Developing a GNSS data analysis tool to characterize the faulty event called Code-carrier incoherence

Guillermo Centelles Hidalgo

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Prof. Agostinho Rui Alves da Fonseca
Dr. Pedro Tavares Moura e Silva

Examination Committee

Chairperson: Prof. Filipe Szolnoky Ramos Pinto Cunha
Supervisor: Prof. Agostinho Rui Alves da Fonseca
Member of the Committee: Prof. João Manuel Gonçalves de Sousa Oliveira

November 2017

Acknowledgments

First of all I want to thank the IST for let me introduce into the world of programming and navigation by satellite. I have been fascinated by the way they coordinate practical and theoretical teaching and from now on it is a university that I will always recommend. Thanks to my supervisor, Agostinho Fonseca for having supported me to make the Master Thesis that I liked the most unconditionally.
Also, I want to thank Deimos Engenharia and Pedro Tavares, who have given me the opportunity to do the Master Thesis with them, I have enjoyed an experience of a real and current project.
I want to thank all my friends who have supported me and encouraged me in the most difficult moments, when no results came out. To de Becerra Sanchez's family, Isa and my father for making me feel that I always have their support.

Especially, and I left it for the end, there are two people who play an important role in my life to whom I would thank a thousand times.
To Irene because the affection and strength she gives me can not be measured with words. She has made me never feel alone in the realization of this project. Thanks my life. And my brother because he gets me to never abandon what I propose to myself, I would never have become as I am if it were not for his dedication, his way of doing things and his desire to see me grow as a person. Thank you both for making me feel that this work is the result of a team that all three of us form.

Abstract

The accuracy, availability and integrity of the Galileo system are of significant concern because of the intended benefits of this system in many applications in a near future, when the system will be fully operative. Thus it is of critical importance that every satellite works and any potential error is minimized and bounded. This thesis analyzes the problem which a type of satellite failure, named code-carrier incoherence, generates and discusses how to overcome it with coherent multipath tracking, using the Galileo E1 frequency band as main frequency and the Galileo E5 frequency band as a secondary frequency. Then, variations with respect to the nominal Multipath receiver environment are detected and afterwards, data is combined to reduce the effect of measurements errors. For that, the antenna calibration has been performed, where the multipath mean computed as a function of elevation angle for a period of days. This work shows a way to effectively use Multipath and its standard deviation that avoids cycle slips and obtains well determined archs without phase ambiguities as a result. A tool is developed in C++ to compute a code-carrier coherence (CCC) value and a Gaussian error propagation of CCC. Finally, a CCC threshold value is established and when the CCC value overpasses it, the code carrier incoherence is detected. Details about the current and future status of Galileo, its data use in receiver stations and in the GPS Tool Kit, the multipath estimation algorithm for Galileo measurements are also noted to provide context for future improvements of this developed tool.

Keywords: Code-Carrier Coherence (CCC), Multipath, Feared event

Resumo

A precisão, disponibilidade e integridade do sistema Galileo são motivo de preocupação, devido aos benefícios pretendidos deste sistema em muitas aplicações num futuro próximo, quando o sistema estiver totalmente operacional. Assim, é de importância crítica que cada satélite funcione e que qualquer erro potencial seja minimizado e delimitado. Esta tese analisa o problema que um tipo de falha de satélite, nomeadamente a incoerência do portador de código, gera, e discute como superá-lo com um localizador de múltiplas trajectórias consistente, usando a banda de frequência Galileo E1 como frequência principal e a banda de frequência Galileo E5 como frequência secundária. Em seguida, são detectadas variações em relação ao ambiente nominal do receptor de múltiplas trajectórias e, posteriormente, os dados são combinados para reduzir o efeito de erros de medição. Para isso, a calibração da antena foi realizada, onde a média de trajectórias é calculada como uma função do ângulo de elevação por um período de dias. Este trabalho mostra uma maneira de usar efetivamente as trajectórias e o seu desvio padrão que evita desvios de ciclo e obtém arcos bem determinados sem ambiguidades de fase como resultado. Uma ferramenta é desenvolvida em C ++ para calcular um valor de coerência do portador de código (CCC) e uma propagação de erro gaussiana de CCC. Finalmente, um valor de limiar de CCC é estabelecido e, quando o valor CCC é ultrapassado, a incoerência do portador de código é detectada. Detalhes sobre o status atual e futuro do Galileo, o uso dos seus dados em estações receptoras e no conjunto de instrumentos GPS, o algoritmo de estimativa de trajectórias para medições Galileo são também assinalados para fornecer contexto para futuras melhorias desta ferramenta.

Palavras-Chave: Code-Carrier Coherence (CCC), Multipath, Evento temido

Contents

List of Acronyms

ARL:UT	Applied Research Laboratories at the University of Texas
BDS	BeiDou Navigation Satellite System
C/A	Coarse Acquisition
CCC	Code-Carrier Coherence
CLS	Collecte Localised Satellites
CNES	Centre National d'Etudes Spatiales
CORS	Continuously Operating Reference Station
CRC	Cycle Redundancy Check
CS	Commercial Service
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
EUREF	European Terrestrial Reference System
FTP	File Transfer Protocol
GGTO	GPS to Galileo time offset
GLONASS	Globalnaya navigatsionnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPSTK	GPS ToolKit
IDE	Integrated Development Environment
IEC	Ionospheric Electron Content
IGS	International GNSS Service
IRNSS	Indian Regional Navigation Satellite System
IST	Instituto Superior Técnico
IST	Instituto Superior Técnico
LOS	Line of Sight
MP	Multipath
MPL	Minimum Performance Levels
NASA	National Aeronautics and Space Administration
NMA	Navigation Message Authentication
OS	Open Service
OS-SDD	Open Service-Service Definition Document
PR	Pseudorange
PRN	Pseudorandom noise
PRS	Public Regulated Service
QZSS	Quasi-Zenith Satellite System

RINEX	Receiver Independent Exchange Format
SAR	Search and Rescue
SBAS	Satellite Based Augmentation System
SoL	Safety of Life
SP3	Standard Product 3
UNAVCO	University-Governed Consortium
WAAS	Wide Area Augmentation System

List of Symbols

$\boldsymbol{A}$	Constant = $40.3^3 s^{-2}$
$\boldsymbol{A_i}$	Ambiguity term
$\boldsymbol{Ant(\phi)}$	Antenna Calibration for elevation angle ϕ
$\boldsymbol{c}$	Light velocity
$\boldsymbol{CCC^i}$	Code Carrier Coherence for satellite i
$\boldsymbol{dT}$	Real-time-derived receiver clock offset
$\boldsymbol{e_M P}$	Code Multipath and noise
$\boldsymbol{f_{carrier}}$	Carrier phase frequency
$\boldsymbol{f_{code}}$	Pseudorandom code frequency
$\boldsymbol{freq}$	Frequency
$\boldsymbol{I_i}$	Term due to ionospheric delay for frequency i
$\boldsymbol{K}$	Integration constant
$\boldsymbol{MP}$	Multipath
$\boldsymbol{MP_{cx}}$	Multipath for a type of data called cx
$\boldsymbol{MP_j^i}$	Biased Multipath for satellite i and receiver station j
$\boldsymbol{N}$	Number of measurements
$\boldsymbol{N_{bins}}$	Number of bins
$\boldsymbol{\overline{CCC}}$	Mean CCC
$\boldsymbol{\overline{MP}}$	Mean Multipath
$\boldsymbol{P_i}$	Code measurement for frequency i
$\boldsymbol{PR}$	Probability
$\boldsymbol{PR}$	Pseudorange
$\boldsymbol{s}$	Term which includes geometric range and the difference of clock offsets
$\boldsymbol{\sigma}$	Standard Deviation

List of Greek Symbols

α	Represents I_i/I_j
$\delta\phi$	Elevation angle increment
ϵ_{MP}	Carrier phase Multipath and Noise
μ	Mean
μ_j^i	Leveled Multipath for satellite i and receiver station j
ϕ	Elevation angle
Φ_i	Carrier phase for frequency i
σ_{ccc}	CCC Standard Deviation
σ_{ccc}^i	CCC Standard Deviation for satellite i
σ_i^j	Standard Deviation of leveled Multipath from Antenna Calibration
$\Theta_c cc$	CCC Threshold

List of Tables

List of Figures

[1] https://play.google.com/store/apps/

Chapter 1

Introduction

Nowadays, most of the people over the world have the possibility to know their position in real time on Earth [9]. That is possible by the Global Navigation Satellites Systems (GNSS), a satellite navigation system with global coverage. Global Navigation is one of the most promising applications that satellites have been used for, and presently, several navigation systems are available:

- Global Positioning System (GPS) provided by the United States of America;
- Galileo by the European Union;
- GLONASS by the Russian Federation;
- BeiDou Navigation Satellite System (BDS) by China.

They compose the list of the uniques global navigation systems. One can check the status of GPS[1] with 32 satellites, the status of Galileo[2], with 18 satellites,the status of GLONASS[3], with 24 satellites , and the status of BDS[4], with 16 satellites, and getting an overview of how many of them are actually operating or what kind of failure or restriction they have. There are some local navigation systems as QZSS by Japan and IRNSS by India.

1.1 Accuracy, Integrity, Continuity and Availability

All of those satellite systems mentioned above are used to provide a good navigation service to users. The combination of every constellation creates a complex system which can provide a better positioning accuracy, integrity, continuity and availability. It is important to have these terms clarified to have a good understanding of the researching efforts in navigation through different approaches.

1.1.1 Accuracy

The accuracy of an estimated or measured position at a given time of a user is the degree of conformance of that position with the true position, speed and time of the user. Accuracy is a statistical measure of the error of the position. The navigation system accuracy has no meaning unless it includes a statement of the uncertainty in the position that applies.

[1] https://www.navcen.uscg.gov/?Do=constellationStatus
[2] https://www.gsa.europa.eu/european-gnss/galileo/system-status
[3] https://www.glonass-iac.ru/en/GLONASS/
[4] https://igscb.jpl.nasa.gov/projects/mgex/Status_BDS.htm

In terms of orthogonal axes, the 95 percent confidence level is used. Vertical accuracy and horizontal accuracy are specified in one-dimensional terms (2 sigma), 95 percent confidence level. [10]

1.1.2 Availability

The availability in satellite navigation systems is the time percentage which the services of the system can be accessible by the user. Availability is a measurement of the capacity of the system to provide a service in a specific position [3]. Signal availability is the time percentage that transmitted navigation signals from external sources are feasible for the user to use them. It is a function of both the environment features and also the technical capabilities of the transmitter characteristics. [10]

1.1.3 Continuity

The system continuity is the ability of the total system to perform its function without interruption during a specific operation. Deeper, continuity is the probability that the system performance is maintained for the duration of a phase of operation, presuming that the system was available at the beginning of that step of the total operation. [10]

1.1.4 Integrity

Integrity is the measure of the trust that can be placed in the correctness of the information given by a navigation system. It includes the system ability to provide timely warnings when the system should not be used for navigation to users. [10]

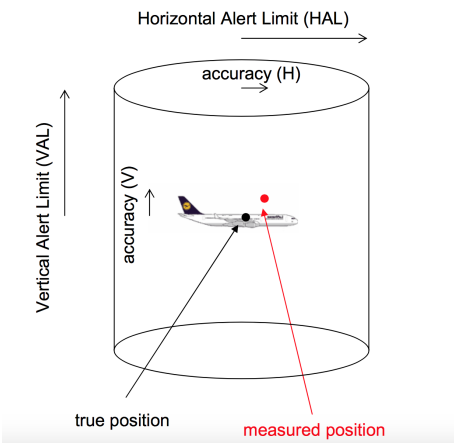

Figure 1.1: Concept of integrity [3]

Integrity exists if it is detected when the measured position leaves the bounds (worst case situation). See Figure 1.1. This timeliness also includes protecting users against satellite faults whenever they occur. Such faults are called feared events, and to adequately protect the user they must be well characterized.

1.2 Feared events

In the navigation field there are many different sources of errors as satellite geometry, satellite orbits, multipath effect, atmospheric effects, clock inaccuracies and rounding errors. Some of them can alter the

accuracy, availability, continuity or even the integrity. These feared events have been studied in static receivers to create models to avoid them in future events. They are cycle slips, multipath and code-carrier incoherence. Although cycle slips is an interesting feared events to study because they can appear in almost every current GNSS receiver, they are not studied deeper in this project. In the case of the cycle slip, one should know that it is a discontinuity in a receiver's continuous phase lock on a satellite's signal. From now on, this project will focus in multipath and mainly in code-carrier incoherence. [11]

1.2.1 Multipath

Multipath interference occurs when the user device receives reflected signals in addition to the direct line of sight (LOS) signal. These interference signals are generally reflected from the ground, buildings like in Figure 1.2 or trees in terrestrial navigation, while signal reflections from the host-vehicle body are more common in airborne and marine applications. [12]

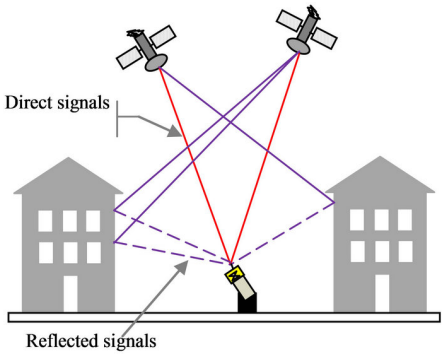

Figure 1.2: Principle of multipath signal in cities [4]

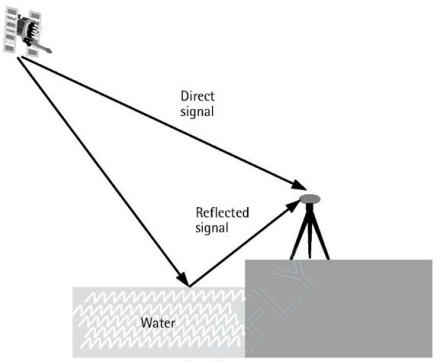

Figure 1.3: Multipath propagation in sea

Two kinds of multipath exist: specular multipath arising from discrete, coherent reflections from smooth surfaces such as standing water, and diffuse multipath arising from diffuse scatterers and sources of diffraction. (The visible glint of sunlight off a choppy sea is an example of diffuse multipath like could happen in Figure 1.3 from [5])

Multipath signals are generally considered undesirable in the GNSS field because they destroy the correlation function shape used for time delay estimation, but can be useful in some cases e.g. for acquisition. Although some wireless communications techniques exploit multipath to provide signal diversity, the key point in GNSS is to efficiently mitigate the multipath effect because only the LOS signal is used for getting the satellite-receiver transit time offset for positioning. [12]

[5] http://what-when-how.com/gps/gps-errors-and-biases-part-1/

1.2.2 Code-Carrier incoherence

For a better understanding of what this faulty event is, it is preferable to explain first, what the code-carrier coherence is. The code-carrier coherence shows the coherence between the code phase and the carrier phase and has a significant role in some applications as carrier phase ranging, carrier phase smoothed pseudorange, etc. For satellite navigation systems, guaranteed code-carrier phase coherence of the satellite transmitted signals is not only important for its ranging function, but also one of the main factors which affects the positioning accuracy of satellite navigation signals. Signal coherence is the relationship between code phase and carrier phase, and phase coherence is the relationship between the phases, i.e., constant phases or different phases. One example of code-carrier coherence boundary is the offset between short-term (less than 10s) code frequency and carrier frequency that has been defined in the U.S. for the WAAS system signal code-carrier phase coherence. : [13]

$$\frac{f_{code}}{1.023MHz} - \frac{f_{carrier}}{1575.42MHz} < 5 * 10^{-11} \tag{1.1}$$

Where the f_{code} is the pseudorandom code frequency and the $f_{carrier}$ is the phase carrier frequency. Code carrier incoherence introduces an additional time varying bias between the code and carrier observables, which will degrade the positioning accuracy when using code and phase data together.
The main factors causing the failure in the code-carrier coherence are:

1. Failure in the satellite. Incoherence between the code emitted by the satellite and the carrier;
2. Ionosphere or propagation path error;
3. Irregularities of the code loop or carrier loop inside the receiver and other factors.

Accordingly, several receivers are used to observe at the same time, therefore as to eliminate the measurement errors caused by the irregularities of the internal code loop or the carrier loop of the receivers through mutual calibration. Finally, the code-carrier incoherence is mainly caused by the delay jitter of satellite signals. [13]

1.3 Thesis layout

The aim of this project is to detect where the feared event of the code-carrier incoherence appears in Galileo satellites. Up to arrive to that point, some steps have been done first, as avoiding cycle slips, computing multipath to track coherence and also variations with respect to the nominal MP receiver environment as a function of the elevation angle. Then the data is combined to reduce the error effects using different receiver stations.

This dissertation will present the current situation of Galileo and its performance in the State of art section, 2, then the methods and materials used are explained in Section 3, the results obtained for intermediate procedures and final plots are shown in Section 4, the discussion where the results are compared with papers and other works in Section 5, the conclusion of achievements and a summary of final results in Section 6 and at the end the future works in Section 7.

Chapter 2

State of art

In this section, the current situation of Galileo and its position in the GNSS world will be explained as its performance. There is also a brief introduction of what is currently known of the code-carrier coherence and its use in Galileo system.

2.1 GNSS and Galileo nowadays

At the moment, GPS and GLONASS are the most extensive and accessible positioning system. They are the only two fully operative systems in 2017. GPS ,which is global opperative since 1978, and GLONASS achieved Russia Global coverage up to 2007 when it had enough available satellites [14]. Until now, GNSS users have had to depend on non-civilian American GPS or Russian GLONASS signals. Galileo, the Europe's Global Satellite Navigation System, is the reliable alternative that, unlike GLONASS and GPS, remains under civilian control.

Figure 2.1: The four systems belonging to GNSS and an example of their satellites: GPS, GLONASS, Galileo and BDS.

In the Figure 2.1 one can find the symbol and an example of satellite of each global navigation satellite. While European independence is a principal objective of the programme, Galileo also gives Europe a place in the GNSS world. The programme is designed to be compatible with all existing and planned GNSS and interoperable with GPS and GLONASS. In this sense, Galileo is positioned to enhance the coverage currently available, providing a more seamless and accurate experience for multi-constellation users around the world. [15]

2.1.1 Galileo status

Satellites GIOVE-A and GIOVE-B for experimental steps were launched in 2005 and 2008 respectively, testing critical Galileo technologies, and also the securing of the Galileo frequencies within the International Telecommunications Union. Operational Galileo satellites were launched in 2011.[1]

Galileo has made important progresses in recent years. Eighteen Galileo satellites are currently orbiting the Earth, and the supporting ground station infrastructure is working properly. As a result, Galileo is now quiet ready to be used.

With the introduction of Galileo Initial Services, Galileo officially moves from a testing phase to a provision of live services. This is the first time that users around the world can be guided using the navigation, positioning and timing information given by Galileo's global satellite constellation.
Galileo Initial Services are the first step on the way to its Full Operational Capability. The Galileo performance will gradually improve as the constellation will increase with additional satellites. Once completed in 2020, users will benefit from its reliability and coverage and its full first-class performance. [15]

Galileo-enabled devices

European satellites are guiding users providing them with global positioning, navigation and timing information. In the advance to Galileo Initial Services, many ahead companies have created Galileo-enabled receivers, modules and chipsets – currently, many of which are available on the market.
Today, 17 companies, which represent more than 95% of the global satellite navigation supply market, produce Galileo chips ready[2]. To have a look of the range of the different devices which are Galileo-enable, there is a widget in the European Global Navigation Systems Agency webpage [3].

Mobile devices

In mid-2016, the Spanish technology company BQ launched the Aquaris X5 Plus, the first European Galileo-ready smartphone. Integrating Qualcomm's Snapdragon 652 processor, the Aquaris X5 is a multiconstellation smartphone that is capable of receiving also GPS, GLONASS and BDS signals. [16]

Since December 2016 Galileo satellite positioning system has been operating as an 'initial service', with 18 of the 30 satellites that will form the system in 2020. Its signal offers higher resolution than GPS and Glonass competitors, and the ability to use Galileo could be an argument to boost new mobile devices with more accurate location. Following mobile specifications, some chips as Kirin 960 eight-core chip are available to receive signals from the constellations GPS (US), Glonass (Russia), BDS (China) and Galileo (Europe).

[1] http://www.esa.int/Our_Activities/Navigation/Galileo/What_is_Galileo
[2] https://www.gsa.europa.eu/galileo/services/initial-services/galileo-enabled-devices
[3] http://www.usegalileo.eu/EN/index.html

The test results has been disappointing. In theory, those new devices must capture all the available satellites, getting the best signals they receive to position themselves more effectively, transparently for the user. The system, so transparent to Galileo, that it can not be seen. Every latest chips can receive signals from all four systems. The question is why the available software does not recognizes them. Which is not apparently the situation.[4]

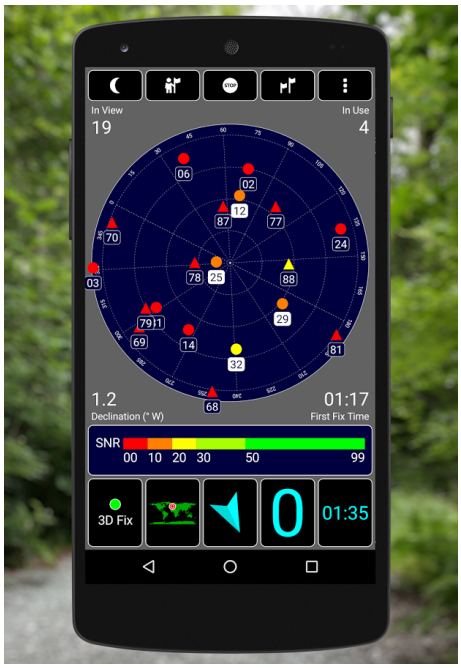

Figure 2.2: Example of GNSS app called GPS Test. As it shows, there are many satellites in view but none of them are Galileo (GPS are the circle shape ones, GLONASS are the triangle shape and Galileo are the diamond shape[6])

As the app shown in Figure 2.2 there are many other apps to test the GNSS status. Unfortunately none of them shows the Galileo System for the moment. The requirements for that are that the mobile device (GNSS receiver) are made to use Galileo to compute its position and also the app can have the choice to include Galileo data. Both hardware and software must have included Galileo as one of the satellite navigation systems to test it.

Autonomous Vehicles

The Galileo project will provide satellite navigation to the autonomous car. This is confirmed by Javier Benedicto, head of the Galileo project at the European Space Agency (ESA), at a conference at the annual Ametic conference.[7]

The agency, in collaboration with Renault, Volkswagen and soon with Seat, research the development of this type of vehicles. The signal from Galileo is capable of reaching places where there is no direct line of sight to the satellite, may be the key to guarantee the safety of the autonomous car, together with the GPS and the 5G network. Galileo is protected from cyber attacks because it is not connected to the Internet and allows to deploy navigation in places where it does not reach 5G. Currently, the 18 satellites in orbit of Galileo provide

[4] http://www.elmundo.es/economia/2017/07/20/59709da3468aeb893a8b4670.html
[7] https://www.kippel01.com/tecnologia/el-proyecto-galileo-orbita-el-negocio-del-coche-autonomo.html

an accuracy of less than one meter and the aim of the ESA is to fit ”a few centimeters”, which will adjust their options for the autonomous vehicle.
More improvements in GNSS constellations are further contributing to the path in the way to the fully autonomous vehicle, with Galileo features such as Open Service NMA and dual frequency are a key role in providing an e cient, reliable, robust and low-cost defence against spoofing attacks or jamming. [16]
The Galileo Open Service (OS) will provide a Navigation Message Authentication feature, called the Open Service Navigation Message Authentication (OS-NMA). Through this feature, users can verify that a navigation message comes from a Galileo satellite and not from a potentially malicious source. [8]

2.1.2 Galileo, improvements

Nowadays, Galileo is providing improved positioning and timing data with significant positive implications for many European services and users. With its satellites working together with GPS and GLONASS, there are more satellites available, meaning more accurate and reliable positioning for end users. In particular, navigation in cities, where buildings can often block satellite signals. The increase in the number of satellites is fundamental for having at least the minimum of satellites required, taking into account that the elevation mask in cities could raise until very high elevation angles [17]. Indeed, Galileo is bringing many other improvement to the GNSS world as:

- The power received from the Galileo signals is more than the double of the power of the C/A code from the current GPS, which decrease tracking noise for both phase and code ranges; [18]

- Each Galileo signal also has a pilot (dataless) component, which can be easily acquired independently, with no data bits decoding. This has several benefits to a data-bearing signal like the GPS CA code, in addition to reduced noise and better tracking integrity at low signal power; [18]

- A better 3-step coding scheme for navigation bits is used (Viterbi convolutional encoding, bit interleaving and cyclic redundancy check (CRC)). This significantly increases the reliability of navigation message decoding when there is interference or with low signal power. [18]

- Galileo’s excellent 30 nanosecond timing accuracy helps to enabling more resilient synchronisation of banking and financial transactions, telecommunication and energy distribution networks which contributes to operate more efficiently; [15]

- It boosts European innovation, contributing to the creation of many products and services, creating jobs and allowing Europe to participate in the EUR 175 billion global GNSS market. [16].
  Satellite navigation has helped the world economic growth of the high-tech industries. It is predicted that the global satellite navigation market will grow by more than 18% up until 2019. The Galileo resiliency is expected to enable many new applications and services that will benefit for increased positioning reliability, in addition to driving economic growth in Europe and beyond. [15]

- Indeed, Galileo provides European citizens with independence and sovereignty, a range of environmental benefits and several new specific services to the Galileo programme (Open Service, Search and Rescue, Commercial Service). Galileo’s Search and Rescue service provides a reduction of the time it takes to detect emergency pain beacon signals from up to three hours to just ten minutes. Since the locations of the beacons are also determined more accurately, people in distress, whether at sea or in the mountains, can be rescued more quickly. [15]

[8] https://www.gsa.europa.eu/newsroom/news/new-funding-opportunity-galileo-open-service-authentication-user-terminal

Mitigating risks

Satellite positioning has become a fundamental service that it is often taken for assumed. Just think what would happen if GNSS signals were suddenly disappeared. Ship and aircraft crews, truck and taxi drivers and thousand of millions of people around the world would suddenly be lost.
Indeed, public utilities, financial and communication activities, humanitarian and security operations and emergency services would all come to a stop. e.i as the use of satellite-based navigation systems continues to expand, the consequences of a signal failure become even greater.
With the addition of Galileo to the global GNSS constellation, these risks are minimised and also ensure better performance and accuracy for the all the users. [15]

2.2 Current Galileo performance

After the Declaration of Galileo Initial Services, in the first quarterly reporting period the measured Galileo Initial Open Service performance figures generally exceeded the Minimum Performance Levels (MPL) targets specified in the OS-SDD by important margins [9]. It has summarized the following data on Galileo:

- Availability of a Healthy Signal, with average monthly values better than 97.33%, is significantly above expectations, where the MPL is of 87%. The “Availability of HEALTHY Signal in Space” is defined, for each Galileo constellation operational satellite, as the percentage of time that the specific satellite broadcasts healthy Galileo Open Service Signals in Space. [19]

- The Space Ranging Accuracy signal shows a 95% monthly accuracy better than 1.07 [m] for individual space vehicles. In concern of the [OS-SDD] MPL, in this case it is also achieved, the threshold is set to 7 [m]; [19]

- Galileo Availability UTC time determination service was achieved compared to the [OS-SDD] MPL target of 87 percent, with a monthly value of 100 percent;

- ”Availability of GPS to Galileo time offset (GGTO) determination was 100 percent in January and March”. February has showed a slightly lower figure of 96.44 percent, although it is still better than the [OS-SDD] MPL aim of 80 percent.

The purpose of this initial OS SDD is to describe the characteristics and performance of the Galileo open system during its initial phase. The initial OS SDD presents the MPLs targeted for such service and defines the conditions under which such MPLs can be reached. [20]
Galileo satellites are broadcasting the time offset between the parallel navigation systems timings, accurate to a few billionths of a second. Formally known as the GGTO the accuracy of the offset is being benchmarked at five nanoseconds or less.[10]

2.3 Code-Carrier Coherence

The quality of GNSS signals directly influences the navigation performance and great importance is attached to the test and evaluation of navigation signal quality at home and abroad. For instance, Stanford

[9] http://gpsworld.com/first-galileo-open-service-performance-report-published/
[10] http://www.esa.int/Our_Activities/Navigation/Galileo_and_GPS_synchronise_watches_new_time_offset_helps_working_together

University has set up a GPS navigation laboratory (GPS Lab) which is committed to monitoring and assessing of the quality of navigation signals, and establishing a satellite navigation signal monitoring system. The European Space Research and Technology Centre (ESTEC) conducted the monitoring and assessment of space signals in its navigation labs using the Galileo system and built up several ground-based observatories in different places. [13] Although modern GPS receivers achieve high pseudorange accuracy in line-of-sight (LOS) conditions, multipath remains a dominant source of ranging error in GNSS. On the other hand, for GPS, the carrier code incoherence is well characterised, but for the new constellation it is not. Concerning on the validation of the monitoring receiver based test method, test results show that the accuracy of the code-carrier coherence test is less than 0.5 degree [13]. As it is explained in the Section 1.2.2. Examples of WAAS monitors include a code-carrier coherence monitor, which checks for divergence of the timing references for code and carrier measurements [21]. Thus, the developed tool for Galileo in this project gains relevance.

Chapter 3

Material and Methods

This chapter explains the procedure and material used to create the tool and which steps to follow to achieve to the same results as shown in 4.

3.1 Galileo

The tool developed in this project is oriented to the Galileo satellite navigation system, thus, it is important having a deep knowledge of what Galileo offers to the world and taking a look of how its internal structure is formed.
The Galileo satellite navigation system will provide five different services: Open Service (OS), Commercial Service (CS), Safety of Life (SoL), Public Regulated Service (PRS), and Search and rescue (SAR). Each service has its application and focuses to a different public. [15]

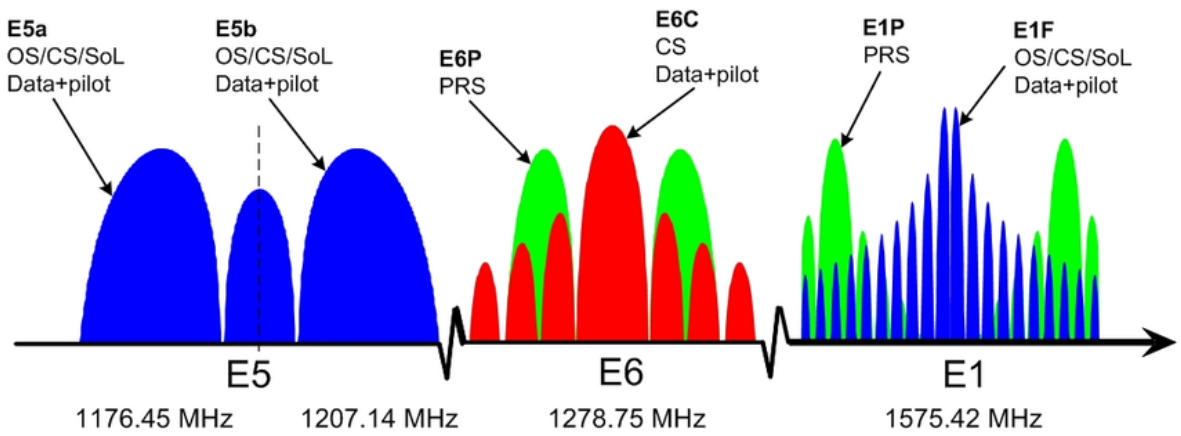

Figure 3.1: Galileo Frequency Spectrum [5]

OS is the most popular GNSS service, it is free of charge and it is not encrypted. It will have world-wide coverage and it is optimized for mass market applications. The OS is provided on three different frequency bands (E1, E5a and E5b). CS is mostly the same as OS but provides an added value by incorporating additional information in the data-stream such as correction data and integrity information. CS is restricted via encryption for commercial exploitation. This service also includes E6. The SoL is adapted for requirements from commercial flight regulations. This has resulted in stringent requirements on integrity monitoring and time

to alert limits. SoL will be provided free of direct user charges but access will probably be restricted to specific user groups by encryption. [5] [17]

The Galileo PRS is encrypted and is ”specifically designed to resist interferences, as spoofing” (an attack in which a person or organization supplants an identity by falsifying data, thereby gaining an illegitimate advantage). With its improved security features PRS allows government authorities to enjoy a continuity of service across a wide range of military and civilian applications with important benefits for public safety and security. PRS will be only available to users authorized by the Member States of the European Union and ESA. [1]

SAR is a support of COSPAR/SARSAT system. ”The added value of the SAR service is the communications link between the person needing rescue and the search and rescue operation centre”. The communication link is double way and allows the comunication of short text messages. The Galileo system will provide a global coverage of this service. [5]

The different services are offered on multiple signals and frequency bands. In total, a Galileo satellite has five different RF-signals on three different frequency bands E1, E5 and E6. As it is shown in Figure 3.1, some of the signals have a data channel (navigation data and ranging code) and a pilot channel (no navigation data, only ranging code) with both the I (data) and Q (pilot) components. In total, there are 10 different signals which are used to define the Galileo services.

3.1.1 Frequency bands chosen

After having an overall idea of the main features of each frequency band, it is possible to explain the choice of the frequency bands used in this project. In the Figure 3.1 it is shown the all the frequencies and also which type of service is provided within each frequency band. Indeed, it is shown which frequency band provide also pilot data. The most popular service, as it is said above is the Open Service, thus, only the frequency bands available for this service are chosen. These are E1, E5a and E5b frequency.

Then, as it is explained in Section 3.5, 2 frequencies are required. The main one is from which the multipath is obtained and the secondary one contributes with its carrier phase. The reason for this choice is that the effect of removing the ionosphere stronger when the carrier frequencies are further away. Moreover, because the E1 frequency also known as L1 frequency is the most widely used around the world. For those reasons the main frequency will be E1 (1575.4 MHz) and the secondary frequency will be E5a (1176.45 MHz).

[1] https://www.gsa.europa.eu/news/prs-secure-opportunity

3.2 GPSTK

The GPS Toolkit (GPSTK) was developed to support GPS applications, it is an open source library and a suite of applications created to the satellite navigation community, to researchers research purposes and not at lower level coding.[2]
The GPSTk suite consists of a set of core library and auxiliary libraries, and a set of applications. The GPSTk provides a big quantity of functions to solve processing problems related to GNSS such as handle standard formats such as Receiver Independent Exchange Format (RINEX) and Standard Product 3 (SP3). The libraries are the basis for deeper advanced applications distributed as a part of the GPSTk suite. [6].

Figure 3.2: GPSTK: The tool kit which helps working on GNSS [6]

Since its first release the GNSS world is evolving into a multi-constellation system: GLONASS has been revived, Galileo is a reality, the Chinese have launched Beidou, the Japanese QZSS, and GPS has slowly been adding the more types of signals to the existing constellation. None of these new systems or signals were operational at the time of the GPSTk's first release, when it was aimed at handling only GPS L1 and L2 signals.

Many changes required to support this multi-GNSS processing have been incorporated into the GPSTk. Examples of this changes in the last years are due to RINEX from version 2 to 3, incorporation of Galileo or Beidou, addition of frequency L5 in GPS, etc. [22]. The Space and Geophysics Laboratory sponsors GPSTk, with the colaboration of the Applied Research Laboratories at the University of Texas at Austin (ARL:UT). GPSTk is a product of GPS research at ARL:UT since before the first satellite was launched in 1978 as a combination of the effort of many software engineers and scientists. Opening source much of their basic GNSS processing software as the GPSTk was decided by the research staff at ARL:UT in 2003. [6]

3.2.1 C++

The GPSTk has an object-oriented design and a C++ implementation, and the original design, which, while sufficient to handle the world of dual-frequency GPS measurements, needed to be extended and expanded to address the multi-GNSS complexities that now exist. This changes have been made in the classes of the library, adding functions and new choices for the objects as TypeID, Satsystem, etc, which may be the inputs for many functions, which just work with the more basic mode, GPS L1 and L2 frequencies. The quantity of current and planned efforts are aimed at improving the reliability, maintainability, and usability of the GPSTk [22].

3.2.2 Doxygen

Doxygen has the GPS ToolKit Software Library Documentation. It describes the architecture and design of the GPS ToolKit (GPSTk), produced by the Space and Geophysics Laboratory (SGL) of ARL:UT. This

[2] //http://www.gpstk.org/bin/view/Documentation/WebHome

documentation is nightly generated from the GPSTk Subversion repository [3]. In this web page, the classes, structures, unions and interfaces with brief descriptions are publicly available. Clicking in each one, its members (public or private) are shown and some information about its hierarchy. It helps the user to know how implement their code, and which class, struct, union or interface fits better. On the other hand, the file list it is also available, where it is openly visible the code of each class. It is useful to know how they work and for doing debugging in the user program.
Due to the fact that there is not a specific manual for using GPSTK, Doxygen is the best way to learn about how to use GPSTK and what it is possible to do and its limitations. There is also some examples which can be imitated for the first steps of a new program.

3.3 International GNSS Service (IGS)

The International GNSS Service (IGS), formerly the International GPS Service, is a voluntary collaboration of more than 200 contributing organizations in more than 80 countries. The IGS global tracking network of more than 300 permanent, continuously-operating GNSS stations provides a rich data set to the IGS Analysis Centers, which formulate precise products such as satellite ephemerides and clock solutions. IGS Data Centers freely provide all IGS data and products for the benefit of any investigator. [1]
Perhaps the most important prerequisite for a successful service and the ease of utilization of IGS products is the standardization of data and product formats. IGS has adopted and developed a number of standard formats. The formats used in this project: RINEX and SP3 undergo regular revisions to accommodate receiver/satellite upgrades, or multi-technique solutions, respectively. [23] [1]

Format name	**IGS Product**	**Sampling**
RINEX	GNSS data	30 sec
SP3	Orbit/Clocks	15 min (900 sec)

Table 3.1: IGS formats [1]

For RINEX and SP3 files, their products and sampling is shown in the Table 3.1.

3.3.1 Receiver Independent Exchange Format (RINEX)

The first proposal for the Receiver Independent Exchange Format (RINEX) was developed by the Astronomical Institute of the University of Berne for the easy exchange of the Global Positioning System (GPS) data to be collected during the first large European GPS campaign EUREF 89, which involved more than 60 GPS receivers of 4 different manufacturers. In August 2015, the CORS network had almost 2,000 stations, belonged by over 200 different organizations, and the network continues to expand.[4] And more than 40 percent of GNSS receivers are Galileo-ready.[5] Most GPS data software use a well-defined set of observables:

- The carrier-phase measurement at one or both carriers (actually being a measurement on the beat frequency between a generated reference frequency of the receiver and the received carrier of the satellite signal);
- The pseudorange (code) measurement, equivalent to the difference of the time of transmission (shown in the time frame of the satellite) and the time of reception (shown in the time frame of the receiver) of a distinct satellite signal;

[3] http://www.gpstk.org/doxygen/
[4] https://www.ngs.noaa.gov/CORS/
[5] http://gpsworld.com/40-of-gnss-receivers-are-galileo-ready/

- The observation time of the receiver clock reading at the instant of genuineness of the carrier-phase and/or the code measurements; Normally the software assumes that the observation time worths for both the phase and the code measurements, and for every satellites observed.

Consequently most of these programmes do not need tha major part of the information that is usually saved by the receivers: They need code, phase and time in the before mentioned definitions, and some station related information like antenna height, station name, etc. [8]

Observables files

Each RINEX file consists on a header section and a data section. The header section is placed at the beginning of the file and contains global information for the entire file. The header section contains the same number of header labels in the right side as the number of lines contained in the header section. These labels are a must and have to follow a protocol.
The format has been optimized to extend the minimum space possible and achieving the requirements independently of the number of different observation types of a specific receiver by indicating in the header the observation types to be stored. Allowing variable record lengths the observation records may then be kept as short as possible in computer systems.
Each Observation file basically contains the data from one site and one period. RINEX observables data files include three fundamental quantities that must be defined: Time, Phase, and Range.

Time

The measurement time is the receiver time of the received signals. It is identical for the phase and range measurements and is identical for all satellites observed at each epoch. It is expressed in GPS time (not Universal Time).

Pseudorange

The pseudorange (PR) is the distance from the receiver antenna to the satellite antenna having into account the receiver and satellite clock offsets:

$$PR = distance + (receiver\ clock\ offset - satellite\ clock\ offset) * c \tag{3.1}$$

where c is the light velocity.
The Equation 3.1 shows that the pseudorange reflects the real behavior of the receiver and satellite clocks. The pseudo-range is stored in units of meters.

Phase

The phase is the whole cycles in the carrier-phase measurement. The measured half-cycles by receivers need to be converted to whole cycles and flagged in the header section by the wavelength factor.
The phase, because of the negative Doppler, changes in the same sense as the range. The phase observations between epochs must be connected by including the integer number of cycles. The phase observations do not contain any systematic drifts from the reference oscillators intentional offsets. The observables are not corrected for external effects as satellite clock offsets, atmospheric refraction, etc.

If the converter software or the receiver adjusts the measurements using the real-time-derived receiver clock offsets dT(r), the coherence of the 3 quantities phase / pseudo-range / epoch must be maintained, i.e. the receiver clock correction must be applied to all 3 observables:

$$Time(corr) = Time(r) - dT(r) \tag{3.2}$$

$$PR(corr) = PR(r) - dT(r) * c \tag{3.3}$$

$$phase(corr) = phase(r) - dT(r) * freq \tag{3.4}$$

Versions

At the present time three major format versions have been developed:
The original RINEX Version 1 presented at and accepted by the 5th International Geodetic Symposium on Satellite Positioning in Las Cruces, 1989.; RINEX Version 2 presented at and accepted by the Second International Symposium of Precise Positioning with the Global Positioning system in Ottawa, 1990, mainly adding the possibility to include tracking data from different satellite systems (GLONASS, SBAS); RINEX Version 3, currently under revision. [8] [24]

System	Freq.Band	Frequency	RINEX 2-character Code			
			Ps.Range	Carr.Phase	Doppler	Sign.Strength
GPS	L1	1575.42	C1,P1	L1	D1	S1
	L2	1227.60	C2,P2	L2	D2	S2
	L5	1176.45	C5	L5	D5	S5
Glonass	G1	1602+k*9/16	C1,P1	L1	D1	S1
	G2	1246+k*7/16	C2,P2	L2	D2	S2
Galileo	E2-L1-E1	1575.42	C1	L1	D1	S1
	E5a	1176.45	C5	L5	D5	S5
	E5b	1207.140	C7	L7	D7	S7
	E5a+b	1191.795	C8	L8	D8	S8
	E6	1278.75	C6	L6	D6	S6
SBAS	L1	1575.42	C1	L1	D1	S1
	L5	1176.45	C5	L5	D5	S5

Figure 3.3: RINEX 2 observation data keys [7]

For each type of data, there is a key to call them. They are different in RINEX 2 and 3 and they include the system, the frequency and the type of data. E.g. In RINEX 2 if the data is the GPS carrier phase for the frequency L2, the key is called L2. As one can check in the Figure 3.3. For RINEX 3, the keys have three digits. The information also includes the channel or code. E.g. if the data is the pseudorange for the galileo frequency E5 with no data (pilot) it is called C5Q. As one can check in the Figure 3.4. Fortunately there is an easy conversion between RINEX 2 keys and RINEX 3 keys, but unfortunately not every RINEX 3 key has a match in RINEX 2. Thus, if is precise to work with both RINEX files, it is mandatory to use type of data which can be managed in both RINEX types.

GNSS System	Freq. Band /Frequency	Channel or Code	Observation Codes			
			Pseudo Range	Carrier Phase	Doppler	Signal Strength
Galileo	E1 / 1575.42	A PRS	C1A	L1A	D1A	S1A
		B I/NAV OS/CS/SoL	C1B	L1B	D1B	S1B
		C no data	C1C	L1C	D1C	S1C
		B+C	C1X	L1X	D1X	S1X
		A+B+C	C1Z	L1Z	D1Z	S1Z
	E5a / 1176.45	I F/NAV OS	C5I	L5I	D5I	S5I
		Q no data	C5Q	L5Q	D5Q	S5Q
		I+Q	C5X	L5X	D5X	S5X
	E5b / 1207.140	I I/NAV OS/CS/SoL	C7I	L7I	D7I	S7I
		Q no data	C7Q	L7Q	D7Q	S7Q
		I+Q	C7X	L7X	D7X	S7X
	E5(E5a+E5b) / 1191.795	I	C8I	L8I	D8I	S8I
		Q	C8Q	L8Q	D8Q	S8Q
		I+Q	C8X	L8X	D8X	S8X
	E6 / 1278.75	A PRS	C6A	L6A	D6A	S6A
		B C/NAV CS	C6B	L6B	D6B	S6B
		C no data	C6C	L6C	D6C	S6C
		B+C	C6X	L6X	D6X	S6X
		A+B+C	C6Z	L6Z	D6Z	S6Z

Figure 3.4: RINEX 3 observation data keys [8]

Galileo data in a RINEX file

For the development of this project, has been mandatory, check several times if different type of RINEX files have or not the appropriate data. In this case, the required data is from the Galileo system. As it is said above, the Galileo system data has the first letter as an 'E' and in the case of the satellites, in the Galileo system, the format is formed by the E followed by the satellite number.
Two steps are important to follow to check if a RINEX file has or not Galileo data:

- First, the RINEX observable file has to have the type system as mixed or Galileo. It appears in the RINEX header as the letter M for Mixed and E for Galileo. Not only that, then one has to check if the data type need is in the list of keys;

- Second, this can seem not necessary but nowadays, most of the files are provide with capability to have Galileo data, even they say in the headers that they have that data, but sadly, at the end there is no data with that information in the body. For that, it is important to check if really there is any satellite in the body providing the proper data.

As an example of what is said above, the Figure 3.5 is an example of a RINEX file, in this case, this a RINEX file version 3.02, which can manage Galileo data. It is shown how the Galileo data is announced in the header, where it is typed that there are many bands collected from Galileo system in the file. Unfortunately, that is not true, as it can be checked in the body. Due to the extension of the file, only the beginning of the body is printed. It can be checked for anybody who looks up the file called "TWTF00TWN_R_20170850000_01D_30S_MO.rnx"

in the FTP directory from NASA[6].

```
     3.02           OBSERVATION DATA    M                   RINEX VERSION / TYPE
ssrcrin-10.1.1x                         20170326 000000 LCL PGM / RUN BY / DATE
(0930225631113) Septentrio specific, please ignore.         COMMENT
TWTF                                                        MARKER NAME
Septentrio                                                  MARKER NUMBER
PolaRx4Pro                                                  MARKER TYPE
Calvin Lin          TL                                      OBSERVER / AGENCY
3008040             SEPT POLARX4        2.9.0               REC # / TYPE / VERS
CR620012101         ASH701945C_M    SCIS                    ANT # / TYPE
 -2994427.4093  4951307.9881  2674498.4821                  APPROX POSITION XYZ
        0.0000        0.0000        0.0000                  ANTENNA: DELTA H/E/N
G   18 C1C L1C D1C S1C C1W S1W C2W L2W D2W S2W C2L L2L D2L  SYS / # / OBS TYPES
       S2L C5Q L5Q D5Q S5Q                                  SYS / # / OBS TYPES
E   16 C1C L1C D1C S1C C5Q L5Q D5Q S5Q C7Q L7Q D7Q S7Q C8Q SYS / # / OBS TYPES
       L8Q D8Q S8Q                                          SYS / # / OBS TYPES
S    8 C1C L1C D1C S1C C5I L5I D5I S5I                      SYS / # / OBS TYPES
R   16 C1C L1C D1C S1C C2P L2P D2P S2P C2C L2C D2C S2C C3Q SYS / # / OBS TYPES
       L3Q D3Q S3Q                                          SYS / # / OBS TYPES
C    8 C1I L1I D1I S1I C7I L7I D7I S7I                      SYS / # / OBS TYPES
J   12 C1C L1C D1C S1C C2L L2L D2L S2L C5Q L5Q D5Q S5Q     SYS / # / OBS TYPES
SEPTENTRIO RECEIVERS OUTPUT ALIGNED CARRIER PHASES.         COMMENT
NO FURTHER PHASE SHIFT APPLIED IN THE RINEX ENCODER.        COMMENT
G 1C                                                        SYS / PHASE SHIFT
G 2W                                                        SYS / PHASE SHIFT
G 2L   0.00000                                              SYS / PHASE SHIFT
G 5Q   0.00000                                              SYS / PHASE SHIFT
E 1C   0.00000                                              SYS / PHASE SHIFT
E 5Q   0.00000                                              SYS / PHASE SHIFT
E 7Q   0.00000                                              SYS / PHASE SHIFT
E 8Q   0.00000                                              SYS / PHASE SHIFT
S 1C                                                        SYS / PHASE SHIFT
S 5I                                                        SYS / PHASE SHIFT
R 1C                                                        SYS / PHASE SHIFT
R 2P   0.00000                                              SYS / PHASE SHIFT
R 2C                                                        SYS / PHASE SHIFT
R 3Q   0.00000                                              SYS / PHASE SHIFT
C 1I                                                        SYS / PHASE SHIFT
C 7I                                                        SYS / PHASE SHIFT
J 1C                                                        SYS / PHASE SHIFT
J 2L   0.00000                                              SYS / PHASE SHIFT
J 5Q   0.00000                                              SYS / PHASE SHIFT
  2017     3    26     0     0    0.0000000     GPS         TIME OF FIRST OBS
 C1C    0.000 C2C    0.000 C2P    0.000                     GLONASS COD/PHS/BIS
DBHZ                                                        SIGNAL STRENGTH UNIT
                                                            END OF HEADER

> 2017 03 26 00 00  0.0000000  0 20
G21  21669376.779 7 113873326.89907      -713.639 7        46.825    21669375.957 6
        36.843    21669376.360 6  88732452.58106      -556.077 6        36.843
G15  22481923.570 7 118143307.46707     -1666.382 7        47.378    22481923.610 5
        34.109    22481923.938 5  92059719.12905     -1298.474 5        34.109    2
2481923.958 6  92059718.11006     -1298.449 6        40.403
G10  23254511.190 6 122203280.36506      3004.569 6        40.982    23254511.097 4
        26.707    23254513.690 4  95223346.29904      2341.223 4        26.707    2
3254512.927 6  95223350.28906      2341.211 6        38.622
R14  21682656.596 6 115580748.41206     -3299.492 6        38.348    21682666.705 6
  89896169.53206     -2566.184 6        37.849    21682666.817 6  89896174.52806
  -2566.205 6        38.366
G20  21447843.544 7 112709193.59007     -1989.772 7        46.674    21447842.686 6
        37.115    21447844.149 6  87825350.08506     -1550.472 6        37.115
S37  37078207.587 6 194847278.48106         2.522 6        40.615
R16  20943830.326 6 111878191.59306      3666.084 6        40.274    20943837.530 6
  87016408.24906      2851.400 6        39.763    20943836.322 6  87016411.24706
   2851.490 6        39.982
S26  39596407.050 5 208080776.61005       -25.458 5        33.449
G32  24320199.884 6 127803505.04906      1780.790 6        37.425    24320200.107 3
        20.976    24320202.728 3  99587158.26303      1387.613 3        20.976    2
4320203.543 6  99587164.27106      1387.641 6        36.650
R15  19538877.740 7 104409824.12307      -709.930 7        44.939    19538887.214 8
  81207676.06608      -552.170 8        48.431    19538888.290 7  81207679.07607
```

(a) Header and beginning of body

```
  -552.144 7        47.743
S40  37137303.938 7 195157809.62307       360.554 7        45.079
S29  37078185.462 6 194847103.06306         2.605 6        39.168
S28  37931741.569 7 199332611.62707         1.022 7        43.536
G24  20496111.479 8 107707783.23508     -1470.365 8        51.563    20496111.368 7
        46.392    20496114.435 7  83928152.96707     -1145.741 7        46.392    2
0496114.633 8  83928152.96008     -1145.715 8        48.474
G25  24371906.857 5 128075239.98905      3246.349 5        35.062    24371906.361 2
        16.377    24371911.662 2  99798911.78202      2529.612 2        16.377    2
4371912.408 5  99798912.77205      2529.668 5        33.228
R20  23121489.893 5 123640944.40505     -2462.975 5        30.475    23121499.993 6
  96165224.41106     -1916.066 6        38.264    23121500.591 6  96165224.44106
  -1916.103 6        37.792
R04  21174436.285 6 113388129.33106      -698.373 6        41.354    21174443.175 7
  88190798.09607      -543.192 7        43.721    21174442.835 7  88190797.10007
   -543.292 7        43.094
G12  22738358.465 7 119490881.12907      2448.103 7        44.683    22738358.367 5
        30.725    22738360.018 5  93109784.60705      1907.612 5        30.725    2
2738360.849 6  93109784.60806      1907.632 6        39.559
G18  21345519.138 8 112171456.21408      2125.203 8        48.006    21345518.490 6
        38.150    21345518.561 6  87406311.05506      1656.005 6        38.150
R05  20756283.959 6 110954212.87706      1596.632 6        39.550    20756291.162 6
  86297754.37706      1241.735 6        40.300    20756290.776 6  86297760.37106
   1241.924 6        40.498
> 2017 03 26 00 00 30.0000000  0 20
G21  21673495.942 7 113894974.61707      -729.591 7        47.260    21673495.120 6
        36.786    21673495.722 6  88749320.90606      -568.510 6        36.786
G15  22491439.730 7 118193315.84807     -1667.576 7        47.705    22491439.761 5
        34.105    22491440.025 5  92098686.64805     -1299.405 5        34.105    2
2491440.459 6  92098685.63706     -1299.302 6        40.119
G10  23237362.351 6 122113166.45606      3003.001 6        41.399    23237362.258 4
        26.742    23237365.483 4  95153127.66504      2339.996 4        26.742    2
3237365.169 6  95153131.65606      2340.042 6        39.027
R14  21701269.767 6 115679971.56106     -3315.335 6        38.838    21701280.692 6
  89973343.03406     -2578.560 6        38.850    21701281.163 6  89973348.02706
  -2578.677 6        38.267
G20  21459238.629 7 112769076.03907     -2002.326 7        46.640    21459237.771 6
        36.904    21459239.247 6  87872011.69406     -1560.253 6        36.904
S37  37078191.293 6 194847201.71406         2.352 6        40.894
R16  20923283.126 6 111768434.64406      3650.959 6        40.222    20923290.648 6
  86931041.75006      2839.633 6        40.291    20923291.220 6  86931044.75406
   2839.668 6        40.215
S26  39596548.349 5 208081532.03705       -25.360 5        33.503
G32  24310040.152 6 127750115.80606      1778.574 6        38.625    24310040.393 3
        21.595    24310043.323 3  99545556.25703      1385.901 3        21.595    2
4310043.596 6  99545562.25806      1385.836 6        37.414
R15  19542940.641 7 104431534.37207      -737.460 7        45.115    19542949.965 8
  81224561.80108      -573.565 8        49.150    19542950.472 7  81224564.81607
  -573.563 7        47.767
S40  37136801.949 7 195155182.08507       360.583 7        44.743
S29  37078172.803 6 194847024.88306         2.130 6        39.264
S28  37931731.955 7 199332575.73207         1.047 7        43.369
G24  20504549.640 8 107752124.18008     -1485.773 8        51.579    20504549.533 7
        46.496    20504552.503 7  83962704.33407     -1157.742 7        46.496    2
0504552.328 8  83962704.32808     -1157.733 8        48.923
G25  24353367.178 5 127977807.93805      3249.189 5        35.452    24353366.727 2
        16.717    24353370.938 2  99722990.69302      2531.825 2        16.717    2
4353371.917 5  99722991.68405      2531.579 5        32.964
R20  23135336.107 5 123714994.24305     -2473.884 5        30.226    23135347.857 6
  96222818.64006     -1923.734 6        39.147    23135348.584 6  96222818.65506
  -1923.723 6        38.303
R04  21178375.790 6 113409223.85206      -707.939 6        41.418    21178382.500 7
  88207204.89907      -550.649 7        44.295    21178382.635 7  88207203.90007
   -550.628 7        43.615
G12  22724405.847 7 119417557.31807      2440.106 7        45.558    22724405.745 5
        30.823    22724407.344 5  93052649.16305      1901.382 5        30.823    2
2724407.385 6  93052649.15906      1901.387 6        40.181
G18  21333405.695 8 112107799.06908      2118.546 8        48.201    21333405.047 6
        38.363    21333405.012 6  87356708.07506      1650.817 6        38.363
```

(b) Continuation of the body (not the end of the body)

Figure 3.5: Example of RINEX file which has Galileo keys in the data type list in the header but that data does not appear in the body.

The reason why it is said that there is no Galileo data in the body is because the Galileo data is provided by the Galileo satellites, these are named as the letter 'E' plus the satellite number and in the Figure 3.5, every new data line begin with a 'G' which mean GPS satellite. There is no line beginning with the letter 'E'. As an example of a good RINEX file, fitable for this project is the one shown in the Figure 3.6. In this, it is shown the keys in the data type list, iqual to the RINEX file in the Figure 3.5, but in this one, one can find the Galileo satellites providing the proper data in the body (E02, E11, E24, E12 and E08 in the first epoch of the day 85 of the year 2017). This RINEX file is named as OZST00USA_R_20170920000_01D_15S_MO.rnx

[6] https://cddis.nasa.gov/Data_and_Derived_Products/GNSS/RINEX_Version_3.html

```
|     3.03           OBSERVATION DATA    M                   RINEX VERSION / TYPE
sbf2rin-11.3.3                          20170403 003513 LCL PGM / RUN BY / DATE
ozst                                                        MARKER NAME
ozst                                                        MARKER NUMBER
DDBM                                                        MARKER TYPE
UNAVCO              UNAVCO                                  OBSERVER / AGENCY
3012360             SEPT POLARX5        5.1.0               REC # / TYPE / VERS
0220356924          TRM59800.80     SCIT                    ANT # / TYPE
 -2574193.3327 -4577156.7783  3609641.2405                  APPROX POSITION XYZ
        0.0083          0.0000          0.0000              ANTENNA: DELTA H/E/N
G   14 C1C L1C S1C C1W S1W C2W L2W S2W C2L L2L S2L C5Q L5Q  SYS / # / OBS TYPES
       S5Q                                                  SYS / # / OBS TYPES
E   15 C1C L1C S1C C6C L6C S6C C5Q L5Q S5Q C7Q L7Q S7Q C8Q  SYS / # / OBS TYPES
       L8Q S8Q                                              SYS / # / OBS TYPES
S    6 C1C L1C S1C C5I L5I S5I                              SYS / # / OBS TYPES
R    6 C1C L1C S1C C2C L2C S2C                              SYS / # / OBS TYPES
C    9 C2I L2I S2I C7I L7I S7I C6I L6I S6I                  SYS / # / OBS TYPES
SEPTENTRIO RECEIVERS OUTPUT ALIGNED CARRIER PHASES.         COMMENT
NO FURTHER PHASE SHIFT APPLIED IN THE RINEX ENCODER.        COMMENT
G L1C                                                       SYS / PHASE SHIFT
G L2W                                                       SYS / PHASE SHIFT
G L2L  0.00000                                              SYS / PHASE SHIFT
G L5Q  0.00000                                              SYS / PHASE SHIFT
E L1C  0.00000                                              SYS / PHASE SHIFT
E L6C  0.00000                                              SYS / PHASE SHIFT
E L5Q  0.00000                                              SYS / PHASE SHIFT
E L7Q  0.00000                                              SYS / PHASE SHIFT
E L8Q  0.00000                                              SYS / PHASE SHIFT
S L1C                                                       SYS / PHASE SHIFT
S L5I                                                       SYS / PHASE SHIFT
R L1C                                                       SYS / PHASE SHIFT
R L2C                                                       SYS / PHASE SHIFT
C L2I                                                       SYS / PHASE SHIFT
C L7I                                                       SYS / PHASE SHIFT
C L6I                                                       SYS / PHASE SHIFT
    15.000                                                  INTERVAL
  2017     4     2     0     0    0.0000000     GPS         TIME OF FIRST OBS
  2017     4     2    23    59   45.0000000     GPS         TIME OF LAST OBS
    78                                                      # OF SATELLITES
 C1C    0.000 C2C    0.000                                  GLONASS COD/PHS/BIS
DBHZ                                                        SIGNAL STRENGTH UNIT
 24 R01  1 R02 -4 R03  5 R04  6 R05  1 R06 -4 R07  5 R08  6 GLONASS SLOT / FRQ #
    R09 -2 R10 -7 R11  0 R12 -1 R13 -2 R14 -7 R15  0 R16 -1 GLONASS SLOT / FRQ #
    R17  4 R18 -3 R19  3 R20  2 R21  4 R22 -3 R23  3 R24  2 GLONASS SLOT / FRQ #
                                                            END OF HEADER

> 2017 04 02 00 00  0.0000000  0 26
G11  24262422.652 6 127499945.13206        36.750    24262421.950 2        16.750
24262420.622 2  99350625.42702        16.750
G24  25015190.974 5 131455723.38605        35.500    25015190.398 3        20.000
25015192.597 3 102433022.51703        20.000    25015192.910 6 102433009.53006
38.250    25015194.255 7  98165006.68907        43.250
G17  20504753.795 9 107753282.90809        54.500    20504753.313 8        51.250
20504751.335 8  83963617.47808        51.250    20504751.429 8  83963610.48108
50.500
G19  20720882.865 8 108889049.21608        49.750    20720881.502 7        45.750
20720879.104 7  84848618.70307        45.750
E02  26753722.524 7 140591790.68207        42.250    26753723.276 7
114116723.08207        44.750    26753725.480 7 104987400.57007        43.000
26753722.665 7 107726186.56807        44.500    26753723.400 7 106356796.64307
46.750
G28  21703672.855 8 114053637.05208        48.750    21703671.845 6        40.750
21703670.493 6  88872986.69506        40.750
R04  24074149.448 6 128916001.79606        39.250    24074155.932 6
100268037.48706        39.250
S33  37291786.380 7 195969863.00007        44.000    37291765.095 7
146341135.26107        46.250
G01  23466051.340 7 123314985.21807        42.250    23466050.916 5        30.000
23466053.316 5  96089631.62105        30.000    23466053.562 6  96089625.62806
41.500    23466052.899 7  92085901.98707        45.500
```

(a) Header and beginning of body

```
E12  25159299.744 7 132213086.04407        43.250    25159297.036 7
107315835.72507        46.250    25159300.161 7  98730595.93007        44.500
25159297.556 7 101306162.30807        45.250    25159298.251 8 100018383.69308
48.000
G30  21520182.209 8 113089362.45308        48.250    21520181.485 6        41.250
21520182.767 6  88121584.44606        41.250    21520183.281 7  88121596.46007
46.250    21520183.741 8  84449873.62608        52.000
E24  24709702.109 8 129850392.38708        48.750    24709694.665 8
105398036.05608        51.250    24709716.335 8  96966282.54208        49.500
24709712.920 8  99495820.64808        50.500    24709714.361 8  98231051.16908
53.000
S31  37095117.981 8 194935277.88608        49.000    37095087.682 8
145569468.31408        52.500
G15  23283092.736 7 122353490.53107        45.500    23283092.415 5        32.250
23283091.394 5  95340396.18905        32.250    23283092.064 6  95340399.19406
40.000
R11  23182685.883 7 123881243.38807        43.500    23182689.922 6
96352108.23406        39.500
G13  21562479.913 8 113311631.52508        48.500    21562479.408 6        39.000
21562478.008 6  88294787.53506        39.000
E11  23225409.253 8 122050382.87708        48.750    23225406.012 8
99066887.85908        51.750    23225408.691 8  91141566.67408        49.000
23225405.934 8  93519167.01208        50.250    23225406.950 8  92330368.41708
52.750
E08  26023004.881 6 136751823.91106        36.750    26023007.815 6
110999859.93106        38.500    26023008.410 6 102119872.77206        36.500
26023004.977 6 104783863.60006        36.750    26023005.958 6 103451864.76106
39.500
R12  20024935.049 8 106969692.22508        51.750
C06  41059221.152 6 213806136.87806        38.500    41059221.024 5
165328463.36005        35.250    41059219.961 5 173734997.75905        35.750
S38  37216669.636 8 195574875.49008        48.750    37216645.423 8
146046673.18608        53.250
R23  19824026.461 8 106045245.54708        53.750    19824032.175 7
82479661.27007        45.500
R03  23674987.141 7 126734034.57207        43.500    23674993.776 6
98571032.11206        39.750
R22  20051311.598 8 107035391.76608        52.750    20051315.732 8
83249737.31708        50.500
S35  37253005.200 8 195765842.22008        48.250    37252982.042 8
146188669.32108        51.000
R13  20953283.935 7 111889402.49307        45.500    20953288.110 6
87025212.25006        41.000
> 2017 04 02 00 00 15.0000000  0 26
G11  24270483.050 6 127542302.58106        36.750    24270482.357 3        18.000
24270481.288 3  99383631.20503        18.000
G24  25005471.133 6 131404649.92706        36.000    25005470.557 3        19.750
25005473.579 3 102393225.07703        19.750    25005473.257 6 102393212.06006
37.000    25005475.371 7  98126867.47507        44.000
G17  20504658.226 9 107752780.97709        54.250    20504657.744 8        51.250
20504655.824 8  83963226.36908        51.250    20504656.128 8  83963219.37308
50.500
G19  20715271.819 8 108859563.14008        50.000    20715270.456 7        45.750
20715268.103 7  84825642.55407        45.750
E02  26747766.171 7 140560492.22607        42.000    26747767.361 7
114091318.51607        44.500    26747769.686 7 104964028.38407        43.000
26747766.746 7 107702204.66307        44.500    26747767.463 7 106333119.59807
46.750
G28  21706912.075 8 114070658.98908        48.750    21706911.060 6        40.500
21706909.660 6  88886250.53906        40.500
R04  24074151.337 6 128916008.99406        39.750    24074155.868 6
100268043.05806        38.250
S33  37291812.568 7 195969993.58507        44.250    37291789.862 7
146341232.83407        46.250
G01  23469163.874 6 123331344.58206        41.750    23469163.441 4        29.750
23469166.570 4  96102379.18104        29.750    23469166.034 6  96102373.20106
41.000    23469166.028 7  92098118.40707        45.500
E12  25164181.519 7 132238739.85207        43.500    25164178.829 7
107336658.63107        46.500    25164181.811 7  98749752.99307        44.500
```

(b) Continuation of the body (not the end of the body)

Figure 3.6: Example of RINEX file which has Galileo keys in the data type list in the header and that data actually appears in the body.

3.3.2 Standard Product 3

The first Standard Product 3 format (SP3-a) was proposed in 1989, with the main purpose of exchanging satellite related data (orbit and clock information). The basic format of an SP3 file is a header, following by a series of records containing the position and clock records for each satellite listed in the header. A second, optional, record contains the satellite velocity and clock correction rate-of-change.

Three major SP3 versions are defined:

- Original SP3-a proposed in 1989;
- SP3 version b, proposed in 1998, defined to allow the combination of GPS orbits and GLONASS orbits;
- And the current SP3 version c, proposed in 2000. [24]

One can find deeper information of the format and how it is composed in the NASA web page [7]. As it is said about the problem of finding Galileo data in the RINEX observable files, there is a similar problem in the SP3 files. Some files have Galileo keys in the satellite list in the header, but then there is no data about Galileo

[7] https://igscb.jpl.nasa.gov/igscb/data/format/sp3_docu.txt

```
#cP2017  3 29  0  0  0.00000000      96 TRACK IGb14 FIT GRGS
## 1942 259200.00000000   900.00000000 57841 0.0000000000000
+   65   E01E02E08E09E11E12E14E18E19E22E24E26E30R01R02R03R04
+        R05R07R08R10R11R13R14R15R16R17R18R19R20R21R22R23R24
+        G01G02G03G05G06G07G08G09G10G11G12G13G14G15G16G17G18
+        G19G20G21G22G23G24G25G26G27G28G29G30G31G32  0  0  0
+          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
++         6  6  5  6  7  6 10  6  7  7  6  6  6  6  5  5  6
++         6  5  5  6  6  5  6  6  6  5  6  6  6  6  6  6  6
++         5  4  6  5  5  5  5  5  5  4  6  5  5  5  6  5  5
++         5  5  5  5  5  5  5  5  5  6  6  5  5  5  0  0  0
++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
%c M  cc GPS ccc cccc cccc cccc cccc cccc cccc cccc cccc cccc
%c cc cc ccc ccc cccc cccc cccc cccc cccc cccc cccc cccc cccc
%f  0.0000000  0.000000000  0.00000000000  0.000000000000000
%f  0.0000000  0.000000000  0.00000000000  0.000000000000000
%i    0    0    0    0      0      0      0      0         0
%i    0    0    0    0      0      0      0      0         0
/* CNES/CLS/GRGS - TOULOUSE,FRANCE - Contact : igs-ac@cls.fr
/* PCV:IGS14_1941 OL/AL:FES2012  NONE     NN ORB:CoN CLK:CoN
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
*  2017  3 29  0  0  0.00000000
PE01   1056.802812  20835.742632 -21001.292437    -30.877402
PE02  -1049.429146 -20840.162924  20995.858203     -0.960878
PE08 -27639.159607  -8276.755219  -6615.274991   6872.936167
PE09  26698.035441  -4872.076625 -11804.902175   7161.672765
PE11  15760.777035  -5475.762784  24454.542188   1687.086040
PE12  18942.995798  14662.088384  17413.706080   1154.782067
PE14 -21614.159796 -13976.277331  -4312.810103   6813.844567
PE18  17990.028113  24063.992709  -7103.059413   7231.184192
PE19  10423.090705  16939.910776  21928.718397      8.225880
PE22 -11597.426933 -27227.177203   -602.736169   4322.642174
PE24  22134.565473 -14736.115745 -13038.359681    348.390079
PE26  -3173.325382  23737.759916 -17388.226105   7492.843426
PE30 -16556.838202  -4204.260955  24160.192700   6940.955118
PR01   9154.351244  10332.706020  21448.277611      9.346562
PR02  23937.199104   3952.864138   7985.708696    242.476585
PR03  23201.249884  -4430.535661  -9639.745594     96.272472
PR04   8451.190471 -10303.892905 -21737.990692    209.379850
PR05 -10504.498969 -10081.636015 -20953.705186    -92.452718
PR07 -22981.419155   4612.166179  10057.864865      1.677646
PR08  -9043.539417  10220.533100  21601.495818    -26.826133
PR10 -16015.068255  14835.996915  13230.340067     17.406963
PR11 -11507.062040  22440.565772  -3942.081214      2.259411
PR13  11717.187715    795.741638 -22666.462116      1.665466
PR14  15935.586677 -16014.865071 -11798.881067    467.739691
PR15  12527.781603 -22225.419989   1619.676501      4.114266
PR16      5.226032 -17459.301199  18618.903483     61.056092
PR17  10887.810507  -2452.610749  22948.322513     33.139232
PR18    325.245483 -17209.821185  18812.826504    144.739682
PR19 -10585.891403 -22963.079811   3337.267973   -177.693916
PR20 -15207.516898 -14395.845859 -14510.691281   -282.837887
PR21 -10867.128090   2316.342872 -22956.036711    108.901363
PR22    663.107379  18145.062772 -17974.742060    232.033860
PR23  11914.033200  22528.224059  -1225.865176     77.598728
PR24  15496.253475  14273.600061  14377.475208   -148.671440
PG01  13468.180229 -15714.948147  16388.180012     52.281242
PG02 -14227.390967  -8017.994344 -20506.857576    450.103073
PG03  22329.595958 -13372.180149  -5237.243121   -105.949605
PG05 -23125.831258   -302.341474 -13321.736968    -53.429385
PG06  -7413.462989 -21059.824505 -14352.466091    330.678553
PG07   7221.007677 -24775.126494  -5078.259865    368.299314
PG08  20725.303439   1016.713004  16687.809044    -54.376061
PG09   1990.137991 -16694.765337 -20586.132203    331.563368
PG10   5046.792791  14399.425437  21761.443546    -86.582386
PG11  13521.926335 -10290.651951  19860.647953   -677.787584
PG12 -24388.958696  10045.963230  -1972.544699    386.350038
PG13 -21707.838824  -9747.638933  11621.595378    -75.759745
```

(a) Header and beginning of body

```
PG14  18240.849990  19190.626137   3285.936211    -64.683114
PG15 -19895.094235   2546.462336  17421.802944   -346.649378
PG16  22747.180678   2086.722199 -14015.893919     30.046966
PG17 -12378.412270 -20622.825035  11793.398128   -186.220859
PG18  -5392.864031  18324.073111  18950.515575    607.353833
PG19 -17377.855283 -19453.325805   3816.069257   -508.983135
PG20 -23231.572460  12859.511474    431.133766    466.014510
PG21  -1221.686353  26443.423471   1503.485938   -515.143871
PG22  24867.225930  -8884.062268   3449.875935     58.661877
PG23  14470.636724  -7865.746089 -20687.364822   -208.620630
PG24 -14238.821142  13202.433016  17930.236663    -32.580844
PG25 -15233.709750  16611.604751 -14014.823727   -360.367419
PG26  15299.082063   8600.780257 -19961.666729   -516.678143
PG27  23061.036778  11538.430441   6769.978794    231.639190
PG28   -506.944637 -15300.285167  22336.241155    600.589261
PG29  -5467.171318  14389.198607 -21649.906481    616.883672
PG30  -1035.138839 -25260.633122   7925.507803    159.739268
PG31   7903.967329  19599.172271 -15768.776308    210.646632
PG32  13637.459291  19923.919789  11130.165690   -367.539481
*  2017  3 29  0 15  0.00000000
PE01     63.031033  22371.703891 -19385.213888    -30.886258
PE02    -55.994959 -22376.637081  19379.669504     -0.961473
PE08 -26933.225239  -8165.413192  -9170.104274   6872.927126
PE09  27654.976493  -4819.690644  -9374.543419   7161.666387
PE11  16678.468750  -3478.415421  24212.012750   1687.095735
PE12  19612.899974  15983.610271  15392.250977   1154.857957
PE14 -21081.887309 -14402.780483  -1491.794539   6813.838526
PE18  17808.579554  23885.340568  -9377.383519   7231.157219
PE19   8409.681962  16716.212476  22940.680025      8.226222
PE22 -11635.490033 -27135.034608   2117.702980   4322.835331
PE24  21285.131455 -13770.021989 -15311.021386    348.749196
PE26  -3806.787249  22209.620429 -19186.240926   7492.803117
PE30 -15239.128871  -5979.586715  24648.915358   6940.927350
PR01   6451.518179  10395.066649  22382.378541      9.346845
PR02  22774.258092   3841.716919  10913.383382    242.477647
PR03  24206.491253  -4589.854070  -6641.371720     96.273520
PR04  11056.233960 -10430.543116 -20474.011855    209.381256
PR05  -7853.704121 -10138.135529 -22058.339429    -92.456689
PR07 -24031.281441   4768.439708   7078.237342      1.676770
PR08 -11612.884319  10345.312486  20269.797836    -26.826368
PR10 -15002.375546  13349.411673  15749.524933     17.406574
PR11 -11519.425646  22767.266940   -748.428186      2.258730
PR13  10356.156259   3265.558094 -23104.138574      1.665449
PR14  15007.166224 -14677.259538 -14466.439706    467.744851
PR15  12351.327432 -22324.539946  -1607.677692      4.114156
PR16    948.027133 -19418.306564  16552.020859     61.056561
PR17  12348.528720    -73.186770  22328.907295     33.141582
PR18   1497.892586 -15083.318357  20512.100862    144.741125
PR19 -10445.354662 -22343.137101   6497.730561   -177.694098
PR20 -16159.459839 -15730.982408 -11849.676988   -282.839253
PR21 -12340.853973    -60.789217 -22320.554378    108.902679
PR22   -438.186190  16086.518496 -19832.680407    232.035079
PR23  11894.644796  22132.560382  -4431.928854     77.600710
PR24  16461.090683  15576.330430  11710.612160   -148.671739
PG01  13434.927854 -13706.643529  18138.183885     52.281982
PG02 -14028.743745 -10405.412611 -19602.903530    450.096663
PG03  22834.601595 -13323.415308  -2426.218980   -105.948757
PG05 -21746.868565  -1191.153125 -15422.465580    -53.427873
PG06  -7087.475931 -22551.839791 -12072.124303    330.682976
PG07   7405.731720 -23982.028801  -7831.377242    368.295339
PG08  22099.625068   2162.227653  14697.049888    -54.376577
PG09   3997.670271 -15390.103718 -21300.422132    331.570262
PG10   2794.721864  15411.542479  21462.022982    -86.576518
PG11  14320.014104  -7984.759227  20338.568787   -677.789907
PG12 -24496.804797   9885.576301    952.339225    386.349388
PG13 -22464.122132 -10704.413181   9089.290774    -75.760972
PG14  17706.968870  19046.803410   6069.646355    -64.684864
PG15 -21432.745994   1381.921457  15691.457070   -346.650334
PG16  21200.526518   2766.125201 -16142.126684     30.047767
```

(b) Continuation of the body (not the end of the body)

Figure 3.7: Example of SP3 file which has Galileo satellite keys in the satellite list in the header and that data actually appears in the body.

satellites in the body.

The Figure 3.7 is an example of a SP3 file which has the Galileo satellite keys in the satellite list in the header and also in the body. One can check that there are several Galileo satellites providing data in the body, concretely the satellites E01, E02, E08, E09, E11, E12, E14, E18, E19, E22, E24, E26 and E30 provide their position and clock data for the first epoch of the week number 1942 and the third day of the week. For that this name of the SP3 is grm19423.sp3.

3.4 Receiver stations

For developing this project, it is required to find receiver stations with some special features:

- Every station chosen has to be located close to the others. There is no a specific numerical limit but they should have the same satellites in view at the same time;
- Every station has to be configured to manage Galileo data and to add it to their RINEX observable files;
- That data, stored in RINEX version 3 files, have to be uploaded online, if not it is not available and testable for everybody.

With the features mentioned above, It is found seven receiver stations in the GPS/GNSS FTP Server Layout provided by UNAVCO from its main webpage. Then one can search mostly every receiver position in the National Oceanic and Admospheric Administration main webpage [2]. It is shown the coordinates for these 7 receiver stations in table 3.2.
Randomly, 5 of them are selected to developing the tool. It is considered that 5 receiver stations are more than enough to have accurate results. Using 7 stations just would increase the computational cost. For testing, the tool can accept the other station data files as input arguments. In 3.8 one can see the position of this stations in a fragment of the world map.

Station ID	X(m)	Y(m)	Z(m)
ISLK	-2,473,741.810	-4,560,945.587	3,698,290.699
OZST	-2,574,193.153	-4,577,158.410	3,609,640.577
P513	-2,669,568.304	-4,504,987.875	3,629,597.378
P558	-2,501,017.874	-4,584,974.606	3,651,264.864
P588	-2,402,808.890	-4,661,826.080	3,618,747.882
ACSX	-2,460,183.687	-4,737,087.405	3,479,422.780
CAT3	-2,540,623.180	-4,682,556.597	3,495,319.263

Table 3.2: Receiver station positions [2]

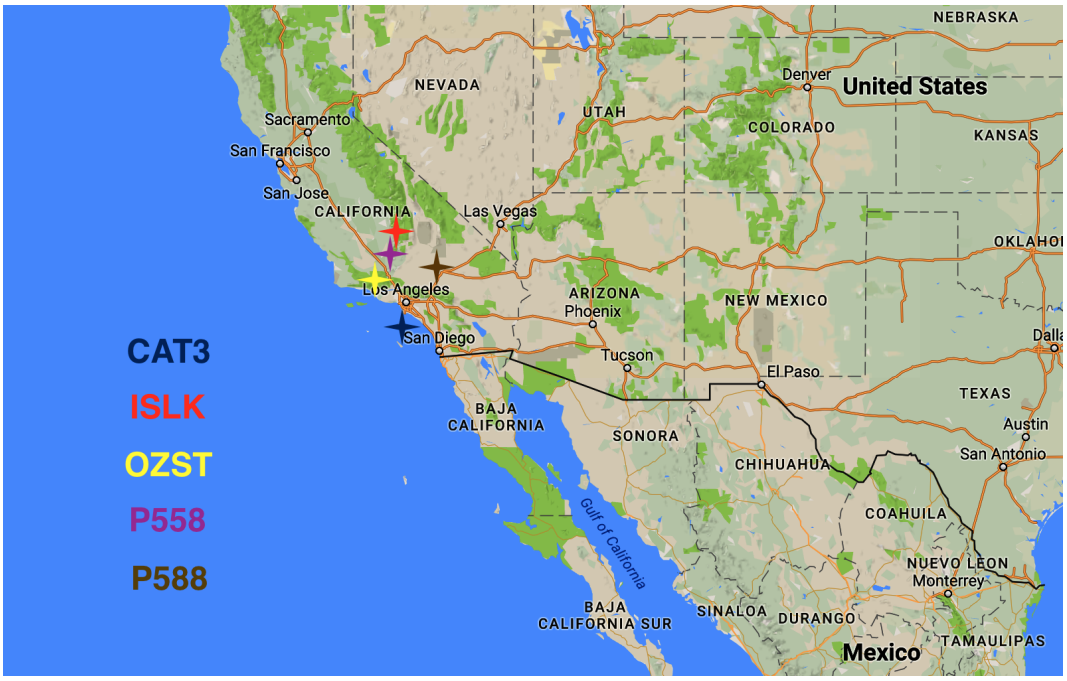

Figure 3.8: Map of the stations used in this project [2]

Also, in Figure [2] there is deeper information about every available receiver station. Clicking in each one, it is possible to access to photographs, site log, even data files as RINEX (The RINEX observables files are not collected from this FTP server because they only have GPS data and neither Galileo nor Glonass data).

(a) ISLK

(b) OZST

(c) P558

(d) P588

(e) CAT3

Figure 3.9: Emplacement of the Receiver Stations used in this project (ISLK, OZST, P558, P588 and CAT3). [2]

3.5 Multipath

3.5.1 Linear combination of observables

Combining the equations of the observables of carrier phase and code measurements, allows to avoid terms as the ionospheric and geometric terms [25]. Then it is possible to focus on the main terms of this project. The pseudorange measurements can be modeled as:

$$P_i = s + I_i + e_{MPi} \tag{3.5}$$

$$P_j = s + I_j + e_{MPj} \tag{3.6}$$

Where the P represents any GNSS code measurement, s is the term which includes the geometric ranges and the difference of clock offsets, I is the term due to the ionospheric delay and e_{MPi} represents the effect of multipath and noise in the code measurement. And the carrier phase measurements are represented as:

$$\Phi_i = s - I_i + A_i + \epsilon_{MPi} \tag{3.7}$$

$$\Phi_j = s - I_j + A_j + \epsilon_{MPj} \tag{3.8}$$

whereΦ is the measured carrier phase, A is the ambiguity term and ϵ_{MP} represents the carrier phase multipath and noise. Subtracting Φ_i from P_i to find MP_i,
where $MP_i = e_{MPi} - \epsilon_{MPi}$

$$P_i - \Phi_i = 2I_i - A_i + MP_i \tag{3.9}$$

$$MP_i - A_i = P_i - \Phi_i - 2I_i \tag{3.10}$$

The following steps are required to avoid the ionospheric term Subtracting Φ_j from Φ_i to solve for I_i:

$$\Phi_i - \Phi_j = I_j - I_i + A_i - A_j \tag{3.11}$$

The Equations 3.9 and 3.11 are geometric-free combinations because the s term is removed with the geometric range inside. The ionospheric delay is proportional to the Ionospheric Electron Content (IEC) and depends on the signal wavelength.

$$I_i = \frac{A}{f_i^2} IEC, I_j = \frac{A}{f_j^2} IEC \tag{3.12}$$

$$\frac{I_j}{I_i} = \frac{f_i^2}{f_j^2} = \alpha \tag{3.13}$$

Where f is frequency. With the constant $A = 40.3m^3s^{-2}$.
Substituting $I_j = \alpha I_i$ into Equation 3.11:

$$\Phi_i - \Phi_j = I_i(\alpha - 1) + A_i - A_j \tag{3.14}$$

$$I_i = \frac{\Phi_i - \Phi_j + A_j - A_i}{\alpha - 1} \tag{3.15}$$

Where I_i is a function of the carrier phases, the ambiguities and the frequencies.

3.5.2 Assessing Multipath

As one can see above, the multipath can be obtained as a function of the other terms of the Equation 3.10, and as the ionospheric effect is not interesting in this project, this can be removed. Substituting 3.15 into 3.10:

$$MP_i - \left\langle \frac{2}{\alpha - 1}(Aj - Ai) + A_i \right\rangle = P_i - \Phi_i - \frac{2}{\alpha - 1}(\Phi_i - \Phi_j) \tag{3.16}$$

After processing MP_i for having a zero mean over elevation, which can be checked in Section 3.5.4, the constant term in each arc $\frac{2}{\alpha-1}(Aj - Ai) + A_i$ can be removed by finding the average over a given orbital arc and then subtracted from the calculated MP_i for each epoch.

$$MP_i = P_i - \Phi_i - \frac{2}{\alpha - 1}(\Phi_i - \Phi_j) \tag{3.17}$$

Where MP_i is the estimation of the code multipath error on a pseudorange P_i, while Φ_i and Φ_j are the carrier phase observables (in units of length) for a α which depends on the frequencies f_i and f_j. j represents any band which is different than i. With multi-frequency Galileo signals, several values of j are possible, but the particular selection of j does not significantly affect the results. [26] [25]

3.5.3 Multipath Template Technique

The multipath effect on GNSS measurements depends on the physical environment and the receiver-satellite geometry. As the Galileo satellites are in nearly circular orbits at an approximate altitude of 23,222 km, they will again be over the same position on the earth's surface at the end of 10 sidereal day (each is approximately 23 hrs 56 mins in length). Thus the viewing geometry is the same each 10 days with respect to solar day, but with a shift of about four minutes per day. When the physical environment remains unchanged from day to day, then the multipath disturbance will be almost constant every 10 days.
This point is interesting from the point of view of forecasting and seeking anomalies. From the model based on several days of multipath measurements, the multipath values and curve shape should be similar, as it shown in [27].

3.5.4 Antenna Calibration

The linear combination of observable equations shown above eliminates the range, clocks, troposphere and the ionosphere delays from the observations. What remains (Equation 3.17) is a combination of code and carrier phase multipath, carrier phase ambiguities, noise, and finally code and carrier antenna effects. The carrier multipath, noise, and antenna effects are much smaller than the corresponding effects for code, thus, carrier multipath and noise can be disregarded. The presence of the carrier phase ambiguities means that it is not possible to perform an absolute calibration of the antenna effects. To eliminate this ambiguity it is used the variation of the multipath observable with elevation to retrieve the multipath measured for a specific angle. Thus, for observables in the same arc, it is computed:

$$dMP_{cx} = MP_{cx}(\varphi - \Delta\varphi) - MP_{cx} \tag{3.18}$$

Where φ is the elevation angle and MP_{cx} is the multipath for a type of data called "cx" (for example, c1 for L1 frequency or c5 for E5 frequency) This removes the constant terms from the multipath combination, namely the ambiguities. Taking this into account, it is possible to define an approximate derivative with relation

to the elevation angle as:

$$\frac{dMP_{cx}}{d\varphi} = \frac{MP_{cx}(\varphi + \Delta\varphi) - MP_{cx}(\varphi)}{\Delta\varphi} \tag{3.19}$$

The derivative multipath data vs the elevation angle can be binned into elevation bins. Every bin has the same width $\delta\varphi$, and it is centered on the a discrete set of angles, referred as φ_i. Moreover, the derivative will be modeled as a constant value on each bin. Averaging all the observations in the same nadir angle bin over time will eliminate the effect of noise, and will provide a better estimate of the derivative of the Antenna Calibration for each nadir angle:

$$\frac{dAnt}{d\varphi}(\varphi_i) = \sum_{\varphi \in \varphi_i \pm \delta\varphi_{bin}} \frac{dMP_x(\varphi)}{d\varphi} \tag{3.20}$$

Where N is the number of observations with nadir elevation angle . After this step, it is left with a quantity that should have very little noise, but it is merely the rate of change with respect to the elevation angle. Integrating this it is found the antenna effect for a given angle, affected by a constant value:

$$Ant(\varphi) = \int_{\varphi_{min}}^{\varphi} \frac{dAnt}{d\varphi}(\delta)d\delta + K \tag{3.21}$$

The constant K is determined by an adequate boundary condition. It it considered a zero mean boundary condition, defined as:

$$\langle Ant(\varphi) \rangle = \frac{1}{\varphi - \varphi_{min}} \int_{\varphi_{min}}^{\varphi} \frac{dAnt}{d\varphi}(\delta)d\delta = 0 \tag{3.22}$$

This means that the integration constant is defined as:

$$K = -\frac{1}{\varphi - \varphi_{min}} \int_{\varphi_{min}}^{\varphi} \frac{dAnt}{d\varphi}(\delta)d\delta \tag{3.23}$$

Finally, because it is dealing with bins, the above expressions are replaced by discrete sums. The Antenna bias at the bin j is given by:

$$Ant(j) == \sum_{i=i}^{j=1} \left(\frac{dAnt}{d\varphi}\bigg|_i \Delta\varphi \right) + \frac{dAnt}{d\varphi}\bigg|_j \frac{\Delta\varphi}{2} + K \tag{3.24}$$

$$K = \frac{1}{N_{bins}} \sum_{j=1}^{N_{bins}} \left[\sum_{i=i}^{j=1} \left(\frac{dAnt}{d\varphi}\bigg|_i \Delta\varphi \right) + \frac{dAnt}{d\varphi}\bigg|_j \frac{\Delta\varphi}{2} \right] \tag{3.25}$$

3.6 Code carrier coherence

Until this point, Multipath has been computing to track coherence, variations with respect to the nominal Multipath receiver environment has been detected, and at the end data is combined from various receivers to reduce the effect of errors. Now, all of mentioned above is joined to compute if there is any distortion in the data results.

Detection of this distortion is made difficult due to the fact that the ranging error caused by such distortion is dependent on the spread spectrum receiver discriminator type, correlator sparcing, and bandwidth. The importance of avoid it is because it is a threat to the integrity of the whole system.

$$ccc^i = \frac{\sum_j \left[\mu_j^i/(\sigma_j^i)^2\right]}{\sum_j \left[1/(\sigma_j^i)^2\right]} \tag{3.26}$$

where μ_j^i is the multipath deviation for reference receiver j and satellite i , and σ_j^i is the multipath error Standard Deviation for reference receiver j and satellite i. A satellite failure is declared when the code-carrier coherence test statistic, ccc^i, exceeds a threshold. [28]

3.6.1 Gaussian error propagation for CCC

If some quantities $x_1, x_2, ...x_n$ are measured with uncertainties $\delta x_1, \delta x_2, ..., \delta x_n$ and then is it required to calculate some other quantity F which depends on x_1 and x_2 and so forth, what is the uncertainty in a function "F"? The answer will show that the uncertainties $\delta x_1, \delta x_2, ..., \delta x_n$ "propagate" to the uncertainty of a function "Q". The error propagation assumes that the quantities $x_1, x_2, ...x_n$ have errors which are uncorrelated and random. (These rules can all be derived from the Gaussian equation for normally-distributed errors and, merely to be able to use them.). [29]
To compute the error propagation for CCC, the variables of the CCC function have to have their own and independent Standard Deviation and they have to be the mean of Gaussian functions.
With that condition:
Let $x_1, x_2, ..., x_n$ be random variables with Gaussian Mean $\mu_1, \mu_2, ..., \mu_n$ and Standard Deviations $\sigma_1, \sigma_2, ..., \sigma_n$.
Let F be a linear function from R^n to R:[8]

$$F(x_1, x_2, ..., x_n) = \sum_i a_i x_i + b \tag{3.27}$$

Then F is a random variable with mean and Standard Deviations given by

$$\mu_F = \sum_i a_i \mu_i + b \tag{3.28}$$

$$\sigma_F^2 = \sum_i a_i^2 \sigma_i^2 \tag{3.29}$$

Applying this procedure for CCC it is relatively simple, because in the Equation 3.26 CCC depends only on μ_j^i and its Standard Deviation.

Since the individual leveled Multipath mean is zero, the mean CCC will be zero. As for the error, the variance is:

$$(\sigma_{ccc}^i)^2 = \sum_j \left(\frac{1}{\sum_k \frac{1}{(\sigma_k^i)^2}} \right)^2 \frac{(\sigma_j^i)^2}{(\sigma_j^i)^4} \tag{3.30}$$

The variance deviation for the CCC becomes simply:

$$(\sigma_{ccc}^i)^2 = \frac{1}{\sum_j \frac{1}{(\sigma_j^i)^2}} \tag{3.31}$$

If every receiver station have the same noise, the global variance would be the Standard Deviation divided by the number of satellites. In this project 5 satellites are used to compute ccc.

$$(\sigma_{ccc}^i)^2 \approx \frac{(\sigma^i)^2}{N} \tag{3.32}$$

[8] https://www.physics.ohio-state.edu/~gan/teaching/spring04/Chapter4.pdf

3.7 Programming code explanation

In Appendix B is shown the developed code with comments. In this section the procedure of what is programmed is explained.

3.7.1 Requeriments

First of all, it is important to make clear what is necessary for developing the tool.

- GPSTK is a dynamic library used as a support for managing GNSS data. It has many advantages, because it has several classes to manage RINEX, use GNSS structures, etc. but it also has some limitations. See Section 3.7.2;
- The main objective, as it is said before, is to compute the ccc. In that formula (Equation 3.26) first, the multipath and the Standard Deviation must be calculated;
- The multipath is calculated as it is explain in Section 3.5. For that it is required the code and phase from Galileo system. The code is required in 1 band and the phase is required in 2 bands. In this project the Multipath is computed as function of elevation angle;
- The code and phase are collected from RINEX observable files. How to collect this type of data is explained in Section 3.7.5. These RINEX observable files must have the convenient data. Not all the RINEX file versions can take Galileo data. See Section 3.3.1;
- For getting the RINEX observable files, one can find them in some FTP servers. See Section 3.7.4;
- The elevation angle is obtained from the data of the antenna position and the satellite position. The antenna position is obtained from the RINEX header and the satellite position from the SP3 files;
- The SP3 are obtain from FTP servers, these files must have also the appropriate data. They must have the Galileo ephemeris;
- The Standard Deviation is computed from the Antenna Calibration and the multipath data. The Antenna Calibration is estimate as it is shown in Section 3.5.4;
- Avoid cycle slips. They can provide wrong measurements of data (phase measurements) and the following error in the multipath estimation. See Section 3.7.3.

3.7.2 GPSTK limitations

Most of the limitations found for using GPSTK are due to the fact that the tool is oriented to the Galileo System. This System is no completely operative, and many of the classes and functions of GPSTK are created to work with GPS (L1 and L2 bands). Thus, working with Galileo system and its bands may have limitations as it is shown further on.

3.7.3 Cycle slip

First, it is required to avoid cycle slip for computing multipath and the Antenna Calibration. In the Antenna Calibration procedure, the input is the derivative of multipath along elevation (Those curves can have similar shape than multipath along time). Avoiding cycle slip is important here because that derivative in those points are extremely high due to the cycle slip, it is shown as a jump in the Multipath curve of the Figure 4.3. To

calculate that, there are some performed classes in gpstk to detect situations where cycle slips could be present. For GPS, and with frequencies L1 and L2 there are a combination of two methods which uses the Melbourne-Wubbena combination and a 2nd order fitting curve using a least mean squares adjustment method. Probably, this combination is the most effective way to detect cycle slip but using them with Galileo is not possible. Instead, other algorithm in GPSTK is used to detect cycle slips. This algorithm will compute the bias between code and phase, and will compare it with a mean bias that is computed on the fly. If the current bias exceeds a given threshold, then a cycle slip is declared. Then, if a cycle slip is declared that epoch is considered invalid, and there is a gap in the elevation data. When that gap appears, the arc is closed and the new data is added for the following arc.

3.7.4 Input files

The input files can appear inside the code or as arguments when the program is run. Two types of files are used: RINEX Observable files and SP3 files.

RINEX Observable files

For using Galileo, as it is said in 3.3.1, although RINEX 2.11 can have Galileo data, it is difficult to find it in RINEX version 2, it has been searched in many stations without success. In GPSTK, the classes which are performed to avoid cycle slips only can have as input RINEX observable files version 2, but unfortunately, they can not accept RINEX version 3. There are two options, create an algorithm to detect cycle slips, which could be hard and time expensive, or obtain Galileo data 2.11 with the required data. It is more feasible to find Galileo data in RINEX 3 (RINEX 2 files do not have Galileo data) and convert them to RINEX version 2.

For using data structures in GPSTK, in concrete in gnssRINEX, it is required to use RINEX observable files version 2, for the moment version 3 is not allowed as the input of this data structure. On the other hand, although RINEX version 2.11 can have Galileo data, the unique version uploaded which have this kind of data is version 3. There are many FTP directories with RINEX data for downloading, some of them, it is supposed they can have Galileo data (The data is Mixed if the RINEX file has more than a unique type of data, for example GPS, Glonass or Galileo), a few of them have Galileo data and neither of them which have Galileo data are RINEX version 2.

The UNAVCO main webpage has a FTP server [9] which has both RINEX 2 and RINEX 3 files for downloading. What is performed is downloading RINEX files version 3 with Galileo data and convert them into RINEX files version 2.11 which can contain this data. There is a free online tool to do this [10]. For the output files, their names are composed for: first, the name of the receiver station, then, the day of the year and dot plus the year. For the example, it is required the RINEX observable file for the receiver station with id P558 for the day 91 of the year 2017. The result is "P558091A00.17O" where the "O" at the end means observable file.

SP3 files

On the other hand, it is mandatory to have proper SP3 files. They must have Galileo ephemeris data. Although Galileo is not operative yet, there are some available ones (Not many). For example CNES (Centre National d'Etudes Spatiales) and CLS (Collecte Localisation Satellites). Since 2007, CNES/GS and CLS teams have been processing regularly GPS and GNSS data from a worldwide network of IGS permanent stations[11].

[9] https://www.unavco.org/data/gps-gnss/ftp/ftp.html
[10] http://www.gps-solutions.com/gnss_converter
[11] https://igsac-cnes.cls.fr/html/products.html

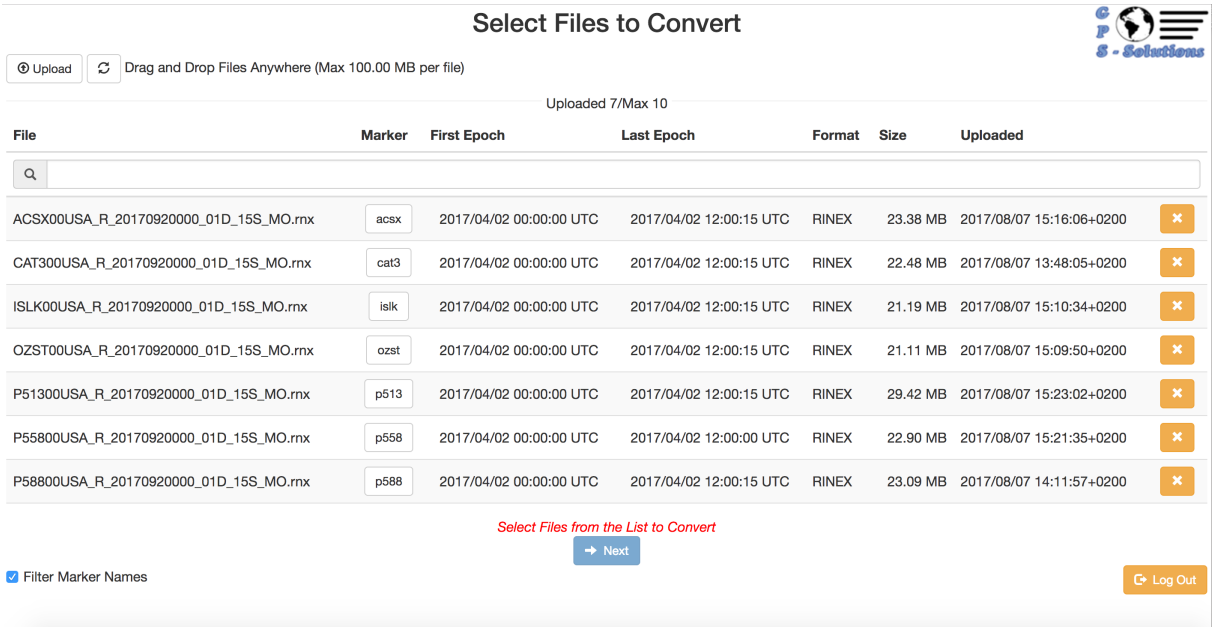

Figure 3.10: RINEX files version 3 uploaded and ready for being converted

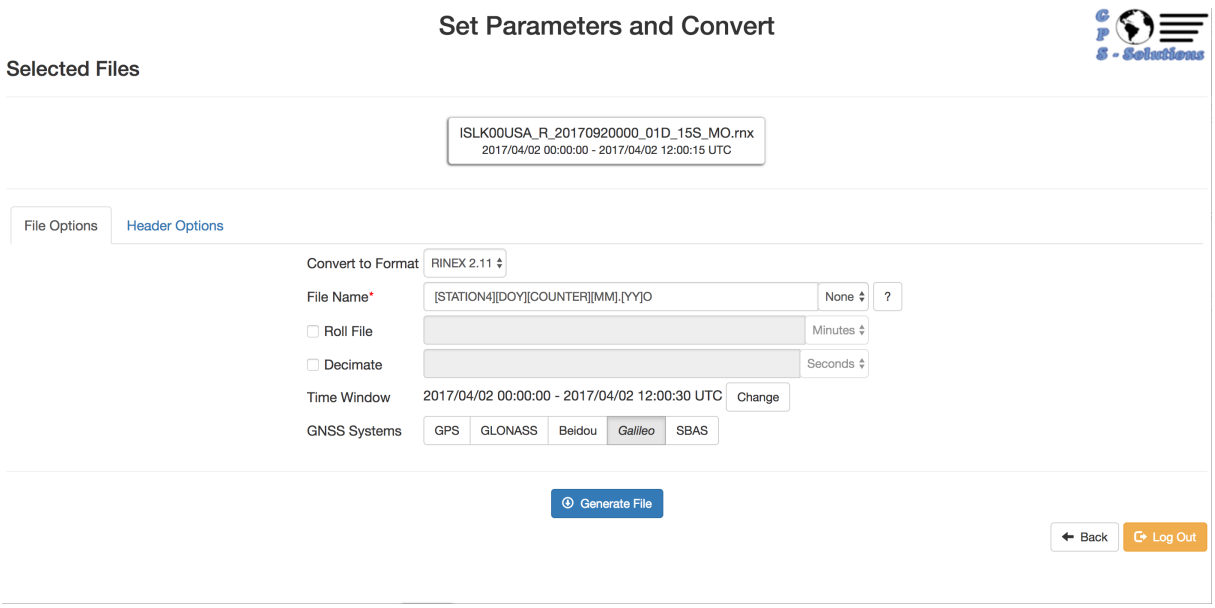

Figure 3.11: RINEX file version 3 loaded. Here, one can choose different options. It is chosen to get as output a RINEX file version 2.11, with the name structure shown and only Galileo data

They compute precise GNSS orbits together with Earth rotation parameters and stations coordinates at the sub centimeter level. There is a SP3 file from the list of files called grm, which has Daily GPS, GLONASS and GALILEO ephemeris/clock at 15-min intervals.

3.7.5 Computing Antenna Calibration

The Antenna Calibration is what is called the mean multipath for a specific receiver station along a concrete number of days. From that definition is possible to define the Standard Deviation forward in Section 3.7.5.

Collecting data

To compute multipath in the section 3.5, it was said that two frequencies are required. In section 3.1.1 it is explained why they are E1 and E5. For the Multipath equation 3.17, the E1 Multipath is computed, for achieving that, the E1 and E5 codes must be obtained but also E1 carrier phase. The GPSTK library let one work reading the observable file epoch by epoch and adding the relevant data and avoiding every type of damaged or incorrect information to a data structure called gnssRINEX. What is required to get this is specify the typeid (kind of data required), then a loop is developed for every present satellite in the available Galileo satellite list. For each satellite, the tool for computing the linear combination of equations and cycle slips is declared and defined (see section 3.7.3). After that, the codes and phases are obtained and the Multipath computed. At the same time the elevation angle is calculated as the elevation angle from the antenna position, which is obtained from the RINEX observable file header, and from the satellite position. The satellite position is obtained from the SP3 files.

Binning Elevation

For representing plots for multpaths depending on elevation, the elevation was binned into integer numbers. As a result, the elevation angle is a discrete variable and thus, multipath values are represented against a range of numbers from 0 to 90 where there are only integer values. To have a value of Multipath for each elevation angle for every arc, while the satellite is in the range of the same whole elevation angle number, the mean of multipath is computed.

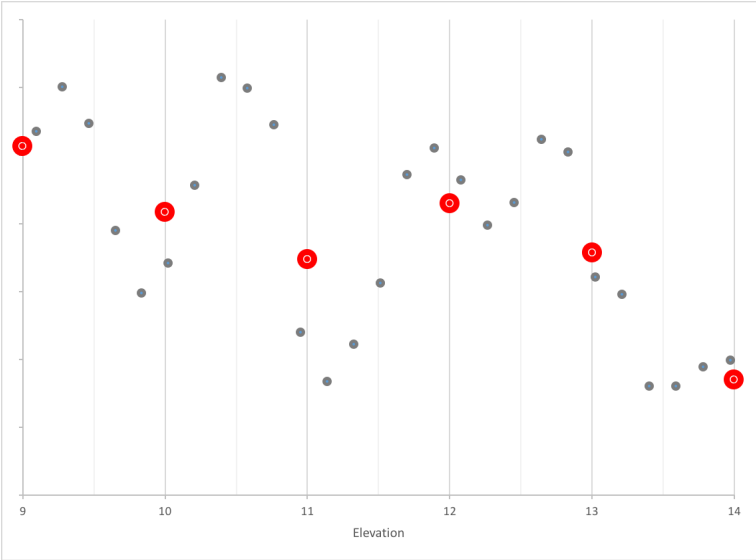

Figure 3.12: Example of how to bin elevation

In the Figure 3.12 is shown an example of a binning elevation. The grey dots represent the multipath computed from the data in the RINEX observable files. They are not fixed in specific elevation values, but they are distanced between 0.2-0.4 degrees one from the previous.
In the Section 4.3.1 it is shown the results for the multipath after binning elevation.

Computing Standard Deviation

After computing the Antenna Calibration, it is possible to compute the Standard Deviation as the deviation of every Multipath values from the Antenna Calibration Multipath. The Standard Deviation is computed as:

$$\sigma^i = \sqrt[2]{\frac{\sum_k^N (\mu_k - Ant)^2}{N-1}} \tag{3.33}$$

Where σ^i is the Standard Deviation for the receiver station i for a specific elevation angle, μ_k is one of the multipath measurements for that specific elevation angle and N is the number of measurements obtained along a specific period of time for an specific elevation angle. The time chosen for getting all the data for computing both the Antenna Calibration and the following Standard Deviation is one week.
This computation is done for every satellite together instead of using one by one.

3.7.6 Computing CCC

To compute the CCC value, it is used the Equation 3.26. For accomplish that, the measured leveled multipath is used in place of μ_j^i , and the Standard Deviation of Multipath as a function of elevation in place of σ_i.
The mean Multipath that is computed identifies any systematic effects in the data. That the Multipath is close to zero indicates that the systematic effect is mostly zero, which is to be expected given the requirements on IGS stations to have good quality.
The Standard deviation of Multipath that was computed in the Section 3.7.5 measures the Multipath noise level as a function of elevation per site. That was done for the multipath values used to compute the Antenna Calibration, here the Standard Deviation is done with the leveled multipath explained in Section 3.7.6. This

new technique consists on:

- Compute Biased Multipath: for each epoch in the pass, compute the MP;
- Compute mean MP per pass: For each pass compute the mean MP;
- Compute leveled MP: For each epoch, subtract mean Multipath from the MP at that epoch;
- Compute leveled MP as a function of elevation: Bin the leveled MP as a function of elevation bins;
- Compute the mean and Standard Deviation of the leveled Multipath for each elevation bin.

Computing leveled Multipath

To compute the leveled Multipath is mandatory to compute first, the Biased Multipath and then, the Mean Multipath.
To compute Biased Multipath: for each epoch in the pass, it is computed the Multipath linear combination for each frequency, using the procedure used in Section 3.5.
To compute Mean Multipath per pass for each frequency: For each pass and frequency the mean Multipath over the full pass is computed as:

$$\overline{MP} = \frac{\sum_t MP^i_{j,f}(t)}{N^i_{j,f}} \tag{3.34}$$

Here N is the number of observations for the pass where f means that belongs to the same pass. Finally, to compute leveled Multipath: For each epoch, subtract $\overline{MP}$ from the MP at that epoch:

$$\mu^i_j(t) = MP^i_j(t) - \overline{MP} \tag{3.35}$$

For each epoch it is important to compute also the elevation, so from the leveled Multipath and elevation it is possible to organize the data in terms of leveled Multipath as a function of elevation.
This procedure provides the Multipath observable per pass.

3.8 CCC threshold

After computing the leveled Multipath and its deviation from the Antenna Calibration Multipath, every variable from the right part of the CCC equation (Equation 3.26) is known. Then, it is possible to compute CCC.

The CCC should be a function of time with zero mean and with noisy shape. Its Standard Deviation is computed through the procedure called Gaussian error propagation explained in Section 3.6.1.
The threshold value depends on how safe the system is required to be. The safer the system the more false alarms there will be. If a probability of 95% of having the data in the correct environment is required, there will be a 5% where the values would overpass this threshold and the flag would be on, in that case, there will be 5% of the values causing false alarms, because the code-carrier incoherence does not really exist.
If the threshold value is considered higher, the system would last longer to detect this feared event but there would be fewer false alarms. If the threshold is establish lower, it would happen the opposite.
To establish a criteria of what threshold is appropriate, a probability of having 95% of good data is chosen but using the worst case standard deviation, which means there is a much smaller probability than 5% of

having a CCC value which overpass the threshold because most of the data will not have the highest Standard Deviation, they will have a lower one. If the Standard Deviation is lower, as it said above the probability of overpassing the threshold is lower than 5%.

$$PR(\mu - 2\sigma \leq X \leq \mu + 2\sigma) \approx 0.9545 \tag{3.36}$$

Where μ is the mean, σ is the Standard Deviation and X is an observation from a normally distributed random variable. Applying to this work, X is the ccc value, μ is the mean of CCC, which should be zero, and σ is the most extreme Standard Deviation of CCC obtained along the time.
The Equation 3.36 shows that the CCC threshold is $\mu \pm 2\sigma$.

Chapter 4

Data Results

In this chapter the results obtained are plotted and shown. It is shown the evolution of the procedure until getting the CCC aimed threshold. In Appendix A it is explained what R is and how it is used to plot the data results.

```
     2.11           OBSERVATION DATA    M (MIXED)           RINEX VERSION / TYPE
sbf2rin-11.3.3      Unknown             20170807 161645 UTC PGM / RUN BY / DATE
SEPTENTRIO RECEIVERS OUTPUT ALIGNED CARRIER PHASES.         COMMENT
NO FURTHER PHASE SHIFT APPLIED IN THE RINEX ENCODER.        COMMENT
Map of observations from RINEX 3 to RINEX 2.11              COMMENT
 E: C1C->C1 C5Q->C5 C6C->C6 C7Q->C7 C8Q->C8 L1C->L1 L5Q->L5 COMMENT
    L6C->L6 L7Q->L7 L8Q->L8                                 COMMENT
ozst                                                        MARKER NAME
UNAVCO              UNAVCO                                  OBSERVER / AGENCY
3012360             SEPT POLARX5        5.1.0               REC # / TYPE / VERS
0220356924          TRM59800.80     SCIT                    ANT # / TYPE
 -2574193.3327 -4577156.7783  3609641.2405                   APPROX POSITION XYZ
        0.0083        0.0000        0.0000                   ANTENNA: DELTA H/E/N
    10    C1    C5    C6    C7    C8    L1    L5    L6    L7# / TYPES OF OBSERV
          L8                                                # / TYPES OF OBSERV
    15.000                                                  INTERVAL
  2017    04    02    00    00   00.0000000     GPS         TIME OF FIRST OBS
  2017    04    02    23    59   45.0000000     GPS         TIME OF LAST OBS
    15.000                                                  INTERVAL
    13                                                      # OF SATELLITES
   E01  1503  1503  1503  1503  1503  1503  1503  1503  1503PRN / # OF OBS
        1503                                                PRN / # OF OBS
   E02  2407  2425  2427  2431  2446  2407  2425  2427  2431PRN / # OF OBS
        2446                                                PRN / # OF OBS
   E05  1319  1320  1315  1325  1324  1315  1320  1315  1325PRN / # OF OBS
        1324                                                PRN / # OF OBS
   E08  2166  2177  2197  2187  2197  2166  2177  2197  2187PRN / # OF OBS
        2197                                                PRN / # OF OBS
   E09  1559  1569  1568  1570  1573  1559  1569  1568  1570PRN / # OF OBS
        1573                                                PRN / # OF OBS
   E11  1003  1010  1010  1017  1019  1003  1010  1010  1017PRN / # OF OBS
        1019                                                PRN / # OF OBS
   E12   622   655   668   661   673   622   655   668   661PRN / # OF OBS
         673                                                PRN / # OF OBS
   E19  1646  1662  1658  1659  1660  1646  1662  1658  1659PRN / # OF OBS
        1660                                                PRN / # OF OBS
   E20  1943                              1936              PRN / # OF OBS
                                                            PRN / # OF OBS
   E22  2254  2254  2254  2255  2255  2254  2254  2254  2255PRN / # OF OBS
        2255                                                PRN / # OF OBS
   E24   556   560   562   560   561   556   560   562   560PRN / # OF OBS
         561                                                PRN / # OF OBS
   E26  2215  2227  2211  2222  2224  2215  2227  2211  2222PRN / # OF OBS
        2224                                                PRN / # OF OBS
   E30  2804  2809  2823  2815  2830  2804  2809  2823  2815PRN / # OF OBS
        2830                                                PRN / # OF OBS
                                                            END OF HEADER
 17  4  2  0  0  0.0000000  0  5E02E08E11E12E24
  26753722.524    26753725.480    26753723.276    26753722.665    26753723.400
 140591790.6820  104987400.5700  114116723.0820  107726186.5680  106356796.6430
  26023004.881    26023008.410    26023007.815    26023004.977    26023005.958
 136751823.9110  102119872.7720  110999859.9310  104783863.6000  103451864.7610
  23225409.253    23225408.691    23225406.012    23225405.934    23225406.950
 122050382.8770   91141566.6740   99066887.8590   93519167.0120   92330368.4170
  25159299.744    25159300.161    25159297.036    25159297.556    25159298.251
 132213086.0440   98730595.9300  107315835.7250  101306162.3080  100018383.6930
  24709702.109    24709716.335    24709694.665    24709712.920    24709714.361
 129850392.3870   96966282.5420  105398036.0560   99495820.6480   98231051.1690
 17  4  2  0  0 15.0000000  0  5E02E08E11E12E24
  26747766.171    26747769.686    26747767.361    26747766.746    26747767.463
 140560492.2260  104964028.3840  114091318.5160  107702204.6630  106333119.5980
  26024077.843    26024081.010    26024080.706    26024078.217    26024079.515
 136757464.8000  102124085.1160  111004438.5590  104788185.8780  103456132.0720
  23222914.039    23222913.459    23222910.955    23222910.644    23222911.763
 122037270.6340   91131775.0710   99056244.8050   93509119.9710   92320449.0950
  25164181.519    25164181.811    25164178.829    25164179.148    25164180.070
 132238739.8520   98749752.9930  107336658.6310  101325819.1310  100037790.6370
  24712976.727    24712990.800    24712969.247    24712987.846    24712989.073
 129867601.0170   96979133.1330  105412004.0970   99509006.4760   98244069.3780
```

(a) Header and beginning of body

```
 17  4  2  0  0 30.0000000  0  5E02E08E11E12E24
  26741808.839    26741811.617    26741809.153    26741809.217    26741809.840
 140529184.9800  104940649.6160  114065906.7900  107678216.0020  106309435.8820
  26025168.639    26025172.307    26025172.446    26025168.432    26025170.490
 136763198.3190  102128366.6230  111009092.3390  104792579.0630  103460469.4190
  23220438.671    23220437.943    23220435.421    23220435.168    23220436.274
 122024261.9030   91122060.7670   99045685.7800   93499152.2550   92310608.0850
  25169079.012    25169079.156    25169076.112    25169076.994    25169077.572
 132264476.2640   98768971.7350  107357548.5650  101345539.2310  100057260.0560
  24716258.812    24716272.917    24716251.498    24716270.236    24716271.297
 129884849.3400   96992013.3690  105426004.3520   99522222.7160   98257117.6170
 17  4  2  0  0 45.0000000  0  5E02E08E11E12E24
  26735848.783    26735851.861    26735850.117    26735849.846    26735850.437
 140497867.6010  104917263.2760  114040486.8550  107654219.5880  106285744.5040
  26026276.932    26026281.534    26026280.260    26026277.083    26026278.850
 136769022.9200  102132716.1550  111013820.0740  104797042.0500  103464875.6780
  23217982.718    23217981.833    23217979.414    23217979.116    23217980.235
 122011355.3910   91112422.7940   99035209.7190   93489262.8550   92300844.3980
  25173991.963    25173991.978    25173989.545    25173989.498    25173990.483
 132290293.7260   98788251.0070  107378504.2990  101365321.4370  100076790.7970
  24719548.423    24719562.657    24719540.957    24719559.766    24719560.800
 129902136.0610   97004922.2840  105440035.7740   99535468.3770   98270194.9040
 17  4  2  0  1  0.0000000  0  5E02E08E11E12E24
  26729887.818    26729890.399    26729888.820    26729888.160    26729889.114
 140466539.9430  104893869.2550  114015058.5980  107630215.3210  106262045.3610
  26027403.529    26027407.416    26027405.999    26027403.649    26027404.423
 136774938.3420  102137133.4850  111018621.5330  104801574.6060  103469350.6200
  23215546.017    23215545.271    23215542.745    23215542.577    23215543.633
 121998550.9730   91102861.0600   99024816.5290   93479451.6830   92291157.9450
  25178920.218    25178920.371    25178917.411    25178917.945    25178918.756
 132316191.8500   98807590.5070  107399525.4950  101385165.4510  100096382.5530
  24722845.405    24722859.523    24722837.935    24722856.507    24722857.639
 129919461.0620   97017859.7650  105454098.2700   99548743.3660   98283301.1400
 17  4  2  0  1 15.0000000  0  5E02E08E11E12E24
  26723925.569    26723928.275    26723926.066    26723925.317    26723926.686
 140435206.7670  104870471.1240  113989625.8370  107606206.7910  106238342.0320
  26028547.128    26028550.778    26028550.143    26028546.501    26028548.331
 136780949.1350  102141622.0710  111023500.4400  104806180.3040  103473897.7630
  23213129.761    23213129.104    23213126.623    23213126.374    23213127.350
 121985853.3860   91093379.1050   99014510.0500   93469722.3710   92281552.3110
  25183864.603    25183864.827    25183861.966    25183862.330    25183863.216
 132342175.1340   98826993.6100  107420615.8220  101405074.7180  100116038.7380
  24726150.124    24726164.613    24726143.104    24726161.529    24726162.698
 129936829.0720   97030829.3820  105468195.6830   99562051.3190   98296439.9250
 17  4  2  0  1 30.0000000  0  5E02E08E11E12E24
  26717961.086    26717963.918    26717961.284    26717960.767    26717962.012
 140403862.5600  104847064.7430  113964184.1150  107582189.8130  106214630.3530
  26029708.888    26029712.090    26029710.454    26029707.370    26029709.433
 136787049.7240  102146177.6970  111028452.1990  104810854.7700  103478512.8130
  23210732.654    23210732.060    23210729.552    23210729.343    23210730.373
 121973257.0810   91083972.7860   99004285.7880   93460070.6720   92272023.3030
  25188824.286    25188824.386    25188821.859    25188822.044    25188822.783
 132368237.8660   98846456.0280  107441770.6330  101425044.8590  100135755.0170
  24729462.553    24729477.033    24729455.449    24729473.739    24729474.908
 129954234.6270   97043827.0300  105482323.5630   99575388.0360   98309607.1070
 17  4  2  0  1 45.0000000  0  5E02E08E11E12E24
  26711994.345    26711997.025    26711994.490    26711993.879    26711995.168
 140372506.7550  104823649.7160  113938732.9800  107558163.9460  106190909.9050
  26030886.142    26030890.171    26030888.781    26030885.577    26030887.349
 136793239.3260  102150799.8090  111033476.2440  104815597.4710  103483195.2160
  23208354.734    23208354.257    23208351.507    23208351.437    23208352.521
 121960761.5640   91074641.7270   98994143.3230   93450496.1860   92262570.5300
  25193799.169    25193798.906    25193795.584    25193796.462    25193797.268
 132394379.2060   98865977.1610  107462989.2570  101445075.2370  100155530.7740
  24732781.797    24732795.847    24732774.473    24732793.046    24732794.129
 129971677.1680   97056852.3000  105496481.4670   99588753.1010   98322802.2750
 17  4  2  0  2  0.0000000  0  5E02E08E11E12E24
  26706025.936    26706029.433    26706026.719    26706026.538    26706027.523
 140341146.7560  104800231.5490  113913278.4360  107534134.8720  106167186.2850
```

(b) Continuation of the body (not the end of the body)

Figure 4.1: RINEX file 2.11 after conversion. Named OZST092A00.17O

4.1 Suitable RINEX Input

After downloading the RINEX observable file version 3 and converting it into version 2 as explained in section 3.7.4, it is shown in the Figure 3.10 and in the Figure 3.11 how to do it, here in the Figure 4.1 the output is shown.
As one can see, there is information of what Galileo data is available in the observable type list. Because it has been created a RINEX Observable File with only Galileo data, there are not GPS nor GLONASS information.
This file fits perfectly as input in the program created of C++. It is appropriate to be manipulated by the GPSTK classes, in concrete, to be the input of data structures. This part became essential because other RINEX version are not allowed as inputs.
The Figure 4.1 shows the RINEX Observable file of the station called OZST for the day 92 of the year 2017. It corresponds with the 2nd of April of 2017, this is important to know because the SP3 file date is typed in week number and day of the week.(WN: 1943 and DoW: 1)
For the following sections, this station (OZST) and that date will be used to show an example of the intermediate results.

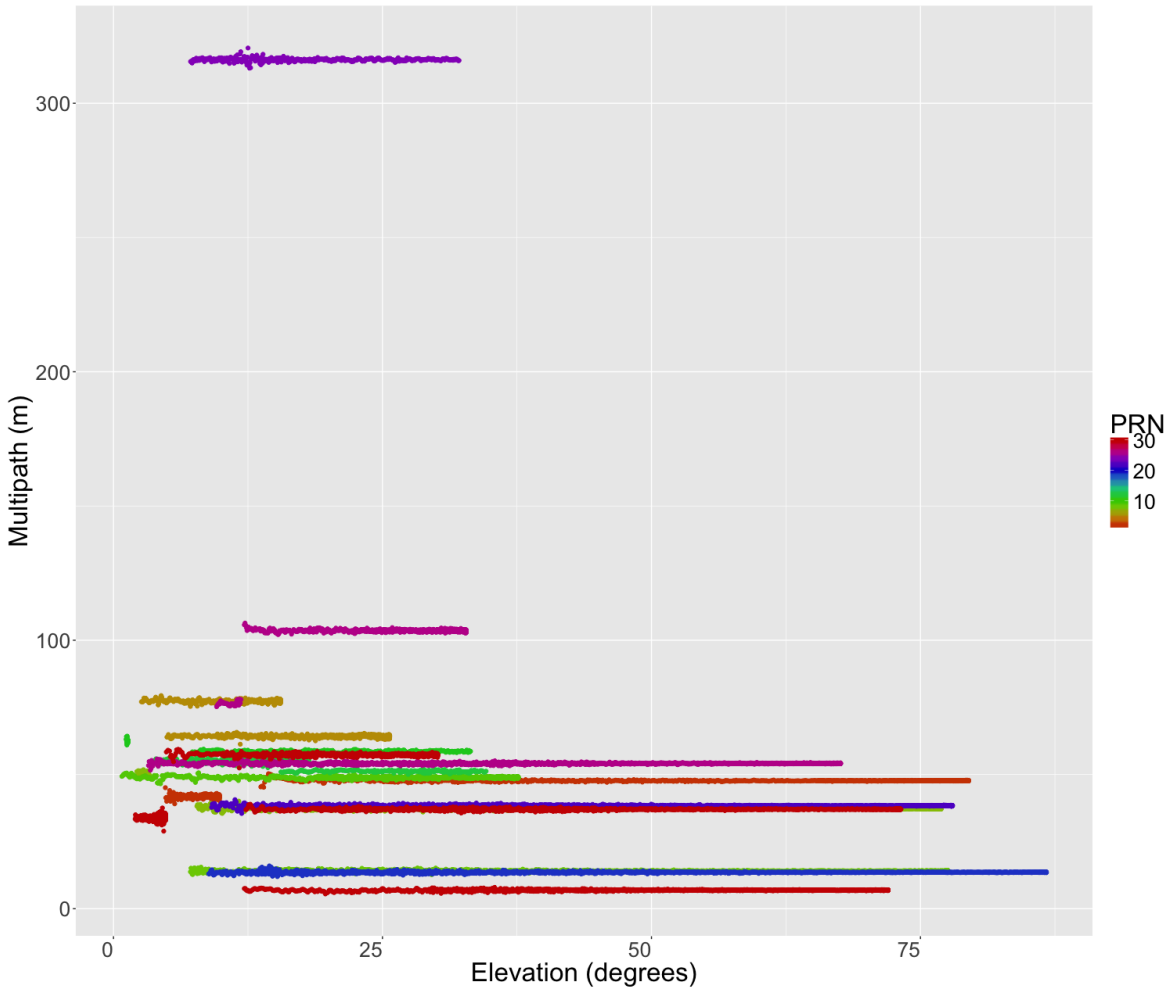

Figure 4.2: Continuous Multipath function of elevation

4.2 Raw Multipath

After computing Multipath for every epoch during a day, the result is a big amount of data with a jump between two consecutive points in elevation of 0.03 degrees approx. and several points to choose to identify a precise Multipath to each elevation angle. In the Figure 4.2 it is shown an example for the station OZST for the day 85 of 2017.

As one can appreciate from the Figure 4.2 the Multipath values normally are noise and more or less constant with the elevation, also there is a jump in the satellite of the color purple, and the most remarkable feature is that the data lines are separate one from the other.

The cycle slip is taken into account for doing the program and it is avoidable using the appropriate algorithm, it is used and works properly for further results. The separation between the curves is due to the term called ambiguity. This ambiguity term is a constant number for each arc but is different for different arcs. Because of that reason, it is necessary to compute all the results shown later.

4.3 Antenna Calibration

To show the results of the Antenna Calibration process, firstly, the discrete Multipath will be shown, then the derivative of Multipath with elevation and at the end the result of the computation of the Antenna Calibration for different quantity of days used and also for different selection of frequency bands.

Continuous function	**Discrete function**
{**Elevation,Multipath**}	{**Elevation,Multipath**}
{15.5858, -11.7399}	
{15.6744, -11.5636}	
{15.7630, -11.9429}	
{15.8516, -11.8559}	
{15.9402, -11.6807}	
{16.0289, -11.5541}	{16, -11.7239}
{16.1176, -11.3875}	
{16.2063, -11.8801}	
{16.2951, -11.7652}	
{16.3839, -11.8710}	
{16.4727, -11.7217}	

Table 4.1: Conversion from continuous Multipath to discrete Multipath function

4.3.1 Discrete Multipath

To compute the discrete Multipath, it was chosen an elevation bin of 1 degree of elevation angle. Then, from the Figure 4.2, for every Multipath consecutive point which share the elevation angles, are collected to compute a discrete Multipath point. e.g. Every consecutive Multipath from the continuous function wich has an elevation value from 15.5 to 16.49 are collected to compute the mean and obtaining the discrete Multipath point for the elevation angle of 16.

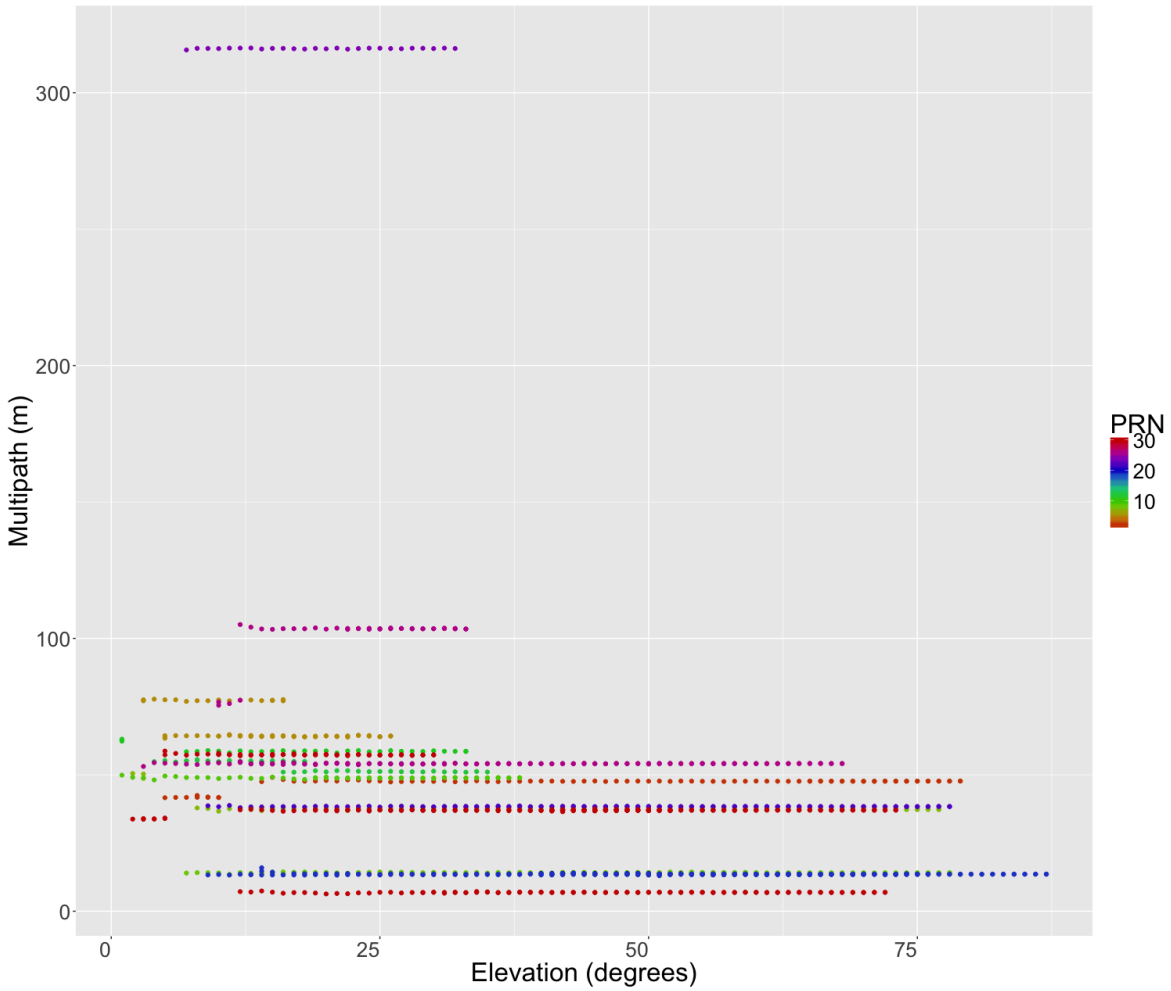

Figure 4.3: Discrete Multipath function of elevation

As a result, the curves shown in Figure 4.3 are shown cleaner, and one can work with them to compute derivatives and treat them as discrete functions.
The Table 4.3 shows an example of how 11 continuous points are converted into just one for the discrete function. This is also explained in the Figure 3.12.
This process is explained in Section 3.5.4 where the discretization is a previous step of the process explained there.

Standard Deviation

The Figure 4.4 shows the standard deviation between each set of points in the continuous Multipath

function (Figure 4.2) and the discrete point computed as the mean. In the Figure 4.4 one can appreciate higher values of standard deviation at lower elevations. The different colors means different satellites, for thus there is not a single value for each elevation value, instead of that, there as many values for each elevation angle value as number of satellites and passes were available at that day.

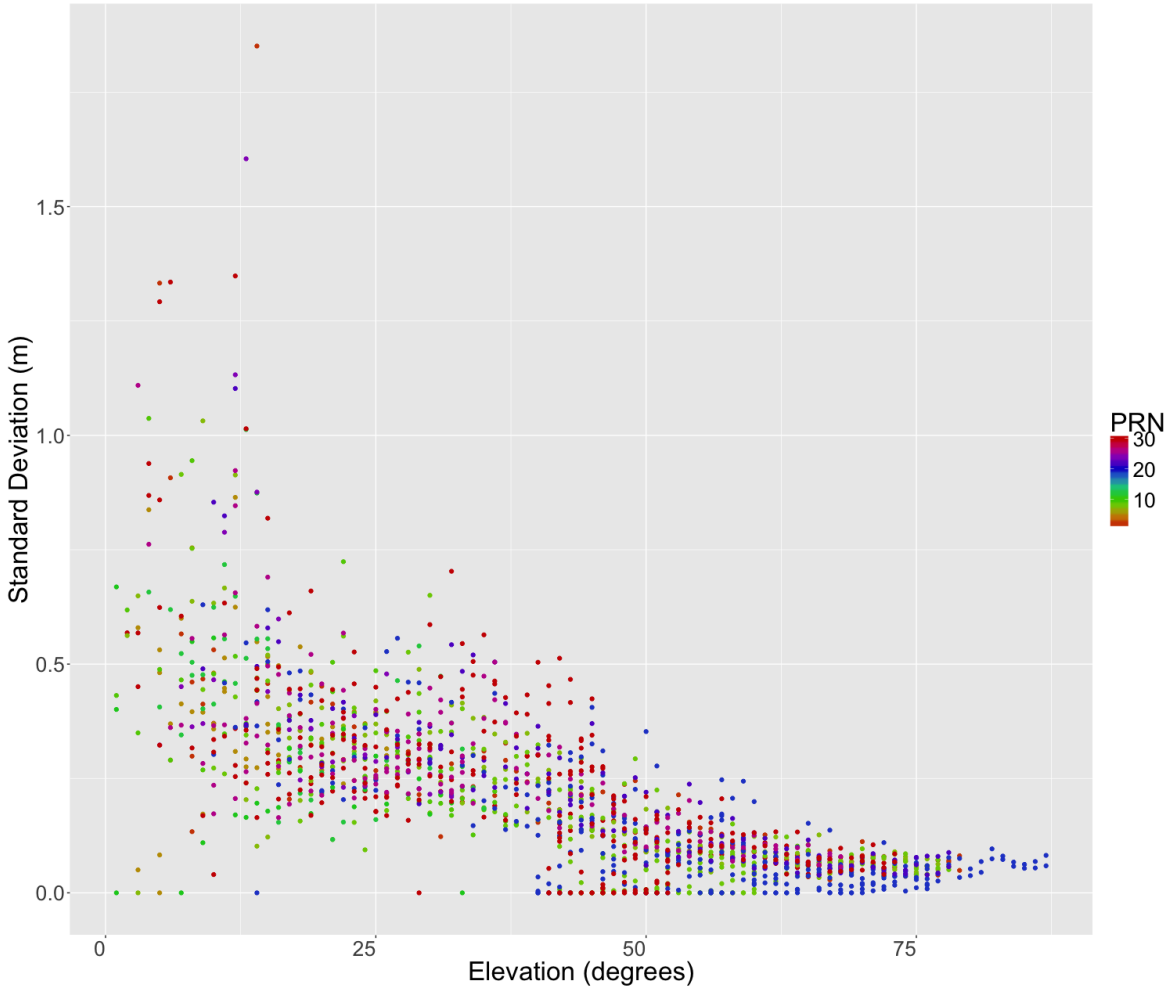

Figure 4.4: Standard deviation of the continuous Multipath versus the discrete Multipath. It is function of Elevation.

4.3.2 Derivative Multipath with Elevation

After computing the discrete Multipath, the derivative Multipath is calculated of each arc between consecutive points. The result is shown in the Figure 4.5. Here, it is shown the derivative of Multipath with elevation for every satellite. It it possible to have more than one value for an specific satellite sharing the elevation angle value.

As it is shown, most of the values are negative, due to the fact that the function has higher Multipath values at lower elevations and lower Multipath values at higher elevation angle values. On the other hand, there are more positive values at lower elevations due to the fact that at lower elevation angles there are more dispersion and the Multipath values are more unpredictable.

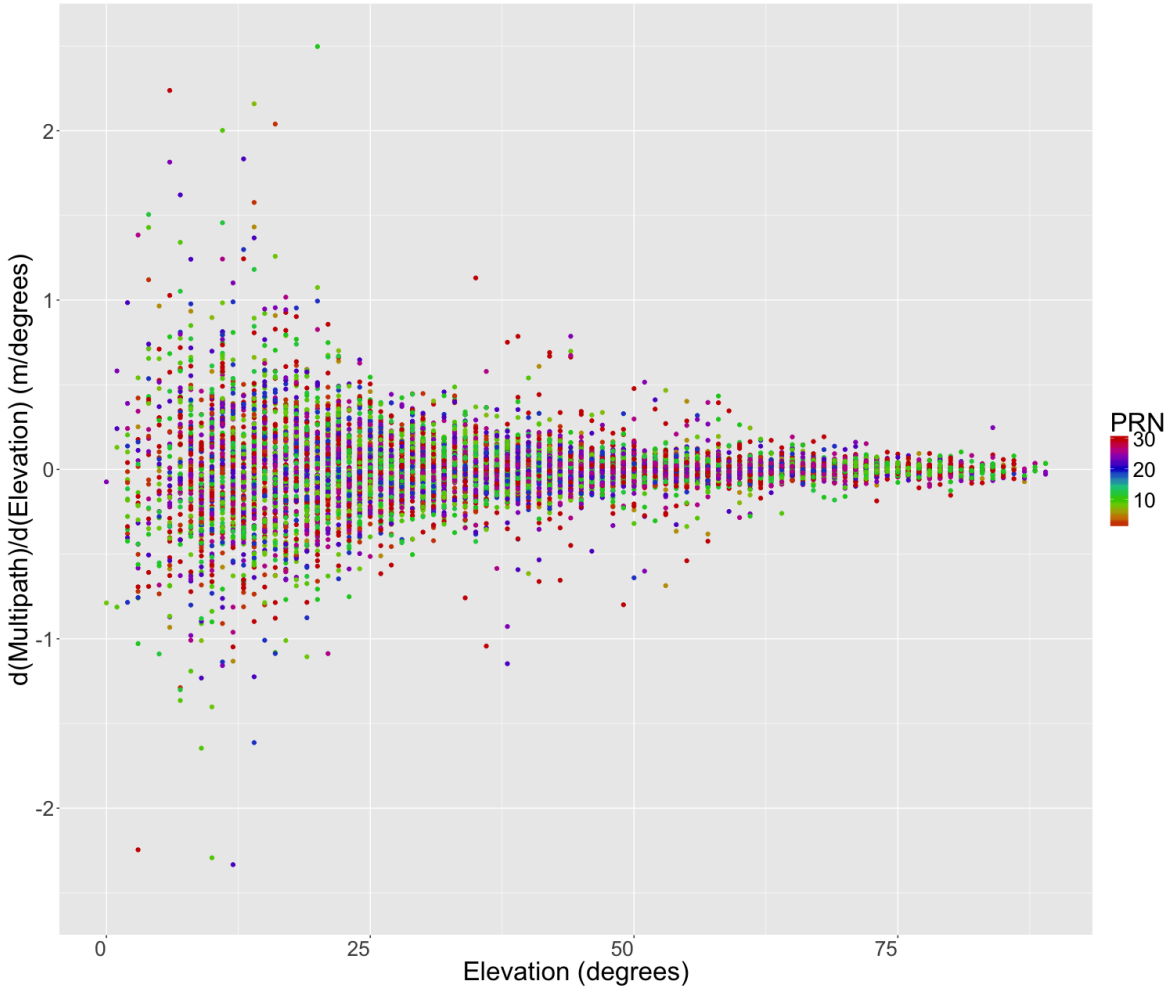

Figure 4.5: Derivative of dicrete Multipath with elevation of all the satellites. It is a function of Elevation.

Then, the mean of those values are computed, having as a result the Figure 4.6, where there is a single value for each elevation angle. For computing this is used the Equation 3.20.
Here, except for lower elevations, all the values are in the negative part, doing the function of mean Multipath decreasing with elevation.

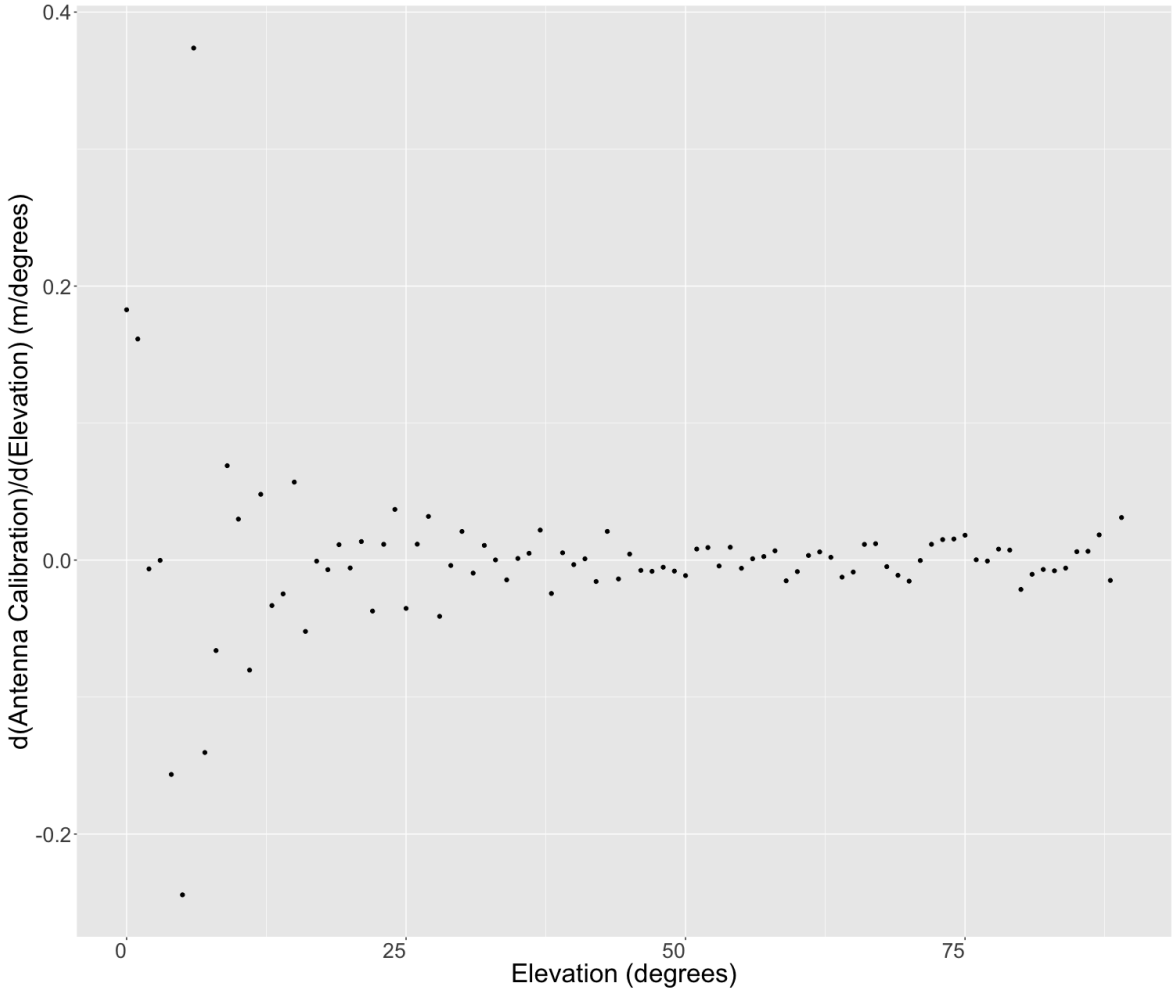

Figure 4.6: Mean of the derivative of discreteMultipath with elevation of all the satellites.The result is the derivative of the Antenna Calibration with Elevation.

As explained above, the values of the standard deviation here are predictable. At lower elevations, the derivative Multipath with elevation are spreaded, then the standard deviation is higher.

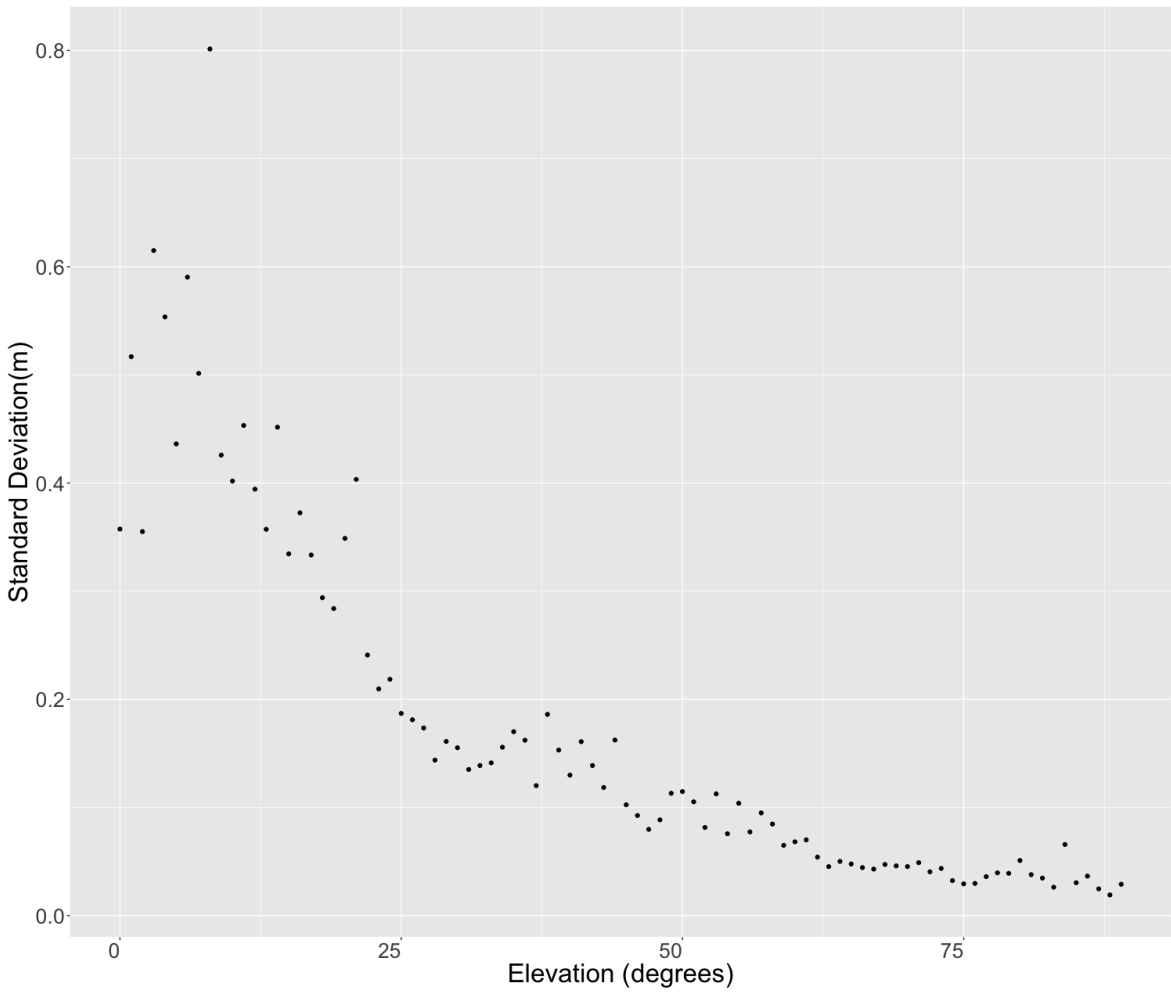

Figure 4.7: Standard deviation of the derivative Multipath with elevation of all the satellites from their mean. It is a function of Elevation.

4.3.3 Multipath of Antenna Calibration

As final results for the Antenna Calibrations, some analysis are done. The previous results are done for a single day. That analysis can be done for as many days as is required. First, the Figure 4.8 shows the quantity of days used to compute the Antenna Calibration. One can check the fact that increasing the number of days, the functions tend to converge, for that it is used 7 days as an appropriate number of days, because it is enough to have good results. More days would increase the computational cost and would not provide more accurate results.

For one and two days, the values depends highly one what features of those days could change a little bit the Multipath values. More than 3 days provide repetitive results wich can be extrapolated to other dates.

This results for both Figure 4.8 and Figure 4.9 are got using the frequency band E1 for the main frequency (Then, the Multipath is called Multipath for the frequency E1) and E5 as the second frequency band.

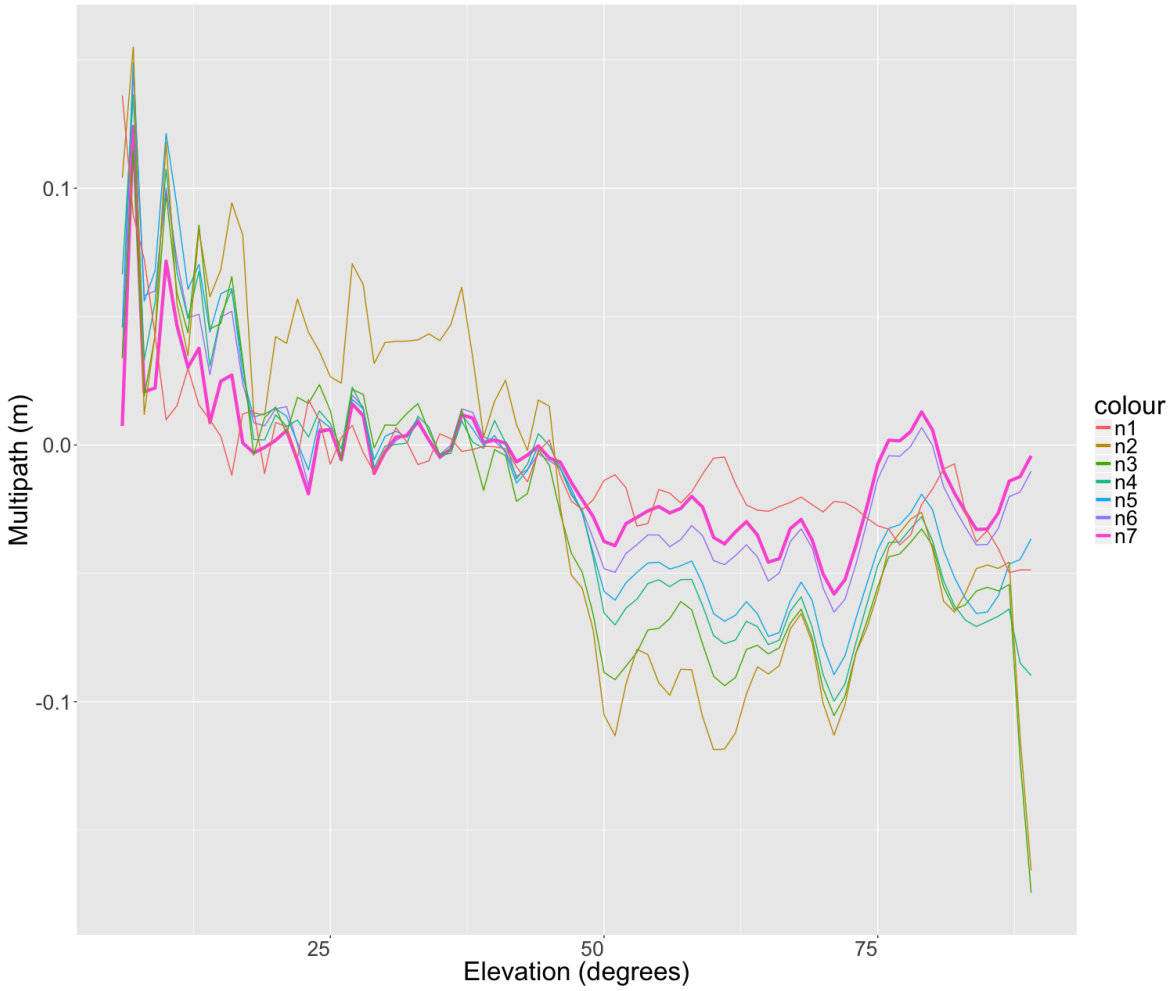

Figure 4.8: Multipath computed by Antenna Calibration from one to seven days, function of elevation.

The Figure 4.9 shows the results of the Antenna Calibration for different receiver stations. As it is shown, the results are quiet similar for every receiver station. The main differences are at lower elevation, probably are due to the mountain, vegetation, etc. which appear near each receiver station. This values are got for future computation on the test part.
As one can see, the Antenna Calibration function is a zero mean function, which has the property of having the sum of all Multipath values the result of zero.

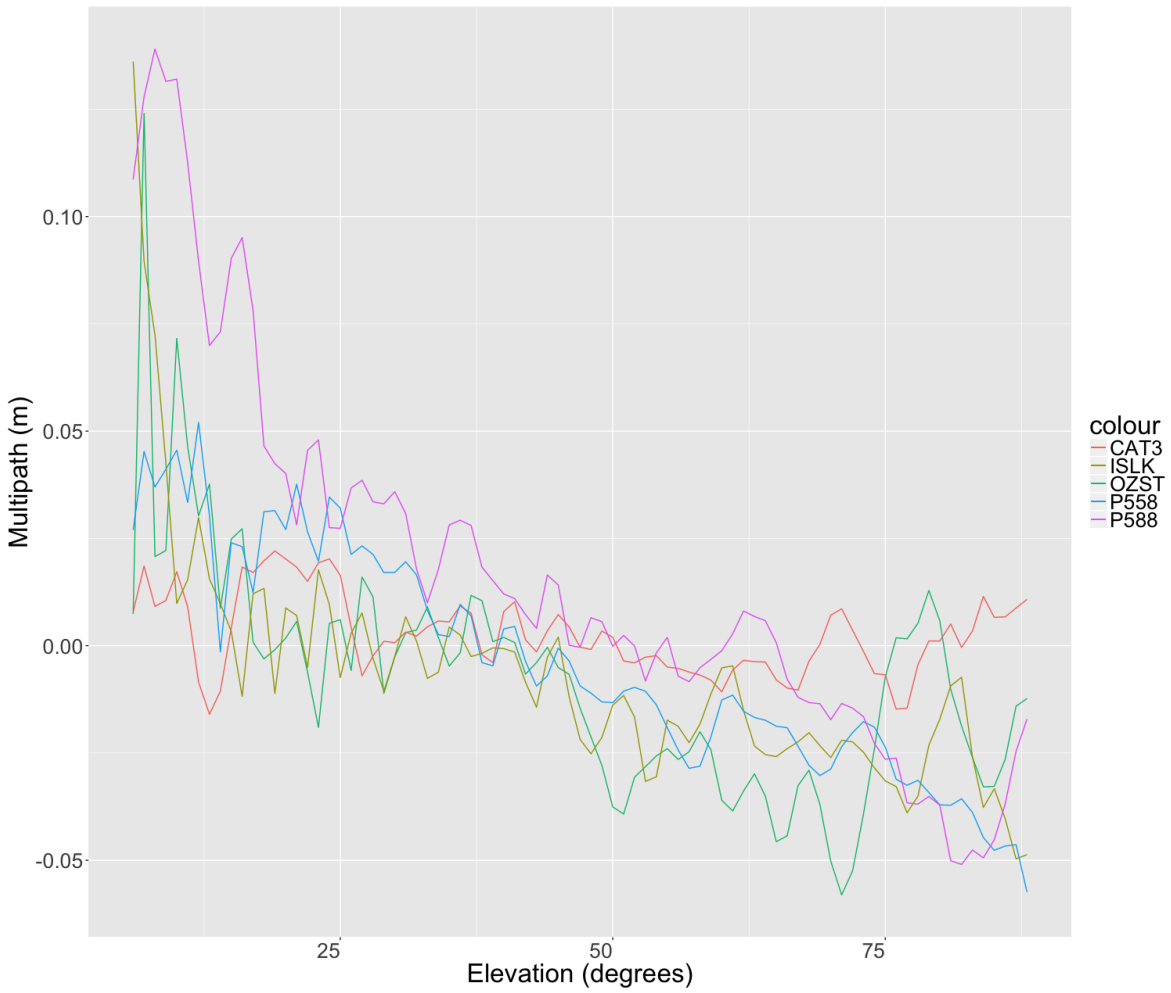

Figure 4.9: Multipath computed by Antenna Calibration for every station for seven days and for the band E1E5, function of elevation.

4.3.4 Standard Deviation

To compute the standard deviation, the procedure explained in Section 3.7.5 is used.
This Standard Deviation is the deviation of the Multipath values during a week from the Antenna Calibration shown in the Section 4.3.3. Where the elevation is binned in whole numbers of elevation angles.
Figure 4.10 shows the standard deviation computed for the Antenna Calibration obtained in Figure 4.9. Despite having very close curves for the Antenna Calibration, there are bigger differences in the standard deviation curves. The shapes of them are very similar but for lower elevation angles they are quiet different. At lower elevation, the terrain could be different and as the azimuth angle is considered, there could be a mountain in one side of the receiver station or other geographic feature on a single point.

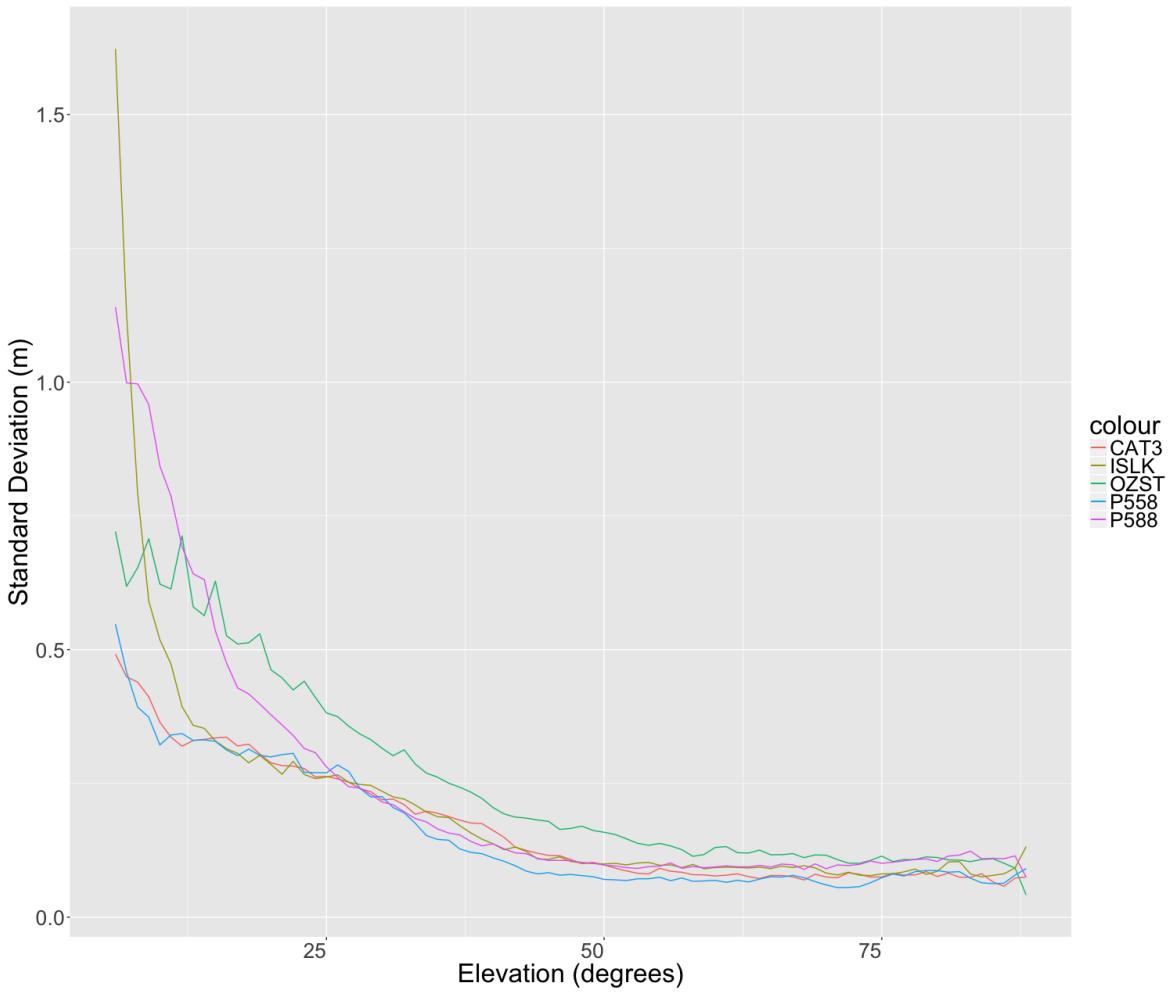

Figure 4.10: Standard Deviation computed by Antenna Calibration for every station for seven days and for the band E1E5, fuction of elevation.

4.3.5 Different Bands

Figure 4.11 shows the results of the Antenna Calibration for the receiver station called ISLK. Here, a comparison of the Antenna Calibration is shown using different combination of frequency bands.
As one can see, the Multipath is lower for E1- E5 and E1- E6. In this project E1- E5 combination is chosen to compute the code-carrier coherence. That is using Galileo E1 frequency band as the main, and Galileo frequency band E5 as the secondary frequency band.
The reason for this choice is that the effect of removing the ionosphere is stronger when the carrier frequencies are further away. Moreover, it is the linear combinations that is more widely used as it is explained in Section 3.1.1.

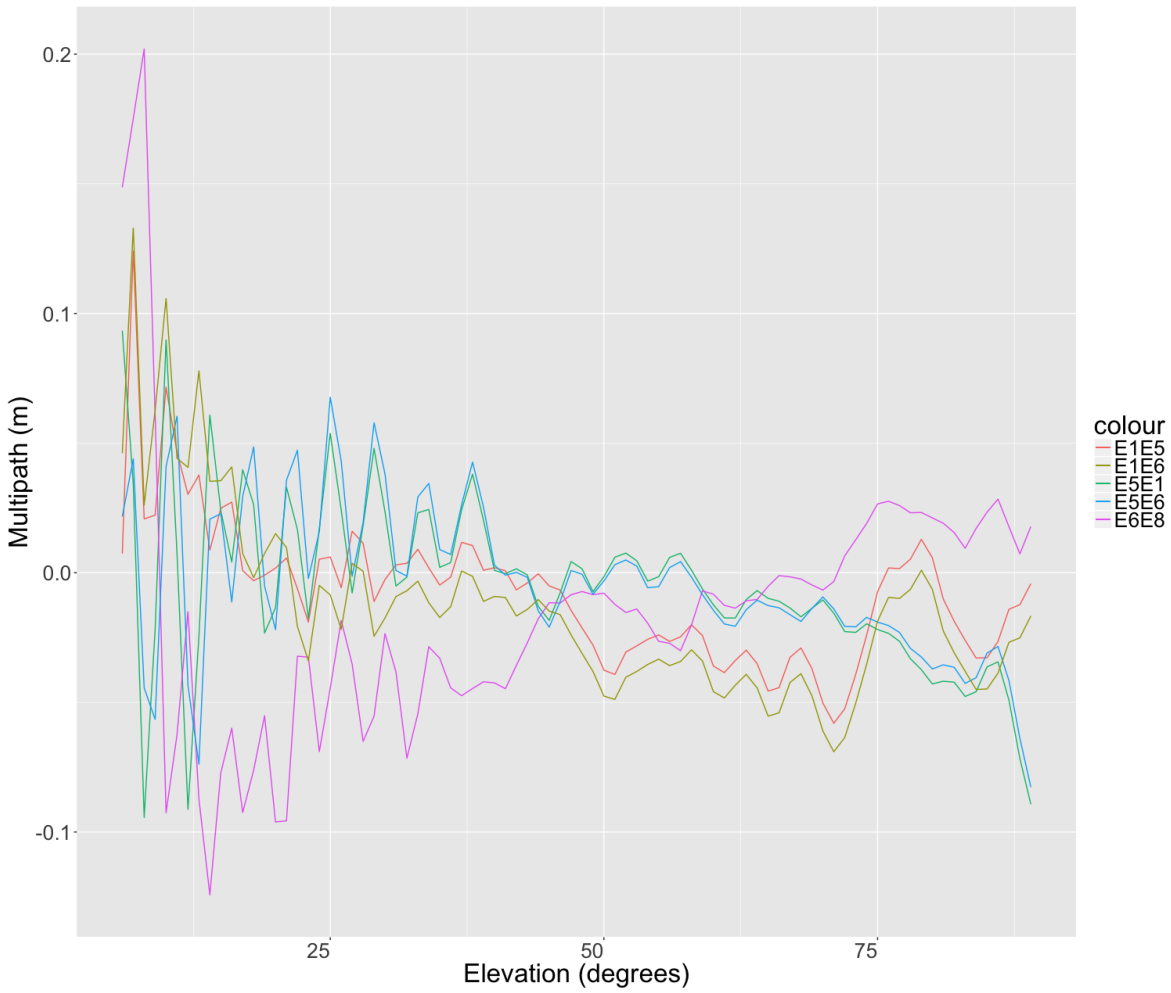

Figure 4.11: Multipath computed by Antenna Calibration for a different combinations of bands, fuction of elevation.

4.4 Code-Carrier Coherence

To compute the test part of the project, it is followed the procedure explained in Section 3.7.6 where it is used the leveled Multipath and the standard deviation. The standard deviation results are provided in Section 4.3.4 and leveled Multipath results will provided then in Section 4.4.1.

4.4.1 Leveled Multipath

The leveled Multipath is explained in Section 3.7.6 and is obtained from the Equation 3.35. For the following plots the E1-E5 frequency band combination is used and the results of using the five receiver station mentioned in Section 3.2 are shown. Then, from the the RINEX observable files of the day 92 of the year 2017 and from the SP3 files from the day 0 and 1 of the week 1943, the Figure 4.12 shows the leveled Multipath for the satellite 30. The Figure 4.13 shows the leveled Multipath for every satellite for the day 92 of 2017.

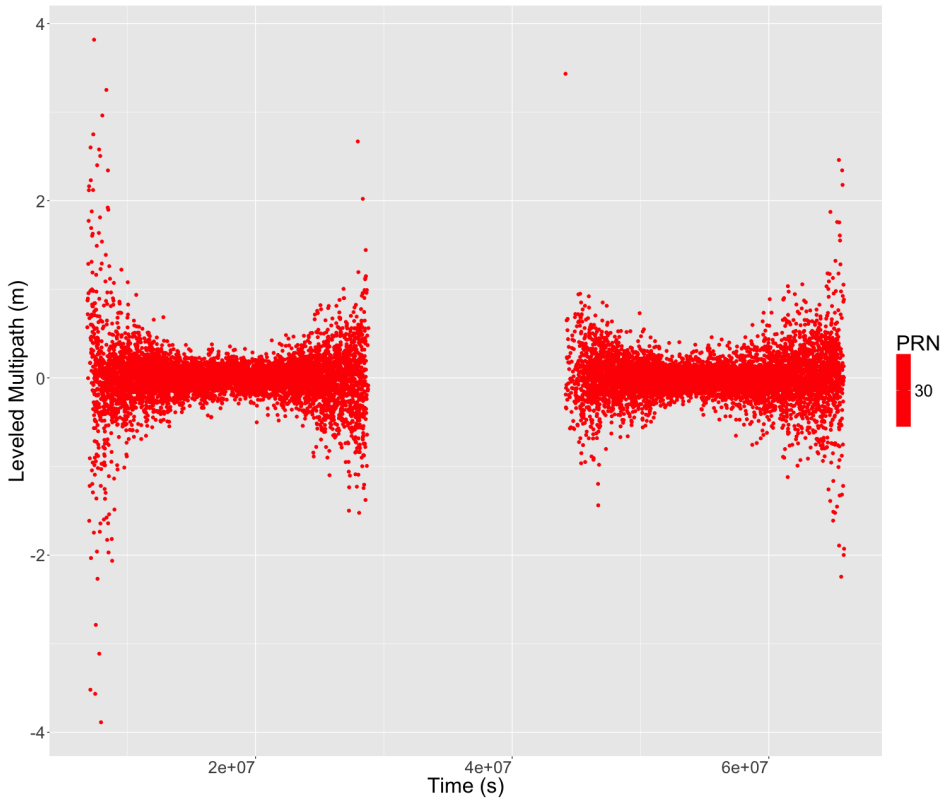

Figure 4.12: Leveled Multipath computed fuction of time for the day 92 of 2017.

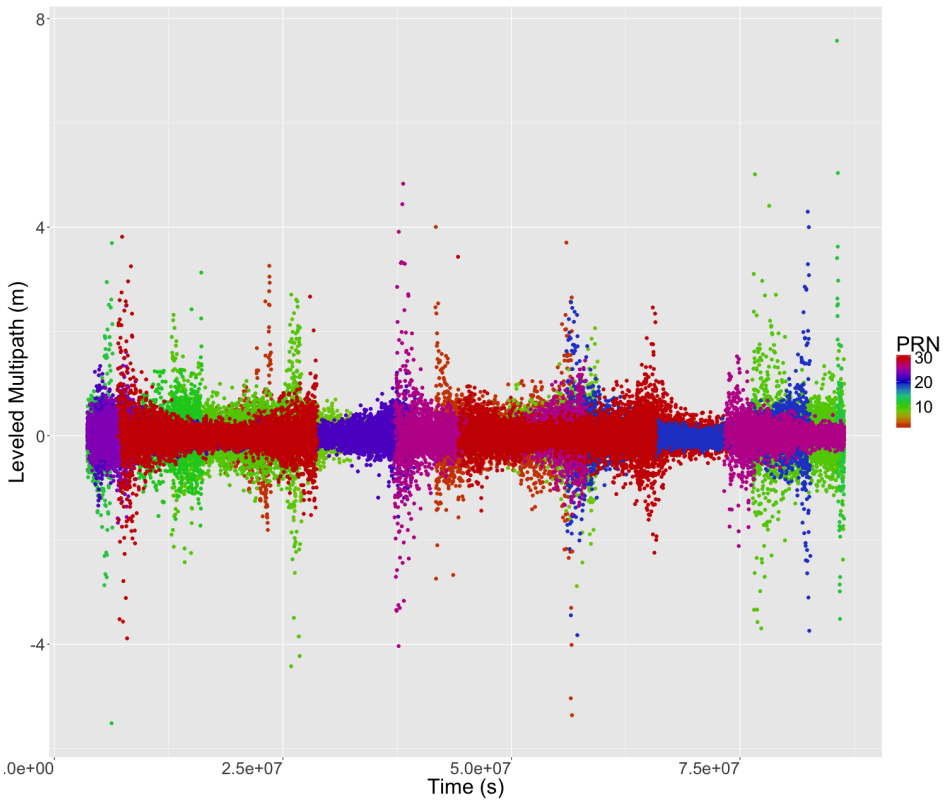

Figure 4.13: Leveled Multipath computed fuction of time for the day 92 of 2017.

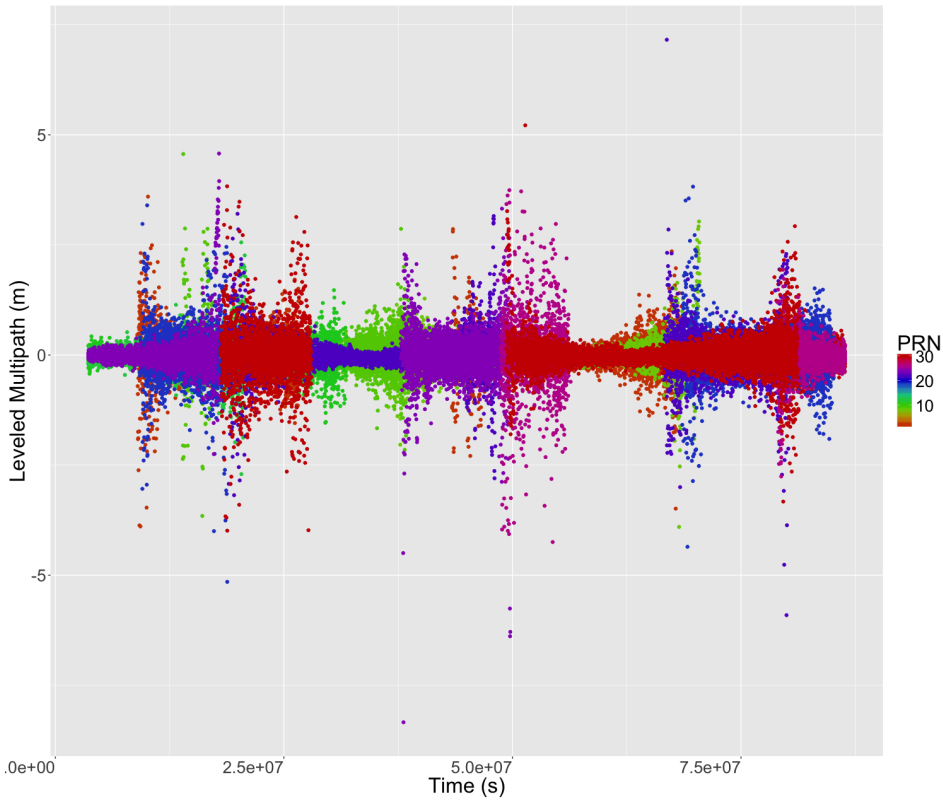

Figure 4.14: Leveled Multipath computed fuction of time for the day 106 of 2017.

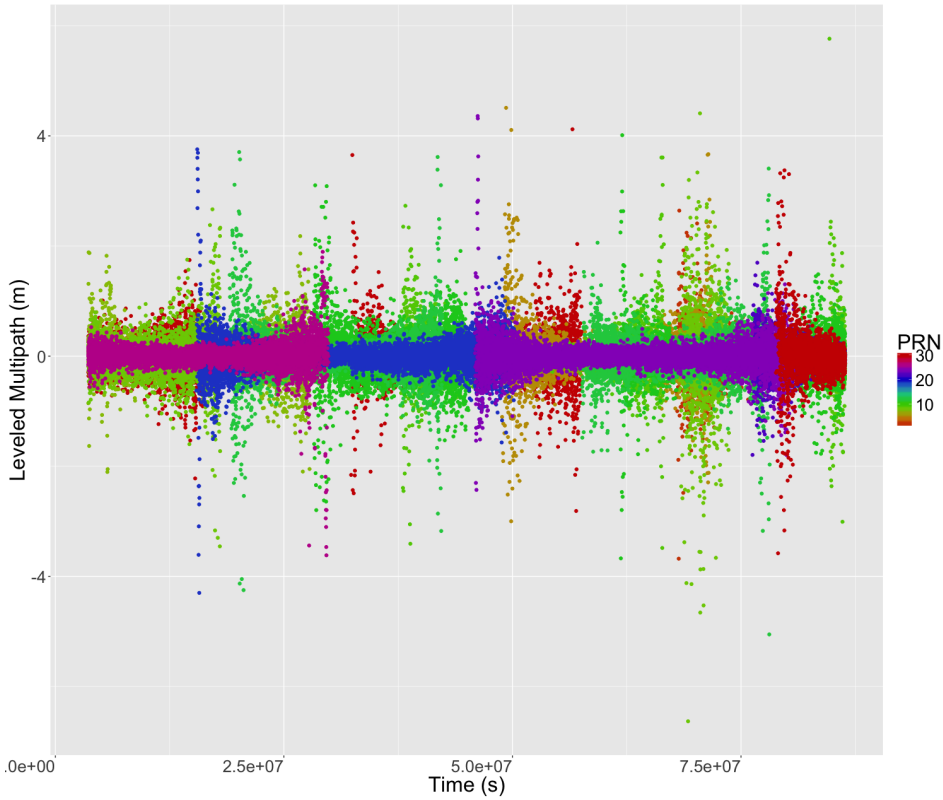

Figure 4.15: Leveled Multipath computed fuction of time for the day 151 of 2017.

The Figure 4.14 and the Figure 4.15 shows the leveled Multipath for the days 106 and 151. The Antenna Calibration is done for the days from 85 to 91. These results have been done to compare the performance of using the Antenna Calibration for a test done 1 day later (day 92), 1 week later (day 106) and some months later (day 151).

4.4.2 Code-Carrier Coherence

This section provides the representation of the Equation 3.26.

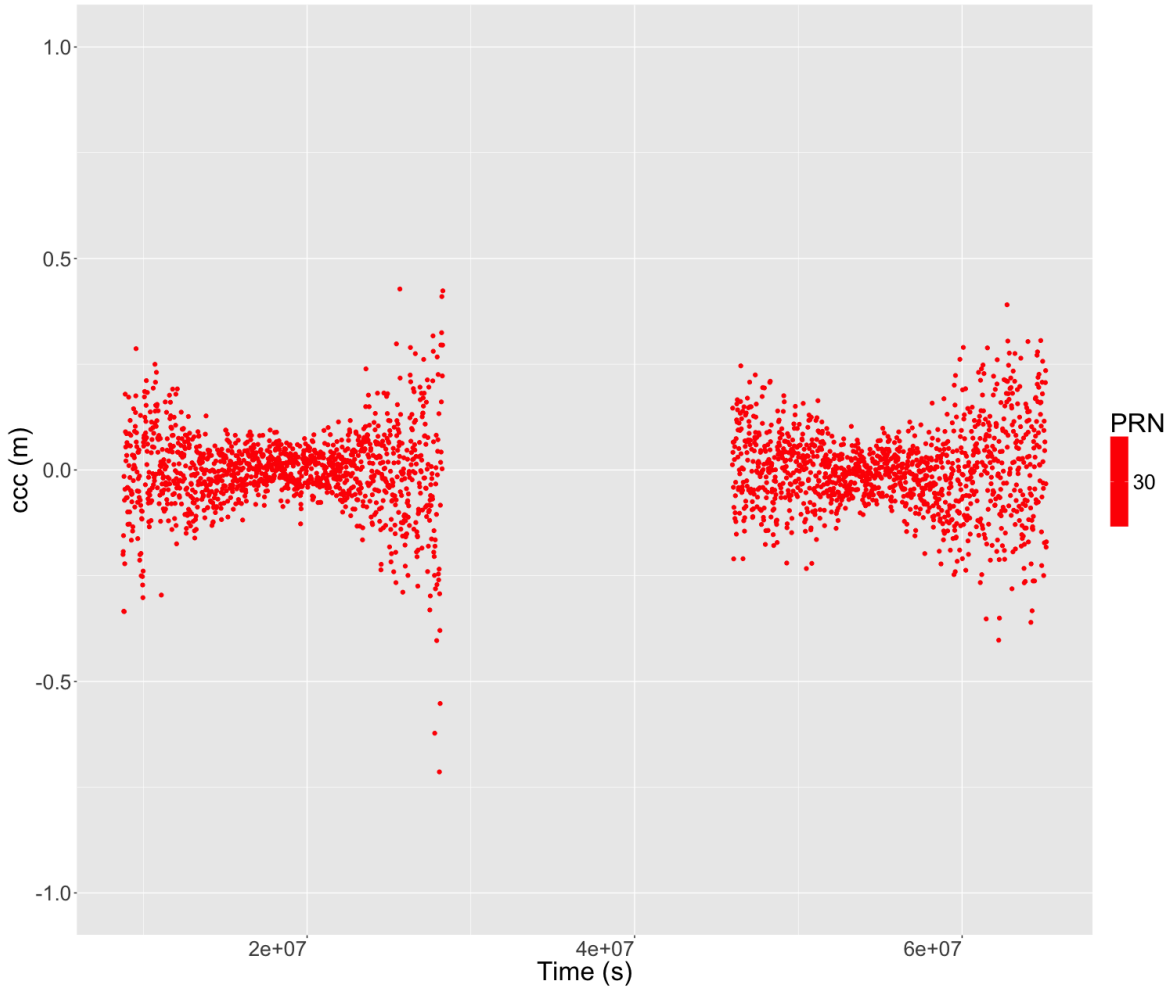

Figure 4.16: Code carrier coherence for the day 92 for the satellite 30.

The Figure 4.16 shows the code carrier coherence of the satellite 30, using the Leveled Multipath represented in the Figure 4.12 and the standard deviation between this Multipath and the Antenna Calibration shown for every station in the Figure 4.9.

The Figure 4.17 shows the code carrier coherence of every available station in the day 92 of 2017, using the Leveled Multipath represented in the Figure 4.13 and the standard deviation between this Multipath and the Antenna Calibration shown for every station in the Figure 4.9.

Figure 4.17: Code carrier coherence for the day 92 for every Galileo satellite.

The Figure 4.17 shows a maximum of 0.53 and a minimum of -0.57.
Then, the Figure 4.18 and the Figure 4.19 provide the code carrier coherence for the days 106 and 151. They use the Leveled Multipath shown in Figure 4.14 and in Figure 4.15.
As one can appreciate, the values at the beginning and at the end of the pass are more scattered than at higher elevations.
The satellite at the beginning of the pass raises from the horizon (because it is fixed an elevation mask of 5, the lowest elevation is supposed to be 5) and increases its elevation up to a maximum and then go down until it disappears again into the horizon.

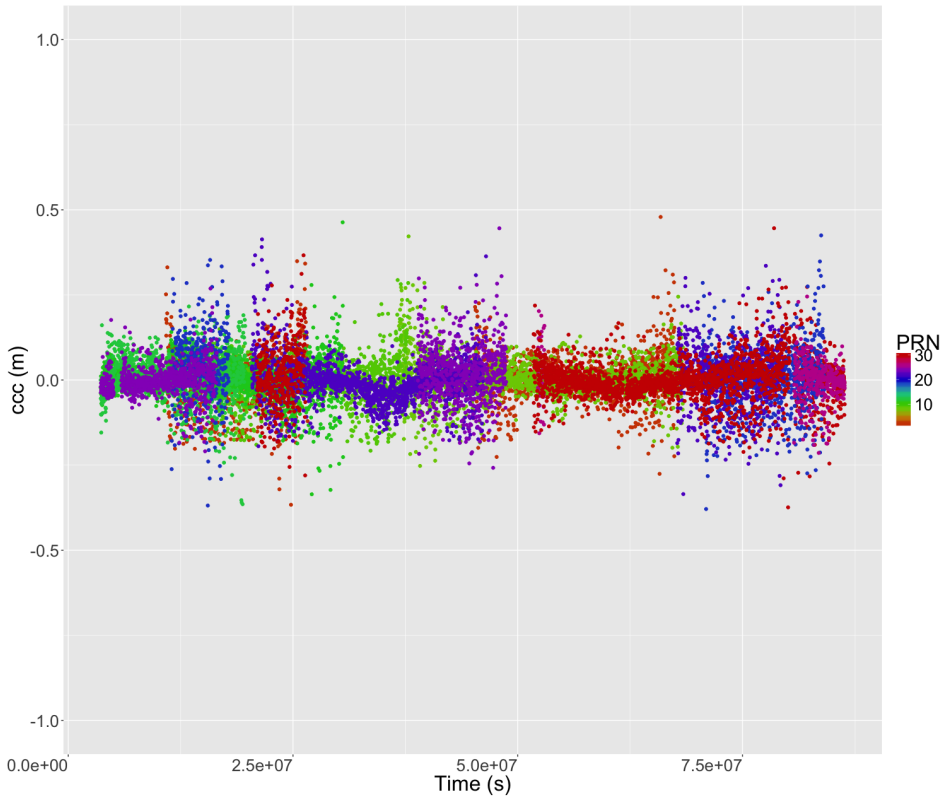

Figure 4.18: Code carrier coherence for the day 106.

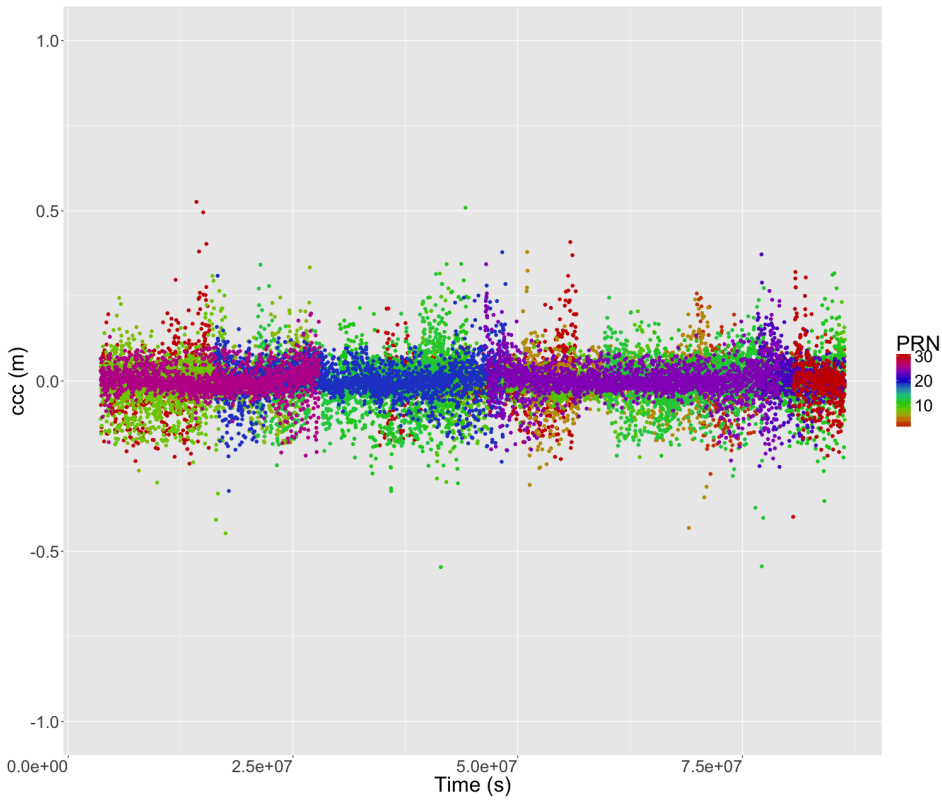

Figure 4.19: Code carrier coherence for the day 151.

The maximum of the values shown in Figure 4.18 and in Figure 4.19 is 0.53 and the minimum is -0.55.

4.4.3 Standard Deviation for Code-Carrier Coherence

To know how the code carrier coherence is evolving, it has been developed a Gaussian error propagation method explained in Section 3.6.1. Now, this Gaussian error propagation is shown as the standard deviation for CCC. Again, the results are obtained for the days 92, 106 and 151, for frequency band combination of E1-E5 and with the use of 5 receiver station.
To establish a threshold of a flag for knowing when a carrier code incoherence is happening, this plots can help to appreciate how big the values can be and if the Antenna Calibration of the week from the day 85 to the day 92 is valid for the day just after that week, the day of the week later and some day after some months. The Figure 4.20 shows the standard deviation of CCC shown in the Figure 4.16 for the satellite 30 for the day 92.

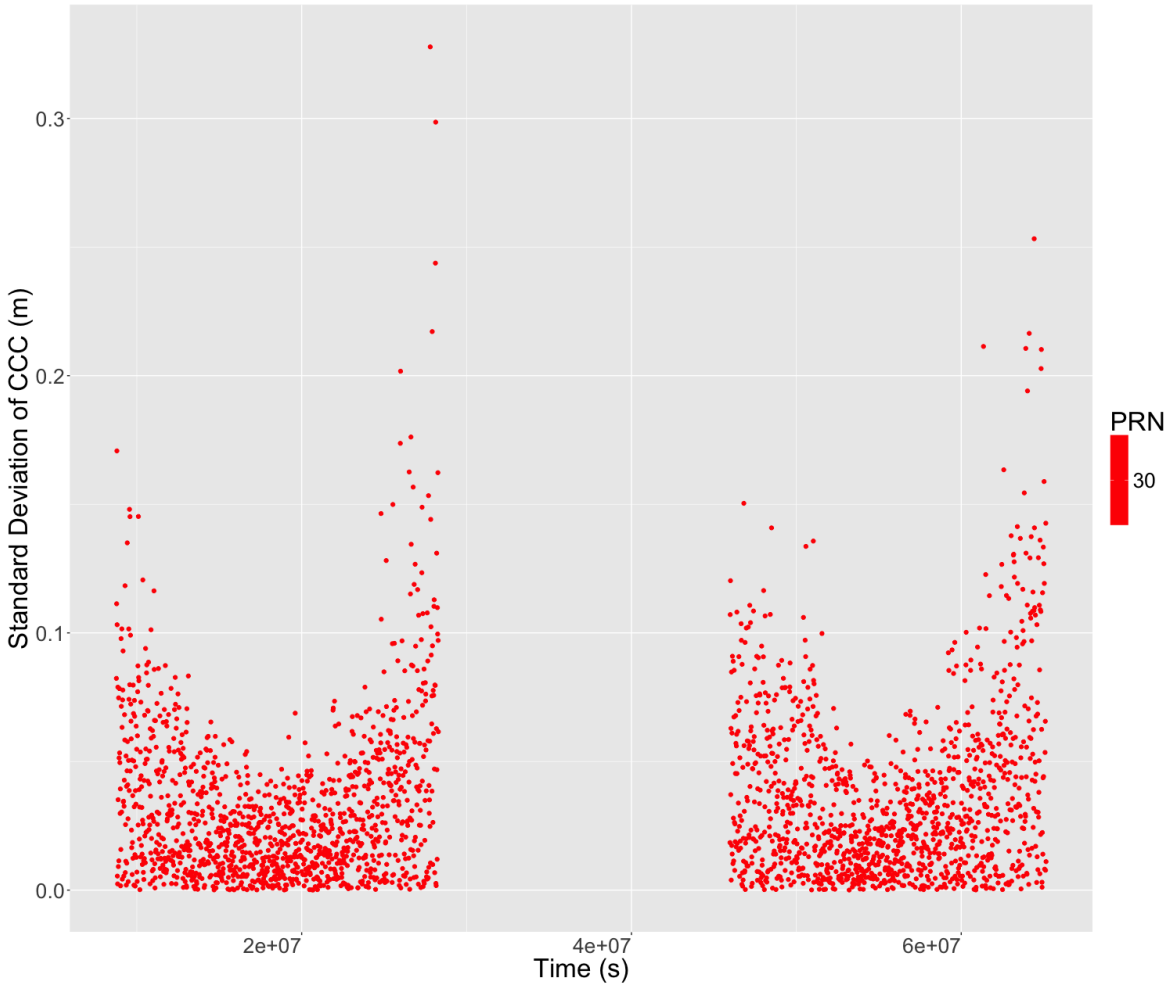

Figure 4.20: Standard deviation of CCC for day 92, for the satellite 30.

As one can see in the Figure 4.20 the values do not overpass the value of 0.4 and just one value overpass the value of 0.3.
The Figure 4.21 shows the standard deviation of CCC of the Figure 4.17.

There are only three values shown in the Figure 4.21 overpass the value of 0.3. One of those is from the satellite 30 also used for compute the Figure 4.20.

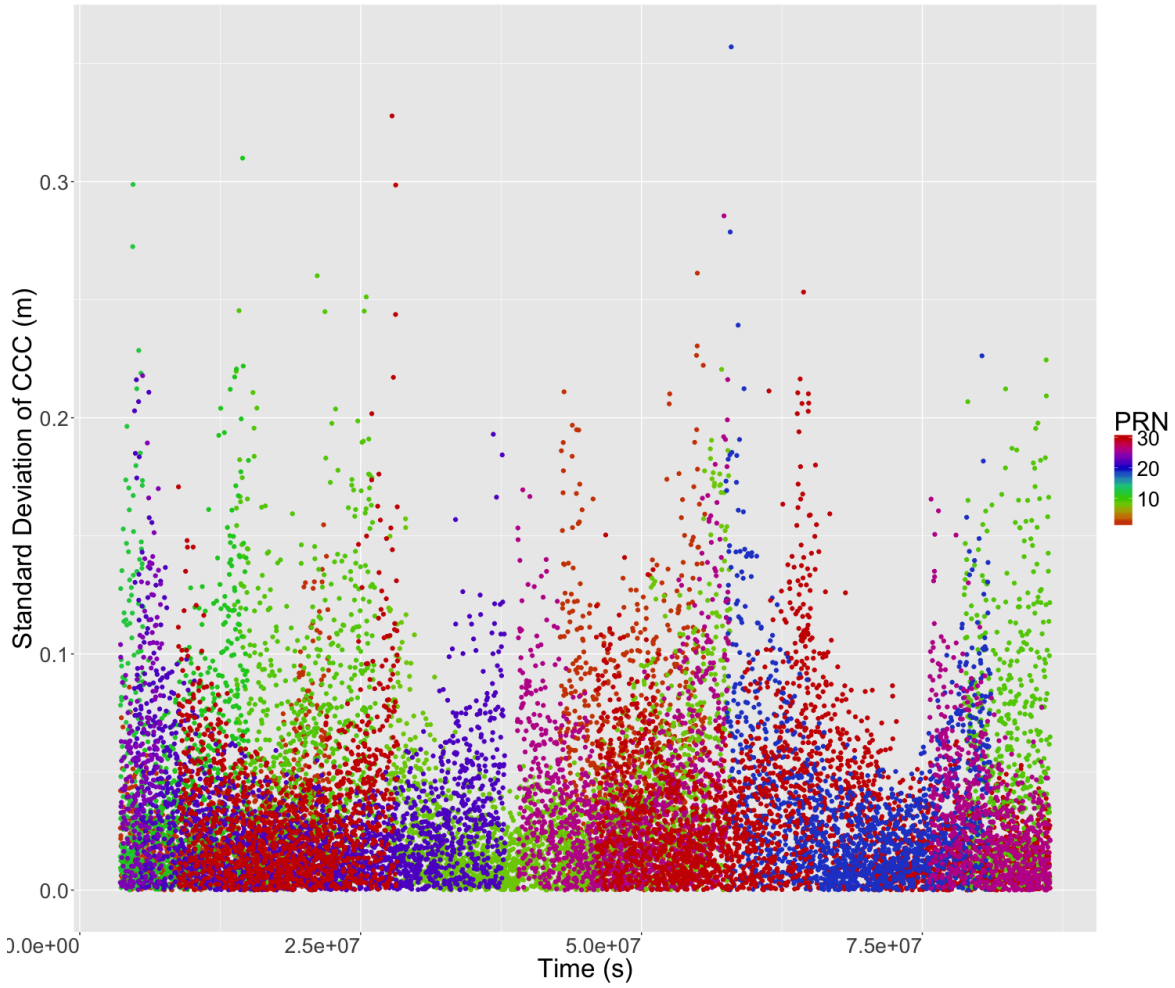

Figure 4.21: Standard deviation of CCC for day 92.

The following figures, Figure 4.22 and Figure 4.23 show the Standard deviation for every satellite for the days 106 and 151.

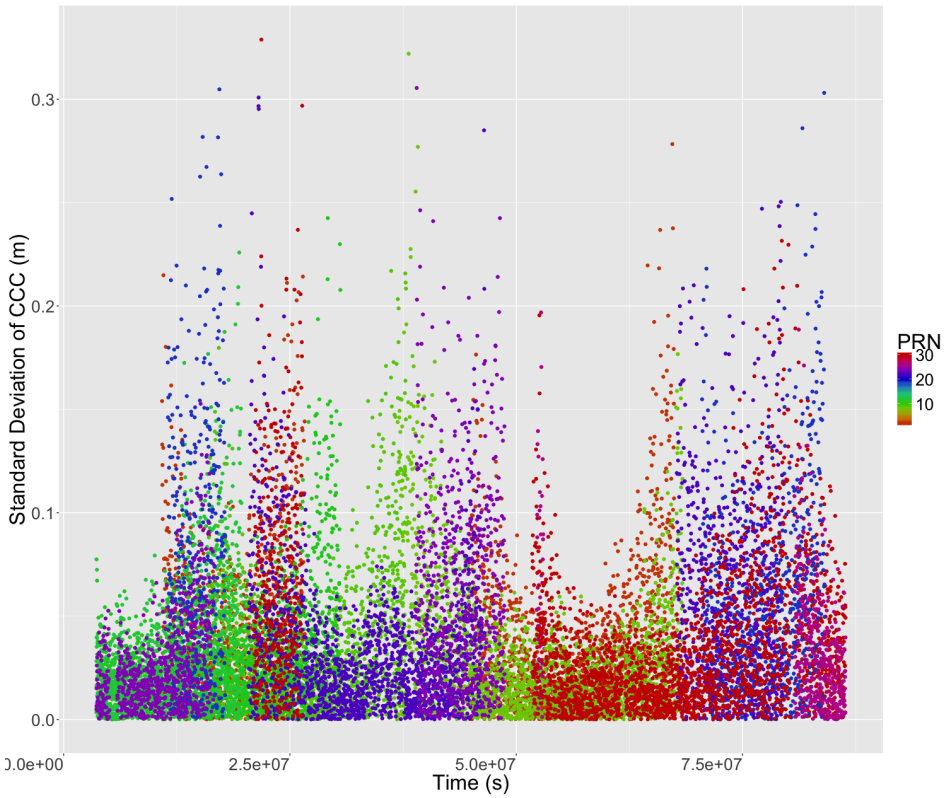

Figure 4.22: Standard deviation of CCC for day 106.

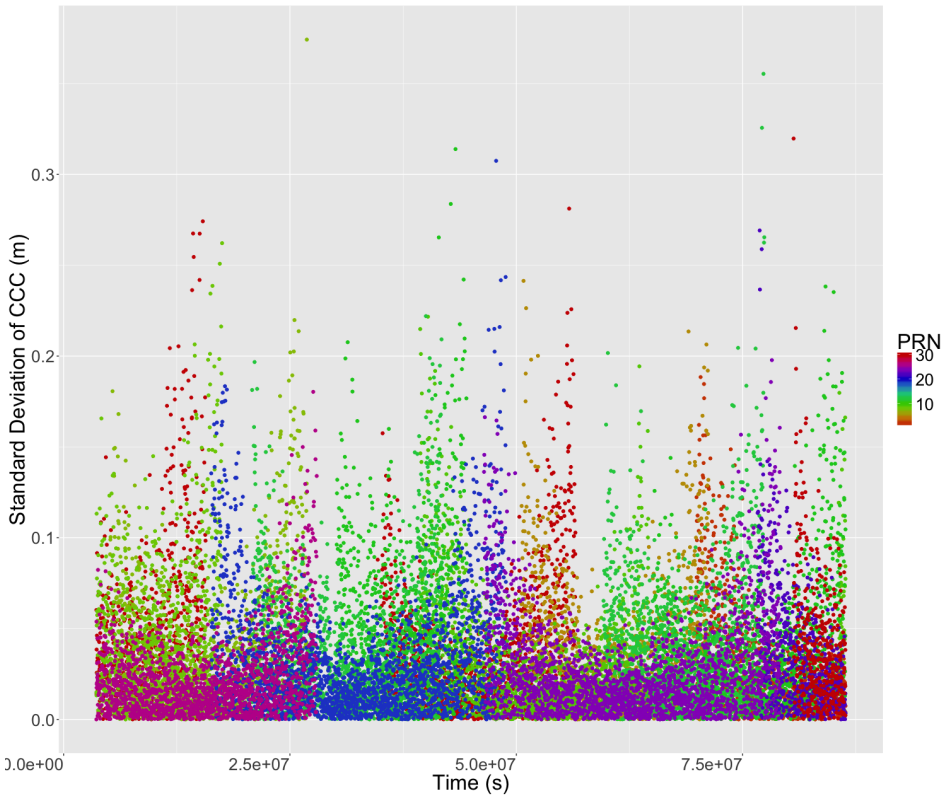

Figure 4.23: Standard deviation of CCC for day 151.

As one can see from the Figure 4.22 and Figure 4.23 is that just few points overpass the value of 0.3 and again, none of them overpass the value of 0.4.

4.5 Code-Carrier Coherence Threshold

After collecting the data shown in the Section 4.4.1, Section 4.4.2 and Section 4.4.3, it is possible to sum some results:

- There are higher values of standard deviation of Multipath at lower elevation, as it is shown in Section 4.4.1 where the dispersion at the beginning and at the end of the pass are higher;
- Leveled Multipath plots show most of the values near zero
- CCC is close to zero and with extreme values of around 0.53 as maximum and -0.57 as minimum;
- Standard Deviation of Multipath is close to zero;
- Only a few points of the Standard Deviation of Multipath are higher than 0.3 and have never been higher than 0.4.

Because the maximum Standard Deviation of CCC is around 0.3, this is considered the most extreme Standard Deviation for the procedure of computing the threshold. As it is explained in Section 3.8 the threshold for CCC is:

$$\Theta_{ccc} = \overline{CCC} \pm 2\sigma_{ccc} \tag{4.1}$$

Where Θ_{ccc} is the CCC threshold, $\overline{CCC}$ is the CCC mean and σ_{ccc} is the Standard Deviation for CCC. The mean can be considered zero as it shown in Figures 4.17, 4.18 and 4.19. Then, the threshold depends only in the extreme value of Standard Deviation which is considered 0.3

$$\Theta_{ccc} = \pm 2\sigma_{ccc} = \pm 0.6 \tag{4.2}$$

Chapter 5

Discussion

As a result of this project, the seeking of detecting the failure of a satellite because of code carrier incoherence has been concluded successfully.

5.1 Galileo Problems

During the project, Galileo data has been deeply searched, analyzed, rejected and finally used carefully. The main problems of using Galileo RINEX and SP3 files are based on the lack of completion of the Galileo system and the inviability of the systems in general and the receiver stations in particular to hold this ”recent” data. On the other hand, although the system requirements are covered to hold Galileo data, the benefits of using a GNSS underdeveloped may be for them not interesting at all. It depends on a balance of adventadges and disadvantages. One of the main disadvantages is adjust a global format as RINEX or SP3 from a specific receiver station to hold Galileo data for just obtaining the data from a few satellites.

5.1.1 Number of Satellites

As the Galileo system is underdeveloped, new satellites become available continuously. Although they can be operative, their data broadcast could be not good enough for the filters imposed to compute the CCC value. From the day 92 to the day 151 the number of satellites has increased in three, having 14 for the days 92 and 106 and 17 for the day 151. Fortunately the number of satellites will increase until having the Galileo constellation fully operative. Nicole, Zofia and Alexandre will be launch in the year 2017. [1]

5.2 Multipath as a mean of computation

As one of the main features of the project, Multipath has been computed along the project many times. That is because it appears in the main equation, Equation 3.26, as the mean of Multipath values, as standard deviation, and even as a result of the Antenna Calibration. For that, before knowing if the code-carrier coherence results have been properly computed it is interesting, firstly, test multipath results.
There is a big difference between the leveled Multipath, the raw Multipath and the discrete Multipath. While the raw Multipath shows the Multipath for a not binned elevation angles and the values are got with ambiguity,

[1] http://space.skyrocket.de/doc_sdat/galileo-foc.htm

leveled Multipath has no ambiguity because the mean has been removed and although discrete Multipath has included the ambiguity term, it is binned.
The results obtained from Antenna Calibration Multipath values are lower than 0.2 meters from the mean, which is in coherence with [30], which in ”Figure 2.14: Illustration of multipath bias per elevation bin” shows values lower than 1 meter ten years ago (2007).

5.2.1 Standard Deviation

On the other hand, the Multipath data used to compute the Antenna Calibration shows a Standard Deviation lower than 2 meters as it is shown in the Figure 4.10 can be compared with [31] where curve shape and maximum values ”Figure 2. L1 Pseudorange Errors for WAAS Network” of their are completely similar as the ones shown in this project. Their Standard Deviation for Multipath L1 is lower than 1.6 meters. In both cases, There are higher values of standard deviation of multipath at lower elevation, as it is shown in Section 4.4.1 where the dispersion at the beginning and at the end of the pass are higher. In this project E1 from Galileo has been used, which is the same band as the L1 from GPS.

5.3 Day of the year

As it is commented several times during the project, the test to compute the CCC value is done for three different days. The first one is the day after the week chosen to compute the Antenna Calibration, the second is the day just after a week after the Antenna Calibration and the third is some day after several months. All of them show results quite similar, for that if there are no any appreciable physical change in the terrain nor in the own receiver station, the same Antenna Calibration can be used for longer time than some days or a week as long as it is safe to trust that there are no changes in the above mentioned.

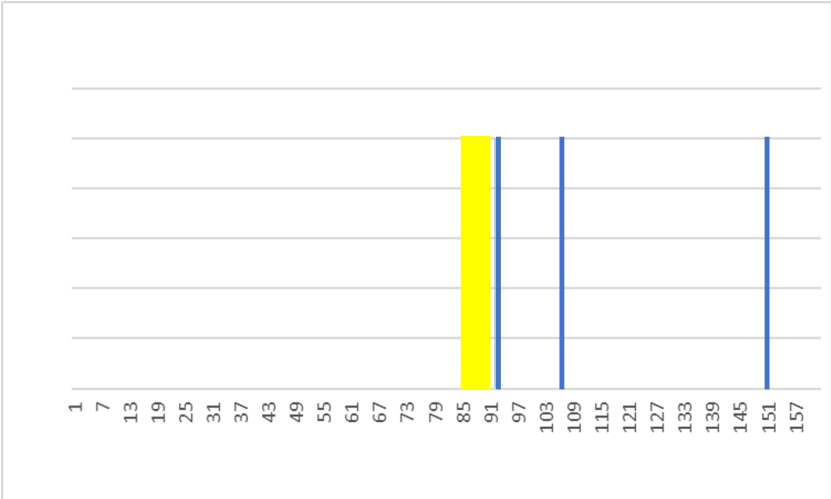

Figure 5.1: Days used to compute the CCC test

In Figure 5.1 it is shown the days used as input of the Antenna Calibration in yellow (from the day 85 to the day 91) and the days 92, 106 and 151 used for the text in color blue.

5.4 Choosing a threshold for Code-Carrier Coherence

As the CCC is a value with a very low mean and great dispersion at low elevation angles, an error propagation is used to measure how to detect whenever the values go away from the normality. In that case, it is said that a code-carrier incoherence exists. To know that, it is a good idea to establish a threshold to know if a value, or a group of them overpass it, and turn on a flag to alert the dependent authority. As it is shown in [28] the aim is to establish a threshold which depends on the user differential range error values that are broadcast by each satellite.
This threshold value have to be computed for every new work which uses different receiver stations and satellites and times. Indeed the threshold value depends on the number of receiver stations. As it is shown in Section 3.6.1 the CCC Standard Deviation is inversely proportional of the square root of the number of satellites. That means that the threshold would decrease with the number of receiver stations.

The threshold obtained in Section 4.5 is $\pm$ 0.6. This value means that if a CCC value along the time goes higher than 0.6 or lower than -0.6 a flag would become on. This threshold values gains coherence because there is no CCC value which exceed those bounds in the plots shown in Section 4.4.2. As it is said there, the maximum is 0.53 and the minimum -0.57.

Chapter 6

Conclusion

This project has developed a method to find code-carrier incoherences in satellites. Specifically, this method has been applied in the current underdeveloped Galileo system. The lack of research around Galileo compared to the whole GPS ecosystem remarks the importance and innovation of the method described. Throughout the project, ambiguities in the carrier phase observable equation have been cleared and arches have been detected, to accomplish that, data with cycle slips has been removed and Multipath has been computed, analyzed and used to track coherence. The E1 frequency band is used as the main frequency and the E5 frequency band as a secondary frequency.

The first part of the tool consists in obtaining the Antenna Calibration of different receiver stations. This Antenna Calibration gets its value by measuring a mean multipath in a seven-day time frame, which allows to reduce out of range value effects and to know how the multipath performance is along elevation angle. This Antenna Calibration is the base to compute the variations in a posteriori computed leveled Multipath. This procedure is done for five different receiver stations. Then, the second part of the tool computes the leveled Multipath for a specific day and its Standard Deviation from the Antenna Calibration. The code-carrier incoherence tracking can be done thanks to the plots of CCC where the values can be considered correct if never exceed from the normal environment. But a better method is to compute the error propagation of CCC and set a threshold. To sum up, the method consists on establish from the error propagation of CCC, a flag which would become ”on” whenever this feared event appears.

It is studied for how long is permissible to use the Antenna Calibration for a specific week. It has been chosen a day, a week and some months later and the results can be considered good for this three test days. In conclusion, if there is no significant change in the receiver station nor in the satellite, there would not be problem of using the same week to compute the Antenna Calibration for a test day method some months later.

The fact of using five receiver stations located in a small area results in a dependency on the elevation. For every single time, every satellite is at similar elevation angle of view for every receiver station chosen in this work. That makes the computation of CCC easier due to the fact that every receiver station is viewing the same satellites. On the other hand, the number of receiver stations is important to establish the threshold, as it is explained in this project, the Standard Deviation of CCC is inversely proportional of the square root of the number of satellites. In conclusion, increasing the number of receiver stations would decrease the threshold. What makes sense because the CCC values become noisy reduced.

The threshold chosen is $\pm$ 0.6, what means that any time a CCC value overpasses it, a possible code-

carrier incoherence can be occurring or there is a value which is out of range. On the other hand, if actually this feared event is occurring, there would be many values overpassing the selected threshold.

In my personal opinion, this value should evolve with experience, depending on how many out of range values are during a period of time, and how better it is to receive many false alarms versus catching this feared event on time (if the threshold is lower than the optimal value, there will be many false alarms, but if the threshold is higher, it will be difficult to catch the code-carrier incoherence on time).

Chapter 7

Future Works

This chapter has been done to include every task which has not been done during the dissertation period. There are some task wh ich can improve or expand the content of this project, but they are hard enough tasks to carry out their own dissertation or because the required tools to perform them are not currently available.

- The tool performed in this project can support every different combination of frequency bands. To analyze the results of the code-carrier coherence method it is used only the combination between E1 and E5 for Galileo. It would be interesting to research the results for other combination of bands to other services apart of the open-service of Galileo[1];
- Unfortunately, there are only few satellites available from the Galileo system. This project could use every satellite in a near future whenever they will be fully operative[2]. This is a very interesting point to test the whole system in case there is any damaged satellite;
- The tool has two steps: first, it computes the antenna calibration from a period of time and then it does a test to compute if there are any code-carrier incoherence in a satellite. An enhance tool would be one which is capable to do it in real time. There is an advantage from this project having the ability to know in real time if there are any damaged satellite because of the code-carrier incoherence;
- Rinex version 2 is used as the input of the tool, as it is explained that is because of the limitation of the GPSTK[3]. This open source library is being improved continuously, thus the tool could be improved when those limitations will disappear.

[1] https://www.gsa.europa.eu/news/prs-huge-potential-europe
[2] http://gpsworld.com/directions-2017-the-year-of-galileo/
[3] https://www.unavco.org/whats-hot/2016/2016-03-25-GDS%20Plan%20for%20GNSS%20Modernization%20Data%20Formats%20and%20Preprocessing%20Tools.html

Bibliography

[1] J. Kouba, Geodetic Survey Division, and Natural Resources Canada, *A guide to using international GNSS service (IGS) products*. 615 Booth Street Ottawa, Ontario K1A 0E9 Canada, September 2015.

[2] U. D. of Commerce, "National oceanic and admospheric administration." https://www.ngs.noaa.gov/CORS_Map/.

[3] C. Gunther and P. Henkel, "Gnss for landing systems and carrier smoothing techniques."

[4] G. Sateesh KumarG. Sasi Bhushana Rao, "Gps errors and biases," April 2013.

[5] J. Rantakokko, P. Handel, F. Eklof, B. Boberg, M. Junered, D. Akos, I. Skog, and H. Bohlin, "Positioning of emergency personnel in rescue operations," 2007.

[6] B. Tolman, R. B. Harris, T. Gaussiran, D. Munton, J. Little, R. Mach, S. Nelsen, and B. Renfro, "The GPS Toolkit: Open Source GPS Software," in *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation*, (Long Beach, California), September 2004.

[7] L. Werner Gurtner (Astronomical Institute, University of Berne), "Rinex: The receiver independent exchange format, version 2.11."

[8] L. Werner Gurtner (Astronomical Institute, University of Berne), "Rinex: The receiver independent exchange format, version 3.03." https://cddis.nasa.gov/Data_and_Derived_Products/GNSS/RINEX_Version_3.html.

[9] L. Privat, "Ti announced new gps chip, cost under 5 dollars," *GPS Business News*, 2007.

[10] Department of Defense, Department of Homeland Security, and Department of Transportation, "System parameters," in *Federal Radionavigation Plan*, 2008.

[11] J. V. Sickle, *GPS for Land Surveyors, Fourth Edition*, ch. 2. CRC Press, 2015.

[12] M. Sahmoudi and R. Landry, "Multipath mitigation techniques using maximum likelihood principle," *Inside GNSS*, vol. 3, no. 8, pp. 24–29, December 2008.

[13] J. Wang, X. Lu, W. Jing, and Y. Bai, "Satellite navigation signal code-carrier coherence analysis," in *Signal Processing, Communications and Computing (ICSPCC)*, IEEE, August 2014.

[14] Information and analysis center for positioning, navigation and timing, "Glonass history." https://www.glonass-iac.ru/en/guide/.

[15] E. G. N. S. S. Agency, "Galileo is the european global satellite-based navigation system." https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system.

[16] "Gnss market report issue 5." https://www.gsa.europa.eu/system/files/reports/gnss_mr_2017.pdf, 2017.

[17] G. M. Moran, "Future gnss: Improved signals and constellations," 2014.

[18] V. K. D. S. Bernd Eissfeller, Gerald Ameres, "Performance of gps, glonass and galileo," 2007.

[19] "Open service, quarterly performance report," January-March 2017.

[20] "Open service service definition document," December 2016.

[21] M. s. Grewal, M. Petovello, and G. Lachapelle, "Waas functions and differential biases," *InsideGNSS*, May/June 2008.

[22] B. Parsons, J. Little, B. Tolman, B. Renfro, S. Nelsen, D. Munton, and B. Harris, *Climbing the GNSS Hill: Lessons from the evolution of the GPSTk*, vol. 3, pp. 2032–2043. Institute of Navigation, 2013.

[23] International GNSS service, "IGS format." http://kb.igs.org/hc/en-us/articles/201096516-IGS-Formats.

[24] ESA and Navipedia, "Interfaces and protocols." http://www.navipedia.net/index.php/Interfaces_and_Protocols.

[25] Andrew SimskyJean-Marie Sleewaegen, "Performance assessment of galileo ranging signals transmitted by gstb-v2 satellites," tech. rep., September 2006.

[26] C. M. Andrew Simsky, Jean-Marie Sleewaegen, "Performance assessment of galileo ranging signals transmitted by gstb-v2 satellites," *ION GNSS, Fort Worth, USA. Session F3*, 2006.

[27] L.-S. Lin, "Applying multipath template technique to improve the real-time ionospheric delay estimation accuracy," *ION GPS 2002, 24-27*, September 2002.

[28] P. Shloss, R. E. Phelts, T. Walter, and P. Enge, "A simple method of signal quality monitoring for waas lnav/vnav," 2002.

[29] Harvard, "A Summary of Error Propagation." http://ipl.physics.harvard.edu/wp-uploads/2013/03/PS3_Error_Propagation_sp13.pdf, 2007.

[30] J. P. Weiss, "Modeling and characterization of multipath in global navigation satellite system ranging signals," 2007.

[31] K. Shallberg, P. Shloss, and E. Altshuler, "Waas measurement processing, reducing the effect of multipath," 2011.

Appendix A

R Programming language

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulation where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.[1] R is a programming language and environment commonly used in statistical computing, data analytics and scientific research. It is one of the most popular languages used by statisticians, data analysts, researchers and marketers to retrieve, clean, analyze, visualize and present data. Due to its expressive syntax and easy-to-use interface, it has grown in popularity in recent years.[2]

A.1 RStudio

RStudio is an active member of the R community. Its belief of freedom and open source data analysis software is a foundation for innovative and important work in science, education, and industry.

Figure A.1: Logotype of RStudio

The many customers who value its professional software capabilities help them contribute to this community.[3] In a few words, RStudio is a new open-source IDE for R. It has interesting features for both new and experienced R developers including code completion, execute from source, searchable history, and support for authoring Sweave documents.
RStudio runs on all major desktop platforms (Windows, Mac OS X, Ubuntu, or Fedora) and can also run as a server which enables multiple users to access the IDE using a web browser.

[1] https://www.r-project.org/about.html
[2] https://www.programiz.com/r-programming
[3] https://www.rstudio.com

A.2 Plotting Results

Due to the big amount of data which have been managed in this projects, it has been necessary to use a tool to extract the information from the output of the program created by C++ Programming Language. The graphs plotted in the results have been done by this program called RStudio.

A.3 R Programming code

```
library(ggplot2)
library(timeDate)
library(timeSeries)
library(fImport)


ContinuousMuvsEl <- read.table("/Users/Guillermo/Documents/master-thesis/calc/MultipathVSElevation.txt", header = TRUE)
ggplot(ContinuousMuvsEl, aes(x=Elevation, y=Multipath, color=PRN)) + geom_point()
+ scale_colour_gradientn( colours = rainbow(10,s=1, v=0.8, start=0, end= max(1,9/10),alpha=1))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="Multipath (m)")

DiscreteMuvsEl <- read.table("/Users/Guillermo/Documents/master-thesis/calc/MultipathVSElevationWithStandardDeviation.txt", header = TRUE)
ggplot(DiscreteMuvsEl, aes(x=Elevation, y=Multipath, color=PRN)) + geom_point()
+ scale_colour_gradientn( colours = rainbow(10,s=1, v=0.8, start=0, end= max(1,9/10),alpha=1))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="Multipath (m)")
ggplot(DiscreteMuvsEl, aes(x=Elevation, y=StandardDeviation, color=PRN)) + geom_point()
+ scale_colour_gradientn( colours = rainbow(10,s=1, v=0.8, start=0, end= max(1,9/10),alpha=1))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="Standard Deviation (m)")

stDevdMudElData <- read.table("/Users/Guillermo/Documents/master-thesis/calc/StandardDeviationDMuDEl.txt", header = TRUE)
ggplot(stDevdMudElData, aes(x=Elevation, y=StandardDeviation)) + geom_point()
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="Standard Deviation(m)")

dMudElData <- read.table("/Users/Guillermo/Documents/master-thesis/calc/DMuDElWithPRN.txt", header = TRUE)
ggplot(dMudElData, aes(x=Elevation, y=DMuDEl, color=PRN)) + geom_point() + coord_cartesian(ylim=c(-2.5, 2.5))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="d(Multipath)/d(Elevation) (m/degrees)")
+ scale_colour_gradientn( colours = rainbow(10,s=1, v=0.8, start=0, end= max(1,9/10),alpha=1))

AntennaCalibrationDays <- read.table("/Users/Guillermo/Documents/master-thesis/data/AntennaCalibrationDays.txt", header = TRUE)
ggplot(AntennaCalibrationDays, aes(x=Elevation)) + geom_line(aes(y=n6, colour ="n6")) + geom_line(aes(y=n5, colour="n5"))
+ geom_line(aes(y=n7, colour = "n7"), size =1.5) + geom_line(aes(y=n4, color="n4"))+ geom_line(aes(y=n3, color="n3"))
+ geom_line(aes(y=n2, color="n2"))+ geom_line(aes(y=n1, color="n1"))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))+ labs(x="Elevation (degrees)", y="Multipath (m)")

AntennaCalibrationStations <- read.table("/Users/Guillermo/Documents/master-thesis/data/AntennaCalibrationDifferentStations.txt", header = TRUE)
ggplot(AntennaCalibrationStations, aes(x=Elevation)) + geom_line(aes(y=OZST, colour = "OZST"))
+ geom_line(aes(y=CAT3, colour = "CAT3")) + geom_line(aes(y=ISLK, colour="ISLK"))+ geom_line(aes(y=P558, color="P558"))
+ geom_line(aes(y=P588, color="P588")) + theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="Multipath (m)")

AntennaCalibrationBands <- read.table("/Users/Guillermo/Documents/master-thesis/data/BandsComparison.txt", header = TRUE)
ggplot(AntennaCalibrationBands, aes(x=Elevation)) + geom_line(aes(y=E1E5, colour = "E1E5"))
+ geom_line(aes(y=E5E1, colour = "E5E1")) + geom_line(aes(y=E6E8, colour="E6E8"))+ geom_line(aes(y=E5E6, color="E5E6"))
+ geom_line(aes(y=E1E6, color="E1E6")) + theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="Multipath (m)")

StDev <- read.table("/Users/Guillermo/Documents/master-thesis/data/StandardDeviation.txt", header = TRUE)
ggplot(StDev, aes(x=Elevation)) + geom_line(aes(y=OZST, colour = "OZST")) + geom_line(aes(y=CAT3, colour = "CAT3"))
+ geom_line(aes(y=ISLK, colour="ISLK"))+ geom_line(aes(y=P558, color="P558"))+ geom_line(aes(y=P588, color="P588"))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="Standard Deviation (m)")

dAndEl <- read.table("/Users/Guillermo/Documents/master-thesis/calc/dAntdEl.txt", header = TRUE)
ggplot(dAndEl, aes(x=Elevation, y=dAntdEl)) + geom_point()
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Elevation (degrees)", y="d(Antenna Calibration)/d(Elevation) (m/degrees)")

CCC <- read.table("/Users/Guillermo/Documents/master-thesis/data/CCC.txt", header = TRUE)
ggplot(CCC, aes(x=Time, y=ccc, color=PRN)) + geom_point()+ scale_colour_gradientn( colours = rainbow(10,s=1, v=0.8, start=0, end= max(1,9/10),alpha=1))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))+ labs(x="Time (s)", y="ccc (m)")
+ coord_cartesian(ylim=c(-1, 1))
ggplot(CCC, aes(x=Time, y=StDev, color=PRN)) + geom_point()+ scale_colour_gradientn( colours = "red")
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
+ labs(x="Time (s)", y="Standard Deviation of CCC (m)")  #+ coord_cartesian(ylim=c(-1, 1))

MuMinusMEan <- read.table("/Users/Guillermo/Documents/master-thesis/data/MuMinusMean.txt", header = TRUE)
ggplot(MuMinusMEan, aes(x=Time, y=Mu, color=PRN)) + geom_point()
+ scale_colour_gradientn( colours = rainbow(10,s=1, v=0.8, start=0, end= max(1,9/10),alpha=1))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))
```

```
+ labs(x="Time (s)", y="Leveled Multipath (m)")

StDevLev <- read.table("/Users/Guillermo/Documents/master-thesis/data/standardDeviationLeveledMu0.txt", header = TRUE)
ggplot(StDevLev, aes(x=Time, y=ccc, color=PRN)) + geom_point()
+ scale_colour_gradientn( colours = rainbow(10,s=1, v=0.8, start=0, end= max(1,9/10),alpha=1))
+ theme(text = element_text(size=25), axis.text.x = element_text(angle=0, hjust=1))+ labs(x="Time (s)", y="ccc (m)")
+ coord_cartesian(ylim=c(-1, 1))
```

Appendix B

C++ Programming code

Here it is attached the C++ programming code use to develop the tool. It is used many guides and online documentation. It is impossible to cite all of them but to rebuild this results, there are some forums[1] where to ask for help for understanding some procedures.[2]

B.1 Antenna Calibration part

Copyright. All rights reserved

B.2 Test Part

```
#include <iostream>
#include <iomanip>
#include <fstream>

// Class for handling satellite observation parameters RINEX files
#include "Rinex3ObsBase.hpp"
#include "Rinex3ObsData.hpp"
#include "Rinex3ObsStream.hpp"


// Class to store satellite precise navigation data from .SP3 files
#include "SP3EphemerisStore.hpp"
#include "Position.hpp"

#include <numeric>

// Class to filter out satellites without required observables
#include "RequireObservables.hpp"

// Class to compute linear combinations
#include "ComputeLinear.hpp"

// This class pre-defines several handy linear combinations
#include "LinearCombinations.hpp"
// Used to decimate data. This is important because RINEX observation
// data is provided with a 30 s sample rate, whereas SP3 files provide
// satellite clock information with a 900 s sample rate.
#include "Decimate.hpp"

#include "Dumper.hpp"
#include "boost/tuple/tuple.hpp"

#include <array>
//class to detect Cycle Slips
#include <OneFreqCSDetector.hpp>


using boost::tuple;
using namespace std;
using namespace gpstk;
```

[1] http://www.gpstk.org/bin/view/Forum/WebHome
[2] http://www.gpstk.org/bin/view/Documentation/WebHome

```
typedef tuple<CivilTime, int, double> Tuple3;
typedef vector< Tuple3 > vectorOfTuple3;
typedef tuple<CivilTime, int, double, double> Tuple4;//civTime, elevation, Multipath, standard deviation
typedef vector< Tuple4 > vectorOfTuple4;

//Global variables
vectorOfTuple4 Multipath0;
vectorOfTuple4 Multipath1;
vectorOfTuple4 Multipath2;
vectorOfTuple4 Multipath3;
vectorOfTuple4 Multipath4;
vector<pair<int, double>> AntennaCalibration0;
vector<pair<int, double>> AntennaCalibration1;
vector<pair<int, double>> AntennaCalibration2;
vector<pair<int, double>> AntennaCalibration3;
vector<pair<int, double>> AntennaCalibration4;



//Declaration of Functions
vector <pair<int, double>> getFromFile(string file);
void computeMultipath(string obsFile, string SP3File1, string SP3File2, int SV, int iValue, int jValue, int i);
vectorOfTuple3 computeMuMinusMeanMu(vectorOfTuple3 MuEl);
vector < tuple<CommonTime, double, double>> computeccc();
void pushBackMultipathAsTupleOf4(vectorOfTuple3 MuMinusMeanMu, int loopCounter);
void saveStandardDeviation(int PRN);

int const numberObsFiles = 5;

int main() {
    vector<pair<int, double>> AntennaCalibration;

    ofstream myFile;
    myFile.open ("ccc.txt",ios::trunc);
    myFile<< "CommonTime" << " " << "ccc" << endl;
    myFile.close();
    myFile.open ("MuMinusMean.txt",ios::trunc);
    myFile<< "PRN" << " " << "Time" << ' ' << "Elevation" <<  ' ' << "Mu" << endl;
    myFile.close();
    myFile.open ("standardDeviationLeveledMu.txt",ios::trunc);
    myFile<< "PRN" << ' ' << "Elevation" << ' ' << "StDev" << endl;
    myFile.close();


    ////////////////////// INPUTS ////////////////////////////////
    // Number of files for computing the antenna calibrations, do not change unless you have
    // already compute the antenna calibration with a different number of files.
    // iValue and jValue are the bands used to compute multipath in the antenna calibration, use the
    // same values in the test. iValue means that you are computing the E iValue=1,5a,5b,5a+b,6  band multipath using the
    // phase of the E jValue band.

    int const iValue = 1, jValue = 5;

    // The file array is the set of the antenna calibration files already computed. For default they are 5.

    array<string, numberObsFiles> AntCalFile = {"AntennaCalibrationISLK.txt", "AntennaCalibrationOZST.txt",
                                        "AntennaCalibrationP558.txt", "AntennaCalibrationP588.txt",
                                        "AntennaCalibrationCAT3.txt"};


//    array<string, numberObsFiles> stDevFile = {"StandardDeviationISLK.txt", "StandardDeviationOZST.txt",
//                                               "StandardDeviationP558.txt", "StandardDeviationP588.txt",
//                                               "StandardDeviationCAT3.txt"};

//    array<string, numberObsFiles> dAntdElFile = {"dAntdElISLK.txt", "dAntdElOZST.txt",
//                                              "dAntdElP558.txt", "dAntdElP588.txt",
//                                              "dAntdElCAT3.txt"};

    // This is what you really can modify easily. Here you can set up the day when you want to test the tool.
    // Change the values for 92,93,94,etc. Make sure that you have the files in the current folder before.

    array<string, numberObsFiles> obsFile = {"ISLK092A00.17O", "OZST092A00.17O", "P558092A00.17O", "P588092A00.17O",
                                        "CAT3092A00.17O"};
//    array<string, numberObsFiles> obsFile = {"ISLK106A00.17O", "OZST106A00.17O", "P558106A00.17O", "P588106A00.17O",
//                                          "CAT3106A00.17O"};
//    array<string, numberObsFiles> obsFile = {"ISLK151A00.17O", "OZST151A00.17O", "P558151A00.17O", "P588151A00.17O",
//                                          "CAT3151A00.17O"};

    // This array is linked with the array above. You must select the same day for the SP3 file and the observable Rinex file.
    // Also, you must add the following day of the SP3 file because it is required by the GPSTK functions. At the end of each day,
    // they need the information of the satellites of the following day.

    array<string, 2> SP3File = {"grm19430.sp3", "grm19431.sp3"};// for day 92 of 2017
//    array<string, 2> SP3File = {"grm19450.sp3", "grm19451.sp3"};//for day 106 of 2017
//    array<string, 2> SP3File = {"grm19513.sp3", "grm19514.sp3"};//for day 151 of 2017


    ////////////////////////////////////////////////////////////////

    SP3EphemerisStore EphList;
```

```
    // Set flags to reject satellites with bad or absent positional
    // values or clocks
    EphList.rejectBadPositions(true);
    EphList.rejectBadClocks(true);

    EphList.loadFile(SP3File[0]);
    EphList.loadFile(SP3File[1]);

    vector<SatID> satList;
    satList = EphList.getSatList(); //List with the available satellites of every contellation


    TimeSystem SP3System = EphList.getTimeSystem();
    cout << "SP3File has as time System: " << SP3System.asString() << endl;

    vector<int> PRNList; //List with the IDs of the available satellites in Galileo
    for(int i=0;i<satList.size();i++){
//          cout << satList[i] << endl;
        if (satList[i].system == SatID::systemGalileo){
            PRNList.push_back(satList[i].id);
        }
    }

    //For getting the standard deviation from files and save them in global variables
    for (int nObsFile = 0; nObsFile < numberObsFiles; nObsFile++) {
        AntennaCalibration = getFromFile(AntCalFile[nObsFile]);
        switch (nObsFile) {
            case 0 : {
                for (int i = 0; i < AntennaCalibration.size(); i++) {
                    AntennaCalibration0.push_back(AntennaCalibration[i]);
                }
                break;
            }
            case 1 : {
                for (int i = 0; i < AntennaCalibration.size(); i++) {
                    AntennaCalibration1.push_back(AntennaCalibration[i]);
                }
                break;
            }
            case 2 : {
                for (int i = 0; i < AntennaCalibration.size(); i++) {
                    AntennaCalibration2.push_back(AntennaCalibration[i]);
                }
                break;
            }
            case 3 : {
                for (int i = 0; i < AntennaCalibration.size(); i++) {
                    AntennaCalibration3.push_back(AntennaCalibration[i]);
                }
                break;
            }
            case 4 : {
                for (int i = 0; i < AntennaCalibration.size(); i++) {
                    AntennaCalibration4.push_back(AntennaCalibration[i]);
                }
                break;
            }
            default:
                cout << "Error because the number of observable files is inappropiate" << endl;
                break;
        }
    }

    cout << "Number of satellites: " << PRNList.size() << endl; //It should be 14
    int m = 0; // Start from 0
    while (m< PRNList.size()) {


        int PRN = PRNList[m];
        m++;
        Multipath0.clear();
        Multipath1.clear();
        Multipath2.clear();
        Multipath3.clear();
        Multipath4.clear();


        for (int nObsFile = 0; nObsFile < numberObsFiles; nObsFile++) {
            computeMultipath(obsFile[nObsFile], SP3File[0], SP3File[1], PRN, iValue, jValue, nObsFile);


            cout << "IMPORTANT TEST" << endl;
            cout << Multipath0.size() << " " << Multipath1.size() << " " << Multipath2.size() << " "
                << Multipath3.size() << " " << Multipath4.size() << endl;



        } // end of for (int j = 0; j < numberObsFiles; j++)
```

```
        if ((Multipath0.size() == 0) || (Multipath1.size() == 0) || (Multipath2.size() == 0)||
            (Multipath3.size() == 0) || (Multipath4.size() == 0)){
            cout << "There is no available information for the PRN: " << PRN << endl;
//                  myFile.open ("ccc.txt",ios::app);
//                  myFile << "Satellite: " << PRN << ' ' << "has no available data" <<endl;
//                  myFile.close();

            continue;
        }
        saveStandardDeviation(PRN);

        cout << "\n\n\nCHECKPOINT\n\n\n\n" << endl;
        vector < tuple<CommonTime, double, double>> CCC = computeccc();

        myFile.open ("ccc.txt",ios::app);
        for (int i = 0; i < CCC.size(); i++) {
            myFile << "Satellite: " << PRN << ' ' << get<0>(CCC[i]) << ' ' << get<1>(CCC[i]) << ' ' << get<2>(CCC[i]) << endl;
        }
        myFile.close();

    } //end of while (m< PRNlist.size())
    return 0;
} //End of main

vector <pair<int, double>> getFromFile(string file){
    char cFirst[100];
    char cSecond[100];
    int first;
    double second;
    vector < pair <int,double>> vectorOfPairs;
    ifstream myReadFile;
    myReadFile.open(file);
    if (myReadFile.is_open()) {
        while (!myReadFile.eof()) {
            myReadFile >> cFirst >> cSecond;
            first = stoi(cFirst);
            second = stod(cSecond);
            vectorOfPairs.push_back(make_pair(first,second));
        }
    }
    myReadFile.close();
    return vectorOfPairs;
}; //End of getFromFile

void computeMultipath(string obsFile, string SP3File1, string SP3File2, int SV, int iValue , int jValue, int loopCounter){

    SP3EphemerisStore EphList;


    // Set flags to reject satellites with bad or absent positional
    // values or clocks
    EphList.rejectBadPositions(true);
    EphList.rejectBadClocks(true);

    EphList.loadFile(SP3File1);
    EphList.loadFile(SP3File2);

    vector<SatID> satList;
    satList = EphList.getSatList(); //List with the available satellites of every contellation


    TimeSystem SP3System = EphList.getTimeSystem();
//    cout << "SP3File has as time System: " << SP3System.asString() << endl;

    vector<int> PRNList; //List with the IDs of the available satellites in Galileo
    for(int i=0;i<satList.size();i++){
//        cout << satList[i] << endl;
        if (satList[i].system == SatID::systemGalileo){
            if (satList[i].id == SV) {
                PRNList.push_back(satList[i].id);
            }
        }
    }


    double Lambdai;
    TypeID psei;
    TypeID phasei;
    TypeID LLIi;
    TypeID CSLi;
    double Lambdaj;
    TypeID psej;
    TypeID phasej;
    TypeID LLIj;
    TypeID CSLj;
    switch (iValue) {
        case 1: Lambdai = C_MPS / L1_FREQ_GPS;
            psei = TypeID::C1;
            phasei = TypeID::L1;
            LLIi = TypeID::LLI1;
```

```
            CSLi = TypeID::CSL1;
            break;
        case 2: Lambdai = C_MPS / L2_FREQ_GPS;
            psei = TypeID::C2;
            phasei = TypeID::L2;
            LLIi = TypeID::LLI2;
            CSLi = TypeID::CSL2;
            break;
        case 5: Lambdai = C_MPS / L5_FREQ_GAL;
            psei = TypeID::C5;
            phasei = TypeID::L5;
            LLIi = TypeID::LLI5;
            CSLi = TypeID::CSL5;
            break;
        case 6: Lambdai = C_MPS / L6_FREQ_GAL;
            psei = TypeID::C6;
            phasei = TypeID::L6;
            LLIi = TypeID::LLI6;
            CSLi = TypeID::CSL6;
            break;
        case 7: Lambdai = C_MPS / L7_FREQ_GAL;
            psei = TypeID::C7; //pseudorange
            phasei = TypeID::L7; //carrier phase
            LLIi = TypeID::LLI7;//Loss of Lock Indicator
            CSLi = TypeID::CSL7;//Cycle Slip Indicator
            break;
        case 8: Lambdai = C_MPS / L8_FREQ_GAL;
            psei = TypeID::C8;
            phasei = TypeID::L8;
            LLIi = TypeID::LLI8;
            CSLi = TypeID::CSL8;
            break;
        default : cout << "Choose a correct frequency band" << endl;
            break;
    }
    switch (jValue) {
        case 1: Lambdaj = C_MPS / L1_FREQ_GPS;
            psej = TypeID::C1;
            phasej = TypeID::L1;
            LLIj = TypeID::LLI1;
            CSLj = TypeID::CSL1;
            break;
        case 2: Lambdaj = C_MPS / L2_FREQ_GPS;
            psej = TypeID::C2;
            phasej = TypeID::L2;
            LLIj = TypeID::LLI2;
            CSLj = TypeID::CSL2;
            break;
        case 5: Lambdaj = C_MPS / L5_FREQ_GAL;
            psej = TypeID::C5;
            phasej = TypeID::L5;
            LLIj = TypeID::LLI5;
            CSLj = TypeID::CSL5;
            break;
        case 6: Lambdaj = C_MPS / L6_FREQ_GAL;
            psej = TypeID::C6;
            phasej = TypeID::L6;
            LLIj = TypeID::LLI6;
            CSLj = TypeID::CSL6;
            break;
        case 7: Lambdaj = C_MPS / L7_FREQ_GAL;
            psej = TypeID::C7;
            phasej = TypeID::L7;
            LLIj = TypeID::LLI7;
            CSLj = TypeID::CSL7;
            break;
        case 8: Lambdaj = C_MPS / L8_FREQ_GAL;
            psej = TypeID::C8;
            phasej = TypeID::L8;
            LLIj = TypeID::LLI8;
            CSLj = TypeID::CSL8;
            break;
        default : cout << "Choose a correct frequency band" << endl;
            break;
    }

    //This is the GNSS data structure that will hold all the
    // GNSS-related information
    gnssRinex gRin;
    // Declare an object to check that all required observables are present
    RequireObservables requireObs(psei);
    requireObs.addRequiredType(phasei);
    requireObs.addRequiredType(phasej);

//    Objects to mark cycle slips
    OneFreqCSDetector markCSi;
    markCSi.setCodeType(psei);
    markCSi.setPhaseType(phasei);
    markCSi.setLLIType(LLIi);//Loss of Lock Indicator
    markCSi.setResultType(CSLi);
```

```
    OneFreqCSDetector markCSj;
    markCSj.setCodeType(psej);
    markCSj.setPhaseType(phasej);
    markCSj.setLLIType(LLIj);//Loss of Lock Indicator
    markCSj.setResultType(CSLj);

    // This object defines several handy linear combinations
    LinearCombinations comb;

    // Object to compute linear combinations for cycle slip detection
    ComputeLinear linear1(comb.pdeltaCombination);
    linear1.addLinear(comb.ldeltaCombination);
    linear1.addLinear(comb.mwubbenaCombination);
    linear1.addLinear(comb.liCombination);
    vectorOfTuple3 vecMu;
    vectorOfTuple3 MuMinusMeanMu;


    try
    {
        int usefulDataCounter = 0;
        int m = 0;
        while (m < PRNList.size()) {
            int mypprn = PRNList[m];

            Rinex3ObsStream roffs(obsFile);
            // Loop through epochs and process data for each.
            // ----------------------------------------
            Dumper dumpObj;

            //Satellite identification: PRN and satellite system
            SatID prn( mypprn, SatID::systemGalileo );

            double ele0 = 0;
            bool flag = false;
            bool flag2 = false;
            int Dele;
            int Dele0;
            double mu0;

            while (roffs >> gRin) {


                Position recxyz = gRin.header.antennaPosition;

                TimeSystem timeSys = TimeSystem::GAL;

                CommonTime time(gRin.header.epoch);

                time.setTimeSystem(SP3System);
                // Let's use the CivilTime class to print time
//                cout << SP3System.asString() << "  " << time << " PRN: " << mypprn << endl;

                CivilTime civtime(time);


                try{
                    // The following lines are indeed just one line
//                      // Satellites without the specific required observations will be deleted
                    gRin >> requireObs      // Check if required observations are present
                        >> linear1          // Compute linear combinations to detect CSL
                        >> markCSi          // Mark cycle slips: LI algorithm
                        >> markCSj;         // Mark cycle slips: Melbourne-Wubbena


                }
                catch(Exception& e)
                {
                    cerr << "Exception at epoch: " << time << "; " << e << endl;
                    continue;
                }
                catch(...)
                {
                    cerr << "Unknown exception at epoch: " << time << endl;
                    continue;
                }
                // Get P1, P2 and L1 observations
                // Let's use the CivilTime class to print time

                try
                {

                    double Li = gRin.body.getValue(prn, phasei);
                    double Lj = gRin.body.getValue(prn, phasej);
                    double Pi = gRin.body.getValue(prn, psei);
                    if ((Li==0) || (Lj==0) || (Pi==0)){
                        continue;
                    }

                    // Compute multipath
                    // ----------------
```

```
                double mu = Pi - Li - 2 * pow(Lambdai, 2) * (Li  - Lj ) /
                                (pow(Lambdaj, 2) - pow(Lambdai, 2));

                usefulDataCounter++; //Adding 1 to the useful data counter

                Position satPos = EphList.getPosition(prn, civtime); //If timeSys is Gal, one error, if it is GPS segmentation fault
                double ele = recxyz.elevation(satPos);

                /////////////// OUTPUT TO TERMINAL ////////////////////
//                cout << civtime << " "
//                     << fixed << setw(7) << setprecision(3)
//                     << " PRN " << myprn << " biased multipath " << mu
//                     << " Pi: " << Pi << " Li: " << Li << " Lj: " << Lj
//                     << "  elevation: " << ele << endl;
                ////////////////////////////////////////////////////

                if (ele <= 5) continue;

                vecMu.push_back(boost::make_tuple(civtime, ele, mu));



                if ((ele - ele0) > 0){
                    Dele = 1;
                } else {
                    Dele = 0;
                }
                if (!flag) Dele0=Dele;

                if (vecMu.size()>2){
                    if (abs(mu- mu0)>10) {
//                      cout<< "\n\n\n\n\n\n Flag2\n\n\n\n\n";
                        flag2 = true;
                    }
                }

                if (((abs(ele-ele0) > 0.2) && flag) || (Dele != Dele0) || (flag2==true)){
                    vecMu.pop_back();
                    MuMinusMeanMu = computeMuMinusMeanMu(vecMu);

                    ofstream ff;
                    ff.open ("MuMinusMean.txt",ios::app);
                    for(int i=0;i<MuMinusMeanMu.size();i++){
                        ff<< myprn << " " << get<0>(MuMinusMeanMu[i]).convertToCommonTime() << ' ' << get<1>(MuMinusMeanMu[i]) << ' ' << get<2>(MuMinusMeanMu[i]) << endl;
//                      cout << myprn << " " << get<2>(vecMu[i])<< ' ' << get<1>(MuMinusMeanMu[i]) << ' ' << get<2>(MuMinusMeanMu[i]) << endl;

                    }
                    ff.close();
                    pushBackMultipathAsTupleOf4(MuMinusMeanMu, loopCounter);

                    vecMu.clear();
                    vecMu.push_back(boost::make_tuple(civtime, ele, mu));
                    flag= false;
                    flag2=false;
                    mu0=mu;
                    continue;
                } //end of if Dele...

                ele0 = ele;
                Dele0 = Dele;
                flag = true;
                mu0=mu;

            }
            catch (SatIDNotFound)
            {
//                  cout << civtime << " ";
//                  cout << "PRN " << myprn << " not in view " << endl;
                continue;
            }
            catch(Exception& e)
            {
//                cerr << "Exception at epoch: " << time << ": " << e << endl;
                continue;
            }

        }//while (roofs >> gRin)

        MuMinusMeanMu = computeMuMinusMeanMu(vecMu);
        ofstream ff;
        ff.open ("MuMinusMean.txt",ios::app);
        for(int i=0;i<MuMinusMeanMu.size();i++){
            ff<< myprn << " " << get<0>(MuMinusMeanMu[i]).convertToCommonTime() << ' ' << get<1>(MuMinusMeanMu[i]) << ' ' << get<2>(MuMinusMeanMu[i]) << endl;
//              cout << myprn << " " << get<2>(vecMu[i])<< ' ' << get<1>(MuMinusMeanMu[i]) << ' ' << get<2>(MuMinusMeanMu[i]) << endl;
        }
        ff.close();
        pushBackMultipathAsTupleOf4(MuMinusMeanMu, loopCounter);


        vecMu.clear();
```

```
            vecMu.clear();


            cout << "Read " << roffs.recordNumber << " epochs of PRN: " << myprn<< " with " << usefulDataCounter
                << " epochs with useful data in the observable file: " << obsFile << endl;

            m++; //counter along prnList which exist in the SP3File
        } // End of 'while'

        return;
    }
    catch (Exception& e)
    {
        cout << e;
        exit(-1);
    }
    catch (...)
    {
        cout << "Caught an unknown exception" << endl;
        exit(-1);
    }

} //End of computeMultipath

//Function that computes the Mean of the Mu and the compute the Mu minus the mean for each elevation angle
vectorOfTuple3 computeMuMinusMeanMu(vectorOfTuple3 MuEl){
    double meanMu=0;
    vectorOfTuple3 MuMinusMean;
    for(int i=0;i<MuEl.size();i++){
        meanMu += get<2>(MuEl[i]);
    }
    meanMu /= MuEl.size();
//    cout << "Mean Multipath\n" << meanMu << "\nMu minus meanMu: \n";

    for(int i=0;i<MuEl.size();i++){
        MuMinusMean.push_back(boost::make_tuple(get<0>(MuEl[i]),get<1>(MuEl[i]),get<2>(MuEl[i])-meanMu));
//        cout << get<0>(MuMinusMean[i]) << " " << get<1>(MuMinusMean[i]) << ' ' << get<2>(MuMinusMean[i]) << endl;
    }
    return MuMinusMean;
}

void pushBackMultipathAsTupleOf4(vectorOfTuple3 MuMinusMeanMu, int loopCounter){
    switch (loopCounter) {
        case 0 : {
//            cout << "\nMultipath 0:\n" << endl;
            for(int i=0;i<MuMinusMeanMu.size();i++) {
                for(int j=0;j<AntennaCalibration0.size();j++) {
                    if (AntennaCalibration0[j].first==get<1>(MuMinusMeanMu[i])) {
                        Multipath0.push_back(boost::make_tuple(get<0>(MuMinusMeanMu[i]),
                                                               get<1>(MuMinusMeanMu[i]),
                                                               get<2>(MuMinusMeanMu[i]),
                                                               abs(AntennaCalibration0[j].second - get<1>(MuMinusMeanMu[i]))));
//                      cout << get<0>(MuMinusMeanMu[i]) << ' ' << get<1>(MuMinusMeanMu[i]) << ' '
//                           << get<2>(MuMinusMeanMu[i]) << ' ' << AntennaCalibration0[j].second << endl;
                        break;
                    }
                }
            }
            break;
        }
        case 1 : {
//            cout << "\nMultipath 1:\n" << endl;
            for(int i=0;i<MuMinusMeanMu.size();i++) {
                for(int j=0;j<AntennaCalibration1.size();j++) {
                    if (AntennaCalibration1[j].first==get<1>(MuMinusMeanMu[i])) {
                        Multipath1.push_back(boost::make_tuple(get<0>(MuMinusMeanMu[i]),
                                                               get<1>(MuMinusMeanMu[i]),
                                                               get<2>(MuMinusMeanMu[i]),
                                                               abs(AntennaCalibration1[j].second - get<2>(MuMinusMeanMu[i]))));
//                      cout << get<0>(MuMinusMeanMu[i]) << ' ' << get<1>(MuMinusMeanMu[i]) << ' '
//                           << get<2>(MuMinusMeanMu[i]) << ' ' << AntennaCalibration1[j].second << endl;
                        break;
                    }
                }
            }
            break;
        }
        case 2 : {
//            cout << "\nMultipath 2:\n" << endl;
            for(int i=0;i<MuMinusMeanMu.size();i++) {
                for(int j=0;j<AntennaCalibration2.size();j++) {
                    if (AntennaCalibration2[j].first==get<1>(MuMinusMeanMu[i])) {
                        Multipath2.push_back(boost::make_tuple(get<0>(MuMinusMeanMu[i]),
                                                               get<1>(MuMinusMeanMu[i]),
                                                               get<2>(MuMinusMeanMu[i]),
                                                               abs(AntennaCalibration2[j].second - get<2>(MuMinusMeanMu[i]))));
//                      cout << get<0>(MuMinusMeanMu[i]) << ' ' << get<1>(MuMinusMeanMu[i]) << ' '
//                           << get<2>(MuMinusMeanMu[i]) << ' ' << AntennaCalibration2[j].second << endl;
                        break;
                    }
```

```
                }
            }
            break;
        }
        case 3 : {
//              cout << "\nMultipath 3:\n" << endl;
            for(int i=0;i<MuMinusMeanMu.size();i++) {
                for(int j=0;j<AntennaCalibration3.size();j++) {
                    if (AntennaCalibration3[j].first==get<1>(MuMinusMeanMu[i])) {
                        Multipath3.push_back(boost::make_tuple(get<0>(MuMinusMeanMu[i]),
                                                               get<1>(MuMinusMeanMu[i]),
                                                               get<2>(MuMinusMeanMu[i]),
                                                               abs(AntennaCalibration3[j].second - get<2>(MuMinusMeanMu[i]))));
//                        cout << get<0>(MuMinusMeanMu[i]) << ' ' << get<1>(MuMinusMeanMu[i]) << ' '
//                             << get<2>(MuMinusMeanMu[i]) << ' ' << AntennaCalibration3[j].second << endl;
                        break;
                    }
                }
            }
            break;
        }
        case 4 : {
//              cout << "\nMultipath 4:\n" << endl;
            for(int i=0;i<MuMinusMeanMu.size();i++) {
                for(int j=0;j<AntennaCalibration4.size();j++) {
                    if (AntennaCalibration4[j].first==get<1>(MuMinusMeanMu[i])) {
                        Multipath4.push_back(boost::make_tuple(get<0>(MuMinusMeanMu[i]),
                                                               get<1>(MuMinusMeanMu[i]),
                                                               get<2>(MuMinusMeanMu[i]),
                                                               abs(AntennaCalibration4[j].second - get<2>(MuMinusMeanMu[i]))));
//                        cout << get<0>(MuMinusMeanMu[i]) << ' ' << get<1>(MuMinusMeanMu[i]) << ' '
//                             << get<2>(MuMinusMeanMu[i]) << ' ' << AntennaCalibration4[j].second << endl;
                        break;
                    }
                }
            }
            break;
        }
        default: {
            cout << "Error in creating the multipath tuples" << endl;
            break;
        }
    }

}

vector < tuple<CommonTime, double, double> > computeccc(){
    int c0=0,c1=0,c2=0,c3=0,c4=0; //counters
    long double num;
    double ccc;
    double stDevCCC;
    double sumStDev;
    vector <long double> invStdDev2; //    [1 / standard deviation^2]
    vector < tuple<CommonTime, double, double> > CCC;


    cout << "Multipath sizes:" << endl;
    cout << Multipath0.size() << " " << Multipath1.size() << " " << Multipath2.size() << " " << Multipath3.size()
        << " " <<Multipath4.size() << endl;
    cout << "Standard Deviation sizes:" << endl;
    cout << AntennaCalibration0.size() << " " << AntennaCalibration1.size() << " " << AntennaCalibration2.size() << " "
        << AntennaCalibration3.size() << " " <<AntennaCalibration4.size() << endl;

    if ( (AntennaCalibration0.size() == 0) || (AntennaCalibration1.size() == 0) || (AntennaCalibration2.size() == 0) ||
            (AntennaCalibration3.size() == 0) || (AntennaCalibration4.size() == 0)){
        return CCC;
    }

    //firstly, choose the first and the last time of the computation
    CivilTime runningTime = max(max(max(max(get<0>(Multipath0[c0]), get<0>(Multipath1[c1])),
                                        get<0>(Multipath2[c2])), get<0>(Multipath3[c3])),get<0>(Multipath4[c4]));
    CivilTime lastTime = min(min(min(min(get<0>(*(Multipath0.end()-1)), get<0>(*(Multipath1.end()-1))),
                                     get<0>(*(Multipath2.end()-1))),get<0>(*(Multipath3.end()-1))),get<0>(*(Multipath4.end()-1)));

    while (runningTime<lastTime) {

        //This moves all the collected multipath data from every receiver to the same time
        while(get<0>(Multipath0[c0]) < runningTime) c0++;
        while(get<0>(Multipath1[c1]) < runningTime) c1++;
        while(get<0>(Multipath2[c2]) < runningTime) c2++;
        while(get<0>(Multipath3[c3]) < runningTime) c3++;
        while(get<0>(Multipath4[c4]) < runningTime) c4++;

//        cout << get<0>(Multipath0[c0]) << get<0>(Multipath1[c1]) << get<0>(Multipath2[c2]) << get<0>(Multipath3[c3]) << get<0>(Multipath4[c4]) << endl;
        if ((get<0>(Multipath0[c0]) == runningTime) & (get<0>(Multipath1[c1]) == runningTime) & (get<0>(Multipath2[c2]) == runningTime)
                                    & (get<0>(Multipath3[c3]) == runningTime) & (get<0>(Multipath4[c4]) == runningTime)){
            array <double,numberObsFiles> Multipaths = {get<2>(Multipath0[c0]),get<2>(Multipath1[c1]),get<2>(Multipath2[c2]),get<2>(Multipath3[c3]),get<2>(Multipath4[c4])};
            array <double,numberObsFiles> stdDev = {get<3>(Multipath0[c0]),get<3>(Multipath1[c1]),get<3>(Multipath2[c2]),get<3>(Multipath3[c3]),get<3>(Multipath4[c4])};
//            cout << Multipaths[0] << Multipaths[1] << Multipaths[2] << Multipaths[3] << Multipaths[4] << endl;
//            cout << "invSqStdDev\n" << c0 << ' ' << c1 << ' ' << c2 << ' ' << c3 << ' ' << endl;
            num = 0;
```

```
            invStdDev2.clear();
            sumStDev=0;

            for (int i=0; i< numberObsFiles;i++){

                invStdDev2.push_back(1./pow(stdDev[i],2));
                num += Multipaths[i]*invStdDev2[i];
                sumStDev += invStdDev2[i];
            }
            stDevCCC = 1./(stdDev.size()*sqrt(sumStDev));
//            stDevCCC = sqrt(pow(stdDev[1],2)/stdDev.size());

            ccc = num/(accumulate(invStdDev2.begin(),invStdDev2.end(),0.0));
            CCC.push_back(boost::make_tuple(runningTime.convertToCommonTime(),ccc, stDevCCC));
            c0++;c1++;c2++;c3++;c4++;
        }
        else {
//            cout << get<0>(Multipath0[c0]) << " " << get<0>(Multipath1[c1]) << " " << get<0>(Multipath2[c2]) << " " << get<0>(Multipath3[c3]) << " " << get<0>(Multipath4[c4]) <<

        }

        CommonTime runningCommonTime = runningTime.convertToCommonTime();
        runningCommonTime +=15;
        runningTime.convertFromCommonTime(runningCommonTime);
//        cout << runningTime << endl;
    }


    return CCC;
}

void saveStandardDeviation(int PRN) {
    ofstream myFile;
    myFile.open ("standardDeviationLeveledMu0.txt",ios::app);
    for(int i=0;i<(Multipath0.size());i++){
        myFile<< PRN << ' ' << get<1>(Multipath0[i]) << ' ' << get<3>(Multipath0[i]) << endl;
    }
    myFile.close();
    myFile.open ("standardDeviationLeveledMu1.txt",ios::app);
    for(int i=0;i<(Multipath1.size());i++){
        myFile<< PRN << ' ' << get<1>(Multipath1[i]) << ' ' << get<3>(Multipath1[i]) << endl;
    }
    myFile.close();
    myFile.open ("standardDeviationLeveledMu2.txt",ios::app);
    for(int i=0;i<(Multipath2.size());i++){
        myFile<< PRN << ' ' << get<1>(Multipath2[i]) << ' ' << get<3>(Multipath2[i]) << endl;
    }
    myFile.close();
    myFile.open ("standardDeviationLeveledMu3.txt",ios::app);
    for(int i=0;i<(Multipath3.size());i++){
        myFile<< PRN << ' ' << get<1>(Multipath3[i]) << ' ' << get<3>(Multipath3[i]) << endl;
    }
    myFile.close();
    myFile.open ("standardDeviationLeveledMu4.txt",ios::app);
    for(int i=0;i<(Multipath4.size());i++){
        myFile<< PRN << ' ' << get<1>(Multipath4[i]) << ' ' << get<3>(Multipath4[i]) << endl;
    }
    myFile.close();
};
```