POLITECNICO
DI TORINO

SCUOLA DI DOTTORATO

Dottorato in Ingegneria elettronica e delle comunicazioni – XXVII ciclo

Tesi di Dottorato

Snapshot Estimation Algorithms for GNSS Mass-Market Receivers

Nicola Linty

Tutore
prof. Letizia Lo Presti

Coordinatore del corso di dottorato
prof. Ivo Montrosset

17 aprile 2015

Summary

This thesis resumes the PhD program carried out in the signal processing, satellite positioning and telecommunication fields, within the Navigation, Signal Analysis and Simulation (NavSAS) group, Department of Electronics and Telecommunications (DET) of Politecnico di Torino, in the period going from January 2012 to December 2014. The main topic of the PhD activity is represented by Global Navigation Satellite System (GNSS) receivers core technologies. In particular, it deals with the design, development, test and performance assessment of innovative architectures, techniques, and algorithms for Global Positioning System (GPS) and Galileo receivers, both professional high-performance and commercial mass-market.

GPS, and in general GNSSs are radio-communication infrastructures, aimed to enable a generic user to compute Position, Velocity and Time (PVT). The signals transmitted by a constellation of satellites are processed by an electronic device, performing trilateration with respect to the satellites, taken as reference points. At least 4 satellites are required to be in Line of Sight (LOS) with the receiver, so as to obtain 4 different signals and to solve the 4 navigation unknowns: *latitude*, *longitude*, *height* and *time*.

Since their first appearance, in the early seventies, GNSS chipsets and devices are gaining a fundamental role in most applications of everyday life, and their global market continues to grow rapidly. In 25 years, GNSS receivers became extremely used worldwide, not only for positioning and navigation purposes, but also for time synchronization, thus spanning an unlimited range of applications, from commercial to scientific, from military to recreational. GNSS mass-market receivers are extremely widespread, produced in very high volume—hundreds of millions just for smartphones and tablets—and sold at a limited price. This variety of applications and possibilities represents the main reason of the continuous growth of the GNSS field: in fact, new systems are emerging beside GPS, such as GLONASS, currently operational and in expansion, Galileo and Beidou. With the latest trends of multi-constellation receivers, the positioning accuracy can greatly improve, as well as its robustness, availability, reliability, but at the expense of a greater complexity and power consumption.

Motivations of the work

The motivations of such a work originate from three different considerations, coming directly from the three different players of the GNSS market.

From one side, the so called **market pull**. Location Based Applications (LBAs) are more and more widespread on the market. Access to position, as well as location and navigation capabilities, are features constantly required by the users, in any situation, from work to entertainment, sport, travel. The strategic role of navigation in the Information and Communication Technology (ICT) and in the smart and connected world we are entering is evident. For all these reasons, a pull from the market, especially the consumer market, can be identified.

Secondly, the **agencies push**. A big effort has been recently made to improve existing systems

(GPS and Glonass modernization) and to design and develop new systems (among all Galileo and Beidou). Now time is mature enough for the agencies that supported the growth of the multi-constellation system to push its massive use. First in designing, prototyping and producing new, small, cheap, powerful, innovative and multi-constellation GNSS receivers, then in using them in everyday life. The effort of European Space Agency (ESA) in promoting Galileo among industries, research institutes and universities is just an example of this. At the same time, the improvements in electronics and miniaturization are important factors contributing to the development of GNSS-related navigation technologies and infrastructures.

As a final point, the increasing **interest** and curiosity exhibited by universities and by research centres toward innovation in the navigation sector. A lot of initiatives, events, conferences, seminars, prices, competitions and trainings are organized every year, encouraging innovation and development of new ideas and applications. There is a growing interest in the study of new architectures and techniques, in looking for new solutions and in exiting from standard paradigms.

Summing up, it can be stated that navigation receivers technologies are of paramount importance in these years. In particular, a category of receivers gathers together all the three components: *mass-market receivers*. Mass-market devices consist of all the GNSS receivers designed for recreational, automotive, social and mobile applications, characterized by low cost, low size and limited accuracy. As will be clearly described in the following, mass-market receivers feature different requirements with respect to standard receivers described in literature; indeed, the requirement placed by the market pull. Mass-market receivers are driven by different requisites, and can exploit the potentialities of new signals and constellations, as claimed by the international and national agencies. Last but not least for the economy of this work, mass-market receivers implement new and particular architectures, algorithms and techniques, which are of great interest from a scientific and engineering point of view. The majority of them can be grouped under the big outfit of *snapshot estimation algorithms*. This justifies and motivates the study, analysis, implementation and test of these kind of techniques, as will be described throughout this thesis.

Thesis outline

More in details, the thesis deals with the analysis, the demonstration and the development of the main architectures and algorithms currently used in mass-market GNSS receivers, i.e. snapshot estimation techniques. The thesis is organized as follows.

- In the first chapter, a general overview on navigation technologies is given. GNSS fundamentals are briefly presented, mainly focusing on GPS and on Galileo systems. Then a non-exhaustive list of GNSS applications is presented.

- The second chapter is devoted to the description of the signal and system model, so as to define a common notation and mathematical background. The GPS L1 and Galileo E1 Open Service (OS) signals are introduced, through a description of their characteristics, components and frequency plan. Code-based measurements and pseudo-range are introduced, and then the structure of a GNSS receiver is presented.

- The strategic role and the huge market share of mass-market receivers are outlined in the third chapter, by describing the differences with professional receivers, mainly on the architectural point of view. Three mass-market drivers are then defined, along with the concept of GNSS assistance. To conclude, a few interesting operating scenarios, used during tests and simulations, are described.

- In the fourth chapter mass-market receivers state-of-the-art techniques are presented, analysed, demonstrated, extended, improved and innovated. A literature survey shows that the majority of the estimation techniques are based on multi-correlation snapshot processing algorithms, characterized by open-loops and a reduced code and frequency search space. Open-loop structures, multi-correlation code delay and Doppler frequency estimation are deeply investigated and extended to the Galileo OS case. Then particular focus is given to am innovative Double Fast Fourier Transform (DFFT)-based frequency estimation technique and to the development of power saving solutions based on "on demand" processing.

- In the fifth chapter, the suitability of the techniques presented in the fourth chapter is demonstrated by implementation, tests and comparisons, using a fully software receiver and performing Monte Carlo simulations. Some results concerning the performance of the multi-correlation unit, the accuracy of the code delay and of the Doppler frequency estimators, the search space windowing and the on demand duty cycle processing are reported and discussed. Afterwards, a few results obtained by processing real Galileo signals with the software receiver are detailed. Finally, a real commercial Galileo-ready mass-market receiver is described along with results and tests.

- The sixth chapter introduces carrier phase measurements, which represent an evolution with respect to code measurements considered in the previous chapters, but are not yet exploited in consumer devices. The benefits and disadvantages of this approach are presented. Then, the problem of cycle slips is analysed, and some simulation results, concerning artificial cycle slips generation, are presented.

- The seventh chapter reports the proof of the maximum Doppler and maximum Doppler rate for GPS and Galileo constellations.

- The eight chapter deals with ethical aspects related to GNSSs.

- Finally, in the appendix, the description of the different hardware and software tools exploited is reported.

Acknowledgements

Part of research and tests campaigns related to this work have been fulfilled in the frame of the European Space Agency (ESA) Education PRESTIGE programme, thanks to the facilities provided by the ESA TEC-ETN section, at the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands. A great thanks goes to all the extraordinary people met in ESA, for welcoming me and for supporting my work. A special thanks to my supervisor Dr. Paolo Crosta for his willingness and helpfulness during all the internship.

My most sincere thanks go to my supervisor Letizia Lo Presti, for her passion for science, technology, engineering, for her enthusiasm towards research and for transmitting all these values to me.

I also would like to express my gratitude to the members of the Navigation, Signal Analysis and Simulation (NavSAS) group who, in various ways, contributed to all the research activities presented in this thesis. I honestly understand the value of your time and the quality of your support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction: Overview on Navigation Technologies

Global Navigation Satellite Systems (GNSSs), such as Global Positioning System (GPS), Glonass or Galileo, are communication satellite systems enabling a generic user located anywhere on the Earth to determine in real time his Position, Velocity and Time (PVT), by means of the Radio Frequency (RF) electro-magnetic signals transmitted by a constellation of satellites. Despite being originally developed for localization, GNSSs are not limited to positioning purposes, but span an unlimited range of applications, from commercial to scientific, from military to recreational. The underlying idea behind satellite navigation is quite simple: some very precise oscillators are orbiting around Earth; their exact positions and their times are broadcast through the Signal-In-Space (SIS). Anybody, equipped with a proper receiver, can receive these signals, estimate their Time Of Arrival (TOA), combine all the information and compute the position on a Earth centred reference system.

This chapter gives a brief and basic overview on navigation technologies. A reader confident with the topic may want to skip this part and jump to the following chapters. In the first section the fundamental concepts of GNSSs are presented. Then, a brief summary of the different GNSS applications is provided. Section 1.3 gives some results, obtained during the PhD, concerning the first Galileo satellites transmission, representing a milestone in the European GNSS program. Finally, the motivations of the work are described.

1.1 GNSS fundamentals

In this section the fundamental concepts of GNSSs are presented, first giving a short historical background, then describing the positioning algorithm exploited by a satellite system. A quick summary of the different global and regional systems is provided, along with the definition of multi-constellation GNSS.

1.1.1 Short history of navigation

Satellite navigation is indeed a high-tech replication of a very ancient tradition and expertise. Our ancestors were experts in looking to the sky to navigate and to mark out time [1]. For example, Phoenicians relied on the few thousands of stars visible to naked eye to traverse the Mediterranean sea. On the other side of the globe, Polynesians exploited the same celestial markers to navigate across thousands of kilometres of open Pacific ocean.

Then, along the 13th century, navigators became less dependent on the sky, thanks to some technology developments. For example the compass, allowing to find directions in the presence of clouds and thus enabling maritime trade all over the year, was introduced also in western countries. Meanwhile, old astronomy instruments evolved into the more practical sextant. North/south latitude could be easily and faithfully reckoned by measuring the angle of prominent stars. On the contrary, west/east longitude estimation was more difficult, since it corresponds to the Earth rotation direction. Since the stars move sideways at a rate of 15 degrees per hour, any timing drift causes large positioning errors, rendering long-distance navigation a hit-and-miss affair.

This problem was considered so intractable and so important that the British Parliament established the famous "Longitude prize" through an Act of Parliament, *the Longitude Act.* 20,000 £ were up for grabs for a method that could determine the longitude within 30 nautical miles (56 km) (equivalent to more than 3 million Euros in 2014). Nevertheless, it was only in 1765 that the self-taught inventor John Harrison crafted the first reliable *marine chronometer*, losing less than 5 seconds every ten days, winning the prize.

New radio ranging systems started to be developed during the Second World War exploiting quartz clocks. Recently, atomic clocks proved to be far more accurate, drifting only a matter of seconds per year and allowing the development of the first global satellite radio-navigation systems.

1.1.2 The positioning algorithm

Uninterrupted Positioning, Navigation, and Timing (PNT) solution is obtained by a GNSS receiver by continuously processing the SIS from the satellites in view. Any navigation solution provided by a GNSS receiver is based on the computation of its distance to a set of satellites, by means of the determination of the propagation time (τ) of the incoming signals travelling at the speed of light, according to the satellite and receiver local clocks. This time difference is transformed into a range information, ρ, by multiplying it by the speed of light in the vacuum. However, since the receiver clock is not synchronized with the transmitters clock, this measure suffers of an error $b = \delta t \cdot c$, where δt is the clock bias. For this reason, ρ is called "pseudo-range".

$$\rho = c(\tau + \delta t)\,. \tag{1.1}$$

The pseudo-range represents a first rough estimate of the true satellite-user range. Figure 1.1 shows this principle for two satellites and pseudo-ranges ρ_1 and ρ_2. In the three-dimensional space, every pseudo-range defines a spherical surface, the centre of which is the satellite. The intersection of at least three of these spheres is a point in space, representing the user position. A fourth measurement is required to solve the time error bias.

Navigation equations can thus be defined based on (1.1), and assume the form [3]

$$\begin{cases} |\rho_1| = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + b \\ |\rho_2| = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + b \\ |\rho_3| = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + b \\ |\rho_4| = \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + b \end{cases} . \tag{1.2}$$

where x_u, y_u, z_u are the user coordinates and $x_{1\ldots4}$, $y_{1\ldots4}$, $z_{1\ldots4}$ are the coordinates of the four satellites. The solution of the system of equations in (1.2) gives the unknown user position and the unknown clock bias. Notice that other errors are present in the range measurement, in addition to the time offset, but in the following they have been neglected.

The signal propagation time used in (1.1) can be computed according to two different measurement principles [4].

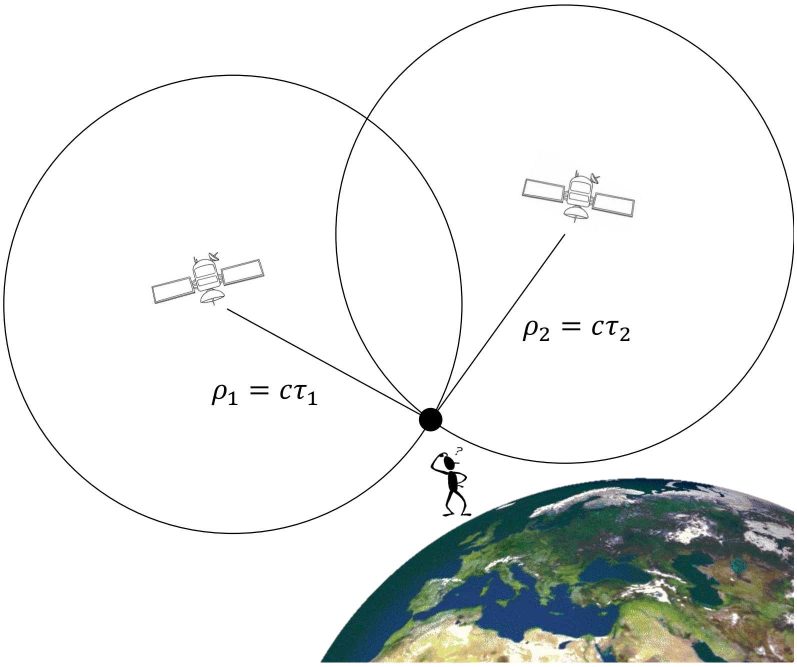


Figure 1.1: General concept of GNSS receivers operations: the user determines its PVT solution by processing the signals transmitted by a set of satellites in view [2].

Code measurements provide an estimate of the SIS transit time by tracking its code phase. They represent the standard technique, used in any navigation receiver, giving a metre-level accuracy. The code measurement principle will be detailed in Section 2.6.

Carrier phase measurements provide an estimate of the SIS transit time by tracking the signal carrier phase, relative to the phase of a locally generated signal. The accuracy they can provide is much higher, of the order of centimetres. However, they are extremely less robust and affected by an ambiguity difficult to solve. For this reason they are only implemented in professional receivers. Their principle is described in Section 6.1.1.

1.1.3 GPS

GPS is the most common, used and studied GNSS. It has been developed by the United States (US) Department of Defence (DoD) in 1973, to overcome the limitations of previous navigation systems (such as TRANSIT), mainly for military purposes. It has become fully operational in 1995 and in January 1999 the US government announced a new GPS modernization initiative that called for the addition of two civil signals.

Similarly to other navigation satellite systems, it consists of three segments: the space segment, the control segment, and the user segment. The main functions of the **space segment** are to transmit radio-navigation signals, and to store and retransmit the navigation message sent by the control segment. These transmissions are controlled by highly stable atomic clocks on board the satellites. The GPS space segment is formed by a satellite constellation with enough satellites to ensure that the user will have at least 4 simultaneous satellites in view, from any point on the Earth surface, at any time. The GPS **ground segment** (or control segment) is responsible for proper system operation. It is composed by a network of Monitor Stations (MS), a Master Control Station (MCS), a backup of the MCS and Ground Antennas (GAs). The MCS processes the measurements received by the MS to estimate satellite orbits (ephemerides) and clock errors, among other parameters, and to generate the navigation message. These corrections and the navigation message are uploaded to the satellites through the GA. The **user segment** consists on L-band radio antennas and receivers, which receive the GPS SIS, determine pseudo-ranges (and other observables), and solve the navigation equations in order to obtain their coordinates to provide a very accurate time. This thesis focuses on the user segment and on receivers. More details on the signal structure can be found in Chapter 2. More details on the GPS system can be found in [5, 6]. GPS current status can be found in [7].

1.1.4 Galileo

Galileo is European Union (UE)'s GNSS, providing accurate navigation signals everywhere in the world and currently being built by the European Space Agency (ESA) [8]. Galileo was born in the 1990s, when the UE identified the need for an independent and civil GNSS. Thirty satellites, including spares, and the necessary ground infrastructure will make up the system when it will be fully up and running. Initial services will be provided in the incoming years, based on a first constellation of 18 satellites.

Galileo is an open, global system, fully compatible and interoperable with GPS but independent from it. Interoperability features with GPS, GLONASS and other external systems is a primary requirement to enable the provision of services based on the combined use of the signals. This means that a user will be able to locate himself with the same receiver from any of the satellites in any system. Its baseline design is composed by a fully deployed constellation of 30 satellites (27 operational + 3 spares), positioned in three circular Medium Earth Orbit (MEO) planes at a nominal average orbit semi-major axis of 29 601.297 km, in order to ensure the entire coverage of the Earth's surface.

More details on the Galileo signal structure are provided in Chapter 2. Galileo current development status can be found in [9].

1.1.5 Other global and regional systems

Although the main interest in this work in on GPS and Galileo signals, other important systems are briefly described hereafter.

Glonass (GLObal'naya NAvigatsionnaya Sputnikovaya Sistema) is the Russian radio-based satellite navigation system. The system was formally declared operational in 1993 and brought to its optimal status of 24 operational satellites in 1995 [2]. The collapse of Russian economy, as well as the reduction in funding for space industry, lead the system to fall into disrepair just after completion. Then, in the early 2000s, the restoration of the system started and in May 2007 a decree on the Glonass navigation system has been signed, to provide the service to Russian and foreign consumers free of charge and without limitations. Glonass modernization is planned, with the launch of a new generation of satellites, including also Code Division Multiple Access (CDMA) signals in addition to the legacy Frequency Division Multiple Access (FDMA) signals (Section 2.3).

The **BeiDou** Navigation Satellite System, also known as Beidou-2, is a project by China to develop an independent GNSS. BeiDou is not an extension to the previously deployed Beidou-1, but a new GNSS, similar in principle to GPS and Galileo.

The **Quasi-Zenith Satellite System (QZSS)**, is a three-satellite regional system, commissioned by the Japanese Government as a National Space Development Program, receivable within Japan. The first satellite *Michibiki* was launched on September 11, 2010.

Indian Regional Navigational Satellite System (IRNSS) is an autonomous regional satellite navigation system owned by the Indian government and being developed by the Indian Space Research Organization (ISRO), which would be under total control of Indian government.

1.1.6 Multi-constellation

Till a few years ago, GPS and GNSS were just synonyms and used interchangeably. Glonass has never been utilized for civilian purposes till 2010, when the 21-satellites constellation has been declared operative again. Around 2011 the first commercial receivers able to receive both GPS and Glonass signals appeared in the market, paving the way towards a multi-constellation scenario [10]. At present, quad-constellation mass-market chipset have been designed [11]. From the user perspective, multi-constellation will allow the use of signals from all satellites in view, irrespective of what system they belong to, bringing a key added value to accuracy, availability, integrity, especially in urban environments. This has been encouraged at system design level, by working towards interoperability and compatibility [12].

Figure 1.2 shows the estimate number of satellites orbiting around Earth from 2009 to 2021, including all the global, regional and augmentation systems. It is clear how in the following years the scientific and industrial GNSS community will have to face with a multi-constellation scenario and get the best from all the different systems.

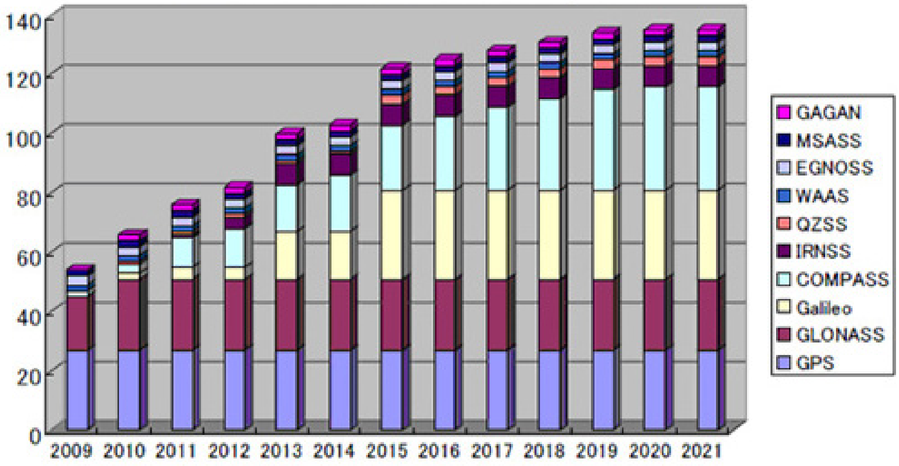


Figure 1.2: Number of SVs available in a multi-constellation system.

At time of writing, the following global, regional and augmentation systems can be considered operational or in development.

Global satellite systems

- **GPS**, owned by the US DoD, operational;
- **Glonass**, operated for the Russian government by the Russian Space Forces, operational;
- **Beidou**/Compass, China's second-generation satellite navigation system, under completion;
- **Galileo**, Europe's navigation satellite system, being built by UE and ESA.

Regional satellite systems

- **QZSS** commissioned by the Japanese Government;
- **IRNSS**, owned by the Indian government.

Augmentations systems

- **European Geostationary Navigation Overlay System (EGNOS)**, the European Satellite-Based Augmentation Service (SBAS);
- **Wide Area Augmentation System (WAAS)**, an air navigation aid developed by the US Federal Aviation Administration;
- **Multi-functional Satellite Augmentation System (MSAS)**, a Japanese SBAS;
- **GPS Aided Geo Augmented Navigation (GAGAN)**, an implementation of a SBAS by the Indian government.

1.2 Applications of GNSS

Satellite navigation touches all aspects of our lives, in dozen of different ways [8, 13]:

Automotive. In vehicle navigation allows drivers to determine their accurate position. Navigation systems can also improve highway operations by improving safety and reducing congestion, leading to a more efficient management of the roadway system; vehicle/highway information exchange is a key factor to improved management and operation of the transportation network. Information on traffic conditions, crashes, adverse weather and road conditions, can be localized and provided to users. At the same time automatic vehicle location improves safety and security in cargo fleet tracking, control, dispatch. Important fuel savings and air pollution reduction can be reached, as well as improvements in asset management. Also road maintenance can benefit of GNSSs.

Aviation. Aviation can use satellite navigation for reliable and accurate positioning worldwide, for precise en-route navigation and precision approaches even in poor weather conditions and where there are no ground based navigation aids. This improves safety and will make possible for airports to grow. Delays can be reduced, more fuel efficient routes can be discovered and in general increased system capacity with enhanced safety can be achieved.

Boats and ships. Maritime GNSS not only helps vessels, large ships, fishing and recreation boats navigate the world's seas and waterways, but also enables functions like *man overboard*, which instantly mark the position of anyone who has fallen into the water. It is also a key technology for harbour approach and entrance, regardless of visibility.

Railroads. GNSSs applied to rail traffic can enhance safety and reduce accidents, increase capacity and efficiency, allowing to reduce the train spacing and the fuel consumption. Rapid rail structure and condition mapping improves maintainable capability, while train control enables location tracking of vehicles and containers, increasing efficiency and capacity.

Agriculture. Precision agriculture uses GNSSs to maximize the use of resources and to minimize the environmental impact in a variety of ways. For example, navigation satellite systems can show farmers where to spray crops, tailoring application of seeds, fertilizer, water, pesticides. Another application concerns precisely levelling of the fields to prevent fluid run-off, or optimizing the ploughing of crop rows.

Sport and entertainment. Climbers, hikers and cyclists can use GNSSs to work out where they are; chances of rescue in case of accident are greatly improved by the ability to provide an accurate position.

Dangerous works. Blades and buckets of heavy equipment used in mining or construction are controlled automatically by GNSSs.

Surveying, mapping, GIS. Sub-centimetre accuracy can be obtained with professional receivers, with up to 100%/300% savings in time, cost, labour. Indeed, most major development projects require surveying, such as rural electrification, telecom tower placement, pipeline installation, dam construction, oil, gas, and mineral exploration, flood plain mapping.

Environmental protection. GNSS applications can help in protecting forest, in enforcing fishing boundaries, in preserving endangered species and habitats, in managing natural resource, in the hazardous clean-up (oil spills, toxic waste).

Scientific research. Satellite navigation systems are more and more used in scientific research. For example for characterization of the atmosphere (radio occultation), performing remote sensing (GNSS scatterometry), but also for monitoring wildlife behaviour, for atmospheric modelling (water vapour content), for oceanic studies (tidal patterns, surface mapping) and for space exploration. Stresses in Earth's crust leading to geological changes (glaciers, tectonic plates, earthquakes, volcanoes), can be measured by tracking the surface movement indicated between GNSS sensors.

Timing. Time stamping is a very useful secondary application of GNSSs and represents an inexpensive alternative to costly, high maintenance timing equipment. GNSS is used in telecommunication networks synchronization and management (phones, pagers, wireless systems, Local Area Networks (LANs), Wide Area Networks (WANs), internet) and in electrical power grid management and faults location. Time is also crucial for financial transactions and for the e-commerce.

1.3 A focus on Galileo IOV

Despite being partially out of the scope of the thesis, I believe that a small mention to the first Galileo acquisition and to the first Galileo-only PVT are of paramount importance. Not only because they happened just during my PhD, when I was in Torino in the first case, and when I was at the hearth of the Galileo program, at European Space Research and Technology Centre (ESTEC), ESA, in the second case, but also because much attention has been given from the GNSS community to Galileo in the last years.

Galileo is currently in the first main phase of the programme, the In-Orbit Validation (IOV) phase. The IOV phase consists of qualifying the system through tests and the operation of two experimental satellites and a reduced constellation of operational satellites and their related ground infrastructure. The first technology in orbit tests were conducted, starting December 2005, within the GIOVE mission, exploiting GIOVE-A and GIOVE-B experimental satellites, at present already dismissed. Later, other 6 IOV satellites have been launched, and at the moment they are broadcasting navigation signals, allowing a wide variety of tests all across Europe, to assess the performance of the system sub-set already deployed.

1.3.1 Galileo first acquisition

On December 12, 2011, one of the two Galileo IOV satellites launched on October 21—the Galileo-ProtoFlight Model (PFM) spacecraft—started transmitting its payload signal on the E1 band over Europe. That same day we were able to acquire and track the E1 signal (Galileo Code Number 11) beginning at 14:46:15 Central European Time (CET) at the Istituto Superiore Mario Boella (ISMB) premises (located in Torino, Italy) with a non-directive antenna, a commercial narrowband RF front-end, and a proprietary software receiver, developed by Navigation, Signal Analysis and Simulation (NavSAS) group [14]. Two days later, on December 14, we received the PFM E5 signal, using a similar experimental set-up.

Since the launch, we were checking the presence of the signal in lab during working hours and we were able to remotely control the system 24/7. Furthermore, we configured the software receiver to send an alert to NavSAS team members' mobile phones in case of successful signal acquisition.

Figure 1.3: First successful acquisition of Galileo OS signals on December 12, 2011 at 14:46:15 CET.

The picture in Figure 1.3 shows the moment of the very first successful acquisition of the signal at ISMB navigation lab.

Figure 1.4 shows the acquisition search space Cross Ambiguity Function (CAF), as a result of post-processing the data captured at that time. The CAF was computed over the 2D acquisition search space, defined along the code delays and Doppler frequency shifts, and obtained performing 11 non-coherent accumulations of 4 ms coherent integration times The presence of the Galileo signal (Code Number 11) is revealed, since the correlation peak rises above the noise floor. However, the code correlation function is not clear enough to recognize the typical BOC(1,1) autocorrelation shape, as the side peaks are buried in the noise floor. More details can be found in [14, 15].

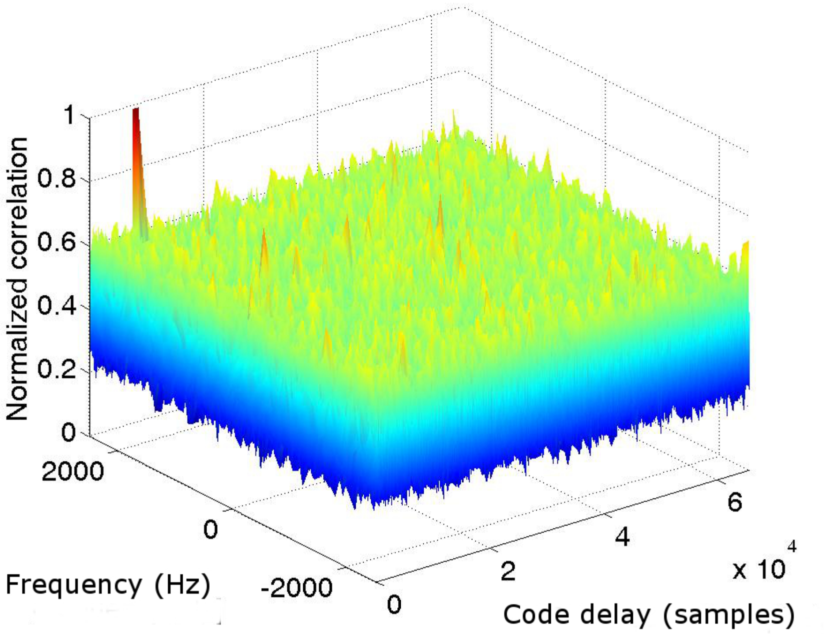
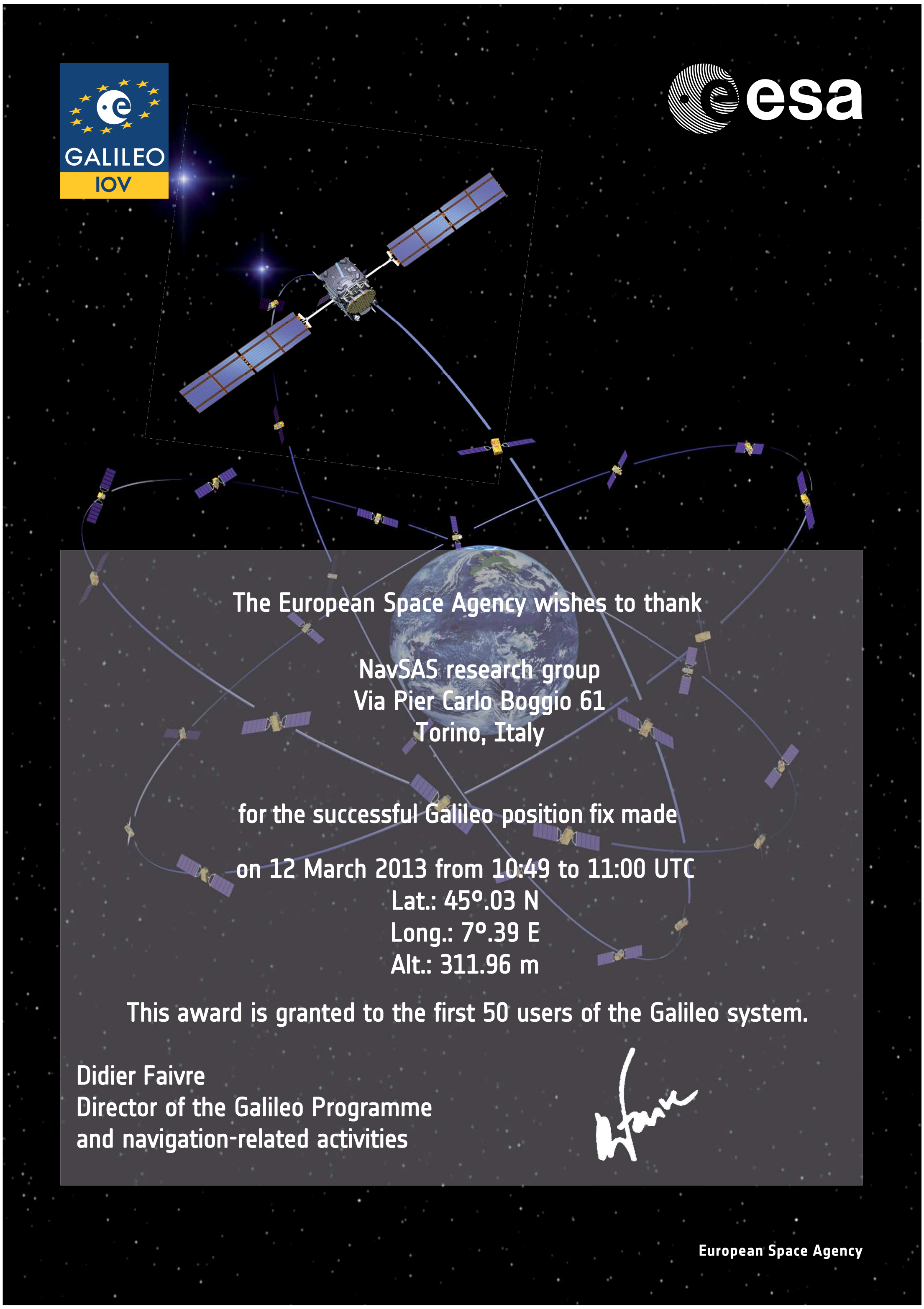


Figure 1.4: Acquisition search space of the first Galileo E1 signal reception on December 12, 2011 at 14:46:15 CET.

1.3.2 Galileo first position fix

On October 12, 2012, two more satellites were launched and on March 12, 2013, the four Galileo IOV satellites were switched on at the same time [16]. Once more, NavSAS was one of the first groups to successfully acquire, track, demodulate the four signals and to compute the Galileo-only PVT, as certified by ESA (Figure 1.5).

At the same time, in ESTEC, ESA, similar tests were carried out. More details are reported in Section 5.10.



GALILEO
IOV
esa

The European Space Agency wishes to thank

NavSAS research group
Via Pier Carlo Boggio 61
Torino, Italy

for the successful Galileo position fix made

on 12 March 2013 from 10:49 to 11:00 UTC
Lat.: 45º.03 N
Long.: 7º.39 E
Alt.: 311.96 m

This award is granted to the first 50 users of the Galileo system.

Didier Faivre
Director of the Galileo Programme
and navigation-related activities

European Space Agency

Figure 1.5: Certificate issued by ESA, July 10, 2014.

Chapter 2

Signal and System Model

This chapter describes the structure and the components of GNSS signals. In particular, this thesis focuses on GPS Coarse Acquisition (C/A) legacy signals in the L1 band and on Galileo Open Service (OS) signals in the E1 band. Nevertheless, it can be stated that most of the techniques can be extended to other signals.

2.1 The GNSS signal

The signal transmitted by GNSS satellites must satisfy several requirements. First of all, it must allow the user to estimate his distance from the satellite, the pseudo-range. The transmitted signal must also carry data containing useful information about the satellite position and velocity, and time. Moreover, it has to be robust against interference, multipath, atmospheric distortions and other types of errors. Finally, it must allow the identification of each satellite in a unique way. RF CDMA signals modulated using Pseudo-Random Noise (PRN) sequences accomplish these requirements and have been thus chosen for navigation purposes. Legacy GNSS signals and they components are detailed in the following.

The analog signal transmitted by a GNSS satellite can be modelled as

$$s(t) = \sum_{i=1}^{L} y_i(t)\,, \tag{2.1}$$

where, $y_i(t)$ are the different signal components. For example, in the case of the E1 Galileo signals, $L = 3$ and $y_1(t)$, $y_2(t)$ and $y_3(t)$ are the E1A, E1B and E1C components, respectively. As shown in Figure 2.1, each signal component, $y_i(t)$, consists of three different terms, a RF carrier, one or more spreading codes and a navigation message. It can be modelled as

$$y_i(t) = \sqrt{2P} d_i(t) c_i(t) \bar{c}_i(t) \cos\left(2\pi f_{RF,i} t + \varphi_i\right)\,, \tag{2.2}$$

where

- P is the transmitted power of the i-th component;
- $d_i(t)$ is the **navigation message**, containing the GNSS time, the satellite ephemeris and health status and the almanacs. It is absent in GNSS pilot signals.
- $c_i(t)$ is the **spreading code**, used for modulating the navigation message.

- $\bar{c}_i(t)$ is the **secondary code**, present in most recent GNSS signals;
- $\cos(2\pi f_{RF,i}t + \varphi_i)$ is the RF **carrier** component;
- $f_{RF,i}$ is the carrier centre frequency;
- φ_i is a phase term used to indicate if the signal component has been broadcast in-phase or in quadrature. For example, the GPS L1 C/A and Galileo E1BC components are broadcast in-phase, with $\varphi_i = 0°$ whereas the Galileo E1A component is broadcast in quadrature, with $\varphi_i = 90°$.

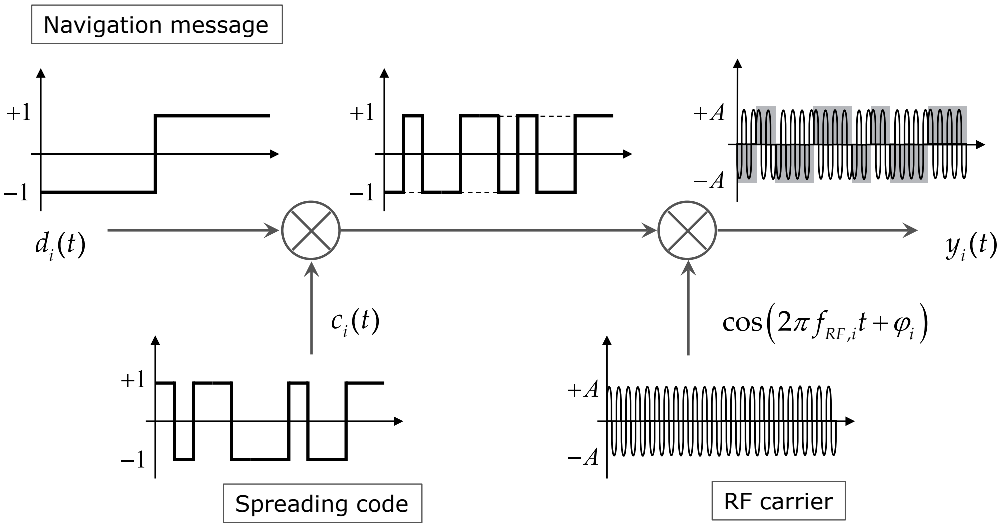


Figure 2.1: Structure of a GNSS signal $y_i(t)$. Three components can be identified: the navigation message $d_i(t)$, the code $c_i(t)$ and the RF carrier $\cos(2\pi f_{RF,i}t + \varphi_i)$.

The signal received by a GNSS receiver is given by the superposition of K useful signals of the form of (2.1) and a noise term; K denotes the number of satellites in view. Each useful signal, $s(t)$, passes through a communication channel that modifies it. At first a delay, usually referred as code delay is introduced. This delay is proportional to the distance between the satellite j and the receiver and affects both the navigation message and the code. For the same reason, a phase term is introduced in the carrier. In addition, because of the relative motion between the user and the satellite, the central frequency of the signal is affected by the Doppler effect. Since each satellite is in a different position and has a different relative velocity with respect to the user, the code delay and the Doppler frequency are different for each Space Vehicle (SV) [17]. Thus the received signal, usually denoted SIS, can be modelled as

$$r(t) = \sum_{j=1}^{K} \tilde{s}_j(t) + \eta(t) = \sum_{j=1}^{K}\sum_{i=1}^{H} \tilde{y}_{ij}(t) + \eta(t),, \tag{2.3}$$

where $\tilde{s}_j(t)$ denotes the signal $s(t)$ broadcast by the j-th satellite after passing through the communication channel and $\eta(t)$ is the noise. The noise can either come from the outside environment and be captured by the antenna (interference), or being generated by the electronic devices inside the receiver (thermal noise). This results in a random process that is usually modelled as an Additive White Gaussian Noise (AWGN), with a flat power spectral density equal to $N_0/2$ W/Hz. So the noise term η is a white sequence, distributed according to a Gaussian process with zero mean and variance equal to σ^2:

$$\eta(t) \sim \mathcal{N}\left(0,\, \sigma^2\right) . \tag{2.4}$$

In (2.3), each signal, $\tilde{s}_j(t)$, is given by

$$\tilde{s}_j(t) = \sum_{i=1}^{H} \tilde{y}_{ij}(t) \, , \tag{2.5}$$

which is the linear combinations of H useful components. Each signal component, $\tilde{y}_{ij}(t)$, can be modelled as:

$$\tilde{y}_{ij}(t) = \sqrt{2C_{ij}} d_{ij}\left(t - \tau_{ij}\right) c_{ij}\left(t - \tau_{ij}\right) \bar{c}_{ij}\left(t - \tau_{ij}\right) \cos\left[2\pi\left(f_{RF,i} + f_{D_{ij}}\right) t + \varphi_i + \phi_{ij}\right] \tag{2.6}$$

where

- C_{ij} is the received signal power;
- τ_{ij} is the code delay introduced of the i-th components of the signal broadcast by the j-th satellite;
- $f_{D_{ij}}$ is the Doppler frequency shift;
- ϕ_{ij} is the carrier phase introduced by the channel.

Because of the quasi-orthogonality of the signal codes broadcast by the different satellites and of the different components, the receiver is able to process each signal component, $\tilde{y}_{ij}(t)$, independently [3]. For this reason, the indexes i and j are removed in the following and a single signal component is considered. Quasi-orthogonality derives from the good correlation properties of PRN sequences.

2.2 Frequency plan

GNSSs transmit in the L band; the complete frequency plan of all signals is depicted in Figure 2.2 [2] In particular the centre frequencies of the most relevant OS signals, denoted f_{RF} in (2.2), are:

- $f_{L1} = f_{E1} = 1575.42$ MHz for GPS C/A and Galileo E1B and E1C signals;
- $f_{L2} = 1227.60$ MHz for GPS L2C signal (GPS modernization);
- $f_{L5} = f_{E5a} = 1176.45$ MHz for GPS L5 (GPS modernization) and Galileo E5a;
- $f_{E5b} = 1207.14$ MHz for Galileo E5b.

2.3 The spreading codes

Several GNSSs use Direct-Sequence Spread Spectrum (DSSS) modulation techniques. The idea is to spread the signal over a wider bandwidth, by multiplying it by a periodical PRN binary sequence, called spreading code, $c(t)$, and known by both sender and receiver [18]. In particular, CDMA modulation is used in the majority of the systems. A spreading sequence of length P is

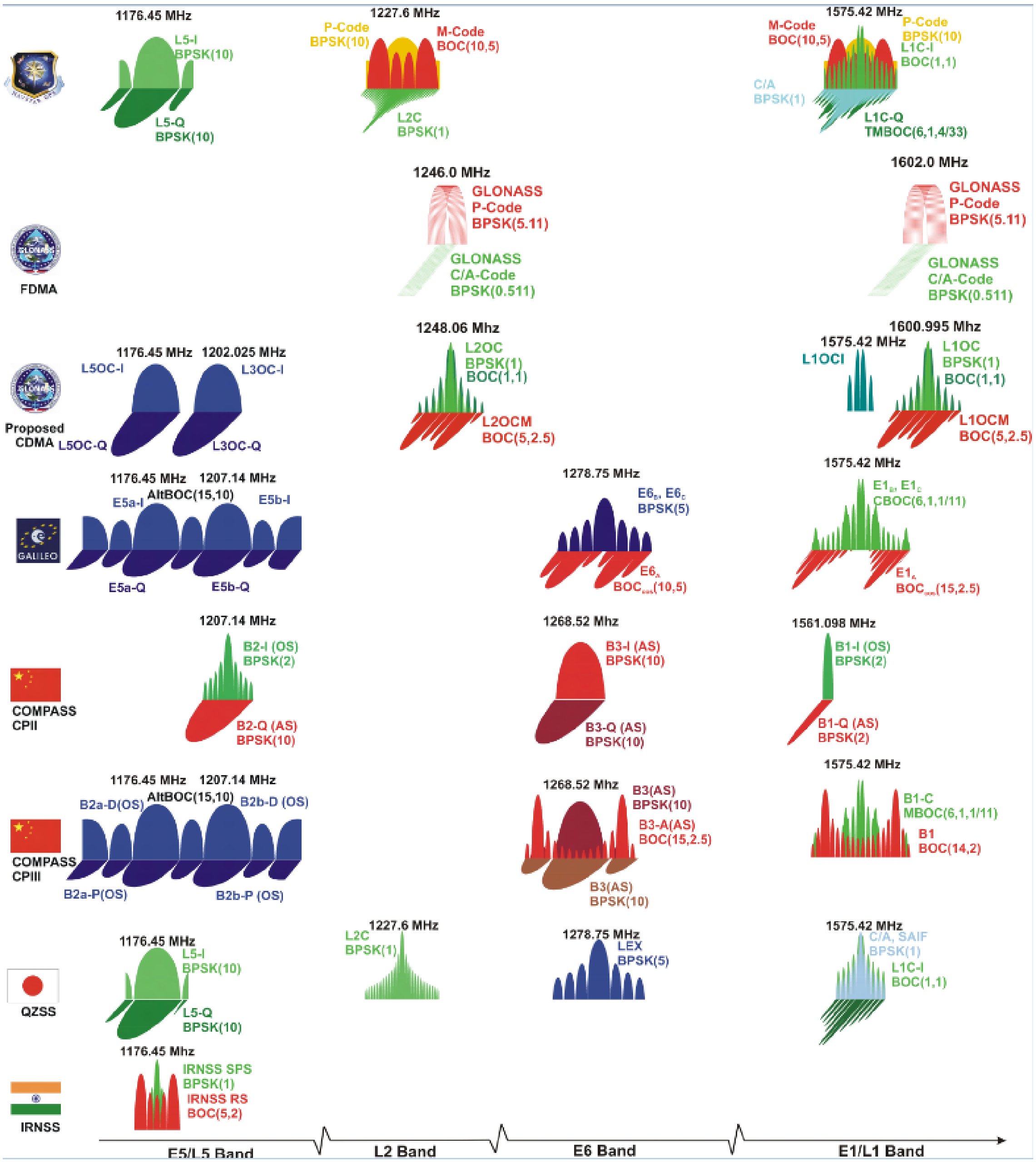


Figure 2.2: GNSS frequency bands [2].

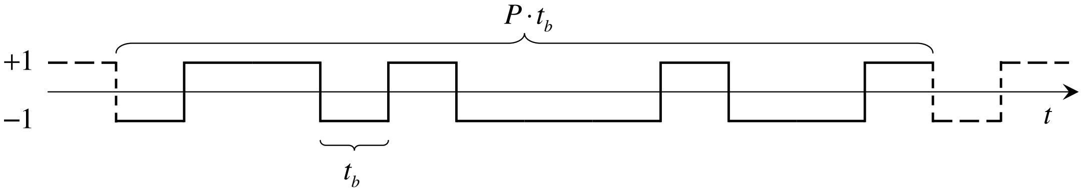


Figure 2.3: Example of a PRN spreading sequence with $P = 12$.

made by P pulses of amplitude $+1$ or -1, called chips. Therefore the chip length, t_b, is the duration of a single chip and the code length is equal to $P \cdot t_b$, as depicted in Figure 2.3. The chipping rate is the frequency of a chip, $1/t_b$.

By multiplying the original signal, $d(t)$, by a spreading code, the bandwidth of the signal increases because of the high chipping rate of the PRN code. In this way, the power of the original signal is spread over a wider bandwidth, giving a stronger resistance against interference, multipath and jamming. In Figure 2.4 the spectrum of the original signal is shown. It is characterized by a bandwidth B_S. When the same signal is modulated by a spreading code, its bandwidth becomes B_C.

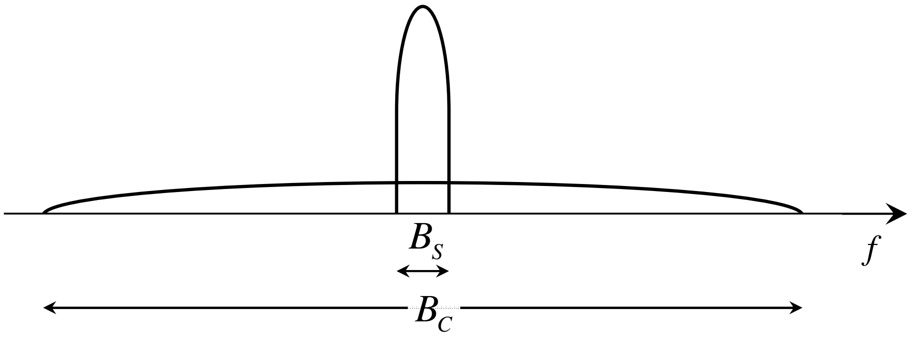


Figure 2.4: Spectrum of a generic signal with bandwidth B_s and of the same signal spread over the bandwidth B_c using a DSSS modulation.

Moreover, most of PRN sequences, among which Gold codes used in GPS and memory codes used in Galileo, exhibit good correlation properties [19]. The correlation is a measure of the similarity between two sequences, which in this case are represented by the PRN sequences. In particular the normalized cyclic cross-correlation measures the similarity between two different sequences cyclically shifted with respect to each other; it is defined as [19]

$$R_{x,y}(l) = \frac{1}{P}\sum_{p=1}^{P} x\big[p\big] y\big[(p+l) \mod P\big] , \tag{2.7}$$

where P is the length of the sequences, x is the first sequence and y is the second sequence. The normalized cyclic auto-correlation measures the similarity between a sequence and a shifted version of itself and it is defined as [19]

$$R_x(l) = \frac{1}{P}\sum_{p=1}^{P} x\big[p\big] x\big[(p+l) \mod P\big] . \tag{2.8}$$

Good PRN sequences are characterized by an auto-correlation equal to 1 for $l = 0$, i.e. when the two sequences are perfectly aligned, and with very low values (ideally equal to zero) for the other values of l. In addition to this, a good PRN code should have very low (ideally equal to zero) cross-correlation values when correlated with a sequence from the same family. These properties introduce two important consequences: since the cross-correlation between different sequences is low, it is possible to share the communication channel between several users, each one distinguished by a different PRN sequence, and then to isolate the signal coming from a single user (CDMA despreading). Moreover since the auto-correlation has a strong peak, it is possible to measure the delay between the incoming sequence by correlating it with a local replica of the incoming signal. This delay corresponds to the transit time of the signal, modulo the code period, that can be converted into a pseudo-range. Thus, the use of PRN sequences in the GPS signal leads to several important benefits:

- resistance to jamming, interference and multipath;
- sharing of the channel between several satellites and thus identification of the transmitting satellite;
- determination of the transit time of the signal, through code phase measurements.

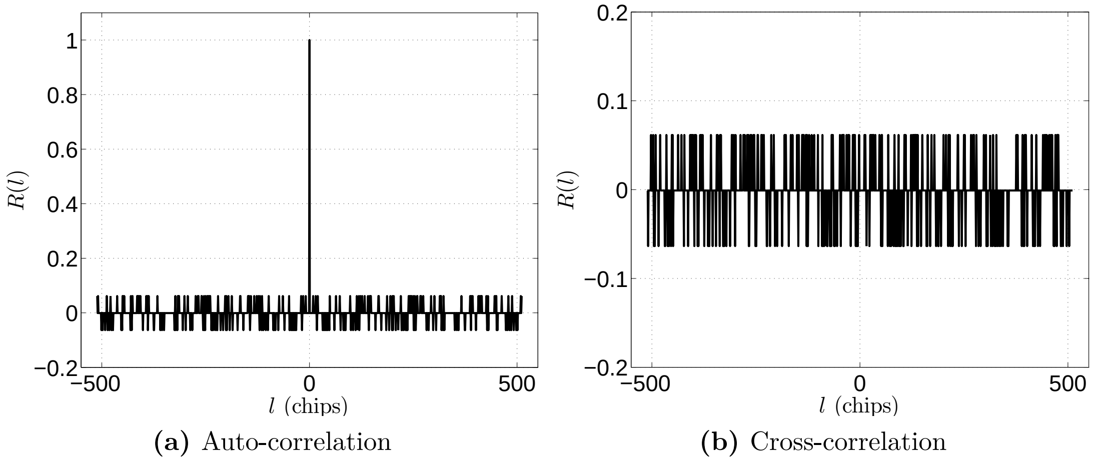
2.4 The GPS C/A signal

The C/A is the GPS signal mainly used for civilian applications in any kind of receiver. It is characterized by Binary Phase-Shifting Key (BPSK) modulation, using a 1023 chips long PRN periodic sequence with chipping rate equal to 1.023 MHz. Therefore an entire period of the code lasts 1 ms [20]. The GPS C/A signal belongs to a family of PRN codes, known as Gold codes. The navigation message is broadcast at a rate of 50 bps. Figure 2.5a shows the auto-correlation of a PRN sequence used in GPS, normalized to the unitary value; as expected there is a peak at $l = 0$, i.e., when the two sequences are perfectly aligned, and low correlation values elsewhere. It can be proved that these values are equal to $-65/1023$, $-1/1023$ and $63/1023$ [19]. Likewise, Figure 2.5b shows the cross-correlation between two different PRN sequences; the values are quite close to zero everywhere, exactly equal to $-65/1023$, $-1/1023$ and $63/1023$ [19].

2.5 The Galileo E1BC signal

The Galileo civil signals in E1 band provides the OS and the Safety of Life (SoL) services. Two signal components are used, conventionally denoted as E1B and E1C channels: they are a data channel (carrying I/NAV message at 250 bps) and a pilot (data-less) channel respectively [21]. The signals use a Composite Binary Offset Carrier (CBOC) modulation, the result of multiplexing a wideband signal, Binary Offset Carrier (BOC)(6,1), with a narrow band signal, BOC(1,1), in such a way that 1/11 of the power is allocated, on average, to the high-frequency components. In case of a BOC(1,1) modulated signal, the minimum theoretical value for the bandwidth is 4.092 MHz, whereas at least 14.322 MHz are needed for receiving also the BOC(6,1) components. Both channels are modulated by a 4092 chips long PRN memory codes, with chipping rate equal to 1.023 MHz. Therefore an entire period of the code lasts 4 ms. In addition, E1C is modulated also by a secondary tiered code of length 25 chips, extending the periodicity to 100 ms.

The shape of the auto-correlation of the Galileo BOC(1,1) and CBOC modulation (along with the GPS BPSK(1)) are reported in Figure 2.6.



(a) Auto-correlation

(b) Cross-correlation

Figure 2.5: Auto-correlation and cross-correlation functions of a PRN sequence used for the C/A signal.

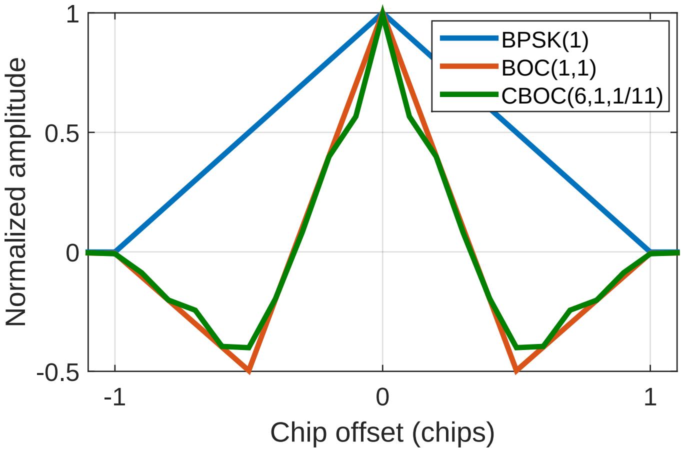


Figure 2.6: Auto-correlation for OS GPS and Galileo signals modulations [2].

2.6 Code phase measurements and pseudo-range

The basic measurement for GNSS receivers is the apparent transit time of the signal from the satellite to the receivers, defined as the difference between the signal reception time, as determined by the receiver clock, and the signal transmission time at the satellite, marked in the navigation message.

In code phase measurements, it is measured as the amount of time shift, modulo the code length, required to align the local code replica to the incoming signal. Three different time scales are involved in this measurement process, as depicted in Figure 2.7.

- **The satellite clock:** signals are generated on board in accordance with this highly accurate clock. All the satellite clocks are synchronized.

- **The receiver clock:** it is usually cheap and very unstable; in addition it is not synchronized with any reference time.
- **The GPS Time (GPST):** it is a common reference time, defined on the basis of a measurement from a set of celsius and rubidium frequency standards in use at the monitor stations. It has to be said that other GNSSs may consider other reference times; for example Galileo introduces the Galileo System Time (GST), a continuous time scale maintained by the Galileo Central Segment and synchronized with International Atomic Time (TAI) with a nominal offset below 50 ns. The GPS Galileo Time Offset (GGTO) is the difference between GPST and GST.

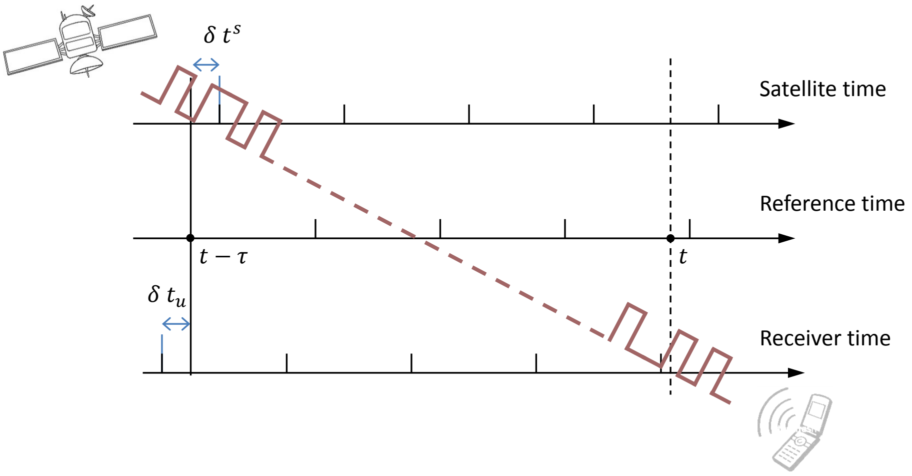


Figure 2.7: A conceptual view of the pseudo-range measurements, with clocks un-synchronized. The apparent transit time is the difference between the signal reception time according to the receiver clock and the satellite transmission time.

Let's suppose that a signal is received at a certain time t; according to the receiver clock, the reception time is $t_u(t)$, where the suffix u stands for *user*. Then, let's define the signal transit time from the satellite to the receiver as τ. The corresponding emission time can be defined as $t^s\,(t-\tau)$, according to the satellite clock, where the suffix s stands for *satellite*. Thus, the measured apparent range ρ can be determined as:

$$\rho(t) = c\left[t_u(t) - t^s\,(t-\tau)\right] \tag{2.9}$$

where c is the speed of light. This is an apparent range, because of the different time scales of transmitter and receiver clocks. It is then denoted *pseudo-range*, since both t and τ are unknown and need to be estimated. Since the distance of a satellite lies between 20 000 and 26 000 km, τ oscillates between 70 and 90 ms. The receiver and satellite time scales can be related by the reference time, as suggested below:

$$t_u(t) = t + \delta t_u(t) \tag{2.10}$$

$$t^s\,(t-\tau) = (t-\tau) + \delta t^s\,(t-\tau) \tag{2.11}$$

where δt_u and δt^s are the receiver and satellite clock biases respectively, as shown in Figure 2.7. These two quantities reflect the amounts by which the satellite and receiver clock are advanced in relation to GPST. The satellite clock bias is estimated by the GNSS control segments and is broadcast in the navigation message.

Therefore, accounting for the clock biases, the measured pseudo-range of (2.9) can be rewritten as:

$$\begin{aligned} \rho(t) &= c\left[t + \delta t_u(t) - (t-\tau) + \delta t^s\left(t-\tau\right)\right] \\ &= c\tau + c\left[\delta t_u(t) - \delta t^s\left(t-\tau\right)\right] . \end{aligned} \tag{2.12}$$

So far the transit time was considered as if the signal was travelling in space; nevertheless, this is not always the case. GNSS signals propagate through atmosphere and in particular they cross two layers with different RF propagation conditions: ionosphere and troposphere. The ionosphere extends from about 60 km to 1 000 km above the Earth's surface while the troposphere covers a lower region, from about 8 km to 40 km above the Earth's surface. They both introduce signal refraction, thus affecting the signal transit time, as follows:

$$c\tau = r\left(t, t-\tau\right) + I_p(t) + T_p(t) \tag{2.13}$$

where:

- $r\left(t, t-\tau\right)$ is the true geometric range between user position (x_u, t_u, z_u) at time t and the satellite position (x_s, t_s, z_s) at time $t - \tau$,

$$r\left(t, t-\tau\right) = \sqrt{\left[x_s(t-\tau) - x_u(t)\right]^2 + \left[y_s(t-\tau) - y_u(t)\right]^2 + \left[z_s(t-\tau) - z_u(t)\right]^2} ; \tag{2.14}$$

- I_p and T_p represent the delay expressed in meters associated to the signal transmission through ionosphere and troposphere respectively.

Summing up, combining (2.12) and (2.13), the pseudo-range can be rewritten as:

$$\rho(t) = r\left(t, t-\tau\right) + c\left[\delta t_u(t) - \delta t^s\left(t-\tau\right)\right] + I_p(t) + T_p(t) + \varepsilon_\rho(t) . \tag{2.15}$$

In (2.15) an additional error term $\varepsilon_\rho(t)$ has been added, to take into account unmodeled effects, such as modelling errors multipath, orbital and ephemeris parameter errors, receiver noise and measurement errors. In particular the GPS C/A code noise amounts to 5 to 200 cm (for Line of Sight (LOS) signal), the GPS Precision (P) code noise amounts to 10 cm and the error due to multipath is bounded to 1 chip [22].

Equation (2.15) is the main equation of code phase measurements; depending on the receiver ability to compensate for ionospheric delay, tropospheric delay and errors, the measured pseudo-range $\rho(t)$ will be closer to the true range $r(t)$.

2.7 GNSS receivers

GNSS receivers are part of the GNSSs ground segment and represent the main focus of this PhD dissertation. They process the received SISs and evaluate the user position.

GNSS receivers are a suboptimal implementation of a maximum likelihood estimator of the signal propagation time. In order to ensure tracking of the signals in each processing channel, receivers are continuously estimating and correcting two parameters:

- the **code delay**, which quantifies the misalignment between the incoming signal and the local PRN code replica;

- the **carrier phase**, or its instantaneous value, the **Doppler frequency**, which reflects the relative motion between user and satellites.

In order to determine these parameters, different estimation strategies can be used, as described in Section 4.1. After synchronization with the incoming signals and demodulation of the navigation message, the receiver is able to determine pseudo-ranges to each satellite, and to compute a navigation solution.

Back in the 1970s, receivers were large analog hardware equipment, targeting military applications. Around 1980 the first commercially available GPS receiver, the STI-5010, built by Stanford Telecommunications, Inc. has been introduced in the market. It was a dual-frequency, C/A and P-code, slow-sequencing receiver. Cycling through four satellites took about five minutes, and the receiver unit alone required about 30 centimetres of rack space. By 1990, a number of manufacturers were offering receivers for positioning, navigation, and timing applications. Already, the first handheld receiver was on the market, the Magellan NAV 1000. Its single sequencing channel could track four satellites.

Nowadays, consumer-grade GNSS receivers have been widely expanded to miniaturized platforms, chipsets, microprocessors, integrated circuits, Digital Signal Processors (DSP), Field Programmable Gate Arrays (FPGA), handheld devices, including integration in most mobile phones. Figure 2.8 shows the evolution of a GPS chipset in ten years, outlining the effort done by manufacturers in miniaturization and integration [23].

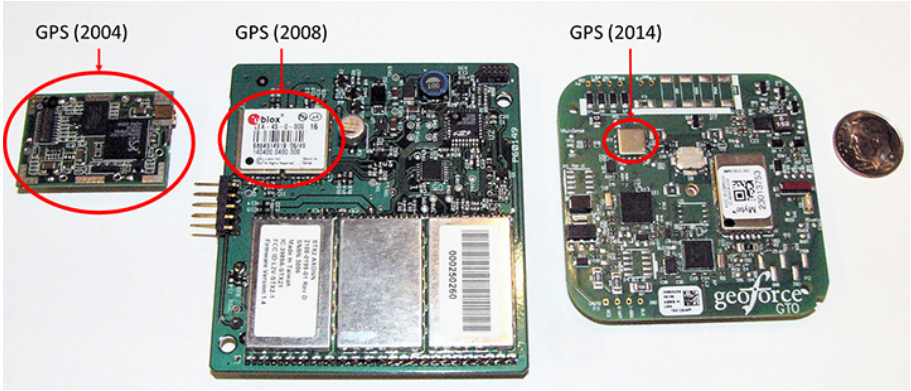


Figure 2.8: GPS chipset evolution: from 2004 (left) to 2008 (centre) to 2014 (right) [23].

Receivers have proliferated thanks to chips and modules miniaturization. In fact, GNSS receivers run in a wide variety of platforms, and the choice results from a trade-off of parameters such as performance, cost, power consumption and autonomy. A billion of GPS receivers are estimated to exist worldwide [24].

Furthermore, the increasing capability of microprocessors has enabled the emergence of Software Defined Radio (SDR) receivers with performance comparable to traditional hardware receivers. SDR refers to an ensemble of hardware and software technologies enabling a reconfigurable communication architecture. The term SDR was coined in 1984 by a team at E-Systems Inc. to refer to a digital baseband receiver; then, in 1991, Joe Mitola independently reinvented the term [25]. Since that time, SDR has gained more and more attention and followers, thanks to the huge opportunities it opens to the prototyping of new architectures, which feature lower development costs, shorter development time and far easier maintainability and upgradability, when compared to the dedicated hardware approaches [26]. With respect to commercial hardware tools, they provide a larger subset of observables related to the signal processing stages of the receiver, as well as a higher grade of flexibility and re-configurability, depending on the user needs.

More details on the receiver architecture are provided in Chapter 4. A wide and detailed description of receivers architecture and operations can be found in [4, 3, 17]. A SDR fully software receiver has been implemented during the PhD activity and is described in Section 5.1.

2.8 The C/N_0

An important parameter to measure the quality of a GNSS signal in is the Carrier-to-Noise density power ratio (C/N_0). The total power received by the antenna, as stated by the Friis transmission formula [27], depends on the satellite transmitted power, the satellite antenna gain, the receiver antenna gain, the free-space losses and the attenuations of the channel. It is denoted with C and in the case of GNSSs is in general very low; for the GPS L1 C/A signal the typical receiver power is approximately -160 dBW [28]. Therefore, the ratio between the received power of the signal carrier and the noise power in a 1 Hz bandwidth can be assumed as measure of the quality of the signal and is called C/N_0 [29].

The C/N_0, despite being a ratio, is expressed in logarithmic terms and its unit of measures is:

$$\left[\left. \frac{C}{N_0} \right|_{\mathrm{dB}} \right] = \left. \frac{[C]}{[N_0]} \right|_{\mathrm{dB}} = \left. \frac{W}{\mathrm{W/Hz}} \right|_{\mathrm{dB}} = \mathrm{Hz}|_{\mathrm{dB}} = \mathrm{dBHz}\,. \tag{2.16}$$

The usual value for the C/N_0 in normal GNSS applications is around $40 - 45$ dBHz, but this is strongly dependent on the quality of the receiver and on the environment, such as the weather conditions, the elevation of the satellite, the position of the receiver with respect to the satellites and the type of signal considered. Standard GNSS receivers can handle signals the C/N_0 of which is in the range $35 - 53$ dBHz. High-sensitivity receivers can process weaker signals.

Part I
Mass-market Receivers

Chapter 3

Strategic Role of Mass-market Receivers

The paramount importance and strategic role of GNSSs in the contemporary society has already been pointed out in Chapter 1. In particular, in Section 1.2, several different applications and uses of the PVT solution have been presented and, in turn, several different receiver types can be distinguished. In the remaining of this part, large importance is given to an emerging category of receivers: the devices devoted to the consumer market, usually called *mass-market receivers* or consumer devices. This category includes:

- receivers for the **automotive market**, such as vehicular navigation devices, Portable Navigation Devices (PNDs) and more recently, vehicular embedded systems;
- Personal Digital Assistant (PDA) devices, recently more known as **tablets**;
- GNSS devices for **recreational** purposes, such as GPS-equipped cameras for geo-tagging pictures or devices for sport and fitness, like GPS watches;
- mobile phones and **smartphones**.

Already in 2009 it was claimed that GPS was found more in phones than in the cumulative number of all other GPS receivers ever built [30]. Figure 3.1, by Frank Van Diggelen, gives an estimate of the number of GNSS devices produced up to 2012. First, as pointed out in [31], the number of military receivers, for which GPS was originally built, is extremely small, less than half a million. The grey bar depicts the industrial and professional uses, such as survey or aviation, and it comes in at about 4.5 million. Surprisingly, the chart scale has to be changed to depict the huge market segments of mass-market devices. Mobile phones still do not fit on the chart.

The reason for these exploding markets can be first found in the improvements in electronics and integration. Secondly, also the increasing availability of new GNSS signals (Section 1.1.6) is a key factor: with Galileo, QZSS, BeiDou, GPS-L1C, and Glonass-CDMA all on their way, the silicon manufacturer are continuing the path towards the fully flexible multi-constellation mass-market receiver [11]. This trend confirms the strategic importance of mass market receivers and of the analysis of their algorithms and performance.

In this chapter first the differences between mass-market and professional receivers are outlines, with a particular focus on the different receiver architecture. Then, in Section 3.2, three important drivers are described. Section 3.3 summarizes the concept of GNSS assistance and finally the typical mass-market scenarios, considered in the following chapters, are introduced.

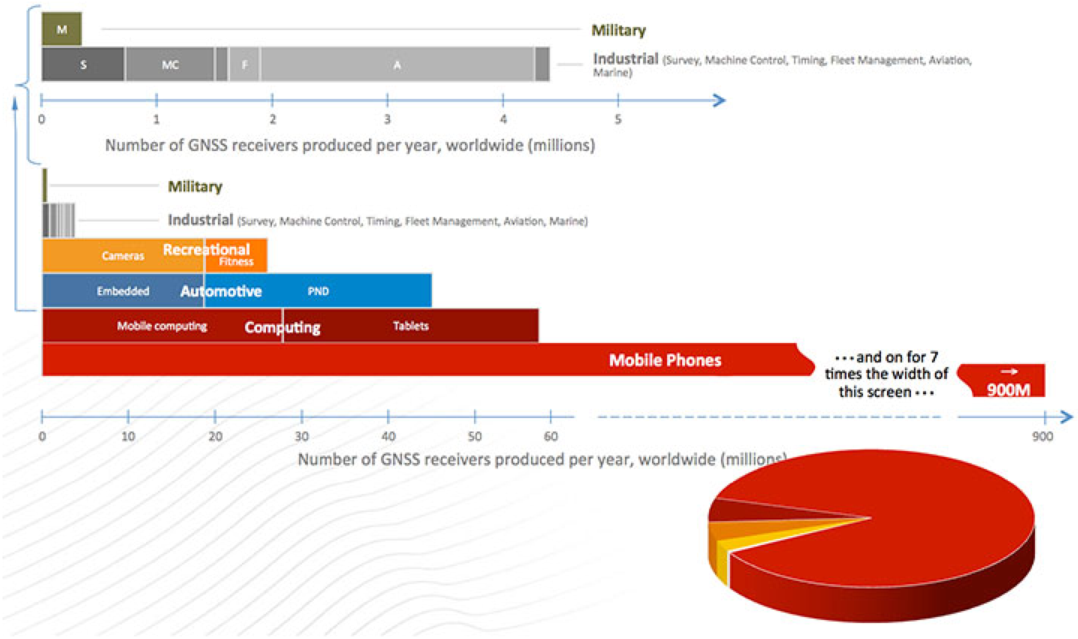


Figure 3.1: GNSS market size, 2012 [31].

3.1 Mass-market vs. professional receivers

Mass-market and professional receivers differ under many aspects. The main diversity is their final purpose: navigation, gaming and social for mass-markets and survey, construction, security for professional devices. This different use justifies their different performance concerning accuracy: metre-level is generally sufficient for the former, while millimetre-level is often required for the latter. As a consequence, their cost is very different; few dollars for a commercial GNSS chip and up to hundreds of thousands of dollars for a professional receiver. Clearly they can also differ in size, which is crucial for GNSS chips embedded in other platforms. What is more, mass-market receivers feature particular architecture and signal processing techniques, in order to comply with mobile and consumer devices' resources and requirements. These techniques are the object of this part of the thesis, and will be deeply investigated in Chapters 4 and 5.

Finally, it has to be said that GPS was originally designed for military use and expected to work outside, with a relatively clear view of the sky. Long initialization times were allowed, as long as the device could work continuously after completing acquisition. On the contrary, today, GNSSs are required to work almost anywhere, often in urban canyon, in light-indoor and indoor scenarios. A generic user typically requires his receiver to work instantly, and the classical continuous operating mode is more and more replaced by **on demand** PVT solution. These requirements pushed the development not only of new devices with ad-hoc algorithms, such as mass-market receivers, but also of **aiding techniques**, as Assisted GPS (A-GPS) (Section 3.3) or integration with inertials.

3.1.1 Architectural differences

The structure and architecture of commercial GNSS chips differs from the traditional acquisition and tracking scheme usually adopted in standard receivers. Standard and consumer receivers share the same RF front-end and navigation block, but differ in the Base Band (BB) stage, as clarified hereafter.

The RF front-end

The first stage of a GNSS receiver consists of the signal conditioning that transforms the RF signal of (2.3) into an Intermediate Frequency (IF) digital sequence. It is operated by the GNSS receiver *front-end*. The input SIS is usually too weak to be processed and needs to be amplified; a Low Noise Amplifier (LNA) is used to limit the amount of noise. There may also be a passive bandpass filter in order to minimize out-of-band RF interference. The frequency of the carrier, f_{RF}, is too high to allow any processing. Therefore the signal is down-converted to IF, at the frequency f_{IF} through a mixer, i.e. by multiplying the incoming signal by a locally generated sinusoid at a frequency f_{LO}, where $f_{LO} = f_{RF} - f_{IF}$. It has to be noted that also the noise part $\eta(t)$ is multiplied by the local sinusoid, but the resulting term is still a white Gaussian noise with the same variance. Then the signal is filtered to remove the high frequency term generated during the multiplication, and assumes the form:

$$r_{IF}(t) = \sqrt{2C}d\left(t-\tau\right)c\left(t-\tau\right)\bar{c}\left(t-\tau\right)\cos\left[2\pi\left(f_{IF}+f_D\right)t+\varphi\right]+\eta_{IF}(t)\,. \tag{3.1}$$

Equation (3.1), φ includes the effect of the channel and the initial phase offset broadcast by the satellite. $\eta_{IF}(t)$ is the down-converted noise term.

At this point, the signal is still in its an analog form and has to be converted into digital samples; an Analog-to-Digital Converter (ADC) is used at this step, with a sampling frequency f_s, leading to the digital version of the signal:

$$r[n] = \sqrt{2C}d(nT_s-\tau)c(nT_s-\tau)\bar{c}(nT_s-\tau)\cos\left[2\pi(f_{IF}+f_D)nT_s+\varphi\right]+\eta_{IF}(nT_s)\,, \tag{3.2}$$

where

- T_s is the sampling interval, $T_s = 1/f_s$;
- n is the digital time after sampling[1], $t_n = nT_s$;

[1] In the following the notation $r(nT_s) = r[n]$ is sometimes used to denote the sampled signal

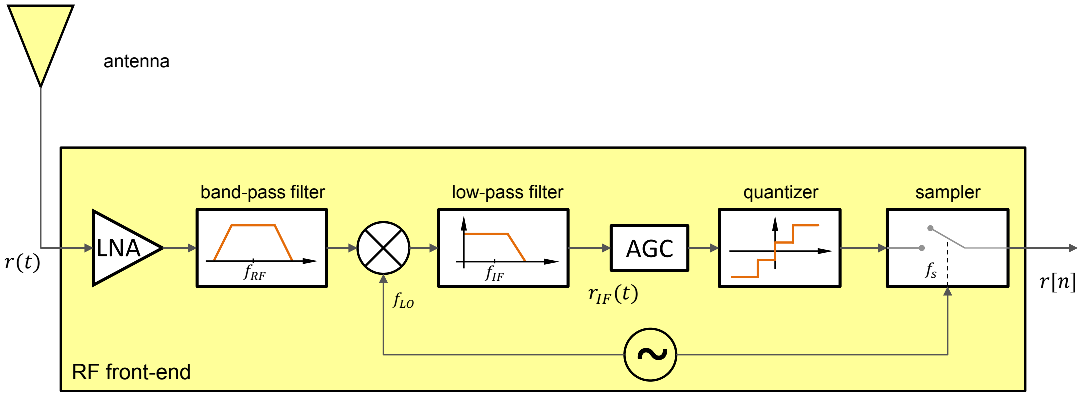


Figure 3.2: Front-end reference scheme.

- f_D the Doppler frequency; this term also includes the oscillator clock drift. In the following this oscillator error will be always assimilated to the Doppler frequency of the signal.
- η is the noise affecting the signal.

The front-end block scheme is reported in Figure 3.2.

The performance of a GNSS receiver strongly depends on how much noise affects the useful signal. The noise can either come from the outside environment and be captured by the antenna (interference), or being generated by the electronic devices inside the receiver (thermal noise) [29]. This results in a random process that is usually modelled as a white Gaussian noise, with a flat power spectral density equal to $N_0/2$ W/Hz, as shown in Figure 3.3. The total power of the noise at the output of the IF stage is then

$$P_\eta = \int_{-B_{IF}}^{B_{IF}} \frac{N_0}{2} = 2B_{IF}\frac{N_0}{2} = N_0 B_{IF}\,, \qquad (3.3)$$

where B_{IF} is the receiver bandwidth.

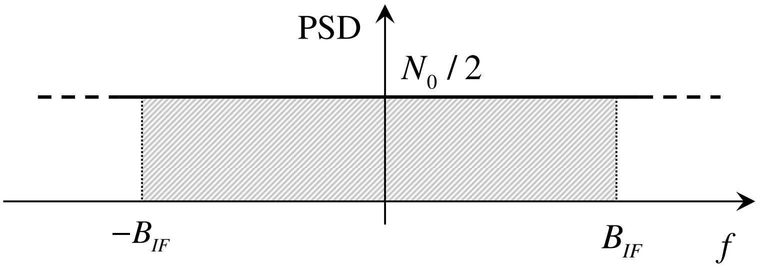


Figure 3.3: Power spectral density of the noise inside a GNSS receiver.

The standard receiver architecture

The standard receiver architecture is based on a 2-step estimation strategy, shown in Figure 3.4.

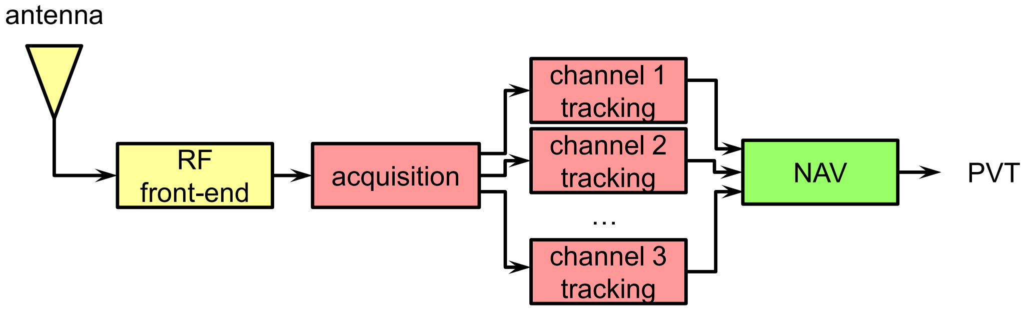


Figure 3.4: Standard GNSS receiver structure.

The IF digital signal of (3.2) first enters an **acquisition** engine; the list of SVs in view is determined and at the same time a first rough estimate of the signals parameters (code delay and Doppler frequency) is computed. An example of acquisition CAF is shown for example in Figure 1.3; the three dimensional search space shows the estimates of a single SV over a grid mapping all the possible values. Then the signals enters the standard **tracking** loop stage, depicted in Figure 3.5. It usually has two concatenated closed-loop structures (Section 4.1.1): a Delay Lock Loop (DLL) for estimating the code delay and a Frequency Lock Loop (FLL) or a Phase Lock Loop (PLL) for estimating the Doppler frequency and the carrier phase. Their purpose is to refine the Doppler and delay estimates, so as to compute the pseudo-ranges. This solution is the most common and studied technique in the GNSS community and is widely described in literature [17, 32, 3, 4].

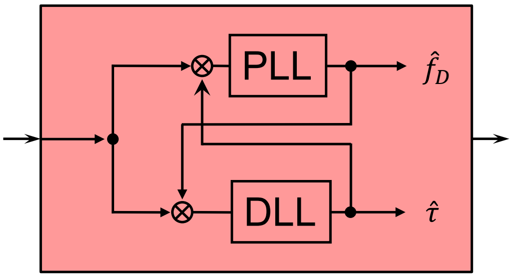


Figure 3.5: Tracking loops concatenation.

The consumer devices architecture

Contrary to standard architectures, consumer devices are based on a *single processing unit*, as shown in Figure 3.6, performing both acquisition and tracking in one shot, in the so called open-loop batch processing snapshot mode [33]. A survey showed that this technique is described in many Broadcom patents, and therefore is likely to be implemented in this category of receivers. This processing unit is usually rather configurable, and thus flexible and adaptable to different situations. It is based on multi-correlators (Section 4.1) and operates according to the open-loop paradigm (Section 4.1.1), thus performing **snapshot processing**. Then, it is more effective from the point of view of on-chip space occupation. The full structure along with all the advantages will be described in details in Chapter 4.

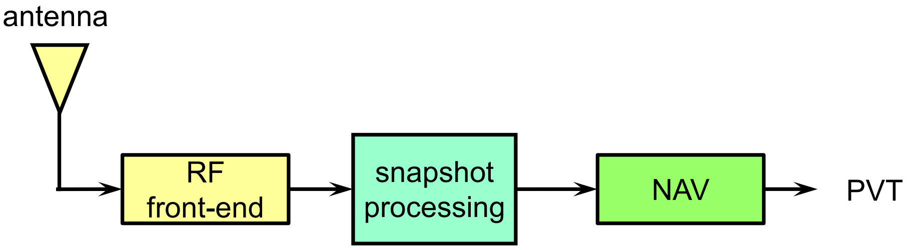


Figure 3.6: Consumer GNSS receivers structure.

3.2 Mass-market design drivers

All mass-market receivers share some design features, directly coming from the user needs. Three main drivers were identified.

Time To First Fix (TTFF)

The TTFF is the time a user has to wait before his device provides a position fix since it is powered on [34]. In other words, the time that a receiver takes to acquire and track a minimum of four satellites, and to obtain the necessary information, from the demodulated navigation data bits or from other sources. It is probably the most important driver in the mass-market design.

The TTFF strictly depends on the architecture of the receiver (such as the number of correlators or the acquisition strategy), on the availability of assistance data (see Section 3.3), and on the broadcast navigation message structure. Consumer producers exploit a base band configurable processing unit, with thousands of parallel correlators generating a multi-correlator output with configurable spacing, depending on the accuracy required [35, 36]. By selecting an appropriate number of correlators, depending on the available assistance data and on the accuracy required, the TTFF varies.

More details on the TTFF along with some results for a real commercial device will be presented in Section 5.9.1.

Sensitivity

Hostile environment is the general word commonly used to describe those scenarios in which open sky and ideal propagation conditions are not fulfilled (see Section 3.3.1). It can include for example urban canyons, where the presence of high buildings limits the SV visibility and introduces multipath, denied environments, where unintentional interference may create errors in the processing, or sites where shadowing of LOS path is present, for example due to trees, buildings and tunnels. In these situations it is necessary to pay particular attention to the signal processing stage; the performance is in general reduced up to the case in which the receiver is not able to compute a fix. More details are provided in Section 5.9.3.

Power consumption

Especially in the case of chips suited for portable devices, running on batteries, the power drain represents one of the most important design criteria. In order to reduce at maximum the power consumption, the different chip manufacturers adopted different solutions. Most of them are based on the concept that, contrarily to a classical GNSS receiver, a mass-market receiver is not required to constantly compute a PVT solution. In fact, most of the time, GNSS chipsets for consumer devices are just required to keep updated information on approximate time and position and to download clock corrections and ephemeris data with a proper time rate depending on the navigation message type and the adopted extended ephemeris algorithm. Then, when asked, the receiver is able to provide a position fix in a short time interval. By reducing the computational load of the device during the waiting mode, the power consumption is reduced proportionally. A complete analysis of power saving architectures is reported in Section 4.3.

3.3 The concept of GNSS assistance

The surplus value of a commercial GNSS chipset is represented by the possibility to exploit assistance information. A-GPS and in general Assisted GNSS (AGNSS) are techniques aimed at improving standard GNSS performance, by providing some information that the receiver would otherwise have derived from the received navigation signal through an alternative communication channel [37]. They ease the task of receiving and processing the SIS. A-GPS capable receivers can minimize time and resources needed for performing a PVT and process much weaker signals that unassisted devices. For example, GNSS receivers' acquisition process can be sped up if a-priori estimates concerning observables such as frequency offset, accurate time, code delay and receiver position are known.

This is widely used also in consumer devices for reducing the search space, as will be detailed in Section 4.1.4. In this case, the location server and cell base station provide assistance data, used by the A-GPS receiver to reduce the frequency and code-delay search. As an example, in Figure 3.7, the full acquisition search space is indicated in dark grey, while the reduced search range is indicated by the white rectangular boxes, defined in the received A-GPS message. The desired satellite signal is found in the intersection of the two boxes. The more precise the assistance data, the narrower the search range, the faster the acquisition time. In particular, assistance always provides some reduction in the frequency search, but there is a reduction in the code-delay search only if fine-time assistance is available.

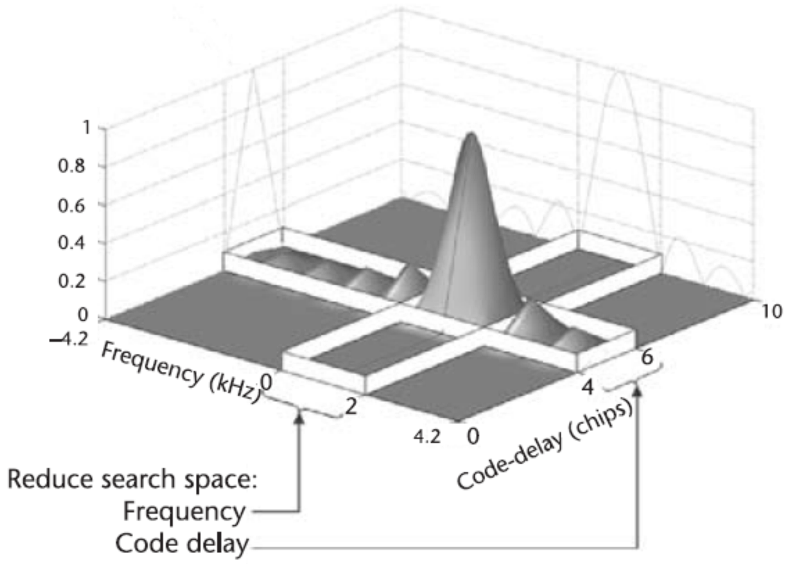


Figure 3.7: Overview of the assisted GNSS principle [37].

Most of the assisting information are generated in a base station, exploiting a high-end receiver. Then the user receiver exploits the assistance data for the following purposes:

- **satellite almanacs** to determine which SVs are in view, at the approximate time and location of the user;
- **signal Doppler** and **Doppler rate** to reduce the frequency acquisition search space and to make the acquisition process more robust to high dynamics, when large coherent integration times are employed;
- accurate **frequency reference** to calibrate the local oscillator;
- accurate **time**, to reduce the code delay acquisition search space;
- **satellite ephemeris data** to determine the satellite positions, required for pseudo-ranges generation;
- **navigation message** to remove the data bits from the received signals, thus allowing extension of coherent integration time beyond the data bit duration;

- **satellite clock errors** to correct the pseudo-ranges computation;
- **differential corrections** to reduce positioning errors due to ionospheric range errors and satellite clock errors.

3.3.1 Assistance requirements in hostile environments

Mass market receivers are often required to work in hostile environments, represented by all those scenarios where open sky and ideal propagation conditions are not fulfilled. Several factors can threaten their performance, like multipath, low visibility of the satellites and weak signal reception. They generally include:

- **urban scenarios**, such as urban canyons in which the satellites visibility is limited by high buildings, and in which heavy multipath threats the correct receiver operations;
- **harsh environments**, where unintentional or intentional interference degrades the received signal;
- **indoor scenario**, where the signals are received with a very low C/N_0.

If assistance and high performing signal processing techniques are not employed, performance can be reduced up to the case in which navigation is not possible. In recent years, many research centres and receiver manufacturers have focused their attention on the investigation and development of High Sensitivity (HS) solutions and signal processing techniques, which can improve robustness and sensitivity in such denied environments.

The main problem to address in these situations is the reduced received power. As it is well known, HS standalone receivers are based on the extension of the coherent integration time T_c. The longer the coherent integration time between the local and the received signals is, the better the de-spreading gain (i.e. Signal-to-Noise Ratio (SNR) improvement) that can be obtained after the correlation process. However, the presence of unknown data bit transitions limits the value of T_c to the data bit duration T_b ($T_\mathrm{c} \leq 20$ ms as for GPS L1 C/A signal, while $T_\mathrm{c} \leq 4$ ms as for Galileo E1B OS signal). The presence of data bits, which introduce sign reversals within the integration window, may cause a partial or even total cancellation of the correlation power. It is well known that, in order to overcome the issue of data bit transitions, non-coherent accumulations can be performed [37]. Non-coherent accumulations are obtained non-coherently summing several coherent integrations. Such a technique is insensitive to bit transitions but suffers the squaring loss issue [37].

Furthermore the sensitivity improvement obtained by increasing the integration time has to be traded-off with the following factors.

1. The increase of the computational complexity, due to the reduction of the acquisition Doppler step width, which leads to an increased search space size, and thus to a larger Mean Acquisition Time (MAT) and TTFF.
2. The incoming Doppler frequency uncertainty and clock stability: significantly longer integration intervals are undesirable as they would require precise knowledge of the incoming Doppler and a very stable frequency reference.
3. Long search periods, which require the use of large banks of correlators and digital signal processing to search for GNSS signals very quickly.

Additionally, an extremely accurate Doppler aiding would not only help in reducing the Doppler search domain, thus decreasing the MAT, but makes the acquisition and tracking stages more robust to high dynamics.

Most of the problems outlined above can be faced with particular signal processing techniques and exploiting external aiding, such as in the case of AGNSS. An assistance requirements analysis is presented in the following of this section.

Data transition The presence of navigation message data can be unsuitable when dealing with denied environments, for different reasons:

- navigation data demodulation cannot be achieved when processing very feeble GNSS signals;
- the data bit length limits the coherent integration time extension, since data bit transitions introduce shifts in the carrier phase leading to signal cancellation.

Thus, an assistance system providing navigation data content is needed in order to perform the data wipe-off at receiver level and to solve both problems. At the same time, pilot signals, offered by modernized GNSSs, allow for longer integration, but secondary code synchronization is required.

Often it is not possible to perfectly wipe-off navigation data, for example because of the limited accuracy of the precise time information coming from assistance. In order to assess the impact of the assistance timing error, a set of semi-analytical simulations, emulating different levels of timing accuracy, has been performed. An Full Educational Library of Signals for Navigation (N-FUELS) GPS L1 C/A simulated signal is considered in this example, characterized by a data bit transition every 20 ms (in the worst case). A coherent integration equal to 20 ms is performed on the input signal. The navigation message information coming from an assistance server is emulated, and a delay τ_d, in the range 0 to 10 ms, is artificially introduced before performing wipe-off. In the ideal case of $\tau_d = 0$ ms the wipe-off is perfect and all the signal power contributes to the correlation peak, while in the worst case $\tau_d = 10$ ms the phase change at half of the integration period completely cancels the signal. Figure 3.8 depicts the correlation SNR during a simulation for different values of the assistance timing error and for different values of signal power, confirming the theoretical assumptions. The SNR is computed as the ratio between the magnitude of the peak and the noise floor standard deviation. As long as the timing error increases, both the magnitude of the peak and the noise floor level decrease, with different trends. In the ideal case (no noise, black line), the SNR is constant until an error of about 8 ms; after this value the SNR starts falling to zero. On the contrary, for real signals, with C/N_0 equal to 50 and 40 dBHz respectively, the SNR decreases much faster as long as the error increases. In the latter case, an error equal to 7 ms already completely cancels the correlation. After this value, the correlation peak is completely buried in the noise floor.

Doppler It is well known that Doppler effect has to be taken into account in order to properly process GNSS signals. A fine carrier wipe-off is required in a GNSS receiver to estimate the signal propagation time, and therefore an accurate Doppler frequency estimation has to be performed.

The unknown Doppler frequency range at the receiver is caused by the combination of three different factors: the Doppler shift from the satellite, from the receiver motions and any frequency offset in the receiver reference oscillator. In static conditions the maximum Doppler frequency is equal to 4.880 kHz for GPS 4.148 kHz for Galileo, as proved in Appendix 7. For terrestrial applications, receiver motion is very small compared to the satellite speeds, so its effect is normally negligible. There is up to approximately 1.5 Hz of Doppler frequency for each 1 km/h of receiver speed [37]; in other words, a receiver motion of 130 km/h extends the Doppler frequency of about 4%. It has to be mentioned that residual Doppler may be still present due to the finite accuracy of the local oscillator frequency, which usually differs from the nominal value of an amount related to the adopted oscillator technology. An additional 1.575 kHz of unknown frequency offset for each 1 ppm of unknown receiver oscillator offset has to be considered. Consumer receivers usually

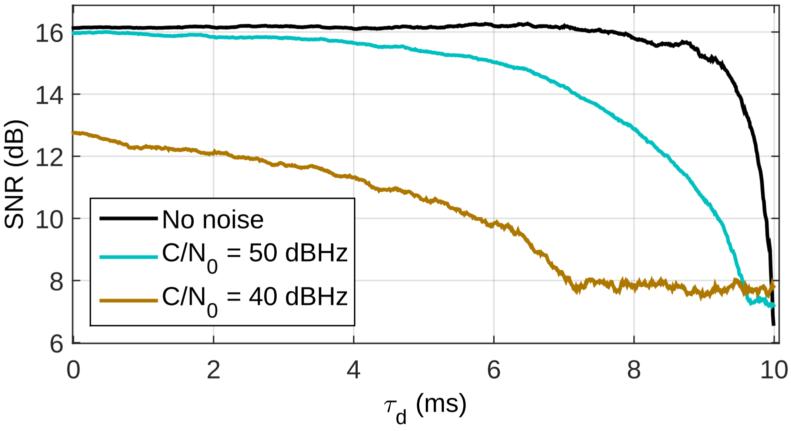


Figure 3.8: Correlation peak magnitude versus assistance timing error.

have Temperature Compensated Crystal Oscillator (TCXO) oscillators with offsets of a few ppm. Results are summed up in Table 3.1.

Table 3.1: Summary of Doppler contributions.

Contribution	GPS	Galileo
Satellite motion	±4.880 kHz	±4.148 kHz
Receiver motion	±1.5 Hz/km/h	±1.5 Hz/km/h
Oscillator offset	±1.575 kHz/ppm	±1.575 kHz/ppm

Further details about Doppler requirements for very low signal power can be found in [38].

Doppler rate Standard GNSS receivers tracking architectures are based on the assumption that the maximum change in the signal Doppler frequency is small, if compared to the inverse of the integration time. The impact of Doppler rate is considered negligible and the Doppler frequency is assumed constant, over a single measurement epoch [39]. In fact, the magnitude of rate of change of a GNSS satellite's relative Doppler frequency is usually relatively small, if compared to the Doppler itself, as confirmed by the results obtained in Appendix 7. For a static user, it can increase up to 0.94 Hz/s for GPS and to 0.64 Hz/s for Galileo, depending on the satellite elevation and on the user's latitude. In the dynamic case, for normal user velocity (up to 150 km/h), it never exceeds 0.98 Hz/s for GPS and to 0.69 Hz/s for Galileo, as proved in Appendix 7. However, when either the receiver velocity or the coherent integration time increase up to non-standard values, the satellite Doppler rate becomes potentially significant and knowing only the Doppler might be useless. The capability of providing the GNSS receiver with an accurate Doppler rate aiding allows the local carrier generator to follow the Doppler variation over time, thus making the acquisition receiver more robust to the high dynamic. Further details about Doppler rate requirements for very low signal power can be found in [38].

3.4 Typical user scenarios

Several different scenarios are defined in order to test the different algorithms proposed, under different aspects and conditions. First, standard AWGN channel is considered, representative of ideal open-sky conditions. Then a Land Mobile Satellite (LMS) model is introduced, adding multipath and fading, representative of harsh environments, such as urban canyons.

3.4.1 AWGN channel

In the first implementation phase of an algorithm it is very useful to deal with signals where no noise is present; despite being a non-realistic condition, it helps in debugging the algorithms and understanding strong and weak points; Then it is possible to move towards non ideal conditions, adding AWGN, changing the code delay, the Doppler frequency and its trend, changing the modulation (from simplest ones like GPS L1 C/A BPSK to more complex ones, as Galileo CBOC), adding navigation data, and so on. In this specific case TTFF (Section 5.9.1), sensitivity (Section 5.9.3) and performance of the joint GPS and Galileo solution have been evaluated, exploiting the capabilities of a signal generator to generate signals with different power levels.

3.4.2 Multipath fading channel

Clearly AWGN conditions are not realistic, because in the real word the signal is affected by several impairments, such as multipath, interference, fading, light indoor and indoor scenarios, low C/N_0, as depicted in Figure 3.9.

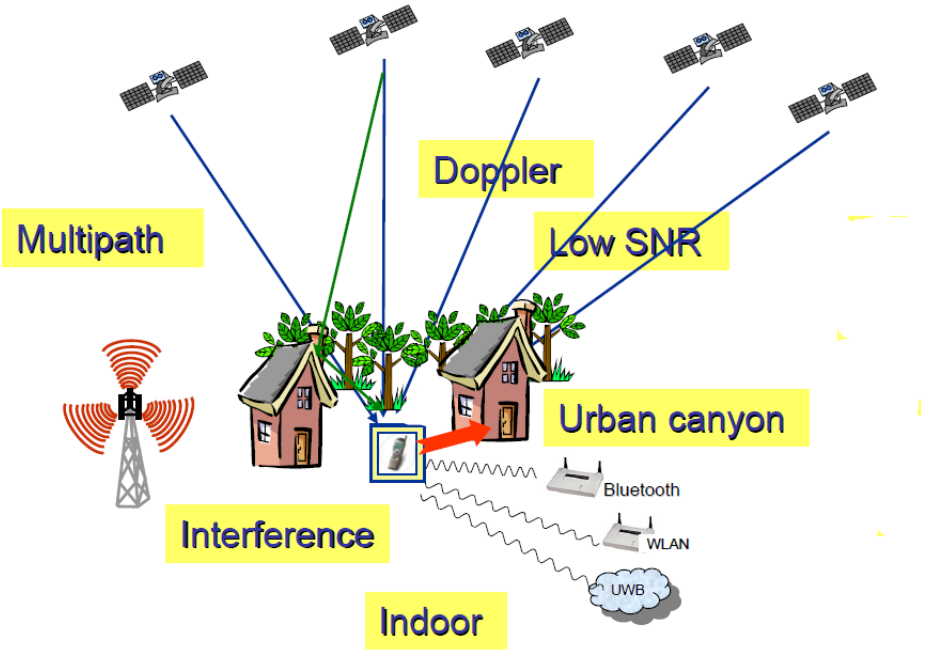


Figure 3.9: The hard work of a GNSS receiver.

In particular, multipath fading has been considered in this work; the definition of the multipath fading channel is based on the model proposed in [40]. This model starts from the statistical model of the LMS channel proposed by [41], whose main characteristics are the following.

- A time variant component which accounts for the slow variations in time of the LOS component due to shadowing and blockage effect. The transition between each state is modelled by a first order discrete time Markov chain. Each state is characterised by a state probability (probability to stay in that state), transition probability (probability to transition to a different state), set of parameters of the Loo distribution as well as the state frame length. The latter accounts for the validity length of each state, or state duration - that is the frame length for which a new state needs to be derived. This depends on the user environment (dynamics, and conditions).

- The signal variations within each state are modelled by a Loo distribution within each state (i.e. for each slow variation conditions). The Loo distribution considers the sum of a log-normally distributed random phasor and a Rayleigh phasor and it is quite useful as it comprises the Gaussian, Rice and Rayleigh distributions as particular cases, hence covering a wide range of applications, from LOS conditions to deep shadowing.

- A time dispersion factor. In fact, the multipath phenomenon not only affects the signal amplitudes but also induces time dispersion depending on the delays of the different replicas. This can be represented by the different instantaneous Power Delay Profiles (PDPs) that account for the fast variations of the signal for the wideband channel.

The new model proposes the following modification.

- The LMS channel is considered to be narrowband and therefore the time dispersion factor is neglected, as all signal variations are now represented by the Loo distribution.

- Instead of considering the 3-state Markov model (LOS, shadowing, deep shadowing), only two states are considered: good and bad, representing a range of LOS-to-moderate shadowing and moderate-to-deep shadowing, respectively. In addition, a semi-Markov chain approach was also considered, where the frame length follows a log-normal distribution.

This new model reports a better fit of this model than the previous one, with respect to the same set of experimental data. This model provides a time series of the receiver signal level that should be read as variations with respect to the LOS level.

3.4.3 Multi satellite multipath fading channel

In particular the implementation of the last version of the two-state LMS simulator, described in [40], is able to generate different but correlated fading for each single SV, according to its elevation and azimuth angle with respect to the user position: especially the angular separation within satellites is crucial, since it affects the correlation of the received signals. This approach is based on a *master-slave* concept, where the state transitions of several slave satellites are modelled according to their correlation with one master satellite, while neglecting the correlation between the slave satellites. The nuisances generated are then imported in the simulator scenario, to timely control phase and amplitude of each simulator channel. LMS time series are generated with a Matlab routine and exploited to assess performance in harsh environments. Figure 3.10 shows an example of power level as generated by this model and used in the simulations.

By using a Matlab implementation of this model provided by ESA, it was possible to generate time series of the signal variations, assuming a certain user velocity of 1 m/s or 50 m/s) and a correlation of about one eighth of the L1 wavelength, herein 0.025 m. The amplitude and phase time-series are then imported in a scenario of the hardware signal generator, to timely control phase and amplitude of each simulator channel.

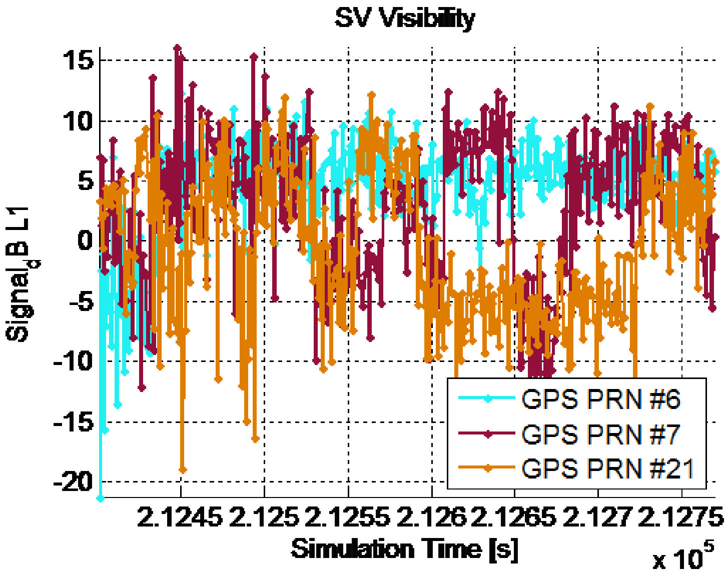


Figure 3.10: Example of different signal correlated powers generated according to the LMS multi-sat model.

3.4.4 Simulator 3GPP scenario

A first attempt to model the hostile scenarios described in Section 3.3.1 has been introduced in the 3GPP standard [42] together with the definition of AGNSS minimum performance requirements for user equipment supporting other AGNSSs than GPS L1 C/A, or multiple AGNSSs which may or may not include GPS L1 C/A. The standard test case supports up to three different constellations and in dual constellation case it foresees three satellites in view for each constellation.

In order to perform TTFF and sensitivity tests, applying the 3GPP standard test case, a GNSS Spirent simulator scenario has been configured with the following characteristics, starting from the nominal constellation:

- 6 SV, of which 3 GPS (with PRN 6,7, 21) and 3 Galileo (with code number 4, 11, 23);
- Horizontal Dilution Of Precision (HDOP) in the range 1.4–2.1;
- nominal power as per corresponding SIS Interface Control Document (ICD);
- user motion, with a heading direction towards 90° azimuth, at a constant speed of 5 km/h.

In addition to the limited number of satellites, a narrowband multipath model, as described in Sections 3.4.2 and 3.4.3, has been introduced. The Multi-SV two-states LMS model simulator has been used to generate fading time series representative of an urban environment.

Chapter 4

Mass-market Algorithms: State-of-the-art and Innovative Techniques

As outlined in Chapter 3, consumer devices are gaining an important role even in the GNSS scientific community. Notwithstanding, a limited documentation concerning their architecture, algorithms and techniques is present in the open literature; besides a few papers, all the know-how is protected by patents, held by the main manufacturers, and mainly focused on the GPS L1 C/A signal.

The architecture described in Section 3.1.1 is considered for analysis and implementation in all this part. Particular focus is given to the estimation of the code delay and of the Doppler frequency affecting the GNSS signal, which is indeed the most important and challenging operation preceding the navigation solution computation.

In Section 4.1 a complete overview on core algorithms for parameters estimation is given, first describing state-of-the-art techniques, then proposing extensions for Galileo OS signals and tailoring them for a software implementation, with final important improvements in performance. In particular, Section 4.2 presents an innovative frequency estimation algorithm. Then, in Section 4.3 power saving and on demand processing are described.

4.1 Multi-correlation snapshot processing

BB algorithms consist of all the GNSS receiver blocks processing data at BB level. They operate on the IF digital signal at the front-end output. The signal is further down-converted at BB by removing the known digital IF frequency, by the process called carrier wipe-off; only the Doppler frequency affecting the GNSS signal and a component due to the non-ideal clock remain as a residual sinusoidal component. Then estimates of the signal code delay and Doppler frequency are computed for each signal in view, along with other intermediate measures useful to evaluate the signal quality (such as the C/N_0 – Section 2.8) or to increase the accuracy (such as the carrier phase). At the same time the navigation message is demodulated.

4.1.1 Open-loop vs. closed-loop

It is well known that the GNSS signal processing performance characteristics (such as accuracy, reliability and robustness) are mostly influenced by the receiver design choice between two architectures:

- traditionally applied **sequential** signal tracking techniques, that implement a sequential **closed-loop** tracking solution

- **snapshot** processing techniques that allow for **open-loop** processing implementations

It is important to mention that closed-loop sequential signal processing is well recognized while open-loop batch processing techniques remain underutilized, despite it has been proved that for a number of practical applications, the use of these techniques efficiently overcomes limitations of the traditional approach [33].

Closed-loop sequential tracking

Figure 4.1 shows a generalized architecture of the sequential tracking stage. It is easy to see that the sequential tracking approach relies on a closed-loop tracking architecture. The sequential tracking stage performs sequential correlation of the incoming and replica signals; the correlator outputs are used to compute raw error signals. Particularly, the code phase error can be computed by using, for example, the early-minus late discriminator while a Costas discriminator can be applied to compute raw value of the carrier phase error. Raw error signals are first smoothed by loop filters that can be implemented as low-pass filters. Smoothed error measurements serve as error signal estimates for the code phase, Doppler frequency shift and carrier phase. These estimates are applied to adjust parameters of Numerical Controlled Oscillators (NCOs) that generate replica signals, with a feedback procedure. The replica code phase, carrier Doppler shift and carrier phase serve as the actual tracking loop measurements.

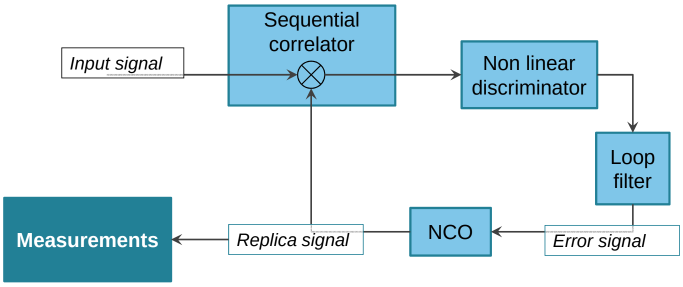


Figure 4.1: Generalized architecture of sequential/closed-loop tracking.

Open-loop batch processing

Figure 4.2 represents a generalized open-loop receiver architecture that utilizes batch processing. Contrary to the sequential closed-loop tracking, batch processing maintains an open-loop architecture. The input signal is correlated with batches of replica signals that are constructed based on the code and frequency search spaces. The correlation is a batch-based correlation and joint time-frequency domain techniques allow parallel correlation computations. The result of the batch-based correlation process is the 3D signal image for the current data batch, where the dimensions are the code delay shift, the Doppler shift and the signal energy. This result is similar to the standard acquisition search space, except for the fact that it usually represents the sub-portion of it containing the signal peak (reduced code and frequency search space), and by the fact that it is obtained exploiting different techniques.

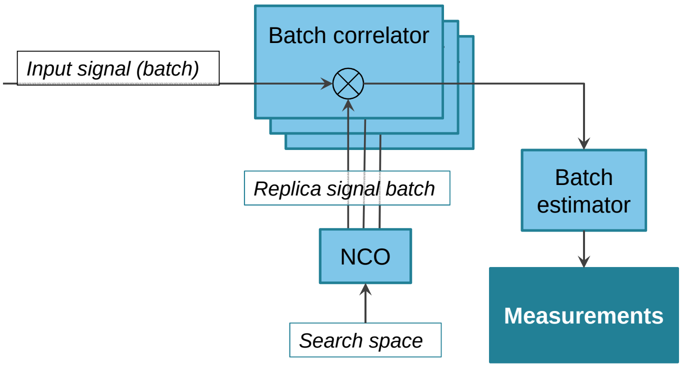


Figure 4.2: Generalized open-loop receiver architecture [33].

According to [33], the main features of the open-loop batch processing approach are summarized as follows:

- improved signal observability as compared to sequential processing;
- capability of parallel computations;
- improved robustness as compared to closed-loop sequential tracking.

In an open loop, a measurement of a parameter at time $n + 1$ may be expressed as

$$P(n+1) = f\left(u_1(n+1), u_2(n+1), \ldots, u_N(n+1)\right) , \tag{4.1}$$

where $u(n)$ is the measurement of a quantity related to P at time n. Thus, a current measurement of P is a function of a finite number of related measurements. In the previous equation, thanks to the statistical independence properties described above, there is no explicit dependence of $P(n+1)$ on $P(n)$ (or $P(n-1)$, $P(n-2)$, etc.). In an open-loop situation, prior measurements, such as $P(n)$, $P(n-1)$, or $P(n-2)$, may just be used to influence the calculation of subsequent measurements. Typically, a prior measurement is used to provide a constrained range, or "window", within which a succeeding measurement may be taken. The location of the centre of the window and/or the width of the current window are typically adjusted as a function of the prior measurements.

For these reasons, in this work only open-loop strategy is investigated, because, despite a higher jitter, it offers improved sensitivity, higher robustness against fading and interference and better stability. Moreover open-loop is described in several Qualcomm patents and is probably widely used in chipsets for mobile devices.

A similar approach, denoted quasi-open-loop architecture, is described in [6]. In this work a traditional tracking structure is used, but instead of updating the NCO at each coherent integration time as in the closed-loop architecture, the update is performed after a certain number of epochs. So, between two updating epochs, the architecture works in open-loop. It is a sort of three-rate scheme, unlike the conventional two-rate scheme; this additional degree of freedom helps in the design of the traditional loop filter parameters, especially in weak signal conditions.

Parameters estimation options

To obtain an estimate of the signal parameters, the code delay τ and the Doppler frequency f_D, one of the following search strategies can be used, each one corresponding to a different search space size and definitions:

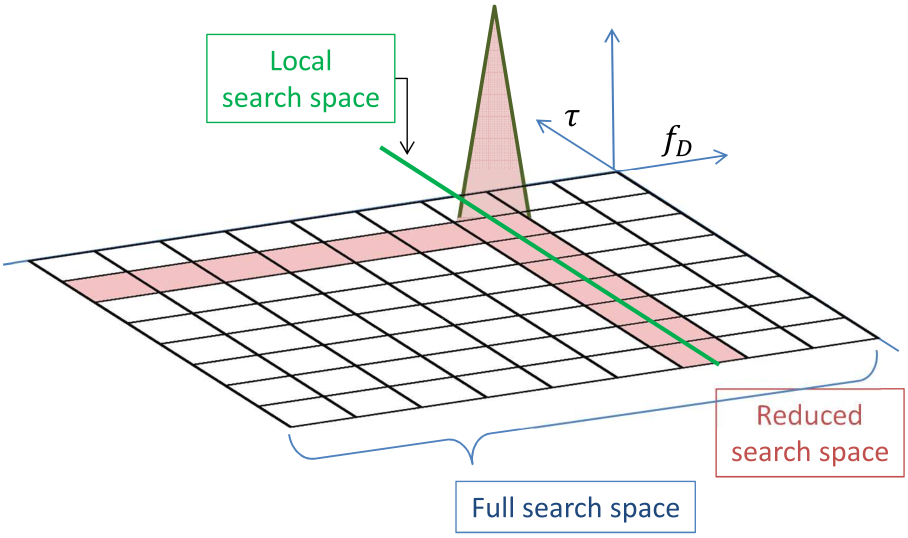


Figure 4.3: Comparison between different search spaces and parameters estimation strategies.

1. **Full search**. When there is no prior knowledge of the signal parameters, a full search has to be performed. This is usually performed during the standard acquisition procedure (as described in Section 3.1.1). All the possible values of the code delay (the full code length) and Doppler frequency (in static conditions, the range −4.88 to 4.88 kHz for GPS, and −4.148 to 4.148 kHz for Galileo) are evaluated, as depicted in blue in Figure 4.3. If the signal is present with a good C/N_0, then the detection probability is high; however, the computational load is large.

2. **Local search**. When rough signal estimates are available, a local search can be performed. This is the case of the traditional closed-loop tracking architectures: the estimates coming from the acquisition are refined and then used to feed the tracking loops. Successively, each epoch's measurement is used to control the following epoch, exploiting the loop feedback. Only a line of the search space is considered, as depicted in Figure 4.3, corresponding to one chip in the delay domain and to a few Hz in the frequency domain. This approach is fast and cheap, but not very robust, since a small variation in the signal dynamic or an increase of the noise level can push the estimates far from the true values. In this case a Loss of Lock (LOL) happens and a new acquisition is required.

3. **Reduced search**. When signal estimates are available, the reduced search can be performed, as an alternative to the local search. This is the case for example of signal estimates coming from AGNSS data or from previous measurement. In this case, sub-portions of the full search space are considered, as shown in pink in Figure 4.3. The signal parameters can correctly being estimated with a reduced computational load, but only if the correlation peak falls within the considered cell, otherwise an *invalid measure* is obtained. It is important to note that each measurement is performed in open-loop and, since no feedback is present, the

estimates of the signal parameters for the current data batch are statistically independent from previous estimates. Statistical independence means that, contrary to the closed-loop local search solution, errors in the current data batch estimates are statistically independent from errors from the previous data batch. What is more, the accuracy and the peak SNR do not depend on the search space dimensions. This is commonly referred to as snapshot processing. The code delay and frequency estimation algorithms described in the following of this section are based on this approach. Section 4.1.4 gives further details on the reduced search space technique, in the light of the estimation techniques described.

4.1.2 Code delay estimation

Code delay estimation is the first task of a GNSS receiver. The information contained in the signal code delay is essential for computing the pseudo-range and thus the PVT, and has to be accurately extracted. The code delay is estimated exploiting the auto-correlation properties of the PRN codes, as described in Section 2.3. For this reason code delay estimation is also referred as "correlation".

The Broadcom solution

One of the most interesting and powerful BB algorithm to estimate the code delay in a mass-market device is described by the following Broadcom patents, by *Charles Abraham*:

- "Method and apparatus for performing signal correlation" US patent 7 190 712 B2 [35];
- "Method and apparatus for performing signal correlation using historical correlation data", US patent 6 819 707 [36].

It has to be noted that a consistent number of Broadcom patents are focused on this receiver unit signal processing algorithms. The signal is processed in open-loop by a BB snapshot processing algorithm; a multi-correlation function, with a certain sampling frequency, is obtained, allowing the estimation of the code delay and the Doppler frequency affecting the signal.

According to [35], the baseband processing unit can be split into three main components:

1. the receiver unit, interfacing the front-end (Section 3.1.1) and in charge of generating correlations. It is based on an array of digital channels able to process signals of different constellations;
2. the co-processor unit, integrated with the receiver, able to estimate signal parameters from the correlation results provided by the receiver. These parameters are estimated after reception of commands issued by the Central Processing Unit (CPU);
3. a CPU accessing the mobile device memories, sensors and I/O circuits and in charge of generating satellite measurements and position.

A multi-correlation unit

The multi-correlation snapshot unit provides all the circuits aimed at providing the carrier wipe-off, the code and carrier NCOs and the correlation values. The output of receiver front-end is coupled to a set of processing channels, shown in Figure 4.4; each channel is dedicated to a different component $\tilde{y}_{ij}(t)$ of the SIS. Each signal is tuned by an NCO, removing the residual frequency component, and generating a BB signal, consisting of In-phase (I) and Quadrature (Q) components.

Some of these operations are similar to the ones performed in a conventional receiver tracking stage; on the contrary, the main differences are outlined below.

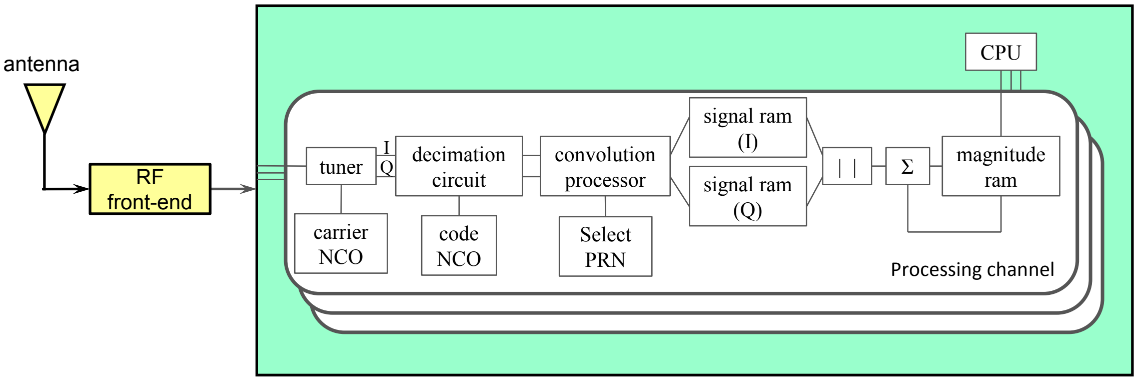


Figure 4.4: BB receiver unit channel structure.

1. The number of correlation values that can be provided is hundreds of times greater. Standard tracking stages usually provide 3 correlation values per channels (the Early (E), the P and the Late (L)), while professional receivers can provide tens of correlators. The use of multiple correlators allows to better analyse the shape of this signal, enabling more advanced techniques as well as strong signal impairments countermeasures, such as multipath rejection capability, which is an essential feature in mass-market receivers.

2. The convolution processor is configurable in terms of:

   (a) number of correlators;

   (b) spacing between correlators;

   (c) integration time.

   Therefore it can provide correlations over a time period ranging from fractions of a primary code epoch up to multiple code epochs, which can be coherently or non-coherently integrated.

3. There is no code generation unit; the code NCO is controlling a decimation circuit that acts before de-spreading with the PRN code. There is no longer the necessity, as in standard tracking loops, to create, for any epoch, a P, an E and a L delayed replica of the local code; the code chip values can be directly read from memory. This approach is particularly appealing in the case of memory codes, as the ones adopter in Galileo. In general The complexity of the unit and the computational burden are reduced

4. Each channel can access directly to its own Random Access Memory (RAM) memory bank where it can store I and Q correlation values.

Afterwards the signal is decimated, by summing the samples over a time interval of half a chip or less. This time is controlled by the code NCO whose rate accounts for the nominal chip rate and the Doppler effect. If the sampling frequency and the decimation frequency are incommensurate, the number of samples summed usually toggles between two values, so that over the long term the correct sample timing is maintained. The decimation factor P determines the decimation frequency, which corresponds to the resolution of the search grid in the code delay domain, as well as the distance between E and L replica code pairs using a standard receiver terminology. In particular, the patents suggest three different resolution modes for processing legacy GPS L1 C/A signals, depending on the assistance information provided to the channel:

- **STANDARD resolution** ($P = 2$) corresponding to 1/2 chip spacing, when no a-priori information about the code delay is provided, giving a rough estimate of the code delay;
- **HIGH resolution** ($P = 5$) corresponding to 1/5 chip spacing, when a rough estimate of the code delay is available, giving an estimate of the code delay;
- **SUPER HIGH resolution** ($P = 10$) corresponding to 1/10 chip spacing, when an accurate estimate of the code delay is available, giving an accurate estimate of the code delay.

The values that can be set for these parameters are limited since they impact the subsequent units and other parameters of the convolution processor.

After this decimation circuit, both the I and Q signals are divided into segments of K chips (KP samples), that are in input to the signal shift registers inside the convolution processor, as shown in Figure 4.5.

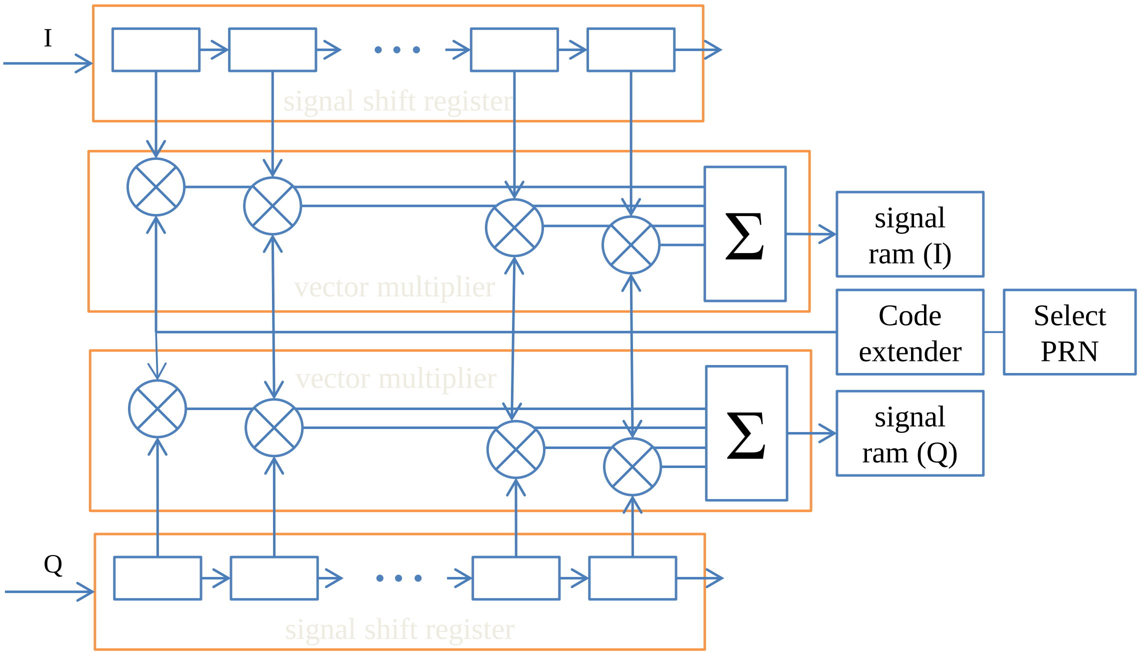


Figure 4.5: Convolutional processor for BB processing.

The convolution processor is a configurable unit that is able to compute in parallel sub-correlations that are then accumulated. This unit works at a higher clock speed, and this allows the computation of many correlation values in parallel, namely KP. More in detail, the signal remains in place in the shift register for many clock cycles so that a series of partial correlations can be performed; KP parallel sub-correlations are computed L times. Each signal vector coming from the shift register is multiplied with a selectable segment of the local reference code, chip-wise stored in a Look-Up Table (LUT). When all the partial correlations are completed, the signal registers are shifted by one sample. By properly combining partial correlations that correspond to a particular time delay, the full correlation can be computed. This is discussed in details in Section 4.1.2.

It has to be noted that the only operation that has to be performed on the memory code is the code extension. The values of K, P and the convolution processor clock are dimensioned according to the reduced search space for the signal detection, according to the following criteria:

- K must be an integer factor of the primary code length (1023 for GPS L1 C/A, 4092 for Galileo E1BC) for example 11, 33, ...
- The time needed by the processor to perform a vector multiplication is essentially given by the reading time of the code segment in the signal RAM, the writing time of the correlation in the memory and the time needed for multiplication and sum that is negligible. The memory access time is around one clock time of the convolution processor, so the total time needed is equal to two clock times ($2T_{\text{clk}}$).
- If no prior information on the signal parameters are present (equivalent of a cold-start acquisition), then all the possible shifts in an entire primary code epoch have to be considered. This leads to the need of performing $M = T_p/K$ partial correlations with consecutive PRN code segments during the reception time of a signal sample, where T_p is the length of the code in chip. The corresponding equation is

  $$M2T_{\text{clk}} = \frac{1}{PR_c}\,, \tag{4.2}$$

  where R_c is the nominal chip rate of the received modulation.

Introducing, f_{clk}, the clock frequency of the convolution processor, the reciprocal of T_{clk}, (4.2) can be rearranged as

$$f_{\text{clk}} = 2PMR_c\,, \tag{4.3}$$

where the clock is a function of the P and K parameters only.

M determines the extension of the code delays that can be searched in parallel. In the case in which a reduced search space is considered, as described in Section 4.1.1, it can be set to a lower value (and denoted L). Thus only a subset of the full convolution is generated. As depicted in Figure 4.6 in the case of GPS L1, this portion contains the signal correlation peak.

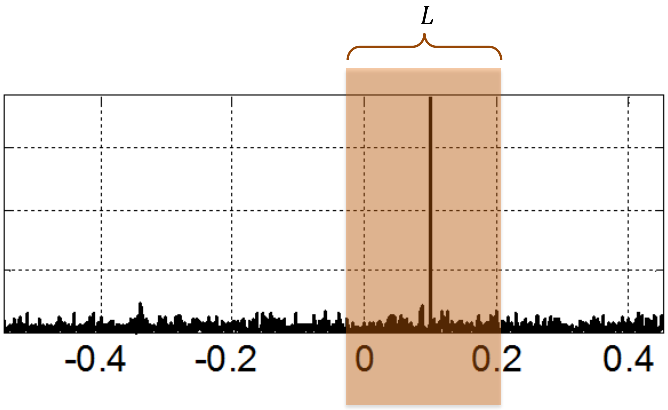


Figure 4.6: Example of code delay search space reduction to a portion of L samples.

Examples of parameter settings for GPS L1 C/A are reported in Table 4.1. The main constraints are:

- Keep the f_{clk} lower than a maximum value around 130 MHz (it is eventually possible to increase this value with new chip technologies; however this could impact on power consumption).

Table 4.1: Parameters for the BB processing sub-correlations for GPS L1 C/A.

R_c (MHz)	P	K	PK	L	f_{clk} (MHz)	Resolution	Parallel complex correlators (PKL)	Correlators spacing
1.023	2	33	66	31	127	Standard	2046	1/2 chip
1.023	5	11	55	12	123	High	660	1/5 chip
1.023	10	3	30	6	123	Super high	180	1/10 chip

- Keep the dimension of the shift register (PK) around 60/70 taps (this could impact on the on-chip occupation).

What is more, by increasing P it is necessary to reduce K, in order to maintain the dimension of the shift register small enough. By reducing K, M must be reduced as well (due to f_{clk} constraint), and the entire code epoch cannot be covered any longer; this is the case in which a new parameter $L < M$ is introduced). By increasing the resolution, the number of equivalent parallel correlators decreases . In addition,the shift register length is not a configurable parameter, at least for a hardware implementation. It is dimensioned to the highest possible values (66 in the previous table) and when the resolution is changed the unnecessary delays are deactivated. Further increasing P can lead to problems with the ADC sampling frequency that generally in this kind of receivers is not higher than tens of MHz (around 16 MHz usually).

Furthermore it has to be noted that the coherent and non-coherent accumulations can be set to different values and the coherent one, in particular, can be also shorter than a signal primary code epoch (< 1 ms). This is particularly used when no information on the Doppler is available: by using two consecutive sub-code epoch correlations, just by looking at the auto-correlation function peak delay variation, the Doppler can be estimated. After this first estimation, the coherent integration time can be increased to improve the SNR. In order to reduce the processing speed and computational effort it is possible to use an even lower number of correlators, as long as sufficient accurate prior aiding information is present. For example, considering standard resolution mode, it is sufficient to keep $K = 32$ and to reduce the parameter L.

Extension to Galileo

The previous considerations can be extended to the Galileo signal, which is not considered in the Broadcom patents and in general in the open literature. In particular, adaptations to the Galileo E1B signal and to the BOC modulation are introduced in this section. In fact the BOC(1,1) modulation can be easily integrated in the code extender. The main differences with respect to the GPS BPSK case are:

1. the primary code epoch is 4 times longer (4092 chips instead of 1023 chips) and therefore the primary code epoch is 4 ms long instead of 1 ms;

2. the chip spacing in the acquisition can be kept to 1/2 chip only in case of narrowband, otherwise should be reduced to 1/2 slot, i.e. 1/4 chip in order to reduce the binning losses;

3. partial correlations over 1 ms (a quarter of a code epoch) are very likely in order to reduce the acquisition time (otherwise the frequency bin should be 4 times narrower than the GPS one).

Different implementations for the standard resolution mode are reported in Table 4.2: it is clear that the processing channel for the Galileo OS signal needs a more complex implementation with either a longer shift register or an increased clock frequency.

Table 4.2: Parameters for the BB processing sub-correlations for Galileo E1B.

R_c (MHz)	P	K	PK	L	f_{clk} (MHz)	Resolution	Parallel complex correlators (PKL)	Correlators spacing
1.023	2	33	66	124	507	Standard	8184	1/2 chip
1.023	2	66	132	62	254	Standard	8184	1/2 chip
1.023	2	132	264	31	127	Standard	8184	1/2 chip

The first combination is not feasible, due to the 4 times increased clock frequency. It is necessary to resort the sub-correlations techniques, with respectively a longer shift register and a shorted portion of the reference code. Thus, either an increased occupation of an higher frequency are necessary to deal with E1B Galileo signals. Furthermore, additional complexity should be added to the processing channel in order to demodulate both data and pilot components. To reduce the computational and hardware complexity, the data channel can be limited to the punctual correlation itself resulting so to an additional correlator, negligible with respect to the other 8 184.

In order to keep the same clock frequency, the third configuration is kept as a reference. Moreover parameters for the high and super-high resolution can be computed. Several degrees of liberty are present in the computation, such as the dimension of the shift register and the total number of correlators. In the design of the operation modes the, drivers chosen are the following:

- to keep approximately the same clock frequency;
- to keep a shift register dimension lower than the standard case;
- to keep the same L of the GPS case;
- to maintain the same proportionality in the correlator number passing from standard to high and super-high resolution modes with respect to GPS.

Table 4.3 reports the results of the computation. These values will be used during the testing phase. It is clear that other numbers are possible for different implementations.

Table 4.3: Parameters for the high definition BB processing sub-correlations for Galileo E1B.

R_c (MHz)	P	K	PK	L	f_{clk} (MHz)	Resolution	Parallel complex correlators (PKL)	Correlators spacing
1.023	2	132	264	31	127	Standard	8184	1/2 chip
1.023	5	44	220	12	123	High	2640	1/5 chip
1.023	10	12	120	6	127	Super high	720	1/10 chip

There are numerous interpolation techniques that can be used to estimate the true time delay using discrete correlation values. For example a least square estimation technique can identify parameters of a signal that best fit the noisy data. The correlation response takes the shape of a raised triangle. One benefit of the high resolution modes over the standard resolution mode is that the raised triangle correlation response is sampled at more than twice the points. Thus, high resolution mode can be used to reliably identify and measure correlation peaks that may not be properly discernible in standard mode. This can be very advantageous when processing low energy signals, especially in indoor scenarios [35]. The different modes can be achieved by dynamically changing the value of the parameters, K, P and L, to obtain the desired resolution. This represents one of the key points of the techniques. In fact, the number of correlator and the width of the code delay window can be dynamically changed according to the actual knowledge of the signal delay without changing the hardware of the receiver.

Post-correlation operations

Once the receiver unit has produced the correlation values, they must be elaborated to produce useful information concerning the code delay. In the original patent, this step is done exploiting a co-processor unit and a CPU. The co-processor unit is in charge of carrying out, upon the reception of an high-level command from CPU, all the needed basic operations based on I and Q correlation results provided by the receiver unit.

The co-processor is responsible for supplying several measurements:

- it locates the points of maximum correlation and return delay measurements;
- it can eventually exploit the correlation peak and the noise statistics to determine the correlation peak SNR;
- it can analyse the frequency by tracking the phase changes from sample to sample, though an averaging, weighted averaging or integration of the complex cross product. The complex cross product is defined as

  $$I_{n-1}Q_n - Q_{n-1}I_n \, , \tag{4.4}$$

  where n is a time index and I and Q are the in-phase and quadrature samples. After an initial frequency estimate is made, the frequency error may be removed from the I and Q correlation results using a complex modulator. By iterating several times this procedure, the frequency estimation process will converge.
- It can analyse phase changes from sample to sample to find the 180° degrees phase shift comprising the 50 bps navigation data stream. The bit transitions can be found by applying a threshold the complex dot product. The complex dot-product is defined as

  $$I_{n-1}I_n - Q_{n-1}Q_n \, . \tag{4.5}$$

Sub-correlation combinations

When working in high resolution and super high resolution mode, partial correlations are computed, i.e. only a sub-portion of the reference code is used. It is indeed necessary to combine partial correlations in a proper way, to build up the full correlations for each delay index, creating the full multi-correlation result. For sake of simplicity, but without losing generality, let's assume a scenario with a 6 chip code, with two samples per chip ($P = 2$), with $K = 2$ and $L = 3$. Therefore the whole code will consist of 12 samples, and will be denoted as `aabbccddeeff`, while the incoming signal will be denoted `AABBCCDDEEFF`. The size of the shift register is equal to 4 samples and for each shift, 3 correlations are computed. At each shift a new sample of the incoming signal enters at the right of the shift register, and a proper sub-portion of the reference code is chosen, as shown in Figure 4.7. In total 12 shifts are needed to produce a complete correlation over a code-epoch, and for each shift 3 parallel correlations are run. The results should combined according to the colours and the lines shown in the figure, in order to compute the full convolution. Each combination, shown in the right part of the figure (only the first 8 out of 12 are shown, but the procedure is the same for the remaining) corresponds to the correlation function point for a certain delay. It has to be noted that in the case of a 1023 chip code, such as the GPS case, with bigger parameters, the scheme is much more complex.

The architecture described in Figure 4.7 illustrates two important properties of the technique. First, it is apparent that less circuitry is required with respect to the matched filter case, usually adopted in standard receivers. Second, the code samples are accessed in fixed segments that are the same for each shift. Thus, a simple LUT can be used, without the necessity to provide code generation circuitry.

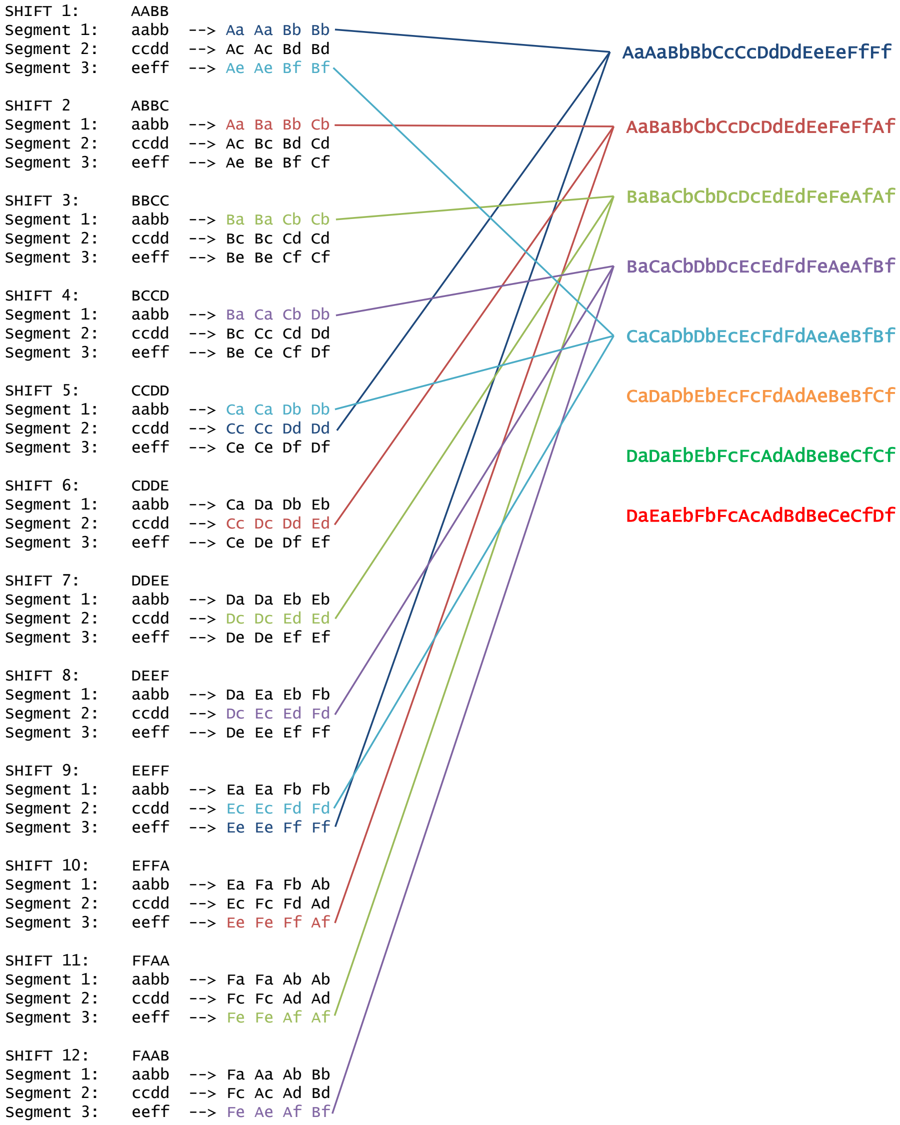

SHIFT 1: AABB
Segment 1: aabb --> Aa Aa Bb Bb
Segment 2: ccdd --> Ac Ac Bd Bd
Segment 3: eeff --> Ae Ae Bf Bf
SHIFT 2 ABBC
Segment 1: aabb --> Aa Ba Bb Cb
Segment 2: ccdd --> Ac Bc Bd Cd
Segment 3: eeff --> Ae Be Bf Cf
SHIFT 3: BBCC
Segment 1: aabb --> Ba Ba Cb Cb
Segment 2: ccdd --> Bc Bc Cd Cd
Segment 3: eeff --> Be Be Cf Cf
SHIFT 4: BCCD
Segment 1: aabb --> Ba Ca Cb Db
Segment 2: ccdd --> Bc Cc Cd Dd
Segment 3: eeff --> Be Ce Cf Df
SHIFT 5: CCDD
Segment 1: aabb --> Ca Ca Db Db
Segment 2: ccdd --> Cc Cc Dd Dd
Segment 3: eeff --> Ce Ce Df Df
SHIFT 6: CDDE
Segment 1: aabb --> Ca Da Db Eb
Segment 2: ccdd --> Cc Dc Dd Ed
Segment 3: eeff --> Ce De Df Ef
SHIFT 7: DDEE
Segment 1: aabb --> Da Da Eb Eb
Segment 2: ccdd --> Dc Dc Ed Ed
Segment 3: eeff --> De De Ef Ef
SHIFT 8: DEEF
Segment 1: aabb --> Da Ea Eb Fb
Segment 2: ccdd --> Dc Ec Ed Fd
Segment 3: eeff --> De Ee Ef Ff
SHIFT 9: EEFF
Segment 1: aabb --> Ea Ea Fb Fb
Segment 2: ccdd --> Ec Ec Fd Fd
Segment 3: eeff --> Ee Ee Ff Ff
SHIFT 10: EFFA
Segment 1: aabb --> Ea Fa Fb Ab
Segment 2: ccdd --> Ec Fc Fd Ad
Segment 3: eeff --> Ee Fe Ff Af
SHIFT 11: FFAA
Segment 1: aabb --> Fa Fa Ab Ab
Segment 2: ccdd --> Fc Fc Ad Ad
Segment 3: eeff --> Fe Fe Af Af
SHIFT 12: FAAB
Segment 1: aabb --> Fa Aa Ab Bb
Segment 2: ccdd --> Fc Ac Ad Bd
Segment 3: eeff --> Fe Ae Af Bf
AaAaBbBbCcCcDdDdEeEeFfFf
AaBaBbCbCcDcDdEdEeFeFfAf
BaBaCbCbDcDcEdEdFeFeAfAf
BaCaCbDbDcEcEdFdFeAeAfBf
CaCaDbDbEcEcFdFdAeAeBfBf
CaDaDbEbEcFcFdAdAeBeBfCf
DaDaEbEbFcFcAdAdBeBeCfCf
DaEaEbFbFcAcAdBdBeCeCfDf

Figure 4.7: Sub-correlation combinations.

Coherent and non-coherent integrations

Two different accumulation strategies can be adopted to average the effect of noise in the receiver and to extract the useful signal from the noise level, both based on integration:

- coherent integration
- non-coherent integration

The accumulation strategy described above is usually referred as coherent integration. The samples of the I and of the Q channel, resulting from the correlations, are independently accumulated. This leads to a SNR improvement and can yield to correlation results averaged over a time period up to several seconds. However, the length of the time windows over which coherent integration can be performed is limited by several factors, including uncompensated Doppler shifts, navigation message, phase shifts introduced by the motion of the receiver, as described in Section 3.3.1. Over tens of milliseconds these phase changes may cause destructive interference that defeats the purpose of coherent integration. For example, a navigation data bit change just half of a coherent integration period can completely cancel out the correlation peak. Therefore, to achieve long averaging intervals, the receiver performs a secondary step of accumulation: the accumulation results are periodically processed by a complex normalizer, able to generate phase insensitive values which can be accumulated for a longer time. This process is referred as non-coherent integration. The operation usually performed on the I and Q correlators output to obtain the non-coherent correlators is the following:

$$\sqrt{s_I^2 + s_Q^2}\,. \tag{4.6}$$

For example, in traditional receiver operating on GPS C/A signals, the following configuration is adopted: 1 ms coherent integration, corresponding to 1 code epoch, or 20 ms coherent integration, corresponding to the length of a navigation data bit (once bit synchronization is achieved), and $>$ 20 ms non-coherent integration.

Synchronization techniques

In order to perform correlation and synchronization between the incoming and the local signal codes, three different approaches can be followed.

1. It is possible to **delay the local code**. In this case the starting point of the reference local code is selected from a particular different line in the code LUT, according to the prior information on the delay. The resolution that can be achieved corresponds to the size of a single line of the LUT, which is equal to the size of the shift register; in this case PK samples.

2. The beginning of the **integration epoch** can be delayed. In this case the input signal used in such integration starts b number of samples later, where b is equal to the prior delay information and the BB processing starts with the first sample of the signal after this delay. The reference code is directly taken from the code LUT, without any processing (except decimation). The resolution that can be achieved corresponds to 1 sample.

3. It is possible to delay the integration epoch, considering fraction of samples. This is the same approach of 2., but the fractional part of the delay information is considered in the engine performing decimation; in this way a resolution lower than 1 sampling time is achieved. The lower bound of the resolution depends also on the integration time. This is the strategy adopted in the implementation of the technique described in Section 5.

Interpolation techniques

Generally, the resolution of the modes described above, despite the benefits brought by increasing the value of P, is not sufficient for many purposes. A possible approach to estimate the delay of the signal with a greater accuracy is the use of interpolation techniques. According to the sampling theory, if a band-limited analog signal with the highest frequency $f_{\max}$ is sampled at a rate of $f_s > 2f_{\max}$, the band-limited signal can be reconstructed from its samples using interpolation [43]. (4.7) describes the concept based on the `sinc` interpolation method.

$$x(t) = \sum_{n=-\infty}^{+\infty} x(n)\,\mathrm{sinc}(t-n)\,, \tag{4.7}$$

where

$$\mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t}, \quad t \neq 0\,. \tag{4.8}$$

In the implementation of interpolation techniques, a bank of sinc functions is multiplied with the vector of points of the multi-correlator function.

In (4.7), the n index refers to nT seconds, where T is the sampling period used to determine $x(n)$ from the original signal, and $x(t)$ is the reconstructed signal: in our case respectively the correlator spacing and the autocorrelation function. The above formula represents a linear convolution between the sequence $x(n)$ and shifted samples of the sinc function. Due to the shifting of the sinc function by integer multiples of T, this results in $x(t)$ having the exact value of a sample located at a multiple of T. This can be seen by observing that the absolute error is always zero at times which are integer multiples of T, in other words at the sample locations. In the following implementation, the sinc function is sampled at a much higher rate than the sampling frequency used for the original function, in order to produce a smoother plotted result. Figure 4.8, indeed, shows the correlation peak of a GPS C/A code, as computed by the algorithm described above with $P = 5$. It also shows the sinc interpolation continuous curve, computed considering the sampled correlation function.

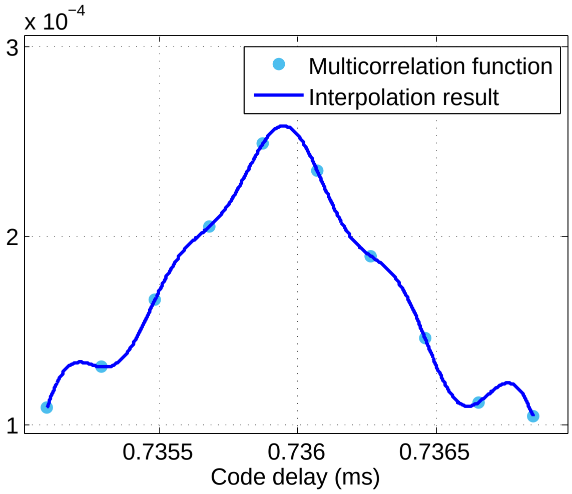


Figure 4.8: Multi-correlation function (correlation peak) and sinc interpolation solution.

It is obvious that it is possible to obtain a higher resolution using an interpolation method like the sinc interpolation [44] but the main point is to understand its limitation. By referring to the Shannon's sampling theorem, $x(t)$ can be uniquely reconstructed from the samples $x(nT)$ only when $x(t)$ is band limited to $\pm 1/2T$. On top of that also the number of $x(n)$ samples used in the convolution is limited and not ranging from minus infinity to infinity.

4.1.3 Doppler frequency estimation

Doppler frequency estimation is the second main task of a receiver. Doppler frequency is useful not only for computing the velocity of the receiver, but also to properly perform the carrier wipe-off in order to remove any residual frequency component in the signal. This operation is fundamental to obtain a good code delay estimate in the processing unit.

In Section 4.1.2, some simple methods (complex cross product) based on consecutive multi-correlation outputs have been presented. Other, like the IQMIX discriminator, are described in [45, 46]. Nevertheless, the most suitable, robust and efficient for such architectures based on snapshot open-loop processing are represented by Fast Fourier Transform (FFT)-based algorithms. They can be found in many mass-market receivers, especially in the cases in which short FFTs are computed and then the processing delay and computational burden is limited. In addition, they can be easily coupled with open-loop snapshot architectures.

MLE FFT-based estimation

A Maximum Likelihood Estimator (MLE)-based method, exploiting FFT and characterized by long coherent integration combined with down sampling is described in the patent [47]. More in details, a long sequence of input values is represented by a smaller number of values, by replacing a number of correlation points with their average. Then, a relatively short length FFT can reveal the low frequency components present in the signal with a high resolution. The main advantage is represented by the large reduction of the computational load, achieved using down sampling and averaging, but without compromising the frequency resolution. Moreover, by changing parameters such as the FFT length and the down sampling ratio, the algorithm can be adapted to different signals and dynamics. Figure 4.9 shows a block scheme of the receiver architecture considered.

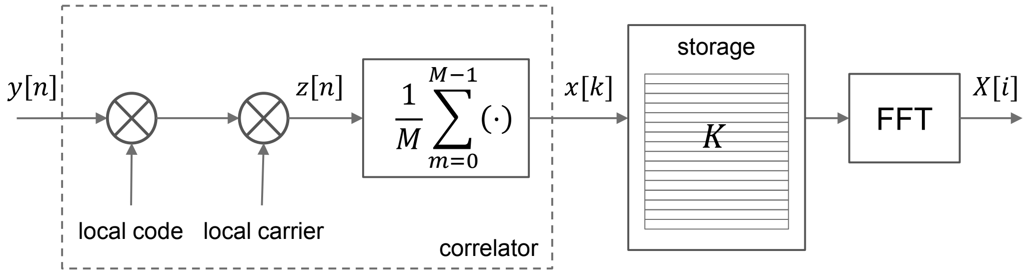


Figure 4.9: Block scheme of the open-loop receiver architecture for MLE Doppler frequency estimation.

The structure of this frequency estimation algorithm has then been modified in the following way to better adapt to modernized GNSS signals.

1. The input signal $y[n]$ is coherently integrated for a certain coherent integration time T_c, usually equal to the primary code period length. Code and frequency wipe-off are performed exploiting respectively a local replica of the spreading sequence and a local carrier at IF.
2. The secondary code synchronization (if a secondary code is present) is estimated and the bit transitions due to phase changes are removed.
3. The bits of the navigation message (if present) are estimated and removed, to avoid bit transitions.
4. Eventually, a certain number of correlation values are averaged and grouped.
5. K consecutive correlation prompt values $x[k]$ are stored in a memory, spanning a total time interval equal to $T_\mathrm{tot} = KT_\mathrm{c}$.
6. A K-point complex FFT, $X[i]$, is computed on the down-sampled sequence, exploiting FFT algorithms, to determine the residual frequency components.
7. The Doppler frequency estimate, corresponding to the frequency index of the FFT peak, is evaluated.

The original patent considers only GPS L1 C/A legacy signals It suggests to collect 80 correlation values at $T_c = 1$ ms, spanning a total duration equal to 80 ms, and to partition them into $K = 16$ groups, with 5 consecutive averaged values in each group. A 16 point complex FFT is then computed. Therefore, despite the reduction of the number of values by a factor 5, leading to a lower computational burden, the total time duration of the sequence is unaltered. However, the presence of phase transitions due to the navigation data requires a further intermediate passage.

This method has been extended to the case of Galileo OS signals in the E1 bandwidth, in particular by taking advantages of the different signal structure. In fact, the OS signal in E1C is modulated by a 4 ms long primary code and a 100 ms long secondary tiered code, and both are public [21]. By processing the pilot signal on E1C, instead of the data signal on E1B, the problem of estimating the data bit transition is solved, allowing computing an arbitrary long FFT.

The input frequency error. A detailed analytical derivation is reported in Section 4.2.1. What is important is to define the *input frequency error*, or frequency offset:

$$\Delta f = |f_0 - f_D| \,, \tag{4.9}$$

where f_D is the true Doppler frequency of the GNSS signal and f_0 is frequency used for the frequency wipe-off. Thus Δf represents a residual frequency component. The objective of the algorithm considered in this section is indeed to estimate it, to accurately compute the Doppler frequency.

Boundaries and resolution. The FFT is able to represent the input signal frequency components in a bounded frequency range, depending on the coherent integration time T_c:

$$\left[-\frac{1}{2T_c} ; \frac{1}{2T_c} \right] . \tag{4.10}$$

This means that the residual frequency error Δf, defined in (4.9), should lie within this range in order to correctly process the signal. To increase this range it is necessary to reduce T_c, with clear negative effects on the averaging process; by reducing the number of input samples of the correlation process the noise level increases. At the same time, the resolution δf of the FFT depends on the number K of points on which it is computed and on T_c, according to the expression

$$\delta f = \frac{1}{K T_c} \,. \tag{4.11}$$

In order to increase the resolution, which affects the accuracy of the estimate, it is necessary to increase either T_c or K, to perform zero padding. In these cases the resolution could be increased as required, at the expenses of an increased computational cost, often not affordable by such architectures. Therefore, a trade-off between resolution, computational complexity and integration has to be considered, according to the specific application and to the characteristics of the signal (C/N_0, dynamics, ...). A third solution to improve the estimate accuracy, without affecting the FFT resolution, is offered by refinement and interpolation techniques, as detailed in the following.

In this work the values $T_c = 4$ ms and $K = 16$ have been chosen, to that $T_{\text{tot}} = 64$ ms. Hence, recalling (4.10), the frequency range is

$$\left[-\frac{1}{2T_c} ; \frac{1}{2T_c} \right] = \left[-\frac{1}{2 \cdot 4 \text{ ms}} ; \frac{1}{2 \cdot 4 \text{ ms}} \right] = [-125 \text{ Hz}; +125 \text{ Hz}] \,, \tag{4.12}$$

and the resolution is

$$\delta f = \frac{1}{K T_c} = \frac{1}{16 \cdot 4 \text{ ms}} = 15.625 \text{ Hz} \,. \tag{4.13}$$

It is suitable to fix T_c equal to the code period length. Lower values are not recommended, since a frequency range of 250 Hz is sufficient for most scenarios, and in order to take advantage of the full correlation gain. Concerning K, a sequence length that is a power of two is considered, since it is the most common FFT algorithm. Small values of K, such as 4 or 8, are not possible, since they would reduce the resolution δf and the total observation period, leading to a reduced sensitivity and robustness. At the same time, large values, such as 32, 64 and more, would bring a considerable increase of the computational load. Furthermore, the effect of the satellite and user dynamic has to be considered (see Chapter 7).

Example. The K-point FFT, $|X[i]|^2$, (which will be analytically derived in (4.41), Section 4.2.1) is plotted in Figure 4.10, to evaluate the Doppler frequency estimate.

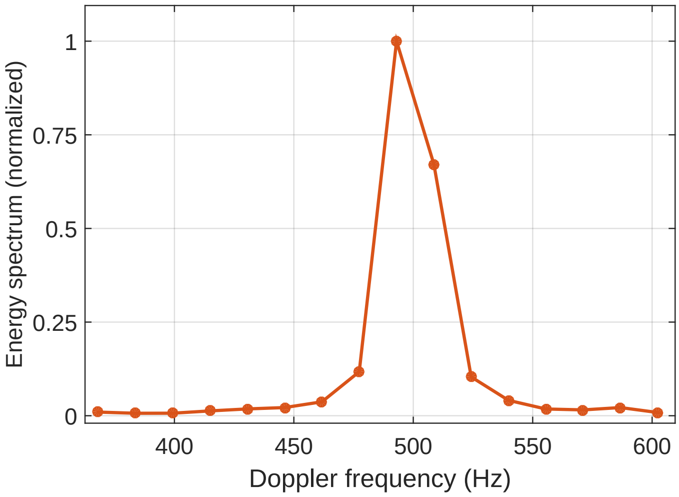


Figure 4.10: MLE FFT-based Doppler estimation with $K = 16$ for a simulated Galileo signal.

In this specific case, a Galileo OS E1 signal has been generated by means of the N-FUELS software signal generator (Section A.1). The signal is characterized by a C/N_0 of 50 dBHz and by a Doppler frequency $f_D = 500$ Hz. The carrier removal has been performed with $f_0 = 493$ Hz, so that an error $\Delta f = 7$ Hz is artificially introduced for testing purposes. Since the value 7 lies in the range defined in (4.12), the frequency component can be estimated. From the figure is clear that the peak of the FFT, corresponding to the value of the true Doppler, lies somewhere in-between the two highest FFT points (493 Hz and 508.6 Hz). It is then estimated by just taking the highest:

$$\hat{f}_{D,FFT} = 493 \text{ Hz} \,. \tag{4.14}$$

Thus, as expected, the estimation error is equal to

$$\varepsilon_{\text{FFT}} = \left| f_D - \hat{f}_{D,FFT} \right| = |500 - 493| = 7 \text{ Hz} \,. \tag{4.15}$$

The standard FFT technique presented above, along with the improvements presented hereafter, have been analysed, implemented in a software receiver, tested and compared. Results are reported in Section 5.4.

Sinc interpolation

To increase the accuracy of the estimate, without redesigning the parameters of the algorithm, it is possible to compute an interpolation on the FFT points. The straightforward solution is to compute a standard Whittaker-Shannon interpolation [48] to the K FFT points; this is justified by the fact that they exactly represent the sampling of a sinc square function. According to the sampling theory, if a band-limited analog signal with the highest frequency $f_{\max}$ is sampled at a rate of $f_s > 2f_{\max}$, the band-limited signal can be reconstructed from its samples using sinc interpolation [43].

$$x(t) = \sum_{k=-\infty}^{+\infty} x(k)\,\mathrm{sinc}(t-k)\,. \tag{4.16}$$

Equation (4.16) describes this concept. In the implementation of interpolation techniques, a bank of sinc functions is multiplied with the vector of points of the multi-correlator function. The sinc function is sampled at a much higher rate than the sampling frequency used for the original function, in order to produce a smoother result.

Figure 4.11 reports an example of sinc interpolation (grey line), obtained using a curve with a resolution equal to 1 000 times the original signal resolution: clearly the accuracy of the result is improved, passing from $\hat{f}_{D,FFT} = 493$ Hz to $\hat{f}_{D,\text{sinc}} \simeq 497.22$ Hz. However, at least three drawbacks should be reported:

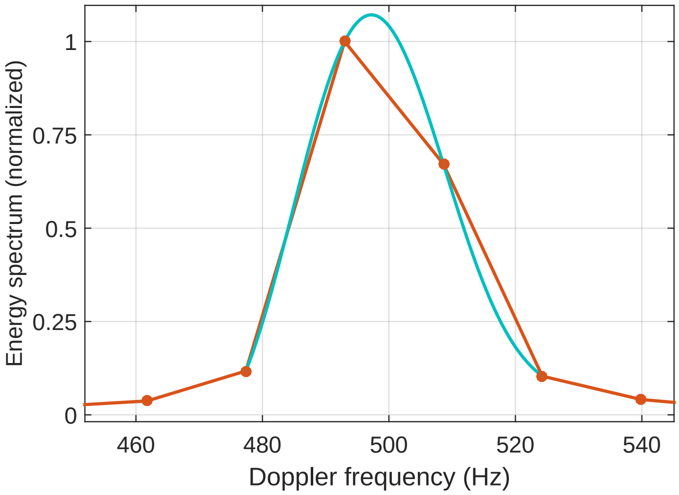


Figure 4.11: Frequency estimation with FFT and sinc interpolation for a simulated Galileo signal.

- it performs in a good way when the C/N_0 of the received signals is high, but when the points are degraded by noise, a different and more robust solution should be considered;
- there are existing solutions ([49, 50]) offering a similar performance with a lower computational burden, thus more suitable for portable devices;
- the accuracy of the solution directly depends on the resolution of the interpolating curve.

Fast nearly ML estimation

Based on the simple equation derived from the mathematical expression of the CAF, a family of novel algorithms is provided, especially concerned to refine the Doppler frequency estimate with respect to that provided by a conventional acquisition method [49]. According to the authors, the algorithm almost reaches the Cramer-Rao Lower Bound (CRLB) and achieves the same performance of a least square method, but requiring a reduced computational burden. This makes it especially suitable for mass-market receivers. In particular, a simple formula is derived from the peaks and the neighbour points of the CAF in the frequency domain. An example of frequency domain representation of the CAF, which corresponds to an FFT, is shown in Figure 4.12, where the x-axis contains the variable Doppler frequency, while the y-axis represents the absolute value of the acquisition test statistic (S). The width of the main lobe depends on the FFT parameters, and in this case is required to be equal to is $2/T_c$; this guarantees that three adjacent points of the frequency domain CAF (A, B, C) are always located in the main lobe.

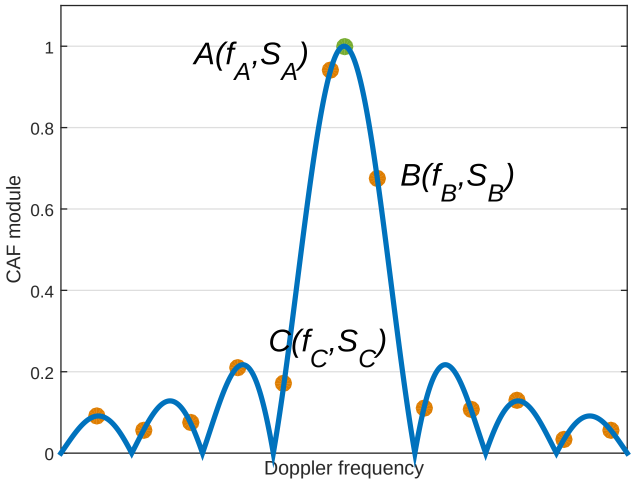


Figure 4.12: Frequency domain representation of the CAF.

The cells A, B and C are characterized by the duplets (f_A, S_A), (f_B, S_B), and (f_C, S_C); in the case of no noise, these duplets can be used to find the true Doppler frequency as follows:

$$f_d = \frac{f_A S_A + f_B S_B + f_C S_C}{S_A + S_B + S_C} \,. \tag{4.17}$$

This expression gives the correct Doppler values, provided that the points are spaced as described above, and indeed represents an average of the three measured points. It can also be used as a first estimator a Doppler even when the signal is affected by noise. However, despite the good performance of the technique, the requirement on the Doppler step cannot be satisfied with the implementation of the multi-correlator unit, and this solution cannot be evaluated. A similar approach has been followed, based on a spectral peak location estimation algorithm [51]. This article presents simple algorithms that provide substantial refinement of the frequency estimation of tones based on FFT samples without the need of increasing the FFT size. The idea is similar to the one presented above, an consist in estimating the frequency based on three FFT samples: f_C , f_A and f_B. A fractional correction term δ is computed and added to the integer peak index f_A, to determine a fine estimate of the peak location. It is proved that this corrected refinement of the original bin location estimate can be surprisingly accurate even in low SNR conditions. A first definition of the correction term, derived from the theoretical expression of the FFT, is given by

$$\delta = \frac{|S_B| - |S_C|}{4|S_A| - 2|S_C| - 2|S_B|} \,. \tag{4.18}$$

However, this simple expression is statistically biased and performs poorly in the presence of noise. Some simple changes can be made, for example using the complex FFT values rather than the magnitude, as follows:

$$\delta = -\Re \frac{S_B - S_C}{2S_A - 2S_C - 2S_B} \,. \tag{4.19}$$

It has be to pointed out that a similar solution is considered in [47].

Quadratic Fit

In [52] several methods are proposed and compared. Among them, a technique based on approximate MLE using FFT and *quadratic interpolation* is considered. The FFT magnitude is interpolated by quadratic polynomial: quadratic fit in the form $y = a + bx + cx^2$ around the FFT maximum can be computed, given the frequencies of the three points around the peak. Then the peak of the quadratic fit, that is also the frequency estimate, is equal to $-b/2c$. This is obtained by solving a system of linear equations, containing a Vandermonde matrix. Given that k is the index of the peak P_0, $k-1$ is the index of the left peak P_- and $k+1$ the index of the right peak P_+,

$$\begin{bmatrix} (k-1)^2 & k-1 & 1 \\ k^2 & k & 1 \\ (k+1)^2 & k+1 & 1 \end{bmatrix} \cdot \begin{bmatrix} c \\ b \\ a \end{bmatrix} = \begin{bmatrix} P_- \\ P_0 \\ P_+ \end{bmatrix} . \tag{4.20}$$

And then

$$\hat{f}_D = \frac{-b/2c - 16/2 - 1}{16T_c} \,. \tag{4.21}$$

It is proved that the estimation accuracy is improved significantly thanks to this post-processing technique.

Discriminator-based estimator

An innovative technique based on the computation of a correction factor to increase the accuracy of the FFT estimate will be presented in Section 4.2.

4.1.4 Search space windowing

As already outlined in Section 4.1.1, it is interesting to reduce both the code delay and the Doppler frequency search spaces, provided that an accurate enough estimate is available.

In the first case, as described in Section 4.1.2, instead of computing a full correlation, equal to the code length, it is possible to compute sub-correlations. L can be vary, as shown in Figure 4.6, so that only the sub-portion of the correlation function containing the correlation peak is computed. The main advantage is the saving of a lot of hardware correlators, or similarly, the reduction of the computational load of the engine.

In the second case, the reduction of the search space is also necessary, because of the properties of the FFT estimation algorithm introduced. As shown in Section 4.1.3, the FFT-based frequency estimation algorithms are only able to detect residual frequency components in the range $[-1/(2T_c); +1/(2T_c)]$. In addition, when the Doppler frequency is close to the borders of this frequency range, the power of the FFT peak is reduced, because the accumulation before the FFT introduces a slow sinc term, as reported in (4.28). Its effect is to attenuate the correlation results, potentially introducing errors in the estimation process. For input frequency errors close to zero this is negligible, but it becomes evident as long as the absolute value of Δf increases. The algorithm proposed shall avoid this last case, by the application of proper windowing schemes and prediction strategies.

Therefore, instead of considering the full code delay search space (1 ms in the case of GPS L1 C/A, 4 ms in the case of Galileo E1B) and the full Doppler search space (from −9 kHz to +9 kHz according to Table 4.4), a reduced search space is managed. This is referred in the following as windowing of the signal. The windowing approach allows a strong reduction of the computational burden. It represents a key feature also speeding up the receiver operations and for reducing the power consumption.

Table 4.4: Example design choices for a receiver acquisition [37].

Frequency contributor	Design range
Satellite Doppler (GPS L1)	−4.2 to 4.2 kHz
TCXO Offset (±3 ppm)	−4.725 to 4.725 kHz
Receiver velocity (up to 160 km/h)	−0.23 to 0.23 kHz
Total frequency search space	−9.2 to 9.2 kHz

In all this work it is assumed that rough information on the delay and Doppler frequency are present at the receiver power on. This could come for example from AGNSS data. If no information on the code delay are present, it is sufficient to run once the processing unit in standard resolution, in order to consider the full code delay search space, and then, when disposing of a measure, moving to a higher resolution mode. If no information on the Doppler frequency is present, it is sufficient to perform once several parallel FFTs, to cover all the Doppler range, with an approach similar to the standard serial acquisition process.

Nevertheless, problems arise when, because of the user-satellite motion, the true Doppler frequency and code delay values evolve with time. The reduction of the search space, coupled to a pure open-loop approach, can lead to a deadlock situation. In fact, if for example the true code delay approaches the boundaries of the reduced code delay search space, then, as soon as it crosses the bound, the correlation peak is no longer detectable and no valid measures can be computed. There is no way that the receiver can re-start correctly "tracking" the signal in this case. The same happens to the Doppler frequency.

The solution is to reinitialize the receiver, i.e. to re-centre the code delay and Doppler frequency search spaces around the last valid estimates as soon as they approach the bounds. This is shown for the case of the frequency in Figure 4.13. The frequency reduced search space is depicted in red; as soon as the estimated Doppler frequency $\hat{f}_D$ reaches the border of the red region, a new wipe off frequency f_0 is defined, based on to the last valid Doppler estimate. This approach is a sort of quasi open-loop [6], in the sense that a loose feedback is generated with a lower rate. In this simple case the rate is determined by the dynamics of the signal.

It is evident that this simple solution is not robust enough to noise and to high dynamics. For example, if the noise is so high that at a certain epoch a false correlation peak is detected, much far from the true one and close to a border, there is the risk that the receiver re-centres the code delay search space around this point for the next epoch, thus excluding the true correct peak and forcing a deadlock condition. At the same time, large dynamics, or, as will be explained in Section 4.3, on demand processing, can fast up the rate of evolution of the signal parameters. It can happen that the real value for a current epoch is outside the search space, because the estimate of the previous epoch was so far away from this one and from the borders that a windowing didn't appear necessary. A filtering technique, such a hatch filter on the last estimates, along with a check on the validity of each estimate, can solve most of the problems. Implementation and results are presented in Section 5.6.

Finally, it is noted that there is a difference between the absence of a correct estimate due to deadlock and the so called invalid measure. Indeed, an invalid measured happens only when the signal is present in the actual search space, but buried in the noise and then not visible.

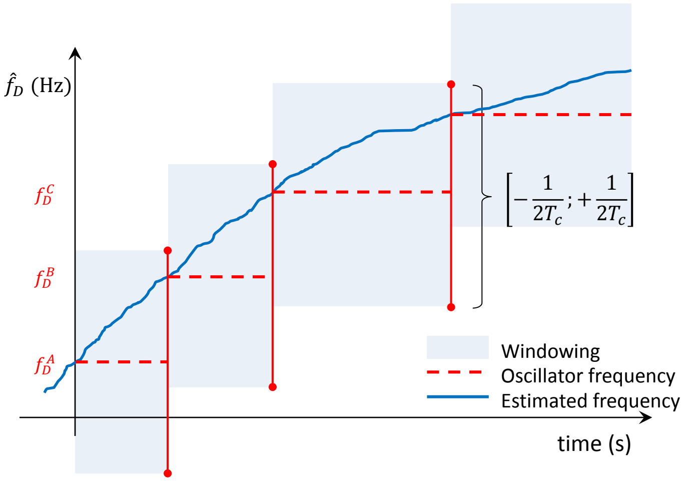


Figure 4.13: Frequency windowing strategy implemented.

4.1.5 Carrier phase estimation

Carrier phase is not a typical value estimated by mass-market receivers. This is due to the fact that the measurements obtained with the techniques described in Section 4.1.2 are not sufficiently accurate to solve the initial ambiguity resolution problem, which is necessary to start carrier phase measurements [53]. With recent improvement and with the first attempt to make dual frequency mass market receivers, the accuracy improved a lot, allowing carrier phase solutions, or use of Real Time Kinematic (RTK) or Precise Point Positioning (PPP) techniques. These methods can produce accuracies at the decimetre level, even on a moving platform, but unfortunately, not in an urban channel, because the carrier phase solutions are vulnerable to cycle slips. Moreover, in urban canyons, most signals are tracked as a reflection. Nevertheless, some carrier phase estimation techniques are considered with this architecture just to prove their theoretical possibility. In addition, phase estimates are used in the fully software routine to decode the navigation message on the data channel. A more complete description concerning carrier phase measurements and cycle slips is provided in Chapter 6.

The phase is computed from the estimate of the Doppler frequency, by accumulating the measure for consecutive epochs (accumulated Doppler): in Figure 4.14 a simplified version of the Galileo E1 pilot channel block scheme, showing only frequency and phase estimation, is reported. The full structure will be presented in Section 5.1.3.

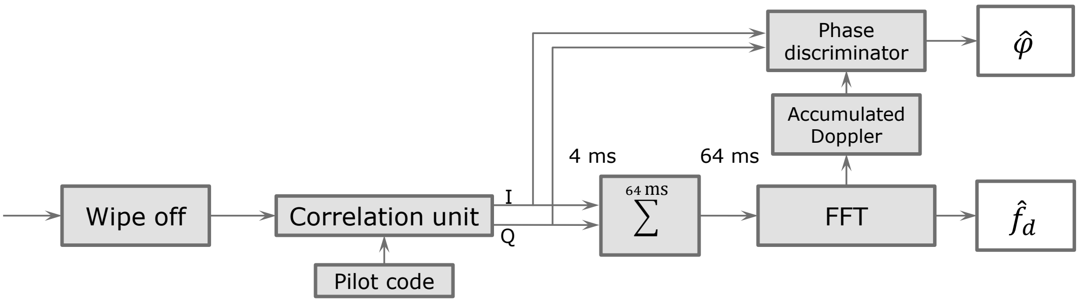


Figure 4.14: Block scheme of phase estimation in the receiver architecture.

The pilot channel, with Galileo E1C codes, is processed with a coherent integration time of 4 ms; 16 non-coherent sums are then accumulated, thus spanning a total time of 64 ms. The Doppler frequency ($\hat{f}_D$) is estimated every 64 ms; at the same time, the carrier phase is estimated every 4 ms with a phase discriminator exploiting I and Q correlation values. In particular, the accumulated Doppler f_A is computed from the estimated Doppler frequency (Figure 4.15) according to

$$f_A = \sum \left(2\pi \hat{f}_D t\right) . \tag{4.22}$$

f_A is then refined with the phase computed with an open-loop phase discriminator from I and Q values (Figure 4.15), obtaining an accurate expression of the phase of the carrier ($\hat{\varphi}$):

$$\hat{\varphi} = \text{atan2}\left(\Re(se^{-jf_A}), \Im(se^{-jf_A})\right) \tag{4.23}$$

where $s = I + jQ$. It is noted that the value of this refinement, i.e. the residual phase, is exactly the loss that would be present when estimating only the frequency and the accumulated Doppler.

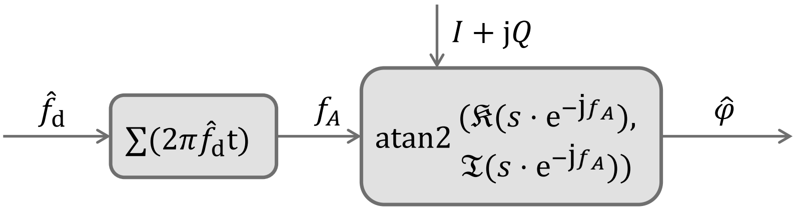

$I + \mathrm{j}Q$
$\hat{f}_\mathrm{d}$
$\sum(2\pi\hat{f}_\mathrm{d}\mathrm{t})$
f_A
atan2 $(\Re(s \cdot \mathrm{e}^{-\mathrm{j}f_A}),$ $\Im(s \cdot \mathrm{e}^{-\mathrm{j}f_A}))$
$\hat{\varphi}$

Figure 4.15: Computation of the carrier phase through accumulated Doppler.

Thanks to this method, the PLL, commonly used to track the phase of a GNSS signal, is not required.

4.2 Advanced open-loop frequency estimation

In this section an innovative MLE interpolation-based technique for Doppler frequency estimation is introduced and deeply investigated. It represents indeed an extension and improvement of the FFT-based technique described in Section 4.1.3, . In particular, a complete analytical analysis is provided, in order to assess the performance and to validate the results. Moreover, a technique based on a second FFT is introduced to solve the problem of the varying accuracy with respect to different initial frequency errors, based on the windowing approach but characterized by a reduced computational demand. The main objective and advantage of this technique is the large reduction of computational load and of power consumption, achieved using down sampling and averaging, but without compromising the frequency resolution, so as to reach the CRLB for Maximum Likelihood (ML) frequency estimation. This makes the proposed algorithm a low complexity and robust solution, allowing real time and on demand applications, for implementation in GNSS receivers based on snapshot processing featuring the consumer and automotive market.

4.2.1 Analytical derivation

In this section the analytical model of the signal is considered and its expression at different stages of the receiver, with reference to Figure 4.9, is derived.

The signal at the output of a GNSS receiver front-end in an AWGN channel, after down-conversion at IF and ADC conversion at frequency f_s, can be modelled as in (3.2). At this point it can be assumed that the navigation message is absent, $d(nT_s - \tau) = 1$; this is the case for example of *pilot signals*, as the Galileo E1C component [21]. Alternatively, it is possible to assume that the navigation message is perfectly wiped-off exploiting some assisting information, as in the case of AGNSS techniques [37]. Furthermore it can be proved that small synchronization errors are negligible and that they can be considered as an additional noise contribution. Similarly, the secondary code term $\bar{c}(nT_s - \tau)$ is dropped from (3.2), assuming that it is absent (as for GPS L1 legacy signals) or that it can be removed. In addition, for sake of simplicity, but without loss of generality, the signal and the noise contributions are handled separately. (3.2) is then rewritten as

$$\tilde{s}[n] = y[n] + \eta[n] = \sqrt{2C}c(nT_s - \tau)\cos(2\pi(f_{IF} + f_D)nT_s + \varphi) + \eta(nT_s)\,. \tag{4.24}$$

The signal enters the correlation block, in which wipe-off is performed in order to remove both code and carrier components, either in a closed-loop or in a snapshot-based open-loop. Let's first assume that noise is absent, $\eta(nT_s) = 0$: only the **useful signal contribution** $y[n]$, i.e. the first term of (4.24), is considered.

$$y[n] = \sqrt{2C}c(nT_s - \tau)\cos(2\pi(f_{IF} + f_D)nT_s + \varphi)\,. \tag{4.25}$$

The signal in (4.25) is first multiplied by a locally generated version of the spreading code, $c(nT_s - \hat{\tau})$, where $\hat{\tau}$ is an estimate of the code delay τ. In general, in realistic situations, $\hat{\tau} \neq \tau$ and a residual cross-correlation affects the post correlation signal. However, when working in steady conditions, this term is usually small and it can be neglected. In this work it is indeed assumed that the code wipe-off is perfect, so that the code component disappears: $c(nT_s - \tau)c(nT_s - \hat{\tau}) = 1$.

Then the signal is multiplied by a local carrier, $e^{-j2\pi(f_{IF}+\hat{f}_D)nT_s}$, in the form of a complex exponential, where $\hat{f}_D$ is an estimate of the Doppler frequency f_D coming from previous measurements.

The signal after code and frequency multipliers is denoted $z[n]$:

$$\begin{aligned}
z[n] &= y[n]c(nT_s - \hat{\tau})e^{-j2\pi(f_{IF}+\hat{f}_D)nT_s} \\
&= \sqrt{2C}c(nT_s - \tau)c(nT_s - \hat{\tau})\cos(2\pi(f_{IF} + f_D)nT_s + \varphi)e^{-j2\pi(f_{IF}+\hat{f}_D)nT_s} \\
&= \sqrt{2C}\frac{1}{2}\left[e^{j2\pi(f_{IF}+f_D)nT_s+j\varphi} + e^{-j2\pi(f_{IF}+f_D)nT_s+j\varphi}\right]e^{-j2\pi(f_{IF}+\hat{f}_D)nT_s} \\
&= \sqrt{\frac{C}{2}}\left[e^{-j2\pi\Delta f nT_s+j\varphi} + e^{-j2\pi(2f_{IF})nT_s-j2\pi(\hat{f}_D+f_D)nT_s-j\varphi}\right].
\end{aligned} \tag{4.26}$$

Then M correlation samples are accumulated; $T_c = MT_s$ is the coherent integration time and $x[k]$ is the result of this accumulation process. Since K instances of $x[k]$ are needed for the FFT evaluation, the samples of $z[n]$ are divided into K groups of M samples each and the index n is rewritten as $kM + m$. In the particular implementation of the technique considered in the following, K consecutive snapshots are exploited. The samples $x[k]$ at the output of the correlator block are equal to:

$$\begin{aligned}
x[k] &= \frac{1}{M}\sum_{m=0}^{M-1} z[kM+m] \\
&= \frac{1}{M}\sum_{m=0}^{M-1}\sqrt{\frac{C}{2}}\left[e^{-j2\pi\Delta f(kM+m)T_s+j\varphi}+e^{-j2\pi(2f_{IF})(kM+m)T_s-j2\pi\left(\hat{f}_D+f_D\right)(kM+m)T_s-j\varphi}\right] \\
&\simeq \frac{1}{M}\sqrt{\frac{C}{2}}\sum_{m=0}^{M-1} e^{-j2\pi\Delta f(kM+m)T_s+j\varphi} \\
&\simeq \frac{1}{M}\sqrt{\frac{C}{2}}e^{-j2\pi\Delta fkMT_s+j\varphi}e^{-j\pi\Delta f(M-1)T_s}\frac{\sin(\pi\Delta fMT_s)}{\sin(\pi\Delta fT_s)}.
\end{aligned} \tag{4.27}$$

In (4.27), the fact that the high frequency term is filtered out by the summation has been exploited, and thus the double frequency term neglected. Since the term $\pi\Delta fT_s$ is usually small, $\sin(\pi\Delta fT_s)$ can be approximated by its argument. Moreover, since $e^{-j\pi\Delta f(M-1)T_s}$ is a phase shift and does not depend on k, it can be included into the phase offset of the incoming signal φ, obtaining a new phase offset θ. $x[k]$ can then be approximated as

$$\begin{aligned}
x[k] &\simeq \frac{1}{M}\sqrt{\frac{C}{2}}e^{-j2\pi\Delta fkMT_s+j\varphi}e^{-j\pi\Delta f(M-1)T_s}M\,\mathrm{sinc}(\Delta fMT_s) \\
&\simeq \sqrt{\frac{C}{2}}\,\mathrm{sinc}(\Delta fMT_s)e^{-j2\pi\Delta fkMT_s}e^{j\theta}.
\end{aligned} \tag{4.28}$$

In the realistic case of non-perfect code wipe-off, an additional term would be present in (4.28). However, its effect on the FFT is negligible, and can be totally considered as an additional noise contribution.

As depicted in Figure 4.9, K instances of $x[k]$ are stored in memory, and then an FFT is computed on these K values to estimate the frequency component. The K-point FFT output, computed on the correlation points $x[k]$, is denoted $X[i]$, with $i \in \mathbb{N}$, $0 \leq i \leq K-1$, and can be computed as:

$$\begin{aligned}
X[i] &= \sum_{k=0}^{K-1} x[k]e^{-j\frac{2\pi}{K}ik} \\
&= \sum_{k=0}^{K-1}\sqrt{\frac{C}{2}}\,\mathrm{sinc}(\Delta fMT_s)e^{j\theta}e^{-j2\pi\Delta fkMT_s}e^{-j\frac{2\pi}{K}ik} \\
&= \sqrt{\frac{C}{2}}\,\mathrm{sinc}(\Delta fMT_s)e^{j\theta}\sum_{k=0}^{K-1}e^{-j2\pi k\left(\Delta fMT_s+\frac{i}{K}\right)} \\
&= \sqrt{\frac{C}{2}}\,\mathrm{sinc}(\Delta fMT_s)e^{j\theta}e^{-j\pi\left(\Delta fMT_s+\frac{i}{K}\right)(K-1)}\frac{\sin(\pi\Delta fKMT_s+\pi i)}{\sin\left(\pi\Delta fMT_s+\frac{\pi i}{K}\right)} \\
&= \sqrt{\frac{C}{2}}\,\mathrm{sinc}(\Delta fT_\mathrm{c})e^{j\theta}e^{-j\pi\left(\Delta fT_\mathrm{c}+\frac{i}{K}\right)(K-1)}\frac{\sin\left(\pi\Delta fKT_\mathrm{c}+\pi i\right)}{\sin\left(\pi\Delta fT_\mathrm{c}+\frac{\pi i}{K}\right)}.
\end{aligned} \tag{4.29}$$

Thus, (4.29) represents the expression of the FFT at the end of the processing scheme of Figure 4.9, in the absence of noise.

It can be proved that the **noise** $h[k]$ at the correlators output is split in the I and the Q components [3]:

$$h[k] = \eta_I + j\eta_Q \tag{4.30}$$

and that η_I and η_Q are independent identically distributed (i.i.d.) Gaussian random variables:

$$\eta_I, \eta_Q \sim \mathcal{N}\left(0,\, \sigma_\eta^2\right)\,, \tag{4.31}$$

with

$$\sigma_\eta^2 = \frac{N_0 f_s}{4M} = \frac{N_0}{4T_\mathrm{c}}\,. \tag{4.32}$$

It is then necessary to compute the FFT of $r[k] = x[k] + h[k]$ in the presence of noise:

$$\begin{aligned} R[i] &= \sum_{k=0}^{K-1} r[k] e^{-j\frac{2\pi}{K}ik} \\ &= \sum_{k=0}^{K-1} x[k] e^{-j\frac{2\pi}{K}ik} + \sum_{k=0}^{K-1} h[k] e^{-j\frac{2\pi}{K}ik} \\ &= X[i] + \sum_{k=0}^{K-1} h[k] e^{-j\frac{2\pi}{K}ik} \end{aligned} \tag{4.33}$$

using the result obtained in (4.29). The noise component $H[i]$ can be evaluated separately.

$$\begin{aligned} H[i] &= \sum_{k=0}^{K-1} h[k] e^{-j\frac{2\pi}{K}ik} \\ &= \sum_{k=0}^{K-1} \eta_I e^{-j\frac{2\pi}{K}ik} + j\sum_{k=0}^{K-1} \eta_Q e^{-j\frac{2\pi}{K}ik} \\ &= H_I[i] + jH_Q[i]. \end{aligned} \tag{4.34}$$

Let's now characterize from a statistical point of view the I and Q FFT samples H_I and H_Q. It is possible to assume that the input noise values are the results of a windowing operation, with a window $w[k]$. In the case under analysis:

$$w[k] = 1, \quad 0 \le k \le K - 1\,. \tag{4.35}$$

It can be proved [54] that the ensemble average of the FFT of Gaussian and white variables with mean μ is equal to:

$$\mathbf{E}[H_I[i]] = \mu W[i]\,, \tag{4.36}$$

where $W[i]$ is the FFT of the window $w[k]$. Since both η_I and η_Q are zero mean, then also their Fourier transform is zero mean, independently from the window applied:

$$\mathbf{E}[H_I[i]] = \mathbf{E}[H_Q[i]] = 0\,. \tag{4.37}$$

The variance is equal to [54]:

$$\sigma_{H[i]}^2 = E_w \sigma_\eta^2\,, \tag{4.38}$$

where E_w is the energy of the window sequence, $E_w = \sum_{k=0}^{K-1} |w[k]|^2$. With the window considered in this case we have $E_w = K$, and then the variance is equal to:

$$\sigma_{H_I}^2 = \sigma_{H_Q}^2 = K\sigma_\eta^2\,. \tag{4.39}$$

Therefore, the noise samples at the output of the FFT block are Gaussian random variables:

$$H_I, H_Q \sim \mathcal{N}\left(0,\, K\sigma_\eta^2\right) = \mathcal{N}\left(0,\, \frac{KN_0}{4T_\mathrm{c}}\right)\,. \tag{4.40}$$

A few important considerations are drawn:

- the variance is constant: each FFT point is statistically characterized by the same noise;
- the variance of the noise samples depends on the window length, i.e. on the size K of the FFT. By reducing the length of the windows, the noise variance can theoretically be reduced, but at the expenses of a reduction of the signal power by a factor K^2, as computed in (4.29).

Summarizing, $R[i]$ represents the expression of the signal at the end of the processing stage, considering both useful signal and noise components, and assuming perfect code synchronization, perfect data wipe-off and absence of secondary code:

$$\begin{aligned} R[i] &= X[i] + H[i] \\ &= \sqrt{\frac{C}{2}}\operatorname{sinc}(\Delta f T_\mathrm{c}) e^{j\theta} e^{-j\pi\left(\Delta f T_\mathrm{c} + \frac{i}{K}\right)(K-1)} \frac{\sin\left(\pi \Delta f K T_\mathrm{c} + \pi i\right)}{\sin\left(\pi \Delta f T_\mathrm{c} + \frac{\pi i}{K}\right)} + H_I[i] + jH_Q[i]\,. \end{aligned} \tag{4.41}$$

The modulus squared of (4.41), $|R[i]|^2$ is used in the following and denoted energy spectrum.

4.2.2 Some examples

In this section the FFT analytically derived in (4.41) is plotted, in order to visualize the effect of different parameters impacting on the estimation accuracy, that is to say the initial frequency error Δf and the C/N_0. In addition, important definitions and theoretical constructs are introduced.

Let's first define P_0 as the highest value of the FFT modulus squared, i.e. its peak:

$$P_0 = \max(|R[i]|^2)\,. \tag{4.42}$$

The frequency corresponding to this peak is denoted f_{P_0} and is the frequency obtained for i_{P_0}:

$$i_{P_0} = \arg\max_i(|R[i]|^2)\,. \tag{4.43}$$

It is noted once more that f_{P_0} corresponds to the Doppler frequency estimate of the FFT technique. However, this estimate is biased, because of the low resolution of the FFT. In fact, despite the fact that the theoretical continuous spectrum of the correlation samples is always the same, the K points of the FFT sample it in a different way depending on the initial frequency error Δf. The error of the estimate f_{P_0}, denoted ε_FFT, is equal to the absolute value of its difference with the true Doppler frequency:

$$\varepsilon_\mathrm{FFT} = |f_D - f_{P_0}|\,. \tag{4.44}$$

Therefore different situations may arise.

1. The correlator multiplier frequency is exactly equal to the true Doppler:

$$\Delta f = 0\,. \tag{4.45}$$

   Therefore, the main FFT peak exactly matches the true Doppler, while all the other points are equal to zero, since they correspond to the zeros of the sinc function. In this borderline case the FFT estimate is exact, $\varepsilon_\mathrm{FFT} = 0$. Figure 4.16a reports an example of this case. The true Doppler frequency is equal to 0 Hz, as well as the correlator multiplier frequency. Clearly also the estimate is equal to 0 Hz and the error is null.

2. Δf is a positive or negative integer multiple l of δf:

$$\Delta f = l \cdot \delta f\,. \tag{4.46}$$

   In this case, again the FFT peak f_{P_0} matches the true Doppler frequency and all the other points are equal to zero, as reported in Figure 4.16b for $l = -1$. Again, the Doppler estimate is correct and $\varepsilon_\mathrm{FFT} = 0$.

3. Δf is equal to half the FFT bin size δf plus eventually a positive or negative integer multiple l of it:

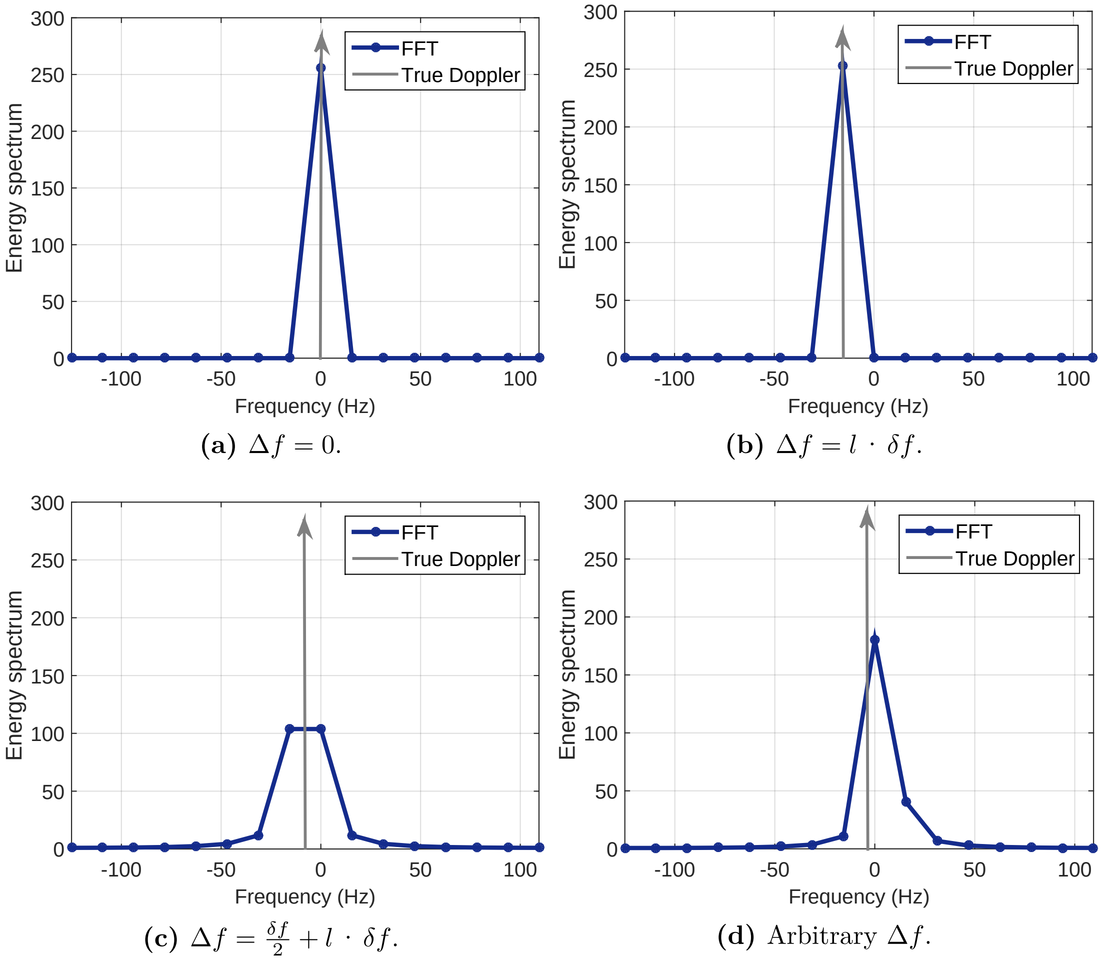
$$\Delta f = \frac{\delta f}{2} + l \cdot \delta f\,. \tag{4.47}$$

In this case the true Doppler falls exactly in between two FFT points, which have indeed the same magnitude. In this case, picking as Doppler estimate the index of the main peak leads to an error:

$$\varepsilon_{\mathrm{FFT}} = \frac{\delta f}{2}\,. \tag{4.48}$$

This corresponds to the worst situation, giving the biggest error, as detailed in Figure 4.16c.

4. All the other intermediate situations. The true Doppler lies in between the two main peaks, as reported as example in Figure 4.16d, and the error $\varepsilon_{\mathrm{FFT}}$ lies in the range $[0; \delta f/2]$.



(a) $\Delta f = 0$.

(b) $\Delta f = l \cdot \delta f$.

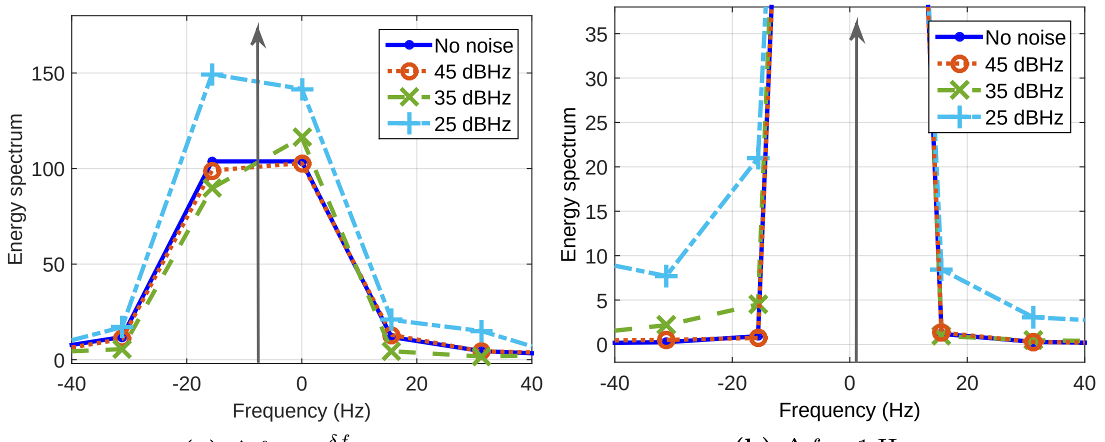
(c) $\Delta f = \frac{\delta f}{2} + l \cdot \delta f$.

(d) Arbitrary Δf.

Figure 4.16: Theoretical FFT for different Δf, $\delta f = 15.625$ Hz.

What is more, the accumulation before the FFT introduces a slow sinc term, as reported in (4.28). Its effect is to attenuate the correlation results, potentially introducing errors in the frequency estimation process. For input frequency errors close to zero this is negligible, but it becomes evident as long as the absolute value of Δf increases. The algorithm proposed shall avoid this last case, by the application of proper windowing schemes (Section 4.1.4) and prediction strategies, according to a quasi-open-loop paradigm [55, 6]. In other words, any time $|\Delta f|$ is larger than a predefined threshold (because of signal dynamics, partial shadowing, light-indoor conditions, etc.), the wipe-off frequency is changed, so as to work in optimal conditions, limiting the larger estimation errors to a few epochs [56].

When the noise contribution is added, the FFT points are affected also by a random noise, as proved in Section 4.2.1. An example for different C/N_0 values and for $\Delta f = -\delta f/2$ and $\Delta f = 1$ is reported in Figure 4.17a and Figure 4.17b respectively.



(a) $\Delta f = -\frac{\delta f}{2}$ **(b)** $\Delta f = 1$ Hz

Figure 4.17: Theoretical FFT for different C/N_0 values, $\delta f = 15.625$ Hz.

4.2.3 A frequency correction term

The idea behind this innovative technique is to compute a frequency **correction term**, based on the values of the FFT points, so as to overcome the limitations due to the FFT resolution (shown in Section 4.1.3) and to improve the rough frequency estimate. In fact, it is theoretically possible to go back to the initial frequency error Δf given $R[i]$, by inverting (4.41) with respect to Δf. The steps of the algorithm are reported hereafter.

1. First a standard FFT is computed and f_{P_0} is estimated, as described above.
2. Successively, the correction factor is computed. It corresponds to an estimate of $\varepsilon_{\mathrm{FFT}}$ and it is denoted $\hat{\varepsilon}_{\mathrm{FFT}}$.
3. Then a decision is taken concerning the sign of the correction $\hat{S} = \pm 1$, i.e. whether to sum or to subtract the correction to f_{P_0}.
4. The Doppler frequency estimate can finally be computed:

$$\hat{f}_{D,\Delta P} = f_{P_0} + \hat{S}\hat{\varepsilon}_{\mathrm{FFT}} \, . \tag{4.49}$$

However, (4.41) is stochastic, non linear, and does not admit a close inverse form. It is then necessary to rely on a numerical approach to compute the correction. According to (4.42), P_0 is the maximum value of the FFT modulus squared and is equal to $|R[i_{P_0}]|^2$. P_- and P_+ are the values of the peak on the left and of the right of P_0. They assume the value $|R[i_{P_0}-1]|^2$ and $|R[i_{P_0}+1]|^2$ respectively. Then, the second highest peak, adjacent to P_0 is denoted P_1 and defined as

$$P_1 = \max(P_-, P_+) = \left|R[i_{P_0} + \hat{S}]\right|^2 . \tag{4.50}$$

In the absence of noise, the true Doppler lies in the middle between P_0 and P_1; therefore, if P_1 is right with respect to P_0, $\hat{S} = +1$ and the correction is summed, whereas if P_1 is left, then $\hat{S} = -1$ and the correction is subtracted.

According to the results presented in Section 4.2.2, P_0 and P_1 can assume different values depending on the initial frequency error, and in particular:

- when $\varepsilon_{\mathrm{FFT}} = 0$, as shown in Figure 4.16a and Figure 4.16b, P_0 assumes the maximum value and P_1 is theoretically equal to 0, since it samples one of the roots of the sinc function. Their ratio is then equal to 0.

- when $\varepsilon_{\mathrm{FFT}}$ is maximum, equal to $\delta f/2$, as outlined in Figure 4.16c, P_0 and P_1 are aligned and their ratio is equal to 1.

This suggests the definition of a new discriminator denoted ΔP, as the ratio of the two peaks:

$$\Delta P = \frac{P_1}{P_0} . \tag{4.51}$$

Other definitions are possible, depending on the requirements of the implementation; for example, the square operation in the definition of P_0 and P_1 could be omitted. The theoretical value of ΔP in the absence of noise for each value of the frequency error $\varepsilon_{\mathrm{FFT}}$ is equal to:

$$\begin{aligned}
\Delta P = \frac{P_1}{P_0} &= \frac{|X[i_{P_0}+S]|^2}{|X[i_{P_0}]|^2} \\
&= \left| \frac{\sqrt{\frac{C}{2}}\,\mathrm{sinc}(\Delta f T_\mathrm{c}) e^{j\theta} e^{-j\pi\left(\Delta f T_\mathrm{c} + \frac{i_{P_0}+S}{K}\right)(K-1)} \frac{\sin\left(\pi \Delta f K T_\mathrm{c} + \pi(i_{P_0}+S)\right)}{\sin\left(\pi\Delta f T_\mathrm{c} + \frac{\pi(i_{P_0}+S)}{K}\right)}}{\sqrt{\frac{C}{2}}\,\mathrm{sinc}(\Delta f T_\mathrm{c}) e^{j\theta} e^{-j\pi\left(\Delta f T_\mathrm{c} + \frac{i_{P_0}}{K}\right)(K-1)} \frac{\sin\left(\pi \Delta f K T_\mathrm{c} + \pi i_{P_0}\right)}{\sin\left(\pi\Delta f T_\mathrm{c} + \frac{\pi i_{P_0}}{K}\right)}} \right|^2 \\
&= \left| e^{-j\pi\frac{S(K-1)}{K}} \frac{\sin\left(\pi\Delta f K T_\mathrm{c} + \pi i_{P_0} \pm \pi\right)}{\sin\left(\pi\Delta f K T_\mathrm{c} + \pi i_{P_0}\right)} \frac{\sin\left(\pi\Delta f T_\mathrm{c} + \frac{\pi i_{P_0}}{K}\right)}{\sin\left(\pi\Delta f T_\mathrm{c} + \frac{\pi(i_{P_0}+S)}{K}\right)} \right|^2 \\
&= \left| \frac{\sin\left(\pi\Delta f T_\mathrm{c} + \frac{\pi i_{P_0}}{K}\right)}{\sin\left(\pi\Delta f T_\mathrm{c} + \frac{\pi(i_{P_0}+S)}{K}\right)} \right|^2 .
\end{aligned} \tag{4.52}$$

The theoretical curve showing the trend of ΔP in the absence of noise versus the FFT error is depicted in Figure 4.18 and is denoted *P-curve.* So, after computing the value of ΔP, the correction factor $\hat{\varepsilon}_{\mathrm{FFT}}$ can be derived from the *P-curve.* Two implementations are possible, either to read the value from a LUT stored in memory or to compute it each time. The first is faster and more computationally efficient, but requires storing capabilities and features limited resolution The Doppler frequency estimate is finally derived, according to (4.49).

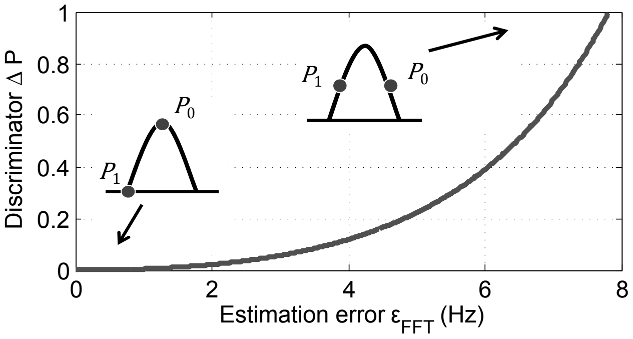


Figure 4.18: The theoretical *P-curve*, in the absence of noise.

In the case considered above, with reference to Figure 4.10, we have $\Delta P \simeq 0.67$. This value corresponds in the *P-curve* to $\hat{\varepsilon}_{\text{FFT}} \simeq 7.031$ Hz. Since in this case P_1 is on the right with respect to P_0, the final Doppler frequency is obtained by summing the correction ($\hat{S} = +1$):

$$\hat{f}_{D,\Delta P} = f_{P_0} + \hat{S}\hat{\varepsilon}_{\text{FFT}} = 493 + 1 \cdot 7.031 = 500.031 \text{ Hz}. \tag{4.53}$$

Summing up, and recalling that $f_D = 500$ Hz, the results reported in Table 4.5 are obtained.

Table 4.5: Summary of frequency estimation for different techniques.

Technique	Estimated Doppler	Error on the estimate
FFT maximum	$\hat{f}_{D,FFT} = 493$ Hz	7 Hz
Discrete sinc interpolation	$\hat{f}_{D,\text{sinc}} = 497.22$ Hz	2.78 Hz
Correction term	$\hat{f}_{D,\Delta P} = 500.031$ Hz	0.031 Hz

The noise impact

Nonetheless, the impact of noise has to be accounted for. Recalling the expression of $H_I[i]$ and $H_Q[i]$, the noise impact does not depend on the frequency, but only on the C/N_0, the coherent integration time and the number of FFT points. However, because of the slope of the *P-curve*, the error on $\hat{\varepsilon}_{\text{FFT}}$ in Hz due a fixed error on ΔP will impact much more where the *P-curve* of Figure 4.18 is flatter, that is for small values of $\hat{\varepsilon}_{\text{FFT}}$; on the contrary, for values of ΔP close to 1, where the curve is steeper, the impact will be lower. This is clarified in Figure 4.19, where it is outlined how the same error in the ΔP axis is translated in an error of different magnitude in the $\hat{\varepsilon}_{\text{FFT}}$ axis. Consequently, the ideal working condition is where the curve is steeper, i.e. in correspondence of initial frequency errors of about $\delta f/2$, in order to reduce the effect of noise in the frequency estimation process. It can be stated that the performance of MLE algorithms changes depending on the frequency difference within the FFT bin, confirming a well known fact in literature [51]. In the following, the impact of this in the estimate accuracy is illustrated, and a solution is proposed.

At the same time, some semi-analytical simulations carried out show how the noise affects the *P-curve*. Figure 4.20 depicts its trend in the absence of noise and when the input signal is characterized by a C/N_0 equal to 35 and to 45 dBHz respectively, confirming that the lower is the C/N_0, the higher is the noise.

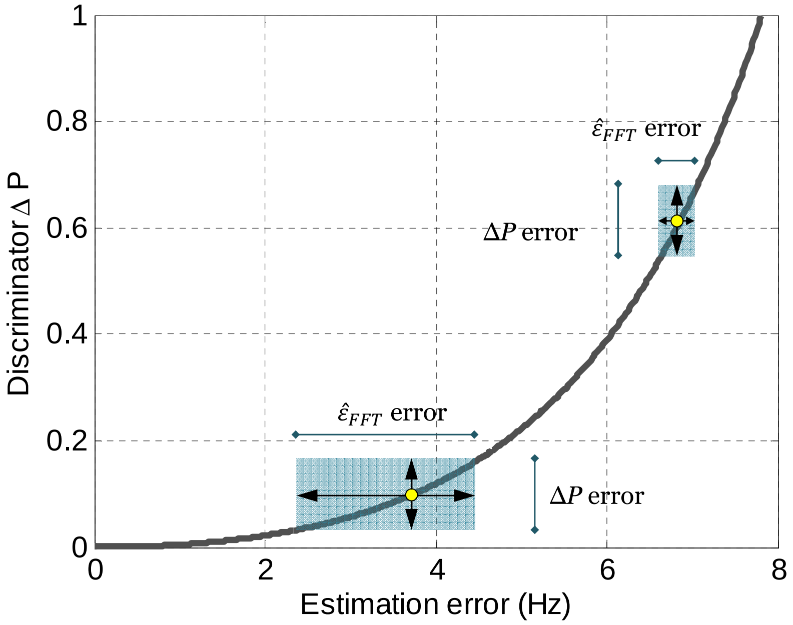


Figure 4.19: Different effect of the noise on ΔP and on the estimate of $\hat{\varepsilon}_{\mathrm{FFT}}$.

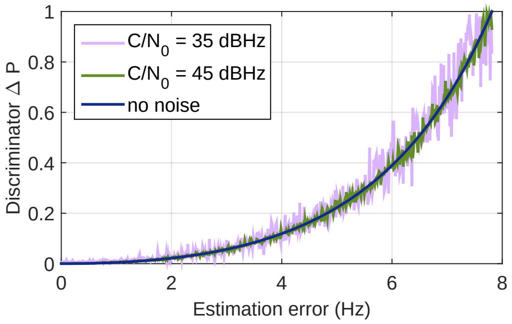


Figure 4.20: Theoretical *P-curve* in the presence of noise.

The problem of the correction sign estimation

It has been shown that when the initial frequency error is close to 0 Hz or to integer multiples of δf, the left and right peaks P_- and P_+ have almost the same value (a very similar situation is depicted in Figure 4.16a and 4.16b). However, in the presence of noise, their magnitude can oscillate, as proved in Figure 4.17b, leading to an error in solving (4.50). In particular, considering Figure 4.17b, where the initial frequency error is close to zero (equal to 1 Hz), it is clear how P_- is lower than P_+ in the case of absence of noise and for $C/N_0 = 45$ dBHz but higher in the case of $C/N_0 = 35$ and 25 dBHz, thus leading to a wrong determination of P_1 and to an error in the estimation of the sign $\hat{S}$. Figure 4.21 shows the event rate of correct estimation of the sign over 10 000 realizations, for an input frequency error in the range $[-\delta f/2; \delta f/2]$ and for three different values of C/N_0. The figure confirms that the sign estimation error increases for values of Δf approaching zero, while is practically absent for $|\Delta f| > 4.5$ Hz, 2.5 Hz and 1.4 Hz for $C/N_0 =$ 35 dBHz, 45 dBHz and 55 dBHz respectively.

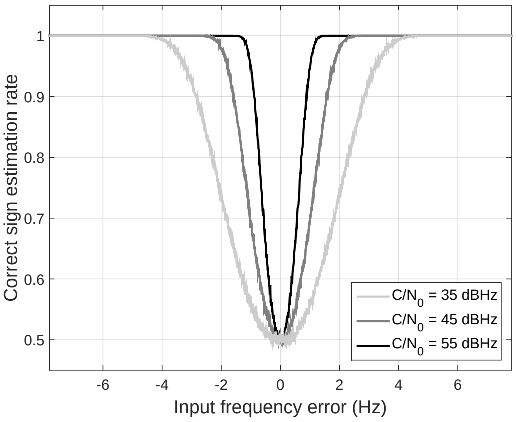


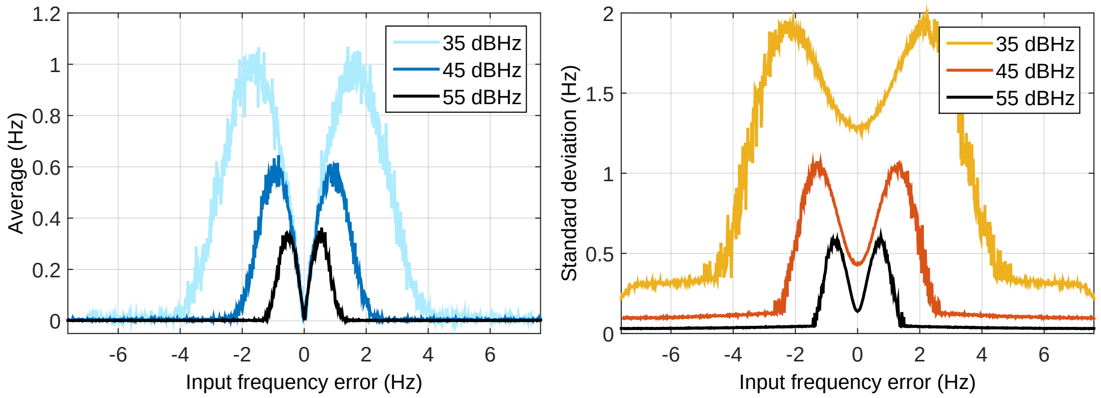
Figure 4.21: Correct sign estimation event rate for 10 000 realizations and for different C/N_0 and Δf.

When this occurs, independently on the accuracy of the correction, an error is introduced: the larger is the value of the correction, the larger is the error.

This is the main reason of the decreased estimate accuracy in the regions where the value of the candidate for the P_1 point is more sensible to noise. Unfortunately, this error occurs more frequently as the noise increases; in addition, the larger is the noise, the larger is its impact also for regions further away from the bin centre. At the same time, as it will be confirmed later on, also this effect is minimum for misalignment frequencies of $\delta f/2 + l$, where l is a positive or negative integer.

Figure 4.22a illustrates this behaviour. A semi-analytically simulation has been carried out to compute the average frequency estimate error, in absolute value, over 1000 realizations, for an input frequency error in the range $[-\delta f/2; \delta f/2]$ and for three different values of C/N_0: 35, 45 and 55 dBHz. First, the symmetric behaviour with respect to the origin is outlined, confirming the periodicity described above. Then, as expected, the error increases as long as the C/N_0 decreases. For example, for $\Delta f = 1$ Hz, the output error is only about 0.07 Hz when the signal C/N_0 is equal to 55 dBHz, but increases to 0.6 Hz for a signal 10 dB weaker and to 0.8 Hz in the case of $C/N_0 = 35$ dBHz. The figure also confirms that this error is more frequent when Δf is close to 0. Nevertheless, it should be reminded that the error is proportional to the size of the correction; so, for values very close to 0, the correction is lower, and despite the fact that the correction is applied in the wrong direction, the average frequency error is limited. As far as the initial frequency error moves away from 0, the size of the correction increases, and so the final error increases as well. After a certain point, corresponding to the maximum error, the probability of wrongly estimating the sign of the correction decreases, and thus the average error decreases. In addition, the frequency region affected by the error in the sign decision increases for lower C/N_0; this is due to the fact that the stronger is the signal, the less noisy are the FFT points, and so the smaller is the probability to mistake P_- and P_+. As stated above, the error is minimum for misalignment frequencies of $\delta f/2 + l$.

Similarly, the standard deviation of the frequency estimate for the same semi-analytical simulation is reported in Figure 4.22b. The same considerations can be drawn: the standard deviation increases for values of Δf close to 0, and slightly decreases for values very close to 0. In addition the maximum standard deviation moves far from $\Delta f = 0$ Hz as long as the C/N_0 decreases, for the same reasons described above.



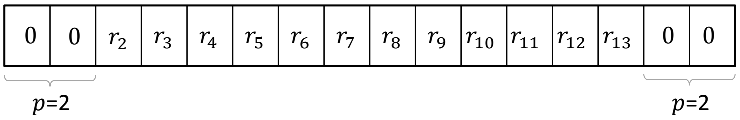
(a) Average

(b) Standard deviation

Figure 4.22: Frequency estimate error vs. the initial frequency error for different C/N_0, $\delta f = 15.625$ Hz.

4.2.4 Zero forcing

A possible solution to overcome this problem is to design a better algorithm for estimating the sign $\hat{S}$. According to some simulations, it is indeed expected that this method can at least reach the same performance for any value of Δf, when solving the issue of the correction sign. For example, instead of taking just the maximum between the P_- and P_+, it is possible to average several points on the left and on the right, to smooth the effect of noise. A good solution, even if more computational demanding, is to compute a second FFT, changing the parameters. For example, by computing an FFT on $K = 32$ points, with some zero padding (16 zeros at the end), two more points are present on the FFT peak, which does not sample the zeros of the sinc function, so they are less affected by noise, allowing to better discriminate the sign. In alternative, to maintain the similar computational burden, it is possible to force to zero some of the K points, so as to reduce the spanning interval and to change the positions of the zeros of the sinc; a different dis-alignment between these zeros and the FFT points is obtained, allowing to better define the shape of the peak.



0	0	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}	r_{13}	0	0

Figure 4.23: Graphical representation of the zero-forcing of a sequence of correlation values in input to the FFT block, with $K = 16$.

Figure 4.23 depicts an example of this approach, denoted *zero-forcing*; $p = 2$ values are overwritten with 0, both at the beginning and at the end of the K long sequence $r[k]$ of correlation values. This is equivalent to window the signal at the end of the correlation with a boxcar function b_p, defined as:

$$b_p[k] = \begin{cases} 1 & p \leq k \leq K-1-p \\ 0 & \text{elsewhere} \end{cases} . \tag{4.54}$$

Let's then define $r_p[k] = r[k]b_p[k]$ the correlation zero-forced samples, and $x_p[k]$ and $h_p[k]$ their signal and noise components respectively, where the subscript p stands for *zero-forcing*. A new FFT on the correlation samples is introduced: the signal component $X_p[i]$ can be computed as:

$$\begin{aligned}
X_p[i] &= \sum_{k=0}^{K-1} x_p[k] e^{-j\frac{2\pi}{K}ik} \\
&= \sum_{k=0}^{K-1} x[k] b_p[k] e^{-j\frac{2\pi}{K}ik} \\
&= \sum_{k=p}^{K-1-p} x[k] e^{-j\frac{2\pi}{K}ik} \\
&= \sum_{k=p}^{K-1-p} \sqrt{\frac{C}{2}} \operatorname{sinc}(\Delta f M T_s) e^{j\theta} e^{-j2\pi\Delta f k M T_s} e^{-j\frac{2\pi}{K}ik} \\
&= \sqrt{\frac{C}{2}} \operatorname{sinc}(\Delta f M T_s) e^{j\theta} \sum_{k=p}^{K-1-p} e^{-j2\pi k\left(\Delta f M T_s + \frac{i}{K}\right)} \\
&= \sqrt{\frac{C}{2}} \operatorname{sinc}(\Delta f M T_s) e^{j\theta} \frac{e^{-j2\pi\left(\Delta f M T_s + \frac{i}{k}\right)p} - e^{-j2\pi\left(\Delta f M T_s + \frac{i}{k}\right)(K-p)}}{1 - e^{-j2\pi\left(\Delta f M T_s + \frac{i}{k}\right)}} \\
&= \sqrt{\frac{C}{2}} \operatorname{sinc}(\Delta f T_\mathrm{c}) e^{j\theta} \frac{e^{-j2\pi\left(\Delta f T_\mathrm{c} + \frac{i}{k}\right)p} - e^{-j2\pi\left(\Delta f T_\mathrm{c} + \frac{i}{k}\right)(K-p)}}{1 - e^{-j2\pi\left(\Delta f T_\mathrm{c} + \frac{i}{k}\right)}} .
\end{aligned} \tag{4.55}$$

Recalling (4.36), it is possible to prove that also in this case the noise is zero mean, for any value of the parameter p. Recalling (4.38), the variance of the windowed noise can be computed. The energy of the window is

$$E_w = \sum_{k=0}^{K-1} |b_p[k]|^2 = \sum_{k=p}^{K-1-p} |1|^2 = K - 2p \,, \tag{4.56}$$

thus

$$\sigma^2_{H_p} = (K - 2p)\sigma^2_\eta \,. \tag{4.57}$$

Thereby, the noise samples at the output of the FFT block in the case of zero-forcing are Gaussian random variables:

$$H_{I,p}[i], H_{Q,p}[i] \sim \mathcal{N}\left(0, \, \frac{(K-2p)N_0}{4T_\mathrm{c}}\right) . \tag{4.58}$$

Summing up, the expression of the zero-forced FFT output is:

$$R_p[i] = X_p[i] + H_{I,p}[i] + jH_{Q,p}[i] \,. \tag{4.59}$$

The main consequence is that by increasing p, the variance of the noise at the FFT output decreases. However, also the amplitude of the FFT points of the useful signal component decreases. In general, since less correlation values are used, some energy is wasted. Nevertheless, as expected, the points $|R[i]|^2$ move, as clarified in Figure 4.24: both P_- and P_+, theoretically equal to 0, increase. This produces several effects on the algorithm for estimating the frequency:

- the probability of getting a wrong estimate of the sign is lower, because P_- and P_+ are further away from the zero of the sinc function;
- the *P-curve* is different, because the ratio ΔP assumes different values;
- the energy of the FFT is lower, some useful signal information is wasted.

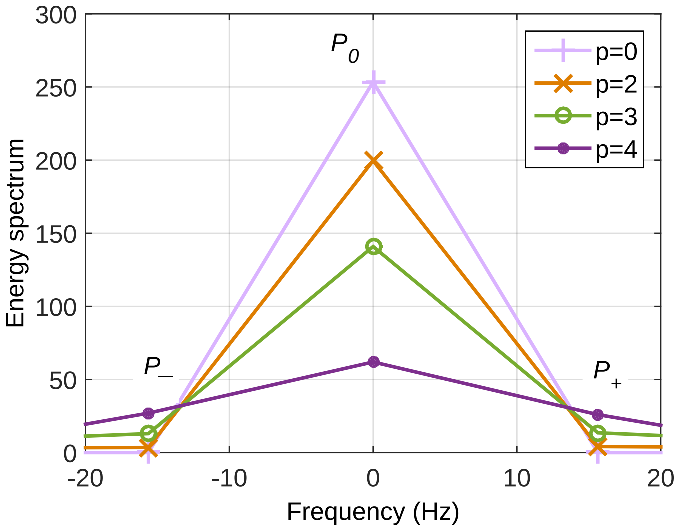


Figure 4.24: Effect of different values of p in the FFT result in the absence of noise.

Wrong sign estimate probability

Table 4.6 reports the values of P_-, P_+, and the absolute value of their difference for different p, in the absence of noise, and for $\Delta f = 0.1$ Hz and $\Delta f = 1$ Hz respectively. Just by looking at the last column it is clear that in the case in which there is no zero-forcing ($p = 0$) the two points are closer. Just by zero-forcing with 1 or 2 zeros per side their difference increases, easing the decision on which is the maximum. Since the noise on the FFT output is constant (as proved in Section 4.2.1), it can be stated that the larger is the difference in absolute value between the two points around the main peak, the larger is the probability of correctly estimating $\hat{S}$.

Table 4.6: Value of P_- and P_+ for different p.

	$\Delta f = 0.1$ Hz			$\Delta f = 1$ Hz		
p	P_-	P_+	$\lvert P_- - P_+\rvert$	P_-	P_+	$\lvert P_- - P_+\rvert$
0	0.0108	0.0105	2.68×10^{-4}	1.19	0.93	0.27
1	4.2	3.5	0.75	8.7	1.1	7.61
2	13.7	12.6	1.12	19.4	8.2	11.25
3	22.0	21.9	1.03	27.9	17.6	10.32

A useful metric for establishing the best value of the parameter p is the *probability of correct sign estimation*, P_S, i.e. the probability that the sign estimate $\hat{S}$ is correct. P_S assumes the maximum value 1 when the sign of the correction is correctly estimated:

$$\hat{S} = \begin{cases} +1 & \text{if} \quad f_{P_0} < f_D \\ -1 & \text{if} \quad f_{P_0} > f_D \end{cases} . \tag{4.60}$$

On the contrary, it can be assumed that the worst situation happens when the noise completely dominates the decision of (4.50). Therefore, the sign estimate is a random variable assuming the values $\hat{S} = +1$ and $\hat{S} = -1$ with the same probability. It can be expected that in half of the cases the sign is correct by chance, and in half of the cases it is wrong, leading to $P_S = 0.5$.

Despite the fact that it is theoretically possible to analytically compute this probability, the problem is hard to handle, and Monte Carlo simulations are exploited to compute *correct sign estimation event rates*. 30 000 simulation realizations are run, and each time the sign is estimated and compared with the real correction sign. The results, for the standard case ($p = 0$) and for zero-forcing with different p, are reported in Figure 4.25. In particular the dependence on the C/N_0 and on the initial frequency error Δf are outlined. As expected, P_S is close to 1 for high values of the C/N_0 and for large values of the initial frequency error. This simulation proves that the value of p giving the highest P_S is $p = 2$.

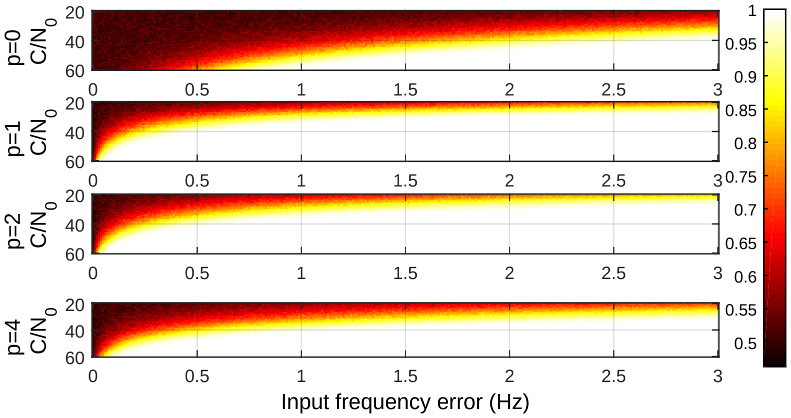
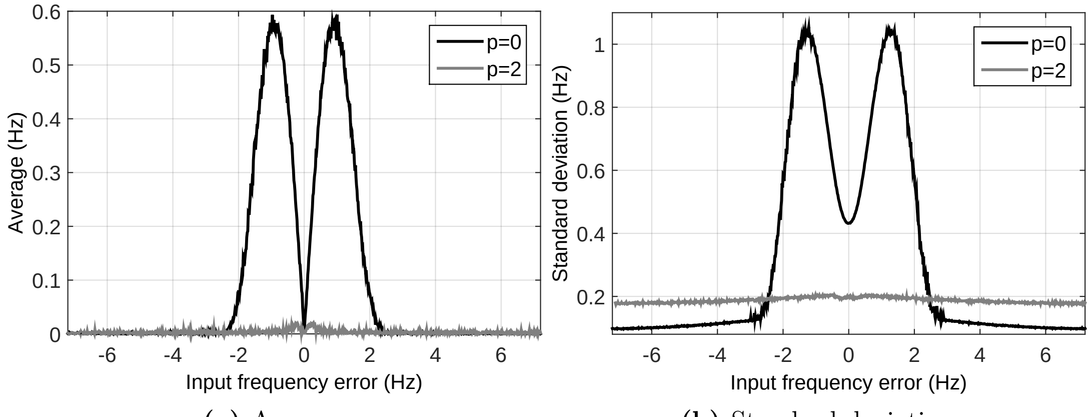


Figure 4.25: Correct sign estimation event rates for 30 000 realizations and for different C/N_0, Δf and p.

The same simulation run in Section 4.2.3 was performed also for the case with $p = 2$. The results, in terms of mean and standard deviation of the frequency estimate, for 10 000 realizations, are reported in Figure 4.26, in the case $C/N_0 = 45$ dBHz. The large errors present in the case without zero-forcing ($p = 0$) and due to the sign are clearly removed. Both the mean value and the standard deviation can be considered constant for each value of the input frequency error. However, the standard deviation is slightly higher than in the standard case: this is due do the fact that some signal energy is lost in replacing correlation samples with zeros. Despite this latter drawback, the technique of zero-forcing is able to solve almost completely the problem of the sign of the correction factor and brings an important improvement in the frequency estimation process.

A new P-curve

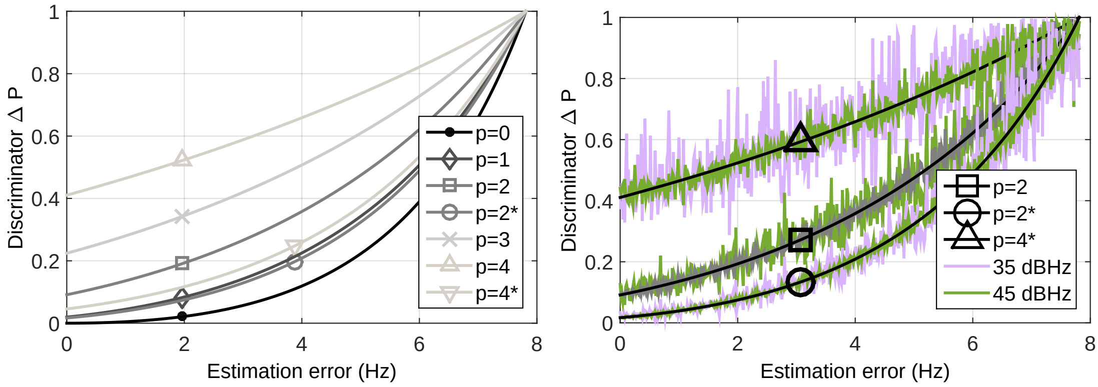
As a consequence of the new values of P_0 and P_1, also the *P-curve* changes aspect. This technique, in fact, despite its main objective is to reduce the error probability in estimating $\hat{S}$, offers also improvements in the correction factor estimation process. Figure 4.27a reports the theoretical *P-curve* in the standard case, when no zero-forcing is performed, and for different values of the parameter p. Figure 4.27b reports the same curve for a subset of cases in the presence of noise. First it has to be noticed that in the case of zero-forcing the curve does not start from the origin; this is due to the fact that for $\Delta f = 0$, P_- and P_+ does not sample the roots of the sinc function. Second, increasing the value of p, the curve tends to a straight line. This has important consequences on the impact of the noise in the correction factor. In fact, recalling the conclusions drawn from Figure 4.19, the impact of the noise in the correction factor estimation is higher where the *P-curve* is flatter; by increasing the value of p, the slope of the curve in these points is increased, thus reducing the noise impact. Third, Figure 4.27b confirms that the impact of noise is stronger when



(a) Average **(b)** Standard deviation

Figure 4.26: Frequency estimate error for different input frequency errors and different values of p.

the value of p is higher.



(a) Absence of noise **(b)** Presence of noise at different C/N_0

Figure 4.27: Frequency estimate error average for different and p and for the double FFT case.

4.2.5 The double FFT

A final improvement can be obtained by introducing a second FFT. Indeed, the full K-point FFT exploits all the signal power and assures good performance in the regions around $\delta f/2 + l$. At the same time, the zero-forced FFT can solve most of the problems in the region around $\delta f + l$ but offers a slightly worse performance in terms of standard deviation of the error. The advantages of both techniques can be obtained by computing two different FFTs and by combining the two contributions, for instance summing their result. This innovative method is hereinafter called Double Fast Fourier Transform (DFFT). It can be proved that since the zeros on the correlation values are inserted symmetrically, no phase changes are introduced. Recalling (4.41) and (4.59), the expression of the DFFT output R_d is equal to:

$$R_d[i] = R[i] + R_p[i]\,, \tag{4.61}$$

where the subscript d stands for *double FFT*.

In Figure 4.27, the theoretical *P-curves* for the DFFT case are reported. The label $p = 2^*$ and $p = 4^*$ identify the *P-curve* of $R_d[i]$ for $p = 2$ and $p = 4$ respectively. It is interesting to notice that the *P-curve* of $R_d[i]$ for $p = 2$ is close to the one of $R_p[i]$ for $p = 1$; however, the latter still suffers for a larger correction sign error. At the same time, Figure 4.27b confirms that the noise is reduced when passing from FFT ($p = 2$) to DFFT ($p = 2^*$).

What is more, the evaluation of two FFTs almost doubles the computational burden, which is often not affordable. By exploiting the linearity property of the FFT, it is mathematically equivalent to first window the signal and then to compute one single FFT, with a large complexity reduction. It can be proved that

$$R_d[i] = \mathbf{FFT}\left\{r[k]\bar{b}_p[k]\right\}\,, \tag{4.62}$$

where

$$\bar{b}_p[k] = \begin{cases} 2 & p \leq k \leq K-1-p \\ 1 & 0 \leq k \leq p-1 \quad \text{or} \quad K-p \leq k \leq K-1 \end{cases}. \tag{4.63}$$

This approach is depicted in Figure 4.28 for $p = 2$ and $K = 16$. In this way the computational burden of the method is kept within the same level of a standard K-point FFT, but the accuracy of the estimates considerably improves, as will be proved in the next section.

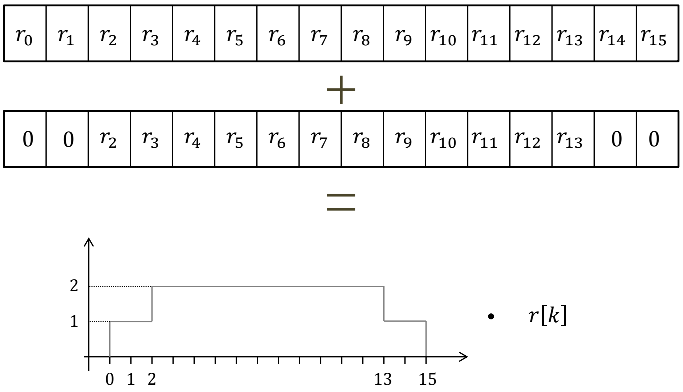

r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
+
0 0 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} 0 0
=
2
1
0 1 2
13 15
• $r[k]$

Figure 4.28: Graphical representation of the windowed DFFT.

4.3 On demand power saving processing

A smart power management is an essential feature for mass-market receivers, since most of the devices equipped with a GNSS chipset (with the exception of navigation receivers for automotive applications) run only on batteries. In fact, despite the recent implementation improvements, mass-market receivers still feature a significant drawback: they are extremely power hungry. Some experiments showed that a general Location Based Application (LBA) running on a smartphone for one hour reduces the power level from 100% to 79%, whereas the same application with GPS disabled, i.e. using alternative positioning technologies, reduces it only to 94%. It has been stated that continuous GPS tracking can deplete a 1000 mAh battery in only 6 hours [57]. The goal of

GNSS chip manufacturers is then to improve at maximum the battery duration by reducing the power consumption.

Different solutions can be adopted as reported in the literature. A possible classification of such solutions is based on the different stage of the receiver in which the power saving layer techniques operate:

- **user-level** solutions, aimed at powering on and off the chip, depending on its real need [58];
- **application-level** solutions, based on switching to alternative location sensing mechanism or sensors, depending on the accuracy required (substitution) or on the user state (suppression) [57];
- **phone-level** solutions, aimed at synchronizing the location requests of different PDAs (piggybacking);
- more **advanced phone-level** solutions, able to directly control the GNSS chip and to trigger it to lower power modes in presence of aiding from inertial or accelerometer signals [59] or of cooperation with other nodes [60].

Most of such strategies are based on the concept that a mass-market receiver based on snapshot processing is not required to constantly compute a PVT solution. In fact, most of the time, GNSS chipsets for consumer devices only need to keep updated information on approximate time and position and to download clock corrections and ephemeris data, at a proper time rate, depending on the navigation message type and the adopted extended ephemeris algorithm. Then, when asked, the receiver is able to provide a position fix in a short time interval. By reducing the computational load of the device during the waiting mode, the power consumption is proportionally reduced.

In this section a comparative analysis of in-chip solutions is given, able to solve the problem at the receiver/algorithm level, as described by some patents hold by consumer manufacturers [61]. Such techniques are based on the definition of a new state, the *sleep state*, in addition to the standard active state (receiver turned on) and off state (receiver turned off). By similarity to a square wave, they are named Duty Cycle (DC) tracking techniques. The validity of those techniques for Galileo signals and under different channel and scenarios is proven exploiting a fully software approach.

4.3.1 Assessment of the state-of-the-art duty cycle power saving techniques

Many different tracking techniques can be adopted, even inside the same chipset, according to changed environmental or dynamic conditions:

- *open-loop tracking*, which requires less resources than standard tracking based on closed-loops.
- *Non-continuous block tracking* as described in some Broadcom patents [35, 36].
- *DC tracking*, in which the tracking only operates on a fraction of data in a given data window and then coast to the beginning of the next data window.

This latter technique is mainly related to some tracking modes covered by several CSR (SiRF) patents. Four hardware-based power states based on five software-based power management modes, enabling power saving features, can be identified. More details can be found in [62, 63].

The hardware-based power states, the power consumption levels of which are reported in Table 4.7, are:

- **Full-power state.** This is the initial state of the receiver where all RF circuitry and base-band modules are fully powered, in which GNSS receivers usually operate. There is a further difference in power consumption during acquisition mode and tracking mode: during acquisition, processing is more intense, thus consuming more power.

- **CPU-only state.** This is the state where the LNA is shut off, but the other parts of the RF section, including the TCXO, are still powered in order to provide a clock to the CPU. This state is entered when the satellite measurements have been collected, so that some modules are no longer needed and can be tuned off, but the navigation solution still needs to be computed.

- **Stand-by state.** In this state, the RF section is completely powered off and the base-band clock is stopped; only a Real Time Clock (RTC) and battery-backed RAM are active. The receiver enters this state once a position fix has been computed, but before shutting down the RTC, a wake-up register is programmed to turn the system on sometime in the future. Alternatively, an external host can wake the receiver up.

- **Hibernate state.** It is a low quiescent power state where only the internal I/O Keep Alive, non-volatile RTC, patch RAM and backup RAM block are powered on, as well as the main supply input. The receiver wakes up from Hibernate state using all internal aiding (like GPS time, ephemeris, and last position) resulting to a fastest possible TTFF in either Hot or Warm start modes.

Table 4.7: Power consumption levels according to the power state.

State	Typical consumption (mA)
Full power (acquisition)	80
Full power (tracking)	60
CPU only	36
Stand-by	1
Hibernate	0.05

The software-based power management modes are:

- **Full Active (FA)** is the usual continuous navigation mode, in which the receiver is always in *full-power stage.*

- **Adaptive TricklePower™ (ATP)** is a duty cycling mode with a user-selectable position update interval, providing high quality accuracy and dynamic motion response at a reduced average current draw. In this mode the receiver stays at *full-power state* for 200 to 900 ms and provides a valid fix. Between fixes the receiver stays in *hibernate state* with 1 to 10 s interval. The receiver stays once in a while in *full-power mode* automatically (typically every 1800 s) to receive new ephemeris data from rising satellites or if received signal levels drop below certain level. The *adaptive* adjective only means that when TricklePower™ (TP) mode is enabled and the fourth strongest satellite signal is below the 26 dBHz threshold, the *full-Power mode* is automatically selected. The TP mode is then re-entered when such signal power is greater than 30 dBHz again. This is just to deal with the ability to track weak signals.

- **Advanced Power Management (APM)** allows power savings while ensuring that the quality of the solution is maintained when signal levels drop. APM does not engage until all necessary information are received. Host can configure the number of APM cycles, time between fixes and max position error. Rest of the time the receiver stays in *hibernate state*. It is designed for use in wireless networks, as it needs assistance: there is no mode in the APM which allows for the collection of ephemeris of the satellite data stream.

- **Push-To-Fix™ (PTF)** allows for infrequent, on-demand user position requests with short TTFF. In this mode the receiver is configured to wake up periodically, typically every 1800 s, for position update and to collect new ephemeris data from rising satellites. Rest of the time the receiver stays in *Hibernate state*. When position update is needed, the host can wake up the receiver and a valid fix is available within 1 to 8 s.

- **Micro Power Management (MPM)** is a sophisticated very low-power management mode that keeps the receiver in hot start conditions while maximizing the time spent in a very low power state, so as to remain below a stated average current level while maintaining a low level of uncertainty in time, frequency, position and ephemeris state. In this mode the receiver is configured to wake up periodically for 18 s, typically every 1800 s, to collect new ephemeris data from rising satellites, and also every 600 s to calibrate RTC. Rest of the time the receiver stays in *Hibernate state*. Given its complexity from a timing point of view and since long sleep states are more difficult to simulate, this mode is not considered for software receiver simulations.

All the last three modes perform similarly in principle but provide different output rates and reliability and they can be referred to the previous patent. They all share a common baseline. The SiRFStarIV receiver starts up in a *hibernate state* and remains there until a signal is sent to it. Then the receiver stays in *full-power mode* until the user's position is fixed and all relevant information is gathered. It takes about 30 to 40 s on average to compute the first position fix and extract other information. The time is shorter in cases with aided or hot starts. When the receiver is ready to carry out normal processing, different sections of hardware can be turned off or un-clocked, depending on the receiver state. After processing is completed, the receiver programs the RTC to wake up at a specified time and then go to sleep by turning off most of the circuitry except the RTC. When the wake up interrupt occurs, the receiver re-starts the system and resumes GPS tasks. The clock accuracy must be very high: the time of wake-up must be accurate to the order of several microseconds.

Extension to Galileo signals structure

All the techniques described were especially designed for GPS L1 C/A signals; nevertheless, as soon as new constellations and modernized signals are available also for the consumer market [10], on demand algorithms can be re-designed and can account for some benefits. The main impact of Galileo signals is indeed the presence of the *pilot signal*: a longer integration time could allow reducing the *full-power state* tracking and then the power drain. Second, the different structure of the navigation message leads to a change in the periodicity of the states when data demodulation is required. For example, the impact on the total average power consumption is expected lower since these updates have a lower repetition rate.

4.3.2 The open-loop approach

The common disadvantage of most of the DC techniques when applied to the traditional receiver structure is that each time that the receiver is set to the active state after an off period, it goes in the so-called re-acquisition state. Even if hot start technology is nowadays consolidated, it is evident that this process is time, energy and resources consuming. Traditional tracking loops, such as PLLs and DLLs, require that, at each epoch, the estimates of the Doppler frequency and of the code delay affecting the signal are kept within a pull-in region, dependent on the equivalent loop bandwidth [5]. When continuous tracking is performed this is not a problem, as long as the dynamics of the user are limited and the C/N_0 is good enough, and provided that the loops are properly designed. If for some reasons (such as unexpected high dynamics, high clock uncertainty, deterioration of the signal quality, ionospheric and tropospheric impairments, or ephemeris errors) the true code delay or the Doppler frequency of the incoming signal drift apart from the values used for performing code and frequency wipe-off respectively, a LOL is declared, and the receiver exits the tracking stage. This situation, that is quite rare in continuous tracking mode, may become much more likely if a sleep state is introduced: hence during this sleep state it is necessary to propagate and predict the estimates of the Doppler frequency and of the code delay. Still, because of all the error sources listed above, it is almost impossible to keep the estimates within the required level of accuracy, so as to avoid LOL.

As a consequence, an **open-loop snapshot** solution has to be adopted for tracking GNSS signals when DC algorithms are employed. Since at each snapshot the code delay and Doppler frequency estimates need to lie within the search space, proper designed estimates propagation algorithms are required.

4.3.3 A proposed technique

Among all the power saving techniques considered, the ATP mode was chosen as a baseline for the extension to Galileo signals, for its adaptation in a software receiver, for implementation and testing. In fact ATP mode is best suited for applications that require:

- navigation solutions at a fixed rate;
- low-power consumption;
- maintaining the ability to track weak signals.

Furthermore it corresponds to the simplest implementation of DC algorithms.

Summarizing, the receiver turns to *full-power state* for the active time to collect data, and then operates in *stand-by state*; the next *full-power state* is initiated by an RTC wake up. Then TP mode continuously cycles through active and sleep state. However, there are some situations where the receiver stays in full-power mode. They are:

- to collect periodic ephemeris data,
- to collect periodic almanac data,
- to collect periodic ionosphere and Universal Time Coordinates (UTC) data,
- to perform RTC calibration,
- to improve navigation result.

After tracking completion, the receiver enters a lower power transitional state in which the RF section is powered down while the base-band processor continues to run (CPU only state). The receiver remains in this baseband-only transitional state while it computes position from the range measurements and then outputs the resulting data. Thus the time spent in this baseband-only state depends upon the time taken to complete the transmission of any messages over the serial host interface.

A DC layer has been added to the receiver structure, exploiting the extreme flexibility offered by software receivers. The states alternation, as described in Section 4.3.1 is maintained, but the fully software approach requires a re-definition of the states, reducing their number to two, as reported in Figure 4.29:

- in the *active state* the receiver works in the normal way, continuously tracking the signal and providing continuous updates of the observables;

- in the *sleep state* the turning off of the hardware components and the RTC setting is emulated, by skipping a proper number of samples of the input file.

Since in a fully software implementation there are no hardware power hungry modules that can be de-activated, the other low power states are removed.

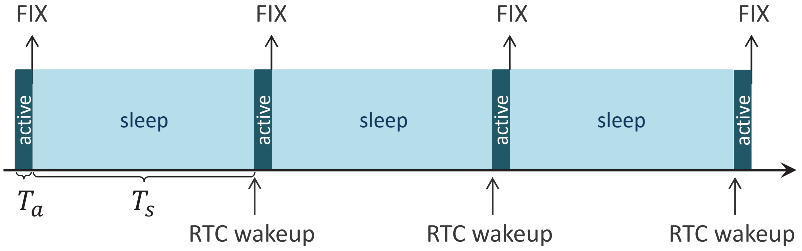


Figure 4.29: TP states alternation as implemented in the software receiver.

The parameters and drivers used to define the length of the states are important design features and they need to be investigated.

T_a**:** The active period length depends on the signal strength, estimated according to the measurements of the previous epoch. Under weak signal conditions, a tracking mode with a long integration is selected. There are in principle no limitations concerning pilot signals, or situations in which assistance data are available. On the contrary, for the legacy GPS C/A signal, the knowledge of Telemetry (TLM) and Handover Word (HOW) data bits at the start of each navigation frame, which last 1.2 s and repeat every 6 s, can help in extending the integration time. Afterwards, the receiver goes to the sleep state for the remaining period of 4.8 s.

T_s**:** The sleep period length is programmed according to the receiver dynamics, mainly acceleration and speed, estimated during the latest active state, as shown in Figure 4.30 [61]. In particular, if any non-zero acceleration is detected, the receiver is not allowed to enter the sleep state and stays in the active state. On the contrary, if acceleration is zero, the receiver is either static or moving with constant velocity; under these latter conditions, the receiver can enter the sleep mode. Therefore, in static condition, the sleep period can be set to the maximum, while its length should be reduced when the velocity increases.

The values recommended by CSR/SiRF are reported in the first two rows of Table 4.8. However, in order to cope with the structure of the multi-correlator block of the software receiver, that works with blocks of 64 ms, a different combination is used as a baseline, and reported in the third row of Table 4.8.

Exploiting the flexibility offered by the software approach, it is possible to test the effect of the length of both states, measured in term of accuracy of the estimates. It was observed that their setting depends on the channel conditions, on the signal strength, on the number of satellites in view, on the user dynamics and finally on the required accuracy.

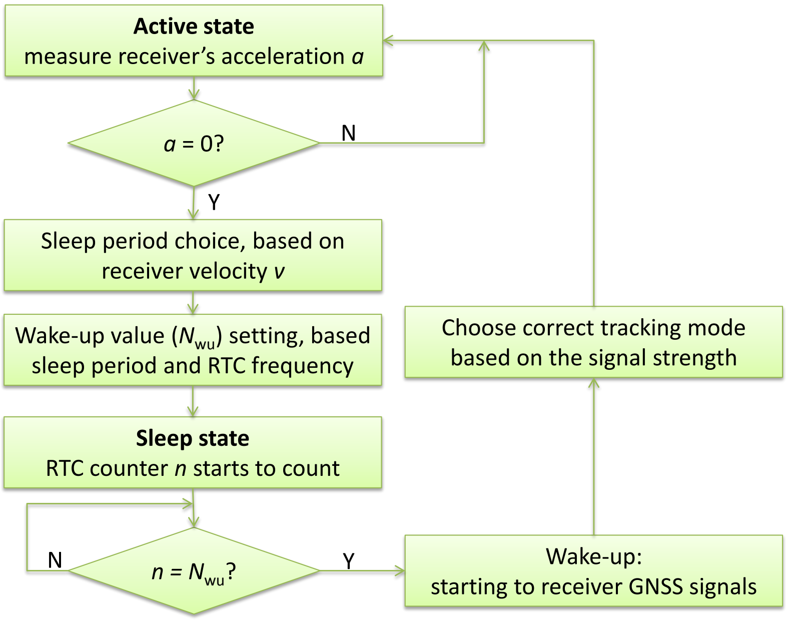

Active state
measure receiver's acceleration a
a = 0?
N
Y
Sleep period choice, based on receiver velocity v
Wake-up value (N_{wu}) setting, based sleep period and RTC frequency
Sleep state
RTC counter n starts to count
N
n = N_{wu}?
Y
Wake-up:
starting to receiver GNSS signals
Choose correct tracking mode based on the signal strength

Figure 4.30: SIRF wake up strategy

It has also to be outlined that whereas in hardware implementations the direct benefit is the power saving, in a software implementation it is not possible to see such an improvement. The reduced power consumption is translated into a shorter processing time for each single channel. It is interesting to note that the DC approach can facilitate the implementation of real time (or quasi-real time) software receivers. In addition, even if the main drawback of using DC tracking techniques remains the decrease of the rate of observables and PVT, often, depending on the application, there are situations in which a position solution each second is more than enough.

4.3.4 Propagation of the estimates and prediction strategies

One of the main problems to face when dealing with DC techniques and open-loop snapshot processing is related to the prediction of the signal parameters. As detailed in Section 4.3.2, a reduced search space, both in the frequency and in the code delay domain, can be defined, but it is necessary that at any epoch the parameters estimates lie within it. It is however likely that during the sleep state the signal changes so much that the these parameters move: Figure 4.31 depicts the transition of the signal from the reduced search space centred around $\left\{f_D^A, \tau^A\right\}$ at epoch A to the reduced search space centred around $\left\{f_D^B, \tau^B\right\}$ at epoch B. If the estimates are not properly updated, the correlation peak cannot be found and an invalid measure is obtained. Therefore it is necessary to predict their trend, and to define a new search space at each RTC wake-up.

Table 4.8: DC values for the ATP mode.

T_a	T_s	Update interval	DC
300 ms	700 ms	1 s	30 %
400 ms	1600 ms	2 s	20 %
64 ms	960 ms	$\simeq$ 1 s	6.25 %

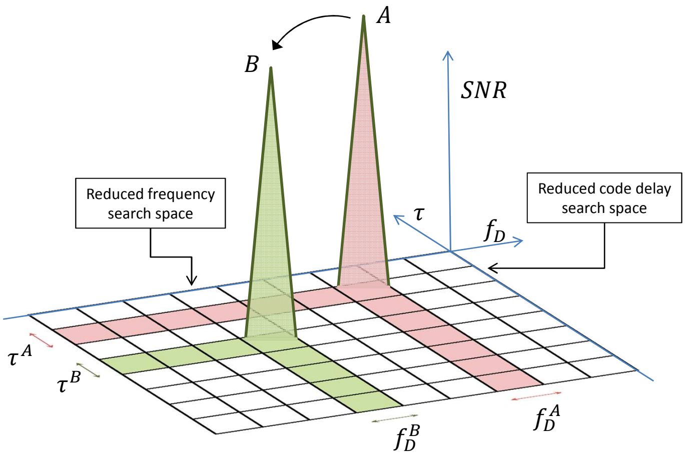

A
B
SNR
Reduced frequency search space
Reduced code delay search space
τ
f_D
τ^A
τ^B
f_D^B
f_D^A

Figure 4.31: Search space transition after long sleep periods.

The magnitude of rate of change of the frequency, for a static user, is up to 0.8 Hz/s [37] and the dimension of the frequency reduced search space is equal to 250 Hz, as reported in (4.12). However, for the vehicular dynamic scenario considered, described in the following (velocity equal to 15 m/s, circular trajectory), the Doppler frequency can increase up to 20 Hz/s, and therefore in a few seconds the Doppler frequency might exit from the pull-in region of the open-loop processing algorithms. Figure 4.32 shows the trend of the Doppler frequency of Galileo PRN 12 during the processing with the software receiver described in Section 5.1. The black line corresponds to the true Doppler frequency of the signal characterized by a high Doppler rate. The orange points correspond to the Doppler as estimated by the software receiver with a duty cycle equal to 3.125 %, when no particular prediction mechanism are exploited: as soon as the true Doppler reaches the bounds defined in (4.12), i.e. after about 14 s, the frequency estimate is clearly wrong, leading to an invalid measure.

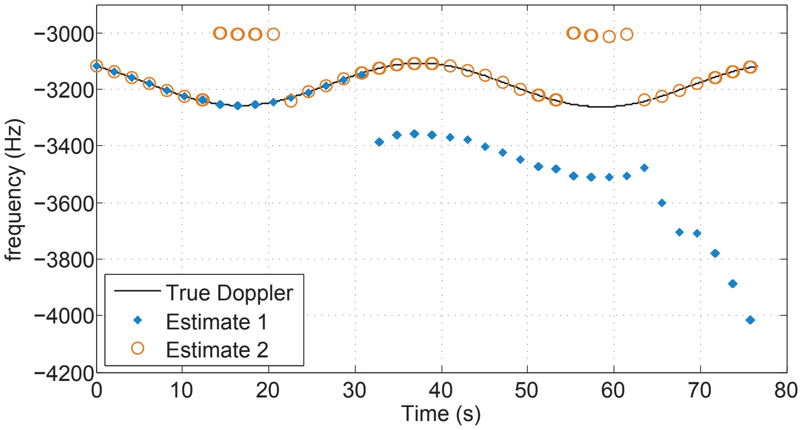


Figure 4.32: Frequency estimates when adapting DC and different prediction strategies in the vehicular case.

A propagation algorithm has been implemented to solve the problem: in particular, the value of the wipe-off frequency is changed when the frequency estimate approaches the range bounds, thus *quasi-closing* the loop, as described in [64]. However, it must be noticed that is it not sufficient to just take the last Doppler estimate and to check whether it is close to the FFT bounds or not; in fact, if for any reason, the estimate is wrong, an irreversible error is introduced, leading to invalid measures for all the following epochs. It is necessary to properly average the last estimates so as to further smooth the effect of noise. Different algorithms were tested, given M values of the frequency estimate $\hat{f}$, so as to compute a new wipe-off frequency candidate $\hat{f}_0$:

1. moving average of the last M values;
2. moving average of the last M values excluding the highest estimates in absolute value;
3. moving median of the last M values excluding the highest estimates in absolute value;
4. hatch Infinite Impulse Response (IIF) filter.

According to some tests, the Hatch filter outperforms the other methods, so it was chosen as a candidate. Summing up, at each epoch m, first the wipe-off frequency candidate is computed

$$\hat{f}_0[m] = w \cdot \hat{f}_0[m-1] + (1-w) \cdot \hat{f}[m] , \tag{4.64}$$

where w is the Hatch filter weight. Then, if this estimate is enough close to the range defined in (4.13), the value of the wipe-off is updated, to re-centre the frequency range around a new value for the following epoch.

This algorithms is able to correctly estimate the frequency, leading to estimates essentially equal to the black line of Figure 4.32. However, it must be pointed out that an inappropriate setting of the parameters, such as the filter weight w, can lead to wrong results. For example, $w = 0.9$, that was proved to be a good value for the pedestrian case, turned out to fail in the vehicular case, as depicted by the blue dots in the figure. On the contrary, with $w = 0.7$, good results are achieved.

A similar technique is used to propagate the code delay estimates. In the multi-correlator software receiver implemented, the code search space covers 10 chips. Assuming that the actual code delay lies exactly in the middle of the search space and a relative velocity between user and satellite of 800 m/s (worst case for a static user), then the code delay covers half of the full search space (towards right or towards left) in $10/R_c \cdot c/v \simeq 5.8$ s, where c is the speed of light. Therefore the sleep period should be lower than 5.8 s to assure that the signal can be declared present; otherwise, a proper prediction strategy, equivalent to the one described below, is exploited.

4.3.5 False alarm and detection probabilities

Another advantage of reducing the search space is represented by the improvement in false alarm and detection probabilities. The expression of single cell detection and false alarm probabilities, also known as single-trial probabilities, is well known [65]. The single cell false alarm probability, denoted P_{fa}, is equal to:

$$P_{\text{fa}}(\mathcal{T}) = \int_{\mathcal{T}}^{+\infty} f_{N_x}(\beta)\, \mathrm{d}\beta , \tag{4.65}$$

where $\mathcal{T}$ is a threshold and $f_{N_x}(\beta)$ is the distribution of the cells N_x under the null hypothesis (no signal).

Then, the global decision false alarm and detection probabilities (P_{FA} and P_{D} respectively), i.e. when analysing the whole reduced search space, can be derived, starting from (4.65). Assuming that the reduced search space is composed by N cells,

$$P_{\text{FA}}(\mathcal{T}) = 1 - (1 - P_{\text{fa}}(\mathcal{T}))^N , \tag{4.66}$$

and

$$P_{\mathrm{D}}(\mathcal{T}) = \int_{\mathcal{T}}^{+\infty} [1 - P_{\mathrm{fa}}(\beta)]^{N-1} f_A(\beta)\,\mathrm{d}\beta\,, \tag{4.67}$$

where $f_A(\beta)$ is the distribution of the cells under the signal presence hypothesis.

From (4.67) and (4.66) it is evident that by reducing the number of cells N the performance increases.

- The false alarm probability P_{FA} decreases as long as N decreases, thus reducing the probability of having an *invalid measure*.
- The detection probability P_{D} increases as long as as N decreases, thus improving the capability of the system to compute a valid estimate.

Figure 4.33 shows the P_{FA} trend versus the P_{fa} at cell level; the four curves are related to four different values of N. It can be observed that P_{FA} at search space level improves together with the reduction of the number of cells evaluated. However it has to be noticed that P_{FA} assumes higher values with respect the false alarm probability at cell level.

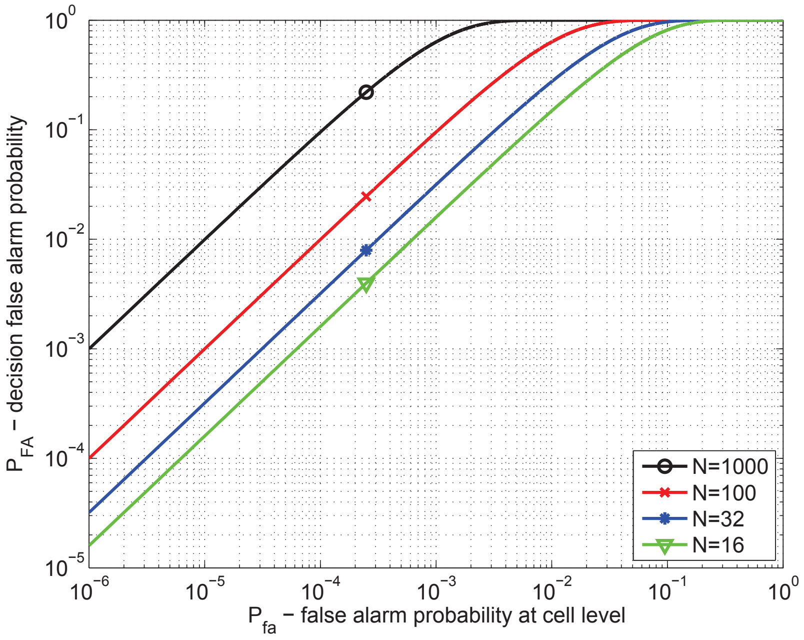


Figure 4.33: False alarm probability at search space level vs. false alarm probability at cell level for different numbers of cells N.