boston.

[ESA, 2006] ESA (2006). The European Geostationary Navigation Overlay System - A Cornerstone of Galileo. European Space Agency.

[Farrell & Barth, 1999] Farrell, J. A. & Barth, M. (1999). The Global Positioning System and Inertial Navigation. McGraw-Hill.

[GPS, 1991] GPS, N. (1991). Interface Control Document. Arinc Research Corporation.

[Navigation, 2002] Navigation, T. (2002). Z Family GPS receivers - Technical Reference Manual. www.thalesnavigation.com.

[Parkinson & Jr., 1996] Parkinson, B. W. & Jr., J. J. S. (1996). Global Positioning System: Theory and Applications. American Institute of Aeronautics and Astronautics, Washington,D.C.

[RTCA, 2001] RTCA (2001). Minimum operational performance standards for Global Positioning System Wide Area Augmentation System airborne equipement. European Space Agency.

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

[Thales, 2008] Thales, N. (2008). `http://www.navtechgps.com/pdf/DG14.pdf`.

[Toran et al., 2004] Toran, F., Ventura, J., Gonzlez, E., Toledo, M., Catalina, A., Barredo, C. & Salonico, A. (2004). http://www.egnos-pro.esa.int/Publications/SISNET/SISNeT_GPSWorld_0404.pdf.