

Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN  
IMPIANTI E SISTEMI AEROSPAZIALI

CICLO XXIV

Settore Concorsuale : 09/A1 INGEGNERIA AERONAUTICA, AEROSPAZIALE E NAVALE

Settore Scientifico Disciplinare: ING-IND/05 IMPIANTI E SISTEMI AEROSPAZIALI

**Design and implementation of a novel multi-  
constellation FPGA-based dual frequency GNSS  
receiver for space applications**

Presentata da: ALESSANDRO AVANZI

Relatore: PAOLO TORTORA

Coordinatore Dottorato: VINCENZO PARENTI CASTELLI

Esame finale anno 2012



---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Research objectives.....	5
1.2	Thesis breakdown .....	7
<b>2</b>	<b>GNSS Receivers for Space Applications</b>	<b>8</b>
2.1	Space mission simulation and receiver requirements .....	8
2.1.1	GNSS Signals .....	10
2.1.2	GNSS constellation assumptions.....	14
2.1.3	Reference application .....	15
2.1.4	Link budget.....	15
2.1.5	Expected Doppler and Doppler Rate .....	30
2.1.6	Visibility and DOP .....	33
2.1.7	Receiver requirements summary .....	40
2.2	GNSS receiver architecture.....	42
2.2.1	IC front-end .....	45
2.2.2	Digital Signal Processing .....	51
2.2.3	Software Architecture.....	56
<b>3</b>	<b>Signal Acquisition and Tracking</b>	<b>59</b>
3.1	Introduction.....	59
3.2	Acquisition and MofN algorithm description .....	61
3.3	Tracking algorithm.....	69
3.3.1	Principles of phase tracking.....	69
3.3.2	PLL and DLL implementation aspects.....	81
3.4	Expected tracking performances.....	93
3.4.1	Simulated scenario for tracking algorithm testing.....	96
3.4.2	Real signal results.....	100
<b>4</b>	<b>Navigation Algorithm</b>	<b>102</b>

---

4.1	Introduction .....	102
4.2	Navigation Algorithm description.....	107
4.2.1	Time .....	107
4.2.2	Reference frame .....	108
4.2.3	Dynamic model.....	110
4.2.4	Numerical Integration .....	121
4.2.5	Measurement processing.....	122
4.3	GNSS broadcasted orbit and clock quality.....	128
4.3.1	Signal-in-Space range error .....	128
4.3.2	The IGS products .....	129
4.4	GNSS observables error modeling.....	135
4.4.1	Code tracking errors.....	136
4.4.2	Ionosphere error .....	136
4.4.3	Multipath errors .....	139
4.4.4	Orbit and clock errors .....	141
4.5	Expected navigation performances .....	142
<b>5</b>	<b>Conclusions</b> .....	<b>146</b>
5.1	Receiver electronic and software architecture.....	146
5.2	Acquisition and tracking algorithm.....	147
5.3	Navigation algorithm.....	147
5.4	Final considerations.....	148
	<b>Bibliography</b> .....	<b>150</b>





# 1 Introduction

The *Global Positioning System*, with acronym *GPS*, is a satellite constellation that provides position and time services to the user through the broadcasting of a *Code Division Multiple Access (CDMA)* signal. The system is operated by the USA Department of Defense (DoD) and is fully operational since 1995. The open service was originally designed to provide position accuracy of about 150 m, while after the deactivation of *Selective Availability (S/A)* (an intentional degradation of the provided service) in May 2000, the achievable position accuracy increased up to the present 10 m. The GPS system is presently composed of 31 Medium Earth Orbit (MEO) spacecrafts at an altitude of approximately 20200 km above the Earth surface, equally distributed on 6 orbit planes with 55 degrees of inclination. The spacecrafts population is constantly updated, and in 2010 the first GPS satellite of generation IIF, with new signal and improved services, has been launched from the Cape Canaveral Air Force Station.

Since the deployment of the GPS infrastructure, the system has become more and more important in many fields, not only related to the navigation, but also for example to the time synchronization of ground infrastructures, such as some segments and components of the UMTS network. This growing importance has motivated other countries and institutions to develop their own infrastructure and services, in order to improve the technology and to gradually obtain some level of independence from the original GPS system. The European Galileo is one of the new systems, which is expected to be operational in 2014. The Chinese COMPASS, composed of 24 MEO satellites and 5 geostationary satellites over China is also expected to be operational before 2020, while the Russian GLONASS is now being updated with new satellites a new signals. In addition to the constellations, services provided by geostationary satellites have been created to augment the navigation an time accuracy provided by the GPS, in order to

satisfy the requirements of specific applications. This group of systems is generally referenced as *Satellite Based Augmentation System (SBAS)*, and is composed of the US *WAAS (Wide Area Augmentation System)*, the European *EGNOS (European Geostationary Navigation Overlay System)*, the Japanese *MSAS (Multi-functional Satellite Augmentation System)* and the *GAGAN* proposed by India. It is expected that before 2030 more than 90 satellites with 20 different open services will compose what is nowadays referred as the *GNSS (Global Navigation Satellite System)* infrastructure.

Since its origin, the navigation services provided by the GNSS system have been applied to a variety of field, some of which are beyond the original scope of the infrastructure. This is the case of the orbit determination of *Low Earth Orbit (LEO)* satellites. The LEO class of space vehicles is generally defined to be up to an altitude of 2000 km above the Earth surface, with a minimum altitude generally not lower than 200 km. They are employed in many different missions, including communication, Earth observation and remote sensing in general, gravimetric and magnetometric sounding, weather monitoring and ocean altimetry. The use of GNSS receivers for on-board satellite orbit determination has become a common solution in new satellite design, as it considerably simplifies the overall architecture if compared to traditional orbit determination systems such as ground based radiometric tracking. The GNSS sensors enable continuous tracking and provide low-noise radiometric measurements onboard the spacecraft. A summary of past and present space missions that make use of on board GNSS sensors is provided in Table 1.

Mission	Year	Receiver	Altitude	Description
Topex POSEIDON	1992	GPSDR (JPL Caltech Motorola)	1337 km	Joint satellite mission between NASA, the U.S. space agency, and CNES, the French space agency, to map ocean surface topography. First mission to demonstrate the in-orbit use of GPS receivers.

---

SRTM	2000	BlackJack (JPL)	233 km	The Shuttle Radar Topography Mission (SRTM) obtained elevation data on a near-global scale to generate the most complete high-resolution digital topographic database of Earth.
CHAMP	2000	BlackJack (JPL)	350 km	CHAMP (CHALLENGING Minisatellite Payload) is a German small satellite mission for geoscientific and atmospheric research and applications.
SAC-C	2000	BlackJack (JPL)	700 km	The SAC-C is an international Earth observing satellite <i>mission</i> developed as a partnership between CONAE and NASA.
Jason-1	2001	BlackJack (JPL)	1336 km	It is the successor to the TOPEX/Poseidon, Jason-1 is a joint project between the NASA (United States) and CNES (France) space agencies.
GRACE	2002	BlackJack (JPL)	460 km	GRACE, twin satellites launched in March 2002, are making detailed measurements of Earth's gravity field which will lead to discoveries about gravity and Earth's natural systems.
ICESat	2003	BlackJack (JPL)	595 km	ICESat (Ice, Cloud, and land Elevation Satellite) is the benchmark Earth Observing System mission for measuring ice sheet mass balance, cloud and aerosol heights, as well as land topography and vegetation characteristics, developed at NASA.
MetOP	2006	GRAS (ESA)	820 km	MetOp is the Europe's first operational polar-orbiting weather satellite. It replaces one of two satellite services operated by the United States National Oceanic and Atmospheric Administration (NOAA).
TerraSAR	2007	IGOR (BRE) Mosaic (EADS)	515 km	TerraSAR-X is the first German radar satellite to be implemented within a Public-Private Partnership (PPP) between the German Aerospace Center (DLR) and Europe's leading satellite specialist Astrium.

---

---

Jason-2	2008	BlackJack (JPL)	1336 km	It is the successor to the TOPEX/Poseidon and Jason-1, Jason-2 is a joint project between the NASA (United States) and CNES (France) space agencies.
GOCE	2009	GRAS (ESA)	295 km	GOCE is dedicated to measuring Earth's gravity field and modeling the geoid with unprecedented accuracy and spatial resolution.
PROBA-2	2009	Phoenix (DLR)	757 km	Proba stands for PProject for OnBoard Autonomy. The Proba satellites are among the smallest ever to be flown by ESA, but they are making a big impact in space technology. Proba-2 is the second of the series, building on nearly eight years of successful Proba-1 experience.

---

**Table 1: Space mission summary**

According to the mission requirements, the operational real-time position accuracy ranges from the 10-15 meters of the pure kinematic solution (following the deactivation of Selective Availability), down to 1-3 meters for the dynamical-filtered solution. Real-time sub-meter positioning has also been proved for both single and dual frequency receivers, with ground experiments by replicating the on board operations. While the 1 to 10 meters solution can be considered adequate for most of the operational needs, specific science or sensing missions require higher accuracies, which are provided by ground-based facilities through the dynamic based processing of dual frequency measurements. In this case accuracies down to the 1-5cm 3D RMS are achievable.

Several GNSS receivers have been developed for the listed space missions (a performance comparison of the devices can be found in [1]), generally showing two different approaches: the first is based on the development of a dedicated chipset, which include also specific features implemented to improve reliability in the space environment, such as the radiation hardening. This is the case of the most popular GNSS receiver for space applications, the BalckJack and its evolution IGOR, developed by the *NASA Jet Propulsion Laboratory (JPL)*. The

second strategy is focused on the adaption of ground based receivers or chipsets for the space application, which are then generally used on board low cost space technology demonstrators [2]. The *Phoenix* receiver developed by the German Aerospace Center (DLR) is part of this family and has been successfully employed for the ESA Proba-2 mission in 2009. It makes use of the Zarlink GP2040 chipset which has gone out of production in 2010. The receivers or chipsets designed for ground applications require low level modifications in order to make them compatible with the space orbit determination, primary because the received signal properties such as the Doppler differs of one order of magnitude, and secondary because the GPS US regulation imposes an *ITAR (International Traffic in Arms Regulations)* on the receiving devices so that they can provide positioning and timing outputs only below 10.000 ft. altitude (18 km) and 1.000 knots (0.5 km/s).

## 1.1 Research objectives

With the described background in mind, in the framework of the development of a small satellite for Earth Observation carried out at the Microsatellite Lab of the University of Bologna [3], named ALMASat-EO, the design, development and prototyping of an FPGA-based GNSS receiver for LEO space vehicle orbit determination has been performed as the main contribution of the PhD work described in this thesis. The target is to define a flexible architecture that strictly follows the paradigm of the *software defined radio receivers*, which is compatible not only with GPS signals, but also with the upcoming GALILEO services. In this context different aspects have been covered:

- *Upcoming system and services contribution to the LEO orbit determination.* The effect of additional signals and services on the process of orbit determination on board LEO satellites has been analyzed through realistic simulations. Not only the new GPS signals have been simulated but also the GALILEO, GLONASS and future COMPASS services, in

order to derive the performance contribution and select which of these system include in the receiver signal processing design.

- *Software defined radio receiver architecture.* The electronic and signal processing design has been based on the software defined radio paradigm; therefore most of the signal processing is performed in the form of software algorithm executed on a general purpose microprocessor, while only what is strictly necessary is executed in hardware in form of FPGA logic. This architecture proposes more challenges if compared the use of ASIC chipsets, but it guarantees higher flexibility so that the receiver can be adapted to different mission requirements and GNSS services.
- *GNSS signal synchronization and tracking algorithms.* Both the synchronization (also referred as acquisition in literature) and signal tracking algorithms have been designed and tested starting from the specific characteristics of the received signal. Third order *Delay Locked Loop (DLL)* and third order *Phase Locked Loop (PLL)* have been implemented with an improved state machine that adapts the loop bandwidth and switches to *Frequency Locked Loop (FLL)* in case of bad signal conditions.
- *Dynamic filtering algorithm for position and time solution.* The GNSS signal processing classic approach can be split in two independent sections: the first includes the signal acquisition, tracking and data demodulation, and produces the GNSS *observables*, i.e. the pseudorange and the carrier phase. The second section makes use of the observables to generate position and time fixes. The ground receivers and early space receivers generally solves for position using a *pure kinematic* solution: the observable are combined on an epoch base in a least square algorithm without any information on the receiver navigation model. When the receiver trajectory can be effectively described with a dynamic model,

such as the case of orbit determination, a dynamic based filtering is preferred since it's able to smooth the measurements noise and improve the fix accuracy. As part of the receiver design, the Kalman filter based estimation algorithm is implemented, evaluating the complexity and accuracy obtained among different algorithm solutions.

## 1.2 Thesis breakdown

This thesis consists of the following sections:

- Section 2: *GNSS receivers for space application*. This section describes the receiver hardware and software architecture starting from requirements definition. The requirements and signal properties are evaluated using a realistic mission simulator that has been developed as part of this work. The GNSS signals and transmitted power assumptions are first introduced, then power budget, signal visibility and other parameters are derived from simulation and presented in the section. This quantities are then translated into receiver hardware and signal processing requirements, and finally the realized hardware prototype is described.
- Section 3: *Signal tracking algorithms*. This section describes the synchronization and tracking algorithm that are implemented in the receiver, starting from a quick introduction on the theory behind *Phase Locked Loop (PLL)* and *Delay Locked Loop (DLL)*. Specific features introduced in the architecture are described, and the resulting tracking accuracy is determined using semi analytic loop simulations.
- Section 4: *Navigation algorithms*. This section explains the navigation algorithms design and expected performances validated with simulations. Different options and the related positioning accuracy and computational complexity are explored and assessed.

---

## 2 GNSS Receivers for Space Applications

### 2.1 Space mission simulation and receiver requirements

ALMASat-EO is the first Earth Observation microsatellite entirely designed, manufactured and assembled by the Microsatellites and Space Microsystems Lab of the University of Bologna [3]. The project has the main goal of designing a space platform for Earth Observation, dedicated to several applications for the weather monitoring and land surveillance, and the spacecraft is the successor of the ALMASat-1 microsatellite, successfully launched onboard the VEGA maiden flight in February 2012. In order to achieve its task, ALMASat-EO will mount an on board optical payload able to take images of a portion of the observed territory of about 150 km<sup>2</sup> with a ground resolution of about 40 m at 650 km altitude.

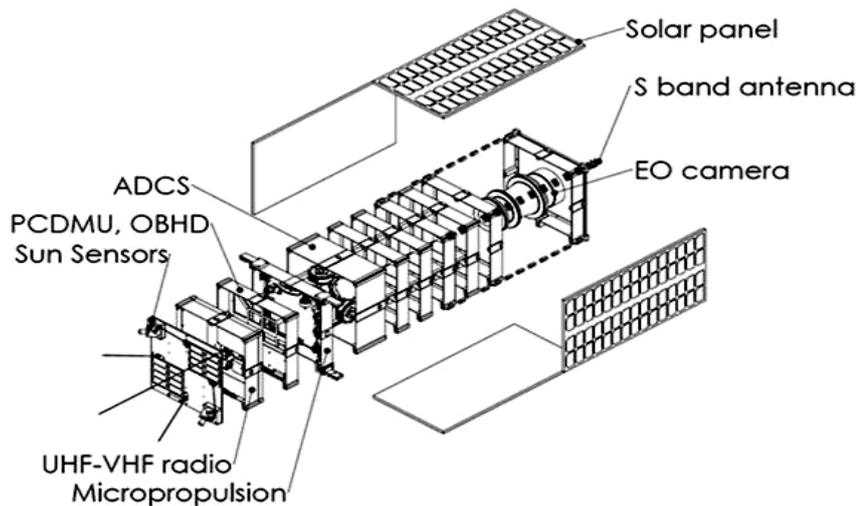


Figure 1: ALMASat-EO CAD exploded view

The need to obtain images of the Earth requires the observation under optimal lighting conditions, thus the selected orbit will belong to the family of sun-synchronous trajectories, with inclination around 98° and 600-800km altitude

above the ground. The capability of the system to correctly identify the portion of the Earth surface that is observed and give geographical reference to the collected data depends on the performances of the on-board orbit determination system. Considering the ground resolution, positioning accuracy from 1m to 10m is required to achieve the desired data geo referencing, thus motivating the selection of GNSS based technology.

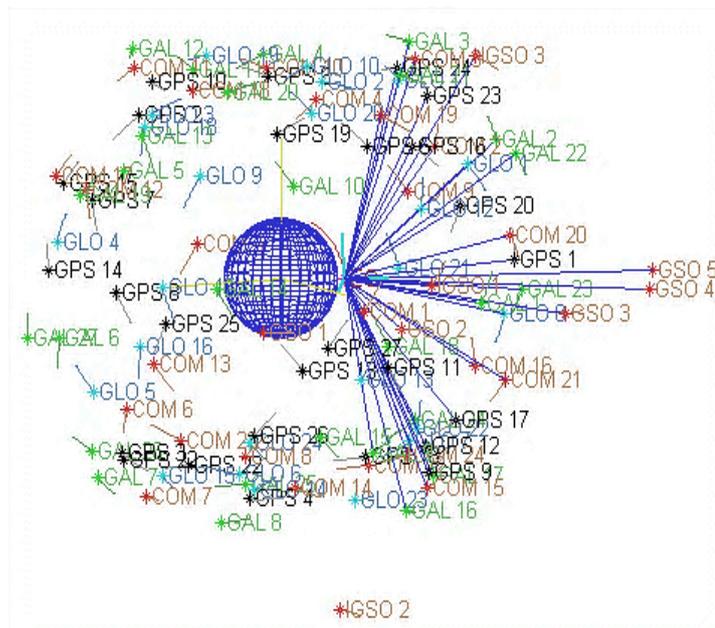


Figure 2: Simulated GNSS constellations and LEO satellite, with line-of-sight signals highlighted in blue

The first task that has been performed in order to identify the receiver requirements, consisted in the development of a full GNSS constellation simulator which generates observables as measured by the receiver that is installed on the LEO spacecraft. The purpose is to get information about relevant design parameters such as

- Signal to noise density ratio CNo, reported in section 2.1.4
- Doppler and Doppler range, reported in section 2.1.5
- Signal visibility and Dilution of Precision DOP, reported in section 2.1.7

and also to design the positioning algorithms having a clear insight on error sources and their effects on the estimated parameters. The navigation algorithm

design is further analyzed in section 4, while the following section presents the simulator assumptions and architecture together with the results on signal parameters, signal visibility and link budget. The simulator includes not only the current GPS system but also the future GALILEO, COMPASS and GLONASS systems, in order to evaluate their influence on the receiver performances and preliminary define the services for which the GNSS space receiver will be designed.

### 2.1.1 GNSS Signals

This section provides a quick overview on the current and future GNSS signals, with frequencies, modulation and services. The work described in the document is mainly focused on the services provided by L1/E1 band when referring to single frequency receivers and to L1/E1 L5/E5a or L1/E1 L2 when referring to dual frequency receivers. The signal description of the GPS and GALILEO Open Services is available in the official ICDs that are public documents. The ICDs of the GLONASS CDMA and COMPASS are not available, and the signal parameters for this study are based on the collected public information from scientific papers and journals.

#### 2.1.1.1 Modernized GPS Open-Service signals

Band	Carrier Freq. (MHz)	Signal	Code Rate (Mcps)	Primary code length (chips)	Secondary code length (bits)	Mod.	Symbol Data Rate (bps)
L1	1575.42 (=1540x 1.023)	L1 C/A	1.023	1,023	No	BPSK(1)	50
L2	1227.6 (=1200 x	L2C	1.023	1,534,500	<i>BPSK(1) Chip-by-chip multiplex of L2CM and L2CL (See L2CM and L2CL)</i>		

	1.023)	L2CM (Data)	0.5115	10,230	No	BPSK(0.5)	50/25
		L2CL (Pilot)	0.5115	767,250	No	BPSK(0.5)	No
L1 & L2	1575.42 & 1227.6	P	10.23	15,345,00 0	No	BPSK(10)	50
		Y	<i>Encrypted version of P-code signal which is broadcast instead of the P-code when anti-spoofing option is active. It is not an Open-Service signal but it can be tracked for iono-free processing in a semi-codeless manner</i>				
L5	1176.45 (=1165 x 1.023)	L5-I (Data)	10.23	10,230	10	BPSK(10)	100/50
		L5-Q (Pilot)	10.23	10,230	20	BPSK(10)	No

Table 2: GPS modernised signals (L1,L2C,L5)

Band	Carrier Freq. (MHz)	Signal	Code Rate (Mcps)	Primary code length (chips)	Secondary code length (bits)	Mod.	Symbol Data Rate (bps)
L1	1575.42 (=1540x 1.023)	L1C <sub>D</sub> (Data)	1.023	10230	No	BOC(1,1)	100
		L1C <sub>P</sub> (Pilot)	1.023	10230	1800	BOC(1,1) or TMBOC(6,1)	50

Table 3: GPS modernized L1C signal

### 2.1.1.2 Galileo signals

Band	Carrier Freq. (MHz)	Signal	Code Rate (Mcps)	Primary code length (chips)	Secondary code length (bits)	Mod.	Symbol Data Rate (bps)
E1	1575.42 (=1540 x 1.023)	E1-B (Data)	1.023	4,092	No	CBOC	250/125
		E1-C (Pilot)	1.023	4,092	25	CBOC	No
E5b	1207.14 (=1180 x 1.023)	E5b-I (Data)	10.23	10,230	4	BPSK(10)	250/125
		E5b-Q (Pilot)	10.23	10,230	100	BPSK(10)	No
E5	1191.79 5 (=1165 x 1.023)	AltBOC	10.23	10,230	4/100/20/ 100	AltBOC(1 5,10)	E5b-I, E5b-Q
E5a	1176.45 (=1150 x 1.023)	E5a-I (Data)	10.23	10,230	20	BPSK(10)	50/25
		E5a-Q (Pilot)	10.23	10,230	100	BPSK(10)	No
E6	1278.75 (=1250 x 1.023)	E6-B (Data)	5.115				1000
		E6-C (Pilot)	5.115				No

Table 4: Galileo signals

### 2.1.1.3 Beidou Compass signals

The COMPASS signals, included in the CP-III system version, will provide global service and will be composed of 5 GEO satellites, 3 IGSO (Inclined Geo-Stationary satellites) and 27 MEO (24 operational and 3 spares). The CP-III COMPASS system will be available before 2020.

Band	Carrier Freq. (MHz)	Signal	Code Rate (Mcps)	Primary code length (chips)	Secondary code length (bits)	Mod.	Symbol Data Rate (bps)
B1	1575.42 (=1540 x 1.023)	B1-C <sub>D</sub>	1.023	(?)	(?)	MBOC(6,1 ,1/11)	50 bps
		B1-C <sub>P</sub>					/100 sps
		B1-A <sub>D</sub>	2.046	(?)	(?)	BOC(14,2)	No data
		B1-A <sub>P</sub>					
B2a	1176.45 (=1150 x 1.023)		10.23			QPSK(10)	
B2	1191.79 5 (=1165 x 1.023)	B2a <sub>D</sub>	10.23	(?)	(?)	AltBOC(15 ,10)	25 bps
		B2a <sub>P</sub>					/50 sps
		B2b <sub>D</sub>					
		B2b <sub>P</sub>					
B2b	1207.14 (=1180 x 1.023)					QPSK(10)	
B3	1268.52 (=1240 x 1.023)	B3	10.23	(?)	(?)	QPSK(10)	500bps
		B3-A <sub>D</sub>	2.5575	(?)	(?)	BOC(15,2. 5)	50bps
		B3-A <sub>P</sub>					No data

Table 5: Compass Beidou CP-III signals

### 2.1.1.4 Glonass signals

Band	Carrier Freq. (MHz)	Signal	Code Rate (Mcps)	Primary code length (chips)	Secondary code length (bits)	Mod.	Symbol Data Rate (bps)
L1 ROC	1575.42	L1OC	(?)	(?)	(?)	BOC(2,2)	(?)
		L1SC	(?)	(?)	(?)	(?)	(?)
L2	1242	L2OC	(?)	(?)	(?)	(?)	(?)
L3	1202.125	L3OC	(?)	(?)	(?)	QPSK(10)	(?)
L5 ROC	1176.45	L5OC	(?)	(?)	(?)	BOC(4,4)	(?)

Table 6: Glonass modernized CDMA signals

## 2.1.2 GNSS constellation assumptions

The following assumptions are taken in order to model the present and future GNSS constellation:

1. GPS
  - a. 27 space vehicles, 6 planes with 4, 5, 4, 5, 4, 5, satellite on each plane. 26559 km semi-major axis, 55 deg inclination. This is the nominal GPS constellation, which is taken as reference. The real constellation presently consists of 31 space vehicles, because some of the spacecraft are continuing to operate beyond their expected lifetime, while new spacecraft have been constantly put in orbit.
2. GALILEO
  - a. FOC constellation: 27 space vehicles, equally distributed on three orbit planes. 29600 km semi-major axis, 56 deg inclination. The FOC is the final GALILEO constellation and is expected to be fully deployed in 2020.
  - b. IOC constellation: 18 space vehicles, equally distributed on three orbit planes. The IOC is the starting point constellation and is expected to be fully deployed in 2014.
3. GLONASS
  - a. 24 space vehicles equally distributed on three orbit planes, 64.8 deg inclination and 25478 km semi-major axis. The CDMA compatible constellation is expected to be operative in 2020.
4. COMPASS
  - a. 24 MEO space vehicles plus 3 spares, equally distributed on three orbit planes. 55 deg inclination and 27878 km semi-major axis. The complete global service is expected to be completed in 2020.

- b. 5 GSO space vehicles, 42146 km semi-major axis.
- c. 3 IGSO satellites, 55 deg inclination, 42146 km semi-major axis.

### **2.1.3 Reference application**

As already described in the previous sections, the reference application that is used for the simulations is the orbit determination for a LEO satellites orbiting on sun-synchronous trajectory, at an altitude between 600 and 800 km. The reference orbit parameters are:

- Altitude 650 km
- Eccentricity 0.00165
- Inclination 98.7 deg
- RAAN 30 deg
- Period 98 min

### **2.1.4 Link budget**

This section describes the link budget analysis performed with the help of the mission simulation software, with the objective of having an estimate of received signal CNo range. The ingredient that are necessary to obtain this estimate are:

- The Equivalent Isotropic Radiated Power EIPR (section 2.1.4.1)
- The transmitting antenna gain pattern (section 2.1.4.2)
- The receiver noise model (section 2.1.4.3)
- The co-channel interference, if applicable (section 2.1.4.4)

When some information is missing, like for example for the new GLONASS CDMA, simplified assumptions are taken by replicating some characteristics of the well-known GPS systems. The link budget analysis results are then presented in section 2.1.4.5.

### 2.1.4.1 Equivalent Isotropic Radiated Power

In order to compute the EIRP, the received power levels that are reported on the systems ICD, expressed for a ground user, have been reversed accounting for the assumptions reported in the interface document. The minimum power received on ground is generally expressed at an elevation above the horizon of  $5^\circ$  or  $10^\circ$  according to the system. Therefore the EIRP has been reconstructed accounting for (the listed parameters are then specified for each system):

- The receiver antenna gain for a reference application.
- The polarization mismatch loss, indicating the worst case loss between receiving an ideal RCHP signal and the real elliptical polarized signal. The value is computed for the GPS system starting from ellipticity data from the ICD, and a constant value has been considered for all the signals, while for the GALILEO is taken directly from in the ICD.
- The atmospheric loss. According to Kaplan [4] is 0.5 dB, and 0.38 dB according to Spilker [5]. 0.4 dB is taken as reference for this study.
- The free space loss

#### ➤ GPS

With reference to the future 2012-2020 constellation, 27 GPS SV are considered for the performance analysis: 10 as part of the block IIR, 7 as part of the block IIR-M and the remaining 10 as part of the IIF. The satellites are distributed on 6 planes, with 4, 5, 4, 5, 4, 5, satellites on each plane. The relevant differences for the link budget between block IIR and IIR-M or IIF are:

- The minimum transmission power for block IIR from the C/A code is increased by 2.71 dB relative to the GPS ICD.
- The minimum transmission power for block IIR-M and IIF for the L2 P(Y) code is increased by 3dB with respect to the block IIR.
- Blocks IIR-M and IIF transmit the L2C signal.
- Block IIF transmits the L5 signal.

The following tables provide the GPS signal power assumption and the computed EIRP for the reference elevation (5 deg).

GPS Block	L1 C/A [dBW]	L1 P [dBW]	L2 C [dBW]	L2 P [dBW]	L5 I [dBW]	L5 Q [dBW]	L1 Ellipticity [dB]	L2 Ellipticity [dB]	L5 Ellipticity [dB]
IIR	-158.5	-161.5		-164.5			1.8	2.2	2.4
IIR-M	-158.5	-161.5	-160.0	-161.5			1.8	2.2	2.4
IIF	-158.5	-161.5	-160.0	-161.5	-157.9	-157.9	1.8	2.2	2.4

**Table 7: Minimum GPS received power on Earth at 5 degrees elevation angle**

Assumptions	L1 [dB]	L2 [dB]	L5 [dB]
Receiver Antenna Gain	3.0 LP	3.0 LP	3.0 LP
Polarization Mismatch Loss	4.0	4.0	4.0
Atmospheric Loss	0.4	0.4	0.4
Free Space Loss	184.44	182.27	182.01

**Table 8: GPS EIRP computation assumptions (LP stands for linearly polarized)**

GPS Block	L1 C/A	L1 P	L2 C	L2 P	L5 I	L5 Q
IIR	30.00	24.29		19.35		
IIR-M	27.29	24.29	23.85	22.35		
IIF	27.29	24.29	23.85	22.35	25.51	25.51

**Table 9: GPS EIRP [dBW]**

## ➤ GALILEO

For the simulations, the fully deployed Galileo constellation is considered (FOC constellation). The FOC system consists of 30 SV (27 operational and 3 spares) positioned in three circular Medium Earth Orbit planes with average semi-major axis of 29601.297 km, and inclination of 56 degrees with reference to equatorial plane. The IOC constellation is also considered to compare the performances,

with 18 SVs equally distributed on 6 orbital planes. The following tables provide the Galileo signal power assumption and EIRP for the reference elevation (10 deg).

GALILEO Signal	Component	Total Received Minimum Power [dBW]
E5	E5a I+Q (50/50% power sharing)	-155.0
	E5b I+Q (50/50% power sharing)	-155.0
E6	E6 CS B+C (50/50% power sharing)	-155.0
E1	E1 OS/SoL B+C (50/50% power sharing)	-157.0

**Table 10: Galileo Minimum received power on Earth at 10 degree of elevation angle**

Assumptions	E5a [dB]	E5b [dB]	E6 [dB]	E1 [dB]
Receiver Antenna Gain	0.0 RHCP	0.0 RHCP	0.0 RHCP	0.0 RHCP
Polarization Mismatch Loss	0.6	0.6	0.6	0.6
Atmospheric Loss	0.4	0.4	0.4	0.4
Free Space Loss	182.75	182.97	183.47	185.28

**Table 11: Galileo EIRP computation assumptions**

GALILEO Signal	Component	EIRP
E5	E5a I+Q (50/50% power sharing)	28.75
	E5b I+Q (50/50% power sharing)	28.97
	Full E5	32.27

E6	E6 CS B+C (50/50% power sharing)	29.47
E1	E1 OS/SoL B+C (50/50% power sharing)	29.28

**Table 12: Galileo EIRP [dBW]**

➤ COMPASS

The Compass constellation is composed as follows: 24 MEO space vehicles plus 3 spares, equally distributed on three orbital planes with 55 deg inclination and 27878 km semi-major axis; 5 GSO space vehicles with 42146 km semi-major axis; 3 IGSO with 55 deg inclination and 42146 km semi major axis. The following tables provide the power assumptions for Compass, and the computed EIRP for the reference elevation (5 deg). The power information are not yet officially published into ICD form, but they have been taken from various publications in conferences and journals. Other assumptions, such as polarization mismatch, have been taken with reference to the GALILEO system.

COMPASS Signal	Component	Total Received Minimum Power [dBW]
B1	$C_D+C_P$	MEO: -158.2 IGSO/GSO: -158.2
	$A_D+A_P$	MEO: -156.2 IGSO/GSO: -158.2
B2	B2a $A_D$ and $A_P$	MEO: -159.0 IGSO/GSO: -160.8
	B2b $B_D$ and $B_P$	MEO: -159.0 IGSO/GSO: -160.8
B3	$A_D$ and $A_P$	MEO: -159.0 IGSO/GSO: -161.3
	B3	MEO: -156.0 IGSO/GSO: -158.3

**Table 13: Compass Minimum received power on Earth at 5 degree of elevation angle**

Assumptions	B1 [dB]	B2 [dB]	B3 [dB]
Receiver Antenna Gain	0.0	0.0	0.0
Polarization Loss	0.6	0.6	0.6
Atmospheric Loss	0.4	0.4	0.4
MEO Free Space Loss	184.89	182.47	183.01
IGSO/GSO Free Space Loss	188.68	186.25	186.80

**Table 14: Compass EIRP computation assumptions**

COMPASS Signal	Component	EIRP
B1	$C_D + C_P$	MEO: 28.09 IGSO/GSO: 31.88
	$A_D + A_P$	MEO: 30.09 IGSO/GSO: 31.88
B2	$A_D$ and $A_P$	MEO: 24.87 IGSO/GSO: 26.85
	$B_D$ and $B_P$	MEO: 24.87 IGSO/GSO: 26.85
	B2 Full	MEO: 30.89 IGSO/GSO: 32.87
B3	$A_D$ and $A_P$	MEO: 25.41 IGSO/GSO: 26.90
	B3	MEO: 28.41 IGSO/GSO: 29.60

**Table 15: Compass CP-III EIRP [dBW]**

### ➤ GLONASS

The GLONASS constellation is assumed to be as follows: 24 space vehicles equally distributed on three orbit planes with 64.8 deg inclination and 25478 km semi-major axis. There are still no information available from any ICD about the

signal power, so the following assumption are taken by replicating some GALILEO characteristics.

GLONASS Signal	Component	Total Received Minimum Power [dBW]
L1	L1OC	-155
	L1SC	-155
L5	L5OC	-155

Table 16: GLONASS CDMA Minimum received power on Earth at 5 degree of elevation angle

Assumptions	L1[dB]	L5 [dB]
Receiver Antenna Gain	3.0	3.0
Polarization Loss	1.0	1.0
Atmospheric Loss	0.4	0.4
Free Space Loss	184.04	181.51

Table 17: Glonass EIRP computation assumptions

GLONASS Signal	Component	EIRP
L1	L1OC	27.44
	L1SC	27.44
L5	L5OC	24.91

Table 18: GLONASS CDMA EIRP [dBW]

#### 2.1.4.2 Transmitting antenna radiation patterns

The transmitting antenna gain pattern for the GPS and the GALILEO are displayed in the Figure 3 and Figure 4. These patterns are assumptions for the analysis of this study and are based on typical L-band antenna models with similar

gain characteristic that are available from literature. For the GLONASS and the COMPASS systems, the GPS reference pattern is taken as the transmitting pattern.

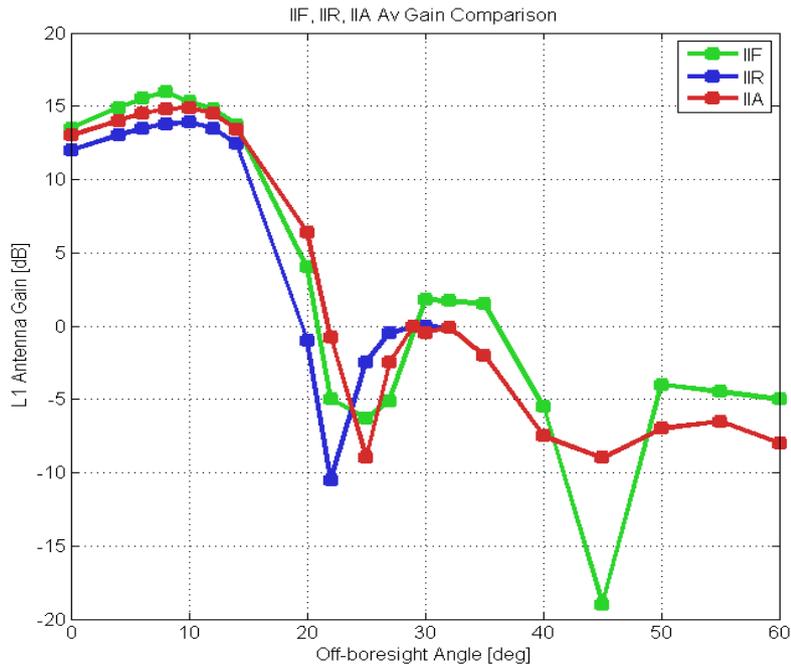


Figure 3: Assumed GPS antenna gain patterns.

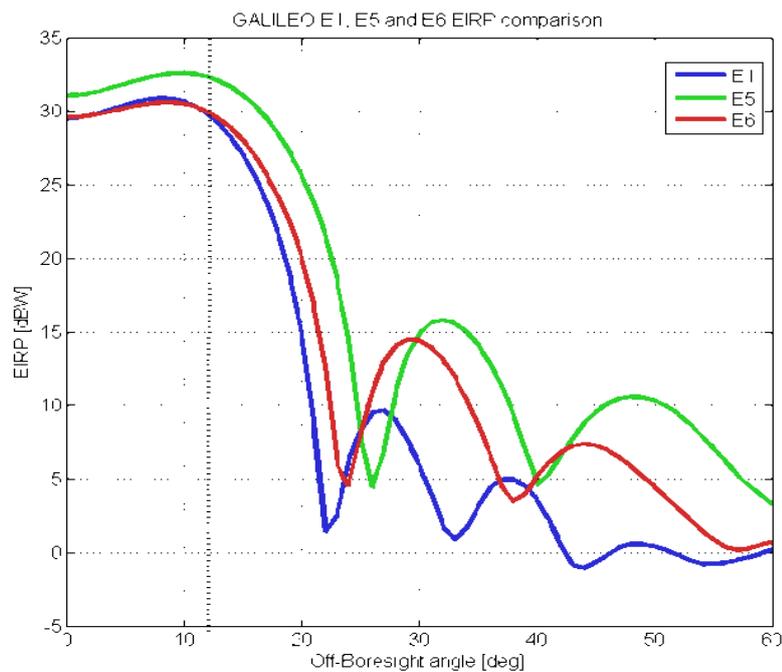


Figure 4: Assumed GALILEO EIRP antenna patterns.

### 2.1.4.3 The receiver noise model

In order to compute the signal CNo at the receiver, a simplified receiver model is implemented which is limited to Antenna-Cable-LNA. The first LNA noise figure determines most of the noise contribution, thus motivating the simplification used in the simulation. The simplified model features are taken from existing literature on GNSS space receivers [1], and from the characteristic of the LNA generally employed in this architectures, such as for example TBD:

- Loss before LNA 0,3 dB
- LNA noise figure 1.5 dB
- LNA gain 28 dB
- Loss other 1.2 dB
- Sky Temperature 60 K

The model then is:

$$CNo = EIRP - FSL - L_{OTHER} + \frac{G}{T} - k_{dB}$$

where  $EIRP$  is the Equivalent Isotropic Radiated Power,  $FSL$  is the free space loss,  $L_{OTHER}$  represents undefined losses and is generally taken as a margin to define un-modelled or unexpected hardware losses,  $k$  is the Boltzmann constant expressed in dB, and  $G/T$  is the receiver figure of merit given by

$$\frac{G}{T} = G - 10 \cdot \log(T_{sys})$$

$G$  is the LNA gain while  $T_{sys}$  is the system equivalent temperature that accounts for the contribution of the antenna and the receiver, and can be computed with

$$T_{sys} = T_{antenna} + T_r$$

$$T_{antenna} = \left( (L_{beforeLNA} - 1)290 + T_{sky} \right) / L_{beforeLNA}$$

$$T_r = \left( (10^{NF/10}) - 1 \right) 290$$

Where  $NF$  is the LNA noise figure expressed in dB. The receiver model is then completed with the receiving antenna gain pattern, expressed with an analytical equation as function of the elevation:

$$G_{ant} = 5.5 - 11.5 \times \left( (90 - e)/70 \right)^2$$

The selected model matches the characteristics of some of the antenna already employed in space mission, such as for example the Sentinel-1 GNSS antenna, and is also practical for computation because of its analytical expression. In order to simplify the simulation, the same gain pattern is used for the different GNSS signals.

#### **2.1.4.4 Co-channel interference**

The purpose of this section is to evaluate the contribution of the co-channel interference to the total C/No, due to the simultaneous presence of multiple signals from different system on the same band. When tracking a single signal on a receiver channel coming from a single GNSS space vehicle, the signals coming from the other GNSS space vehicles on the same bandwidth represents only additional noise that reduces the CNo. Considering the increasing number of satellites from different systems that are planned to be deployed in the future years, an analysis on this effect is recommended to quantify the possible CNo degradation. The following table provides simulated mean con-channel interference power during 10 orbits, meaning that the provided value is not the worst case but an average over time, and represents the unwanted power received from the specified system while tracking one signal on one receiver channel.

Signal	Central frequency [MHz]	Transmission bandwidth [MHz]	Reference bandwidth* [MHz]	Mean Co-Channel Interference Power** [dBW]
<b>GPS</b>				
L1 C/A	1575.42	20.46	4.092	-149.2
L1 P	1575.42	20.46	20.46	-153.3
L2 P	1227.6	20.46	20.46	-
L2 C	1227.6	20.46	4.092	-165.7
L5 I+Q	1176.45	24.00	20.46	-156.1
<b>GALILEO</b>				
E1 OS/SoL	1575.42	24.553	8.184	-146.1
E5a	1176.45	20.46	20.46	-
E5	1191.795	51.15	51.15	-141.8
E5b	1207.14	20.46	20.46	-
E6	1278.75	40.92	40.92	-144.4
<b>COMPASS</b>				
B1/E1 C <sub>D</sub> +C <sub>P</sub>	1575.42	32.736	8.184	-146.0
B2/E5 Full	1191.795	51.15	51.15	-142.7
<b>GLONASS CDMA</b>				
L1 OC	1575.42	20.46	4.092	-146.0
L5 OC	1176.45	20.46	20.46	-146.0

**Table 19: GNSS Mean Co-Channel Interference Power**

\* containing > 90% of the signal power

\*\* mean over 10 receiver orbit simulation, constellation composition described in section TBD, mask elevation angle set to 0 deg during simulations

The overlapping bands for different systems and signals are highlighted with colours in the table Table 19. For the purpose of interference computation, only the bands that are used by all the system are considered, since this is the case

where the higher co-channel interference is expected to be present. According to the specified reference bandwidth on which we assume to receive more than 90% of the interference power, the power can be expressed in terms of the mean equivalent noise temperature summing the contribution on each band. For example for the GPS system only, we can sum the contribution of the P signal overlapped with the C/A signal.

$$T_{GPS-L1} = \left( \frac{P_{GPS-CA}}{B_{GPS-CA}} + \frac{P_{GPS-P}}{B_{GPS-P}} \right) / k$$

The same concept applies to other signal of interest, providing the total mean equivalent noise temperatures that are summarized in the table Table 20.

Band	GPS	GPS + GALILEO	GPS + COMPASS	GPS + GALILEO + COMPASS	GPS + GALILEO + COMPASS + GLONASS
L1/E1/B1	23	45	45	67	111
E5	< 1	9	8	17	26

**Table 20: Mean Equivalent Noise Temperature Related to Co-Channel Interference [K]**

The reported temperature then is summed to the receiver system temperature  $T_{sys}$  according to the model of the previous section. If we compute the receiver figure of merit with parameters as reported in the previous section, we can see that in the case of four systems (GPS + GALILEO + GLONASS + COMPASS) the mean C/No degradation of a single receiver channel which is tracking for example one L1 C/A signal from one GPS SV is around 2.5 dB. This mean result is in accordance with literature publications [6], which reports worst case interference of about 3dB. According to this value we do not consider the degradation relevant for the purpose of this study, therefore neglecting its contribution for the rest of the analysis.

### 2.1.4.5 Link budget analysis

According to the assumption described in previous sections, mission simulations have been performed in order to evaluate the received signal C/No. The Figure 5 provides the minimum receiver C/No as a function of the signal elevation for different GPS and GALILEO signals. The differences between the signals are mainly related to the differences in antenna gain patterns and transmitted EIRP.

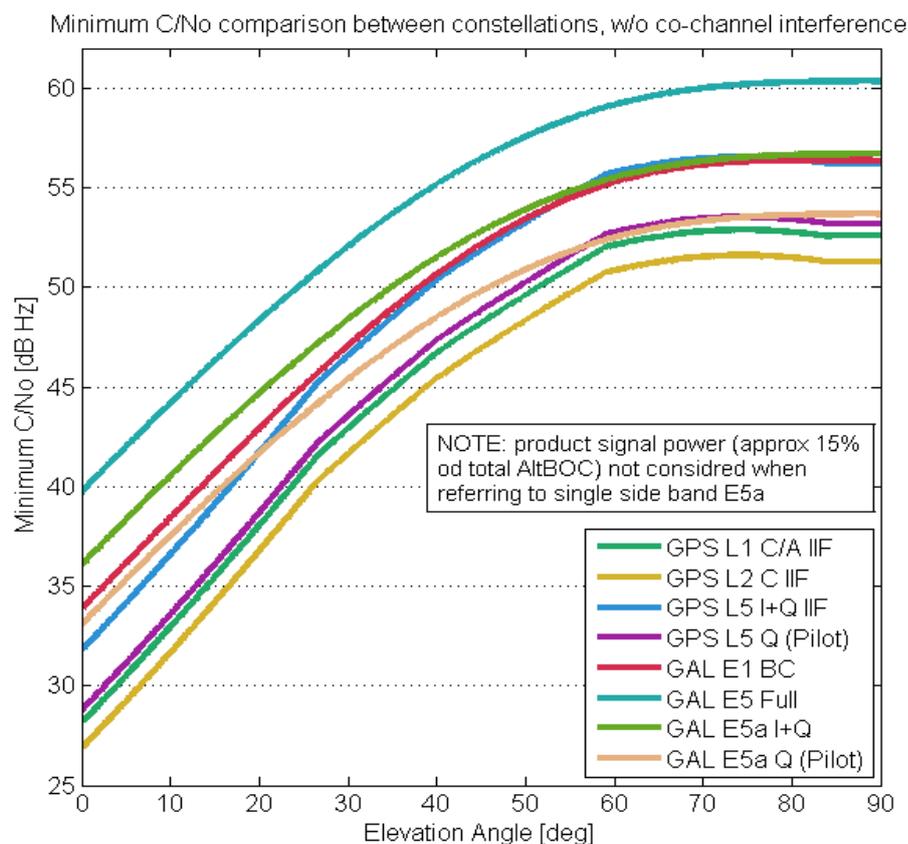
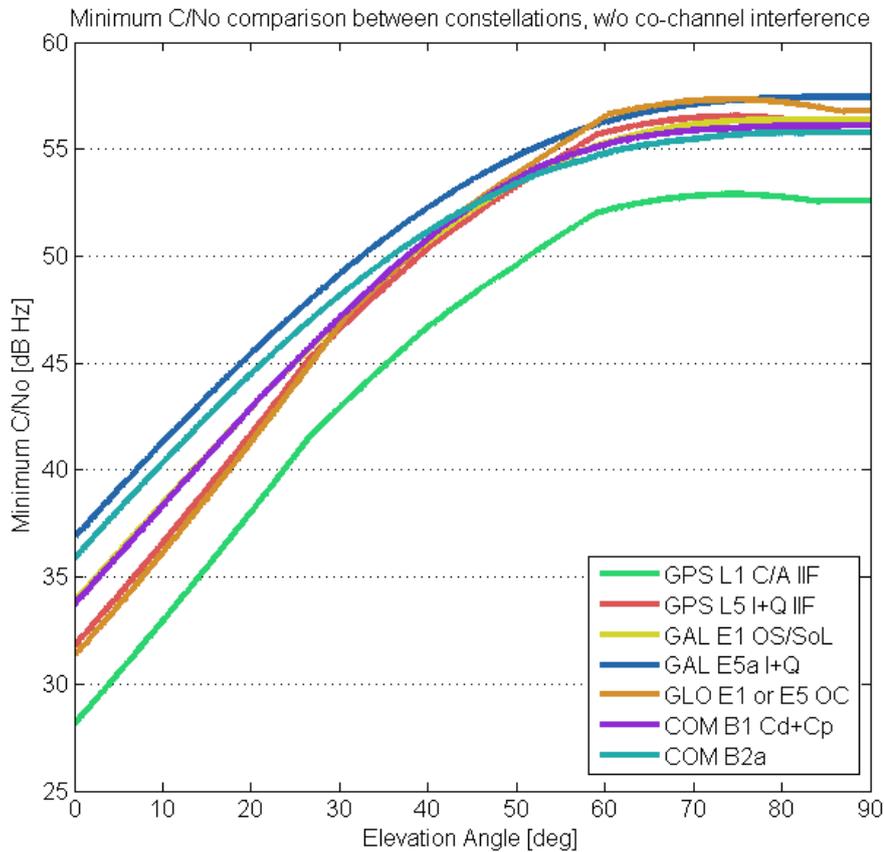


Figure 5: GPS and Galileo minimum C/No comparison

The Figure 6 provides the minimum receiver C/No as a function of the signal elevation for different GPS, GALILEO, COMPASS and GLONASS signals. The differences between the signals are mainly related to the differences in antenna gain patterns and transmitted EIRP.



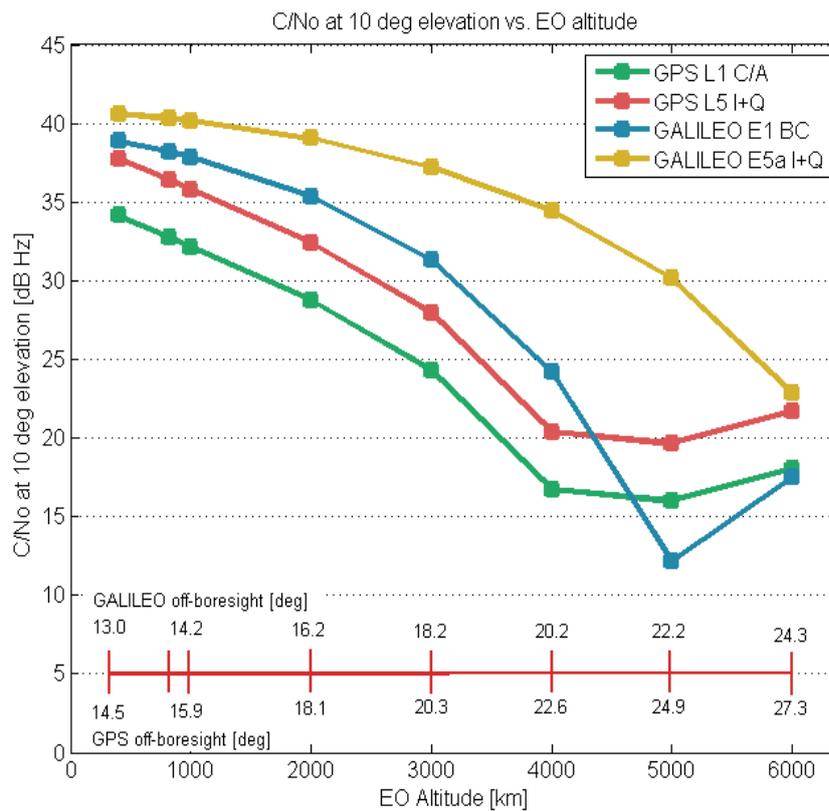
**Figure 6: GPS, GALILEO, GLONASS, COMPASS minimum C/No comparison**

A sensitivity analysis for the minimum received C/No at 10 deg of elevation with respect to the LEO altitude has also been performed. The purpose of the analysis is to identify the masking angle above which all satellite in view can be tracked and their signal decoded, according to reference receiver sensitivity taken equal to 27 dB Hz. This is the minimum required CNo to decode the navigation message, however, to track the signal is sufficient to have a C/No down to 20 dB Hz depending on the receiver implementation. These values, the minimum CNo for signal message decoding and the minimum CNo for signal tracking, are derived from existing GNSS receiver as reported in Table 21. This sensitivity analysis has been restricted only to the GPS and GALILEO systems, since they are the only two systems with official published ICD and transmitting antenna pattern data,

while for the GLONASS and COMPASS, GPS or GALILEO alike assumptions have been taken.

Receiver	Sensitivity	Developer
Pivot	27 dB Hz	NASA GSFC
PolaRx2	30 dB Hz	Septentrio
TOPSTAR 3000	20 dB Hz	TAS-F
Viceroy	37 dB Hz	General Dynamics
GPSPOD	30 dB Hz	RUAG
Mosaic	30 dB Hz	EADS Astrium

**Table 21: Existing GNSS space receiver and their sensitivity**



**Figure 7: GPS and GALILEO C/No at 10 deg elevation vs. LEO altitude**

As already mentioned, if we assume the receiver sensitivity to be taken equal to 27 dB Hz, then from the previous CNo curves we can conclude that all signals can be tracked and decoded down to 0 deg elevation for the reference application (650 km altitude). At the same time we can conclude that the GPS C/A code can be decoded down to 10 deg up to 2500 km altitude while the Galileo E5a signal can be decoded down to 10 deg elevation up to 5500 km altitude. The Galileo E1 signal can be decoded down to 10 deg of elevation up to 3000 km altitude. From the link budget analysis, we can finally conclude that no specific techniques for weak signal tracking and decoding are required for the GNSS receiver implementation up to a mission altitude of about 2500-3000 km, and that a target sensitivity of about 35 dB Hz is sufficient for the receiver operations, to track all the signal with a masking angle of approximately 10 deg. The generated CNo curves, will also serve to preliminarily evaluate the code and carrier tracking errors, and consequently the positioning errors generated by the estimation algorithms.

### 2.1.5 Expected Doppler and Doppler Rate

This section shows some relevant properties of the received GNSS signals, in terms of Doppler, Doppler rate and range rate, that are derived from mission simulations. These quantities are important when designing the receiver acquisition and tracking algorithms, since they define the acquisition time and the tracking steady state errors. The relevant parameters are summarized in Table 22 and Table 23 for the GPS and GALILEO systems and different altitudes of the LEO mission, while similar values can be derived for the GLONASS and COMPASS systems.

	400 km altitude		817 km altitude		1000 km altitude	
Max. Values	GPS	GALILEO	GPS	GALILEO	GPS	GALILEO
<b>L1/E1 Doppler</b>	44 kHz	42 kHz	43 kHz	41 kHz	42 kHz	40 kHz

<b>L2 Doppler</b>	34 kHz	-	34 kHz	-	33 kHz	-
<b>L5/E5 Doppler</b>	33 kHz	32 kHz	33 kHz	31 kHz	32 kHz	30 kHz
<b>E5a Doppler</b>	-	31 kHz	-	31 kHz	-	30 kHz
<b>L1/E1 Doppler Rate</b>	75 Hz/s	66 Hz/s	69 Hz/s	60 Hz/s	67 Hz/s	58 Hz/s
<b>L2 Doppler Rate</b>	58 Hz/s	-	54 Hz/s	-	52 Hz/s	-
<b>E5 Doppler Rate</b>	57 Hz/s	50 Hz/s	52 Hz/s	46 Hz/s	51 Hz/s	44 Hz/s
<b>L5/E5a Doppler Rate</b>	-	50 Hz/s	-	45 Hz/s	-	43 Hz/s
<b>Range Velocity</b>	8.3 km/s	8.0 km/s	8.2 km/s	7.8 km/s	8.0 km/s	7.7 km/s
<b>Range Acceleration</b>	0.014 km/s <sup>2</sup>	0.013 km/s <sup>2</sup>	0.013 km/s <sup>2</sup>	0.011 km/s <sup>2</sup>	0.013 km/s <sup>2</sup>	0.011 km/s <sup>2</sup>

**Table 22: Maximum Doppler, Doppler rate and range acceleration at different LEO altitudes (1)**

	2000 km altitude		4000 km altitude		6000 km altitude	
Max. Values	GPS	GALILEO	GPS	GALILEO	GPS	GALILEO
<b>L1/E1 Doppler</b>	41 kHz	38 kHz	38 kHz	35 kHz	36 kHz	36 kHz
<b>L2 Doppler</b>	32 kHz	-	30 kHz	-	28 kHz	-
<b>L5/E5 Doppler</b>	31 kHz	29 kHz	29 kHz	27 kHz	28 kHz	25 kHz
<b>E5a Doppler</b>	-	29 kHz	-	26 kHz	-	25 kHz
<b>L1/E1 Doppler Rate</b>	57 Hz/s	48 Hz/s	46 Hz/s	37 Hz/s	41 Hz/s	31 Hz/s
<b>L2 Doppler Rate</b>	45 Hz/s	-	36 Hz/s	-	32 Hz/s	-
<b>E5 Doppler Rate</b>	43 Hz/s	37 Hz/s	35 Hz/s	28 Hz/s	31 Hz/s	24 Hz/s

<b>L5/E5a Doppler Rate</b>	-	36 Hz/s	-	28 Hz/s	-	23 Hz/s
<b>Range Velocity</b>	7.7 km/s	7.3 km/s	7.3 km/s	6.7 km/s	6.9 km/s	6.4 km/s
<b>Range Acceleration</b>	0.011 km/s <sup>2</sup>	0.009 km/s <sup>2</sup>	0.009 km/s <sup>2</sup>	0.007 km/s <sup>2</sup>	0.008 km/s <sup>2</sup>	0.006 km/s <sup>2</sup>

**Table 23: Maximum Doppler, Doppler rate and range acceleration at different LEO altitudes (2)**

When acquiring and synchronizing the GNSS signals, the first step consists of a search algorithm, a two dimension process that spans the SV unique *Pseudo-Random Noise PRN* codes and the carrier Doppler frequency. The simplest of these algorithms is the serial search, that divides the two dimensions space in discrete cells, which are serially evaluated against their probability of detection. More details on the synchronization algorithm are provided in section 3. For the preliminary evaluation described in this section, the time required to perform a simple serial search can be computed starting from the Doppler data of the table Table 22, thus indicating if the serial search is feasible or if more complex algorithms are to be implemented. The complete serial search time with 1ms integration period (which is adequate since no weak signal conditions are foreseen), and with no a priori information that can be exploited to restrict the search space, is:

1. GPS L1 C/A

- 8 channels: 20 min
- 10 channels: 16 min
- 12 channels: 13 min

1. GALILEO E1-B

- 8 channels: 70 min
- 10 channels: 55 min
- 12 channels: 45 min

The differences between the two representative signals are due to the different length of their spreading codes. These time values indicates that a parallel search technique is fully recommend, or alternatively a priori information (GNSS system almanac and rough receiver position) is required to restrict the serial search space and reduce the search time. Doppler rate has to be taken into account when selecting tracking algorithm for the steady state error. Third order PLL and DLL are recommended to null steady state error associated with phase and range accelerations.

### **2.1.6 Visibility and DOP**

This section provides the visibility results for the GPS, GALILEO, COMPASS and GLONASS system, related to the reference user application. Minimum, maximum and average number of visible satellites with a masking angle of 10 deg are reported as resulted from system simulation. This analysis is essential in order to define the receiver number of channels, which is a compromise between the capability to acquire all the signal that are in view and the receiver implementation complexity. Then the minimum, maximum and average *Dilution Of Precision DOP* is provided for the considered systems and their combinations. This quantity is a pure geometric term that describes how the errors associated with the raw measurements are translated and amplified into the receiver positioning errors.

The Figure 8 and Figure 9 displayed the visibility time series for the GPS and GALIELO systems, and for the COMPASS and GLOANSS systems, while Figure 10 provides representative visibility histograms. The minimum, maximum and average signals in view are then summarized in Table 24.

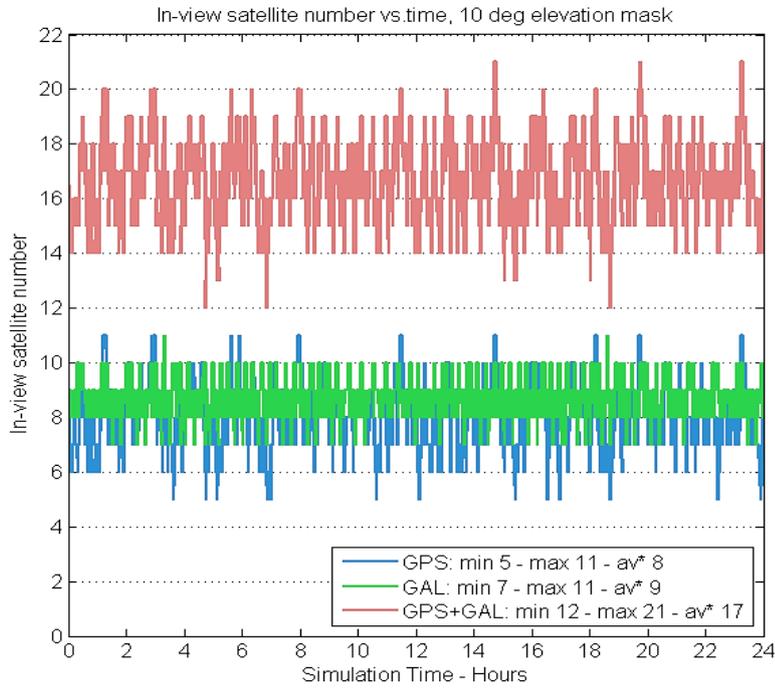


Figure 8: GPS and GALILEO visibility time series

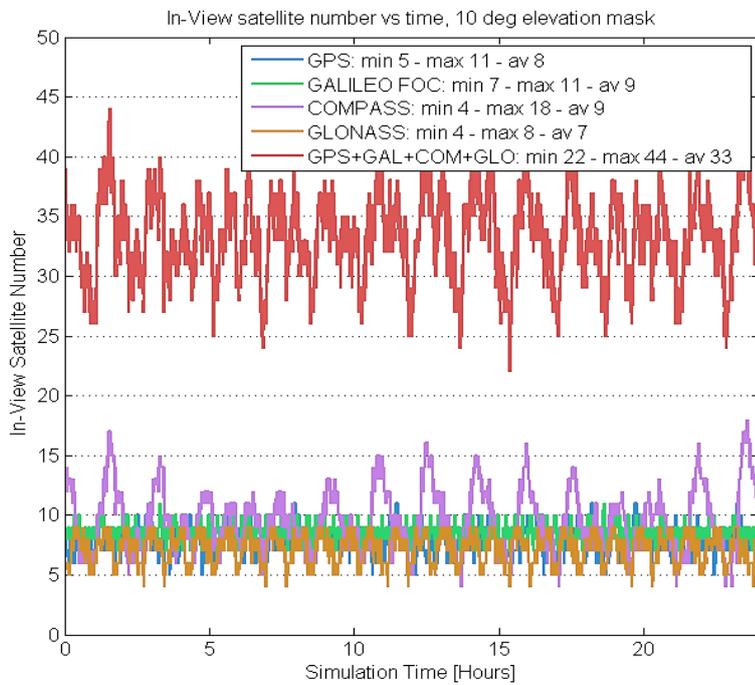
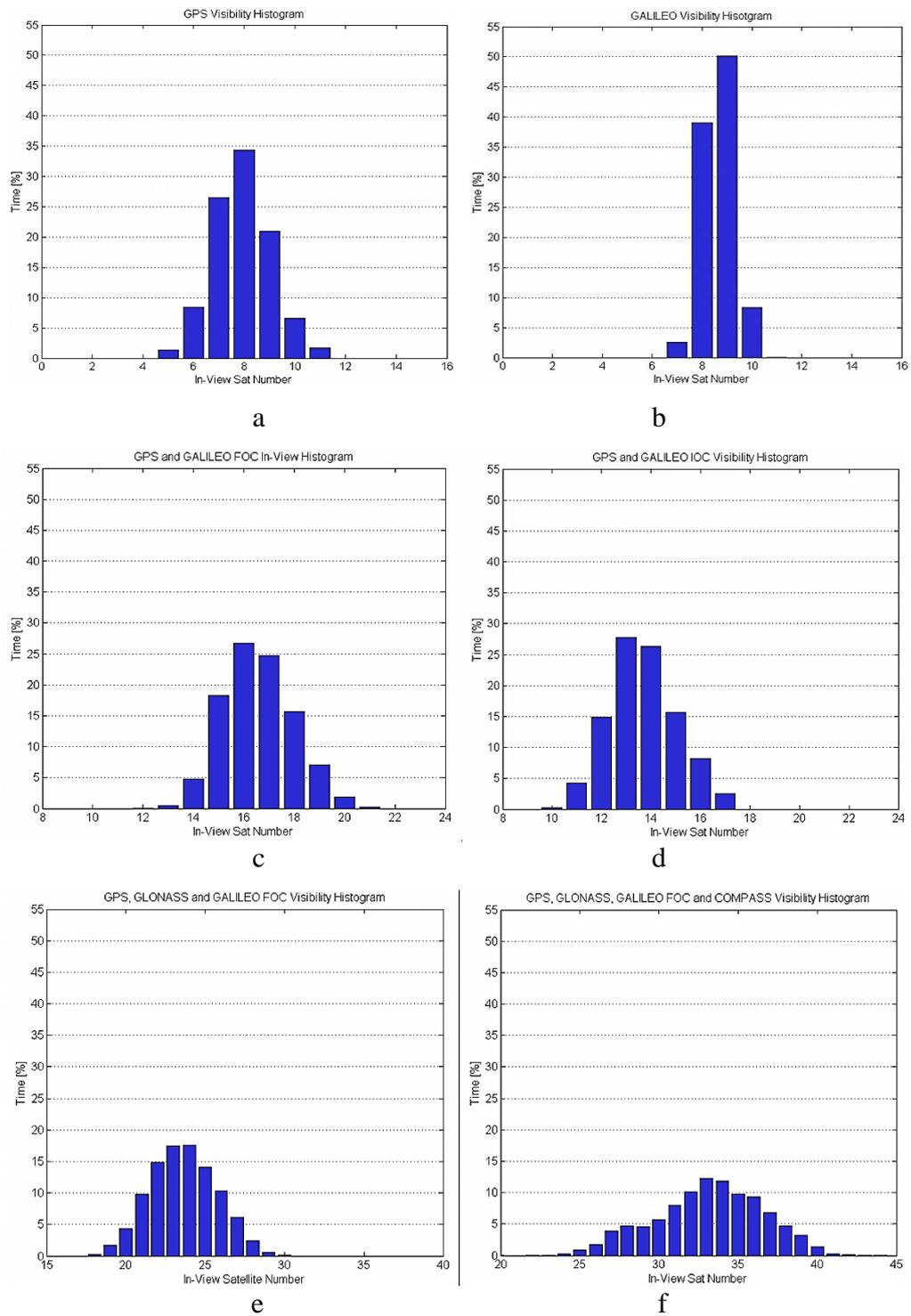


Figure 9: GPS, GALILEO GLONASS and COMPASS visibility time series



**Figure 10: Visibility histograms: a) GPS histogram, b) GALILEO FOC histogram, c) GPS + GALILEO FOC histogram, d) GPS + GALILEO IOC histogram, e) GPS + GLONASS + GALILEO FOC histogram, f) GPS + GLONASS + GALILEO FOC + COMPASS histogram**

	Min	Max	Average*
GPS	5	11	8
COMPASS	4	18	9
GLONASS	4	9	7
GALILEO IOC	4	8	6
GALILEO FOC	7	11	9
GPS+GALILEO FOC	12	21	17
GPS+GALILEO IOC	10	17	14
GPS+GLONASS	10	20	15
GPS+GALILEO FOC+GLONASS	18	30	24
GPS+GALILEO IOC+GLONASS	15	26	21
GPS+GALILEO FOC+COMPASS+GLONASS	22	44	33
GPS+GALILEO IOC+COMPASS+GLONASS	19	41	30

**Table 24: All constellation visibility statistic (\* rounded to nearest integer)**

Related to the provided visibility statistics, the Table 25 shows the percentage of time in which the receiver is capable of tracking all the signal from satellites that are in view, for a given number of receiver channel.

Channels	8	10	12
GPS	71%	98%	100%
GALILEO FOC	42%	99%	100%
GALILEO IOC	100%	100%	100%

Channels	16	18	20
GPS+GALILEO FOC	50%	91%	99%
GPS+GALILEO IOC	97%	100%	100%
Channels	22	24	26
GPS+GALILEO FOC+GLONASS	31%	66%	91%
GPS+GALILEO IOC+GLONASS	80%	96%	100%
Channels	34	36	38
GPS+GALILEO FOC+GLONASS+COMPASS	64%	83%	95%
GPS+GALILEO IOC+GLONASS+COMPASS	90%	97%	100%

**Table 25: % of time in which the receiver track all signals in view with given number of channels**

According to the information presented in this section about the visibility analysis, the following conclusion can be drawn:

1. Single Constellations

- A 10 channel receiver is capable to track all satellite in view more than 90% of the time.

2. Dual Constellations

- GPS+GALILEO FOC and GPS+GLONASS: 18 channels receiver is capable to track al the satellite that are in view more than 90% of the time
- GPS+GALILEO IOC: 16 channels receiver is capable to track al the satellite that are in view more than 90% of the time
- GPS+COMPASS: 22 channels receiver is capable to track al the satellite that are in view more than 90% of the time. The increment

with respect to the other constellations is due to the GEOS and IGSOs located over the Asian region.

### 3. Three Constellations

- o GPS+GALILEO FOC+GLONASS: 26 channels receiver is capable to track all the satellites that are in view more than 90% of the time

### 4. Four Constellations

- o GPS+GALILEO FOC+GLONASS+COMPASS: 38 channels receiver is capable to track all the satellites that are in view more than 90% of the time

The visibility analysis described in the section, is not sufficient to define the receiver number of channels: having clear that the minimum number of signals to compute the user position is four, the additional signals contribute to improve the quality of the final position estimation through the DOP parameter.

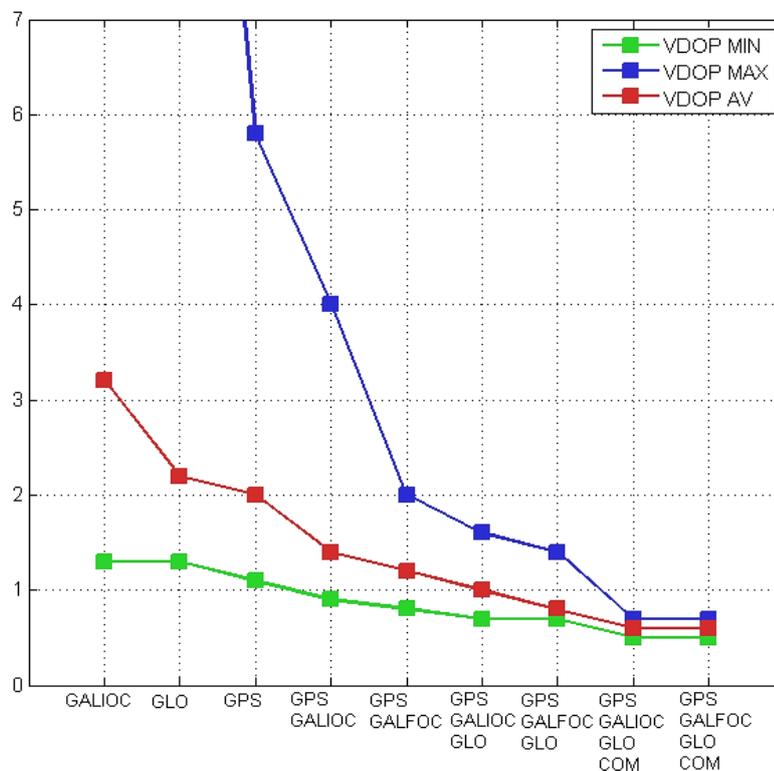
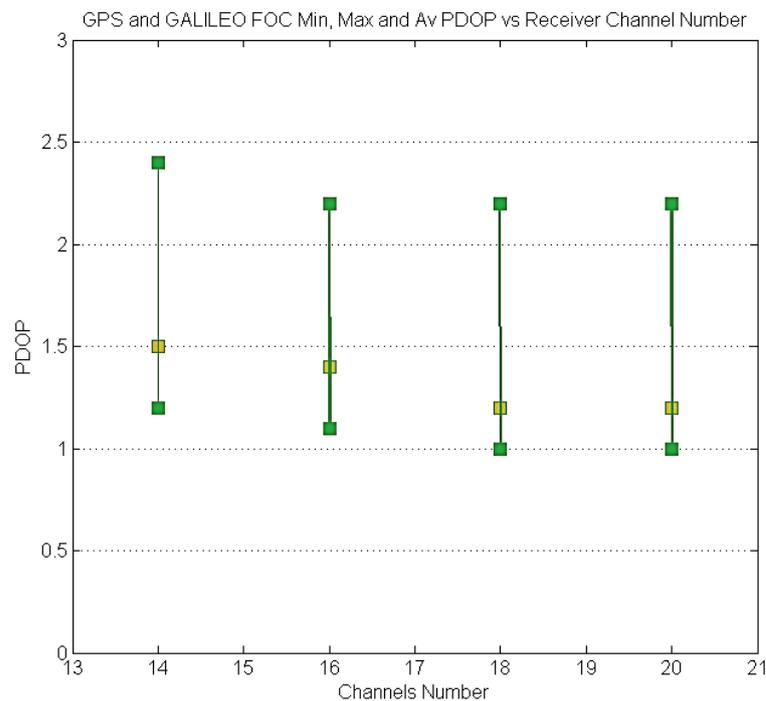


Figure 11: VDOP (min, max and average) as function of different systems combinations

Only relevant improvements in the DOP justify the increasing number of channel with the related implementation complexity. Therefore an analysis of the DOP resulting from the combination of multiple systems is required to understand if all signal in view have to be acquired, or only a subset of them is sufficient to obtain the required position accuracy. The Figure 11 displays the *VDOP* (*Vertical Dilution of Precision*, is a measure of how the satellites geometry contributes to error of the local vertical position component) for different single and multi-constellation combinations. It is displayed that up to the combination of two systems, for example GPS and GALILEO, the improvement in DOP is relevant, while it becomes much smaller when adding other system capabilities to the receiver. We can therefore assume that the additional complexity of having a receiver compatible with three or four systems, each of them with their signals and spreading code, is not justified by the improved satellite geometry quality.



**Figure 12: PDOP (min, max and average) as function of receiver number of channels for a GPA + GALILEO FOC receiver**

As opposite to the ground user, where the higher number of GNSS space vehicles could provide benefits in heavily shadowed environment, two systems are sufficient for the space user to improve the DOP, especially its maximum value, while avoiding to excessively increase the architecture complexity. If we decide to reduce the receiver compatibility only to the GPS and GALILEO systems, according to the previous visibility analysis 18 channels are sufficient to track all the signal in view more than 90% of the operating time. We could also decide to reduce the number of channels because of specific limitation of the signal processing IC, with a related increasing of the DOP values as displayed in Figure 12.

### 2.1.7 Receiver requirements summary

According to the ALMASat-EO mission specifications and the mission simulation described in the previous sections, the Table 26 summarizes the GNSS receiver requirements.

Requirement	Value	Notes
Position error 3D RMS	Better than 10m	3-sigma
System compatibility	GPS GALILEO	
Signals	GPS L1 L2 GALILEO E1	Second frequency L2 to be used only for tests purposes; the operative condition includes only the L1/E1 frequency with GPS and GALILEO observables.
Antenna	3dBi LP	Or 0dBi RHCP
First LNA Noise Figure	1.5dB	
First LNA Gain	25dB	
Number of channels	14-18	Actual number of channels depends on the available resources on the selected FPGA model.

Search strategy	Assisted	Additional information shall be provided to the GNSS receiver in order to speed up the search process, such as GNSS satellite almanac and LEO satellite rough position estimation through TLE.
Probability of false alarm	1%-5%	
PLL architecture	III Order	
DLL architecture	III Order	
Estimation algorithm	EKF based	
Data interface	CAN – RS232	
Power consumption	<10W	
Power interface	5V DC	

**Table 26: Receiver requirements summary**

## 2.2 GNSS receiver architecture

This section provides an overview on the GNSS receiver electronic architecture and prototyping: components and physical characteristics of the receiver prototype are presented, focusing on the most critical aspects, the RF front-end and the FPGA logic design. The software architecture, the structure of drivers and user applications is also described in the section. The receiver prototype is a dual front-end software defined receiver, based on the Xilinx Virtex5 FPGA FXT series, which incorporates a 32-bit PowerPC PPC440 in form of hard-processor, therefore not encoded using the FPGA logic. The receiver block diagram is displayed in Figure 13, while the prototype PCB is displayed in Figure 14. The lowest FPGA speed grade guarantees operations up to 400MHz, and although the processor does not incorporate built-in FPU, external soft FPU can be attached giving approximately 35 MFLOPS. The first prototype, named *Gemini Alpha*, features 64MB of 200MHz DDR2 SDRAM and 16MB of Flash memory, which is used to store both the FPGA configuration file and the application software. The DDR2 ram is used both to load the operating system and to implement a disk where to record the file system, the so called *RAMdisk*.

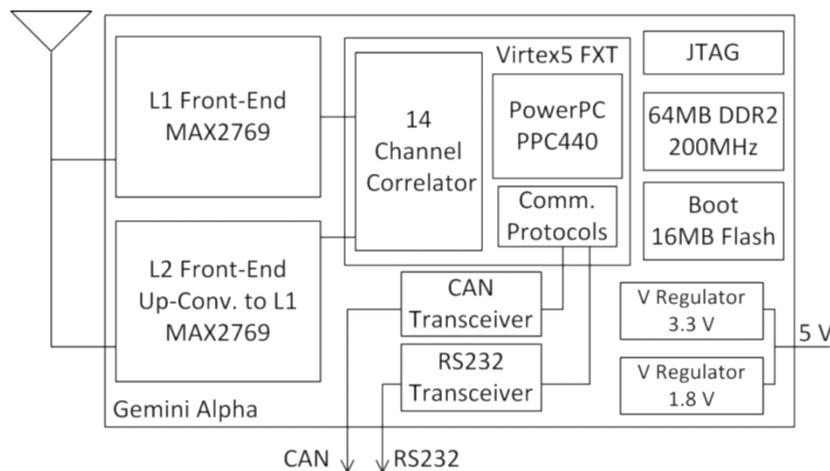


Figure 13: GNSS receiver prototype block diagram

Two RS232 and one CAN interface are used to configure and get data from the receiver: the RS232 protocol is implemented in FPGA logic and is part of the

*Intellectual Property* libraries provided by Xilinx, while the CAN uses and external chipset provided by Microchip and connected using the *Serial Peripheral Interface SPI*.

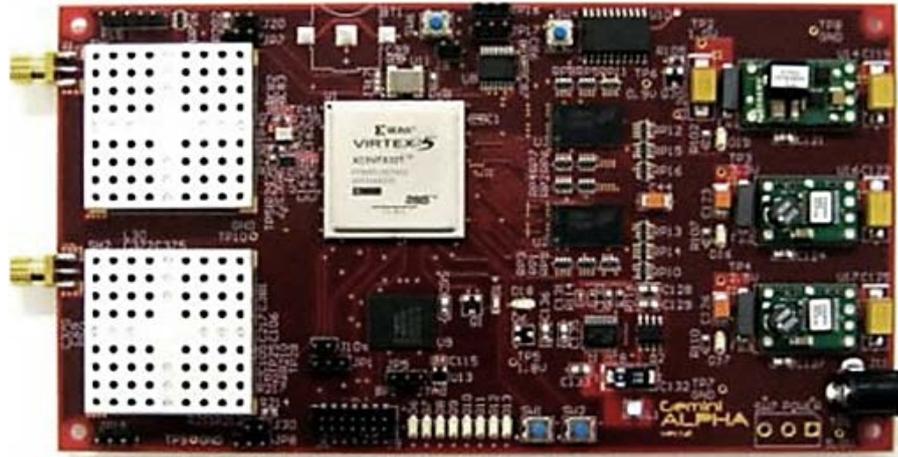


Figure 14: GNSS receiver prototype PCB

The PCB itself is a 12 layer board, with dimension of 90x164x1.5mm: the layer stack is displayed in Figure 15 while Figure 16 provides an overview of the circuit CAD design from the Altium CAD software.

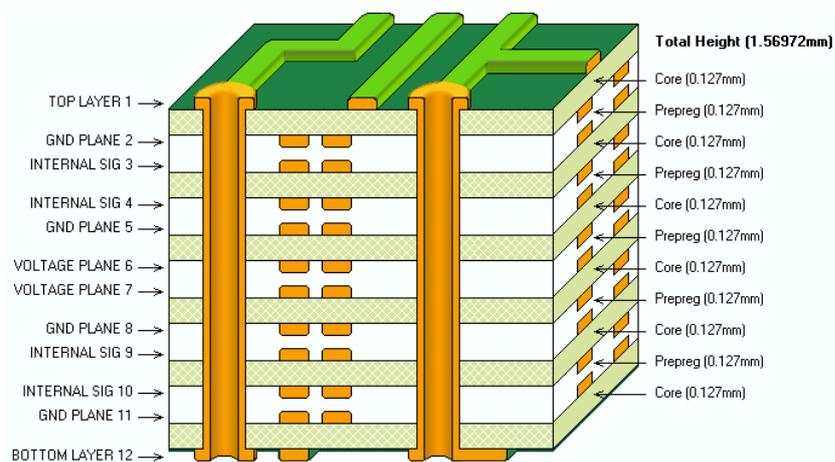


Figure 15: GNSS receiver prototype layer stack from Altium Stack Manager

For the PCB manufacturing the following criteria have been applied as a constrain to the manufacturing company:

- Material: FR4 class epoxy glass.

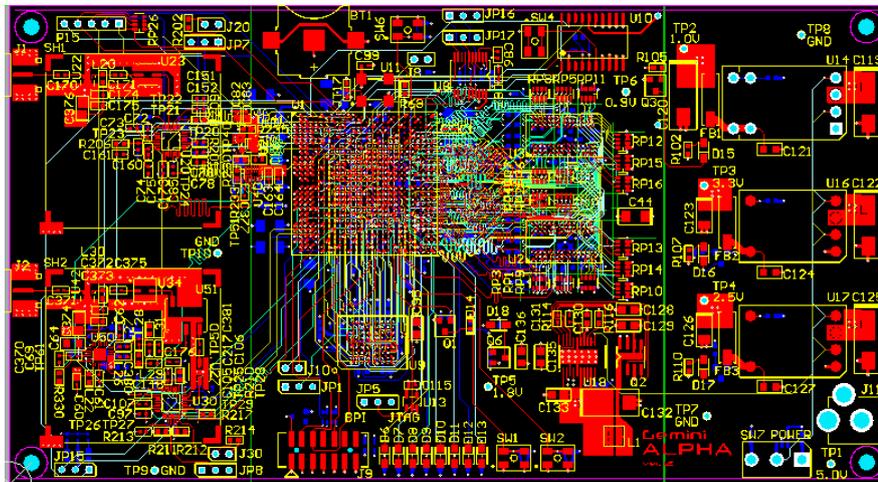


Figure 16: GNSS receiver prototype CAD (internal layers not shown)

- 0.5 Oz copper (17.5um) for external and internal layers.
- Finish: electro-less nickel immersion gold.
- Mask colour: red.
- Finished board thickness:  $62 \pm 7$  mils.
- Control TRACE-GND impedance of  $50\text{Ohm} \pm 10\%$  for all internal traces of width 4 mils.
- Control TRACE-GND impedance of  $50\text{Ohm} \pm 10\%$  for all external traces of width 8 mils.

The first receiver prototype was designed with the objective of validating the GNSS architecture; therefore it does not take into account any strategy that is usually applied to the electronic systems for space applications, which are aimed to reduce the power consumption and provide more reliability against the space environment. The final device realizations will include power saving strategies and radiation mitigation techniques, but the general guidelines can be summarized in the following considerations. The radiation environment is composed of high energy particles that can be divided into 4 groups: Galactic Cosmic Rays, Trapped Radiation, heavy nuclei associated with solar events and Neutrons. The associated energy goes from a minimum of 40KeV for electrons to a maximum of 1MeV for protons, neutrons and heavy ions. Their effects on electronics are in

most cases associated with charge deposition in the IC semiconductors, and the associated phenomena goes from the shift of MOS transistor threshold voltages, to the more destructive polarization of parasitic BJT's inside CMOS structure that leads to high current absorption. The events are classified, according to their causes, in cumulative or single. Cumulative effects are caused by low energy radiation that slowly adds charges to the semiconductor, while single events are associated with high energy particles that causes immediate failure. The firsts are generally not a serious threat in short LEO missions, and they can be mitigated by protecting critical IC with high-Z shielding materials such as tungsten. On the opposite side, the single events, even though having low probability, represent a serious threat, since they are practically unavoidable. What is implemented is generally only aimed to recover the system in a post-SEU situation. The simplest guideline to follow for non-critical systems such as the receiver, where a short gap in available data does not affect dramatically the satellite performance, is to perform scheduled maintenance, or in other words to perform periodic FPGA reconfiguration and receiver reset despite the presence of an event. This, together with the build in logic redundancy (there are 14 identical correlator channels) represents a sufficient strategy to achieve the desired system reliability for a short LEO mission, without having to withstand the huge costs associated with a radiation hardened IC development.

### **2.2.1 IC front-end**

Both the receiver front-ends are based on the Maxim MAX2769 IC that incorporates a single stage down-converter and two two-bit ADCs for the in-phase and quadrature components. The first front-end is designed for the GPS L1 or GALILEO E1 signal, while the second is designed for the L2 signal through up-conversion from 1227.6MHz to 1575.42MHz. The L2 signal is used only for test purposes and has been included in the prototype receiver in order to verify the capability to use the same front-end IC. However the operative condition of the receiver includes only the L1/E1 carrier with both GPS and GALILEO

observables. The selected front-end is the first IC on the market which is ready for the GALILEO signal, since it features an IF bandwidth up to 4.2MHz, required to capture the main lobe of the GALILEO E1 signal. Internal details of the selected component are displayed in Figure 17.

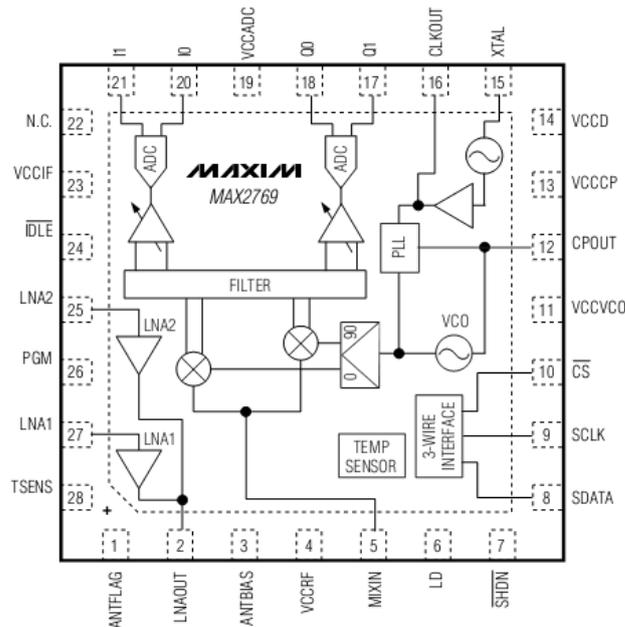


Figure 17: MAX2769 pin configuration and block diagram, reproduced from component datasheet

As displayed the LO tuning frequency is generated by a PLL with external filter connected to CPOUT, provided that a reference oscillator is passed to the IC at the XTAL input. The selected oscillator is a Connor Winfield clipped sine wave TCXO at 16,368MHz, which is an integer fraction of the required LO frequency. With selected TCXO and PLL configuration, the front-end provides an IF at 4,092MHz with approximately 4MHz 3dB bandwidth, sampled at 16,368MHz with two bit coding for both the I and Q channels. The data are streamed at the I1 I0 and Q1 Q0 pin while the reference clock is provided at the CLKOUT pin. I and Q samples are provided because the IC is capable also of zero-IF conversion, but since in will be used in low-IF mode, only the I output is streamed to the FPGA. The selection of the proper crystal oscillator is an important factor because it influences directly the phase fluctuation of the PLLs, both the one inside the front-

end used to generate the LO frequency, and the loop used to track the GNSS signals. For the selected oscillator family the Figure 18 shows the phase noise, a typical quantity that is provided by oscillator manufacturers to describe their devices. It represents a measure of spectral density of phase fluctuations, expressed in  $dBc/Hz$ , where  $dBc$  is the power respect to the carrier.

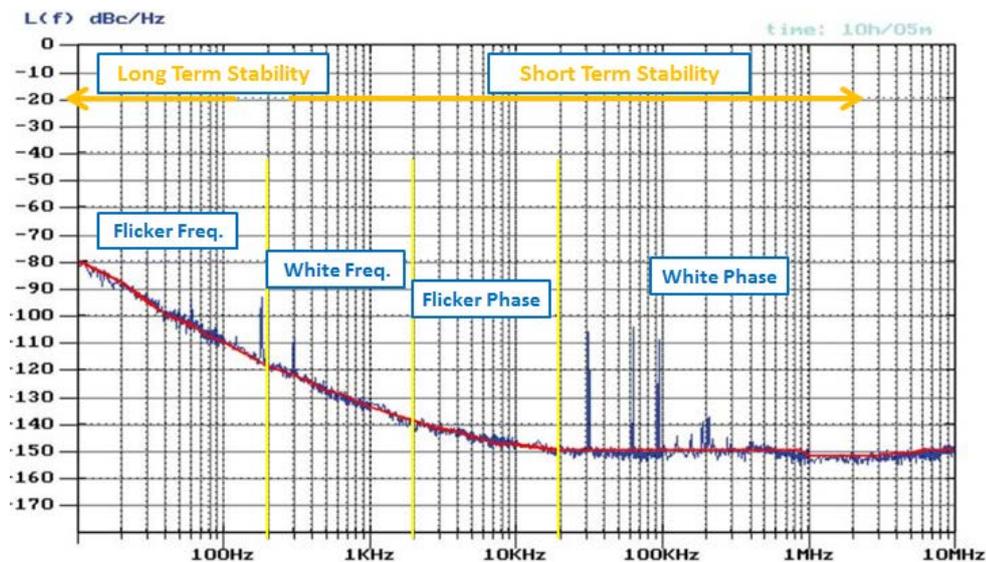
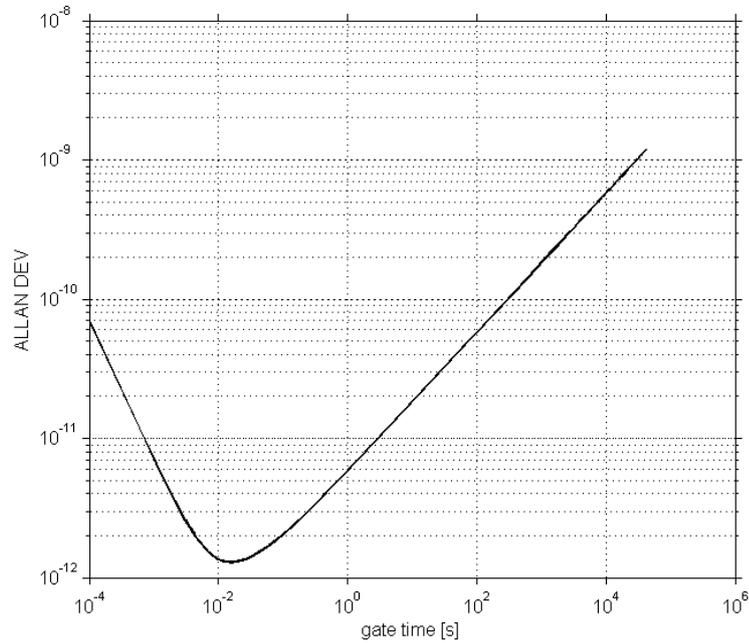


Figure 18: Connor Winfield TCXO phase noise, reproduced from the component datasheet [TBD].

The picture shows also some of the noise types that generally affects the oscillators, which can be identified with the slope of the provided quantity. In order to define the influence of the oscillator on the PLL, some empirical equations relates the PLL jitter to the employed oscillator Allan deviation [4], while a conservative rule of thumb states that Allan deviation better than  $1 \times 10^{-10}$  at the given PLL gate time (the inverse of the PLL bandwidth) provides reliable PLL operations. The algorithm to translate the phase noise diagram into an approximated oscillator Allan deviation is not reported here, but can be found in [7]. With this algorithm applied to the selected TCXO, the Allan deviation is displayed in Figure 19.



**Figure 19: Connor Winfield TCXO Allan deviation.**

Since the PLLs bandwidth, both the front-end one and GNSS signal tracking one, is between 1Hz and 20Hz, the gate time of interest are below 1s. Therefore we can assume that the selected TCXO is compliant with reliable operations of the PLL according to the reported rule of thumb.

The LNA1 and LNA2 which are part of the MAX2679 are not used in the receiver prototype, but a single external LNA has been mounted because of its higher gain and lower noise figure. The Figure 20 displays the front end diagram with external components up to signal quantization and digital conversion, performed inside the MAX2679, while Table 27 summarizes the features of the selected components and internal MAX2679 stages.

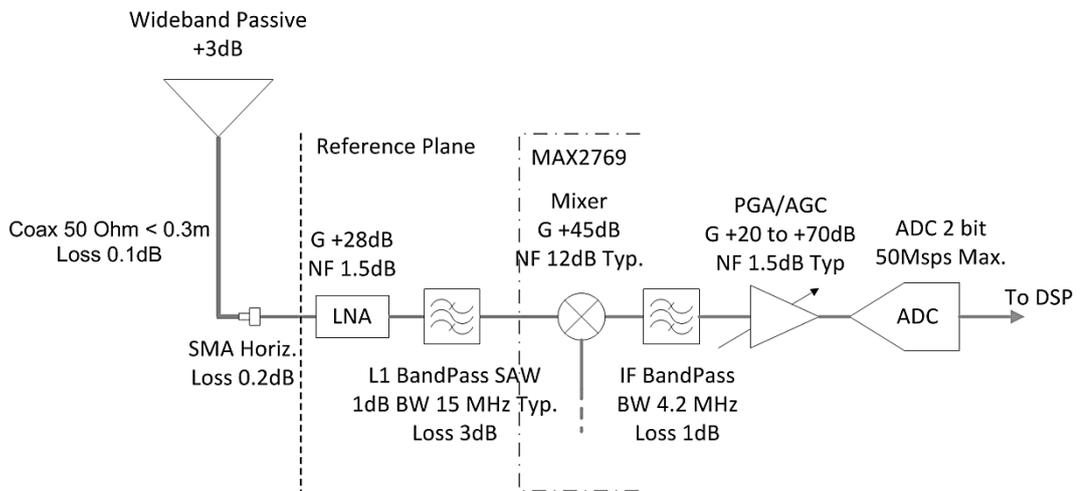


Figure 20: Front end RF cascade

As already introduced, the front-end for the L2 signal uses the same IC as the L1 front-end, with the addition of an up-conversion stage, since the MAX2769 is designed for the L1 frequency. The up-conversion stage is composed of a VCO which generates the LO frequency, and a mixer. The selected up-conversion mixer and VCO characteristics are summarized in Table 28.

Component	Parameter	Value
LNA SGL-0622Z	GAIN	28 dB
	NOISE FIGURE	1.5 dB @ 1575MHz
	POWER	10.5 mA @ 3.3V
	REVERSE ISOLATION	-28 dB
	INPUT IIP3	-13 dB
L1 BANDPASS SF1186B-2	CENTER FREQ.	1575.42 MHz
	1dB PASSBAND	15.3 MHz
	PASSBAND VARIATION	MAX 1.0 dB
	INSERTION LOSS	3 dB
	MAX SIGNAL LEVEL	+10 dBm
TCXO	SUPPLY	2 mA @ 3.3 Vdc

D32G	FREQUENCY	16.368 MHz
	FREQ. STABILITY	±0.5 ppm
	OUTPUT VOLTAGE	1.0 V pk-pk
MIXER MAX2769	GAIN	+45 dB
	NOISE FIGURE	10-14 dB
IF BANDPASS MAX2769	CENTER FREQ.	4 MHz
	3dB PASSBAND	2.5 – 4.2 MHz
	STOPBAND ATT.	49.5 dB at 4 MHz
PGA/AGC MAX2769	GAIN	+20 dB to +70 dB
	NOISE FIGURE	1.5 dB Typ.

**Table 27: L1 front end components**

Component	Parameter	Value
L2 BANDPASS NSVS1108	CENTER FREQ.	1227.6 MHz
	1dB PASSBAND	Typ. 20 MHz
	PASSBAND VARIATION	1.0 dB
	INSERTION LOSS	Typ. 2.8 dB
VCO ADF4360-7	FREQ. RANGE	350 – 1800 MHz
	SUPPLY	26.5 mA @ 3.3 Vdc
	DIGITAL INTERFACE	SPI
MIXER MCA1-24+	LOSS	Typ. 6.1 dB
	LO-RF ISOLATION	40 dB
	LO-IF ISOLATION	25 dB

**Table 28: L2 front end components**

## 2.2.2 Digital Signal Processing

The FPGA logic incorporates the GNSS core, which is in charge of GNSS IF signal processing, and the system peripherals that are required for the receiver operations, as shown in Figure 21.

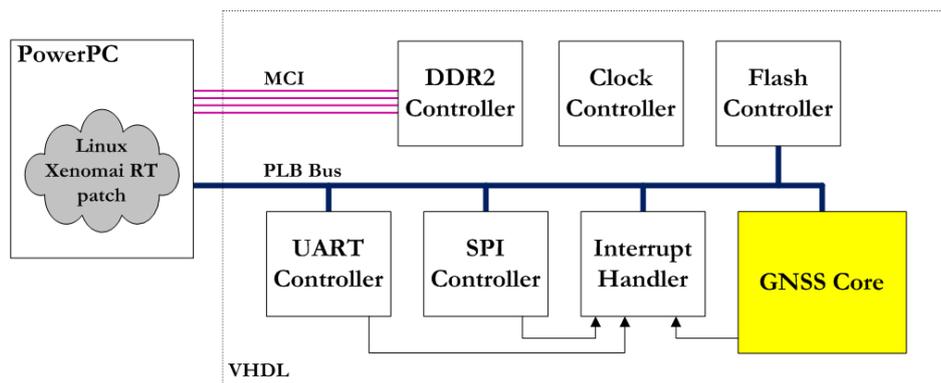


Figure 21: Receiver logic diagram

Except for the GNSS core, which has been coded in VHDL as part of the project, the other peripherals are part of Xilinx Ips and they have been assembled in a macro project using the editors provided by the FPGA manufacturer.

Name	Bus No.	IP Type	IP Version	IP Classification
ppc440_0		ppc440_virtex5	1.01.a	Processor
fcv_v20_0		fcv_v20	1.00.a	FCB Bus
plb_v46_0		plb_v46	1.04.a	PLBV46 Bus
DDR2_SDRAM_16Mx32		ppc440mc_ddr2	2.00.c	Memory Controller
xps_bram_if_cntlr_1		xps_bram_if_cntlr	1.00.b	Memory Controller
FLASH_8Mx16		xps_mch_emc	2.00.a	Memory Controller
xps_bram_if_cntlr_1_bram		bram_block	1.00.a	Memory
apu_fpu_virtex5_0		apu_fpu_virtex5	1.01.a	Peripheral
gnss_core_0		gnss_core	1.00.a	Peripheral
jtagppc_cntlr_0		jtagppc_cntlr	2.01.c	Peripheral
proc_sys_reset_0		proc_sys_reset	2.00.a	Peripheral
LEDs_8Bit		xps_gpio	1.00.a	Peripheral
Push_Buttons_18Bit		xps_gpio	1.00.a	Peripheral
xps_intc_0		xps_intc	1.00.a	Interrupt Controller
xps_spi_0		xps_spi	2.01.b	Peripheral
xps_timebase_wdt_1		xps_timebase_wdt	1.01.a	Peripheral
xps_timer_1		xps_timer	1.01.b	Peripheral
RS232		xps_uart16550	2.01.a	Peripheral
RS232_2		xps_uart16550	2.01.a	Peripheral
clock_generator_1		clock_generator	3.01.a	IP
input_bufg_0		input_bufg	1.00.a	IP
input_bufg_gnss_clk		input_bufg	1.00.a	IP
input_bufg_gnss_i0		input_bufg	1.00.a	IP
input_bufg_gnss_i1		input_bufg	1.00.a	IP
FLASH_8Mx16_util_bus_split_0		util_bus_split	1.00.a	IP

Figure 22: Logic design peripherals

The actual design is more complex than the diagram displayed in Figure 21, since it includes clock managing, JTAG interface and other units. The complete peripheral list is displayed in Figure 22. The GNSS core is accessible from the PowerPC through the PLB data bus, while its signals are collected using the interrupt handler, which translates a number of interrupts coming from the different units into a single interrupt for the PowerPC processor. The interrupt handler driver is then in charge of interrogating the unit in order to distinguish between the interrupt sources. The GNSS core is synchronized with the general purpose processor using two separate interrupt lines, which provide pulse per second precise information and trigger the dump of new data coming from the GNSS processing. The GNSS core implements a 14 channels correlator with I and Q branches, and equally spaced early-prompt-late accumulators: the basic units of each channel are displayed in Figure 23.

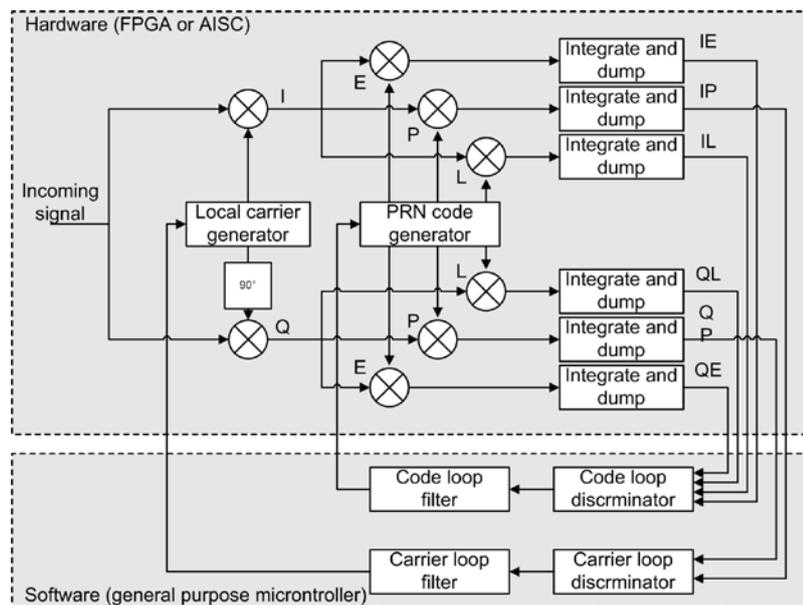


Figure 23: Logic correlator

The incoming signal, the 2 bit in-phase digital stream at the front-end output, is first numerically mixed with a local generated carrier replica at the Intermediate Frequency IF, both the in-phase I and quadrature channel Q. Then the result is correlated with a local spreading code: three spreading code versions are

generated by the logic, distanced of a fraction of the chip length. The output of the correlation is used by the software in order to control the code and carrier NCO. More details on the process of signal acquisition, tracking and demodulations using the described correlator are given in section 3, which provides also information on how the tracking of the GNSS signal is translated into GNSS observables (the carrier phase and pseudorange). The signal processing section of the core is clocked at 16.368MHz, which is the sampling frequency of the digital 4.092MHz IF GNSS signal at the front-end output and the 30-bit code and carrier NCO provides frequency control with accuracies of 0.071138 Hz/unit and 0.035569 Hz/unit respectively. The first version of the spreading code generator, which is used for architecture validation, includes only the LFSRs for the GPS C/A signal generation. The target code generator unit contains the LFSRs for the C/A code generation and primary and secondary memory allocations for the Galileo B+C code generation. The result of the VHDL coding have been validated using post-routing simulation, which provides the details on internal signals and bus values. The early, prompt and late samples of codes for a given PRN number are displayed in Figure 24, as result of simulation. Similar validation has been performed for all the entities that compose the correlator channel, namely the code and carrier NCOs, the I and Q binary carrier generator, the carrier mixer (Figure 25), the accumulators and counters, and the read/write register that are used to configure and get data from the correlator channel. The GNSS core and the channels are independently controlled using a series of 32-bit and 16-bit register, as is usually implemented in most commercial Ics. The Table 29 summarizes the control registers providing a description of their functions; the registers are accessible from the software application running on the PowerPC, and are used to configure the core and get data from GNSS signal processing.



Registry	Bit	RW / R	Function
PROG_MEAS_INT	32	RW	24-bit counter upper limit, to be used to set the measurements interrupt period.
MEAS_STATUS	32	RW	Indicates if new measures are available from code and phase counters.
PROG_ACCUM_INT	32	RW	24-bit counter upper limit, to be used to set the accumulation interrupt period.
ACCUM_STATUS	32	RW	Indicates if new accumulated data is available from the channels
SYSTEM_SETUP	32	RW	Interrupt enabling, unit reset and general setup.
CH <sub>x</sub> _CONTROL	32	RW	Channel enabling, reset and general setup.
CH <sub>x</sub> _CARR_DCO	32	RW	30-bit DCO value to set local carrier frequency.
CH <sub>x</sub> _CODE_DCO	32	RW	30-bit DCO value to set local code frequency.
CH <sub>x</sub> _CARR_PHASE_COUNTER	32	RW	Fractional phase counter, reports the integral of carrier Doppler
CH <sub>x</sub> _CODE_PHASE_COUNTER	32	RW	Counts the number of locally generated chips, integer and fractional.
CH <sub>x</sub> _I_EARLY	16	R	Integrated In-Phase Early
CH <sub>x</sub> _Q_EARLY	16	R	Integrated Quadrature Early
CH <sub>x</sub> _I_PROMPT	16	R	Integrated In-Phase Prompt
CH <sub>x</sub> _Q_PROMPT	16	R	Integrated Quadrature Prompt
CH <sub>x</sub> _I_LATE	16	R	Integrated In-Phase Late
CH <sub>x</sub> _Q_LATE	16	R	Integrated Quadrature Late
CH <sub>x</sub> _EPOCH	16	R	Counts the number of epochs at 1.5 seconds interval and resets every week.
CH <sub>x</sub> _EPOCH_CHECK	16	R	Instantaneous vale of epoch counter

**Table 29: GNSS core read /write registers (CH<sub>x</sub> refers to channel number x, therefore there are CH1 to CH14)**

### 2.2.3 Software Architecture

Following the software defined radio paradigm, most of the signal processing and the navigation solution computation are performed by software algorithms, while only the signal correlation and integration is implemented with the logic described in the previous section. The results of the signal correlation are processed by a software application in order to close the loop and control the code and carrier NCO, and the same software application extracts the GNSS observables from the GNSS core counters and computes the navigation solution, i.e. the LEO spacecraft position, velocity and on-board clock offset respect to the GPS time. The application software is executed under the control of an embedded Linux OS modified with the Xenomai real time patch, which ensures minimum latency between the front-end IC and the application drivers. The Linux OS provides the benefit of a solid infrastructure to implement multi thread application, and to make use of the drivers such as RS232, timers and SPI that are already part of the operating system, with the drawback of a much more complex driver implementation with respect to an embedded stand-alone application.

The first issue when loading to operating system to the target board, is to make the software aware of the underlying hardware, providing information about the physical memory allocations of the peripherals and the bus structure where this peripheral are attached. This task is performed by loading at boot time a device tree generated using Xilinx tools: the tree is a text file that enumerates and link all the peripherals and provides specific information about each device. Then a set of drivers which are part of the operating system, named *Open Firmware Platform Drivers* or *OF Platform Drivers*, are in charge of device tree interpretation and information extraction. With the help of this driver, the operating system is able to identify the starting point of the physical addresses bank that are used to control and get data from the GNSS core.

The GNSS core driver for the Linux OS has been implemented as part of the receiver prototyping, and it provides an interface between the core and GNSS

application software that is executed in the user space. The Linux OS inhibits the direct access to the hardware from the user space, therefore the core register and interrupts lines are not directly available from high level software. In order to overcome this limitation, and keeping at the same time the simplicity of application development in the user space, the GNSS core device driver has been implemented and its structure is displayed in Figure 26.

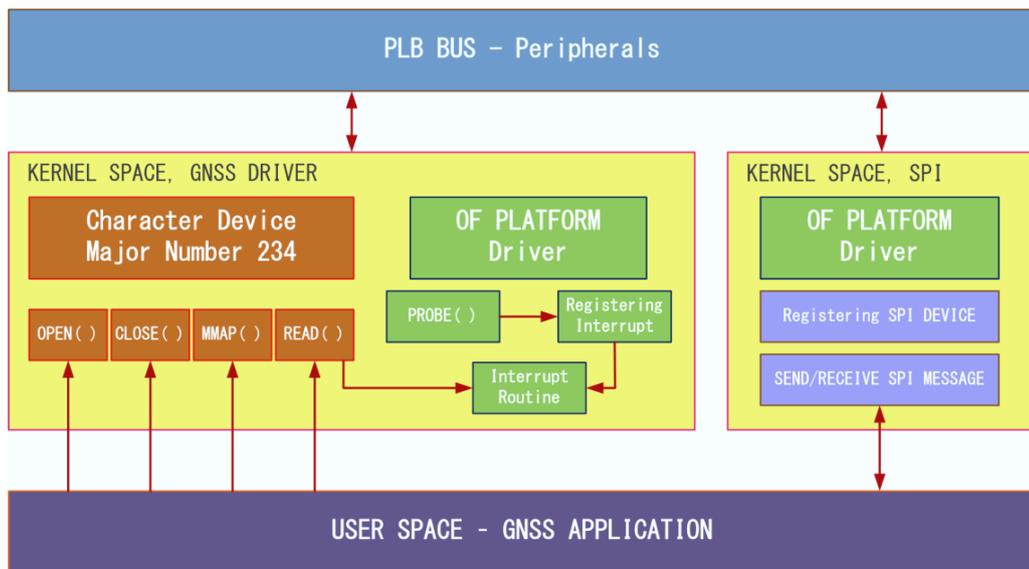


Figure 26: GNSS application device driver structure

The interface between the GNSS application and the peripherals is composed of two sections: the SPI driver is used to configure the front-end and other external components that are compatible with this interface, while the GNSS driver is used to configure and access the core that process the GNSS signal. As displayed in Figure 26, the GNSS driver is built from two Linux infrastructures: the character device driver, which provides to the user space the classic calls `OPEN()`, `CLOSE()`, `MMAP()` and `READ()`, and the OF Platform Driver which includes the `PROBE()` call that is used to extract relevant information from the device tree and to register the interrupt requests and related routines. The two fundamentals calls that let the user space application access the GNSS core registers are the `MMAP()` and the `READ()`. The `MMAP()` is used to remap the register into user space memory, having from that point on direct access to the control registers, while the

READ() function is a blocking call that is released at the interrupt occurrence. This driver trick is used to overcome the inability of the Linux OS to map interrupt routines to their interrupt sources directly from user space.

The GNSS user space application is a multi-thread software based on POSIX threads API, patched with Xenomai real-time OS layer. The software organization in parallel threads is displayed in Figure 27.

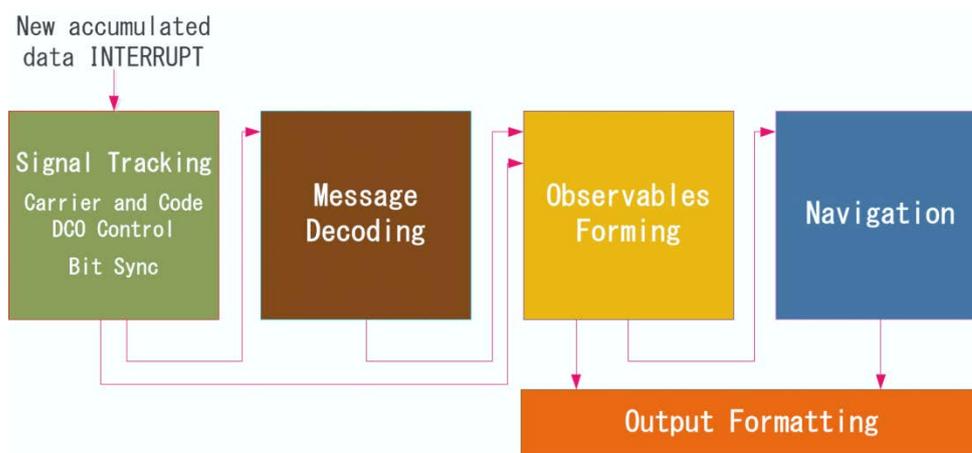


Figure 27: GNSS application software structure

The signal tracking and bit sync algorithm is triggered and executed at the new accumulated data interrupts: at first the new data coming the correlators are read by the application, which then closes the tracking loop feedback by processing the dumped information. More details on tracking algorithms are provided in section 3. The bit stream that is demodulated as part of signal tracking, is then passed to the message decoding thread, which search for frames preamble and then extracts relevant content from GNSS modulated data. The decoded information together with counters values that are read by the interrupt routine are used by a separate thread to form the GNSS observables, the pseudorange and carrier phase, the signal Doppler and an evaluation of  $C_{no}$ . The operations described since here are performed for every receiver channel. Finally the observables are processed by the navigation engine that provides user position, velocity and clock offset respect to the GPS time. More details on the navigation algorithm are provided in section 4.

## 3 Signal Acquisition and Tracking

### 3.1 Introduction

GNSS space vehicles provide Code Division Multiple Access (CDMA) services through the synchronous broadcasting of Direct Sequence Spread Spectrum (DSSS) signals in the L band. Each signal is composed of a unique PRN spreading code (or a time multiplexed combination of codes, or again a superimposition of a PRN sequence and a binary subcarrier) with good correlation/cross-correlation properties, and a navigation message, which contains necessary information to compute the navigation solution (space vehicles position, clock, code disambiguation time tags and other service information), which are phase modulated on the selected carrier. The signal at the receiver antenna can be modelled as

$$r_{RF}(t) = \sum_{i=1}^L y_{RF,i}(t) + n_{RF}(t)$$

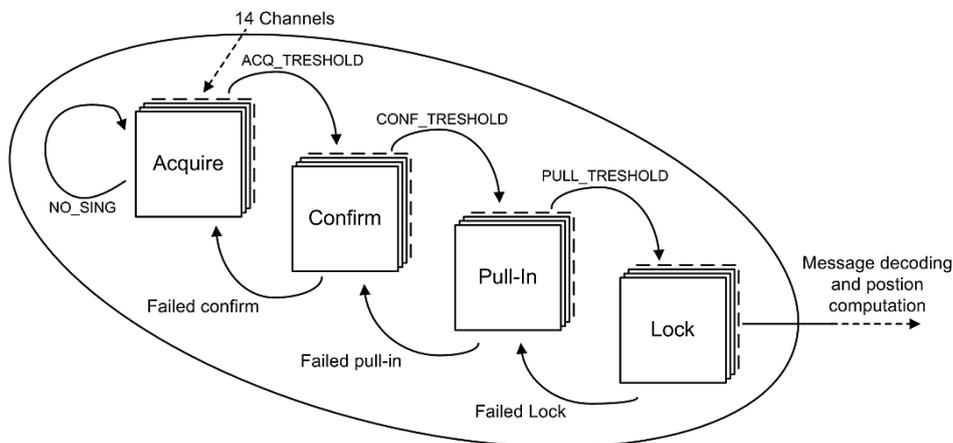
$$y_{RF,i}(t) = A_i c_i(t - \tau_i) d_i(t - \tau_i) \cos(2\pi(f + df_i)t + \varphi_i)$$

where  $L$  is the number of signal that are in view at the receiver antenna,  $n_{RF}$  is the noise,  $A$  is the signal amplitude,  $c$  is the spreading sequence which is assumed to take values of  $-1$   $+1$ ,  $d$  is the message data bit,  $df$  is Doppler effect affecting signal  $I$  and  $\varphi$  is a random phase. The specific characteristics of the spreading codes and carrier frequencies for the different systems have been already reported in section 2.1.1. The noise term is assumed to be Additive White Gaussian Noise with power spectral density equal to  $No/2$ , while each received signal is characterized by the power  $C=A^2/2$ . In terms of signal power, the GPS signal architecture guarantees approximately  $-130\text{dBm}$  of power at a  $0$  gain antenna, while the thermal noise power density is  $-173.97\text{dBm/Hz}$ . For a  $2\text{MHz}$  noise bandwidth (the null-to-null bandwidth of the GPS L1 C/A signal) at a medium temperature of  $17^\circ\text{C}$  the noise power is  $-110.97\text{dBm}$ , meaning that the power of GPS signal is well below the

noise level. As already mentioned in previous section, the receiver front end is in charge of signal filtering, down conversion to an *Intermediate Frequency IF*, quantization and digital conversion, translating the analogue input into a numerical sequence, which can be modelled as

$$r[n] = y[n] + n[n] = Ac(n - \tau_0)d(n - \tau_0) \cos(2\pi(f_{IF} + df)n + \varphi_0) + n[n]$$

where, in order to simplify, we assume that the front end bandwidth only affects the noise, providing a noise variance equal to  $NoB_{IF}$  when the selected sampling frequency is twice the front-end bandwidth, while leaving unchanged the spreading code pulse shape. The synchronization of the described CDMA signal is a two-step process, which exploits the correlation/cross correlation properties of the received signal by correlating the incoming sequence with a locally generated replica, through the correlator processing unit that has been described in section 2.2.2. The first step is the signal acquisition: the signal coming from a specific space vehicle is searched, spanning the carrier Doppler frequency and the spreading code phase with a sequence of discrete cell tests.



**Figure 28: Acquisition and tracking state machine**

If the acquisition algorithm gives positive answer, indicating the presence of the signal and providing also a rough estimation of the Doppler and code phase, then the signal handling is passed to the tracking algorithm, which is in charge of fine signal alignment, i.e. fine tuning of locally generated carrier phase and code phase

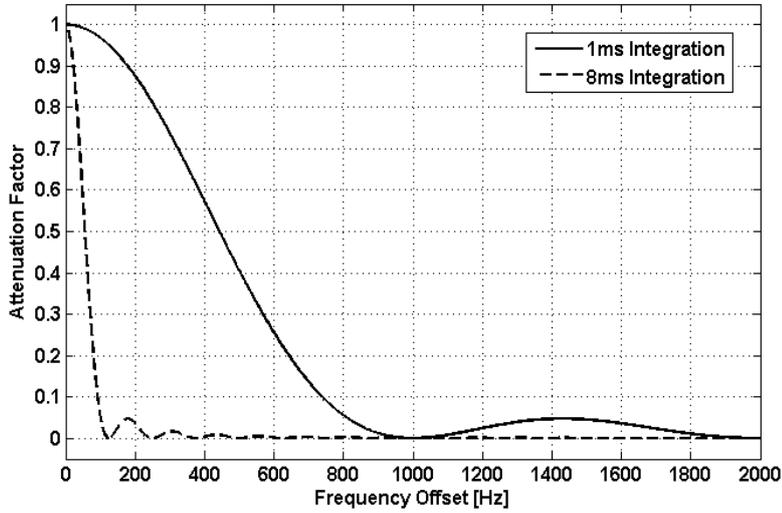
to match the incoming signal, and finally of message bit demodulation. The implemented synchronization and tracking algorithm is more complex than the described two-step process, but it is based on the same principles. The actual algorithm is implemented as a state machine, displayed in Figure 28, with the purpose of providing more robustness against time varying channel conditions. The content of each state is described in the following sections.

### **3.2 Acquisition and MofN algorithm description**

The GNSS signal acquisition is a search process, which requires replication of both the spreading code and the carrier. For the GPS case for example, the L1 C/A code is a sequence of 1023 chips, therefore 1023 phases have to be searched during acquisition which is generally performed at 0.5 chip increments. For the GALILEO case, the E1-B code is longer than GPS, precisely 4092 chips. For the second search dimension, the carrier Doppler, the span has been defined in section 2.1.5 for the space receiver case and is approximately  $\pm 45\text{kHz}$  around the center frequency. The maximum Doppler search step depends on the selected correlation integration period: this is because the offset that is present between the real carrier and the Doppler hypothesis produces an attenuation of the correlation peak which depends on the integration period. This attenuation is given by [5]:

$$\alpha(\delta F) = \frac{\sin^2(\pi N \delta F)}{(\pi N \delta F)^2}$$

where  $\delta F$  is the Doppler offset and  $N$  is the number of integrated samples during the correlation. The attenuation is displayed in Figure 29 for the cases of 1ms signal integration and 8ms signal integration (the actual number of samples depends on the sampling frequency which is generally 3 or 4 times the IF).



**Figure 29: Correlation peak attenuation due to Doppler offset during Doppler discrete search**

Longer integration periods are generally required when acquiring in weak signal conditions in order to increase the probability of detection, which is the case of ground application in shadowed environment, for example urban canyons or light indoors. However, it has been shown in previous sections that the space application of GNSS receiver for orbit determination does not include this signal conditions, and therefore 1ms integration period with 500Hz Doppler search step can be configured for the search process, for both the GPS and GALILEO signals. According to the correlator diagram provided in section 2.2.2, during the search phase the prompt  $I$  and  $Q$  samples are integrated and their envelope (the sum  $I^2+Q^2$ ) is evaluated against a defined threshold  $\beta$ : the presence of signal is revealed for the selected test cell if the envelope is higher than the threshold, otherwise the hypothesis is rejected. The probability that the envelope is greater than the selected threshold when the signal is present is called *detection probability*, while the probability that the envelope is greater than the selected threshold when the signal is absent is called *false alarm probability*. They can be expressed as

$$P_d(\beta) = P(S > \beta | H_1)$$

$$P_{fa}(\beta) = P(S > \beta | H_0)$$

where  $H_1$  represent the hypothesis of a signal correctly aligned while  $H_0$  represents the absence of the signal.  $S$  is the result of correlation with locally generated carrier and code replica, it is the sum of  $I^2$  and  $Q^2$ , and can be expressed as [8]:

$$S(\tau, df) = |Y(\tau, df)|^2 = \left| \frac{1}{N} \sum_{n=0}^{N-1} r[n]c[n-\tau] \exp\{-j2\pi(f+df)n\} \right|^2$$

where  $N$  is the number of integrated samples and the local carrier,  $I$  and  $Q$  components, has been expressed with complex exponential notation. When no signal is present,  $S$  is the square of the norm of a complex Gaussian random variable, or the sum of squares of Gaussian random variables. It can be derived [8] that the variance of the imaginary and real part of  $Y$  is given by

$$\text{Var}[\text{Re}\{Y(\tau, df)\}] = \text{Var}[\text{Im}\{Y(\tau, df)\}] = \sigma_n^2 = \frac{\sigma_{IF}^2}{2N}$$

and that their expected value is zero. The sum of their squares, which gives  $S(\tau, df)$ , has a probability density function which follows a Rayleigh distribution (or equivalently a  $\chi^2$  distribution with 2 degrees of freedom and composing random processes with zero mean) [4], and it is given by (it can be proved using basic properties of Gaussian random variables)

$$f_0(s) = \frac{s}{\sigma_n^2} \exp\left\{-\frac{s^2}{2\sigma_n^2}\right\}$$

Therefore the false alarm probability is given by

$$P_{fa}(\beta) = \int_{\beta}^{+\infty} f_0(s) ds = \exp\left\{-\frac{\beta^2}{2\sigma_n^2}\right\}$$

The threshold  $\beta$  that has to be configured in the tracking algorithm, can be identified starting from the selected false alarm probability. It depends also from the noise variance at the output of the correlators: the actual numerical value is

related to ADC bit coding and the derived digital representation of the ADC values (for example ADCs could provide 2-bit or 4-bit sampling, changing the numerical representation of the same analogue signal at the ADC inputs), and therefore is generally measured trying to acquire a signal which is known to be absent. In order to define a preliminary quantity, we can also identify the noise variance with the following simplified consideration: the digital output of the front-end produces 2-bit a digital stream, and the quantization values corresponds to levels +3, +1, -1 and -3. The automatic gain control adjust the input gain so that the percentage of values approximately follows the rules:

01	+3	15%
00	+1	35%
10	-1	35%
11	-3	15%

The digital stream is mixed with locally generated 2-bit carrier replica, which is encoded with 8 phases per cycle as:

$$\begin{aligned}
 \textit{In-Phase:} & \quad 11 \ 10 \ 00 \ 01 \ 01 \ 00 \ 10 \ 11 & \quad (+2 \ +1 \ -1 \ -2 \ -2 \ -1 \ +1 \ +2) \\
 \textit{Quadrature:} & \quad 00 \ 01 \ 01 \ 00 \ 10 \ 11 \ 11 \ 10 & \quad (-1 \ -2 \ -2 \ -1 \ +1 \ +2 \ +2 \ +1)
 \end{aligned}$$

The carrier mixer produces and output with 3-bit for the magnitude, representing the values 1, 2, 3 and 6 plus 1-bit for the sign. The values are the accumulated by the integrator. When the input is +3, then the mixer sequence is

$$+3 \times \textit{In-Phase:} \quad +6 \ +3 \ -3 \ -6 \ -6 \ -3 \ +3 \ +6$$

which has zero mean and a variance of 22.5, while when the input is +1 the variance is 2.5. Then, according to the percentage assigned by the automatic gain control, we have when no signal is present

$$\textit{Var}[Re\{Y(\tau, df)\}] = \textit{Var}[Im\{Y(\tau, df)\}] = (22.5 \times 0.3) + (2.5 \times 0.7) = 8,5$$

The compute value can be used as a first post-correlation noise variance guess to defined the acquisition threshold, having selected the probability of false alarm. Considering the non-critical employment of the GNSS receiver and its received

signals, multiple false alarms do not raise any specific problem. Therefore the probability of false alarm can be selected to be quite high, between 1% and 5%, assuring also to not to discard any signal when it's present, because of too stringent threshold. According to the properties of the Rayleigh distribution, the mean value of  $S$  as estimated with the previous simplified assumptions, is approximately 10,7 (because of low values, the integrated samples are multiplied by a constants, in order to have enough digits when representing the number with an integer). When the signal is present, the variance of  $Y$  does not change, because the amount of noise coming from the input signal remains the same, but the expected value increases because of a non-zero correlation. It can be shown that the expected value, in case of perfect alignment of the code phase and carrier Doppler is [8]:

$$E[Re\{Y(\tau, df)\}] = \frac{A}{2} \cos\varphi_0$$

$$E[Im\{Y(\tau, df)\}] = \frac{A}{2} \sin\varphi_0$$

Then the probability density function follows a Rice distribution (or equivalently a  $X^2$  distribution with two degrees of freedom and composing random processes with non-zero mean [4]), which can be expressed as:

$$f_1(s) = \frac{s}{\sigma_n^2} \exp\left\{-\frac{s^2 + A^2}{2\sigma_n^2}\right\} I_0\left(\frac{sA}{\sigma_n^2}\right)$$

where  $A$  is the signal amplitude and  $I_0$  is the modified Bessel function of first kind and zero order. The probability of detection is then expressed by the integration of  $f_1(s)$ , which is given by the Marcum Q-function of first kind

$$P_d(\beta) = Q_1\left(\sqrt{\frac{A^2}{\sigma_n^2}}, \sqrt{\frac{\beta^2}{\sigma_n^2}}\right)$$

If we define the pre-detection signal to noise ratio as

$$SNR = \frac{E^2[\text{Re}\{Y(\tau, df)\}]}{\text{Var}[\text{Re}\{Y(\tau, df)\}]} = \frac{A^2}{4} \frac{1}{\sigma_n^2} = \frac{A^2}{4} \frac{2N}{\sigma_{IF}^2} = \frac{NC}{N_0 f_2/2} = 2 \frac{C}{N_0} T_C$$

where  $T_C$  is the integration period, and also we define a normalized threshold as

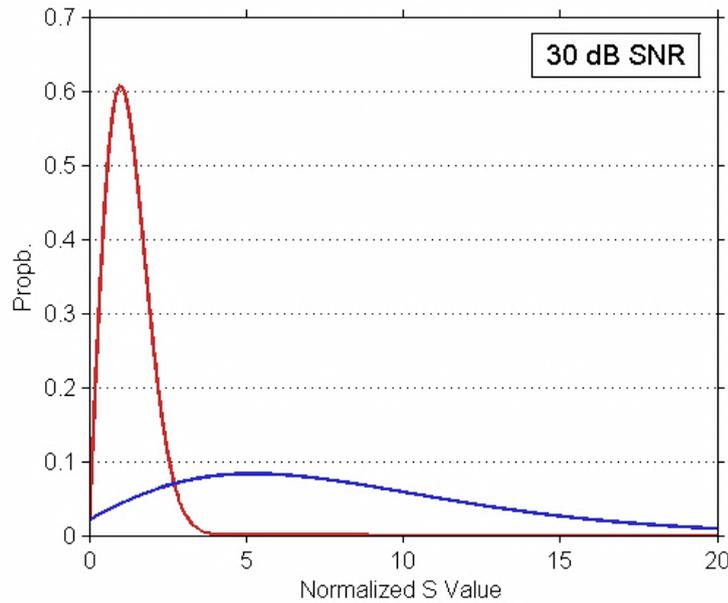
$$\beta' = \frac{\beta^2}{\sigma_n^2}$$

then the probabilities of detection and false alarm becomes

$$P_{fa}(\beta') = \exp\left\{-\frac{\beta'}{2}\right\}$$

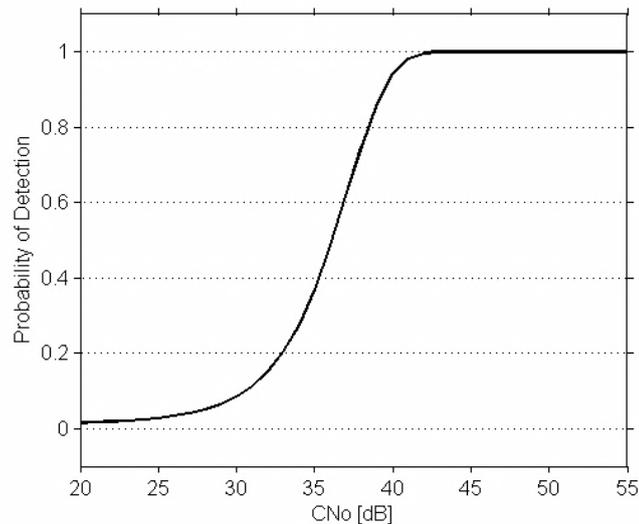
$$P_d(\beta) = Q_1(\sqrt{4SNR}, \sqrt{\beta'})$$

The equations shows that the probability of detection depends only on the pre-detection  $SNR$ , and therefore on the signal to noise density ratio, having fixed a normalized acquisition threshold.



**Figure 30: Probability of detection (blue) and probability of false alarm (red) density functions, for a given Cno of 30dB**

The Figure 30 shows the mentioned probability density functions, for a given  $C_{no}$  of 30 dB, while the Figure 31 shows the probability of detection with a fixed acquisition thresholds that gives 1% of probability of false alarm, as function of the input signal  $C_{no}$ .

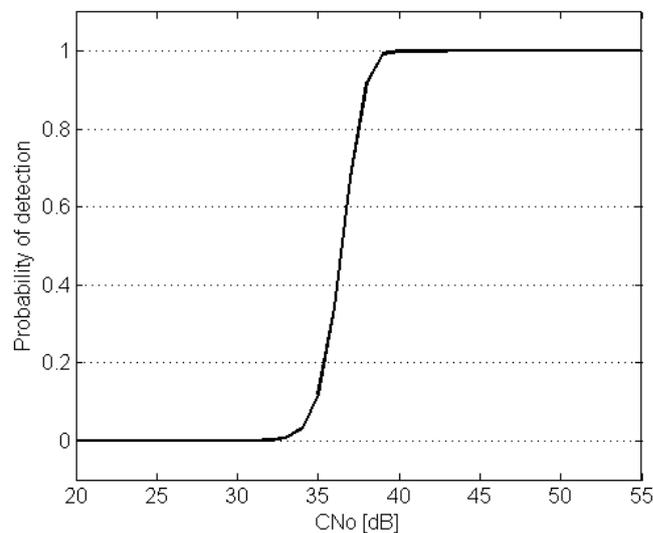


**Figure 31: Probability of detection for a fixed threshold giving 1% of probability of false alarm, as function of input  $C_{no}$**

According to the state machine presented in Figure 28, the positive answer from a code phase and Doppler hypothesis analysis triggers a confirmation algorithm, which is mainly used to reduce the propagation of false alarms through the processing chains. The confirmation is given by an M of N algorithm: N hypothesis trials are performed keeping the Doppler and code phase constant, and the presence of signal is confirmed only if a minimum of M tests are successful, i.e. the value of correlation  $S$  is higher than the threshold for M times. The successive trials can be modelled as Bernoulli trials, where the number of successful tests is given by a binomial distribution. The overall probability of false alarm is then the sum of the probability of all successful tests, given the absence of signal:

$$P_{FA} = \sum_{n=M}^N \binom{N}{n} P_{fa}^n (1 - P_{fa})^{N-n}$$

For a given single trial  $P_{fa}$  equal to 1%, the overall confirmation probability given 10 trials and requesting 6 successes, drops to about  $10^{-3}$ , while the probability of detection as function of Cno is displayed in Figure 32.



**Figure 32: Probability of detection of 6 successes out of 10 trials in presence of a signal at the given Cno, for a single trial probability of false alarm of 1%**

Compared to Figure 31, the probability of detection shows a sharper edge with increasing Cno: this is because at low Cno, the low probability of single trial detection reduces the amounts of successful tests below the requested value of 6. The “6 of 10” are the numbers of successes and trials selected for the software implementation of the confirmation algorithm: no formal analysis has been performed to optimize these values, but they have been chosen according to what suggested in specialized literature [4][5][8] and basing on practical experience. When the confirmation algorithm provides positive answer, the signal handling is passed to the tracking algorithm which is in charge of fine code and carrier phase

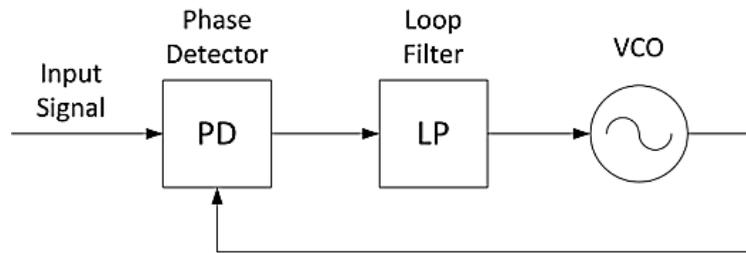
alignment; the theory and practical application of signal tracking are described in the following sections.

### **3.3 Tracking algorithm**

The tracking algorithm is in charge of refining the Doppler and code phase rough estimation provided during acquisition as result of a discrete search process, and keep the local replica aligned with the incoming signal in terms of carrier and code phase. Two algorithms works together in order to achieve this task: one is dedicated to code phase and is generally named *Delay Locked Loop DLL*. The second is dedicated to the carrier and can be in form of *Frequency Locked Loop FLL* or *Phase Locked Loop PLL*, or more often a combination of them. Regardless of the nature or the purpose of the tracking loops implemented in the GNSS receiver, a common principle is at the base of their architecture: an error is generated to represent “misalignment” between the incoming signal phase and its local replica, which is then processed by a feedback structure that finally controls the local replica *Numerically Controlled Oscillator NCO* (named also *Voltage Controlled Oscillator VCO* or *Digitally Controlled Oscillator DCO* according to the implementation). The following section summarizes the general principles of phase tracking, which are valid for DLL, PLL or FLL, while section 3.3.2 provides practical details on what is actually implemented in the GNSS receiver, and how the theoretical models are translated into the operative algorithm.

#### **3.3.1 Principles of phase tracking**

The basic elements of a Phase Locked Loop are displayed in Figure 33 [9]. It consists of three units: (I) a discriminator or phase detector, (II) a loop filter and (III) a Voltage Controlled Oscillator VCO.



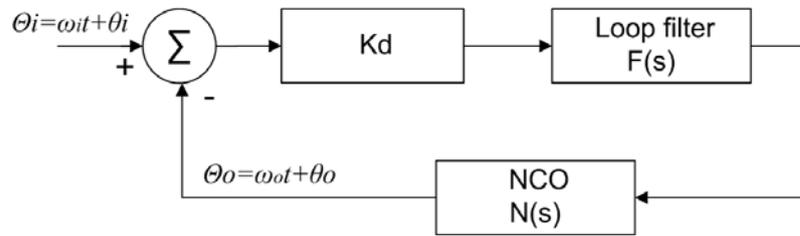
**Figure 33: Basic Phaselock Loop**

The phase detector compares the periodic input signal with the output of the VCO, providing a measure of the phase error between the two. The error is then filtered in order to reduce the tracking noise, and the resulting error signal is applied to the VCO in order to control its output frequency. When the loop is *locked*, the average frequency of the VCO equals the average frequency of the input, but phase lock does not imply zero-phase error. Steady state error and fluctuating phase error are generally both present, and if their sum is excessive, they can cause loss of lock. The salient properties of the loop are:

- *Bandwidth*, that represents the capability of the loop to process the information contained in the input signal. For simple loop implementations the bandwidth is fixed and can be mathematically expressed and linked to loop parameters, while more complex algorithms imply a variable bandwidth that adapts itself to the input signal conditions.
- *Linearity*. Regardless the algorithm complexity, every phase locked loop is nonlinear. The mathematical instruments to deal with this nonlinearity are too complex for practical implementation, and can provide usable results only under strong simplifications. Fortunately, most locked loop of interest can be analysed using linear techniques, when they are in lock conditions. The linear analysis is also considered sufficient to derive design guidelines of most PLLs. Several important mechanism related to loop nonlinearity have to be taken into account, during loop design and analysis.

In order to proceed with a theoretical analysis, which is aimed to produce a mathematical relation between the loop parameters and the loop behaviour during

signal tracking, two simplifications are required: the first is to neglect the effect of the non-linearity, or in other words to assume that the loop is working “near” the lock state. The second simplification is to assume that the loop works in continuous time domain, while in reality it processes a discrete stream of data. The implications of the passage between continuous time domain and discrete time domain will be further analysed in the section 3.3.2, dedicated to the implementation aspects. Under this assumption the loop model can be displayed with Figure 34.



**Figure 34: Phaselock Loop Linear Model**

The phase detector, which is generally the source of loop non linearity, has been replaced with the gain  $K_d$ , while the other elements are represented by their continuous time domain linear transfer functions. With these elements, the closed loop transfer function can be expressed as

$$H(s) = \frac{K_d F(s) N(s)}{1 + K_d F(s) N(s)}$$

The NCO transfer function is generally given by

$$N(s) = \frac{K_o}{s}$$

when analysing the PLL or by a pure gain  $K_o$  when dealing with the FLL, while the loop filter transfer function can be selected in order to give to the loop the desired properties. The most common filter transfer function employed in GNSS receivers and reported on the related literature [4], gives second order type two Phase Locked Loop. For the case of GNSS receiver for space application, third order loops has also been investigated, and their design procedure developed as

part of the receiver implementation. The employed loop filter transfer functions are provided in Table 30, while the sections 3.3.1.1 and 3.3.1.2 focuses on second order and third order loops characteristics and design parameters.

Order	Type	Filter
PLL II	II	$F(s) = \frac{\tau_2 s + 1}{\tau_1 s}$
PLL III	III	$F(s) = \frac{(\tau_2 s + 1)^2}{(\tau_1 s)^2}$
FLL II	II	$F(s) = \frac{\tau_2 s + 1}{(\tau_1 s)^2}$

**Table 30: Loop filter transfer functions**

### 3.3.1.1 Second order Phase Locked Loops

According to what has been described in the previous section, the loop filter that gives second order PLL is

$$F(s) = \frac{\tau_2 s + 1}{\tau_1 s}$$

and then the closed loop transfer function is then given by

$$H(s) = \frac{\frac{KdKo}{\tau_1} (\tau_2 s + 1)}{s^2 + \frac{KdKo}{\tau_1} \tau_2 s + \frac{KdKo}{\tau_1}}$$

We now define two parameters, natural radian frequency  $\omega_n$  and damping actor  $\zeta$ , as follows:

$$\omega_n = \sqrt{\frac{KdKo}{\tau_1}}$$

$$\zeta = \frac{\omega_n \tau_2}{2}$$

The natural frequency is related to the velocity of the loop: the higher it is the faster the loop will be in responding to input changes. But at same time the higher the natural frequency is, the higher the output phase will overcome the input signal while trying to following it. On the opposite an high damping ratio means that the output will be slowed while trying to follow the input reference. With this parameter definition, then

$$H(s) = \frac{2\omega_n\zeta s + \omega_n^2}{s^2 + 2\omega_n\zeta s + \omega_n^2}$$

which is the most common representation of the second order tracking loop transfer function that is reported in literature [4]. From the definition of natural radian frequency and damping factor, we can derive the first important design parameter, the loop bandwidth, which provides a measure on the capability of the loop to pass or stop the information carried at its input. It is important to notice that the bandwidth refers to the phase error information extracted by the correlation processing, and not the GNSS signal itself. Generally speaking, there are various definitions of a filter bandwidth, each one of them underline a particular aspect of the filter. We now focus on two of those definitions: 3dB bandwidth and *Noise Equivalent Bandwidth, NEB*. The exact definitions of the 3dB bandwidth index is: “*the radian frequency at which the amplitude of the harmonic transfer function of the loop is  $1/\sqrt{2}$  times the amplitude of the same function at frequency equal to 0*”. The harmonic transfer can be easily derived from the Laplace transfer function reported in the section. It is a function of frequency  $f$ , and it represents the response of the loop when the input is a sinusoid signal of frequency  $f$ . The  $1/\sqrt{2}$  ratio used in the definition corresponds to a 3dB attenuation of the output. This bandwidth tells us “how much signal” enters the loop: suppose for example that the 3dB bandwidth of our PLL is 1HZ while the phase of the received signal is oscillating, due to the relative motion of the receiver to the satellite, with a frequency of 2Hz. In this case our PLL is not able to follow the changing phase because the 2Hz changing doesn’t enter our 1Hz

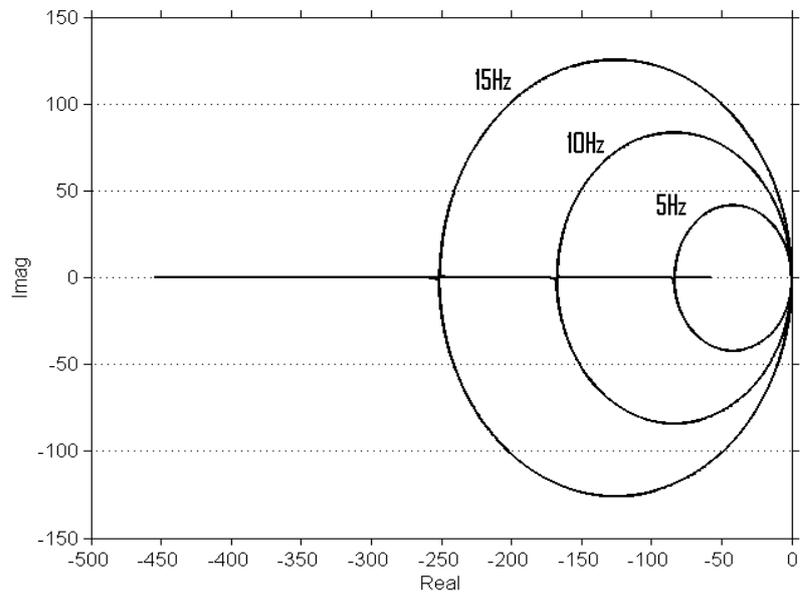
3dB bandwidth PLL. So this index is somehow related to the capability of intercepting fast significant changes in the information carried by the input signal. Regardless of the mathematics involved, the equation of the 3dB bandwidth for a second order tracking loop is

$$B_{3dB} = \omega_n \sqrt{1 + 2\zeta^2 + \sqrt{2 + 4\zeta^2 + 4\zeta^4}}$$

The definition of Noise Equivalent Bandwidth NEB is “*the width of a fictitious rectangular filter such that the power in that rectangular band is equal to the actual power of the output signal in an infinite band*”. This index tells how much wideband noise is passed by the filter, since the definition consider all the spectrum of the output processed signal, not only limited to a 3dB attenuation. For this reason NEB is usually greater than the 3dB bandwidth. The equation of NEB is given by

$$B_{NEB} = \omega_n \frac{\zeta + 1/4\zeta}{2}$$

From practical point of view, generally only one of the bandwidth is used to describe the loop architecture. They are both reported in this text because both of them can be found in specialized literature.

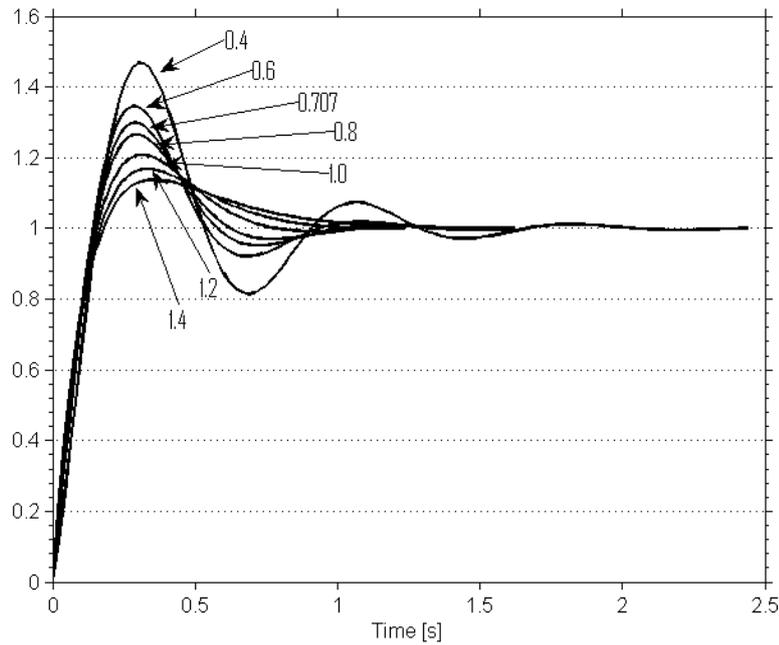


**Figure 35: Second order loop root locus**

Generally speaking, the damping factor and bandwidth have to be selected in order to assure loop stability and the proper stability margin. For the specific case on second order phase locked loop, it can be easily shown that it is unconditionally stable regardless of the actual selected values, providing that both the damping factor and the natural radian frequency are positive values. As a confirmation, the Figure 35 displays the root locus of the open loop transfer function, with fixed Noise Equivalent Bandwidth equal to 5Hz, 10Hz and 15Hz, and varying damping factor, showing that the real part of the roots are non-positive for every value of the feedback gain.

The second order loop step impulse response can be computed in closed form, providing also analytical expression for some of the response parameters such as maximum overshoot and peak time. The response is given by

$$\theta_{STEP}(t) = 1 + \frac{1}{\sqrt{1 - \zeta^2}} e^{-\omega_n \zeta t} \sin(\omega t - \Psi)$$



**Figure 36: Second order loop step impulse response**

with  $\omega = \omega_n \sqrt{1 - \zeta^2}$  which is the radian frequency of the response and  $\Psi = \tan^{-1}((\sqrt{1 - \zeta^2})/\zeta)$ . The step response is displayed in Figure 36 for Noise Equivalent Bandwidth equal to 1Hz and varying damping factor. The last element to consider when selecting the loop parameters, is the steady state error. The steady state error is defined as the difference between the loop input and the loop output when the transient period shown in Figure 36 is over, so when the settling time is passed. Three different standard loop inputs are used to find three corresponding steady state errors, so the behavior of the loop is completely described if we assume that a generic input can be a combination of these three standard inputs. These three standard inputs are (the time domain function and the Laplace transform are given below):

- phase position

$$\theta_i(t) = Cp \quad t \geq 0 \qquad \theta_i(s) = \frac{Cp}{s}$$

- phase ramp (frequency position)

$$\theta i(t) = Cvt \quad t \geq 0 \quad \theta i(s) = \frac{Cv}{s^2}$$

- phase acceleration (frequency ramp)

$$\theta i(t) = Cat^2 \quad t \geq 0 \quad \theta i(s) = \frac{2Ca}{s^3}$$

To find the mathematical equation for the steady state errors with the three inputs, it is convenient to use another form of loop transfer function, which put the error in evidence. If we consider the error phase  $\theta_e$  as the output of the loop, and the incoming signal phase  $\theta_i$  as its input then the transfer function will be

$$G(s) = \frac{\theta e(s)}{\theta i(s)} = \frac{\theta i(s) - \theta o(s)}{\theta i(s)} = 1 - \frac{\theta o(s)}{\theta i(s)}$$

which can be written in terms of natural radian frequency and damping ratio

$$G(s) = \frac{s^2}{s^2 + 2\omega_n \zeta s + \omega_n^2}$$

To compute the steady state error using the transfer function, the final value theorem is used, which can be expressed as follows: assume to have a generic transfer function  $G(s)$  with an input function  $i(s)$ . The steady state output in the time domain  $o(t)$  will be, if it exists,

$$o_{STEADY} = \lim_{t \rightarrow +\infty} o(t) = \lim_{s \rightarrow 0} sG(s)i(s)$$

Applying this, we can now compute the steady state error for our second order transfer function in the three input cases:

- phase position:

$$\lim_{s \rightarrow 0} sG(s) \frac{Cp}{s} = \lim_{s \rightarrow 0} \frac{Cps^2}{s^2 + 2\omega_n \zeta s + \omega_n^2} = 0$$

- phase ramp (frequency position)

$$\lim_{s \rightarrow 0} sG(s) \frac{Cv}{s^2} = \lim_{s \rightarrow 0} \frac{Cvs^2}{s(s^2 + 2\omega_n \zeta s + \omega_n^2)} = 0$$

- phase acceleration (frequency ramp)

$$\lim_{s \rightarrow 0} sG(s) \frac{2Ca}{s^3} = \lim_{s \rightarrow 0} \frac{2Cas^2}{s^2(s^2 + 2\omega_n \zeta s + \omega_n^2)} = \frac{2Ca}{\omega_n^2}$$

We found that the phase position error and the phase ramp error are always zero, while the phase acceleration is a constant. This is always true for a transfer function expressed as the selected PLL, which in control theory is classified as second order, type two. Second order refers to the highest degree of the polynomial expression  $1 + KdF(s)N(s)$ , which is the transfer function denominator, while the type refers to the number of poles located at zero in the polynomial expression  $F(s)N(s)$ . If we use a different filter in our tracking loop, the steady state errors change according to the following table.

	Step	Ramp	Acc.
II Order Type 1	0	constant	ramp
II Order Type 2	0	0	constant
III Order Type 3	0	0	0

**Table 31: Steady state errors for various transfer functions**

### 3.3.1.2 Third order Phase Locked Loops

Giving the peculiarities of the receiver space application and the associated high dynamical stresses, and considering the steady state errors determined in the previous section, third order type three loop is selected for the PLL

implementation in order to null the error associated with phase acceleration. The implemented analogue filter transfer function is given in the following equation.

$$F(s) = \frac{(\tau_2 s + 1)^2}{(\tau_1 s)^2}$$

Unfortunately the loop parameters selection that comes from linear approximation is not as simple as the second order case, since the concepts of natural radian frequency and damping factor are lost in the third order transfer function. The design shall rely on classical control theory instruments such as open loop Bode plots and root locus, and find approximations in order to define a set of meaningful design parameters. The open loop transfer functions and related phase are given by

$$G(s) = \frac{KdKo}{s} \left( \frac{\tau_2 s + 1}{\tau_1 s} \right)^2$$
$$\angle G(s) = \frac{-3\pi}{2} + 2 \tan^{-1}(\omega_c \tau_2)$$

The unit gain cross over frequency  $\omega_c$ , displayed in open loop function bode plot of Figure 37, can be selected as an indicator of the loop bandwidth, and once it is fixed according to the application requirements, the parameter  $\tau_2$  can be determined by assigning a positive phase margin to the equation, thus assuring also the loop stability.

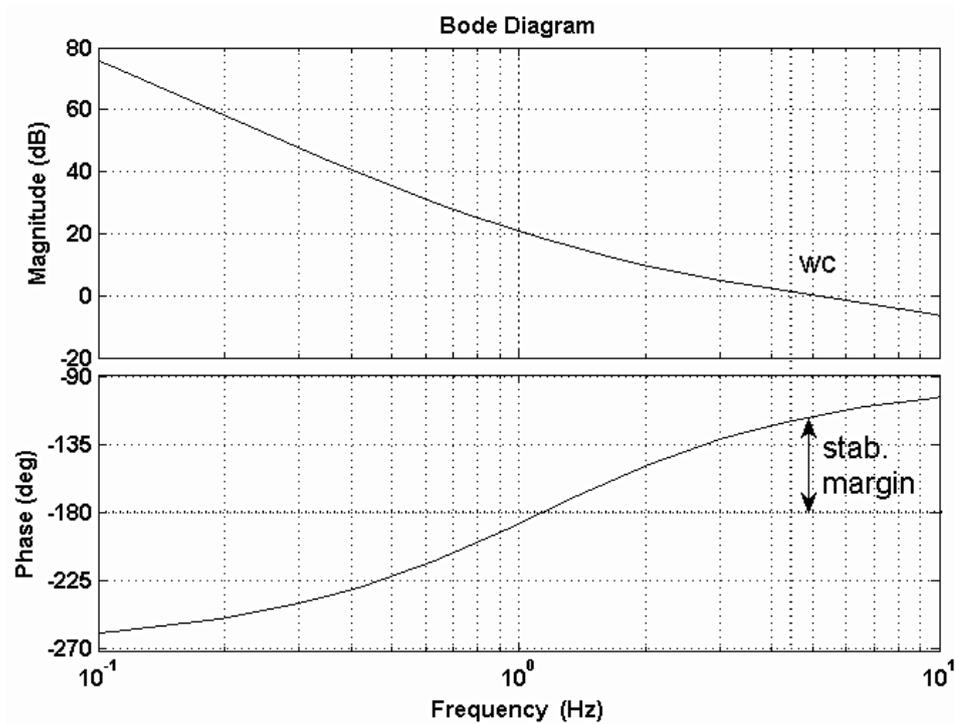


Figure 37: Third Order Loop Bode diagram, with cross over frequency equal to 4.5Hz

If the loop is in the stable region, i.e. the unit gain cross frequency is larger than the inverse of  $\tau_2$ , it can be approximated by a second order loop obtaining the following equation.

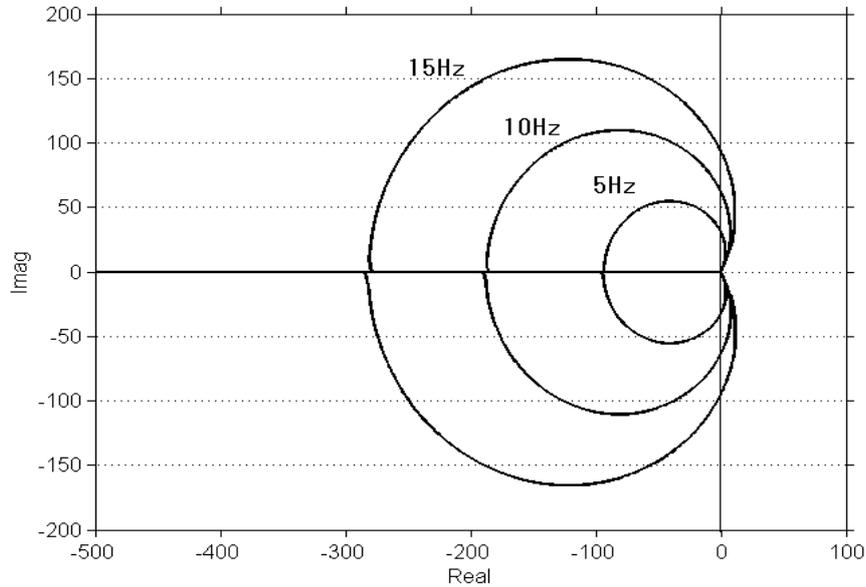
$$\omega_c \cong KdKo \left( \frac{\tau_2}{\tau_1} \right)^2$$

We can therefore compute the parameter  $\tau_1$ , and define a relation to a fictitious damping factor  $\zeta$ , expressed by the equation

$$\tau_2^{-1} = \omega_c \left[ \sqrt{16\zeta^4 + 1} - 4\zeta^2 \right]$$

The correct selection of the damping factor requires a trial and error procedure, where the phase margin is first fixed, then the damping factor is computed through the approximate equation, and again the phase margin is modified if the resulting damping factor does not fit the requirements. As already stated in the section,

third order loop is conditionally stable given a set of design parameters, as shown also in the root locus of Figure 38.



**Figure 38: Third order loop root locus**

The Figure 38 displayed the root locus for three fixed values of unity gain cross over frequency with varying stability margin: it is displayed that for a certain set of parameters the real part of poles is positive, thus providing loop instability. It is therefore necessary, once the parameters have been selected, to check for the loop stability in order to guarantee the correct loop behaviour.

### **3.3.2 PLL and DLL implementation aspects**

In the previous sections the theoretical background of the signal tracking has been presented, with details on the loop filter selection, design parameters and general features. This section provides details on the implementation aspects, on how the continuous time domain transfer functions are translated into operative equations, and which set of parameters has been selected for the loops. The procedure used to generate the observables from the tracking information is also presented at the end of the section.

The filter transfer functions have to be software implemented, since they are in charge of processing the correlation output coming from the GNSS core and their output is used to drive the carrier and code NCOs. In order to make the continuous time domain functions implementable, they have to be translated into discrete time domain, or equivalently, they have to be expressed in form of  $z$ -transform which is the discrete time domain equivalent of the Laplace transform. Once the Laplace transform is given as in our case, there is not a unique way to obtain the corresponding  $z$ -transform, because of two reasons: the first is that different approximations of the continuous time domain can be selected giving different transformations (this aspect will be clarified shortly), and the second is that a number of unit delays is generally required for the discrete time domain representation since in a discrete process the information at any instant  $t$  can only be used to define the system behavior at instant  $t+1$ . The simplest method to obtain the  $z$ -transform for the Laplace representation is to use the triangular transformation expressed as

$$s = \frac{1 - z^{-1}}{\Delta T}$$

where  $z^{-1}$  represents a one-step delay in the discrete stream. Alternatively the bilinear or Tustin transformation can be used:

$$s = \frac{2}{\Delta T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

These different forms are related to a different way to approximate the first order linear non-homogeneous differential equation. We can see that two different approximations give two different relations between Laplace and  $z$ -transform:

$$\frac{dy}{dt} = x \quad \rightarrow \quad \frac{y_i - y_{i-1}}{\Delta T} = x_i \quad \rightarrow \quad s = \frac{1 - z^{-1}}{\Delta T}$$

$$\frac{dy}{dt} = x \quad \rightarrow \quad \frac{y_i - y_{i-1}}{\Delta T} = \frac{x_i + x_{i-1}}{2} \quad \rightarrow \quad s = \frac{2}{\Delta T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

We can now focus on the second order loop filter Laplace representation:

$$F(s) = \frac{\tau_2 s + 1}{\tau_1 s} = \frac{\tau_2}{\tau_1} + \frac{1}{\tau_1 s}$$

It is clear that the filter itself is the sum of a proportional part and an integration part represented with  $1/s$ . If we now apply the bilinear transformation, we obtain

$$F(z) = \frac{\tau_2}{\tau_1} + \frac{1}{\tau_1 \frac{2}{\Delta T} \frac{1-z^{-1}}{1+z^{-1}}}$$

where  $\Delta T$  is the sampling period. We can manage the equation, relating the input of the filter  $x$  to its output  $y$ , to obtain

$$y_i - y_{i-1} = \frac{\tau_2}{\tau_1} (x_i - x_{i-1}) + \frac{\Delta T}{2} \frac{1}{\tau_1} (x_i + x_{i-1})$$

$$y_i = y_{i-1} + \frac{\tau_2}{\tau_1} (x_i - x_{i-1}) + \frac{\Delta T}{2} \frac{1}{\tau_1} (x_i + x_{i-1})$$

where  $y_i$  is the new NCO value coming from the processing of the correlation value  $x$ . The proportional and integral part can be clearly identified in the discrete filter form, but still the given equation does not represent what is actually implemented in the GNSS software. The problem is that the dumped information at time instant  $I$  cannot be used to set the NCO at the same time instant, but at future time instant  $i+1$ . Or in other terms, the present value of the NCO can be set using only past information, therefore unit delays have to be inserted both in the proportional and integral part to make the equation implementable. With the delays, the equation becomes:

$$y_i = y_{i-1} + \frac{\tau_2}{\tau_1} (x_{i-1} - x_{i-2}) + \frac{\Delta T}{2} \frac{1}{\tau_1} (x_{i-1} + x_{i-2})$$

which now satisfies also the non-concurrent time requirement from input to output. If we apply the same transformation for the third order loop filter and

other loop filters that can be employed in the GNSS tracking, we obtain the implementable equations that are summarized in Table 32. The Table 33 provides an overview of the relevant loop parameters from continuous time domain analysis.

	Filter	Discrete time domain implementable filter equation
PLL II	$F(s) = \frac{\tau_2 s + 1}{\tau_1 s}$	$y_i = y_{i-1} + \frac{\tau_2}{\tau_1} (x_{i-1} - x_{i-2}) + \frac{T}{2\tau_1} (x_{i-1} + x_{i-2})$
PLL III	$F(s) = \frac{(\tau_2 s + 1)^2}{(\tau_1 s)^2}$	$y_i = 2y_{i-1} - y_{i-2} + \frac{\tau_2^2}{\tau_1^2} (x_{i-1} - 2x_{i-2} + x_{i-3}) + \frac{T\tau_2}{\tau_1^2} (x_{i-1} - x_{i-3}) + \frac{T^2}{4\tau_1^2} (x_{i-1} + 2x_{i-2} + x_{i-3})$
FLL II	$F(s) = \frac{\tau_2 s + 1}{(\tau_1 s)^2}$	$y_i = 2y_{i-1} - y_{i-2} + \frac{T\tau_2}{2\tau_1} (x_{i-1} - x_{i-3}) + \frac{T^2}{4\tau_1} (x_{i-1} + 2x_{i-2} + x_{i-3})$

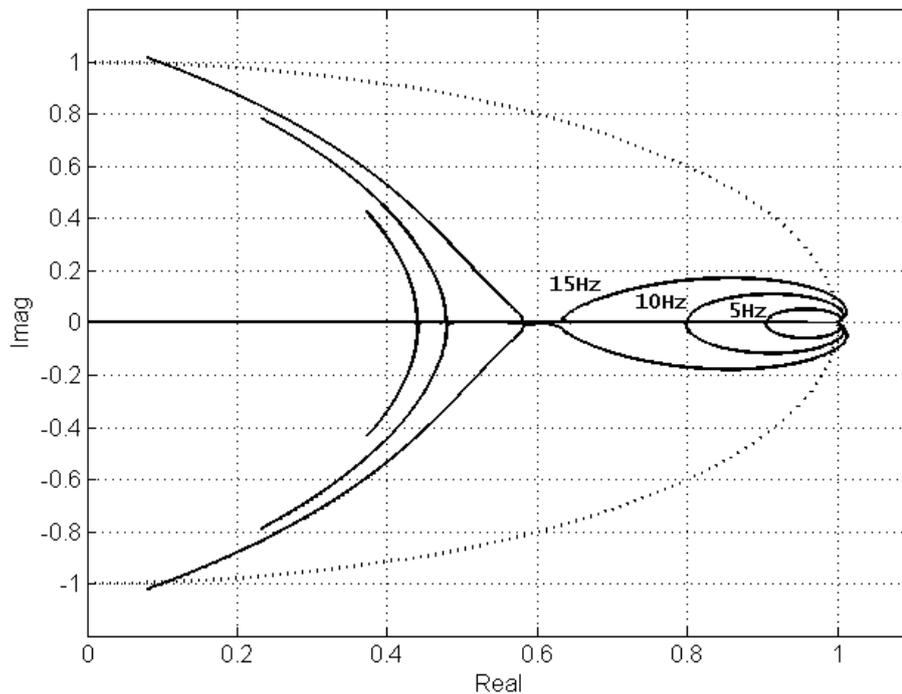
**Table 32: Basic loop filter implementable equations**

Order	Relevant Parameters	
II	Natural Radian Frequency	$\omega_n = \sqrt{\frac{KdKo}{\tau_1}}$
	Damping factor	$\varepsilon = \frac{\omega_n \tau_2}{2}$
	Equivalent Noise Bandwidth	$B = \omega_n \frac{\varepsilon + 1/4\varepsilon}{2}$

III	Unity Gain Cross Over Frequency	$\omega_c \cong KdKo \left(\frac{\tau_2}{\tau_1}\right)^2$
	Phase Margin	$\varphi \cong -\frac{3\pi}{2} + 2\tan^{-1}(\omega_c\tau_2)$

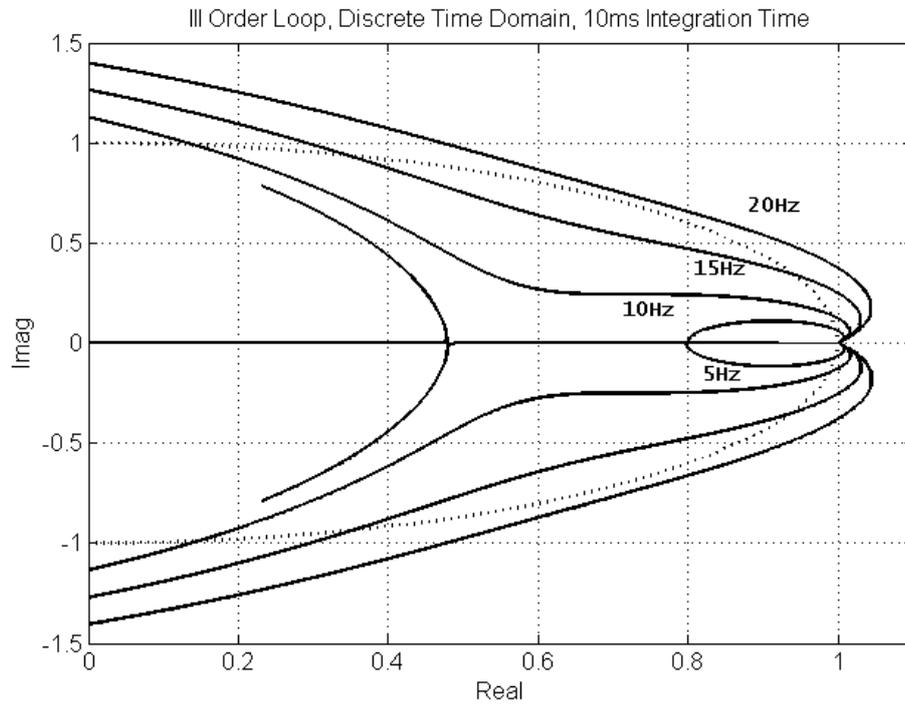
**Table 33: Basic loop filter relevant design parameters**

Since the filter equations have been changed, from continuous time domain to discrete time domain and by including the unit delays, the new form should be checked for stability, while we can simplify the parameters selection by still getting support from the continuous time domain theory. In order to check the stability, the complete loop open loop transfer function has to be translated into discrete time domain, and then its root locus evaluated. The Figure 39 displays the root locus for the third order loop transfer function: it is shown that the translation to discrete time domain introduces new instability regions in the loop, recalling that for the z-transform, the closed loop is stable if the poles of the open loop transfer function are placed inside the unity circle centred at the origin.



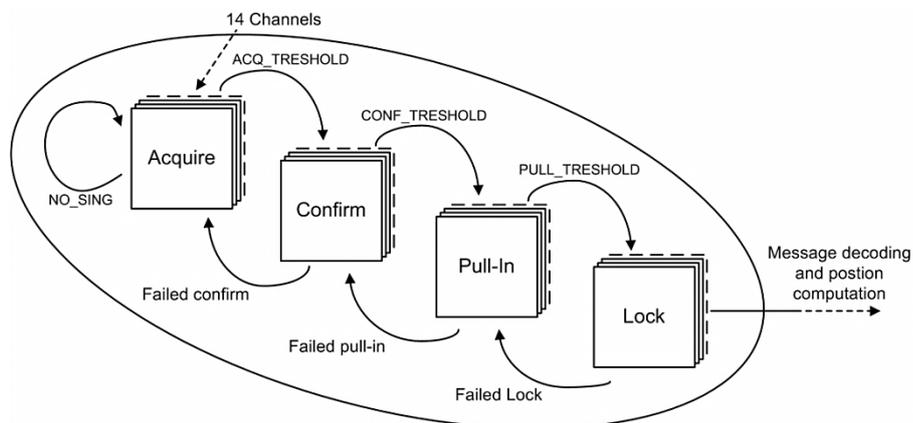
**Figure 39: Third order loop discrete time domain root locus, 1ms integration**

The same behaviour can be observed for the second order loop, which was unconditionally stable in continuous time domain. What can also be observed is that the stability also depends on the  $\Delta T$  parameters, the time distance of two consecutive discrete data processed by the loop. The Figure 39 was related to a loop working on a integration period of 1ms, therefore the correlation output was a stream of data distanced by 1ms. If the integration period is increased, for example to 10ms, then root locus changes according to Figure 40. It is shown for example that for a given loop unity gain cross over frequency of 20Hz, the loop is unconditionally unstable, while in other cases instability regions are enlarged with respect to the 1ms integration period case.



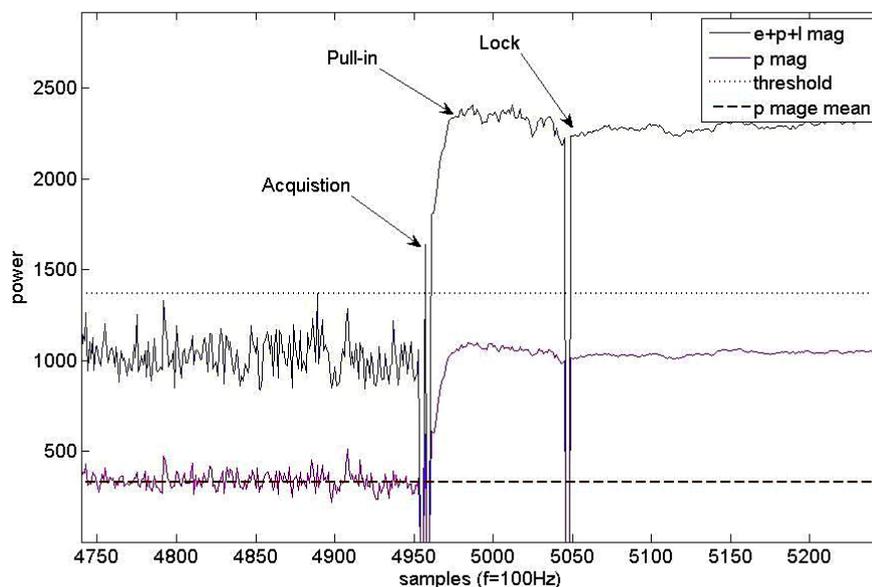
**Figure 40: Third order loop discrete time domain root locus, 10ms integration**

Having identified the implications of discrete time domain implementation of the loops, and the limits related to the parameter selection because of discrete loop stability, the parameters can be configured in order to guarantee the correct loop behaviour.



**Figure 41: Acquisition and tracking state machine**

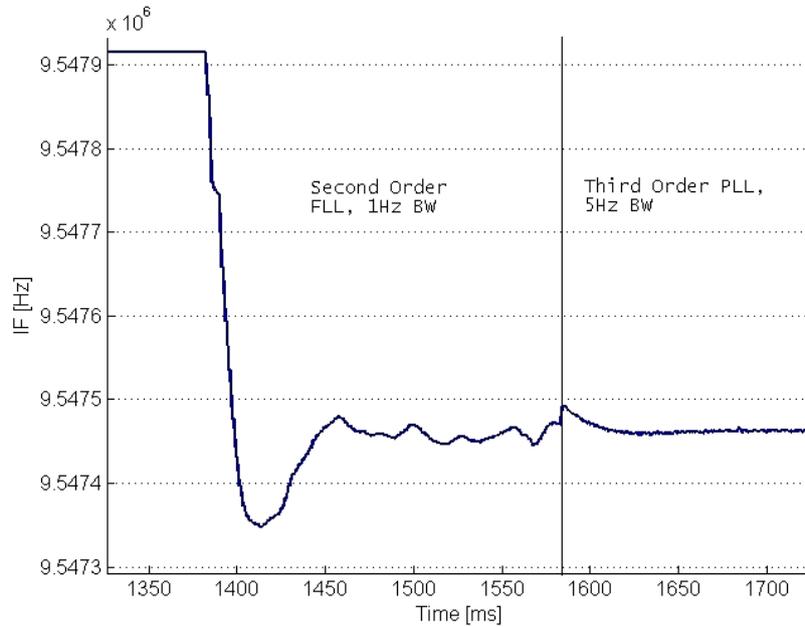
The software state machine displayed in Figure 41 tells that after signal confirmation, the signal handling is passed to the PULL-IN state and then to the LOCK state: both states are composed of FLL/PLL and DLL, but their bandwidths are configured so that the PULL-IN is faster while the LOCK state provide more accurate signal tracking: this ensures that at the beginning of tracking the phase errors are rapidly reduced using higher bandwidth, while tracking variance in LOCK state is reduced when the tracking error is small thanks to a lower bandwidth.



**Figure 42: Signal power from prompt correlator branch during signal acquisition, pull-in and lock.**

The concept is clarified in Figure 42, which displays the prompt signal power during signal acquisition, pull-in and lock. When stepping from pull-in to lock, the loop bandwidth is reduced, therefore providing higher signal tracking accuracy. Moreover the pull-in state carrier phase tracking algorithm is not a simple PLL, but a sequence of FLL followed by a PLL: the first loop reduces most of the frequency error at the beginning of tracking, when the frequency hypothesis coming from the acquisition and confirmation could contain an error up to 500Hz. Then, once the frequency is acquired, the PLL comes in place and

acquires the signal carrier phase. The concept is illustrated in Figure 43, where the sequence of FLL and PLL effect of frequency tracking is displayed.



**Figure 43: Carrier frequency tracking during pull-in state**

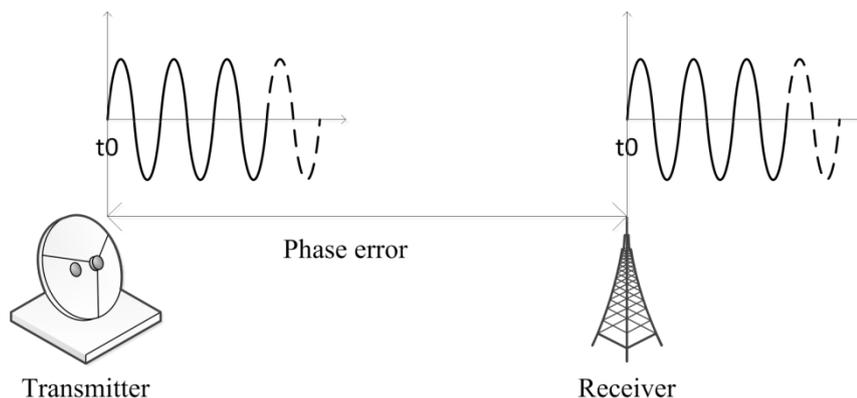
Having identified the overall structure of the tracking algorithms, the separation between PULL-IN and LOCK state, and the composition of FLL and PLL for the pull-in tracking of the carrier phase, the Table 34 summarizes the parameters selection for the different loops that have been implemented in the GNSS receiver using the discrete time equations reported in Table 30. The selection of parameters is mainly based on reports of specialized literature about the application of GNSS receiver in space [1][10], and on previous experience on FPGA based GNSS receivers [11].

State	Loop	Order	Bandwidth	Integration Time	Discriminator
PULL-IN	FLL	II	1 Hz	1 ms	ATAN based
	PLL	III	10 Hz	1 ms	ATAN based
	DLL	III	4 Hz	1 ms	E-L Power

LOCK	PLL	III	5 Hz	1 ms	ATAN based
	DLL	III	0.8 Hz	10 ms	E-L Power

**Table 34: Selected Loop Parameters**

Once the described tracking algorithms have locked the code and carrier phases, the GNSS observables can be generated from tracking information, and the GNSS message can be decoded in order to get the data that are necessary for user position, velocity and clock computation. The process of message bit decoding after code and carrier phase locking is straightforward, and is well described in specialized literature [4], therefore not described here. It is based on bit boundary detection by histogram evaluation, then on frame synchronization based on known preamble detection. What is generally not clearly detailed is how the GNSS chipsets translates the tracking information into the GNSS observables, the carrier phase and the pseudorange. The first of the two observables, the carrier phase, is simple to generate: it is also called *Accumulated Delta Range*, since it is basically the integral of the carrier Doppler measured in carrier cycles. The carrier phase is only capable of measuring relative movements between the transmitter and the receiver, but not absolute distance, because of the its ambiguous nature.



**Figure 44: Simplified transmitter and receiver model**

In order to understand this concept, we can think to simplified situation where the transmitter and the receiver are fixed in space, and shares the same ideal clock,

therefore there is no clock error between the two. At a certain time instant the receiver and transmitter starts to generate their perfectly synchronized carriers (the transmitter actually transmits this carrier, while the receiver only generates a local carrier). The situation is displayed in Figure 44. Because of the space distance of the two devices, the transmitted carrier is received with a phase offset with respect to locally generated carrier: this phase offset is the distance divided by the carrier wavelength (we assume that there are no other propagation delays), and composed by a number of integer cycles plus a fractional cycle. When the receiver align its local replica phase, it is only capable of compensating for the fractional cycle, since it cannot distinguish between integer cycles. What it does is to modify the frequency of the local carrier oscillator since the PLL is locked. If we integrate the correction value of the NCO with respect to nominal frequency, we would find the fractional phase that has been compensated to align the local replica to the incoming signal. This is the first measure of the carrier phase, a fractional cycle. From this point on, every time the receiver or transmitter move along the line of sight, generating a Doppler on the carrier frequency, the NCO compensates for this Doppler. By integrating its correction value we can have an estimate of the displacement. Summarizing, by integrating the NCO correction value (the Doppler) for example using an accumulation register, the carrier phase provides a measure of the mutual displacement between transmitter and receiver with respect to the initial condition, when the PLL has first locked.

The pseudorange forming, derived from code tracking, follows a similar principle but with the important difference that it is not ambiguous, but it's a measure of the distance between the receiver and transmitter. The implementation is also slightly different, since it is based on counters. The procedure for the GPS case is as follows (it is similar for other signals),

1. A free running counter, called  $Tr$ , is clocked with the nominal signal code rate, 1.023MHz for the GPS L1, and is reset at the beginning of

---

each GPS week. It provides a measure of local receiver time during the GPS week (which start every Saturday/Sunday transition).

2. Another counter, called  $T_i$ , is clock with the local code replica NCO, which also has a nominal value that gives 1.023MHz.
3. In our ideal condition, with the receiver and transmitter fixed in space and clocked with the same source, at a certain instant  $t$  both counters start to count. Since they are clocked with the same exact frequency, their values will also be perfectly aligned. At the same instant the transmitter also start signal transmission, while the receiver start its local code replica generation.
4. The code form the transmitter is received: because of the distance, the two code will not be aligned, but the received code will be in late with respect to its local replica, meaning that when we receives the first chip of the sequence, we are already generating the  $n$ -th chip of the sequence. Now the DLL comes in place: it slows down the local generated code replica, reducing the NCO value, practically waiting for the two codes to be aligned. When  $m$ -th chip received correspond to the  $m$ -th chip locally generated, the two codes are aligned and the DLL is in lock.
5. Since during the procedure the NCO value has been reduced to slow down the local replica, now the counter  $T_r$  is ahead of counter  $T_i$ , because  $T_r$  is constantly clocked at 1.023MHz while  $T_i$  is clocked with the NCO. The difference between  $T_r$  and  $T_i$  represents the flight time of the signal, so the distance between the receiver and transmitter, called pseudorange.

In the real case there is a first complication: the GPS L1 C/A sequence is approximately 300km long, meaning that the received signal form the GPS satellite is a sequence of identical codes that are 300km long. This means that even the code is ambiguous, because as happens for the carrier phase, we cannot distinguish between integer sequences, and the code is not long enough to

represent the receiver-satellite distance unambiguously (this distance is approximately 20000km). To solve this issue the modulate message comes in place: every frame of the message contains the transmission time of the first bit front of the next frame. Since each frame is long enough to cover the satellite-receiver distance unambiguously, when we receive the first bit of the frame we know exactly its transmission time (it was transmitted as part of the previous frame), therefore we can set the  $T_i$  counter with this value and from that point on it will represent the exact transmission time of the received signal. Then the pseudorange will be the difference between of  $T_r$  and  $T_i$ . Finally, since the receiver and transmitter clocks are not aligned, and various propagation effects delays the incoming codes, the counters will also include these errors, which can then be separated from the geometric observables during the navigation estimation process.

### **3.4 Expected tracking performances**

In order to evaluate the performances of the implemented tracking algorithms, the ideal test platform would have consisted of a signal generator which is capable of simulating the receiver trajectory and provided at its output the RF signal as it would be receiver at the output of the real antenna. Since this expensive instrumentation, was not available at the faculty laboratories, and alternative approach has been followed. A signal software simulator has been developed, in a semi-analytic framework, which provides the realistic environment where to test the tracking algorithms applied to the GNSS space application. After this, the algorithms have been tested on real signal, but received on ground, without the characteristics of the signal received in space. The following sections briefly describes this approaches and the main results, recalling that the main purpose is to characterize the tracking algorithms in terms of tracking noise respect to the input signal  $C_n$ .

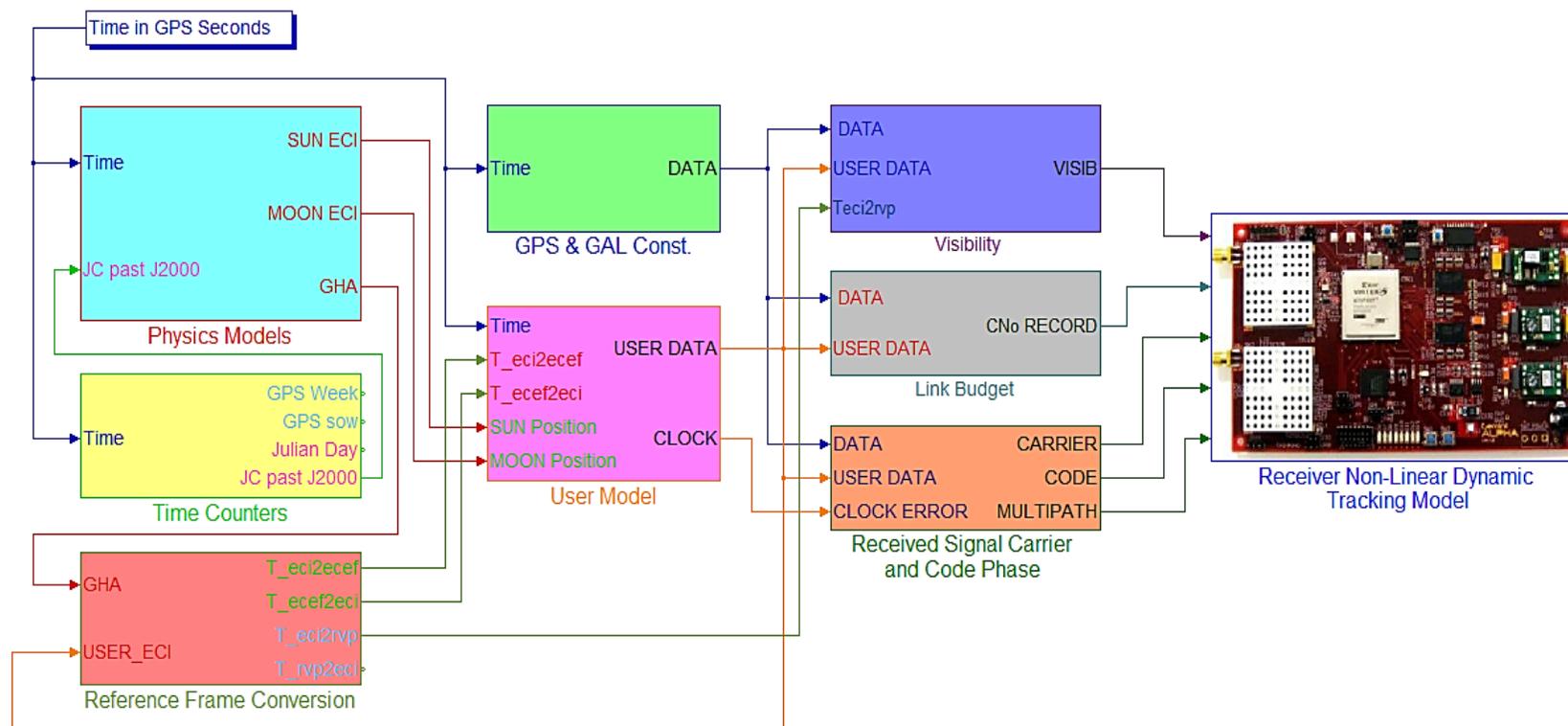


Figure 45: Semi analytic signal tracking simulator diagram

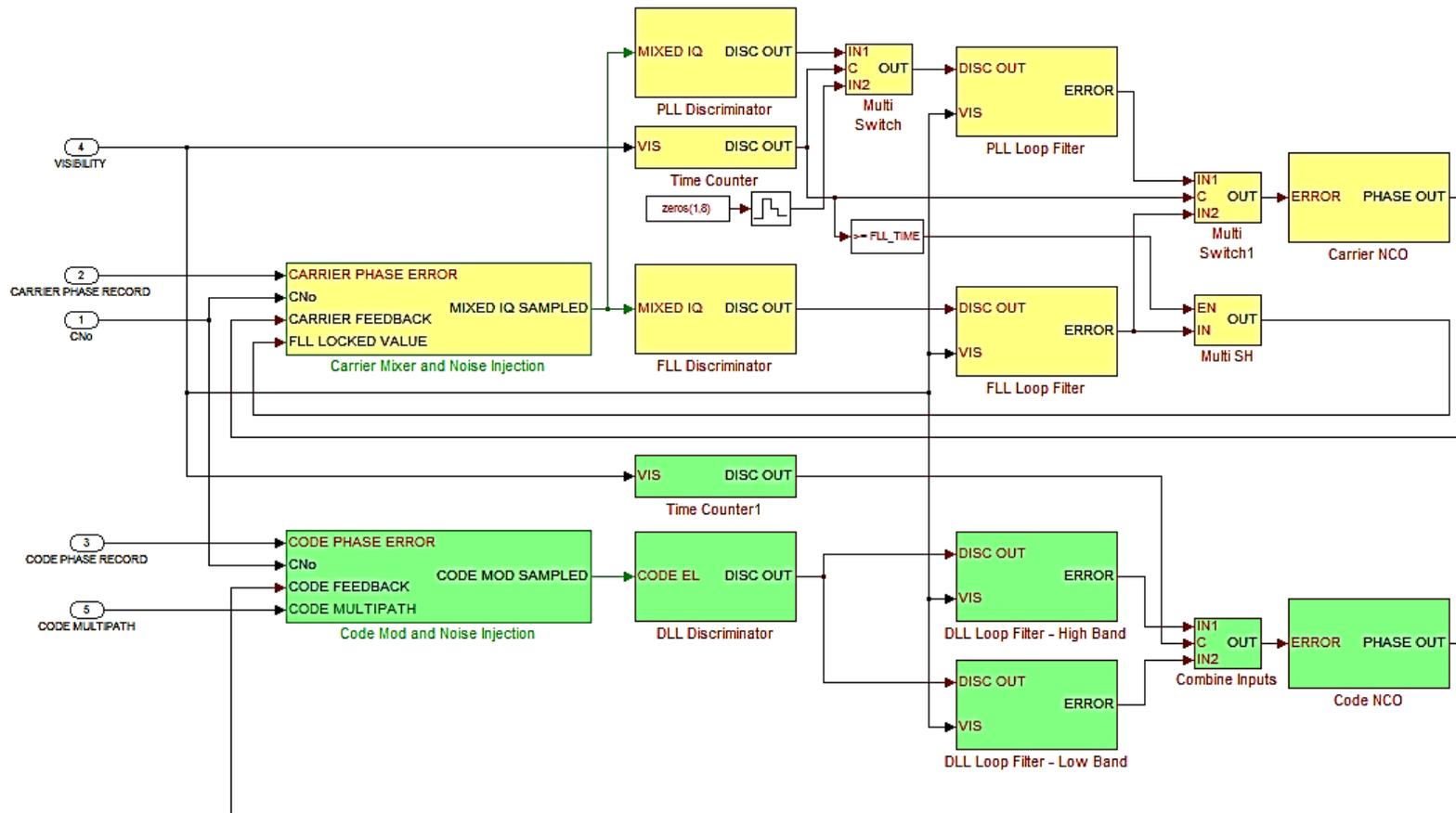
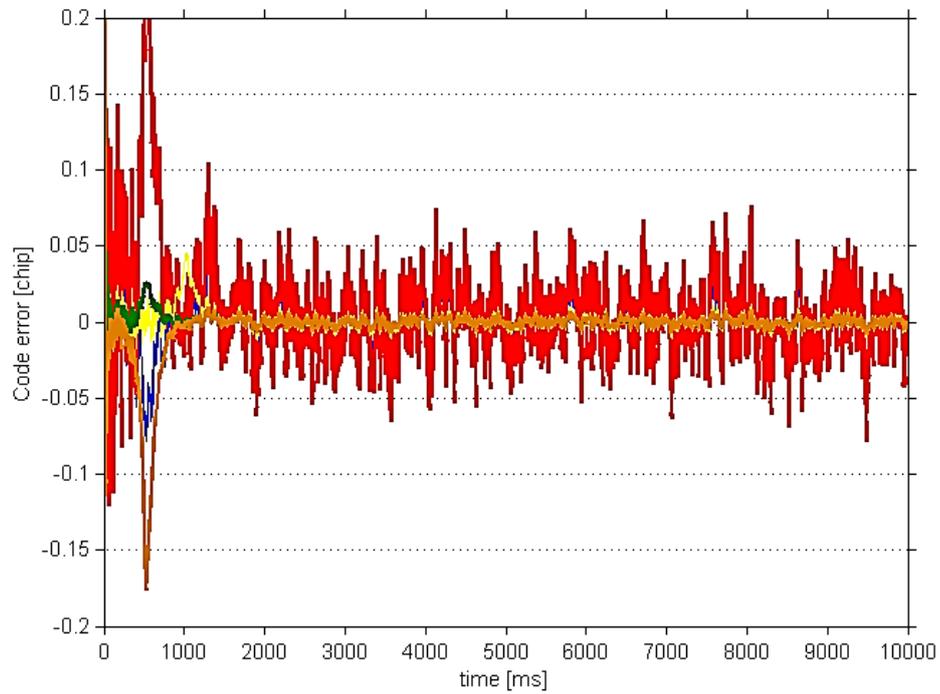


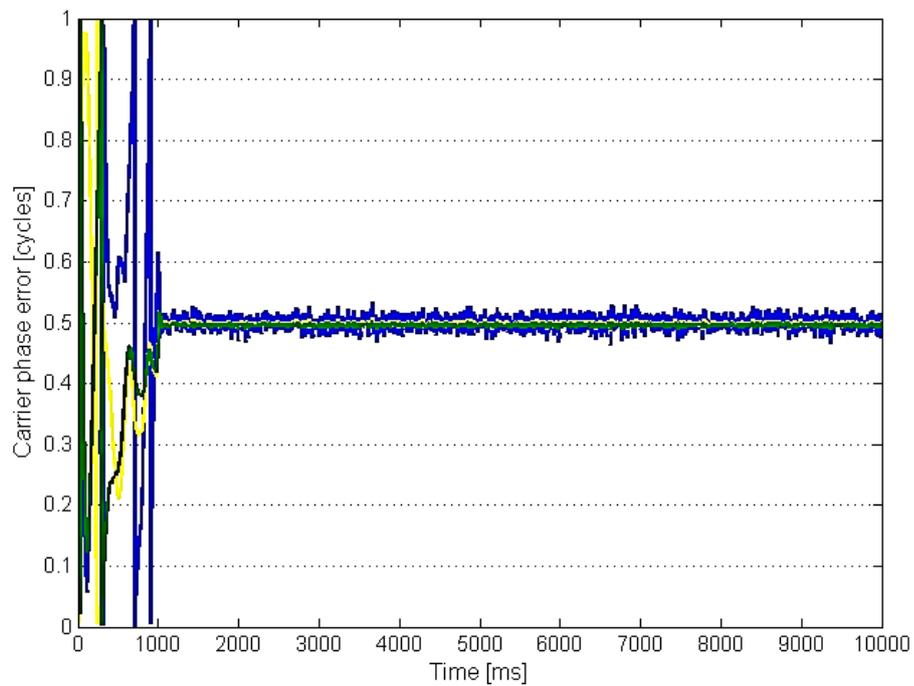
Figure 46: Simulated tracking loop diagram

### 3.4.1 Simulated scenario for tracking algorithm testing

The Figure 45 displays the diagram of the developed simulator, written in Matlab/Simulink environment, for the GPS and GALILEO signal tracking analysis. The simulation framework is semi analytic, meaning that it does not generate the RF signal, which should be then processed by the correlators, but it models directly the output of the correlators, as function of propagation code phase delays, carrier Doppler and received signal Cno. Using this framework, the amount of data that has to be processed is substantially reduced, since the RF signal is generally sampled with frequency in the order of the several MHz, while the correlator output is generated every millisecond or with even longer periods if longer accumulation is employed for weak signals. This simulator is an extensions of what has been already described for preliminary mission analysis. It basically propagates in time the positions of the GNSS vehicles and of the user vehicle (the LEO satellite), it then computes signal visibility, Cno and signal Doppler, and with this information it simulates the output of correlators from the 14 channels of the GNSS receiver. This correlator data stream is then processed using the tracking algorithms described in the previous sections. The drawback of this type of implementation, is that it is difficult to apply to the modeled correlator output the complex phenomenon that affect the propagation channel, such as multipath fading or ionosphere effect, which are generally modeled as channel impulse response functions to be applied directly to the RF signal. But since the simulator is strictly dedicated to tracking analysis in nominal conditions, it is considered sufficient to derive the tracking properties of loops. The block diagram that actually implements the tracking loops is displayed in Figure 46. Both FLL/PLL sequence and PULL-IN/LOCK loops bandwidth transitions are modeled in the simulator, as described in the previous section, while the bit decoding algorithms and the counters used to generate the observables are not included. A realization of the tracking output is displayed in Figure 47 for the code phase error, and Figure 48 for the carrier phase error.



**Figure 47: Code tracking error from simulation, expressed in code chips**



**Figure 48: Carrier phase tracking error from simulation, expressed in cycles**

From Figure 47 we can notice the transition between DLL in PULL-IN state and LOCK state, which is some channels is put in evidence by a spike in the code tracking errors: since the loop filter includes an integrator, it requires few milliseconds to reach convergence when switching from high to low bandwidth, therefore increasing temporarily the error associated with code tracking. The different scales in the errors from different channels is simply due to the different  $C_{no}$  of the received signal on that particular channel. In Figure 48 we can notice the transition between the FLL, where there is no phase control but only frequency locking, and the PLL where the phase locking is finally achieved with a standard deviation of a small fraction of a cycle. Having prepared the simulation environment, a rigorous performance analysis would have implied a high number of simulations with varying application conditions, in a Monte-Carlo framework, in order to have an estimate not only of the tracking accuracy but also of its uncertainty. However, the focus of the project was the receiver prototyping, and the implementation of a series of tools for mission analysis, tracking analysis and navigation analysis. Therefore the accurate analysis of the implemented methods and the sensitivity to the mission parameters will be performed in a second phase, together with the receiver PCB flight model design. For the purpose of receiver prototyping, a preliminary evaluation of the algorithms tracking accuracy using the developed simulator is given in Figure 49 for the code phase, and Figure 50 for the carrier phase.

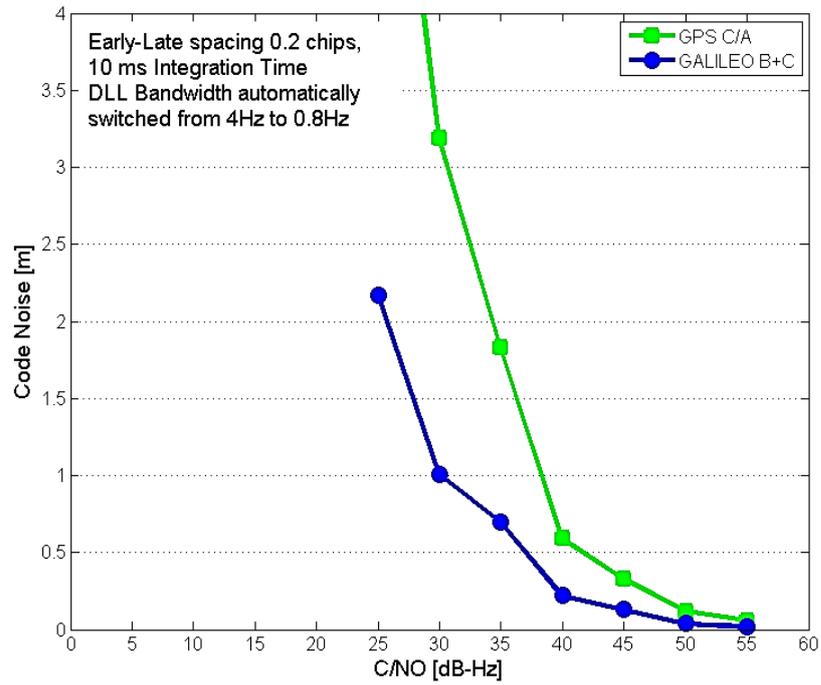


Figure 49: Code tracking error in meters as function of the input signal Cno

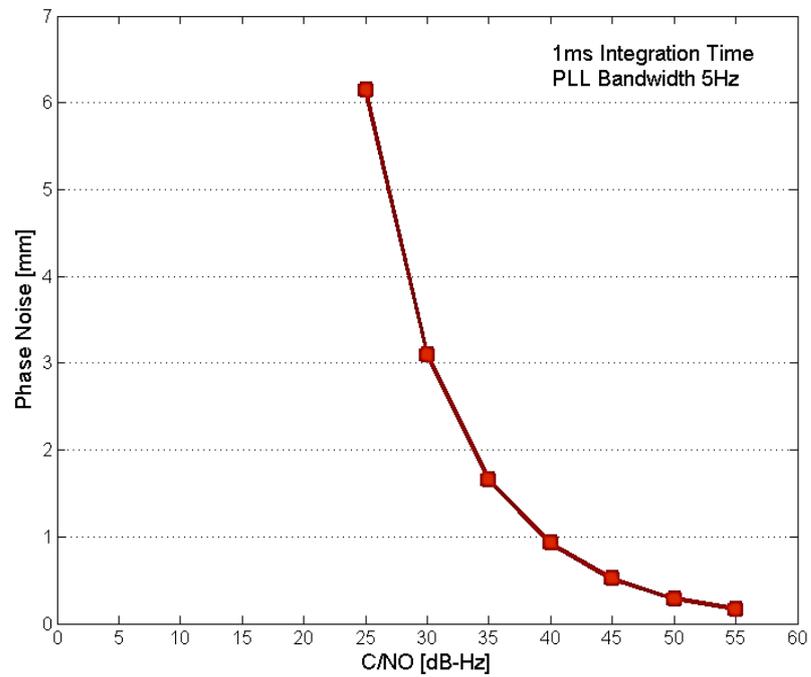


Figure 50: Carrier tracking error in millimeters as function of the input signal Cno

### 3.4.2 Real signal results

The GPS real signal tracking has been performed only to validate the signal tracking architecture: since real conditions cannot be reproduced, and the algorithm that provides an estimate of the received Cno is unreliable, the test is performed only to check general algorithm functionalities rather than characterizing the receiver performances, for which an RF simulator is required. Since there is no access to real carrier and code phase as happens in simulation, they are only evaluated from the output of the loops discriminators, which generates the error function that drives the NCO, before the loop filtering. Figure 51 displays the carrier phase error expressed in degrees generated by the PLL discriminator of one receive channel during GPS signal acquisition. Up to approximately 1400ms no signal is present, and the serial search process is still running.

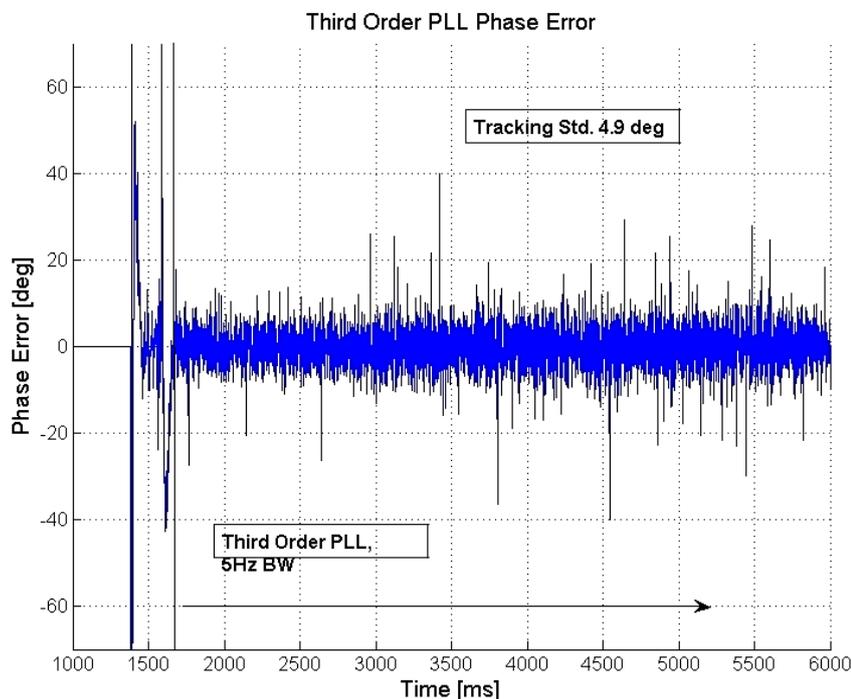
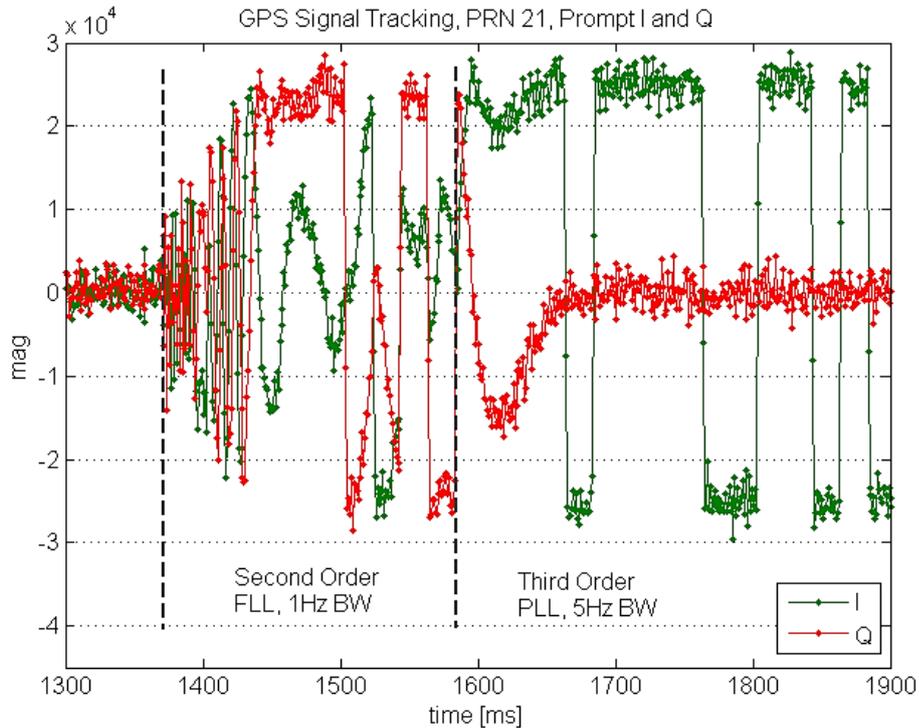


Figure 51: Phase error during real GPS signal acquisition

Then a signal is detected and confirmed, and we see the action of FLL between time 1400 and 1600, where there is no phase control. Finally the phase is acquired

by the third order PLL. The tracking standard deviation is approximately 4.9 degrees, which corresponds for the L1 carrier to roughly 2.5mm.



**Figure 52: I and Q prompt values during signal acquisition and tracking**

Corresponding to the phase tracking displayed in Figure 51, the Figure 52 shows the prompt I and Q channels values during the same time instants for the selected channel. During FLL operations, the frequency is acquired but the energy is still divided into in-phase and quadrature component due to the phase error between the local carrier replica and incoming signal. Then the phase is acquired by the third order PLL: for the GPS L1 signal the energy is only present in the in-phase component, where the bit transition are clearly evident from the in-phase value changes of sign.

---

## 4 Navigation Algorithm

### 4.1 Introduction

The purpose of the navigation algorithm is to process the observables, pseudoranges and carrier phases produced during signal tracking and SVs positions extracted from signal message, in order to compute the receiver position, velocity and clock offset with respect to the GPS time. Most of ground receivers and early space receivers perform this task using a *pure kinematic* method: the observables are combined epoch per epoch in a non-linear least square algorithm to compute the receiver position, time and velocity [4]. In an effort to improve the accuracy of the onboard navigation solution for the case of the space receiver, the use of Kalman filter and dynamical orbit model has been studied by various authors since the first application of this kind which dates back in 1992, taking advantage of the well-known model of the satellite orbits that can be exploited to smooth the error introduced by the observables noise.

In *Potti et al. 1995* [12] a Kalman filter based navigation algorithm is described in details, providing also accuracy results from simulations based on the MetOP-1 orbit. The main objective of the implementation is to smooth the effect of the Selective Availability, and the obtained accuracy is in the order of 15 to 30m. For this purpose the standard state vector (position, velocity, receiver's clock and phase constants) is augmented with one-per-channel bias parameters modelled as a first order or second order Gauss-Markov process. The filter dynamic model only includes the spherical gravity acceleration plus  $J_2$  and  $J_3$  terms. Sensitivity to orbit manoeuvres is also discussed by the author, which finally proposes to ignore the presence of any intentional disturbance in order to simplify the filter.

In *Hart et al. 1996* [13] the GPS Enhanced Orbit Determination Experiment (GEODE) flight software is described, which implements a Kalman filter based navigation algorithm with a 20 m accuracy goal in presence of Selective

Availability. The filter processing is split in two sections: the first is the filter state vector estimation that is executed every 30 seconds and that propagates the current state estimates and corrects it with the GPS measurements; the second section is a less accurate state propagation that runs every second. Compared to the previous author, the filter dynamical model is much more complete, since it accounts for gravity harmonic expansion up to order and degree 30, solar and lunar gravity perturbation and atmospheric drag. The atmospheric drag correction coefficient is included as part of the estimated vector.

In *Yunck 1996* [14] both least square and Kalman filter based estimation theories are described in details, covering real time on board estimation and ground based precise estimation. Reduced dynamic implementation by means of state augmentation with Gauss-Markov process is also presented as a solution to overcome dynamic model limitations. The effect of GPS orbit errors on the estimated solution are also presented by the author, showing that the errors in the GPS orbit are attenuated by roughly a factor of two in the dynamic solution of an orbiter at 1300 km altitude. At lower altitudes the dynamic model error grows especially of the atmospheric drag, and at the Space Shuttle altitude (300 km) the kinematic solution is nearly optimal due to the large dynamic modelling errors. The GRAPHIC code carrier combination is also introduced by the author as a solution to remove the ionospheric error in single frequency solutions.

In *Hass et al. 1999* [15] the GPS Onboard Orbit Determination Software (GOODS) is described. The algorithm implementation is similar to what is described by the previous authors, with a filter update of 30 seconds. The GOODS algorithm is tested not only in simulation but also on real data coming from the TOPEX/Poseidon mission. The achieved position accuracy results goes from 4.8 m 3D in simulation to 3.6 m 3D for the real data processing.

*Goldstein et al 2001* [16] proposes an update of the GEODE flight software, and shows also positioning results based on real data coming from the OrbView-1 mission, with Selected Availability deactivated (with an improvement of about 3

---

m). The updates are related to an approximation of the gravity harmonic expansion, to the use of code/phase combination for the single frequency ionospheric correction, and to the state vector augmentation with empirical accelerations. With these updates and Selective Availability deactivated, the 3D accuracy using the TOPEX/Poseidon real data (which are less affected by the ionosphere error) is in the order of 1 m.

*Reichert et al. 2002* [17] describes the RTG (Real-Time GIPSY developed at JPL) flight software and its performances with application to the SAC-C data and CHAMP data. It is shown that using global differential correction in ground experiments, 30 cm 3D RMS accuracy is achievable when processing dual frequency measurements, while using the broadcasted GPS ephemeris (the differential corrections are not available onboard in real-time), the meter level solution can be obtained as already introduced by the previous author. Compared to the previous implementation, the RTG uses gravity harmonic expansion up to order and degree 70, the DTM 94 atmospheric drag model, a solar radiation pressure model together with Earth orientation, polar motion and relativity model. In addition the RTG has the capability to utilize the reduced dynamic technique in which empirical accelerations are estimated in order to account for any dynamics left unmodelled. The state estimate includes then the correction for atmospheric drag and solar scale modelled as constants, while the empirical acceleration are modelled as stochastic processes with correlation time of 65 minutes and steady state process noise level of 15, 200 and 100 nanometers per seconds for the radial, cross-track and along-track components, respectively.

*Montenbruck et al. 2005* [18] further develops the concept of reduced dynamic orbit determination, introduced by *Wu et al. 1991* [19], that implies the augmentation of the state vector with empirical accelerations in order to overcome the dynamic model limitation. The author implements this method in a ground based POD software tool named GHOST [20], which performs both least square estimation and Kalman filter based estimation. Since the application is developed for off-line precise orbit determination, the associated dynamic model is fully

detailed, including gravity harmonic expansion up to order and degree 100, solid earth and ocean tides, lunar and solar gravity perturbations, solar radiation pressure, atmospheric drag, antenna phase offsets. Furthermore, precise clock and satellite ephemeris obtained from IGS [21] and CODE [22] are used in the computation. Both single frequency (GRAPHIC combination) and dual frequency measurement are tested for the POD, using data coming from the GRACE mission. Both single and dual frequency POD provides position accuracy in the order of 10 cm, and the associated empirical accelerations are between 20 and 300 nm/s<sup>2</sup>. Then *Montenbruck et al. 2008* [23] focuses on real time navigation onboard LEO satellites, proposing dynamic model simplification respect to the POD implementation already developed for the GHOST software, in order to reduce the computational workload. The workload is tailored for the ARM7TDMI processor with 30MHz clock frequency, and 30 s filter update period. Sensitivity analysis is proposed by the author respect to the implemented model complexity (mainly the order of gravity harmonic expansion) and the used clock and ephemeris data (broadcasted, IGS ultra-rapid [21], JPL real-time [24] and CODE final [22]). The algorithm is tested on data coming from CHAMP, GRACE, TerraSAR, ICESat, SAC-C and MetOP missions. The obtained accuracy is in the order of 1 m using the lightest dynamic model implementation and the broadcasted clock and ephemeris (with variations according the satellite altitude since higher altitude satellites are less affected by the ionosphere error), while using the JPL real time data and the 70x70 gravity harmonic expansion, the 3D position accuracy improves up to 40 cm RMS for the single frequency GRAPHIC combination.

*Bock et al. 2008* [25] focuses on single frequency precise orbit determination, comparing three different techniques to reduce the effect of the ionosphere. The first one is based on ionosphere modelling, the second is based on the code/phase GRAPHIC combination and the last technique directly estimates the ionosphere as part of the filter processing. The conclusion is that the preferred method is either the GRAPHIC combination or the direct estimation. It is also stated that the

quality of code measurement is a crucial aspect to define the final accuracy. The algorithms are tested on the GRACE and MetOP data. The obtained accuracy is between 10 and 20 cm, recalling that these results are related to ground processing using POD software tool. The single frequency techniques comparison is however still valid also for real time onboard processing.

*Montenbruck et al. 2008* [26] focuses on the antenna phase center calibration for the precise positioning of LEO satellites. It is shown that the ground calibration of the LEO antenna is recommended in order to obtain the best possible a priori phase pattern when the precise positioning requirements are stringent, while correction factors can be estimated as part of the filter adjustment during the estimation process, by locking at the phase residuals during the computation. Even if the technique is applied to least square estimation in a POD software tool, the provided analysis and results are relevant also for the onboard real time orbit determination processing.

*Choi et al. 2010* [27] proposes an alternative algorithm for the filter implementation based on Unscented Kalman Filter UKF instead of Extended Kalman Filter EKF. The algorithm is tested on real data coming from the CHAMP and KOMPSAT-2 missions. Regardless of the final accuracy, the general conclusion is that there is small different between the two implementation, that does not justify the implementation of estimation filters tailored for nonlinear system. These results indicate that EKF based algorithm represents the best choice for the estimation process, due to the specific conditions of the application: the process model is nonlinear, but this nonlinearity is weak especially for short integration time periods, so that linear approximation perfectly matches with the desired model accuracies.

As result of the reported literature review, the following section describes the architecture of the navigation algorithm implemented in the FPGA based GNSS receiver. The same mission simulator developed for preliminary requirements analysis and extended for tracking algorithm development, has been used also to

verify the algorithm performances in different conditions, using different signal combination. The details about GNSS observables realistic error modelling are provided in section 4.3. Then the navigation results are reported in section 4.4, where the expected algorithm performances during the mission operations are evaluated, and the requirements compliancy is verified.

## 4.2 Navigation Algorithm description

The algorithm described in this section starts from the work done by *Montenbruck et al. 2009* [28] for the Sentinel-3 mission, that includes the contributions given by the reviewed authors. The Sentinel-3 mission is part of the ESA's Global Monitoring for Environment and Security program (GMES) and is scheduled to be launched in 2013; therefore we can assume that its on-board filter represents the *state-of-the-art* technology. The real-time navigation filter is similar to what already described in the previous section, and has been successfully implemented for the PROBA-2 mission [29], and tested on data coming from CHAMP, GRACE, TerraSAR, ICESat, SAC-C and MetOP missions. The described algorithm is a reduced dynamic estimation filter; therefore it includes a dynamic model of the orbit, and a filtering procedure that combines the a priori information given by the model with the measurements from the GNSS receiver. The alternative algorithm, which is mentioned in the following sections, is based only on GNSS observables processing without any dynamic model information, and is named *kinematic solution*.

### 4.2.1 Time

The preferred choice for the time reference system is the GPS time, an atomic time scale that differs from the *International Atomic Time TAI* in the chosen offset. This offset is constants, therefore there are no leap seconds that changes in time to be added or subtracted to the timescale. Moreover the relation with other time scale such as *Coordinated Universal Time UTC* is or *Terrestrial Time TT* is simple since they are only separated by an integer number of seconds. Finally, the

on-board filter estimations are implicitly tied to the GPS time scale, since the receiver clock offset that is part of the state vector is related to this time scale.

### 4.2.2 Reference frame

The modeling of LEO spacecraft trajectories involves two different types of reference frames. An inertial reference frame is the preferred choice to describe the equations of motions and the position of third bodies such as the Sun and Moon. However the description of the Earth gravity field requires an Earth-fixed reference frame, since the required gravity field model does not show cylindrical symmetry with respect to the Earth rotation axis. Moreover, the navigation estimation coming from GNSS measurements are implicitly tied to an Earth fixed reference frame, since the broadcasted ephemeris provide space vehicles position coordinates in this frame. The *International Celestial Reference Frame ICRF* and the *International Terrestrial Reference Frame ITRF* are the realization of the mentioned frames. Whether we decide to express the equations of motion in the inertial reference frame or in the Earth fixed reference frame (by taking into account apparent accelerations), the mathematical transformation between the two frames is required, and its complete expression is (according to the standard IAU1980)

$$\mathbf{r}_{ITRS} = \mathbf{\Pi}(t)\mathbf{\Theta}(t)\mathbf{N}(t)\mathbf{P}(t)\mathbf{r}_{ICRS}$$

where the rotational matrices  $\mathbf{P}$ ,  $\mathbf{N}$ ,  $\mathbf{\Theta}$  and  $\mathbf{\Pi}$  describe the coordinates changes due to precession, nutation, Earth rotation and polar motion respectively. The related models accounts for:

- Precessions, describing secular variations in the Earth axis orientation and related change in the vernal equinox.
- Nutation, describing the short term variation in the Earth axis orientation and related change in the vernal equinox.

- Earth's rotation in relation to the UT1, for which difference between UT1-TAI is required and is provided by the IERS Earth Observation Parameters (EOP) [30].
- Polar motion, for which measured polar coordinates are required and are provided by the IERS Earth Observation Parameters (EOP) [30].

The complete and exhaustive description of the concepts behind these transformations can be found in [31], but some simplification is required in order to implement the transformation for the on board real time algorithm. Fortunately it is not necessary to fully evaluate the expression above, since good results can be obtained by neglecting the influence of precession and nutation, and considering a constant rotation rate of  $0.7292115 \times 10^{-4}$  rad/s about the *Celestial Ephemeris Pole CEP* axis, so that the update of UT1-TAI parameter is not required. The polar motion is still required to translate the *CEP* rotation into *ITRF*, which pole position is taken as fixed reference (*IRP*), but the motion parameters from IERS can be uploaded on board through the telemetry data with low frequency, for example once per week. The remaining effects can be well compensated by empirical acceleration estimated as part of the state vector. It is particularly convenient to express the equation of motion in the Earth fixed frame *ITRF* since the GNSS space vehicle positions, provided as part of the signal message, are given in this reference frame. Since this is not an inertial frame, apparent acceleration is to be taken into account. In particular additional centrifugal and Coriolis acceleration are given by

$$\Delta \ddot{r}_{CC} = -2\omega \times v - \omega \times \omega \times r$$

where  $r$  and  $v$  are expressed in *ITRF*. According to the approximation taken before (we assume a constant rotation rate about the *CEP* axis, and we only take into account the polar motion), the rotation vector can be expressed as

$$\omega = \Pi(t) \begin{pmatrix} 0 \\ 0 \\ \omega_{\oplus} \end{pmatrix}$$

Since the equation of motion are expressed in *ITRF*, Sun and Moon positions generally given in an inertial reference frame, have to be translated in Earth fixed coordinates, using an approximated rotation matrix that includes Earth rotation, precession and polar motion, while the nutation can be ignored.

### 4.2.3 Dynamic model

The on board real time navigation filter is built on a high grade trajectory model that accounts for the following accelerations:

- Gravitational forces
  - Earth gravity field
  - Sun and Moon
  - Earth tides
- Non-gravitational forces
  - Atmospheric drag
  - Solar radiation pressure
- Apparent forces
  - Coriolis and centrifugal
- Empirical accelerations

Each component is briefly described in the following sections. According to the available computational resources on the GNSS receiver, the models can be simplified to reduce the workload, especially the Earth gravity harmonic expansion.

#### 4.2.3.1 Earth gravity field

The Earth gravity acceleration is modeled as a harmonic expansion expressed in spherical coordinates, given by:

$$\ddot{r} = \Delta \frac{GM_{\oplus}}{r} \sum_{n=0}^{\infty} \sum_{m=0}^n \frac{R_{\oplus}^n}{r^n} \bar{P}_{nm}(\sin \phi) (\bar{C}_{nm} \cos(m\lambda) + \bar{S}_{nm} \sin(m\lambda))$$

where  $n$  and  $m$  are the degree and order respectively,  $r$  is the spherical coordinate's radial component,  $\phi$  the geocentric latitude and  $\lambda$  the geocentric longitude,  $P_{nm}$  is the associated Legendre polynomial and  $C_{nm}$  and  $S_{nm}$  are the geopotential coefficients. Geopotential coefficients with  $m = 0$  are called zonal coefficients since they described the part of the potential that is not dependent on the longitude. All  $S_{n0}$  are equal to zero because of their definition while for the other zonal terms the notation  $J = -C_{n0}$  is generally adopted. The other geopotential coefficients are named tesseral when  $m < n$  or sectorial when  $m = n$ .

In the computation of the gravity potential, several recursion relationships can be used, simplifying the processing of the polynomials. The complete mathematical treatment of the recursive equations can be found in [31], which provides implementable equations to model the Earth gravity harmonic expansion.

The selected model that provides the expansion coefficients is the GGM0xS, derived from the analysis of data coming from the GRACE mission. This geopotential is at the base of the adopted IERS conventional model of 2010 [32], the EGM2008 model, with C20 C30 and C40 coefficients modification as suggested by *Cheng et al 2008* [33].

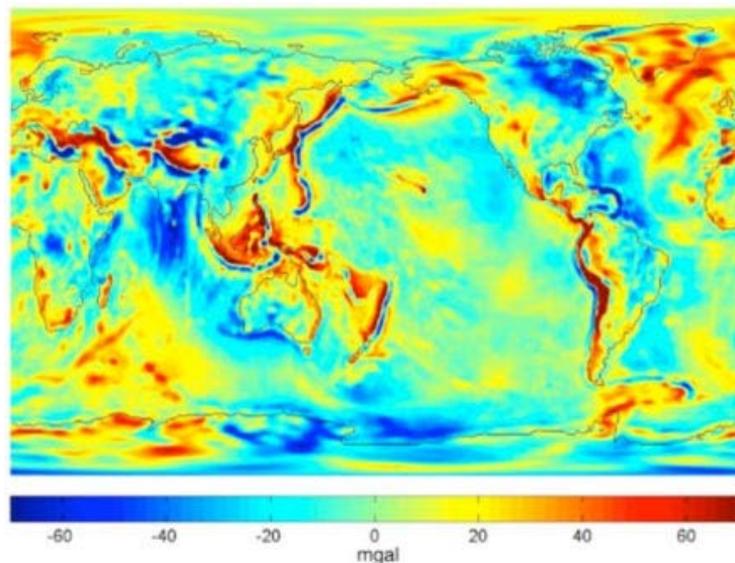


Figure 53: Gravity anomalies from four years (2003-2006) of GRACE data (GGM03S). Reproduced from [34].

The current release of the GRACE based model is the GGM03S [34], which is usable up to order and degree 130, while providing coefficients up to degree and order 180. Practical implementation that provides gravity acceleration description with an accuracy of  $10^{-10}$  km/s<sup>2</sup> (that is approximately the acceleration caused by the solar radiation pressure on a space vehicle with a reference area-to-mass of 0.01 m<sup>2</sup>/kg), suggests that for an 800 km altitude space vehicle a 30x30 gravity harmonic expansion is sufficient, while for a 400 km altitude space vehicle a 50x50 expansion is required [28]. These levels of expansions are suggested by *Montenbruck et al. 2009* [28] as a result of a residual error evaluation given from series truncation at different altitudes, recalling that for a given harmonic of degree  $n$ , the associated acceleration decreases with the  $n+1$  power of the space vehicle distance from the Earth. Therefore higher satellites will be less subjected to higher order harmonic perturbations. The  $10^{-10}$  km/s<sup>2</sup> target accuracy of the dynamic models is again proposed by *Montenbruck et al. 2009* [28] and *Montenbruck et al. 2008* [23] as a compromise between accurate orbit modeling and the actual compatibility of the algorithm with the on board computational resources. When integrating the model, this errors accumulates causing approximately 5m 3D position error after one orbit period for a LEO satellite (between 200km and 2000km altitude). This level of accuracy is necessary to heavily constrain the solution and reduce the errors associated with the GNSS observables. The reference epoch of the GGM03S gravity field model is 1 January 2005. Terms for which rates were modeled have been mapped to this epoch, so they were not fixed to the IERS convention. These terms include  $C20$ ,  $C30$ ,  $C40$ ,  $C21$  and  $S21$ . From practical point of view the time dependent variation of these coefficients can be ignored, which is adequate if the gravity field reference epoch is within 5-10 year from the epoch of interest.

#### **4.2.3.2 Sun and Moon gravity effect**

Gravitational acceleration caused by the Sun and the Moon are in the order of  $10^{-9}$  km/s<sup>2</sup> for a satellite around 800 km altitude. Therefore their effect requires to be modeled as part of the filter, but their coordinates need not to be known with very

accurate precision to meet the required acceleration accuracy. A simple series expansion can be used with reference accuracy of about 1%. For both the Sun and the Moon, the mathematical models and series can be found in [31]. The provided positions are given in an inertial frame with reference to the ecliptic; therefore a first rotation is required to compute Cartesian coordinates with respect to the equator. Then, since the coordinates refers to mean equinox of epoch J2000, the precession need to be taken into account to correct for the secular changes in the reference axis. Finally, in order to obtain *ITRF* coordinates, approximated Earth rotation and polar motion have to be applied. The complete transformation is expressed as:

$$\mathbf{r}_{ITRS} = \mathbf{\Pi}(t)\mathbf{R}_z(GMST|t_0)\mathbf{R}_z(\omega_{\oplus}(t - t_0))\mathbf{P}(t)\mathbf{R}_x(-\varepsilon) \begin{pmatrix} r_{SM}\cos(\lambda_{SM})\cos(\beta_{SM}) \\ r_{SM}\sin(\lambda_{SM})\cos(\beta_{SM}) \\ r_{SM}\sin(\beta_{SM}) \end{pmatrix}$$

where  $\varepsilon = 23.43929111$  degrees, *GMST* stands for *Greenwich Mean Sidereal Time* and is related to UT1 by a conventional relation,  $r_{SM}$ ,  $\lambda_{SM}$  and  $\beta_{SM}$  are the spherical coordinates of the Sun or the Moon given by the series expansion. In the formulation the Earth rotation is expressed with respect to an epoch  $t_0$  for which actual UT1 time can determined, and selected possibly in close vicinity to the expected mission epoch. With this approximation the UT1-TAI parameter does not need to be periodically updated. The gravitational acceleration associated with the Sun or the Moon can be then expressed as

$$\ddot{r} = GM \left( \frac{s-r}{|s-r|^3} - \frac{s}{|s|^3} \right)$$

where  $r$  and  $s$  are the geocentric coordinates of the satellite and of the Sun or Moon. Besides the gravitational effects, the Sun and Moon positions are used to model the solid Earth tides and the shadowing effect for the solar radiation pressure computation.

### 4.2.3.3 Earth tides

The gravitation forces of the Sun and the Moon act on the body of the Earth, leading to a time varying deformation of the planet, and therefore to a variation of its gravity potential. The small period variations of the solid body are called *solid Earth tide*, while the ocean response to the time varying lunisolar perturbation is called *ocean tide*. The acceleration perturbation caused by this phenomenon is around  $10^{-10}$  km/s<sup>2</sup> for a satellite orbiting at 800km [31]; therefore the effect modeling is required to meet the desired gravity field accuracy.

The changes induced by the solid Earth tide in the free space gravity potential, are most conveniently modeled as variation in the standard geopotential coefficients  $C_{nm}$  and  $S_{nm}$ , as function of the relative position between the Earth, the Sun and the Moon. The normalized coefficient corrections are given by [32]

$$\Delta\bar{C}_{nm} - i\Delta\bar{S}_{nm} = \frac{k_{nm}}{2n+1} \sum_{j=2}^3 \frac{GM_j}{GM_{\oplus}} \left(\frac{R_e}{r_j}\right)^{n+1} \bar{P}_{nm}(\sin \Phi_j) e^{-im\lambda_j}$$

where  $R_e$  is the equatorial radius of the Earth,  $GM_{\oplus}$  is the gravitational parameter of the Earth,  $GM_j$  is the gravitational parameter of the Moon ( $j = 2$ ) and the Sun ( $j = 3$ ),  $r_j$  is the distance from the geocenter of Moon or Sun,  $\phi_j$  is the Earth fixed geocentric latitude of Moon or Sun,  $\lambda_j$  is the Earth fixed geocentric longitude of Moon or Sun and  $k_{nm}$  is the Love number. The Love numbers are used to describe the contributions of tides to the gravity potential, and considering the effect of ellipticity and of the Coriolis force due to Earth rotation three  $k$  parameters,  $k_{nm}^{(0)}$ ,  $k_{nm}^{+}$  and  $k_{nm}^{-}$ , are required to characterize the changes, except for  $n = 2$  where  $k_{nm}^{-}$  is zero due to mass conservation. In fact the equation given above only provides coefficient correction when  $n = 2$  or  $n = 3$  for all  $m$ , while the computation of the changes in the degree 4 coefficient caused by the degree 2 tide is given by

$$\Delta\bar{C}_{4m} - i\Delta\bar{S}_{4m} = \frac{k_{2m}^{(+)}}{5} \sum_{j=2}^3 \frac{GM_j}{GM_{\oplus}} \left(\frac{R_e}{r_j}\right)^3 \bar{P}_{nm}(\sin \Phi_j) e^{-im\lambda_j}$$

At a first approximation the tidal gravity potential leads to an elastic deformation of the Earth, therefore the Love numbers are real numbers. The Earth is however elastic only to first order and anelasticity of the mantle causes the Love number to have a small imaginary part, that represent the fact that there is a small phase offset between the tidal potential and the tide phenomenon. However for practical implementation, the anelastical effect can be neglected using the real Love numbers provided in the IERS convention 2010 [32] and the above equation to compute coefficients corrections.

$N$	$m$	$k_{nm}$	$k_{nm}^+$
2	0	0.29525	-0.00087
2	1	0.29470	-0.00079
2	2	0.29801	-0.00057
3	0	0.093	-
3	1	0.093	-
3	2	0.093	-
3	3	0.094	-

**Table 35: Nominal values of Solid Earth tides external potential Love numbers**

The provided Love numbers, that describes how the tides potential caused by the Sun and the Moon is translated into free space Earth gravity potential, can be considered constant values only at first approximation. If the Earth tide phenomenon is examined in details, one would find the Love number values depend on the frequency of the tidal potential excitation on the Earth, dividing the potential itself in diurnal, semi-diurnal and long term contributions. In a rigorous computation the geopotential coefficient correction would be a dual step process, providing at first a nominal value correction, and then a frequency related correction especially for coefficients  $C20$ ,  $C21$  and  $S21$ . The detailed model can be found in IERS Convention 2010 [32].

Ocean tide also plays an important role in satellite geodesy, although its magnitude is ten times smaller than the Earth solid tide. The detailed treatment can be found in [32], however for practical implementation the ocean tide contribution can be neglected to simplify the computation. Moreover, even the Earth solid tide modeling can be further simplified for the filtering implementation, using the simple  $k_{20}$  approximation described in *Rizos and Stolz 1985* [35]. With this simple approximation, the additional acceleration is modeled as

$$\dot{r} = \frac{k_{20}}{2} \frac{GM_j}{r_j} \frac{a_e^5}{r^4} \left( 3 - 15 \cos^2 \theta \frac{\vec{r}}{r} + 6 \cos \theta \frac{\vec{r}_d}{r_d} \right)$$

where  $\theta$  is the angle between the geocentric vector  $r$  (spacecraft position) and the geocentric vector  $r_d$  (disturbing body).

One more thing to mention about the Earth tides and the geopotential coefficients in general is that the degree 2 zonal tide generation potential has a mean value that is nonzero. This time independent potential produces a permanent deformation and a consequent time independent contribution to the geopotential coefficient  $C_{20}$ . If the contribution is included in the coefficient, it is called *zero tide*, while if the contribution is not included the coefficient is *tide free*. In case of *zero tide* coefficients, the Earth tide correction cannot be implemented with the formulation above since the time independent part would be counted twice. Therefore it is necessary to correct for the permanent part in order to compute the  $\Delta C_{20}$ . The permanent part is generally provided with the gravity model or can be found in [32] as specified the IERS conventional model.

#### 4.2.3.4 Atmospheric drag

The atmospheric forces are the largest non-gravitational forces that acts on LEO satellites, and at the same time they are the major cause of dynamic modeling errors, since their accurate modeling is difficult for various reasons. The most important of these reasons is that the physical properties of the upper atmosphere are not understood in details, and the most accurate models depends on local

parameters such as the solar flux or the exospheric temperature, which are generally not available on board a LEO spacecraft and are difficult to predict accurately. In general the atmospheric models can be divided into two types: the models of the first type are characterized by their dependence on altitude and eventually on angular distance with respect to the Sun pointing vector, and they are independent on any other parameter. The models of the second type are also dependent on the amount of energy coming from the Sun, which shows seasonal and secular variations due to a number of effects. A variety of models exists that differs of about the 20% at 300km altitude and even more at higher altitude, each characterized by a certain level of complexity and by the need of additional parameters. The description of applicable models can be found in [31]. Considering the variability among the models, both in terms of provided density accuracy (which is generally around the 15%) and in terms of computational complexity, it appears fully justified to select for the estimation process a simple model that does not depend on external information, with the additional benefit of reduced computational complexity. For this purpose a widely used standard atmosphere is represented by the Harris-Priester model [31], that is described in this section.

The dominant atmosphere force acting on the LEO spacecraft, named drag, is directed opposite to the velocity of the satellite with respect to the atmosphere itself, which in first approximation can be considered to rotate together with the Earth (the maximum error associated with these approximation is of the order of 40%). Additional force component directed perpendicular to the velocity vector, named lift, can be safely neglected in most cases. The drag can be expressed as

$$\dot{r} = \frac{1}{2} C_D \frac{A}{m} \rho v^2 \mathbf{e}_v$$

where  $A$  is the cross-section reference area of the spacecraft,  $m$  is its mass,  $v$  the velocity magnitude,  $\rho$  is the local atmosphere density and  $C_D$  is the drag coefficient. The drag coefficient depends on the interaction of the atmosphere with the body of the spacecraft and is related to the properties of its surfaces and

on its shape. The range is usually between 1.5 and 3.0. Since the parameter is difficult to be measured even in laboratory experiments, a first tentative value of 2.2 fits with most of the cases. Another relevant parameters is the area to mass ratio  $A/m$ , where the area is a reference quantity that is fixed in order to evaluate the  $C_D$  parameters. A realistic value for a spacecraft is around  $0.01\text{m}^2/\text{kg}$ , which provides  $1\text{m}^2$  of reference area for a  $100\text{kg}$  space vehicle.

The acceleration due to the drag depends on the atmosphere density; therefore the modeling of these property is one of the challenging task of modern orbit determination algorithms. The Harris-Priester model [31] is a simple atmospheric model based on the properties of the upper atmosphere as determined from the solution of the heat conduction equation under quasi hydrostatic conditions. While neglecting the explicit dependence of semi-annual and seasonal latitude variations it has been extended to include the diurnal density bulge. The apex of this bulge is delayed by approximately 2 hours, equivalent to a location  $30^\circ$  to the east of the subsolar point. The apex and antapex densities are computed by exponential interpolation between tabulated values as function of the altitude over the reference ellipsoid, and are given by

$$\rho_m(h) = \rho_m(h_i) \exp\left(\frac{h_i - h}{H_m}\right)$$

$$\rho_M(h) = \rho_M(h_i) \exp\left(\frac{h_i - h}{H_M}\right)$$

where  $h_i < h < h_{i+1}$  is the altitude. The corresponding scale height are given as

$$H_m(h) = \frac{h_i - h_{i+1}}{\ln(\rho_m(h_{i+1})/\rho_m(h_i))}$$

$$H_M(h) = \frac{h_i - h_{i+1}}{\ln(\rho_M(h_{i+1})/\rho_M(h_i))}$$

The diurnal density variations from the apex to antapex due to solar radiation are computed using cosine function, and are given by

$$\rho(h, \psi) = \rho_m(h) + (\rho_M(h) - \rho_m(h)) \cos^n\left(\frac{\psi}{2}\right)$$

where  $\psi$  is the angle between the satellite position vector and the apex of the diurnal bulge, and  $n$  is used to take into account the effect of latitude, taking value of 2 for low inclination orbits and 6 for polar orbits. Finally a good approximation for the height neglecting the polar motion is given by

$$h = r - r_s$$

$$r_s = \frac{R_e(1-f)}{\sqrt{1 - (2f - f^2)\cos^2\delta}}$$

where  $r$  is the magnitude of the satellite position vector,  $R_e$  is the equatorial radius of the Earth,  $f$  is the Earth's flattening coefficient and  $\delta$  is the declination of the satellite which is assumed to equals the geocentric latitude of the subsatellite point. The Figure 54 displayed the apex and antapex densities as function of the height for mean solar activity derived from the Harris-Priester model.

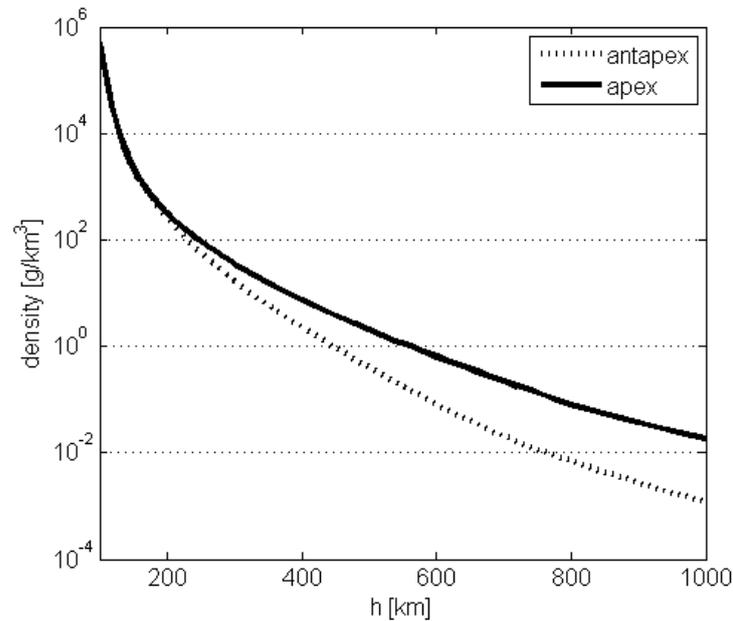


Figure 54: Harris-Priester density profile for mean solar activity.

#### 4.2.3.5 Solar radiation pressure

A satellite that is exposed to the solar radiation experiences a small force that arises by the absorption or reflection of photons. The associated acceleration is in the order of  $10^{-10}$  km/s<sup>2</sup> on a space vehicle with a reference area-to-mass of 0.01

$\text{m}^2/\text{kg}$ . If absorbed, the pressure is the power flux density divided by the speed of light while if the radiation is reflected, the radiation pressure is doubled. In most of the cases a combination of reflection and absorption phenomenon describes the situation, and the reflective coefficient  $\varepsilon$  is used to quantify the percentage of occurring reflection. In the general case, the orientation of the satellite surfaces should be taken into account, and a simplified macro model of the spacecraft structure should be used for precise orbit determination applications. However, in order to simplify the on board computation, we can assume that the reflecting surface is always normal to the direction of the Sun. With this assumption, the acceleration of the spacecraft due to solar radiation can be expressed as

$$\ddot{r} = -P_{SUN} C_R \frac{A}{m} \frac{s}{|s|^3} AU^2$$

where  $P_{SUN}$  is the solar radiation pressure at 1 AU distance and is equal to  $4.56 \times 10^{-6} \text{ Nm}^{-2}$ ,  $A$  is the spacecraft reference area and  $m$  its mass,  $s$  is the Sun-Spacecraft vector pointing to the Sun and  $C_R$  represents the property of the absorbing/reflecting surface. The Table 36 provides the coefficient for some representative materials.

Material	$\varepsilon$	$1 - \varepsilon$	$C_R$
Solar Panel	0.21	0.79	1.21
High-gain antenna	0.30	0.70	1.30
Aluminum coated mylar solar sail	0.88	0.12	1.88

**Table 36: Reflective  $\varepsilon$ , absorption  $1 - \varepsilon$  and solar radiation pressure coefficient  $C_R$**

As for the drag, a standard value of  $0.01\text{m}^2/\text{kg}$  is taken as reference for the area to mass ratio in the pressure model. The described model has been derived under the assumption of full illumination by the Sun. However for most LEO satellite partial or total eclipse occurs when the spacecraft passes the night side of the Earth. In order to simplify the computation, the partial eclipse phase can be ignored accounting only for a cylindrical shadow model behind the Earth, and

assuming also a simplified Earth model that does not account for the atmosphere and for the geoid oblateness.

#### 4.2.3.6 Empirical accelerations

In order to compensate for dynamic model error and simplifications, empirical acceleration can be introduced as part of the state vector and updated in the estimation process, implementing the so called reduced dynamic orbit determination. This acceleration are generally given in an orbital reference frame aligned with radial, along track and cross track direction, and have to be rotated in order to be included in the *ITRF* based model equations. In order to statistically represent these accelerations, a first order Gauss-Markov process is generally implemented. This process is exponentially auto-correlated and is characterized by two parameters: the time constant  $\tau$  and the steady state variance  $\sigma$ . For the Kalman filter implementation, the time updated equation of the process is given by

$$m_j = \exp\left(-\frac{t_{i+1} - t_i}{\tau_j}\right)$$

while the associated variance is given by

$$q_j = (1 - m_j^2)\sigma_j$$

where  $j$  covers the radial, along track and cross track directions that are generally characterized by different parameters. For example, *Reichert et al. 2002* [17] proposes stochastic processes with correlation time of 65 minutes and steady state process noise level of 15, 200 and 100 nanometers per seconds for the radial, cross-track and along-track components, respectively.

#### 4.2.4 Numerical Integration

A variety of methods have been developed for the numerical solution of ordinary differential equations, which provides different level of accuracy and requests different level of computational effort. While the computational requirement is not a concern for the offline precise orbit determination, it has to be carefully

---

evaluated for the real time onboard applications. With the hardware available on modern GNSS space receivers, time step around 30s is achievable, while if the navigation solution is required at higher data rate, the actual processing time is likely to exceed the selected time step. However the motion of LEO satellites is sufficiently smooth to allow interpolation between points computed by the filtering algorithm. Therefore higher frequency intermediate step can be computed using the appropriate interpolator. Based on existing literature on real time on board processing, a 4<sup>th</sup> order Runge-Kutta integrator with Richardson extrapolation and Hermite interpolation [28] offers a consistent 5<sup>th</sup> order integrator/interpolator that is well suited for real time navigation systems. If a reference time step of 30 seconds is select, then it is likely that the interpolation would not be required considering the available hardware resources, leading to a clean 4<sup>th</sup> order Runge-Kutta implementation. The details on RK4 implementation are well known and are not reported in this text, but they can be reviewed in [31]. The actual time step generally depends on the mission requirements, i.e. the requirements of the on board computer or the requirements of the science payload or of the science community that will make use of the data. Based on literature about past and present space missions, generally the fix requirement is between 1Hz (meaning one fix per second) and 0.3Hz (meaning one fix every 30 seconds).

#### **4.2.5 Measurement processing**

Based on the actual GNSS receiver implementation, the real time navigation filter can be based on either dual frequency measurements or single frequency measurements, and the processing can relies on pseudorange only or can include the carrier phase. While the use of dual frequency receiver is mandatory to eliminate the ionospheric delay in the pure kinematic solution, the same does not apply for the real time navigation filter. Here three different combination of ionosphere free linear combination of basic GNSS pseudorange (C/A, P1, P2) and carrier phase measurements (L1, L2) can be considered, together with the standard single frequency L1/E1 pseudorange observable.

- *Single frequency L1/E1 pseudorange only*

- *GPS  $\rho_{CA}$*

- *GALILEO  $\rho_{BC}$*

- *Ionospheric free dual frequency pseudorange combination*

$$\rho_{P12} = 2.54\rho_{P1} - 1.54\rho_{P2}$$

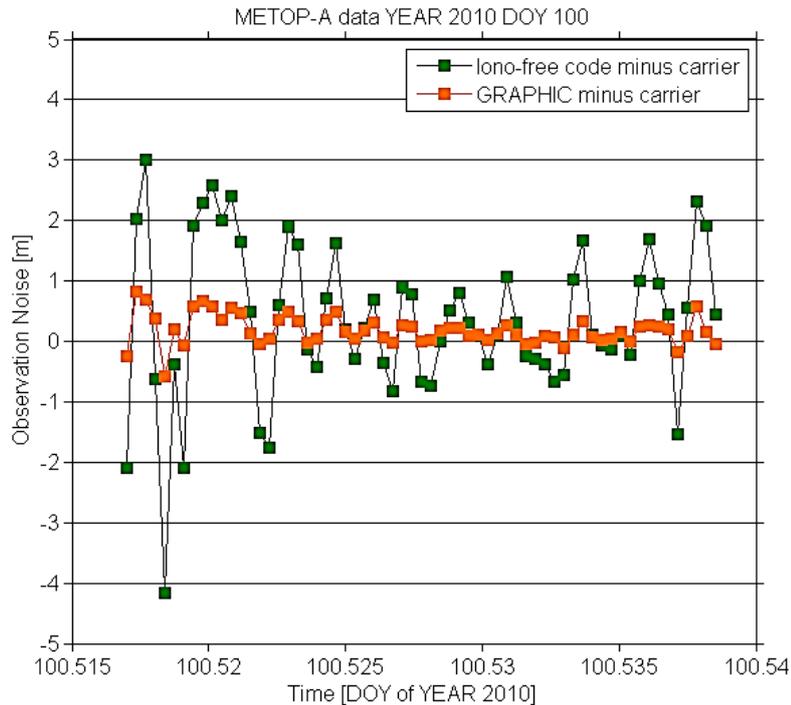
- *Ionospheric free dual frequency carrier phase combination*

$$\rho_{L12} = 2.54\rho_{L1} - 1.54\rho_{L2}$$

- *Single frequency GRAPHIC combination*

$$\rho_{C1L1} = (\rho_{CA} + \rho_{L1})/2$$

The latter combination is known as Group and Phase Ionospheric correction GRAPHIC [14], and according to [25] represents the best choice to attenuate the effect of the ionosphere for single frequency receiver. This method makes use of the fact the carrier phase measurements exhibit a ionospheric error which is equal in size but opposite in sign to the pseudorange delay. Compare to the classic dual frequency combination which amplifies the individual measurements noise by a factor of three (both thermal noise and multipath are summed by the combination), the noise of GRAPHIC combination is only one half of the corresponding pseudorange noise. The Figure 55 displays the noise difference for the ionosphere free code combination and the GRAPHIC combination, obtained by differencing these measurements with the ionosphere free carrier phase combination. The displayed measures have been individually adjusted to remove the bias introduced by the carrier phase measures and are referred to the MetOP GRAS receiver during year 2010.



**Figure 55: Ionosphere free code combination noise compared to GRAPHIC combination noise.**  
MetOP GRAS data of year 2010, DOY 100. GPS PRN number 5.

However, the use of the GRAPHIC combination introduces an additional bias parameter in the computation due to the ambiguity associated with the carrier phase measurements. This bias remains constant from the beginning of the computation, apart from eventual cycle slips that can occur in the phase measurement. Starting from a priori estimate of this bias which is generally computed through code/carrier difference, the actual value can be estimated as part of the filter implementation. The estimated bias is usually not constrained to any value, but they are kept floating during the computation, and tuned by assigning the appropriate process noise. This selection provides the benefit of allowing the biases to absorb any unmodelled measurement error, but does not fully exploit the potential of the very accurate carrier phase measurements. Regardless of the selected measurement combination, a data editing is generally required to discard bad measurements that can occur during the tracking. The simplest technique is based on the comparison between the actual measurements and the predicted

measurements derived by from the propagated state vector. Following a residual test, which takes into account the propagated state uncertainty and the expected measurement noise covariance, the accepted observations are used to update the filter state.

The total filter state vector is composed of

- the instantaneous position and velocity of the spacecraft barycenter (6 elements),
- the empirical accelerations in radial, along track and cross track directions (3 elements),
- the receiver clock offset (1 element),
- for single frequency GRAPHIC combination, 1 bias parameter per channel.

Additionally, correction coefficient for the atmospheric drag and solar radiation pressure can be estimated as part of the algorithm, if the properties of the LEO space vehicle are not known with enough accuracy. The filter itself is an Extended Kalman filter, which combines the a priori information given by the dynamic model with the measurements coming from the GNSS receiver in a suboptimal sequential algorithm. The non-optimality is given by the fact the Kalman filter is designed for linear systems that are only affected by Gaussian noise both in the process and the measurements, while the extension to nonlinear system implies a certain degree of approximation, which in case of EKF is a first order approximation. In fact the computation of the Kalman gain and the covariance matrix associated with the estimation are performed by linearizing the system around the last estimated state point. For weak nonlinear system such as the orbit model with time updates in the order of seconds, the use of EKF is fully justified, and no definitive evidence is given in literature on the eventual benefits coming from different type of filters. Eventual benefits could be related to the fact that the noise that affects the process and the measurements are not ideal (white noise with

zero mean), since for example most of the noise that affects measurements is in form of bias caused by uncorrected ionospheric delays, multipath and residual of GPS orbits and clocks. The filter is initialized from a pure kinematic solution that involves only pseudoranges, which exhibits a representative accuracy of better than 10m and 10cm/s when an adequate number of GPS satellites are tracked. The equation that realizes the Kalman based estimation (the combination between a priori propagation and the measurements) is given by:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-)$$

where  $\hat{x}_k^-$  is the a priori propagated state,  $K_k$  is the Kalman gain,  $z_k$  is the vector of measurements and  $H_k$  is the matrix that translates the a priori state estimation into the related a priori measurement. The provided filter is based on a linear discrete system assumption with additive white Gaussian noise, that can be generally expressed as

$$\begin{aligned} x_{k+1} &= \Phi_k x_k + u_k \\ z_k &= H_k x_k + v_k \end{aligned}$$

where  $u_k$  is the process noise, white Gaussian with zero mean, represented by the variance/covariance matrix  $Q_k$  and  $v_k$  is the measurements noise, white Gaussian with zero mean, represented by the variance/covariance matrix  $R_k$ . The filter is then completed with an equation that provides estimation of variance/covariance matrix associated with the estimation error, given by

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k$$

In case of the orbit de termination filter, both the process equation and the measurements equations are non-linear and the noises are not ideal. They can be generally expressed as

$$\begin{aligned} x_{k+1} &= F(x_k, u_k) \\ z_k &= G(x_k, v_k) \end{aligned}$$

where both  $F$  and  $G$  are non-linear, discrete time domain functions. In order to compute the Kalman gain and the estimation error variance/covariance matrix,

they have to be linearized around a given point (generally the estimation provided at the preceding time step) in order to have the  $\Phi_k$  and the  $H_k$ . On the opposite, in order to compute the a priori propagated state and the a priori measurement, the full non-linear equations can be used. Since the state transition matrix only appears in the estimation of the error variance/covariance matrix, it can be simplified by neglecting some of the perturbations (such as for example the Sun and Moon gravitational effects and the gravity harmonic expansions) and assuming some of the states as constants. The final transition matrix in case of the GRAPHIC combination, is given by,

$$\Phi_k = \begin{pmatrix} \Phi_y & S_a & & & \\ & & mI_{3 \times 3} & & \\ & & & 1 & \\ & & & & I_{n \times n} \end{pmatrix}$$

and in case of single frequency pseudorange only

$$\Phi_k = \begin{pmatrix} \Phi_y & S_a & & \\ & & mI_{3 \times 3} & \\ & & & 1 \end{pmatrix}$$

where  $\Phi_y$  is the transition matrix of position and velocity,  $S_a$  is the sensitivity matrixes of empirical acceleration given the position and velocities,  $m$  is the exponential time correlation of the empirical accelerations and  $n$  is the number of ambiguities. The ambiguities and receiver clock offsets are modeled as constants. While generally the measurements error variance/covariance matrix is known, because we know the order of magnitude of the errors associated with the measurements, the filter can be tuned by acting on the process noise matrix  $Q$ . The tuning of the filter is an important step in the filter design, which is generally performed in simulation. Since the EKF is build up on non-ideal assumptions, it can experience divergence. Only adequate tuning can guarantee correct operations of the filter and can avoid filter divergence.

### 4.3 GNSS broadcasted orbit and clock quality

The positioning accuracy provided by the described real-time on board algorithm is limited by the quality and type of GNSS spacecraft orbits and clocks data that are used in the computation, which are derived from the GNSS broadcasted message. The used of broadcasted information is not the only available solution, since more precise orbits and clocks are available from other sources, which could be uploaded to the LEO spacecraft through the telemetry data in order to improve the quality of the on-board real time estimation. The purpose of this section is to provide an estimate of the accuracy of the broadcasted GNSS satellites orbits and clock, and also to identify possible alternatives as realistic solutions to improve the on-board estimation accuracy. With the help of this overview, the flight hardware and protocols will be designed according to the selected method.

#### 4.3.1 Signal-in-Space range error

The analysis of GNSS error budgets are generally accomplished using two indexes: the *User Range Error (URE)* which accounts for system errors and user related errors such as the multipath, and *Signal-In-Space Range Error (SISRE)* which accounts only for those error that are related to the signal and service structure itself, i.e. clock and ephemeris errors, atmosphere and ionosphere errors. If working with ionosphere free combination, either dual frequency or single frequency code/carrier, the *SISRE* can be expressed as [36] (considering also that for space users the atmosphere error can be neglected)

$$SISRE = \sqrt{\sigma^2(\Delta r_R - \Delta c\delta t) + \frac{1}{7^2}(\sigma^2(\Delta r_T) - \sigma^2(\Delta r_N))}$$

where  $\Delta r_R$ ,  $\Delta r_T$ ,  $\Delta r_N$  are the ephemeris errors component in radial, along track and cross track direction,  $\Delta c\delta t$  is the clock error, and  $1/7 \approx \sin(8^\circ)$  in the along track and normal component accounts for the average angle between the line-of-sight and the radial direction for the observers on the surface of the Earth and its immediate vicinity. These error formulation is able to characterized both the

contribution of the position and clock error, and therefore it appears more suitable for the analysis than looking at separate orbits and clock. The orbit only *SISRE* is given by

$$SISRE_{ORB} = \sqrt{\sigma^2(\Delta r_R) + \frac{1}{72}(\sigma^2(\Delta r_T) - \sigma^2(\Delta r_N))}$$

When comparing GNSS clocks solution coming from different sources, a detrending is generally required to remove biases and drifts in the provided solutions which are common to all satellites. The different trends between solution sources are related to different realization of the GNSS time scale, which affect the user clock estimation but not the position accuracy. With this concepts in mind the following sections provides an overview of present products sources and product characteristics, which are considered to be of interest for their application to the real time on board navigation of LEO satellites.

### 4.3.2 The IGS products

The International GNSS Service (IGS, formerly the International GPS Service) started to provide precise GNSS satellites position and clock in the 1994, when the GPS systems had almost reached its fully operational status. The IGS has developed a worldwide system comprising satellite tracking stations, Data Centers, and Analysis Centers to put high-quality GPS data and data products on line within a day of observations. For example, following the Northridge, California, earthquake in January 1994, analysis teams using IGS-supplied data and products were able to quickly evaluate the disaster's immediate effects by determining station displacements accurately to within a few millimeters. The IGS global network of permanent tracking stations, each equipped with a GPS receiver, generates raw orbit and tracking data. The Operational Data Centers, which are in direct contact with the tracking sites, collect the raw receiver data and format them according to a common standard, using a data format called Receiver Independent Exchange (RINEX). The formatted data are then forwarded to the Regional or Global Data Centers. To reduce electronic network traffic, the

---

Regional Data Centers are used to collect data from several Operational Data Centers before transmitting them to the Global Data Centers. Data not used for global analyses are archived and available for online access at the Regional Data Centers. The Global Data Centers archive and provide on-line access to tracking data and data products. Both the network of IGS ground stations and the quality of the resulting products have continuously increased in the past decade.

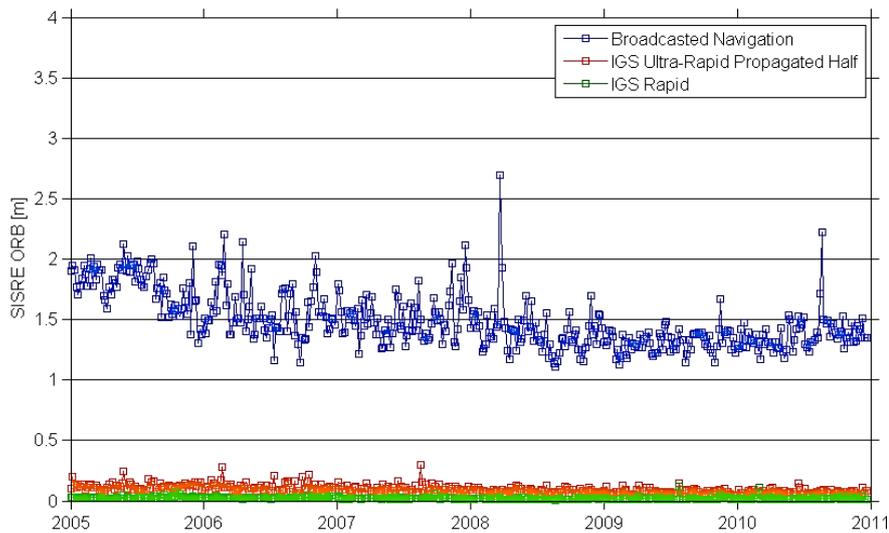
The final IGS product is available between 12 and 18 days after the end of the GPS week, and have a reported position accuracy of about 2.5cm 1D RMS, while the clock accuracy is in the order of 20ps STD or better than 1cm. The rapid products are available within 17 – 41 hours past each day while providing almost the same accuracy. In response to the increasing need of for near real time application to make use of faster delivered products, the IGS proposed also the ultra-rapid products since 2001. The service provides updated ephemeris and clock data every 6 hours (at 03, 09, 15 and 21 UTC) and the information covers a sliding window with 24h of data coming from observation and 24h of data coming from predictions. These 24h plus 24h are centered for the four daily files respectively at 00, 06, 12 and 18 UTC. Therefore the useful predictions after each update covers the interval 03-09, 09-15, 15-21 and 21-03 UTC, as displayed in Figure 57. According to the IGS the observed portion of the orbits are presently accurate in the order of 3cm 1D RMS while the clock is in the order of 50ps STD or better than 2cm (actual RMS value is higher due to the bias that is present in the clock products and is around 1m). The predicted portion of the ultra-rapid orbit is accurate in the order of 5cm 1D RMS while the clock, which is difficult to predict due to its stochastic nature, has similar accuracy than the broadcasted one. All type of IGS product are provided in sp3 file format, and the orbit and clock data are spaced by 15m. This allows accurate polynomial interpolation of the GPS satellite position at the time of the GPS measurement, while the accuracy provided by the linear or quadratic clock interpolation is limited. According to [36] the orbit interpolation error from a 15m grid is negligible, while when interpolating the clock from a 15m grid to 5m or 30sec grid, an additional error is

introduced of about 6cm. The actual error depends on the interval size and on the stability of the clock. Supplementary to the sp3 file, the IGS provides also 5min distanced clock for the stations and the GPS satellites. The following Table 37: IGS Products Summary provides a summary of the mentioned products.

IGS data and products		Accuracy	Latency	Updates	Sample Interval
Broadcast	Orbit	100 cm 1D RMS	real time		daily
	Clock	150 cm RMS 75 cm STD			
Ultra Rapid Predicted	Orbit	5 cm 1D RMS	real time	03, 09,15, 21 UTC	15 min
	Clock	90 cm RMS 45 cm STD			
Ultra Rapid Observed	Orbit	3 cm 1D RMS	3 hours	03, 09,15, 21 UTC	15 min
	Clock	1 cm RMS 2 cm STD			
Rapid	Orbit	2.5 cm 1D RMS	17 – 41 hours	17 UTC daily	15 min
	Clock	2 cm RMS 1 cm STD			15 min or 5 min
Final	Orbit	2.5 cm 1D RMS	12 – 18 days	every Thursday	15 min
	Clock	2 cm RMS 1 cm STD			15 min or 5 min

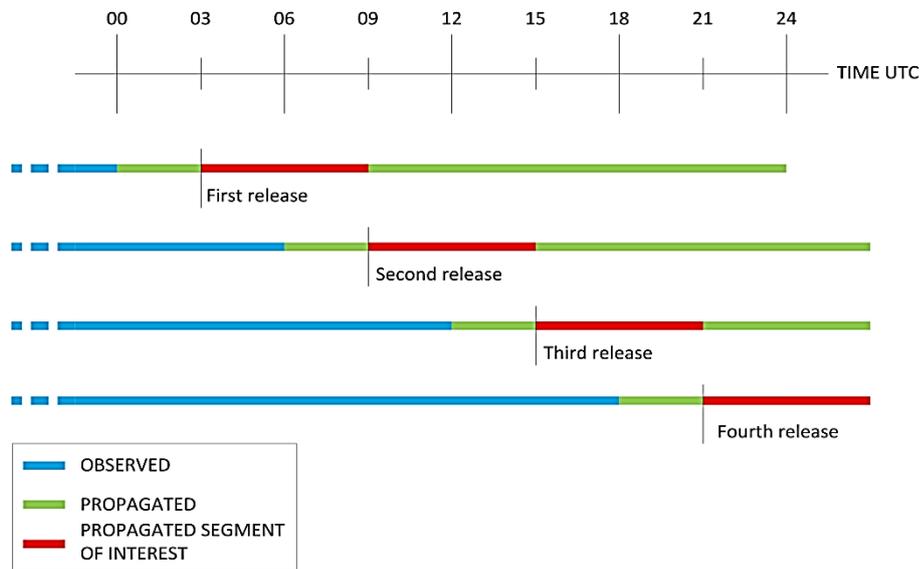
**Table 37: IGS Products Summary**

As part of the present work, an evaluation of the IGS representative products together with the GPS broadcasted ephemeris has been performed, and the resulting SISRE ORB and CLOCK standard deviation (which are more representative than the 1D RMS values provided by the IGS) are displayed in Figure 56.

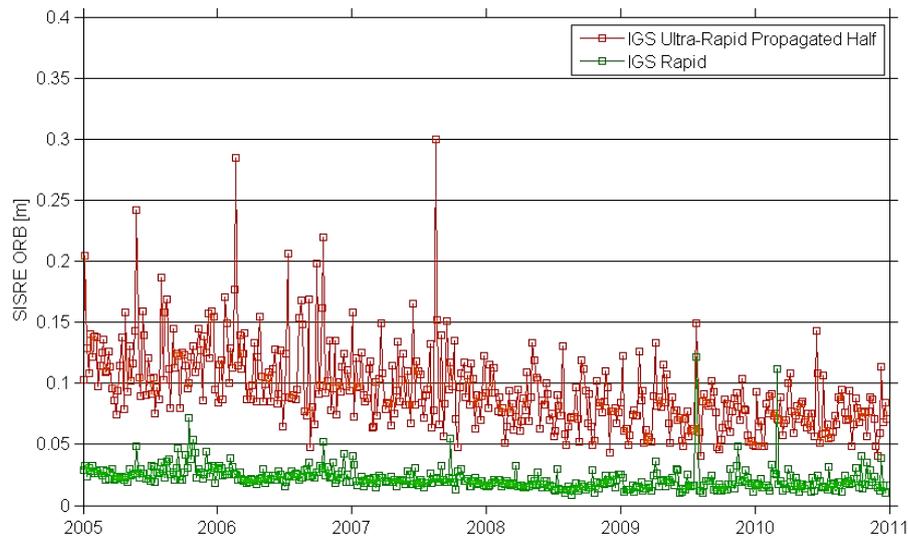


**Figure 56: SISRE ORBIT comparison between Broadcasted Navigation, IGS Ultra Rapid Propagated Half and IGS Rapid evaluated against IGS final**

The Figure 56 shows that the GPS broadcasted SISRE ORBIT, representing the error on the user range given by the spacecraft position errors, is presently around 1.3m (which is compatible with the 100cm 1D RMS presently declared by the IGS) at that it has improved from 2005 of about half a meter. If we look at the IGS products, particular interest for the present study is given by the propagated part of the ultra-rapid product. What is displayed in the Figure 58 is exactly the 6 hour propagation error of interest: the release timeline of this IGS data is further described in Figure 57.



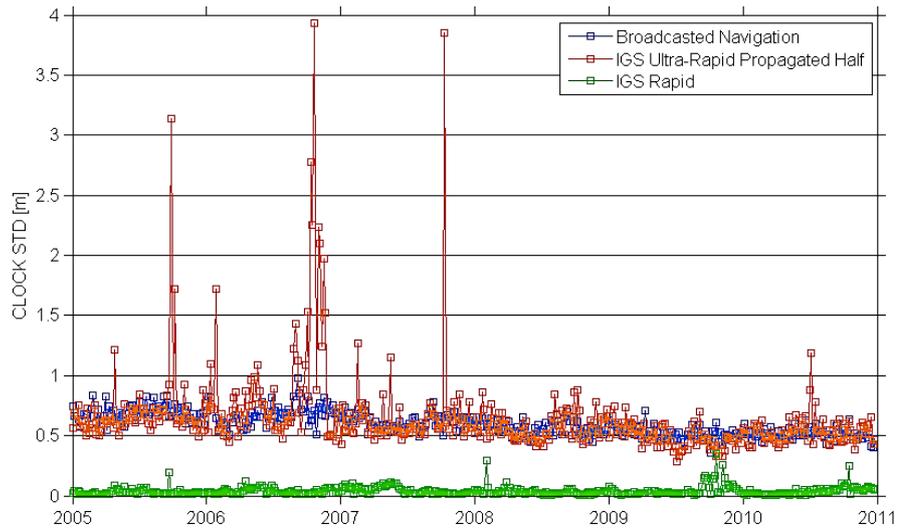
**Figure 57: Release timeline of the IGS ultra rapid product**



**Figure 58: SISRE ORBIT comparison between IGS Ultra Rapid Propagated Half and IGS Rapid evaluated against IGS final**

According to the Figure 58, the SISRE ORBIT of the propagated half of the ultra-rapid product is below 10cm, and has improved from the 2005 of more than 1m. The interest in this product is represented by the fact that the data could be uploaded on a LEO spacecraft 4 time per day, a practice that seems to be perfectly

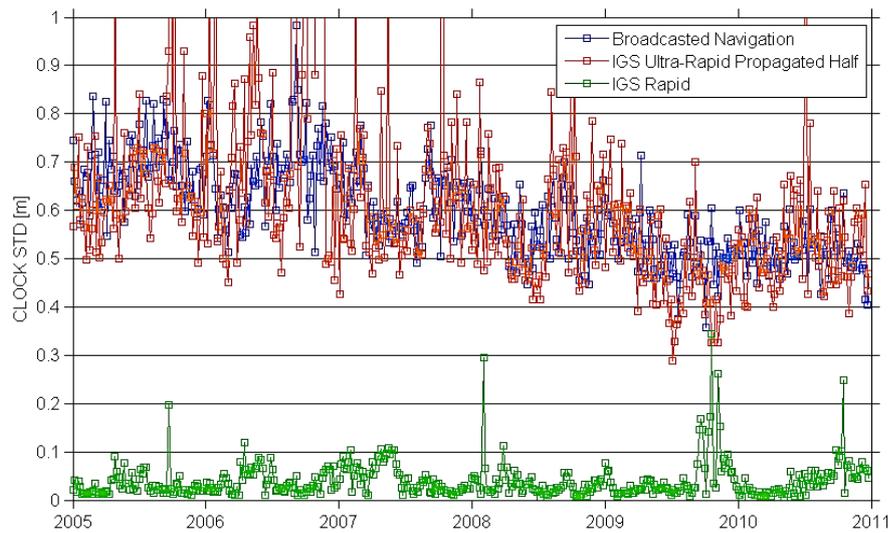
compatible with the present technology without any relevant modification of the communication infrastructure. The same concept does not hold for the clock prediction, since this computation is much less reliable due to the stochastic nature of the clock errors.



**Figure 59: CLOCK STD comparison between Broadcasted Navigation, IGS Ultra Rapid Propagated Half and IGS Rapid evaluated against IGS final**

As shown in Figure 59 and Figure 60 (which display the same data with different scale on the error axis), the propagated clock data error from the IGS ultra rapid product matches exactly the clock error that is available from the GPS broadcasted message, which accuracy is presently around 50cm. Therefore this data would not provide any benefit with respect to the broadcasted message for the on board orbit determination.

The clock error provided by the IGS rapid product is accurate better than 5cm, as displayed in Figure 60, but it is important to notice that frequent outliers are present in the solution, in proportion of around the 5% of the data.



**Figure 60: CLOCK STD comparison between Broadcasted Navigation, IGS Ultra Rapid Propagated Half and IGS Rapid evaluated against IGS final**

From the IGS data analysis it seems clear that the propagated half of the ultra-rapid product could provide position accuracy enhancements in the on board orbit determination due to the improved accuracy of the GNSS spacecraft positions, at the price of four data upload per day on the LEO satellite. For the clock error, the propagated IGS data does not provide any benefit, so the broadcasted information can be used.

#### 4.4 GNSS observables error modeling

This section presents the error sources that are modeled in the system simulator, and are used to generate the GNSS observables (pseudoranges and carrier phases). These observables are then used by the estimation algorithms (least squares or EKF based) to compute the navigation solution, which can be compared with the true position to define the navigation errors. The implemented errors are:

- Code and carrier tracking error
- Ionosphere error
- Multipath error

- GNSS SVs orbits and clock errors

According to the listed errors, the GNSS observables can be modeled as

$$P = \rho + I + c(dt_r - dt^s) + M_p + \varepsilon_P$$

$$L = \rho - I + c(dt_r - dt^s) + \lambda N + \varepsilon_L$$

where  $P$  is the pseudorange and  $L$  the carrier phase,  $\rho$  is the geometric range,  $I$  the ionospheric delay,  $dt^*$  are the receiver and transmitter clock errors,  $M$  is the multipath code delay,  $N$  is the carrier phase integer ambiguity and  $\varepsilon^*$  includes both the thermal error and the residual ephemeris and clock errors. Additional sources of errors, such as antenna phase center variations, phase wind-up, hardware biases, or relativistic effects are not modeled in the simulation.

#### 4.4.1 Code tracking errors

Only the thermal noise error is considered in the simulation, while the DLL steady state errors are neglected by assuming a third order PLL and third order DLL that nulls the error associated with range acceleration. The effect of jerk on the code tracking is then negligible. The code and carrier tracking errors have already been estimated for the employed PLL and DLL, and are reported in section 3.4.1.

#### 4.4.2 Ionosphere error

As reported on [5], around the 10% of the total ionosphere delay is due to the outer ionosphere, so called protonosphere or plasmasphere, a layer that is mainly composed of protons that spans between 1000km and 22000km approximately. The Figure 61 displays the mentioned ionosphere structure.

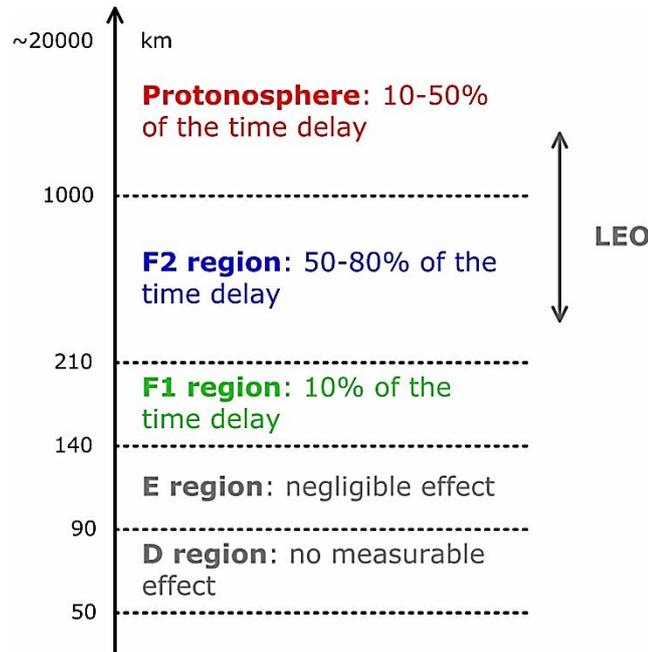


Figure 61: Ionosphere structure diagram

The LEO satellite are positioned from 200 km to 2000 km altitude and they are generally in the middle of the F2 region right above of the TEC peak that is positioned between 300 and 400 km. For this reason, while trying to replicate the error caused by the ionosphere, both the ionosphere itself and the contribution of the plasmasphere is taken into account. The simplest model that can be used to replicate the ionosphere electron density content, is the Chapman profile with a Lear mapping function. The Chapman profile describes the electron density of the topside ionosphere, the F2 region, which provides most of the delay contribution, and reconstructed profiles are generally accurate up to an altitude of about 1000km. The standard 3-parameters Chapman layer electron density as function of altitude is given by:

$$N(z) = NmF2 \cdot \exp\left(\frac{1}{2}\left(1 - \frac{z - z_m}{H} - \exp\left(-\frac{z - z_m}{H}\right)\right)\right)$$

where  $NmF2$  is peak electron density of the  $F2$  region,  $z_m$  is the altitude of the peak and  $H$  is a scale height. By integrating the provided density along the altitude coordinate, the Vertical *Total Electron Content*  $VTEC$  can be estimated, and then

mapped to *Slant Total Electron Content STEC* (the electron content of the line-of-sight signal path) using the Lear mapping function. The mapping function is given by:

$$m(E) = \frac{2.037}{\sqrt{\sin^2 E + 0.076} + \sin E}$$

where  $E$  is the GNSS satellite elevation over the local horizon. Once the *STEC* has been determined, the ionospheric delay for the L1 frequency is given by

$$I(E) = 0.162m \cdot \frac{STEC}{TECU}$$

where *TECU* stands for *Total Electron Content Unit* and is equal to  $10^{16} \text{ e}^-/\text{m}^2$ . The default values applied to simulate the electron density are summarized in Table 38.

Parameter	Value
$NmF2$	$1e12 \text{ m}^{-3}$
$z_m$	350 km
$H$	110 km

**Table 38: Ionosphere model parameters**

The listed parameters corresponds to about 5 m vertical delay on ground at L1 frequency, which decreases to 10 TECU at 450 km altitude (approximately 1.6 m of vertical delay). Then the contribution of the plasmasphere is added to the modeled upper ionosphere, in a percentage that is configurable by the user with a default value of 10% of the ground vertical TEC (the model is therefore consistent with the study presented in [37], even if strongly simplified). Given the parameters, Figure 62 displays modeled the VTEC as function of LEO satellite altitude.

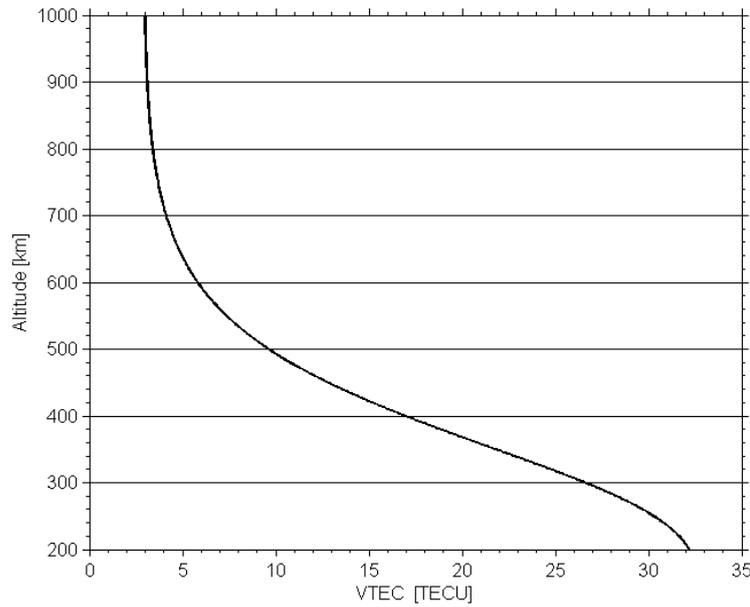


Figure 62: VTEC profile as function of altitude

### 4.4.3 Multipath errors

The implemented multipath error is produced by a combination of a time correlated noise process and exponential model as function of the elevation, which is tuned on the basis of real data available in literature from GNSS receiver flight experiences. The model itself is:

$$M(E) = w \cdot e^{-\alpha(E-E_0)}$$

The parameters are fixed as follows:

- Noise standard deviation 0.3 m
- Noise time correlation 20 s
- Exponential decay with elevation 0.03 m/deg
- Reference zero elevation 10 deg

The model is simplified so is only function of the elevation and not of the signal azimuth, and is considered the same for all the signals since for such short delays, as expected on a spacecraft with dimension of a few meters, the multipath envelope is similar for all the modulations. With this model, the simulated

multipath delay is displayed in Figure 63 in comparison to real multipath measured using GRAS receiver data on board Metop-A. It is displayed that the simulated multipath reproduces the real phenomenon in a realistic way, but without any predominant azimuth direction. Figure 64 displays also the multipath realization as function of the elevation.

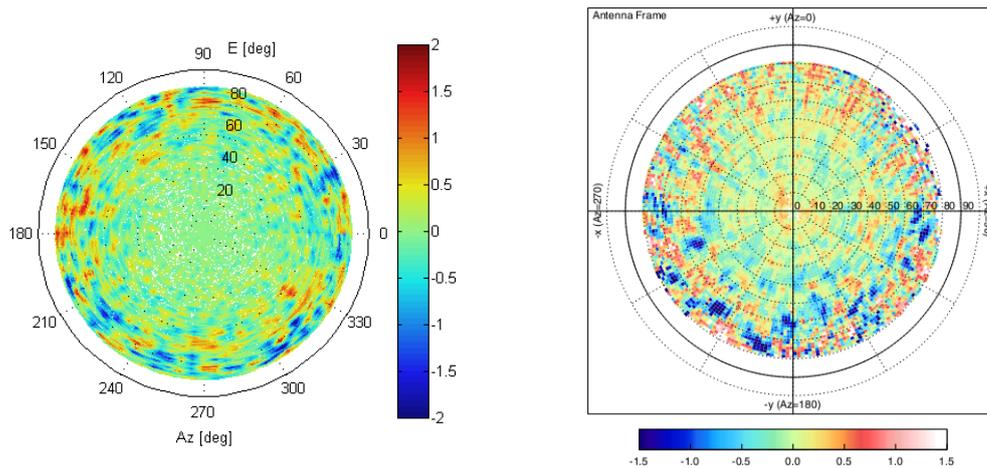


Figure 63: Simulated code multipath (left) and code multipath from MetOP-A (right) reproduced from [10].

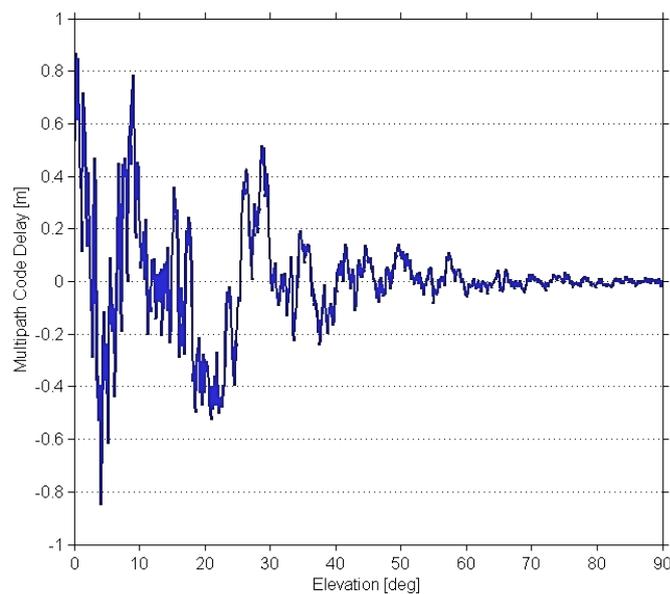


Figure 64: Multipath realization as function of the signal elevation

#### 4.4.4 Orbit and clock errors

The last source of errors comes from the broadcasted orbit and clock information residuals that affect the measurements. For the GPS system the residuals comes from actual measurements already described in section 4.3, while for the GALILEO the residual are as specified by the system. For the GLONASS and COMPASS it is assumed that the residual are the same as the GPS system. The default values are summarized in the Table 39.

System	Parameter	Value
GPS	SISRE ORBIT	1.2 m
	3D RMS ORBIT	1.7 m
	CLOCK STD	0.5 m
GALILEO	SISRE ORB	0.2 m
	3D RMS ORBIT	0.3 m
	CLOCK STD	0.3 m
GLONASS	SISRE ORB	1.2 m
	3D RMS ORBIT	1.7 m
	CLOCK STD	0.5 m
COMPASS	SISRE ORB	1.2 m
	3D RMS ORBIT	1.7 m
	CLOCK STD	0.5 m

Table 39: Broadcasted residual error

## 4.5 Expected navigation performances

Having defined the structure of the navigation algorithm, which estimates user position, velocity and clock bias respect to the GNSS time, and having also implemented realistic observables sources of errors, a series of simulations have been performed in order to accurately tune the filter, and finally to determine the accuracy of the estimation.

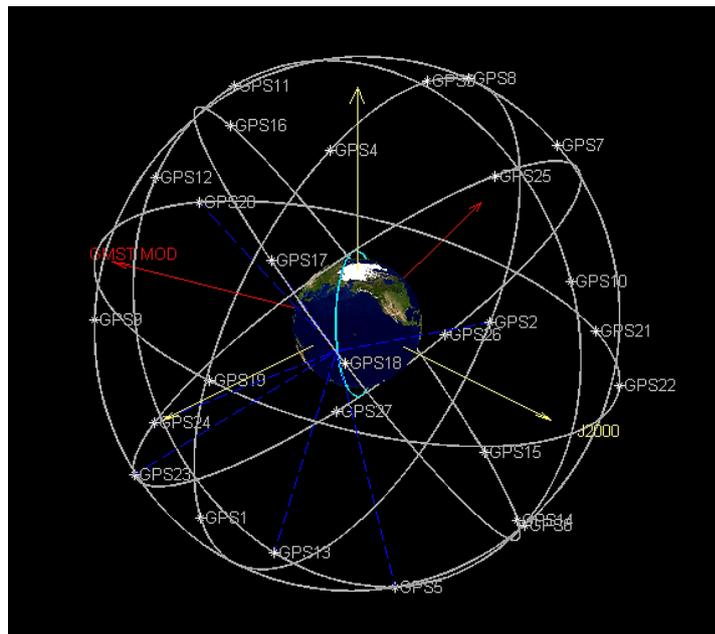


Figure 65: Simulated Navigation Scenario

The purpose was to validate the navigation algorithm architecture, and have a preliminary estimation of the receiver performances based only on simulation, while in order to measure the real receiver navigation accuracy, an RF signal simulator is required. Even if the prototyped GNSS receiver is compatible with only GPS and GALILEO, navigation solution has been computed also taking into account one or two additional systems, in order to verify their contribution to the final estimation. The Figure 65 displays the radial position RMS accuracy for the pseudorange only solution, given different combination of observables. Radial statistic is provided since it is the most noisy position component.

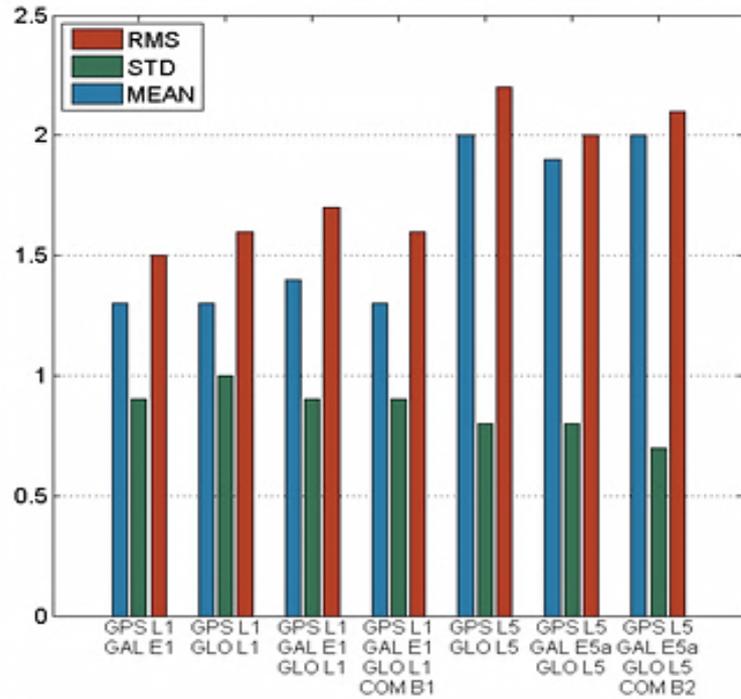
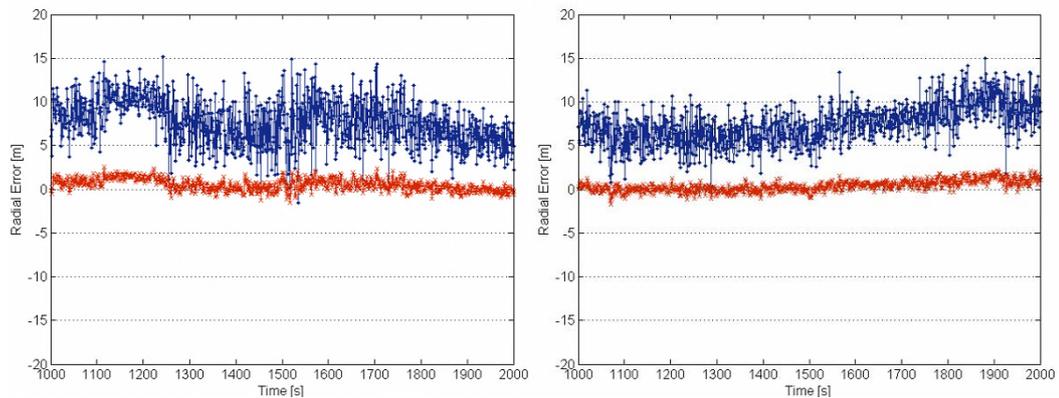
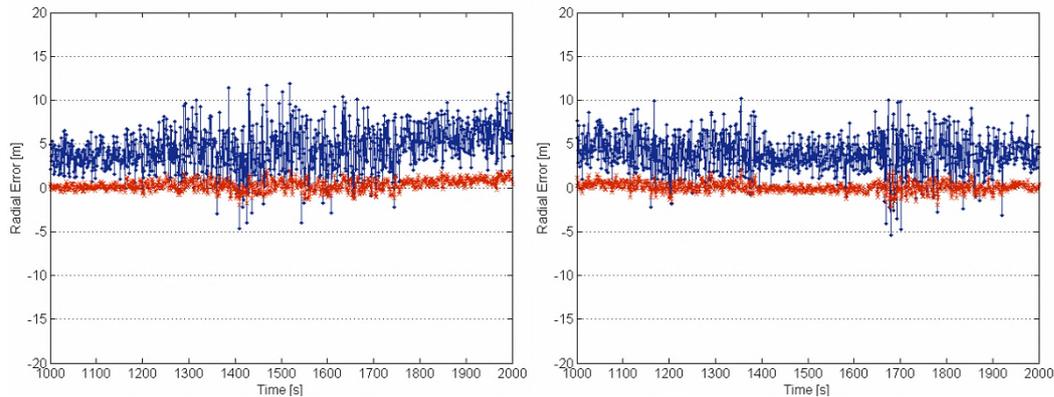


Figure 66: Navigation radial component accuracy, for different observables combinations, form EKF pseudorange only solution.

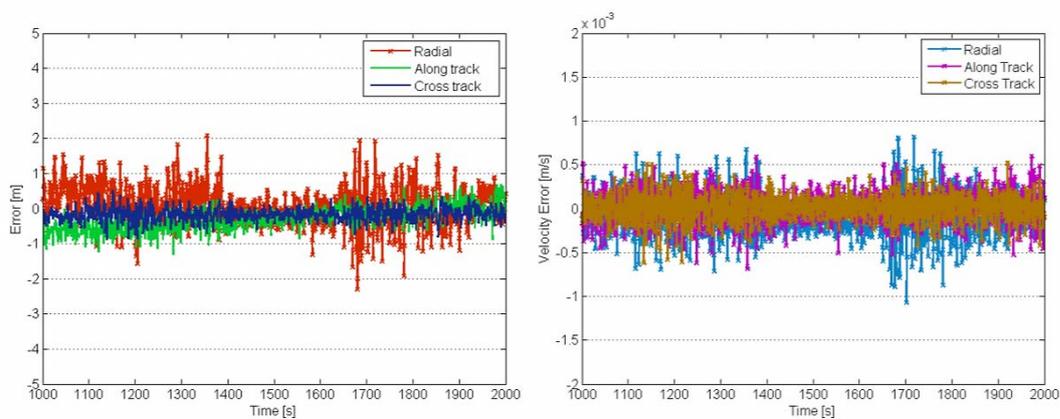
A time series realization of the estimated position error, is given in Figure 67 as an example, for the GPS and GALILEO L1/E1 observables. Similar time series have been obtained for the other observables. Figure 67 compare the EKF based solution with *pure kinematic* solution error, obtained without any orbit dynamical model constrain.





**Figure 67: Radial position error time series, form different simulations. Comparison between pure kinematic solution (blue) and EKF based solution (red)**

As shows, the pure kinematic solution obtained with an un-constrained algorithm, is heavily biased respect to the EKF solution, due to the nature of the GNSS measurements, which are mostly affected by noise in form of bias rather than zero-mean noise. Is displayed also that the EKF based navigation algorithm is required to comply with the LEO mission requirement, which requires positioning error below 10m. Figure 65 displays also what has been already concluded in section 2: including additional GNSS systems and observables provides negligible accuracy improvements with respect to the selected GPS and GALILEO configuration. Figure 68 displays the positioning error components in a time series realization, and the estimated velocity errors. Finally, Table 40 summarizes the estimation errors obtained with EKF based filter.



**Figure 68: Positioning error components (right) and velocity error components (left) time series**

---

Position [m]	Estimation Method	
	<i>Nonlinear least square</i>	<i>EKF Reduced Dynamics</i>
Radial Mean	4.5	1.3
Along Track Mean	0.5	0.0
Cross Track Mean	0.7	0.2
Radial Std.	3.3	0.9
Along Track Std.	1.4	0.7
Cross Track Std.	0.7	0.2
3D RMS (1-sigma)	5.9	1.8
3D RMS (3-sigma)	10.9	3.7
Time Bias Std.	2.3	1.5

**Table 40: Preliminary navigation accuracy evaluation**

## 5 Conclusions

The thesis has proposed the preliminary design and prototyping of an FPGA based GNSS receiver for space applications, covering all the aspects that are related to this subsystem design. The starting point has been the mission analysis performed through the development of a dedicated software simulator. Then the covered aspects have been

- The electronic and software receiver architecture.
- The signal acquisition and tracking algorithms.
- The navigation algorithms.

Each point and related results are summarized in following section. Then final considerations and project follows up are given.

### 5.1 Receiver electronic and software architecture

The proposed receiver architecture is based on the software defined radio paradigm: most of signal processing is performed in software while only what is strictly necessary is done by hardware. Specifically, an FPGA has been employed to perform GNSS signal correlation, which is an heavy computational task to be performed in software. The receiver has been prototyped, and the VHDL code that performs signal correlation has been implemented, together with the software drivers. Few minor bugs have been identified, mostly related to wrong footprints of various components, that will be solved for the flight model fabrication. The L2 front-end will also be removed for the flight model since it is not necessary to achieve the required positioning accuracy, The FPGA-PowerPC architecture with the support of the Linux OS has also been validated, and the Linux device drivers that interface the correlators to the software have been implemented: correct interrupt handling and register access have been verified with hardware testing. The hardware still requires a validation with a GNSS signal generator, in order to

test the tracking and navigation algorithms. Moreover the selected FPGA model is almost fully routed for the project implementation, which generally has to be avoided because of inefficient logic resources placement. Therefore a larger model should be selected from the same family, possibly pin-to-pin compatible.

## **5.2 Acquisition and tracking algorithm**

Acquisition and tracking algorithms have been studied starting from the tracking loop theoretical background: most of the ground applications make use of second order tracking loops, since generally the dynamic is not so relevant to justify more complexity in the design. Therefore, as part of the thesis, a design procedure for third order tracking loop has been consolidated, starting from the continuous time domain description, up to the time discrete time domain implementable equations and physical meaning of parameters. The proposed tracking architecture is a combination of second order FLL and third order PLL for the signal carrier, and a third order DLL for the spreading code. The signal processing from software side has been implemented as a state-machine with acquisition, confirmation, pull-in and lock state, in order to make signal tracking more reliable against variable channel condition. The designed algorithms have been verified against space conditions using a simulations tool, and on real signal coming from a ground receiver. As for the hardware, they still require validation with an RF signal generator.

## **5.3 Navigation algorithm**

The navigation algorithm has been designed in order to achieve the required positioning accuracy: simple pure kinematic solution was not sufficient, so a dynamic based filtering has been implemented. The implemented filter has been tested against different combination of observables, in order to verify the contribution of future constellation to the accuracy of on board positioning. It has been found that the mentioned contribution is negligible, and does not justify the increased complexity of a receiver compatible with all the systems. Moreover,

---

other than the broadcasted GNSS orbits and clock, which will be used as the default solution, other products accuracy has been investigated and a possible strategy for their used on board identified. The navigation algorithm architecture has been validated in simulation, and the provided accuracy preliminary evaluated. Having clarified which is the required architecture among many possibilities and many levels of complexity, the next step is to consolidate the performance evaluation with a number of analysis, in a Monte-Carlo framework. This analysis will be performed in the following project stage, together with the design of the flight hardware.

## 5.4 Final considerations

The FPGA based GNSS receiver for on board orbit determination has been prototyped, and the related algorithms architecture verified. As side products of the design process, analysis tool for the mission, the tracking performances and the navigation performances have been implemented. The various stages of the work have been published with the following papers:

- Avanzi, P. Tortora, “Design and implementation of a new spaceborne FPGA-based dual frequency GPS and Galileo software defined receiver” NAVITEC 2010, ESTEC, Noordwijk, The Netherlands.
- Graziani, A. Avanzi, P. Tortora, “Results from the scenario simulation of the radio occultation experiment on board ALMASat-EO mission” NAVITEC 2010, ESTEC, Noordwijk, The Netherlands.
- Avanzi, P. Tortora, “Design and development of a GNSS spaceborne receiver for Earth Observation satellites orbit determination”, 8th IAA Symposium on Small Satellites for Earth Observation, April 04 - 08, 2011, Berlin, Germany. *Awarded for best presentation.*
- Avanzi, P. Tortora, A. Garcia-Rodriguez “Design and implementation of a novel multi-constellation FPGA-based dual frequency GNSS receiver for space applications” ION GNSS 2011, 20-23 September 2011, Oregon Convention Center, Portland, Oregon (USA).

As already mentioned, the presented projects constituted the preliminary design of the ALMASat-EO GNSS subsystem: hardware design refinement together with a more detailed tuning of the navigation algorithm shall be performed to manufacture the flight hardware.

---

## Bibliography

- [1] Montenbruck O., Garcia-Fernandez M., Williams J.; Performance Comparison of SemiCodeless GPS Receivers for LEO Satellites; GPS Solutions (2006). DOI 10.1007/s10291-006-0025-9.
- [2] Montenbruck et al. “GPS Orbit Determination for Micro-Satellites – The PROBA-2 Flight Experience” AIAA Guidance, Navigation, and Control Conference 2 - 5 August 2010, Toronto, Ontario Canada.
- [3] A Graziani, N Melega, P Tortora, “A Low-Cost Microsatellite Platform for Multispectral Earth Observation,” in Small Satellites For Earth Observation: Selected Contributions, Springer 2008, ISBN 978-1-4020-6942-0.
- [4] Elliot D. Kaplan “Understanding GPS Principles and Applications”, Artech House Mobile Communications.
- [5] Bradford W. Parkinson, James J. Spilker Jr. “Global positioning system: theory and applications, Volume 1”
- [6] JVP Gisbert “Interference assessment using up to date public information of operating and under development GNSS system” Fourth European Workshop GNSS Signals 2009
- [7] PC. Chang et al. “Allan variance estimated by phase noise measurements”, 36<sup>th</sup> annual Precise Time and Time Interval Meeting, Dec. 2004.
- [8] D. Borio “A Statistical Theory of GNSS signal Acquisition”, PhD Thesis, Politecnico di Torino
- [9] F. M. Gardner, “Phaselock Techniques, third edition” John Wiley and Sons, 2005.
- [10] Montenbruck et al. “Tracking and orbit determination performance of the GRAS instrument on MetOp-A” GPS Solution (2008) 12:289–299.
- [11] A. Avanzi, “FPGA based GNSS receiver: tracking system analysis and open source code review”, MsC Thesis, Università di Bologna
- [12] Potti et al. “An Autonomous GNSS-based Orbit Determination System for Low-Earth Observation Satellites”, ION GPS 1995, September 12 - 15, 1995 Palm Springs, CA.
- [13] Hart et al. “Global Positioning System (GPS) Enhanced Orbit Determination Experiment (GEODE) on the Small Satellite Technology Initiative (SSTI) Lewis Spacecraft”, ION GPS 1996 September 17 - 20, 1996 Kansas City, MO.
- [14] Yunck T.P. “Orbit Determination”, in B.W. Parkinson and J.J. Spilker Jr. (eds.), *Global Positioning System: Theory and Applications*, Vol. II, Ch. 21 (AIAA, Washington, D.C., 1996.
- [15] Hass et al. “Real-Time High Accuracy GPS Onboard Orbit Determination For Use On Remote Sensing Satellites”, ION GPS '99, 14-17 September 1999, Nashville, TN.

- 
- [16] Goldstein et al. “Real-Time, Autonomous, Precise Orbit Determination Using GPS”, NAVIGATION: Journal of The Institute of Navigation Vol. 48, No. 3, Fall 2001 Printed in the U.S.A.
- [17] Reichert et al. “Toward Decimeter-Level Real-Time Orbit Determination: A Demonstration Using the SAC-C and CHAMP Spacecraft”, ION GPS 2002 September 24 - 27, 2002 Oregon Convention Center Portland, OR.
- [18] Montenbruck et al. “Reduced dynamic orbit determination using GPS code and carrier measurements”. Aerospace Science and Technology 9 (2005) 261–271.
- [19] S.C. Wu, T.P. Yunck, C.L. Thornton, “Reduced-dynamic technique for precise orbit determination of low Earth satellites” J. Guidance Control Dynam. 14 (1) (1991) 24–30.
- [20] Wermuth M., Montenbruck O., van Helleputte T. “GPS High Precision Orbit Determination Software Tools (GHOST)”, 4th International Conference on Astrodynamics Tools and Techniques; 3-6 May 2010, Madrid (2010).
- [21] <http://igsceb.jpl.nasa.gov/>
- [22] [http://www.aiub.unibe.ch/content/research/gnss/code\\_\\_\\_research/index\\_eng.html](http://www.aiub.unibe.ch/content/research/gnss/code___research/index_eng.html)
- [23] Montenbruck et al. “Precision real-time navigation of LEO satellites using global positioning system measurements” GPS Solutions (2008) 12:187–198.
- [24] <http://www.gdgps.net/>
- [25] Bock H. et al. “GPS single-frequency orbit determination for low Earth orbiting satellites” Advances in Space Research 43 (2009) 783–791.
- [26] Montenbruck et al. “Antenna phase center calibration for precise positioning of LEO satellites” GPS Solutions (2009) 13:23–34.
- [27] Jong-Yeoun Choi, Sang-Jeong Lee “Precision Assessment of Near Real Time Precise Orbit Determination for Low Earth Orbiter” J. Astron. Space Sci. 28(1), 55-62 (2011).
- [28] Montenbruck et al. “Onboard Real-Time Navigation for the Sentinel-3 Mission” ION-GNSS-2009, 22 Sep. -25 Sep. 2009, Savannah, USA (2009).
- [29] Montenbruck et al. “GPS Orbit Determination for Micro-Satellites – The PROBA-2 Flight Experience” AIAA Guidance, Navigation, and Control Conference 2 - 5 August 2010, Toronto, Ontario Canada.
- [30] <http://www.iers.org/>
- [31] Montenbruck O., Gill E. “Satellite Orbits, Models Methods Applications” Springer 2000.
- [32] [http://www.iers.org/nm\\_11216/IERS/EN/Publications/TechnicalNotes/tn36.html](http://www.iers.org/nm_11216/IERS/EN/Publications/TechnicalNotes/tn36.html)
- [33] Cheng, M., J. C. Ries, and B. D. Tapley, “Variations of the Earth's figure axis from satellite laser ranging and GRACE” , J. Geophys. Res., 116, B01409, doi:10.1029/2010JB000850.
- [34] <http://www.csr.utexas.edu/grace/>

- [35] Rizos, C. and Stolz, A. (1985). "Force modelling for GPS satellites orbits." 1<sup>st</sup> International Symposium on. Precise Positioning with GPS, Rockville, Maryland, United States of America, 1, 87-98.
- [36] Montenbruck O. Gill E. Kroes R. "Rapid orbit determination of LEO satellites using IGS clock and ephemeris products" GPS Solutions (2005) 9: 226–235 DOI 10.1007/s10291-005-0131-0.
- [37] E. Yizengaw, M.B. Moldwin, D. Galvan, B.A. Iijima, A. Komjathy, A.J. Mannucci "Global plasmaspheric TEC and its relative contribution to GPS TEC" Journal of Atmospheric and Solar-Terrestrial Physics 70 (2008) 1541– 1548.