École polytechnique de Louvain

Study of GPS L1 C/A acquisition channels for a low-Earth orbit FPGA-SDR GPS receiver satellite

Author: **Pol Maistriaux**
Supervisors: **Jean-Didier Legat, Luc Vandendorpe**
Readers: **Jérôme Louveaux, Antoine Gauthier, Thomas François**

Academic year 2020–2021

Master [120] in Electrical Engineering

Acknowledgments

First, I would like to express my gratitude to my two supervisors Prof. Jean-Didier Legat and Prof. Luc Vandendorpe, for allowing me to work on this subject that has kept my interest over this year and for their support during this master thesis.

Second, thank you to the company Aerospacelab for trusting a master student with this task and for providing a good first experience in the world of work.

In particular, I would like to thank some employees of Aerospacelab : Antoine Gauthier for always being available to discuss the project, Summit Karki, Thomas Pairon and Thomas Francois which have answered my many questions and given me insightful advice.

I also would like to thank Prof. Jérôme Louveaux for agreeing to be a reader of my thesis.

Lastly, I would like to thank my family and my friends for their support during this year, with a special thank to Julien Pirard and Maxime Paquet.

Abstract

This master thesis provides a study of the GPS L1 C/A signal acquisition. The goal is to develop an acquisition channel for a GPS receiver which will be embedded on a low-Earth orbit satellite of the company Aerospacelab. A software-defined radio implemented on a field programmable gate array is used to optimize the performances and resource consumption of the design. To this end, a state-of-the-art of the GNSS technology is first established, with a detailed description of the GPS L1 C/A signal. Second, the theoretical structure of the acquisition channel of a GPS receiver is explained and the different acquisition algorithms and their specificities are discussed. Afterwards, the available hardware is presented. The actual VHDL implementation on the design tool Vivado from Xilinx is then discussed with a focus on the non-idealities inherent to the practical design of the processing chain. Finally, the testing of an acquisition channel is performed on captured GNSS data, allowing to validate the correct behavior of the system.

Contents

List of Abbreviations

ADC Analog-to-Digital Converter

AMBA Advanced Microcontroller Bus Architecture

AWGN Additive White Gaussian Noise

AXI Advanced eXtensible Interface

BPSK Binary Phase-Shift Keying

C/A Coarse / Acquisition

CDMA Code Division Multiple Access

CORDIC Coordinate Rotation Digital Computer

CPU Central Processing unit

DSP Digital Signal Processor

EIRP Equivalent Isotropic Radiated Power

EM Electromagnetic

FFT Fast Fourier Transform

FPGA Field Programmable Gate Array

GNSS Global Navigation Satellite System

GPP General Purpose Processor

GPS Global Positioning System

HDL Hardware Description Language

HOW Handover Word

IF	Intermediate Frequency
I/O	Input/Output
IP	Intellectual Property
IFFT	Inverse Fast Fourier Transform
LFSR	Linear-Feedback Shift Register
LPF	Low-Pass Filter
LSB	Least Significant Bit
LUT	Lookup Table
MAC	Multiply And Accumulate
MSB	Most Significant Bit
NAV	Navigation message
PCPS	Parallel Code Phase Search
PFS	Parallel Frequency Search
PLL	Phase-Locked Loop
PVT	Position Velocity Timing
RAM	Random-Access Memory
RFFE	Radio Frequency Front-End
SDR	Software-Defined Radio
SV ID	Space Vehicle Identifier
SVN	Space Vehicle Number
TLM	Telemetry word
VCO	Voltage-Controlled Oscillator
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits

Chapter 1

Introduction

The **Global Positioning System**, GPS, is a technology providing user with position, velocity and timing information, or PVT [1]. It has been developed in the United States and is available worldwide. This system is part of the of the **Global Navigation Satellite System**, GNSS, that includes multiple other standards, e.g. Galileo developed in the European Union, GLONASS in Russia, BeiDou in China. However, GPS remains nowadays the most widespread navigation system. It uses multiple satellites orbiting around the Earth at an altitude close to 20,000 kilometers. Thirty-one satellites are currently operating for GPS and twenty-four of them are required to obtain a full coverage of the Earth surface.

An important aspect of GNSS technologies is the communication unidirectionality: messages coming from GPS satellites contain all the information required at the receiver to compute the PVT. Therefore, after detection of a GPS satellite, receivers only extract the needed information from the signal, but no response signals are sent back.

Even though the GPS system launched in 1973, several improvements and evolutions have followed up to now, mostly at the receiver side with new and optimized algorithm for satellite acquisition as well as Assisted-GPS.

To extract its PVT, a GPS receiver must first acquire a satellite. All of them have a unique identifier and code, allowing to distinguish them. Due to the satellites motion, signal losses are frequent at the receiver, which must be able to identify and decode signals from other new satellites. A GPS receiver system is thus composed of an acquisition module responsible for this task, and a tracking module which keeps the lock on the identified satellites. Both are decomposed in multiple channels, allowing to acquire and track multiple satellites at the same time.

In the meantime, the development of Earth observation satellites for the monitoring of Earth surface has been accelerating. The company **Aerospacelab** [2] is specialized in the extraction of geospatial intelligence using their own satellites. Those data allow to improve decision-making in various domains, from ecosystems preservation using environmental indicators to territorial defence with strategic insights. It remains important to be able to localize those satellites orbiting in **low-Earth orbit**, or LEO. This is why the GPS technology is now being adapted to space vehicle tracking.

Figure 1.1: Representation of the applications performed by Aerospacelab Earth observation satellites [2]

In parallel, digital signal processing has become a standard for multiple applications, including GPS, with the development of **software-defined radio**, or SDR. **Field programmable gate arrays**, or FPGAs, provide high level of parallelization for computation, allowing to optimize the processing performances of the receivers.

This master thesis provides a study of the GPS L1 C/A signal acquisition. The objective is to develop an acquisition subsystem of a GPS receiver that will be embedded on a low-Earth orbit satellite of the company **Aerospacelab**. A FPGA-based SDR will be used for the implementation of the processing chain. This master thesis will be structured as follow.

In the first part of this work, the state-of-the-art concerning the GPS system will be established. The first chapter will present in details the GPS technology and more specifically the **GPS L1 C/A** signal. The effects of working on low-Earth orbit instead of the Earth surface will also be analyzed and quantified. The second chapter will cover the theoretical GPS acquisition processing chain with a presentation of the different algorithms described in the literature. In the third chapter, the hardware used for this design will be presented with the different possible configurations.

In the second part of this master thesis, the actual implementation developed with the software **Vivado** from the company **Xilinx** will be explained. First, a presentation of the acquisition channel architecture implementing the algorithm will be made. Next, the different modules designed for the acquisition system will be discussed with their working principle and specificities. A focus will be directed on the non-idealities inherent to the practical implementation. The last chapter will present the testing of the obtained design with captured GNSS data. The setup for this data capture will be detailed and the result of the acquisition channel will be shown and discussed.

Finally, a conclusion will summarize this work by going over the achieved milestones. A discussion over the possible improvements for the design will be made to conclude this master thesis.

Chapter 2

State-of-the-art of GPS technology

In the first part of this master thesis, a detailed introduction to the GPS technology is provided. An intuitive approach to the position determination is first given, based on a two dimensions case that is next extrapolated to three dimensions.

Next, the frequency band employed for the actual implementation, the **GPS L1 band**, will be detailed with the different signals transmitted over it. A quick introduction will be made on the **GPS Navigation message**, which contains the required data for correct PVT extraction and non-idealities compensation.

Afterwards, the focus will be on the license-free **GPS L1 C/A** signal which uses the C/A code to allow for identification, range estimation and data decoding. A description of the modulation scheme used and of the channel access method will be provided with an in-depth analysis of the C/A code properties and generation.

The last sections will present the **GNSS observables**. They are the available information on which the position computation is performed, involving multiple non-idealites caused by signal propagation through different atmosphere layers and from technical limitation of hardware. Lastly, the effects of the low-Earth orbit of the receiver will be studied, regarding the experienced Doppler shift and the available received power.

2.1 GNSS positioning

In order to determine their position, GPS receivers perform **trilateration**. This method allows to determine the position of an object based on the knowledge of its distance to known points, which is different from the more generally known triangulation. This last term refers to another technique of position determination in which the angles between multiple points are used. For trilateration, the required information is the distance between the transmitters and the receiver. Those distances are derived from the travelling time of the electromagnetic waves which are computed as the difference between the actual (or absolute) time of reception and the sending time of the signal. This last one can be extracted from the navigation message of the GNSS satellites. Multiplying the travelling time with the speed of the wave, here the speed of light c, gives the apparent distance to the satellite, R_p.

If we refer to a 2D problem, the knowledge of the distance to a point, ρ, constraints our possible positions to a circle around this point. The precise position can be found knowing the distances separating us from two reference points since the solutions become the intersections of two circles, as seen on the left of Figure 2.1. Two solutions are still possible but one of them can be discarded with an additional information : in the case of the GPS receiver, the device is on low-Earth orbit.

If an uncertainty δ on the distances to the reference points is considered, the solution becomes an hyperbola which foci are the reference points, as shown on the right of Figure 2.1. An hyperbola is the set of points such that for any point of the set, the difference of the distances to the foci is constant. This definition is mathematically written in Equation 2.1 with ρ_i being the true euclidean distances. The scenario where an uncertainty is included is realistic in our case since the wave is propagating at the speed of light. Thus, a small timing error can have a non-negligible impact on the computed distances which are therefore called **pseudoranges**.

$$\begin{cases} R_1 = \rho_1 + \delta \\ R_2 = \rho_2 + \delta \end{cases} \longrightarrow R_1 - R_2 = \rho_1 - \rho_2 = constant \tag{2.1}$$

To compensate for this error, the distance to a third reference point is used. The two final solutions are provided by the intersection of the set of hyperbolas obtained from the three reference points, see Figure 2.2. To go from this 2D to a 3D case, a fourth reference point needs to be added to cover the additional dimension. The above reasoning is still valid, however circles become spheres and hyperbolas become hyperboloids.

GPS receivers perform trilateration and the coordinates of the satellites are known from the navigation messages included in their communications. At any point in time, four satellites need to be acquired to get a position calculation. In the

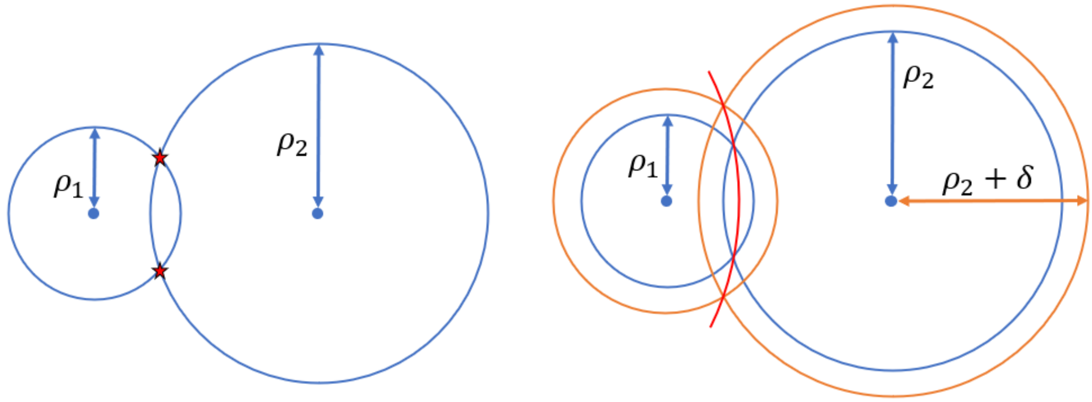

Figure 2.1: 2D trilateration problem. On the left, the red stars represent the possible solutions. On the right, an uncertainty δ has been added and the solution becomes an hyperbole. Only one branch has been represented, the other being less likely since it involves a much higher δ.

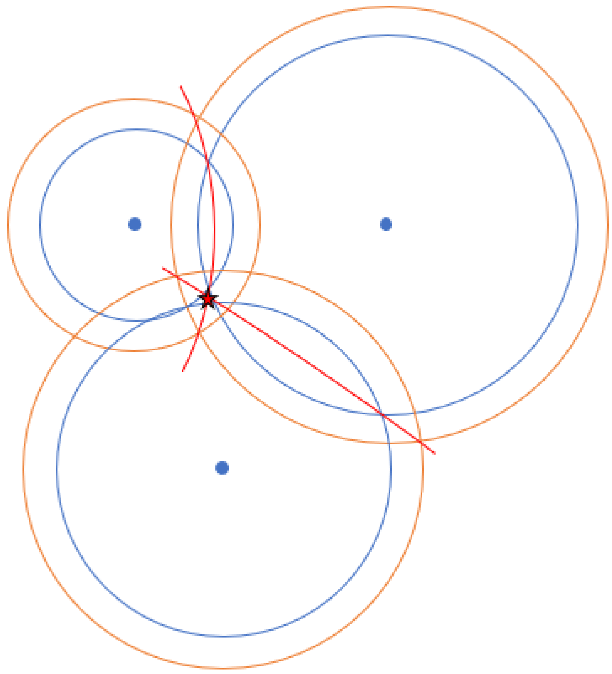

Figure 2.2: 2D trilateration with 3 reference points to compensate for the distance error. The two solutions are the intersections of the hyperbolas. Only one has been represented to ease readability.

end, a system of equations is solved using different available methods. From the two remaining solutions at the end, one is on low-Earth orbit. To obtain an accurate pseudorange, the clocks of the different devices must be precisely synchronized. GNSS satellites therefore embed expensive atomic clocks with very low drift. Additionally, the communications of the satellites include a correction error on its clock to reach highest precision.

2.2 GPS L1 Band

In this section, we are going to introduce the frequency spectrum used by the GPS technology. GPS satellite transmitters operate in the L band which covers frequencies from 1 to 2 GHz. The spectrum is divided in multiple bands used by different systems. This work will employ the GPS L1 band centered at a carrier frequency f_0 of 1575.42 MHz. This band is one of the most used for nowadays application [3].

A GNSS signal includes three different components. First, the carrier is the sinusoidal waveform transmitted at the central frequency in the L band. Second, ranging codes are bit sequences used to extract the travel time of the signal and to acquire satellites, i.e. synchronize with them. Most GNSS technologies use **code division multiple access**, or CDMA. It is a channel access method where multiple transmitters can use the same communication channel at the same time and over the same frequency band without collisions. To this end **pseudorandom noise codes**, or PRN codes, are employed. Those codes have a high auto-correlation but almost null correlation with a shifted version of themself or with other codes. This allow to distinguish transmitting satellites and to synchronize the receiver. Finally, the navigation message provides information on the state of the satellite e.g., its position, its clock offset and its almanac [4].

Four signals are transmitted over the GPS L1 band : the **Coarse/Acquisition code**, or C/A code, knwon as civilian code, the **L1C** signal designed for interoperability with Galileo E1, the Precision/Secure (P/Y) and the Military (M) code, both reserved by cryptographic techniques to military and authorized civilian users.

In this work, we will only be interested in the C/A code since it is designed for civilian use and is still one of the most widespread codes. It uses PRN codes of the family of Gold codes. They consist of 1023 bits, named chips, transmitted at 1.023 Mchips/s and are therefore repeated every millisecond. As explained above, the PRN codes of different satellites are orthogonal, meaning low cross-correlations. Due to the wave speed, a chip of the C/A code is equivalent to 293m and this

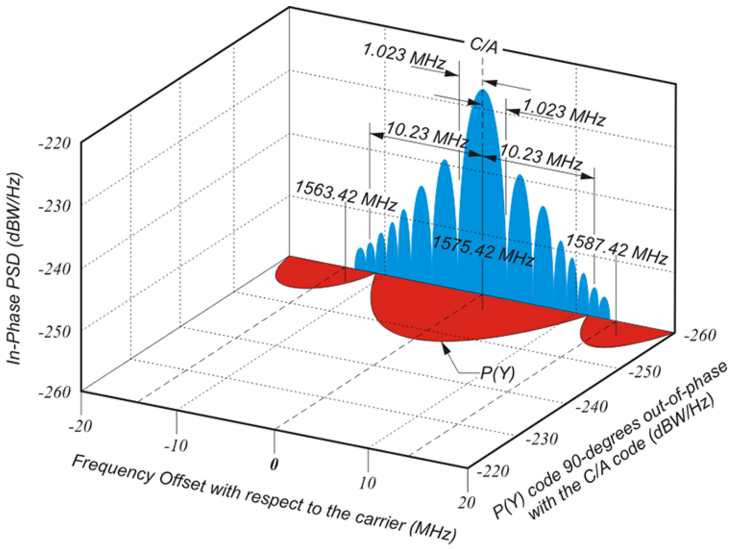

Figure 2.3: GPS L1 band spectrum representation from *GPS for Land Surveyors* [4], with the C/A and P/Y codes shown

accuracy can be extended with additional methods. To differentiate satellites, each of them has a unique and permanent space vehicle number (SVN) and all operating satellites have an additional space vehicule identifier (SV ID).

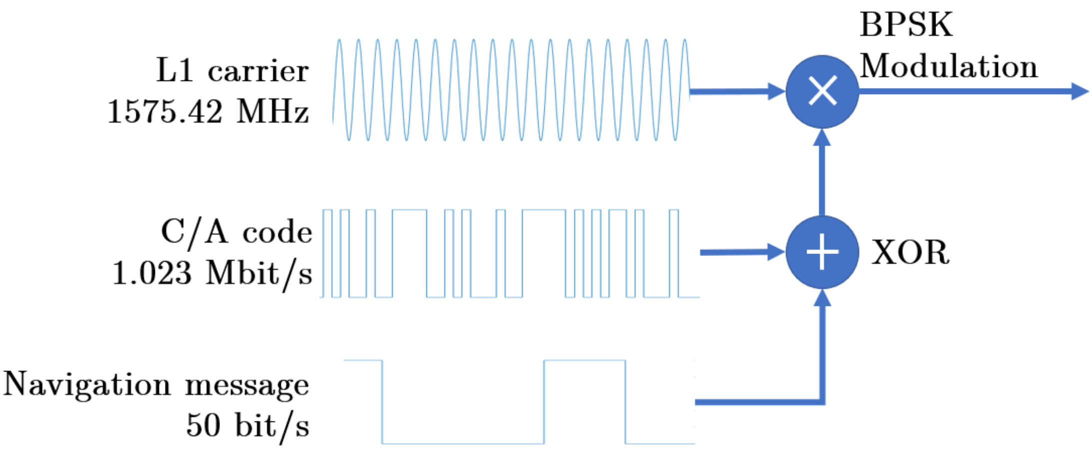

Figure 2.4: GPS L1 C/A modulation signal generation

To construct the transmitted C/A signal, a navigation message which rate is 50 bits per second, or bps, is XORed with the PRN code, i.e. exclusive-or logic function.

The result is then modulated by the carrier at 1575.42 MHz using **binary phase shift keying**, or BPSK modulation. The process is summarized on Figure 2.4. The following sections will present each part of the signal involved.

2.3 GPS Navigation message

As mentioned earlier, a navigation message (NAV) is transmitted over the carrier frequency. It contains multiple information necessary for PVT computation and additional information to obtain higher accuracy. Each navigation message is composed of 25 frames for a total payload of 37.5 Kbits. At 50 bps, the complete message takes 12.5 minutes to transmit. A frame with a size of 1500 bits include 5 subframes. Each subframe (300 bits) provides information on different aspects, e.g. clock correction, ephemeris (i.e. position over time), almanac, satellite health status, using 10 words of 30 bits. Moreover, a subframe always starts with a telemetry word, TLM, which allows for synchronization thanks to a fixed preamble. It is followed up by a handover word, HOW, providing information about the time in the GPS week and the index of the subframe. A general representation of the navigation message is shown on Figure 2.5 and a summary of the information about the length and duration of its subparts is available on Figure 2.6.

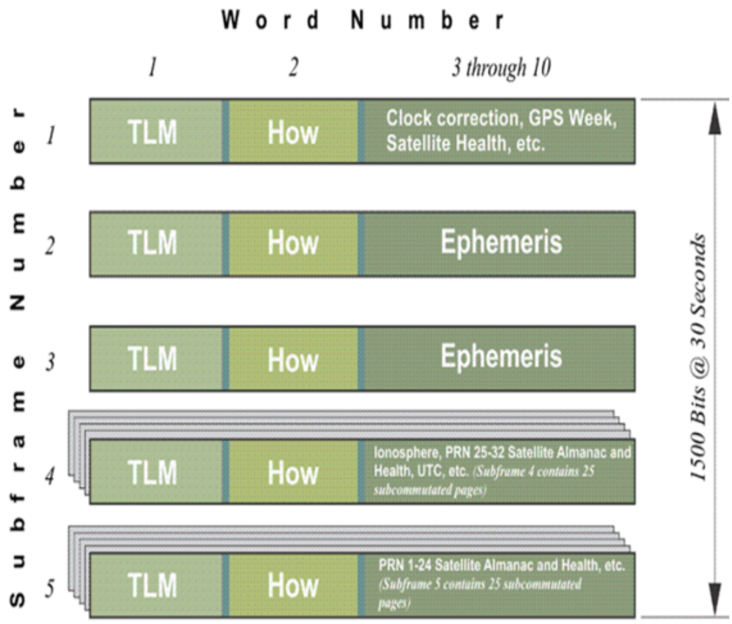

Figure 2.5: Representation of the GPS navigation message for a complete frame from *GPS for Land Surveyors* [4]

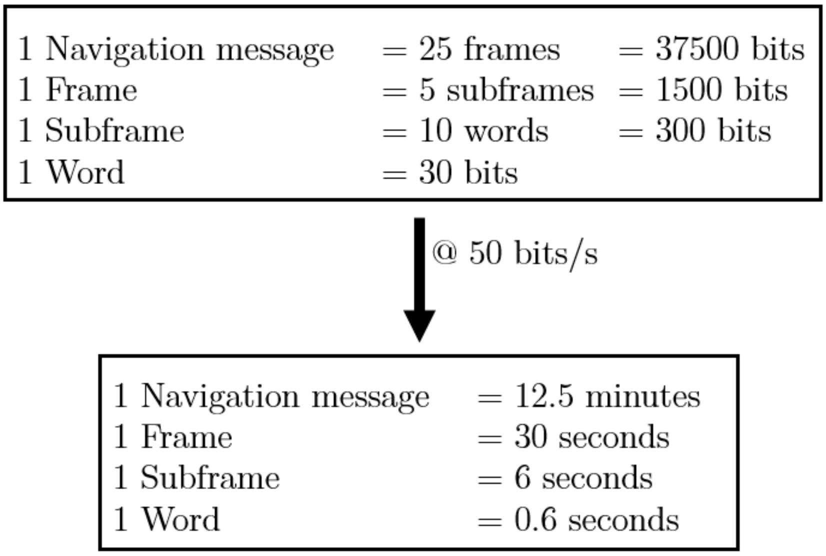

Figure 2.6: Size and duration of the different parts and sub-parts of the GPS navigation message

2.4 GPS L1 C/A modulation

The GPS L1 C/A signal is modulated using BPSK, as represented on Figure 2.7. It uses two phases which are separated by 180°. On a constellation diagram, the signal at the transmitter can be plotted with only an in-phase component, as in Figure 2.8. The information-bearing signal to be transmitted is $D(t) * x(t)$ and is purely real at the transmitter. It is the combination of the PRN code $x(t)$ and the navigation message $D(t)$. The output of the transmitter can be written as in Equation 2.2, with f_0 being the carrier frequency of 1.575 GHz.

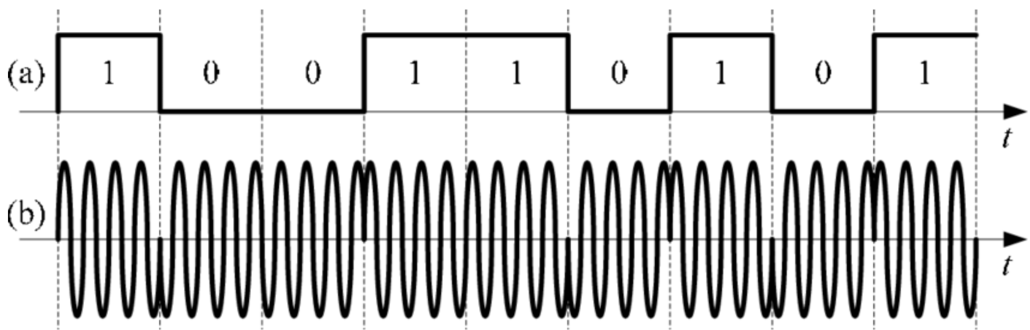

Figure 2.7: BPSK modulation representation [5]

$$b_{out}(t) = D(t) * x(t) * cos(2\pi f_0 t) \tag{2.2}$$

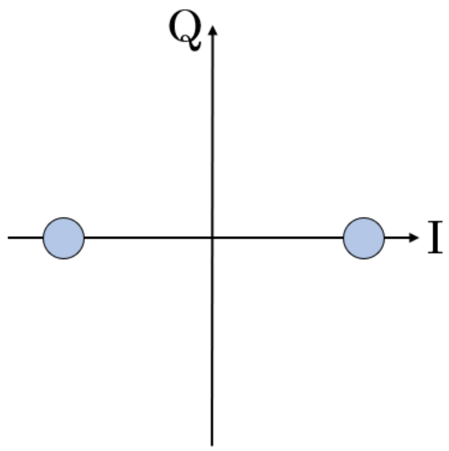

Figure 2.8: BPSK constellation at transceiver

Of course during transmission a phase shift will be experienced and thus the constellation will turn around the origin, as seen in Figure 2.9. The captured signal, however, is always real. Neglecting the amplitude of the received signal, its expression is given in Equation 2.3.

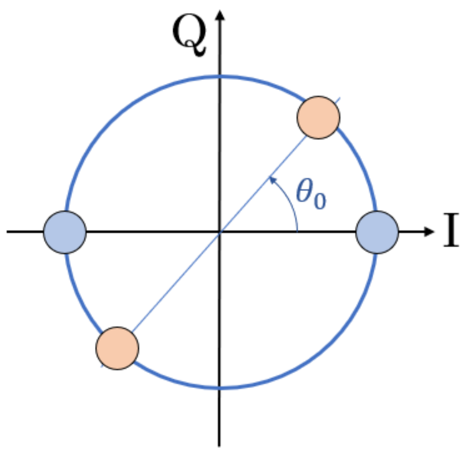

Figure 2.9: BPSK constellation at receiver

$$\begin{aligned} r_{in}(t) &= \Re\left\{D(t-\tau) * x(t-\tau) * e^{2\pi(f_0+f_d)t+\theta_0}\right\} \\ &= D(t-\tau) * x(t-\tau) * cos(2\pi(f_0+f_d)t+\theta_0) \end{aligned} \tag{2.3}$$

- τ is the delay due to signal propagation.
- $D(t)$ is the navigation message.
- $x(t)$ is PRN code.
- θ_0 is the phase shift accumulated during signal propagation.
- f_d is the Doppler shift experienced by the signal, explained later on.

2.5 Coarse/Acquisition code analysis

The coarse/acquisition PRN code, which will be referred to as the PRN code or sequence, is a Gold code with 1023 chips period. Gold codes, named after Robert Gold, are sequences with the property of high auto-correlation values when aligned and small bounded values otherwise. This provides the ability to detect when sequences are aligned at the receiver and satellite to obtain a value of the time of transmission and to compute the pseudorange. Moreover, Gold codes also have small bounded cross-correlations and each satellite uses a different PRN code. Therefore, satellites can be identified based on their PRN codes. In the following section, a budget of those cross and auto-correlations is provided. This will help increase the **signal-to-noise ratio**, abbreviated SNR, at the expense of an higher bandwidth since the bit rate is multiplied by 1023, a bit of information being replaced by a complete PRN sequence.

2.5.1 PRN code property

The auto-correlations of the 37 first PRN codes have been computed and are shown on Figure 2.10. The bottom graph is in log scale and zoomed to properly estimate the correlation gain. It is the ratio between the highest correlation value, also named correlation peak, and the highest secondary peak.

It can be seen on the figure that the auto-correlation when a lag is introduced does not exceed a certain value. In fact, the auto-correlation of the Gold codes can take only a finite set of values which are listed in Table 2.1. From this we extract a correlation gain of 23.94 dB.

Auto-correlation [chips]	-65	-1	63	1023

Table 2.1: Auto-correlation set of values for C/A PRN codes

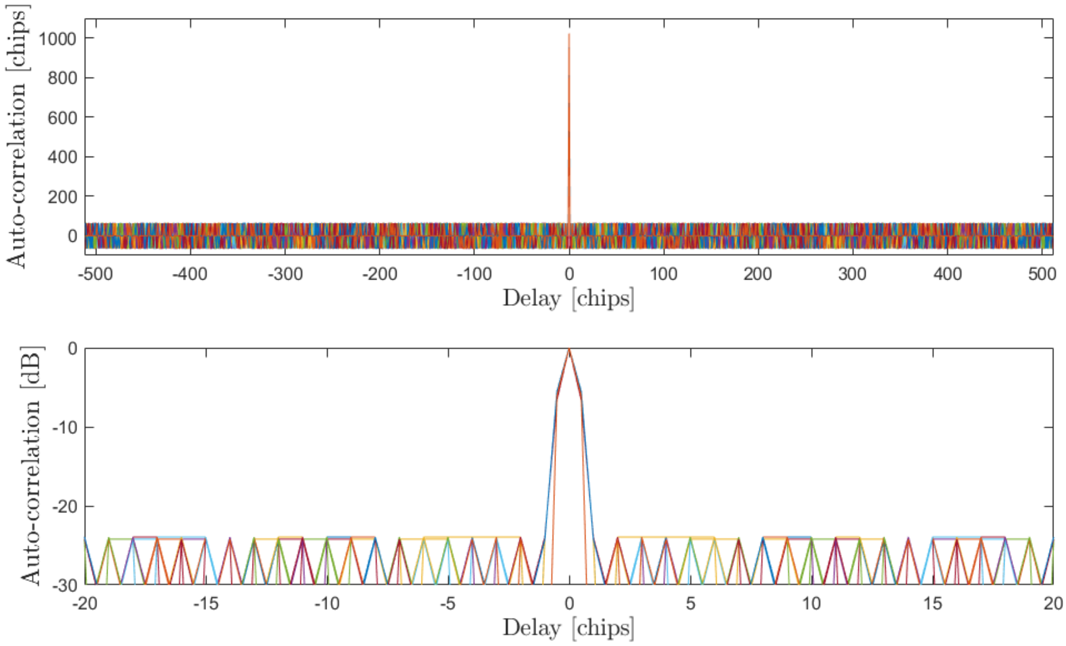

Figure 2.10: Auto-correlations of the 37 first C/A PRN codes

Cross-correlations between 3 satellites are depicted on Figure 2.11. The same finite set of values is obtained for the cross-correlation of different satellites but with the absence of a peak when the sequences are aligned.

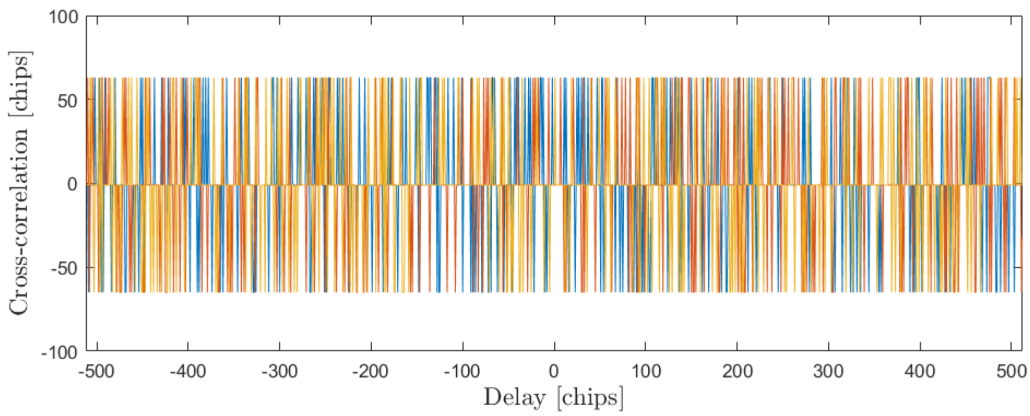

Figure 2.11: Cross-correlation of 3 different C/A PRN codes

2.5.2 PRN code generation algorithm

To generate the PRN codes, two 10 stages linear-feedback shift registers, or LFSR, are used [6]. They are 10 bits registers which are sequentially shifted to the right. When shifting, the bit furthest to the right becomes the output and the bit furthest to the left becomes the result of a linear function. This last part creates the feedback system. This function is represented by a polynomial in which the exponent represent the bit index in the register (going from 1 to 10). In our case, the XOR function is used as linear function. Finally, a LFSR needs an initial state called seed. Here, two LFSRs, **A** and **B**, are used with the polynomials expressed in Equations 2.4.

$$\begin{aligned} A(x) &= x^{10} + x^3 + 1 && \text{with register seed 1111111111} \\ B(x) &= x^{10} + x^9 + x^8 + x^6 + x^3 + x^2 + 1 && \text{with register seed 1111111111} \end{aligned} \tag{2.4}$$

In the polynomials above, the "+1" terms refer to the input bits on which the feedback is performed. To obtain the PRN code, two more steps are needed. A XOR between two bits of **B** is operated. The indexes of those two bits depends on the satellite for which the PRN code is generated, and are referred to as the **code phase selector**. Finally, its result is then XORed with the output bit of **A**. The generation scheme is described on Figure 2.12.

2.6 GNSS observables

The computation of the PVT is based on the knowledge of the pseudoranges to the satellites, R_p. They are computed by multiplying the calculated travelling time δt with the speed of light c as expressed in Equation 2.5. The terms $t_s(T_1)$ and $t_r(T_2)$ correspond respectively to the time of signal transmission measured by the satellite and the time of signal reception measured by the receiver.

$$R_p = c * [t_r(T_2) - t_s(T_1)] \tag{2.5}$$

As explained in Section 2.1, clock uncertainties introduce an error on the observed distances. Those pseudoranges are hence not equivalent to the geometric distances between receivers and satellites. The expression of the geometric distance as function of the 3D Cartesian coordinates of the receiver (x_r, y_r, z_r) and of the satellite (x_s, y_s, z_s) is given in Equation 2.6.

$$\rho = \sqrt{(x_r - x_s)^2 + (y_r - y_s)^2 + (z_r - z_s)^2} \tag{2.6}$$

Moreover, multiple other factors impact the pseudorange. Taking them into account, a more complete range equation is proposed in Equation 2.7 [3].

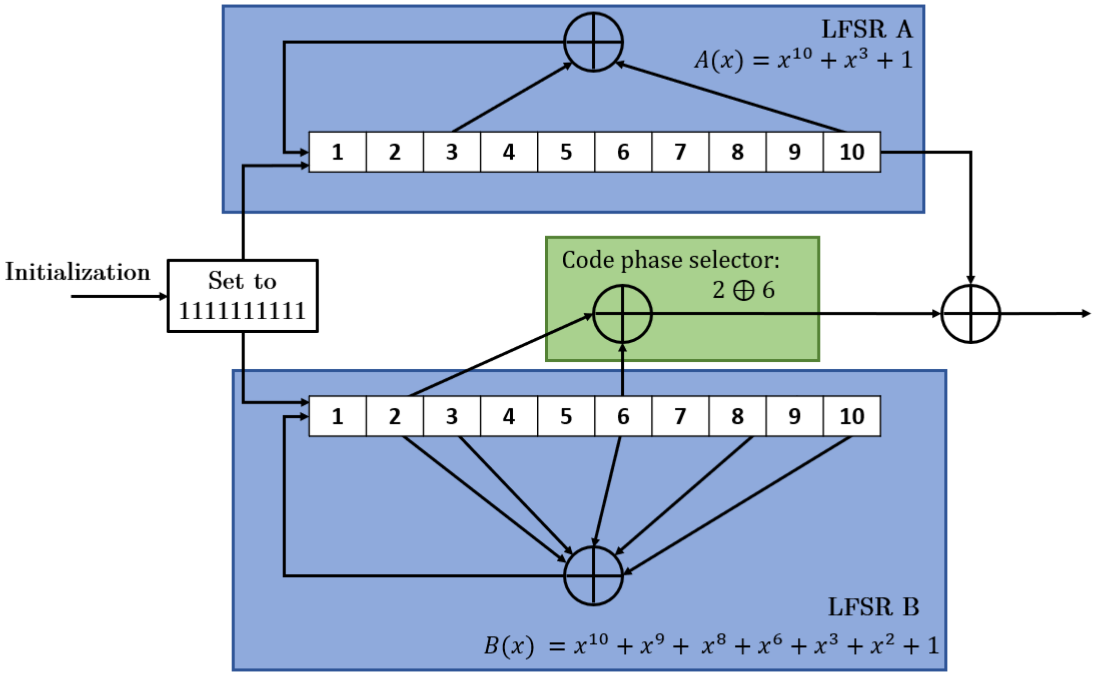

Figure 2.12: PRN code generation algorithm representation for satellite with code phase selector : $6 \oplus 2$

$$R_p = \rho + c * (dt_r - dt_s) + T + \alpha_f STEC + K_{P,r} + K_{P,s} + M_P + \epsilon_P \tag{2.7}$$

The terms of the equations above are defined as :

- ρ : the geometric distance between the receiver and the satellite.
- dt_r : the receiver clock offset due to internal clock drift.
- dt_s : the GNSS satellite clock offset due internal clock drift.
- T : the tropospheric delay.
- $\alpha_f STEC$: the ionospheric delay.
- $K_{P,r}$ and $K_{P,s}$: the instrumental delay at the receiver and the GNSS satellite.
- M_P : the multipath effect.
- ϵ_P : the noise at the receiver.

The troposhpere is the lowest Earth atmosphere layer, not exceeding 20 km in height. Its effects on the GNSS signals appears as an extra delay in the measurement of the signal, depending on the temperature, pressure, humidity and on the antennas location. The troposphere is a non-dispersive media for electromagnetic waves up to 15GHz, i.e., the tropospheric effects are not frequency dependent for the GNSS signals. Hence, the carrier phase and code are affected by the same delay. Even if the troposphere is at a lower altitude than the LEO, the signal might pass through it when the LEO and GPS satellites face different parts of the Earth surface, see Section 2.7.2.

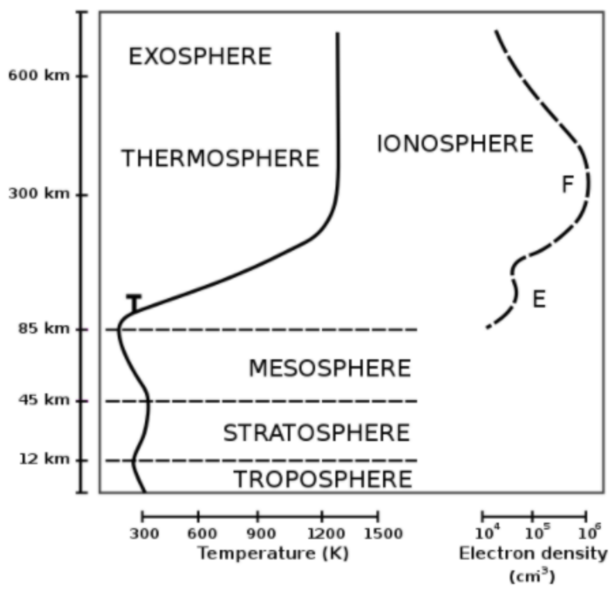

Figure 2.13: Representation of the atmosphere layers and relationship with the ionosphere

Another part of the atmosphere layers coming into play is the ionosphere, the combination of the upper layers starting at approximately 50 km. As the name suggests, it is the ionized part of the upper atmosphere. The propagation speed of the GNSS electromagnetic signals in the ionosphere depends on its electron density, which differs between day and night due to respectively sun radiations and the recombination process [3]. This media is dispersive in opposition to the troposhere, the term α_f being frequency dependent. A representation of the ionosphere and of the atmosphere layers is given on Figure 2.13.

In addition, instrumental delays due to inherent imprecision in devices are added. Multipath effects are important on the ground but will not be as impacting when working on low-Earth orbit. Each term from Equation 2.7 does not have an equivalent impact, and not all of them will need to be compensated.

Finally, a **Doppler shift** is experienced by the signal. It is the frequency variation of the received signal due to the velocities of the transmitter and receiver relative to one another. The receiver must estimate the Doppler shift of the received signal from a satellite to be able to properly decode its message. GNSS satellites travel at a high speed with respect to the Earth-centered reference system. This speed of around 3.9 km/s is determined by the height of their orbit. For a ground receiver, the Doppler shift experienced does not exceed 5kHz. But due to LEO space vehicle higher speed, the Doppler shift in our application must be reevaluated. A qualitative study of this effect will be performed in Section 2.7.1.

2.7 Effects of low-earth orbit GPS

The context of a low-Earth orbit (LEO) satellite requires a study of the effects it will induce on the GPS receiver. The main change is the Doppler shift due to the difference in the receiver velocity when going from the ground to LEO. Indeed, to maintain their orbit LEO satellites need to travel at higher speed. In addition, GNSS satellites radiation pattern ensures the highest received power on the Earth surface. An analysis will hence be conducted to quantify the available received power at the receiver. For the equations that follows, the notations below will be used :

- f_0 is the transmission frequency, here 1.575 GHz.
- $R_E = 6370km$ is the Earth radius considering a perfect sphere.
- $R_R = 300km$ is the altitude of the GPS receiver in our study case.
- $R_S = 20200km$ is the altitude of GPS satellites.
- v_s is the velocity of the GPS satellite.
- $v_{s,r}$ is the velocity of the GPS satellite in the receiver direction.
- v_r is the velocity of the receiver.
- $\mu = GM = 398600.5km^3/s^{-2}$ is the standard gravitational parameter equal to the product of the gravitational constant G and the mass M of the body (in our case the Earth).
- $c = 299.79 * 10^6 m/s$ is the speed of light.

2.7.1 Doppler shift range quantification

The goal in this section is to quantify the maximum Doppler shift that can be experienced at the LEO GPS receiver. A representation of the problem for the worst case scenario inducing this maximum Doppler shift is depicted on Figure 2.14. The Doppler shift being dependent on the objects velocity towards each other, the worst case scenario happens when the tangential trajectory of the receiver is in line with the GPS satellite. First, we need to calculate the velocity of the satellites based on their orbit height as expressed in Equation 2.8 where r is the orbit height and v the resulting speed. This equation assumes a circular orbit.

$$v = \sqrt{\frac{\mu}{r}} \tag{2.8}$$

The orbital eccentricity measures the deviation from a circle of an object's orbit. Earth orbital eccentricity is 0.017 making the assumption relevant. The different velocities are computed in Equations 2.9, 2.10 and 2.11.

$$v_r = \sqrt{\frac{\mu}{R_E + R_R}} = 7.73 km/s \tag{2.9}$$

$$v_s = \sqrt{\frac{\mu}{R_S + R_R}} = 3.87 km/s \tag{2.10}$$

$$\begin{aligned} v_{s,r} = & \quad v_s * cos(90^\circ - \alpha) \\ = & \qquad v_s * \frac{R_E + R_R}{R_E + R_S} = 0.97 km/s \end{aligned} \tag{2.11}$$

The Doppler shift can then be derived based on equations of the relativistic Doppler effect to take into account the high speed of objects and the electromagnetic (EM) nature of the transmitted waves. This expression and its result are provided in Equation 2.12.

$$\begin{aligned} f_D = & \left[\sqrt{\frac{1+\beta}{1-\beta}} - 1\right] * f_0 \quad \text{with} \quad \beta = \frac{v_{s,r} + v_r}{c} \\ = & 45.722 kHz \end{aligned} \tag{2.12}$$

With the same reasoning for opposite directions, it can finally be proven that the Doppler shift range for LEO satellites goes from $-46kHz$ to $46kHz$. For an Earth surface context, the Doppler shift has an usual value below $\pm 5kHz$. Thus, there is almost a factor ten that will impact the device acquisition of new satellites.

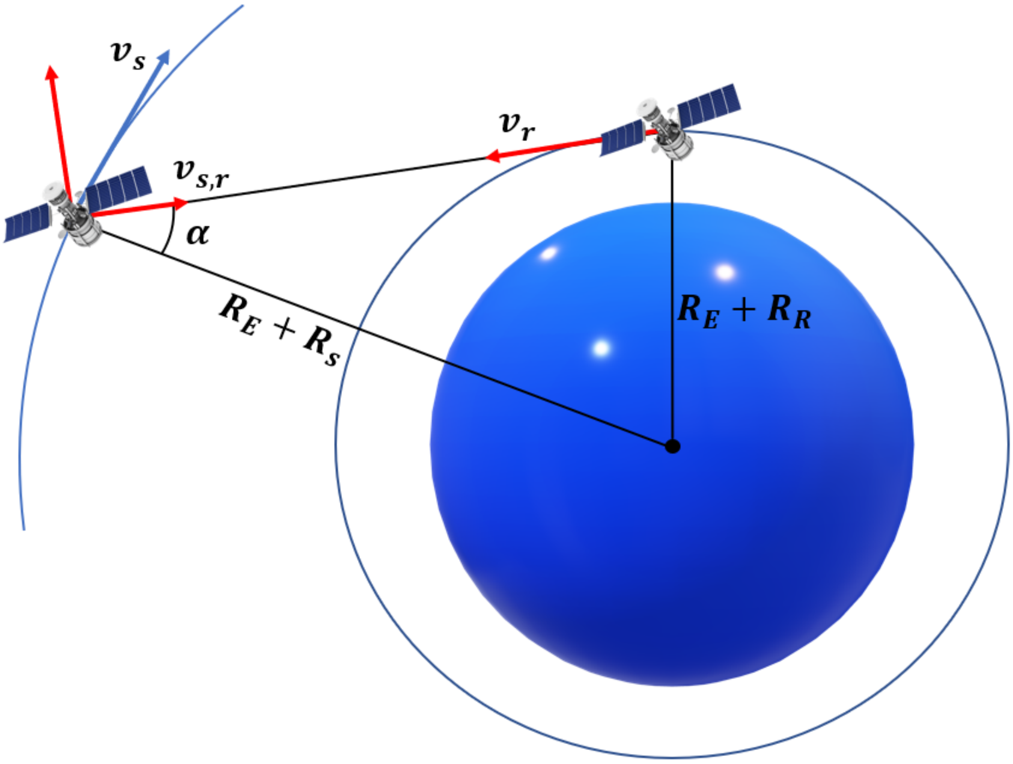

Figure 2.14: Doppler shift worst case scenario representation for LEO satellites

2.7.2 Received power quantification

Since GPS is developed by the U.S government, there is a lack of information regarding the specifications of GPS satellites. The interface specification **IS-GPS-200** [7] defines a minimum received power in the worst case for the different frequency band. Variations in path loss due to longer observed distances are compensated by a higher antenna gain. This guarantees a minimum received power of -158.5 dB at the Earth surface. This value is measured at the output of a $3dBi$ linearly polarized antenna. This section will discuss the impact of low-Earth orbit on the received power. Since the U.S. government does not share all the specifications of GPS satellites, researchers have reverse engineered some of those parameters which will be used for this development, see [8].

First, the received power P_R at the receiver can be computed based on the equivalent isotropic radiated power, abbreviated $EIRP$, and the receiver antenna gain G_r. Losses need to be taken into account, e.g. the free path loss L_s, the atmospheric loss L_{atm} and the polarization mismatch loss L_p. Equation 2.13 is derived from [8].

$$P_r[dB] = EIRP + G_r - L_s - L_{atm} - L_p \tag{2.13}$$

In this equation, the $EIRP$ is computed as the input power of the transmitter antenna multiplied by the antenna gain in the receiver direction. The input power is in general considered to be 27 Watts [9]. The antenna gain depends on the elevation angle α which is the angle seen at the GPS satellite between the center of the Earth and the receiver direction. For a ground receiver, the maximum elevation angle is equal to 13.8° while for a LEO receiver it increases to 14.5°. Based on tables from [9] and [8], the minimum observed antenna gain at this angle is 12 dB.

The received antenna gain is considered 3 dBi for a linearly polarized antenna. However, this gain is cancelled by the polarization mismatch loss equal to 3 dB since the signal is polarized circularly.

Next, the **free-path loss** accounts for the signal power reduction due to the distance travelled by the wave. It depends on the wavelength λ of the signal and therefore on its frequency. Equation 2.14 expresses it in decibels, where R is the distance separating transmitter and receiver. The worst case scenario for R, i.e. the longest distance, is depicted on Figure 2.15.

$$L_s = 20 * log\left(\frac{\lambda}{4\pi R}\right) = 20 * log\left(\frac{c}{4\pi R * f_0}\right) \tag{2.14}$$

From this worst case scenario, the distance R can be derived as done in Equation 2.15, which gives a value of 27800 km. With our carrier frequency of 1.575 GHz, this gives a free path loss of -185.2 dB.

$$R = \sqrt{(R_E + R_S)^2 - R_E^2} + \sqrt{(R_E + R_R)^2 - R_E^2} \tag{2.15}$$

Finally, the atmospheric loss is the result of gas absorption , rain, clouds, ... Its value is generally considered to be below 0.5 dB of loss [9]. The receiver power can then be quantified based on Equation 2.7.2 and is equal to -159.5 dBW which is only one decibel lower than the standard for IS-GPS-200. The effect of LEO on the received power is therefore minor and, in addition, multipaths and obstructions have not been considered here while they would affect more a ground receiver.

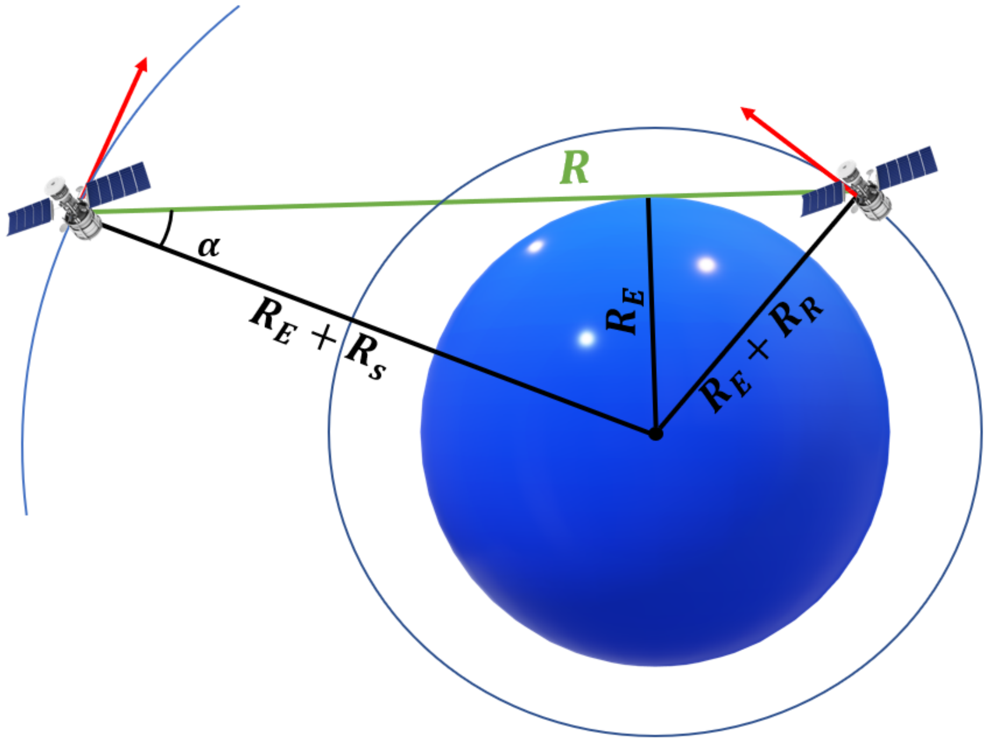

Figure 2.15: Representation of the worst case scenario regarding the distance separating the GPS and LEO satellites

Chapter 3

State-of-the-art of GPS acquisition channels

It has already been introduced that GPS receivers need to synchronize with GPS satellites to be able to extract all possible information, e.g. navigation message, pseudorange. Two tasks are performed for GPS satellite synchronization : **acquisition and tracking**. This master thesis is centered on the first one. The acquisition system is necessary to get a lock on a satellite. It identifies several unknowns required to properly decode the received message and obtain the pseudorange.

In this chapter, a state-of-the-art of GPS acquisition channels will be conducted. Whereas the previous part introduced the needed concepts for the GPS L1 C/A signal, this part will focus on the structure of the acquisition processing chain of a GPS receiver.

First, the acquisition objectives will be introduced through the GPS signals unknowns it tries to identify. From there, the acquisition search space representing the possible combinations of those unknowns will be presented.

Afterwards, the signal path along the processing chain will be followed to provide a theoretical analysis of the involved operations. Therefore, the **radio-frequency front end** will be discussed even though it is common to the tracking and acquisition systems. Next, the demodulation of the signal considering the Doppler effect, named **Doppler wipeoff**, will be discussed. To detect the satellite signal, the receiver then performs a correlation which will be mathematically analysed.

After this theoretical presentation of the acquisition operation, three algorithms implementing its principle will be described and then compared based on their performances and resources consumption.

In the last sections, methods to improve the **signal-to-noise ratio** will be explained with the advantages and drawbacks they involve. The decision logic for acquisition approval will be intuitively and mathematically described. An overview of the interest for **FPGA-based SDR receiver** will be discussed in the end.

3.1 GPS signal unknowns

As explained earlier, a GPS receiver needs to know its distance to 4 satellites. For this, it must be able to decode their signal. However, this received signal has multiple unknowns that must be extracted. First, the receiver does not know the GPS satellite it receives signal from. It can be identified by finding the PRN code used for CDMA encoding. Also, the delay between the PRN code of the received signal and the one generated in the receiver, caused by the signal propagation, must be determined. It will be referred to as the **code delay** and it is a mandatory information to synchronise the two codes and extract the navigation message. Moreover, a Doppler shift affects the signal and, as explained previously in Section 2.7, it will impact the decoding of the message even more for a low-Earth orbit positioning system. If the Doppler shift is not properly compensated during demodulation, the residual low frequency component will interfere with the PRN code and the navigation message .

Acquisition base principle is to correlate the incoming signal with a generated PRN code, referred to as PRN replica, by assuming the value of the unknowns. This correlation output value should rise if the assumed values are close to the real ones. Indeed, PRN codes have only high auto-correlations value when they are aligned with themself. Therefore, for a reachable satellite, a demodulation compensating the real Doppler shift followed by a correlation considering the correct code delay should result in a high correlation peak.

3.2 Acquisition search space

Since each GPS satellite signal that reaches the receiver experienced a specific Doppler shift and code delay, the receiver tries to determine those two parameters for every known satellite. This is performed serially, meaning that each satellite PRN code is tested independently for every possible values of the two unknowns. To acquire different satellites at the same time, the acquisition processing chain is duplicated. Each instantiation is called an **acquisition channel**.

The combinations of Doppler shifts and code delays for a specific satellite can be represented on a 2D grid, named acquisition search space [10], [11]. Figure 3.1 represents it, each lines intersection being a combination, referred to as an acquisition trial. An acquisition cell defines the range of values (on x and y axis) between the acquisition trials.

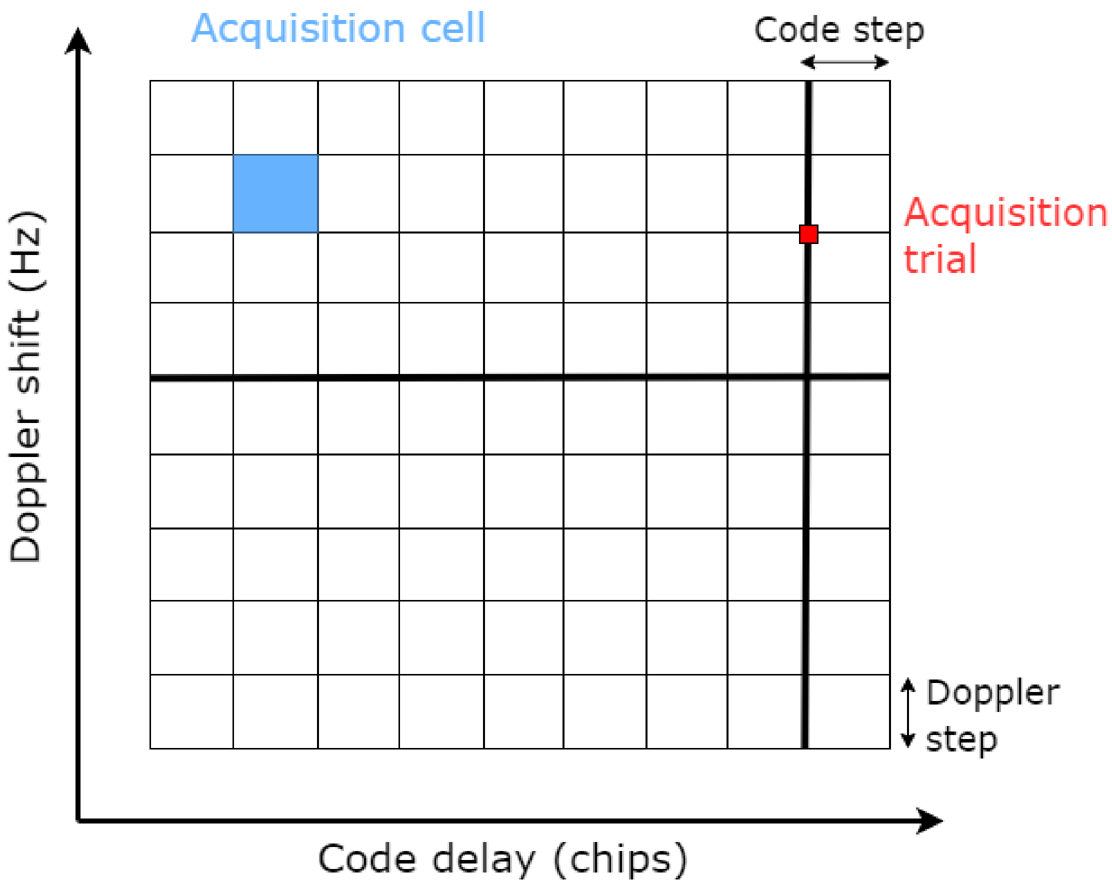

Figure 3.1: 2D representation of the acquisition search space, with the red and blue square representing respectively an acquisition trial and an acquisition cell

The search space needs multiple parameters to be computed. The **Doppler range** $f_{d,max}$ is the maximum Doppler shift considered for the acquisition. The **Doppler step** $f_{d,step}$ and **code step** τ_{step} are respectively the frequency and code delay variations between two acquisition trials. The Doppler range is below 5 kHz on the Earth surface but has been calculated at 46 kHz on low-Earth orbit due to the space vehicle high velocities as seen in Section 2.7. The Doppler and code step define the resolutions achievable. Indeed, with a small value of those parameters, the unknowns determination becomes highly accurate and further GPS operations, e.g. tracking, will be easier. High values risk to lower the acquisition result, decision-making on satellite presence becoming harder. The advantage is the algorithm execution time: lower number of acquisition trials need to be tested to go over the whole search space [12].

Now that the search space is characterized, a metric must be derived in order to compare the result of the different acquisition trials. This is the goal of the receiver processing chain, which is explained in the following sections.

3.3 Radio frequency front end

When the signal reaches the receiver it is captured by an antenna. The radio frequency front end, abbreviated RF front end or RFFE, includes all the building blocks that will process the signal at its received incoming frequency and downconvert it for further processing to an intermediate frequency, IF, and mathematically expressed f_{IF}. Receivers employing IFs are referred to as superheterodyne receivers [10]. The RF front end uses filters and mixers to process the signal, and **analog-to-digital converters**, or ADCs, for digitization. In this thesis, the **MAX2769C** chip [13], represented on Figure 3.2, will be studied. This RF front end produces in-phase and quadrature samples, also named **I and Q** samples, both encoded on 2 bits. An additional configuration allows to use only the I samples on 3 bits, but it will not be discussed here.

$$r_{cos}(t) = cos(2\pi(f_0 - f_{IF})t + \theta_r) \tag{3.1}$$

$$r_{sin}(t) = sin(2\pi(f_0 - f_{IF})t + \theta_r) \tag{3.2}$$

To perform the downconversion, the RFFE multiplies the input signal with a sine and cosine functions. Those two signals, expressed in Equations A.2 and A.3, are produced by a phase-locked loop, or PLL, and a voltage-controlled oscillator, or VCO, at a frequency $f_0 - f_{IF}$. Then, adding a **low pass filter**, or LPF, the desired part of the spectrum can be isolated. Otherwise, a high frequency content would interfere with the digitization and the processing. With the input signal r_{in} expressed in Equation 2.3, we can provide the expressions of the resulting I and Q components at the output of the LPF. They are given in Equations 3.3 and 3.4, where h_I and h_Q are respectively the I and Q outputs, h_{LPF} is the transfer function of the LPF, and θ_r is the phase shift of the RFFE local oscillator. The mathematical development of those equations can be found in Appendix A.1.1.

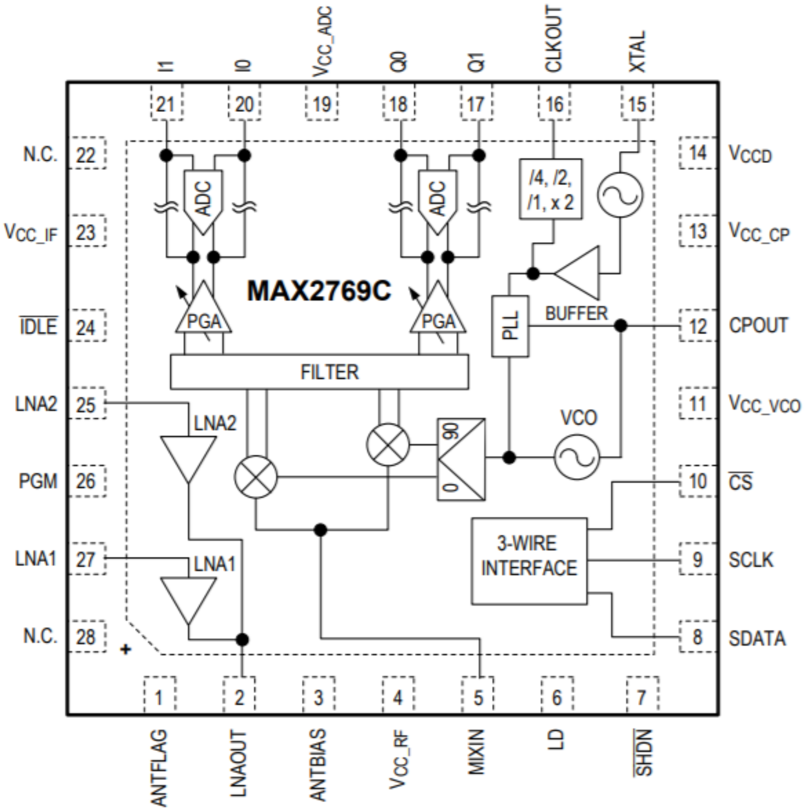

Figure 3.2: Schematic of the MAX2769C [13]

$$
\begin{aligned}
h_I(t) &= h_{LPF}(t) \circledast [r_{in}(t) * r_{cos}(t)] \\
&= D(t-\tau) * x(t-\tau) * \frac{1}{2} * cos(2\pi(f_{IF}+f_d)t + \theta_0 - \theta_r)
\end{aligned}
\tag{3.3}
$$

$$
\begin{aligned}
h_Q(t) &= h_{LPF}(t) \circledast [r_{in}(t) * r_{sin}(t)] \\
&= -D(t-\tau) * x(t-\tau) * \frac{1}{2} * sin(2\pi(f_{IF}+f_d)t + \theta_0 - \theta_r)
\end{aligned}
\tag{3.4}
$$

Finally, those signals are sampled by the ADCs. To keep the developments simple, the signals will still be analysed in continuous time.

3.4 Doppler wipeoff

Since the signal has been affected by a Doppler shift f_d, it must be considered in ordrer to perform the downversion from the intermediate frequency to baseband.

This downconversion, performed considering a trial Doppler shift value $\hat{f}_d$, is referred as **Doppler wipeoff**.

To perform the Doppler wipeoff, an IQ demodulation need to be performed. If only the in-phase or quadrature is used, information loss appears. Indeed, the input signal and the local oscillator used for downconversion may not be in phase. Without taking any care, it would lead to an unknown residual phase shift in baseband. The information contained in the signal would decrease the closer the residual phase shift θ is to $\pi/2$. A representation of this issue on an IQ diagram is depicted on Figure 3.3. When the signal is fully in quadrature, no information can be retrieved from the in-phase component, and inversely. By using an both the I and Q components for demodulation, all the information of the signal is maintained.

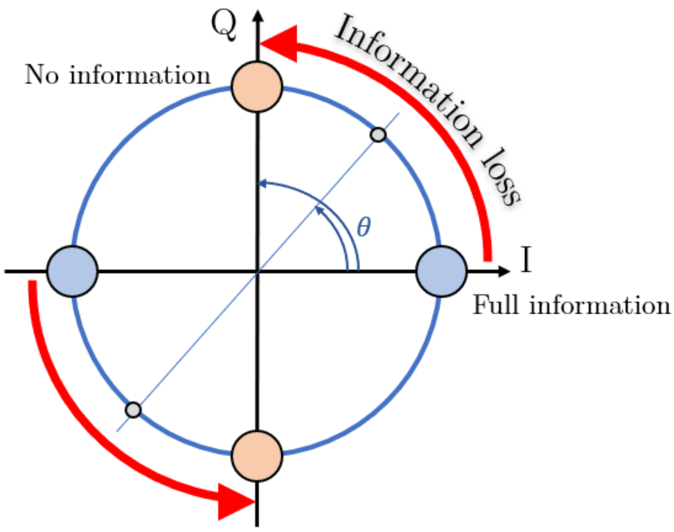

Figure 3.3: IQ diagram of the signal and impact of only working with the in-phase or quadrature component

The IQ demodulation scheme is shown on Figure 3.4. The underlying mathematical development is provided in Appendix A.1.2, derived from the output of the RF front end, see Equations 3.3 and 3.4. The final results are given in Equations 3.5 and 3.6, where $h_{I,out}$ and $h_{Q,out}$ are the signals at the end of the of the Doppler wipeoff. Annotations are used and need to be set:

- $\hat{f}_d$ is the trial Doppler shift used for Doppler wipeoff.
- $\hat{\theta}$ is the phase shift of the local oscillator used by the Doppler wipeoff.

- $\Delta f_d = f_d - \hat{f}_d$ is the remaining error at the output and thus the residual frequency that will be propagated to the next steps of the acquisition channel.
- $\Delta\theta = \theta - \hat{\theta}$ is the phase shift between the samples and the local oscillator used by the Doppler wipeoff.

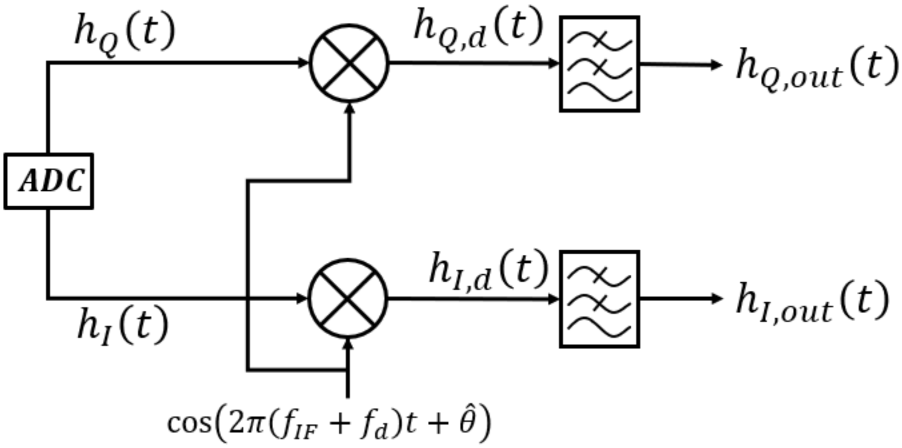

Figure 3.4: IQ demodulation while using I and Q samples

$$\begin{aligned} h_{I,out}(t) &= h_{LPF}(t) \circledast \Big[h_I(t) * cos(2\pi(f_{IF} + \hat{f}_d)t + \hat{\theta})\Big] \\ &= D(t-\tau) * x(t-\tau) * \frac{1}{2} * cos(2\pi\Delta f_d t + \Delta\theta) \end{aligned} \tag{3.5}$$

$$\begin{aligned} h_{Q,out}(t) &= h_{LPF}(t) \circledast \Big[h_Q(t) * cos(2\pi(f_{IF} + \hat{f}_d)t + \hat{\theta})\Big] \\ &= -D(t-\tau) * x(t-\tau) * \frac{1}{2} * sin(2\pi\Delta f_d t + \Delta\theta) \end{aligned} \tag{3.6}$$

From those equations, the expression of the complex term h_{out} can be written as done in Equation 3.7. It can be observed that the modulus of h_{out} is independent on the phase shift. We thus have a **non-coherent demodulation** since the information about the phase is lost. The modulus operator will therefore be applied at the end of the acquisition algorithm to suppress residual phase shift dependency.

$$\begin{aligned} h_{out}(t) &= h_{I,out} + jh_{Q,out} \\ &= D(t-\tau) * x(t-\tau) * \frac{1}{2} * e^{-j2\pi(\Delta f_d t + \Delta\theta)} \\ |h_{out}(t)| &= \left| D(t-\tau) * x(t-\tau) * \frac{1}{2} * e^{j2\pi\Delta f_d t} \right| \end{aligned} \tag{3.7}$$

3.5 Mathematical correlation expression

With the signal now in baseband, the expression of the correlation can be derived. In the end, we hence obtain a 3D representation of the search space with the resulting value of the correlation for each acquisition trial.

A **correlation** between two sequences consists in first their multiplication element-wise. The result is then integrated (or accumulated) over their duration. The entire correlation function is derived by circularly shifting one of the sequence. In our case, a progressively shifted PRN replica is multiplied with the Doppler wipedoff signal and then accumulated over the duration of the PRN code, T_{CO}. From the input complex signal given in Equation 3.7, the theoretical correlation expression can be derived, as shown in Equations 3.8 [11], where $\hat{\tau}$ is the shift applied to the PRN replica.

$$\begin{aligned} h_{corr}(\hat{\tau}, \hat{f}_d, \tau, f_d) &= \int_0^{T_{CO}} x(t-\hat{\tau}) * h_{out}(t)\, dt \\ &= e^{-j\Delta\theta} * \int_0^{T_{CO}} D(t-\tau) * x(t-\hat{\tau}) * x(t-\tau) * e^{-j2\pi\Delta f_d t}\, dt \end{aligned} \tag{3.8}$$

To suppress the residual phase shift $\Delta\theta$, the modulus operator is applied on the correlation result, as shown in Equations 3.9 and 3.10. For simplicity, the notation $\Delta\tau = \tau - \hat{\tau}$ is introduced and the navigation message $D(t)$ has also been removed. The navigation message has a rate of 50 bps, meaning that it has a one half probability to change every 20 PRN periods. This is thus not a decisive term at the moment and it will be discussed later on in this work. Finally, the 3D result of this expression for the entire search space is shown on Figure 3.5, where it has been normalized by the highest correlation value, i.e. the correlation peak, and expressed in dB.

$$|h_{corr}(\Delta\tau, \Delta f_d)| = \left| \int_0^{T_{CO}} x(t-\hat{\tau}) * x(t-\tau) * e^{-j2\pi\Delta f_d t}\, dt \right| \tag{3.9}$$

$$h_{corr}(\Delta\tau, \Delta f_d)[dB] = 10 * log \left(\frac{|h_{corr}(\Delta\tau, \Delta f_d)|^2}{T_{CO}^2} \right) \tag{3.10}$$

To see the impact of the Doppler and code step, we can derive the expression of the correlation assuming that one unknown has been determined, i.e. either $\Delta\tau$ or Δf_d is close to 0 [14]. Below, T_{chip} is the duration of 1 chip of the PRN code: $T_{CO}/1023$. First, we assume that the code delay has been correctly identified. A PRN code is a sequence of "+1" and "−1", meaning that when multiplied with an aligned replica

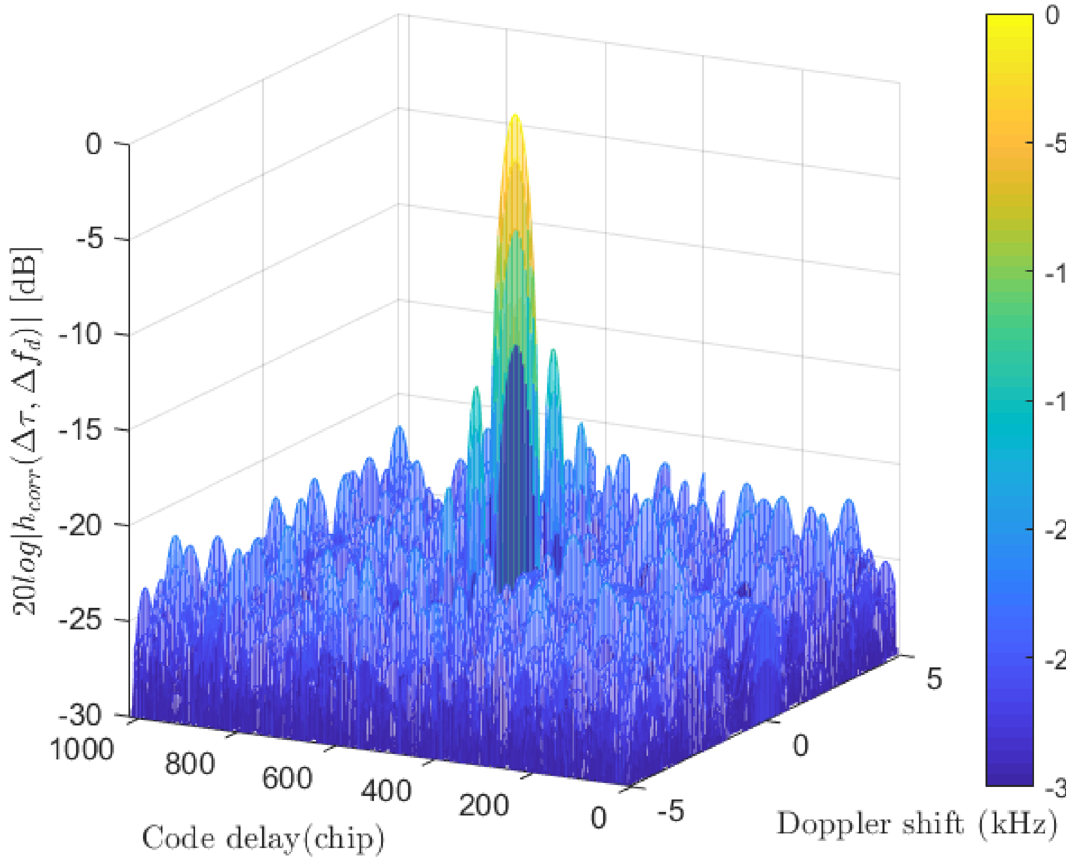

Figure 3.5: $|h_{corr}(\Delta\tau, \Delta f_d)|^2$ expressed in dB : correlation result over the entire search space

of itself, the result is constant. Therefore, Equations 3.11 can be written. The full development is available in Appendix A.2. The plot of this function is represented on Figure 3.6 in log scale.

$$\begin{aligned} |h_{corr}(\Delta\tau = 0, \Delta f_d)|^2 &= \left| \int_0^{T_{CO}} x^2(t) * e^{-j2\pi\Delta f_d t} \, dt \right|^2 \\ &= T_{CO}^2 * \left| \frac{sin(\pi\Delta f_d T_{CO})}{\pi\Delta f_d T_{CO}} \right|^2 \end{aligned} \tag{3.11}$$

From the expression of the correlation in this particular case, the effect of the Doppler step, denoted $f_{d,step}$, can be quantified. We can assume that the worst case scenario for the correlation peak position happens when it is located precisely in

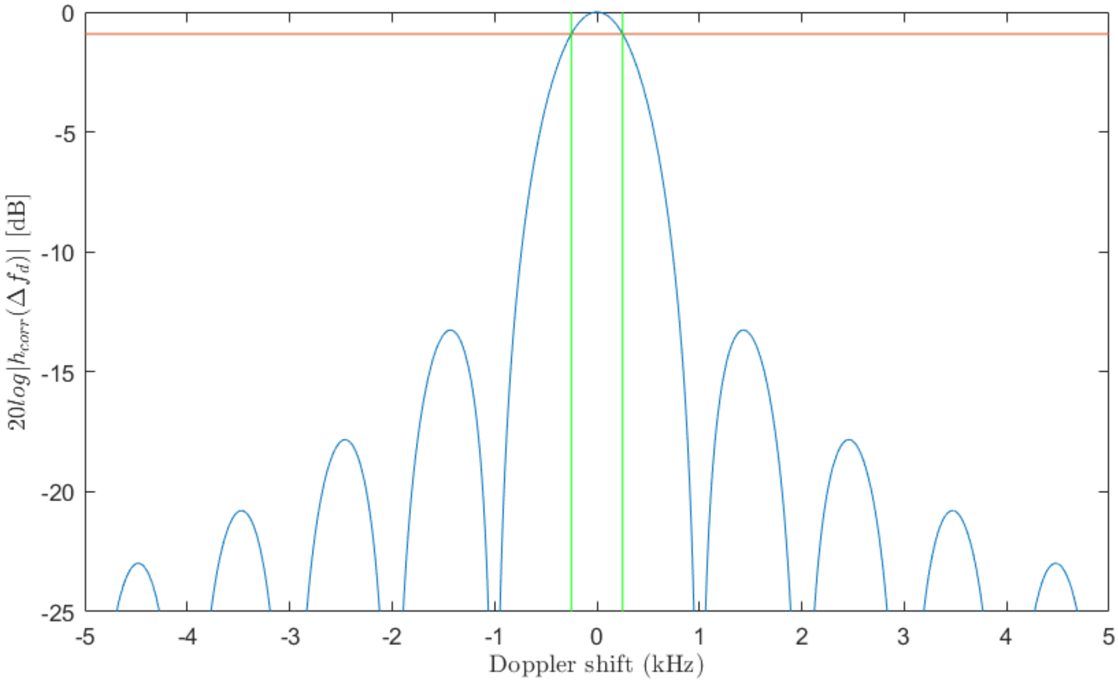

Figure 3.6: $20log|h_{corr}(\Delta f_d)|$ when $\Delta\tau = 0$. Green lines represent the position of the Doppler shift trials resulting in the worst case scenario for the correlation when $f_{d,step} = 500$ Hz. Their intersection with the orange line shows the observed correlation result.

the middle of an acquisition cell. The peak is then positioned at ${}^{f_{d,step}}/_{2}$ of the two closest Doppler shift trials, as represented on Figure 3.6. The standard value of the Doppler step equal to 500 Hz has been used for the plot and for the quantification of Equations 3.12, where Δf_d has been replaced by half $f_{d,step}$. This choice of parameter implies a loss of 0.9 dB on the correlation.

$$20 * log10\left(\left|\frac{2sin({}^{\pi f_{d,step}T_{CO}}/_{2})}{\pi f_{d,step}T_{CO}}\right|\right) = -0.91dB \tag{3.12}$$

Now we assume that the Doppler shift was accurately compensated. The resulting expressions are derived in Equation 3.13 and the correlation function is plotted on Figure 3.7.

$$\begin{aligned} |h_{corr}(\Delta\tau, \Delta f_d = 0)|^2 &= \left|\int_0^{T_{CO}} x(t) * x(t - \Delta\tau)\, dt\right|^2 \\ &= \begin{cases} \left|T_{CO} * \left(1 - \frac{|\Delta\tau|}{T_{chip}}\right)\right|^2, & \text{if } |\Delta\tau| \leq T_{chip} \\ 0, & \text{else} \end{cases} \end{aligned} \tag{3.13}$$

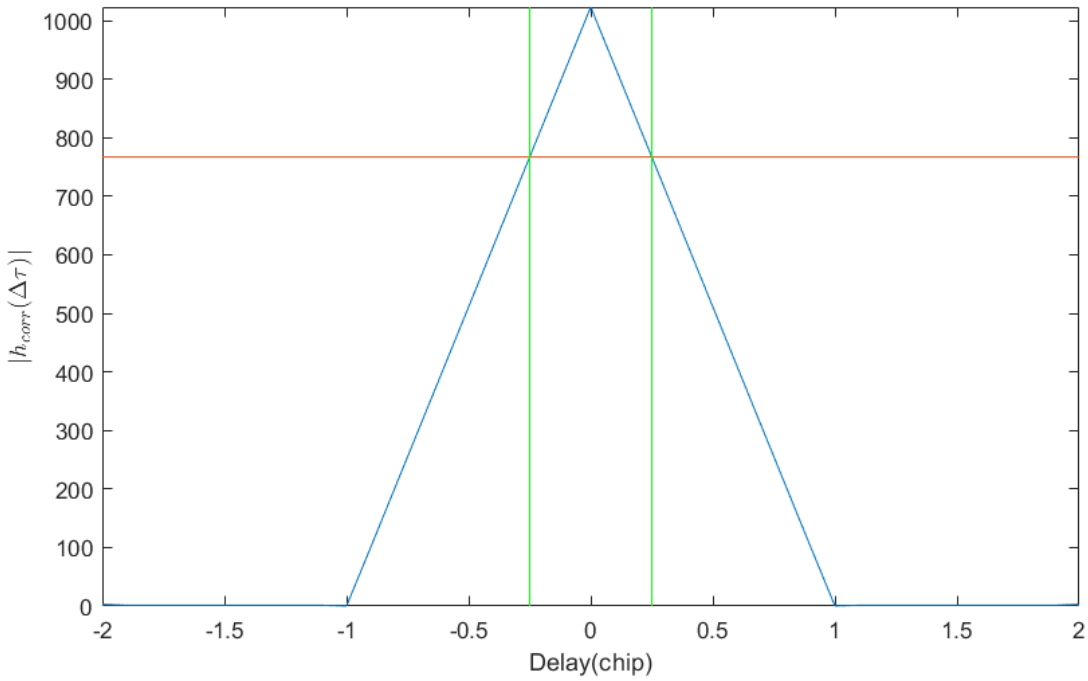

Figure 3.7: $|h_{corr}(\Delta\tau)|$ when $\Delta f_d = 0$. Green lines represent the position of the code delay trials resulting in the worst case scenario for the correlation when $\tau_{d,step} = {}^{T_{chip}}/_{2}$. Their intersection with the orange line shows the observed correlation result.

The same reasoning as before can be applied to observe the impact of the code step. Equation 3.14 quantifies the impact on the correlation when replacing $\Delta\tau$ by its worst case value, half τ_{step}. For a standard code step value of half a chip, it results in a loss of 2.5 dB.

$$\begin{gathered}\left(1 - \frac{|\tau_{step}|}{2T_{chip}}\right) = 0.75 \\ 20 * log(0.75) = -2.5dB\end{gathered} \tag{3.14}$$

3.6 Algorithms

Now that the acquisition principle, through the correlation, has been properly mathematically described, the algorithms implementing it can be analysed. First, the most basic algorithm will be presented, which straightforwardly applies the concept of the previous section. The two others build on properties of the correlation and on the spectrum of the signal to increase the overall acquisition performances. Their analysis will include a description of their processing chain as well as a quantification of their performances and resource consumption.

3.6.1 Serial search acquisition

Serial search acquisition analyses every acquisition trials sequentially. The Doppler wipeoff is performed with a trial Doppler shift value $\hat{f}_d$, based on the IQ demodulation, resulting in the two signals $h_{I,out}$ and $h_{Q,out}$. The correlation for a trial code delay $\hat{\tau}$ is calculated by shifting the PRN replica by this delay and performing a multiply-and-accumulate operation, or MAC. It is done separately for $h_{I,out}$ and $h_{Q,out}$. As explained before in Section 3.4, the modulus operator is then used to obtain phase shift independence. When finished, the trial code delay is changed to obtain the full correlation function for the specific Doppler shift trial, which is afterwards modified too [9]. The overall serial search acquisition working principle is depicted on Figure 3.8.

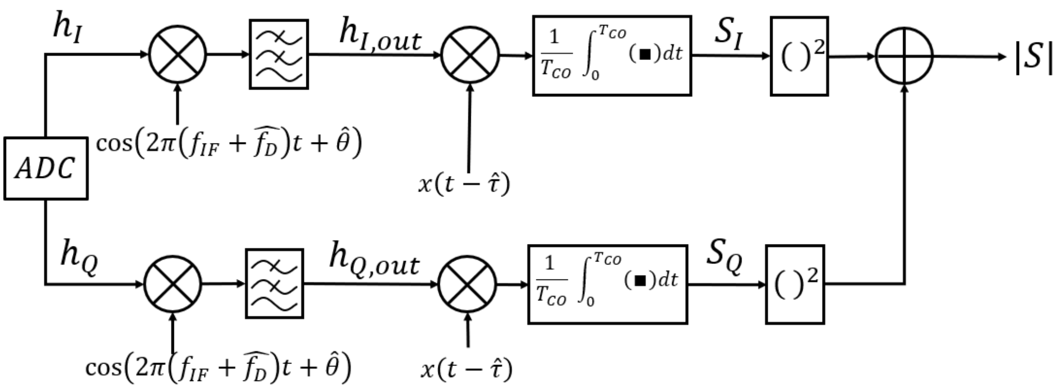

Figure 3.8: Serial search acquisition algorithm

Based on Equations 3.5 and 3.6 of the Doppler wipeoff outputs from Section 3.4, we can derive the expressions for this algorithm and prove their equivalence to the ones obtained with the mathematical analysis of Section 3.2. First the two correlations outputs S_I and S_Q, respectively from the in-phase and quadrature components, are derived in Equations 3.15 and 3.16.

$$
\begin{aligned}
S_I &= \int_0^{T_{CO}} h_{I,out}(t) * x(t-\hat{\tau})\, dt \\
&= \int_0^{T_{CO}} D(t-\tau) * x(t-\tau) * \frac{1}{2} * cos(2\pi\Delta f_d t + \Delta\theta) * x(t-\hat{\tau})\, dt
\end{aligned}
\tag{3.15}
$$

$$
\begin{aligned}
S_Q &= \int_0^{T_{CO}} h_{Q,out}(t) * x(t-\hat{\tau})\, dt \\
&= -\int_0^{T_{CO}} D(t-\tau) * x(t-\tau) * \frac{1}{2} * sin(2\pi\Delta f_d t + \Delta\theta) * x(t-\hat{\tau})\, dt
\end{aligned}
\tag{3.16}
$$

The modulus operator is then applied on the complex signal formed by S_I and S_Q. This is equivalent to summing the square of those signals, as expressed in Equations 3.17.

$$\begin{aligned}|S|^2 =& |S_I + j * S_Q|^2 = S_I^2 + S_Q^2 \\ =& \frac{1}{4} * \left| \int_0^{T_{CO}} D(t-\tau) * x(t-\tau) * x(t-\hat{\tau}) * e^{-j2\pi \Delta f_d t}\, dt \right|^2\end{aligned} \tag{3.17}$$

One can observe that this result is similar to the one obtained in Equation 3.9. Serial search acquisition is highly limited by its computation time. If we denote $S_{p,Chip}$ as the number of samples per chip of the PRN code, $1023 * S_{p,Chip}$ samples are needed for an acquisition trial. Due to the discrete nature of the signal, the minimal τ_{step} is equal to the sample duration, inversely proportional to $S_{p,Chip}$. For a maximum considered Doppler shift $f_{d,max}$, the number of acquisition trials and samples needed are given by Equations 3.18 and 3.19.

$$N_{trials} = \frac{2 * f_{D,max}}{f_{D,step}} * 1023 * S_{pChip} \tag{3.18}$$

$$\begin{aligned}N_{samples} =& N_{trials} * (1023 * S_{pChip}) \\ =& \frac{2 * f_{D,max}}{f_{D,step}} * (1023 * S_{pChip})^2\end{aligned} \tag{3.19}$$

The two next algorithms build on the serial search principle to increase the performances both in terms of computation time and of resources. They rely on analysing at the same time all possible values of one unknown, i.e. Doppler shift or code delay, for a specific value of the other. It drastically reduces the number of computations even though the implementations become more complex.

3.6.2 Parallel frequency search acquisition

The purpose of **parallel frequency search**, or PFS, acquisition is to analyse simultaneously all Doppler shift trials for a specific code delay trial. It therefore only needs to iterate over one dimension of the search space [12]. This algorithm does not use Doppler wipeoff. Even though a PRN replica is used, the goal is to analyse the spectrum of the signal. The algorithm processing chain is presented on Figure 3.9. The output of the RF front end is directly multiplied with the PRN replica. The signal at the input of the **fast Fourier transform**, abbreviated FFT, is described in Equation 3.20:

$$h_{FFT,in}(t) = D(t-\tau) * x(t-\tau) * x(t-\hat{\tau}) * cos(2\pi(f_{IF} + f_d)t + \theta) \tag{3.20}$$

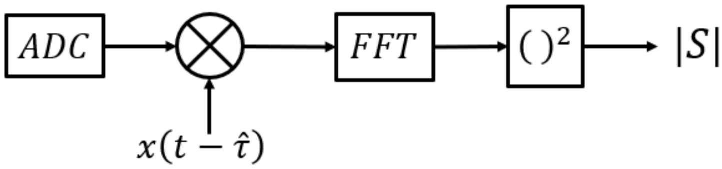

Figure 3.9: Parallel frequency search acquisition algorithm

If we assume the two PRN sequences are aligned, the equation could be simplified to only the cosine term. By applying the Fourier transform, only the cosine spectrum should appear in the case assumed. This spectrum contains a delta positioned at $(f_{IF} + f_d)$. By knowing the IF value, we can easily deduce the Doppler shift. Lastly, the output of the FFT is squared for sign independence.

With this algorithm, the Doppler shift step is equal to the FFT resolution, denoted Δf_{FFT}, which is directly proportional to N_{FFT}, the FFT length. This length can be calculated based on the sampling rate and on the Doppler step targeted, as shown in Equation 3.21 [12].

$$N_{FFT} = \frac{f_s}{\Delta f_{FFT}} = \frac{f_s}{f_{d,step}} \tag{3.21}$$

Finally, the number of acquisition trials and of samples needed can be recomputed as done for serial search algorithm, the resulting expressions are given in Equations 3.22 and 3.23.

$$N_{trials} = 1023 * S_{p,Chip} \tag{3.22}$$

$$\begin{aligned} N_{samples} =& N_{trials} * N_{FFT} \\ =& \frac{f_s}{f_{d,step}} * (1023 * S_{p,Chip}) \end{aligned} \tag{3.23}$$

3.6.3 Parallel code phase search acquisition

Usually, the number of Doppler shift trials is far smaller than the number of code delays, i.e. a standard GPS receiver has around 20 Doppler shift trials and at least 2046 code delay trials [9]. The last algorithm, **parallel code phase search** acquisition, or PCPS, is using properties of the correlation to process all possible code delays at once for a specific Doppler shift trial. The implementation is more complex than the two previous algorithms but requires fewer iterations.

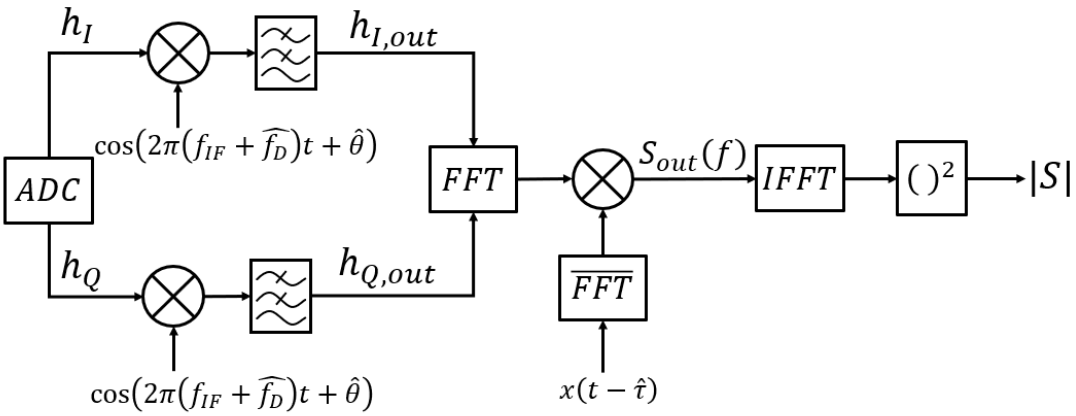

Figure 3.10: Parallel code phase search acquisition algorithm

First, Doppler wipeoff is performed, followed by the correlation. From a mathematical point of view, a correlation in time domain between two sequences is equivalent to the multiplication of those sequences in the frequency domain. To use this principle, the FFT of the received demodulated signal and of the PRN replica are first computed. The result of the first one is then multiplied with the complex conjugate of the second. Finally, the inverse Fourier transform is used to convert back the result in time domain and the modulus of the signal is computed. PCPS is represented on Figure 3.10.

The derivation of all expressions is performed in Appendix A.2.2. The output expression is given in Equation 3.24 and is similar to the result of Section 3.5.

$$
\begin{aligned}
S(\hat{\tau}) =& e^{j(2\pi\Delta f_d\hat{\tau}+\Delta\theta)} * \int_0^{T_{CO}} x(\hat{\tau}) * x(t+\hat{\tau}-\tau) * e^{j2\pi\Delta f_d t}\, dt \\
|S(\hat{\tau})|^2 =& \left| \int_0^{T_{CO}} x(t) * x(t+\hat{\tau}-\tau) * e^{j2\pi\Delta f_d t}\, dt \right|^2
\end{aligned}
\tag{3.24}
$$

Finally, the number of acquisition trials and of samples needed are expressed in Equations 3.25 and 3.26.

$$
N_{trials} = \frac{2 * f_{D,max}}{f_{D,step}} \tag{3.25}
$$

$$
\begin{aligned}
N_{samples} =& N_{trials} * (1023 * S_{p,Chip}) \\
=& \frac{2 * f_{D,max}}{f_{D,step}} * 1023 * S_{p,Chip}
\end{aligned}
\tag{3.26}
$$

3.6.4 Comparison

From the description of the three algorithms that has been made, we can extract and compare the resources needed for each of them. Table 3.1 gathers those information with the expressions of N_{trials} and its resulting values for the standard choice of parameters $f_{d,step}$ equal to 500 Hz and $S_{p,Chip}$ equal to two (τ_{step} is thus half a chip). The number of multipliers and FFT blocks are also given to express the resource usage of the hardware. We see that PCPS requires the most of them, followed by PFS and serial search.

Algorithm	N_{trials}	$N_{standard}$	**Multiplier**	**(I)FFT**
Serial search	$\frac{2*f_{D,max}}{f_{D,step}} * 1023 * S_{p,Chip}$	376,464	4	0
PFS	$1023 * S_{p,Chip}$	2046	1	1
PCPS	$\frac{2*f_{D,max}}{f_{D,step}}$	184	3	3

Table 3.1: Theoretical comparison of the acquisition algorithms. $N_{standard}$ is the number of acquisition trials for the set of parameters: $f_{d,step} = 500$ Hz, $S_{p,Chip} = 2$ and $f_{D,max} = 46$ kHz

From this table, we clearly see that PCPS is the most efficient algorithm in term of computation time due to its very low number of acquisition trials. In addition, PCPS optimization can virtually reduce the number of (I)FFT to two. Indeed, the FFT of the PRN replica only needs to be calculated once for the entire search space of a GPS satellite, since the PRN code used does not change over it. The PCPS acquisition algorithm has therefore be the choice for the implementation of the design that will follow.

One could argue that serial search low complexity would allow for very high computation speed, which is true. However, the satellite acquisition is performed in real time. The processing chain must thus wait available data to perform each acquisition trials, limiting the performance of serial search.

3.7 Signal-to-noise ratio improvements

The propagation of the signal through different mediums and the receiver non-idealities induce the presence of noise at the receiver. In this work, the noise will be assumed to be an **additive white Gaussian noise**, abbreviated AWGN. It will be integrated by the processing chain and risk to cause false alarms. Moreover, an insufficient SNR would not allow the detection of satellites at all since the correlation result would be indistinguishable from the noise.

To increase the output SNR, and thus reduce the noise effects on the detection performance, two methodologies can be employed and will be discussed. First, the effect of the number of samples used during accumulation will be explained. Next, non-coherent integrations will be introduced.

3.7.1 Accumulated number of samples

The AWGN presence at the input of the correlation induces the accumulation of this noise. If N denotes the number of samples accumulated, the correlation output S is a sum of N normal random variables that are uncorrelated and independent. Two hypothesis can now be made regarding the acquisition trial [15], [14]:

- The hypothesis H_0 assumes that the satellite signal with the trial parameters of the acquisition, i.e. code delay and Doppler shift, is not present.
- The hypothesis H_1 assumes the contrary, the satellite signal is present and the trial parameters are correct.

The distributions resulting from those hypothesis are mathematically expressed in Equation 3.27, where S is the correlation output, A is a factor describing the amplitude of the received signal and σ^2 is the variance of the AWGN at the input. From the equation, we can observe that the distribution of S has a mean proportional to the number of samples accumulated under H_1. Otherwise, a zero mean is observed. In both cases, the variance is equal to the one of the AWGN multiplied by N.

$$S \sim \begin{cases} \sum_{n=1}^{N} \mathcal{N}(0,\sigma^2) = \mathcal{N}(0, N\sigma^2) \quad \text{under } H_0 \\ \sum_{n=1}^{N} \mathcal{N}(A,\sigma^2) = \mathcal{N}(AN, N\sigma^2) \quad \text{under } H_1 \end{cases} \tag{3.27}$$

Based on this expression we can analyse the impact of N on the SNR. When it is increased, the correlation result for an acquisition success increases proportionally. However, the positive and negative random noise observations does not accumulate as effectively. Based on the variance and mean of S, the SNR can be derived, as seen in Equation 3.28. The overall SNR increases linearly with N.

$$
\begin{aligned}
SNR_{out} &= \frac{E[S|H_1]^2}{Var[S|H_1]} \\
&= \frac{A^2 N}{\sigma^2}
\end{aligned}
\tag{3.28}
$$

To increase N, two possibilities exist. We can capture more samples on a PRN code by increasing $S_{p,Chip}$. We can also accumulate the correlation output multiple times, which is referred to as coherent integrations. Increasing $S_{p,Chip}$ allows to maintain the time spent on an acquisition trial, but the computation load becomes higher. The opposite is observed for the number of coherent integrations: since we accumulate multiple output results, the time spent on each acquisition trial becomes higher but the computation load does not increase. A trade-off between the two allows to have a correct increase of the SNR at a reasonable price in term of required time and computational load.

To validate the SNR dependencies on those two parameters, Monte Carlo simulations were performed. For this, GPS L1 C/A data were generated using Matlab. AWGN was added and the result was passed through an acquisition channel developed on Matlab, from which the output SNR is extracted. One hundred Monte Carlo simulations were ran for different values of $S_{p,Chip}$ and of the number of coherent integrations. Figures 3.11 and 3.12 compare the theoretical SNR improvement of Equations 3.28 with the one extracted from the simulations. In both cases, theory and practice are aligned, making those parameters pertinent.

Unfortunately, coherent integrations have a major drawback. Due to the navigation message that can change every 20 PRN periods, the coherent integrations become destructive if a bit transition appears. The next method will allow to integrate over a longer duration at the expense of a lower SNR gain.

3.7.2 Non-coherent integrations

The second methodology to improve the SNR is to implement non-coherent integrations. It consists in first squaring the output of the coherent integrations in order to eliminate the navigation message and phase shift dependencies. Then, the result is accumulated K times, as expressed in Equation 3.29.

$$
S_{nc} = \sum_{n=1}^{K} |S|^2 \tag{3.29}
$$

Since the noise is squared too, it will add up constructively, and a positive bias replaces the zero-mean, explaining why non-coherent integrations are less efficient

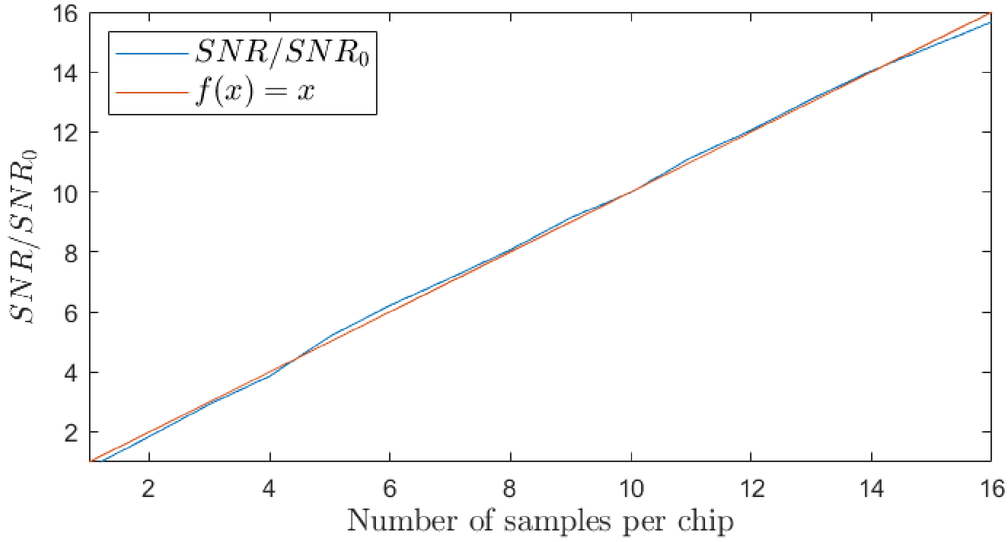

Figure 3.11: SNR improvement as function of $S_{p,Chip}$. The orange curve shows the theoretical relation while the blue one has been obtained using 100 Monte Carlo simulations with an input SNR of -20dB.

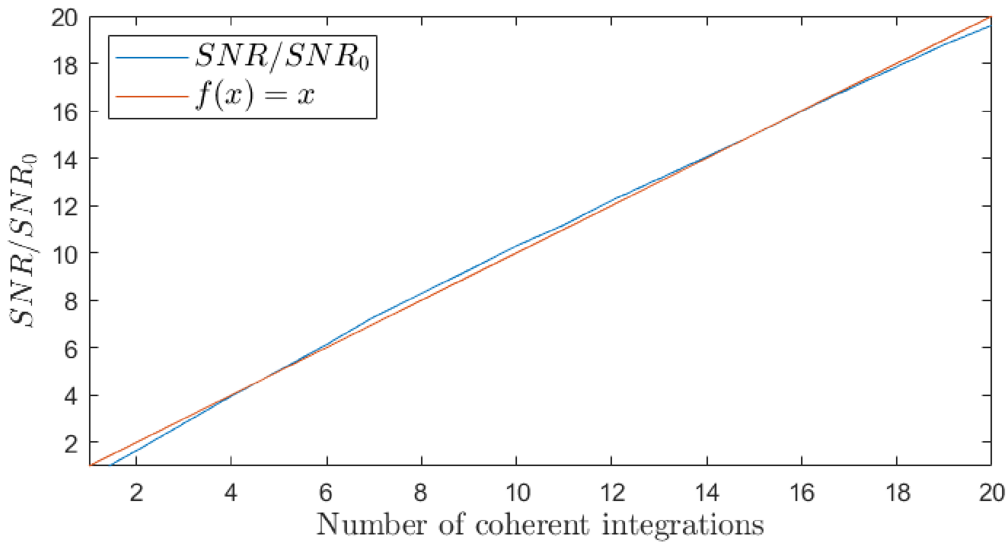

Figure 3.12: SNR improvement as function of the number of coherent integrations. The orange curve shows the theoretical relation while the blue one has been obtained using 100 Monte Carlo simulations with an input SNR of -15dB.

[14]. The distribution of S_{nc} is the sum of $2K$ squared normal variables. When normalized, this is equivalent to a Chi-squared χ^2 distribution with $2K$ degrees of freedom, and a noncentrality parameter λ under H_1 while it is centered under H_0.

The mathematical expression of the distribution is provided in Equation 3.30. The factor two in the number of degrees of freedom comes from the decomposition in real and imaginary components of S as derived in Equation 3.7.

$$\frac{S_{nc}}{N\sigma^2} \sim \begin{cases} \sum_{k=1}^{2K} \mathcal{N}^2(0,1) = \chi^2_{2K} & \text{under } H_0 \\ \sum_{k=1}^{2K} \mathcal{N}^2\left(A\sqrt{N}/\sigma, 1\right) = \chi^2_{2K,\lambda=\frac{2KNA^2}{\sigma^2}} & \text{under } H_1 \end{cases} \tag{3.30}$$

If $K >\simeq 8$, then the central limit theorem can be applied and the Chi-square distribution can be approximated by the normal distribution [14], as expressed in Equation 3.31.

$$\frac{S_{nc}}{N\sigma^2} \sim \begin{cases} \mathcal{N}(2K, 4K) & \text{under } H_0 \\ \mathcal{N}\left(2K(1+A^2N/\sigma^2), 4K(1+2A^2N/\sigma^2)\right) & \text{under } H_1 \end{cases} \tag{3.31}$$

The SNR can now be calculated. Due to the squaring and the positive bias of the noise, it becomes the ratio between the mean of S_{nc} under H_1 minus the noise bias, and between the standard deviation of S_{nc} under H_0. The expression is given in Equation 3.32. We can observe that the SNR only depend on the square-root of the number of non-coherent integrations, explaining their reduce efficiency compared to coherent integrations. As performed in the previous section, simulations were ran to compare the theoretical and practical relation of the SNR as a function of K. The result is shown on Figure 3.13. It is interesting to see that the relation, found using the central limit theorem, fits even for very low values of K.

$$\begin{aligned} SNR &= \frac{E[S_{nc}|H_1] - E[S_{nc}|H_0]}{\sqrt{Var[S|H_0]}} \\ &= \frac{A^2}{\sigma^2}\frac{2KN}{\sqrt{4K}} \\ &= \frac{A^2}{\sigma^2}N\sqrt{K} \end{aligned} \tag{3.32}$$

3.8 Decision logic

Based on the result of the acquisition algorithm, a decision must be made on whether or not a GPS satellite has been detected for a given Doppler shift and code delay. To do so, the hypothesis H_0 and H_1 presented in Section 3.7.1 will be used.

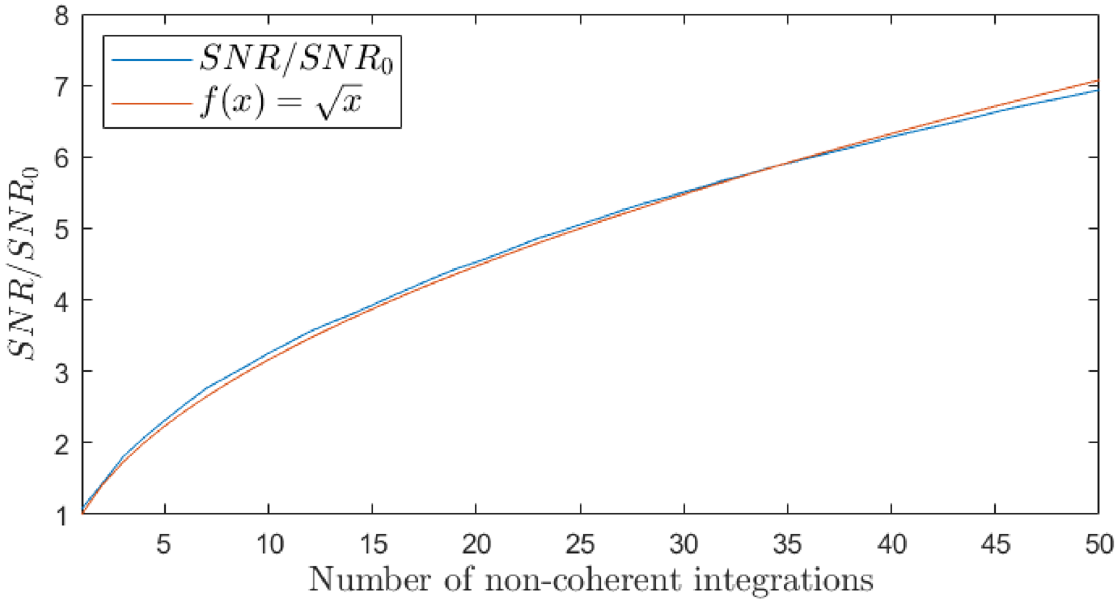

Figure 3.13: SNR improvement as function of the number of non-coherent integrations K. The orange curve shows the theoretical relation while the blue one has been obtained using 100 Monte Carlo simulations with an input SNR of -20dB.

From them, Table 3.2 gathers the different detection cases. First, we have the probability of correct detection P_d. Second, there is the probability of false alarm P_{fa}, when an absent signal is concluded as present. Next, we have the probability of correct dismissal P_{cd}, and finally the probability of missed detection P_{md}.

If we define a threshold value T_h for the detection based on the final output S_{nc}, the equations of the different probabilities explained above can be written as shown in Equations 3.33.

	Signal not present (H_0)	**Signal present** (H_1)
Signal not detected	Correct dismissal (P_{cd})	Missed detection (P_{md})
Signal detected	False alarm (P_{fa})	Correct detection (P_d)

Table 3.2: Detection cases and their probability

$$
\begin{aligned}
P_d &= P(S_{nc} > T_h|H_1) \\
P_{fa} &= P(S_{nc} > T_h|H_0) \\
P_{md} &= P(S_{nc} \leq T_h|H_1) \\
P_{cd} &= P(S_{nc} \leq T_h|H_0)
\end{aligned} \tag{3.33}
$$

Our goal is to minimise the false alarm probability. Indeed, a false detection is costly since it would trigger the tracking system for no reason. Therefore, based on a desired false alarm probability, the threshold value can be derived from F, the cumulative distribution function of S_{nc} normalized under H_0 [15]. As a reminder, when normalized by $N\sigma^2$, S_{nc} follows a Chi-square distribution with $2K$ degrees of freedom. The threshold expression is developed in the Equations 3.34.

$$
\begin{aligned}
P_{fa} &= 1 - P(S_{nc} \leq T_h) \qquad \text{with } \frac{S_{nc}}{N\sigma^2} \sim \chi^2_{2K} \\
&= 1 - F\left(\frac{T_h}{N\sigma^2}, 2K\right) \\
\Longleftrightarrow \frac{T_h}{N\sigma^2} &= F^{-1}(1 - P_{fa}, 2K)
\end{aligned} \tag{3.34}
$$

We have therefore defined a threshold value T_h for the output of the acquisition algorithm. Optimization of the SNR will allow to obtain an high probability of detection.

3.9 FPGA-based SDR receiver

The acquisition working principle have now been presented with various algorithms and improvement techniques. Multiple hardware nowadays exist to perform those operations. For a long time, analog components were used inside common receivers with filters, mixers and so on. Due to the fast development of transistors with scaling going in the nanometer range, operating frequencies and overall performances of digital components began to compete with analog ones.

From this digital rise emerged **software defined radios**, abbreviated SDRs. It is a radio communications system in which most of the processing is performed in software, therefore digitally. The objective is to achieve high flexibility of the receiver. Indeed, whereas an analog chip is almost fixed when laid out (except in case of trimming), the software embedded by a digital chip can be easily updated via communication interfaces. Hence, SDRs allow a higher level of correction and of adaptation to different environment if, for example, a higher SNR is required. Still, analog electronics are needed since the data captured in the real world are by nature continuous. RF front-end is in our case performing those operations and its ADC is the frontier between the analog and digital domains.

SDRs can use different types of hardware. General purpose computer, or GPP, can be used for an easier implementation with traditional software. However, **Field Programmable Gate Arrays**, or FPGAs, are well suited for signal processing. A FPGA is an integrated circuit constitued of a large number of logic blocks that can be interconnected. Its structure can be reconfigured using a hardware description language (HDL) to perform complex operations. FPGAs offer many advantages over more traditional CPU solutions :

- As for the pure software, they are reconfigurable, meaning their processing can be adapted to work at a different frequency or with more samples depending on the need.
- Conversely to a CPU which works sequentially on data (even if parallelism mechanisms are implemented), FPGAs can work in a true parallel fashion, with multiple operations performed simultaneously. This increases the processing speed and the latency which is critical in space application.
- For the same clock frequency, FPGA throughput is higher since there is no OS overhead and each task performed is optimize directly to take as few cycle as possible. It thus limit the power consumption too.
- Logic blocks of a FPGA can be configured to create a soft-core which can perform simple tasks of a traditional CPU. It can thus operate some software that can be provided via an interface.

However, FPGAs have the drawback of being more complex to implement. HDL programming needs to take into account the register-level abstraction layer. Moreover, each task is case specific and traditional libraries are not available.

Chapter 4

Hardware description

In this part, the materials provided by Aerospacelab will be analysed and the different configurations under which it can be used will be described. The RF front-end is provided with different intermediate frequencies, sampling rates and ADC resolutions. As discussed in the previous chapter, acquisition parameters and ADCs resolution can have an impact on the obtained SNR. A careful choice of those parameters can thus reduce the complexity of the algorithm. This helps to limit the required ressources, which are restricted for space operations, and increase the performances of the acquisition.

First, a description of the hardware will be made for the two main components : the RFFE and the FPGA. Next, the specifications of the acquisition will be discussed, by deriving the output SNR from the hardware and acquisition parameters. It will be based on the development performed in previous chapters.

4.1 Radio frequency front end

In this project the MAX2769C chip is used as the radio frequency front-end. It is a next generation GNSS receiver covering multiple frequency bands, including the employed L1 band. To be noted, this RFFE can be used for other GNSS technologies such as Galileo, BeiDou and Glonass [13]. The chip includes a dual-input LNA, mixers, filters, a programmable gain amplifier (PGA) and a multibit ADC. To perform the downconversion to IF, a voltage controlled oscillator (VCO), a crystal oscillator and a frequency synthetiser are embedded.

A fractional clock divider is included in the clock path towards the ADC to obtain a large range of sampling frequencies from the reference input clock. Two registers of 12-bits, L_{count} and M_{count}, allow to derive the ADC sampling frequency based on

Equation 4.1 [13]. The evaluation kit of the MAX2769C chip uses a 16.368 MHz TCXO (temperature compensated crystal oscillator), which is a multiple of the C/A code rate (1.023 MHz). It is therefore easy to obtain a sampling frequency multiple of the C/A code rate and this will be used in the designed implementation.

$$\frac{f_{out}}{f_{in}} = \frac{L_{count}}{4096 - M_{count} + L_{count}} \tag{4.1}$$

Many parameters of the RF front end are tunable. The IF filter following the mixer which downconvert the signal from 1.575 GHZ to the IF has an adaptable order, either a 3rd-order Butterworth filter for a reduced group delay or a 5th-order Butterworth for a steep out-of-band rejection. It can be configured as a lowpass or bandpass filter with a bandwidth choosen between 2.5, 4.2 or 9.66 MHz. When used as a bandpass filter, multiple center frequencies for the filter are available : 3.9, 7.1 and 10.7 MHz. The filter bandwidth is important when discussing the SNR at the output of the ADC since a larger bandwidth will imply a higher noise power.

The programmable gain amplifier, abbreviated PGA, is used to reach a proper input range on the signal at the input of the ADC. The gain of the PGA needs to be configured in order to make the most use of the range of the ADC. A gain not sufficient would not trigger the most-significant bits, or MSBs, of the ADC, lowering the precision on the samples value and hence the SNR. Same approach can be used for a too high gain with, in this case, the least-significant bits, or LSBs, being unused. To ensure an appropriate gain of the PGA, the MAX2769C provides an automatic gain control, or AGC, which optimally configures the PGA gain through a feedback loop. To do so, it counts the number of magnitude bits over 512 ADC clock cycles. The result is compared to a reference value stored as a control word, and the PGA gain is tuned until the compared value are equal. The datasheet of the MAX2769C provides the optimal reference value for each ADC resolutions.

4.2 Field programmable gate array

For this project, the SDR receiver is implemented on a FPGA. As mentioned previously it allows high level of parallelization on the signal processing, leading to higher performances for a more complex design. Furthermore, Aerospacelab has already tested a FPGA for radiations, as it will be embedded on their satellites. Indeed, the electronic components in space will be subject to a higher level of radiations compare to common purpose FPGAs. Microcontrollers would be the other solutions for the signal processing. However, small MCUs may not support the program due to its computation load. Larger MCUs studied by Aerospacelab would represent an overshoot in terms of resources, performances and cost. For all those reasons, FPGAs are an elegant solution.

The tested FPGA is the Artix-7 FPGA Development Board from Xilinx. The characteristics are presented in Table 4.1 where the Artix7-35 and Artix7-100 are compared, the last one being its latest version [16].

FPGA	**Artix 7**	**Artix 7-100**
Logic slice	5200	15850
Block RAM	1800 Kbits	4860 Kbits
DSP Slices	90	240
Internal clock	450 MHz	450 MHz

Table 4.1: Available resources on the Artix 7-35 and Artix 7-100

Logic slices of the FPGA contain four look up tables (LUT) and eight flip-flops. Their number is a limiting factor of the FPGA and of its parallelization property. The DSP slices consist of a multiplier followed by an accumulator. It is used in many signal processing blocks of Xilinx, e.g. the FFT block and the Cordic which will be used in this implementation. The devices also include a of random access memory (RAM) for buffers and fast read/write operations.

For the implementation of the SDR on the receiver, the hardware description language VHDL has been used on the software suite Vivado produced by Xilinx. It is the software development tool used at Aerospacelab for hardware programming and thus the obvious choice for an easy handover of the project at the end of the master thesis. Vivado Design suite includes an associated development environment, a visual interface to construct block diagram and an IP catalog with a large selection of blocks e.g. FFT, Cordic, RAM.

4.3 Specifications

From the information regarding the hardware provided above, the specifications of the acquisition system can be derived. The SNR at the output of the MAX2769C will be quantified based on its datasheet, presented earlier. Next, a possible output SNR for the acquisition processing chain will be computed by specifying the acquisition parameters, e.g. Doppler and code step, number of integrations.

4.3.1 RF front end output SNR

To compute the SNR obtained at the output of the RFFE, the carrier-to-noise ratio first need to be quantified. In Section 2.7.2, the received power has been determined equal to −159.5 dBW in the worst case scenario. From this, the carrier-to-noise ratio can be derived as done in Equation 4.2, where T_{sys} is the the system noise temperature and k is the Boltzmann constant.

$$ {}^{C}/_{N_0}[dB] = P_r - 10 * log(k * T_{sys}) \tag{4.2} $$

The system noise temperature represents the quantity of thermal noise generated and captured by the system, in this case the antenna and the RFFE. For facility, we will assume a passive antenna since it does not bring any gain, making it the worst case scenario. The noise temperature is tightly linked to the noise factor which expresses the variation of SNR between the input and output of a device. T_{sys} is expressed in Equation 4.3, where T_{ant} is the antenna noise temperature, in our case considered to be the sky noise temperature between 3 to $10K$ [17]. T_0 is the standard noise temperature of 290 K and F is the cascaded noise factor of the MAX2769C provided in its datasheet and equal to 1.4 dB.

$$ \begin{aligned} T_{sys} &= T_{ant} + T_0 * (F - 1) \\ &= 120K \end{aligned} \tag{4.3} $$

The carrier-to-noise ratio is therefore equal to 48.3 dBHz. If the 2.5 MHz filter bandwidth is selected, the SNR at the output of the RFFE can be expressed as shown in Equation 4.4.

$$ \begin{aligned} SNR[dB] &= {}^{C}/_{N_0}[dB] - 10 * log(BW) \\ &= -15.7dB \end{aligned} \tag{4.4} $$

4.3.2 Acquisition output SNR

Now that all the implementation losses and gains have been expressed, a computation of the theoretical acquisition output SNR can be derived. For this, Table 4.2 summarizes all the parameters affecting the SNR discussed in this master thesis.

The table also includes a quantification of its resulting value based on a standard set of parameters used in GPS receivers [9].

Parameter	Impact on SNR	Value	Quantified impact [dB]
Input SNR			-15.7
Number of sample per PRN Code $S_{p,code}$	$1023 * S_{p,Chip}$	2046	33.1
Number of coherent integrations N_{co}	N_{co}	5	7
Number of non-coherent integrations K	$\sqrt{K}$	5	3.5
Doppler step $f_{d,step}$	$\left\lvert \frac{2sin(\pi f_{d,step} T_{CO}/2)}{\pi f_{d,step} T_{CO}} \right\rvert$	500 Hz	-0.91
Code step τ_{step}	$\left(1 - \frac{\lvert\tau_{step}\rvert}{2T_{chip}}\right)$	$T_{chip}/2$	-2.5
Output SNR			24.9

Table 4.2: Calculation table for the acquisition output SNR

We see that for this selection of parameters, a resulting SNR of 24.5 dB is theoretically achieved. In the other way, this table allows to derive the parameters of the implementation based on the desired SNR.

Chapter 5

Acquisition channel implementation

In this chapter, the implementation of the previously described FPGA-based SDR acquisition channel will be explained and characterized. First, a possible architecture of the complete GPS receiver will be introduced. The controller interface with the acquisition, allowing to configure the acquisition channels, will be discussed. Next,the **acquisition channel architecture**, derived from the PCPS algorithm, will be explained based on a block design, each block performing a part of the algorithm.

The communication protocol between the blocks, **AXI4-Stream**, will then be presented. A description of the behavioral testing methodology of the individual blocks of the system will be made. Afterwards, each section will discuss a specific block, or group of blocks, with their inputs and outputs, their working principle and their implementation. Specificities of each block will be presented.

To be noted, this implementation of an acquisition channel is a proof of concept. The objective is to obtain a test result which confirms the possibility to acquire a satellite with a FPGA-based SDR implementation. The actual implementation is therefore not completely optimized. Coherent integrations have not been implemented in this design but are an improvement path for the follow-up of this master thesis. However, non-coherent integrations are part of the design since they allow to obtain a result independent from the navigation message and from the phase shift, which is essential.

5.1 Architecture of the receiver

The possible architecture of a complete FPGA-based SDR GPS receiver is represented on Figure 5.1. First, the radio-frequency front end is provided with an antenna at the beginning of the processing chain. The acquisition, tracking and navigation decoder are labelled as hardware since they can benefit from the high level of parallelization reachable with a FPGA. The controller and PVT equations resolutions would be performed in software for facility since their computation is less frequent and harder to implement in hardware.

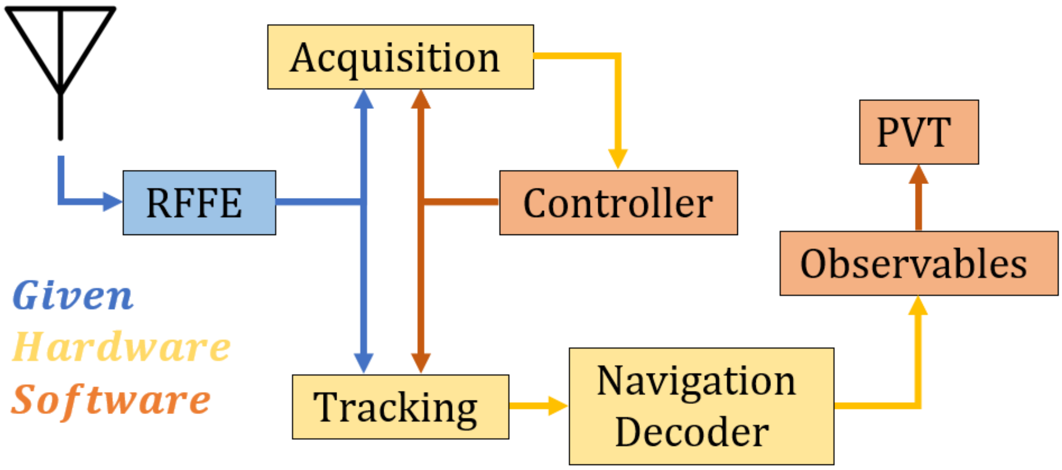

Figure 5.1: General architecture of a FPGA-based SDR GPS receiver

The controller is the part of the system responsible for the coordination of the different operations. It knows the actual state of the whole design, and can thus send control signals to the acquisition channels. The results of the channels are forwarded to the controller so that it can launch the tracking of the acquired satellites. The interface between controller and acquisition channels will now be specified.

The controller can modify multiple acquisition parameters via a number of control signals. The Doppler step can be tuned to change the precision on the Doppler shift determination while the Doppler range can be updated if real-time event requires it. The number of non-coherent integrations can be modified to trade computation time for SNR. Finally, the controller decides the satellites looked for. Thus, it provides the code phase selector, explained in Section 2.5.2, which is needed to generate the specific PRN code of a GPS satellite.

5.1.1 Architecture of the acquisition channel

As mentioned previously, the PCPS algorithm will be used and its practical block design can be seen on Figure 5.2. Zero padding has been added before the FFT since this operation is much more efficient for specific number of points, e.g. power of two or four, especially for hardware implementation. Indeed, a N points Radix-2 FFTs for example can be implemented as an interconnection of smaller $^{N}/_{2}$ points FFTs, reducing the complexity and the computation time.

A buffer is introduced before the complex multiplier and implemented as block RAM. Indeed, the PRN replica spectrum is not modified until the entire acquisition search space has been analysed. Regenerating its spectrum for each correlation performed would induce a high performance loss. The Doppler wipeoff internal operation has been changed to use the Cordic block from the IP catalog of Xilinx. Its working principle will be explained in the block-by-block analysis.

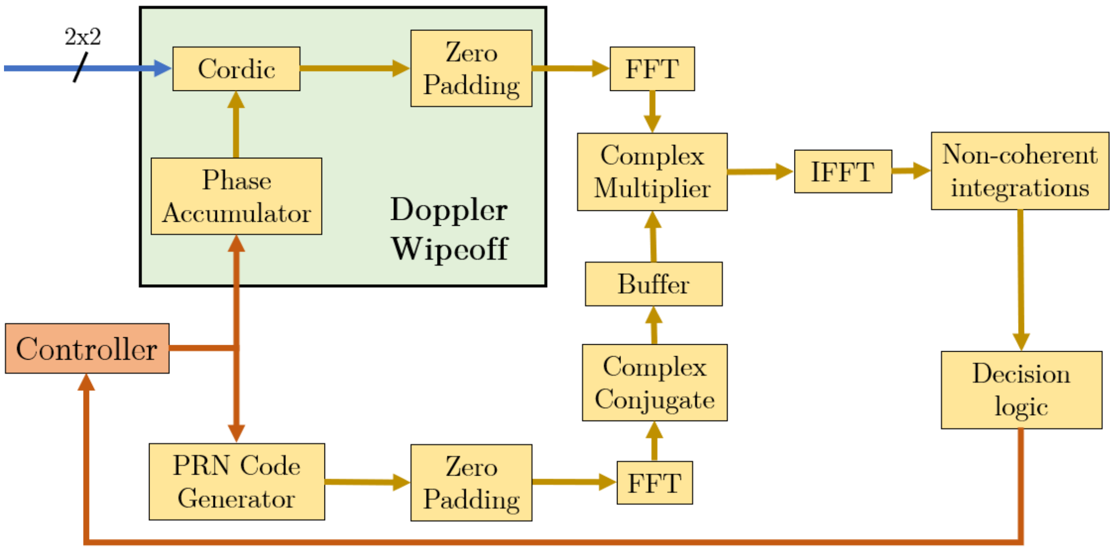

Figure 5.2: Block design of an acquisition channel using PCPS. The inputs are I and Q samples both encoded on 2 bits

5.1.2 Communication protocol : AXI4-Stream

The Advanced eXtensible Interface (AXI) is a high-performance communication protocol linking multiple masters and slaves with high parallelism and at high frequency. It is part of the ARM Advanced Microcontroller Bus Architecture (AMBA)

3 and 4 [18]. In this project, the used protocol is AXI4-Stream, an interconnect infrastructure of the AXI type which allows the reliable communication between a master and at least one slave.

AXI4-Stream transmits a fixed number of bits over the bus at each clock cycle where an handshake is performed, i.e. when the master and slave are respectively ready to send and receive data. For the handshake, AXI4-Stream uses additional signals. Those employed in this master thesis are specified in Table 5.1. The handshake process is completed if *TREADY* and *TVALID* are high on the same rising edge of the clock. The value on the *TDATA* bus is then stored at the slave via a register. This is shown on Figure 5.3. To be noted, the *TLAST* signal is only used if the data word are transmitted as frames. It is especially useful in case of GPS acquisition since data are processed as blocks of PRN code.

Signals Name	**Bus width (bits)**	**Source/Dest.**	**Description**
TDATA	Data length	Master to Slave	Data word to exchange
TVALID	1	Master to Slave	Enabled when the Master is ready to send a data word
TREADY	1	Slave to Master	Enabled when the Slave is ready to receive a data word
TLAST	1	Master to Slave	Enabled on the last data word of the frame.

Table 5.1: AXI4-Stream signals used in this implementation

5.1.3 Methodology for the testing of the system

For the testing of the acquisition blocks, a testbench is implemented in VHDL, instantiating the desired block as unit under test (UUT). The testbenchs allow to provide input data from a text file and to write the output into another one. Control signals for the UUT are generated internally and input signals are the result of the previous blocks. A general representation of the testing is depicted on Figure 5.4.

For the Doppler wipeoff, which is the beginning of the processing chain, input data are generated with Matlab : the *cacode* function generates the PRN code of the desired satellite, which is then multiplied with a high frequency sine wave, at IF

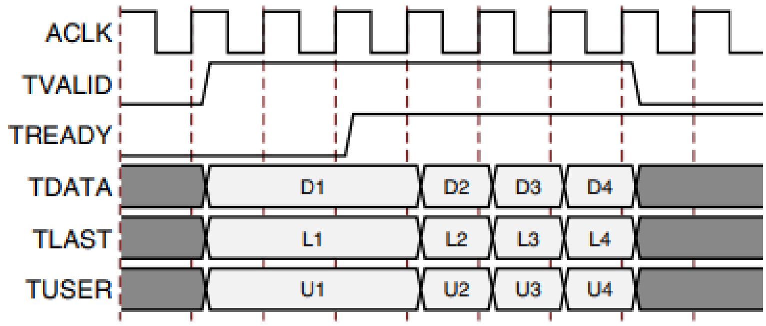

Figure 5.3: AXI4-Stream protocol handshake [19]. TUSER is an optional signal, used to transmit additional information in parallel to TDATA. It is used mostly for debugging.

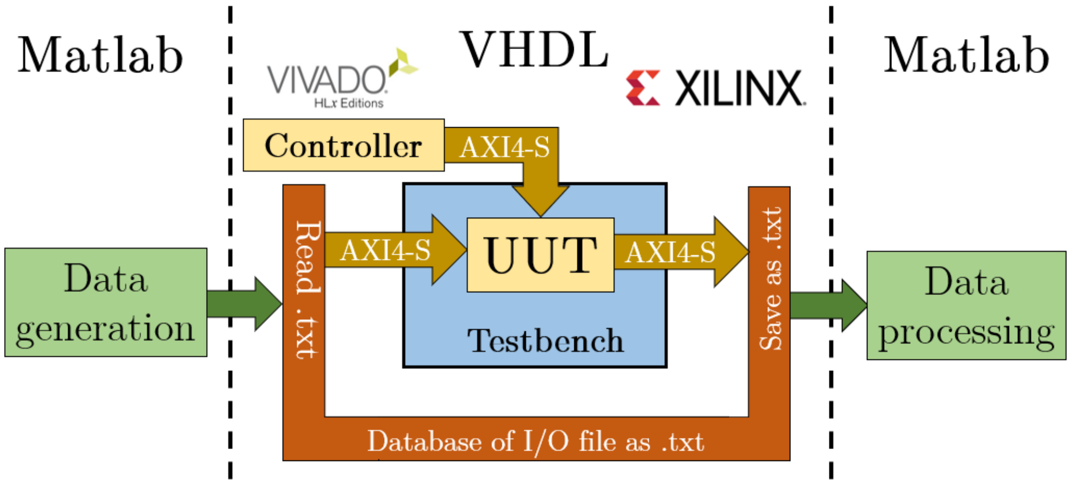

Figure 5.4: General block testing representation

plus a fixed Doppler shift. The resulting signal is then resampled on 2-bit to match the RFFE ADC output.

To test the output of the UUT, a complete acquisition channel implementation as first been made on Matlab. It allows to know the expected behavior of the implementation. UUTs outputs are extracted and then forwarded to the Matlab processing chain. Its result is then compared with the complete Matlab implementation.

Multiples cases are tested with to validate the correct behavior of the block. All the blocks that will be described in this chapter showed proper results with the behavioral testing which will therefore only be discussed in specific cases. However, a testing on real GNSS data will be performed in the next chapter. For this design, eight samples are used per chip. This high values compensates the absence of coherent integrations in this proof of concept. However, this parameter is easy to modify using the block design of Vivado.

5.2 PRN code generator

The PRN code generator is directly implemented from the algorithm presented in Section 2.5, using two LFSRs and XOR logic operations. An AXI4-S interface with the controller is added to provide the PRN code phase selector needed to generate the PRN code of a specific satellite. Each output bit of the PRN code generator is then replicated during multiple cycles in order to obtain the desired number of samples per chip. The output is then forwarded to the FFT using AXI4-S.

A counter tracks the number of output data so that when the complete PRN code has been generated, zero padding is added. It completes the frame in order to allow the use of a Radix-2 or Radix-4 FFT. There is only one chip missing between 1023, the number of chip in the PRN code, and the next power of two, 1024. Therefore, a quantity of zeros equal to the number of samples per chip are appended at the end of the code. When finished, the *TLAST* signal is triggered, identifying the end of the frame.

5.3 Doppler wipeoff

The Doppler wipeoff is the first block of the acquisition chain. It downconverts the signal from IF to baseband and compensates a trial Doppler shift. It is implemented using the **Cordic** algorithm, provided by Xilinx as an IP, coupled with a phase accumulator.

5.3.1 Cordic Algorithm

The Cordic algorithm, standing for **Co**ordinate **R**otation **Di**gital **C**omputer, allows the computation of trigonometric and hyperbolic functions, multiplications, square roots, by using only basic arithmetic operations and lookup tables. In our case, the algorithm is used for complex samples rotation, which is represented on Figure 5.5 and expressed in Equation 5.1 [20].

$$\begin{aligned} x_r &= x_i cos(\phi_r) - y_i sin(\phi_r) \\ y_r &= x_i sin(\phi_r) + y_i cos(\phi_r) \end{aligned} \tag{5.1}$$

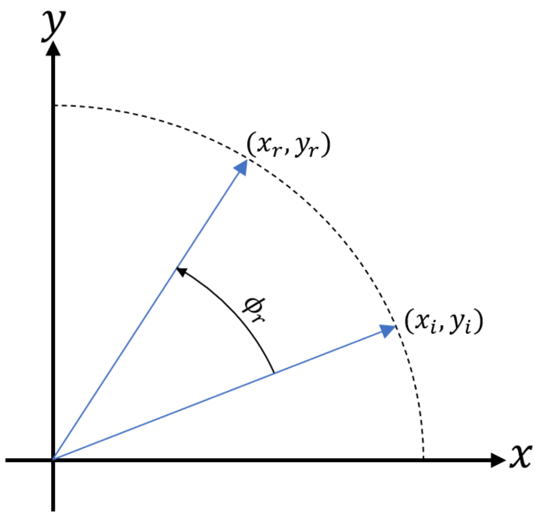

Figure 5.5: Representation of vector rotation as performed by the Cordic algorithm

Since the frequency is the derivative of the phase, see Equation 5.2, it translates to a phase accumulation between each sample by a phase increment ϕ_c. As shown in Equation 5.3, it is dependent on the IF, the Doppler shift trial and the sampling frequency. A downconversion to baseband can be interpreted as the compensation of ϕ_c. Coordinate rotation is thus performed on each sample with a different phase, in order to obtain a steady constellation on the phase diagram. A representation of this principle is shown on Figure 5.6. The remaining constant phase is not important if the modulus is applied at the end.

$$f = \frac{1}{2\pi}\frac{d\phi(t)}{dt} \tag{5.2}$$

$$\begin{aligned}
\phi[n] &= \phi[n-1] + 2\pi \int_0^{\frac{1}{f_s}} (f_{IF} + \hat{f_D})\, dt \\
&= \phi[n-1] + 2\pi \frac{f_{IF} + \hat{f_D}}{f_s} \\
&= \phi[n-1] + \phi_c
\end{aligned} \tag{5.3}$$

5.3.2 Phase accumulator

To provide the phase to compensate for, a second block is needed: a phase accumulator. Its purpose it to integrate the phase increment to obtain the phase rotation value, $\phi[n]$ in Equation 5.3, that will be performed by the Cordic. As discussed in Section 5.1, the Doppler step and range, and the value of the IF are provided to the acquisition system by the controller. Based on their value, the phase accumulator can compute the phase increments corresponding to the frequency downconversion

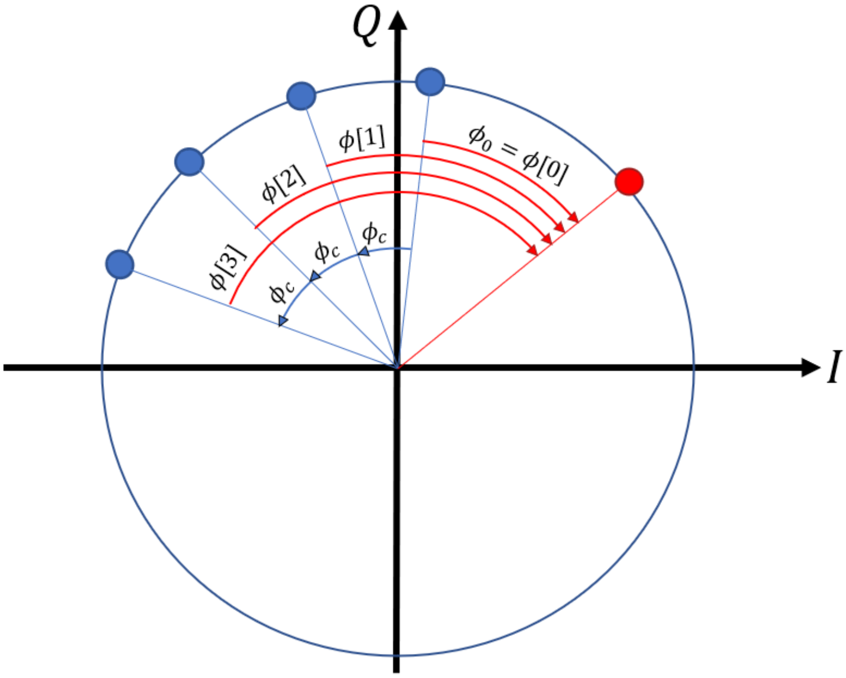

Figure 5.6: Representation of the phase compensation through complex rotation. The result is a steady point on the phase diagram, in red

of each Doppler shift trials $(f_{IF} + \hat{f_D})$.

The bitwidth used to represent the phase need to be discussed. Even if the input signal is sampled on two parts of two bits, the phase requires a higher bitwidth to handle the accuracy desired with the Doppler step. The Cordic block IP of Xilinx takes an input phase between $-\pi$ and π, encoded in the 2QN format. This format uses $N-3$ bits for the decimal part [20]. Based on the expression of ϕ_c, the minimal Doppler step Δf achievable with a phase bitwidth of N is derived in Equation 5.4.

$$\begin{aligned} \Delta f[N] &= \frac{2^{(N-3)} f_s}{2\pi} \\ \Delta f[16] &= \frac{2^{(13)} f_s}{2\pi} \end{aligned} \tag{5.4}$$

From this expression, a Doppler step resolution of 159Hz can be reached with 16 bits when using a sampling frequency of 8.184 MHz. This is below the standard value of 500 Hz used in GPS ground receiver. 16 bits is also a common value since it can be encoded on 2 bytes.

An additional counter is included to divide the data stream into frames. As for the PRN code generator, zero padding is added and followed by the $TLAST$ signal.

5.4 Fast Fourier Transform

The FFT is provided by Xilinx in its IP catalog [21] and implements the Cooley-Tukey FFT algorithm. It can use either the Radix-2 or Radix-4 decomposition. The first one has been chosen here for its lower resource usage. The Xilinx block also implements the inverse FFT which will be discussed in the meantime since it shares the same specificities.

5.4.1 Output bit growth

For the Radix-2 decomposition, it is performed using $log_2(N_{FFT})$ stages where N_{FFT} is the transform size, each one performing additions and substractions, increasing the data bitwidth by one. The maximum bitwidth at the output is given as the input bitwidth + $log_2(N_{FFT})$ + 1. Moreover, the processing chain also includes a complex multiplier, potentially doubling its input bitwidth, and an IFFT. The complete output bitwidth for a 16 bits input is evaluated in Equation 5.5 for the smallest FFT length usable.

$$\begin{aligned} N_{bit,out}[N_{FFT}] &= 2 * (16 + log_2(N_{FFT}) + 1) + log_2(N_{FFT}) + 1 \\ N_{bit,out}[1024] &= 2 * (16 + 10 + 1) + 10 + 1 \\ &= 65 \end{aligned} \tag{5.5}$$

The output bitwidth is here multiplied by a minimum four factor which is too resource consuming. A scaling can be applied on the FFT block to divide the result of each stage by a factor 1, 2 or 4 and maintain the input bitwidth. An analysis of the FFT block output has thus been performed to determine the scaling factor minimizing the loss information at the output. Indeed, the part of the data bitwidth conveying the most information might be concentrated towards the MSBs or the LSBs.

The methodology is equivalent for the two used FFTs and the IFFT. It is done as follows. The block is first used in unscaled mode, with the full output bitwidth. The output is extracted to Matlab and a bit index analysis determines the positions of the highest and lowest used bits. The result can then be plotted on an histogram. This is shown for the FFT following the Doppler wipeoff block on Figure 5.7.

We see that the used bitwidth grows up to 25 bits. The scaling to apply must be at least nine to keep all the used MSBs from the indexes 10 to 25. Using this method

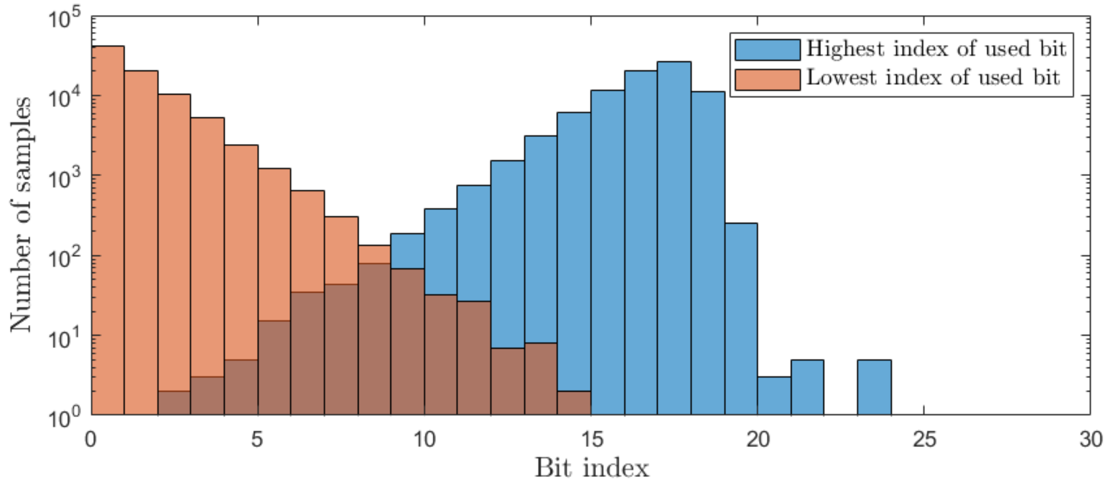

Figure 5.7: Highest and lowest indexes of the used bits for the output of the FFT following the Dopplerwipeoff in unscaled mode

for all FFT and IFFT blocks, the scaling factors are chosen. For the two FFTs, it is selected to be 10. The output of the IFFT only uses the 16 LBSs, hence no scaling is required.

5.4.2 Zero Padding effect

Addition of zeroes before the FFT block is made to meet the specifications of the block of Xilinx. However, this has a drawback at the output of the processing chain that can be observed even on the Matlab model. Indeed, since the received signal code and the generated PRN replica are unaligned, zero padding is thus not placed at the same index. A simple representation of the problem is given in Figure 5.8 where two initially identical sequences **A** and **B**, with **B** circularly shifted by a delay τ, have been zero padded (blue rectangles).

If we perform the correlation, we see that the two sequences are never perfectly aligned. This is represented on the figure, two delays for the correlation, τ and $\tau + padding$, share a partial alignment of the sequence. This leads to a reduction of the auto-correlation property. The reduction factor depends on the initial delay between the sequences, τ. In the worst case scenario, it is equal to half the sequence length and the codes are maximum half aligned. This is equivalent to a SNR loss of 6 dB which needs to be taken into account.

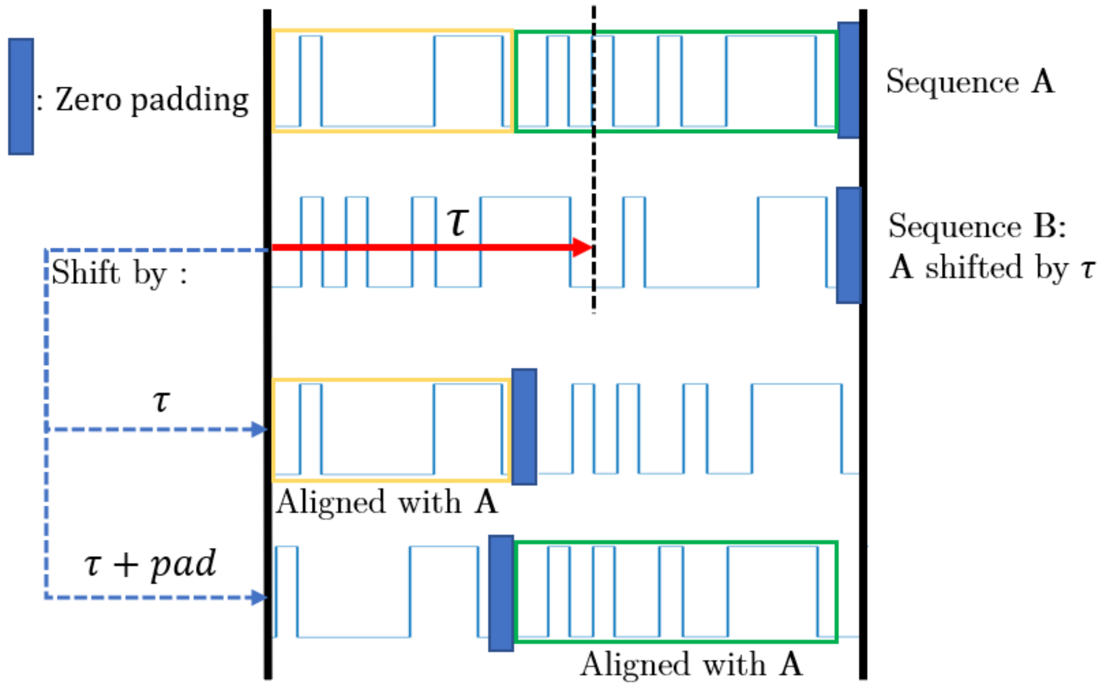

Figure 5.8: Zero padding effect representation on the correlation result for two identical sequences, **A** and **B**, unaligned by a delay τ and padded. The two bottoms sequences are shifted versions of **B**, only partially aligned with **A**

5.5 PRN replica buffer

When the FFT of the PRN code has been computed, the next step in the algorithm is to compute its complex conjugate and forward it to the complex multiplier. As mentioned earlier, it would be highly energy consuming to generate the spectrum of the PRN code each time a frame of the received signal is processed.

The solution chosen is to use random-access memory, **RAM**, as buffers to store the result of the FFT. Every acquisition channel would have its own buffer, each one can then try to acquire a different satellite. Since iterating over the entire search space requires a lot of frequency domain correlations, the combination of PRN code generator and FFT would mainly be in sleep state. A single instantiation of those two blocks could thus be used for the whole acquisition system, forwarding its result to a given acquisition channel. However, This solution comes at the price of the addition of a RAM block in each acquisition channel. Its memory size is defined by the maximum number of samples used by a PRN code.

Since this buffer is mainly read and not written, a single port RAM is used. A finite-state machine, or FSM, is designed to encapsulate the RAM such that reading or writing operations are always performed over the entire buffer before another operation can take place. It avoids corruption of the memory. The schematic of the FSM is presented on Figure 5.9. For the reading operation, a counter has been added both for the address selection in the memory and the *TLAST* signal generation which marks the end of the frame and of the buffer reading.

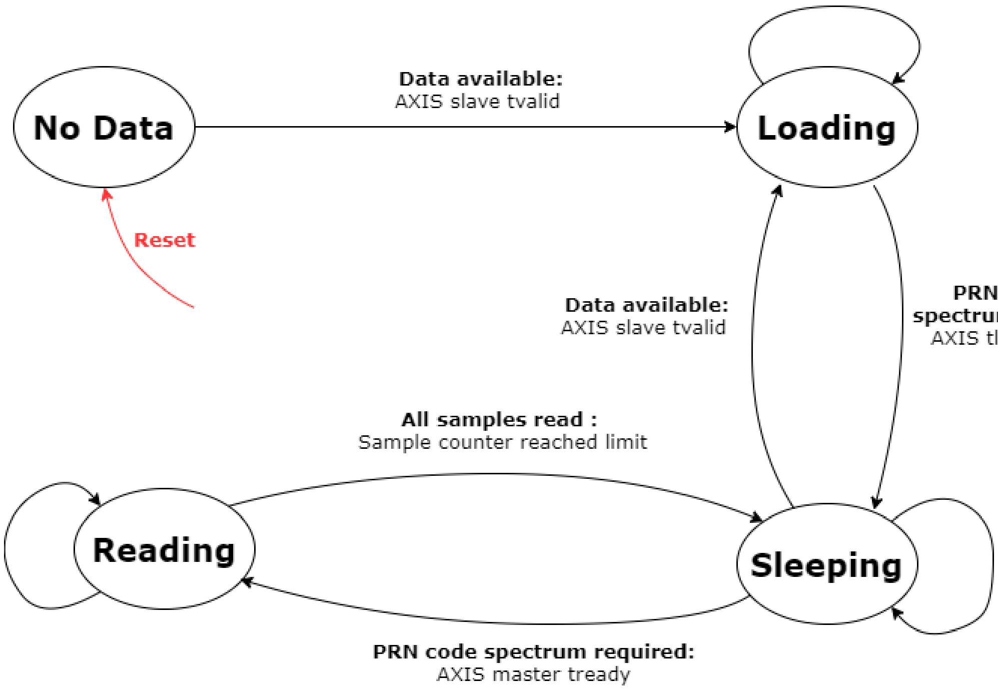

Figure 5.9: FSM of the PRN replica buffer. The red arrow implies that this state transition can come from any of the FSM states. Writing operation are referred to as the *Loading* state

Finally, since the frequency domain correlation requires to use the complex conjugate of the spectrum, its value is computed before being written into the RAM. It is done by simply taking the opposite of the sample imaginary part.

5.6 Complex Multiplier

The complex multiplier block has been selected in the IP catalog of Xilinx. It is connected to the PRN replica buffer and to the FFT following the Doppler wipeoff, for reminder see Figure 5.2. Both streams are encoded as 32 busses with a 16 bits imaginary and real parts. However, as for the FFT blocks, if no measure are taken, the output width grows to 33 bits for each part. A similar analysis to the one performed for the FFT and IFFT has been conducted to identify the bit index range at the output which conveys the most information. The result showed that the bit indexes 12 to 27 were the most important. Truncation of the output was thus performed to isolate those 16 bits.

5.7 Non-coherent integrations

To implement non-coherent integrations, we first need to obtain the modulus of the correlation result. The Cordic algorithm can again be used, here to calculate the magnitude of the complex number at the output of the IFFT. Indeed, the Cordic block in *Translate* configuration uses micro-rotations up to the point where the complex data is only on the real axis. The output real part is therefore the magnitude of the complex data. An internal scaling compensation allows to maintain the initial bitwidth.

Next, the Cordic result needs to be accumulated over multiple frames. A buffer is hence required to store the result during the accumulation. For this, a RAM block is employed again. To accumulate, the prior result is read in the memory, summed with the received data, and then written back. Each memory address corresponds to an index of the correlation result. A counter is used to synchronize the index in the received frame with the address in memory for the read and write operations.

The FSM scheme used for the implementation is represented on Figure 5.10. The state *FirstLoading* is only accessed for the first output frame of a Doppler shift trial, and does not involve read operations on the buffer. Indeed, at this moment the content of the memory is the one of the previous Doppler shift trial, thus out-of-date. For the following output frames, the succession of reading and writing operations are performed with the states *ReadingCell* and *WritingCell*. When the buffer is completely loaded, reading of the buffer has priority so that the result of the non-coherent integrations is not lost.

Finally, an AXI4-S interface with the controller allows to change the number of non-coherent integrations between acquisition trials.

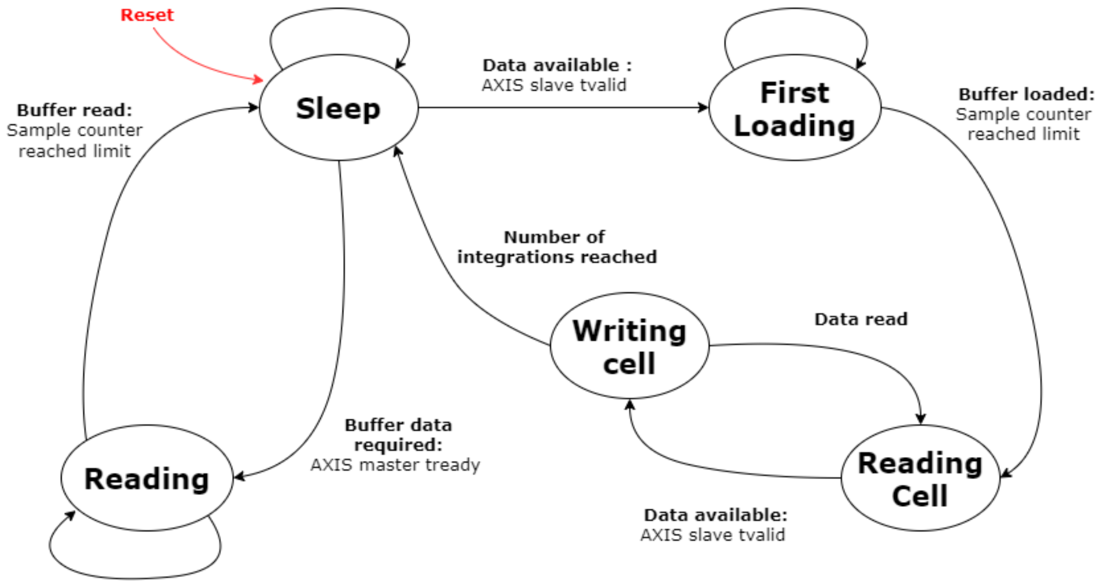

Figure 5.10: FSM of the non-coherent integrations buffer. The red arrow implies that this state transition can come from any of the FSM states

5.8 Discussion of the implementation

To conclude this part, we can summarize the important aspects tackled and the objectives achieved. From the parallel code phase search acquisition algorithm, the practical block design has been derived, adapted to fit the constraints of the hardware, e.g. zero padding addition to use Radix-2 FFTs. The working principle of each block has been explained and the important specificities of this FPGA implementation have been discussed to facilitate the handover of the project to Aerospacelab. The bit growth and the loss due to zero padding are the two most critical aspects of the design at this point. In the end, a working VHDL implementation of a GPS acquisition channel was obtained and will be tested on GNSS data in the next chapter.

Improvement paths can however be suggested. Coherent integrations should be implemented to help reach an higher signal-to-noise ratio. To do so, the same principle as for non-coherent integrations can be applied. A buffer using a RAM would be used to accumulate the result of the IFFT. Next, the use of a sampling frenquency providing a Radix-2 number of points per frame could be studied to avoid the need of zero padding and its induced loss. It would imply the addition of an interpolator to properly resample the generated PRN code replica. The next steps of the GPS acquisition system development should optimize this design and perform a synthesis to quantify the used resource of the FPGA.

Chapter 6

Implementation testing on GNSS data

This final chapter will be dedicated to the testing of the implementation developed previously. The objective here is to show the correct behavior of the actual acquisition channel on captured GNSS data. To validate the design, a satellite acquisition will be performed over a defined search space. The extracted output data from the channel implemented on Vivado should show a peak value significantly higher than the noise for the correct GPS signal unknowns. However, since GNSS data for LEO GPS receivers are unavailable, a ground antenna has therefore been used for this test.

In the first section, the acquired GNSS data will be detailed with a description of the setup used for their capture. Afterwards, an explanation is provided on the pre-processing performed to match the output data format of the MAX2769C chip. In a second time, we will present the result of the acquisition channel on those formatted data and then provide a discussion.

6.1 GNSS test data description

As mentioned above, the lack of data set for low-Earth orbit GNSS receiver constrains our test to samples captured on the Earth surface. Section 2.7.2 has discussed the received power, and a theoretical difference of 1 dB was observed between ground and LEO GPS receivers. This does not include additional signal propagation effects such as multipath or obstruction which are more dominant for ground receivers. A conclusive result for ground captured data should thus be promising for the low-Earth orbit scenario.

6.1.1 Hack-RF One setup

The setup for the data capture includes a HackRF one [22] which is a software defined radio peripheral capable of transmission and reception in large frequency spectrum, from 1MHz to 6GHz. It has been provided by Aerospacelab and is an open source platform. A GPS active external antenna from Mikroe, the MIKROE-3373 [23], is connected to the HackRF one. To obtain a sufficient accuracy for GPS data capture, a 0.5PPM 10MHz TCXO module add-on [24] designed for HackRF is connected to ensure ultra-low phase noise and high frequency stability.

This setup is finally connected to a PC running the open source software-defined receiver, GNSS-SDR [25]. It allows to configure the HackRF one and to extract the processed data. The configuration file used for the data capture is available in Appendix B. The complete setup is depicted on Figure 6.1.

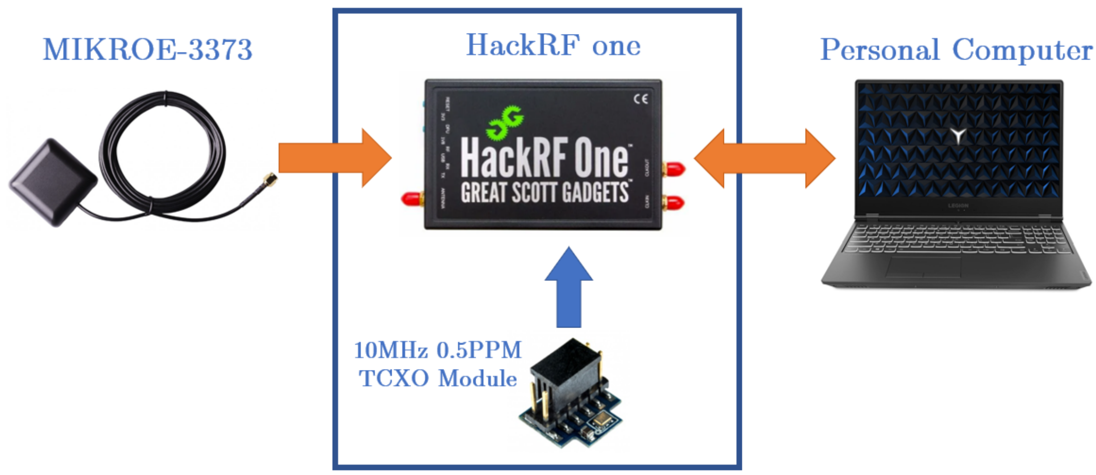

Figure 6.1: GNSS data capture setup

6.1.2 Data extraction and pre-processing

The data received from the HackRF one are not output with same format as the MAX2769C chip. First, the sampling frequency is dictated by the TCXO module added which is 10MHz. Therefore, an interpolation is performed via Matlab to obtain the required sampling frequency, here selected to be 8.184 MHz which results in eight samples per chip of the PRN code.

Second, the output resolution of the HackRF one is much higher than the one employed by the MAX2769C. The data has thus been resampled on 2 bits for both the I and Q components.

GNSS-SDR also provides the results of its acquisition system, with the ID of the detected satellite (referring to its PRN code), the Doppler shift and the code delay observed. This allows to select the good code phase selector for the PRN code generation and to know the expected results of the acquisition channel. For the following test, the satellite with PRN code 18 is being acquired, with a Doppler shift of 500Hz and a code delay of 324 chips returned by GNSS-SDR. However, a deviation from this Doppler shift will be tolerated for the test result since a Doppler step of 500 Hz was used by GNSS-SDR.

6.2 Acquisition channel testing

For this test, the acquisition channel implemented on Vivado uses eight samples per chip and the number of non-coherent integrations K was fixed to ten. Based on the configuration of the phase increment, see Section 5.3.2, the Doppler step is fixed to 477 Hz. It is the closest value to the standard 500 Hz that can be used. Since the real Doppler shift is known to be around 500 Hz thanks to GNSS-SDR, the search space will be constrained to a range of $\pm$ 3 kHz to limit the simulation time on Vivado.

To allow a comparison with a working GPS signal acquisition, Matlab has been used to perform the acquisition with the same parameters and input data as the Vivado implementation. The normalized result from Matlab is depicted on Figure 6.2 where a SNR of 9.6 dB is extracted. The observed Doppler shift and code delay are close to the ones anticipated, 477 Hz and 323.5 chips.

Next the Vivado acquisition channel has been simulated. The result is depicted on Figure 6.3. The extracted SNR reaches 9.9 dB, for a determined Doppler shift of 477 Hz and a code delay of 323.5 chips.

The acquisition result from Vivado shows a significant peak value at a precise point, higher than the noise level. This point refers to the correct GPS signal unknowns. Moreover, the results from Vivado and from Matlab match in term of shape and of SNR.

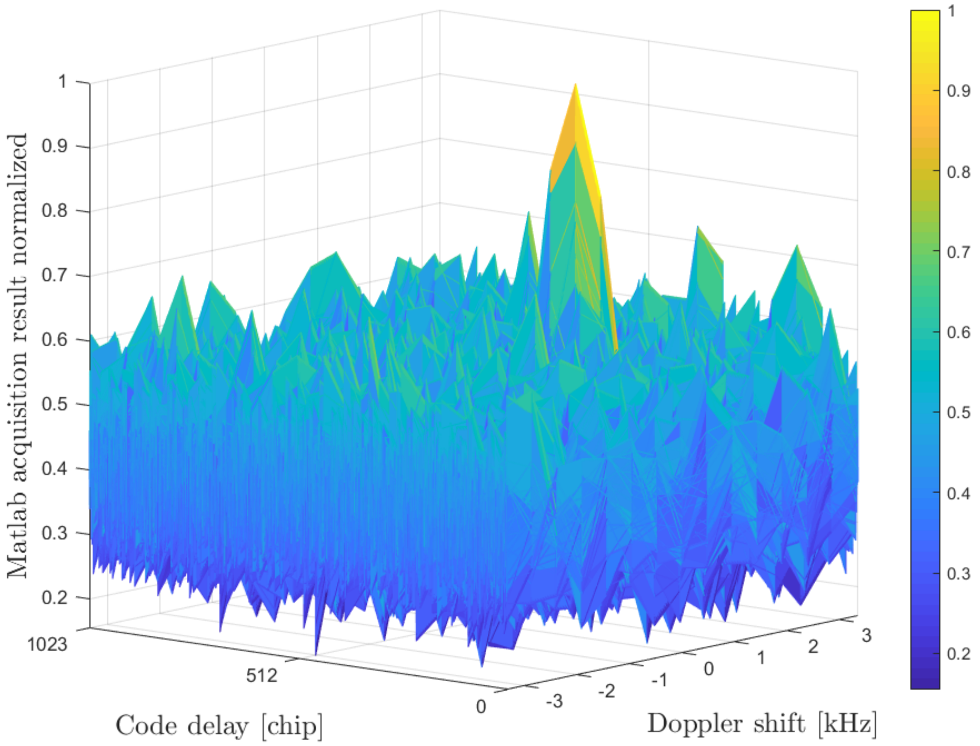

Figure 6.2: Matlab acquisition result for the detection of the satellite with the PRN code 18, experiencing a code delay of 324 chips and a Doppler shift of 500 Hz. The used parameters were $f_{d,step} = 477$ Hz and $K = 10$. The peak is positioned in $f_d = 477$ Hz and $\tau = 323.5$ chips

6.3 Discussion of the result

The test result obtained is conclusive with a correct identification of the GPS signal unknowns and a SNR of 9.9 dB for GNSS data captured on the ground. This validates the correct behavior of the implemented acquisition channel and is promising for the complete development of the FPGA-based SDR GPS receiver. Furthermore, we can extrapolate this result to the LEO GPS acquisition case.

First, as mentioned previously the received signal power at the receiver would theoretically be one decibel lower than for a ground receiver. However, this does not take into account obstructions due to clouds and human or natural obstacles, e.g. trees, building. Also, reflection and multipath becomes less important at the LEO altitude.

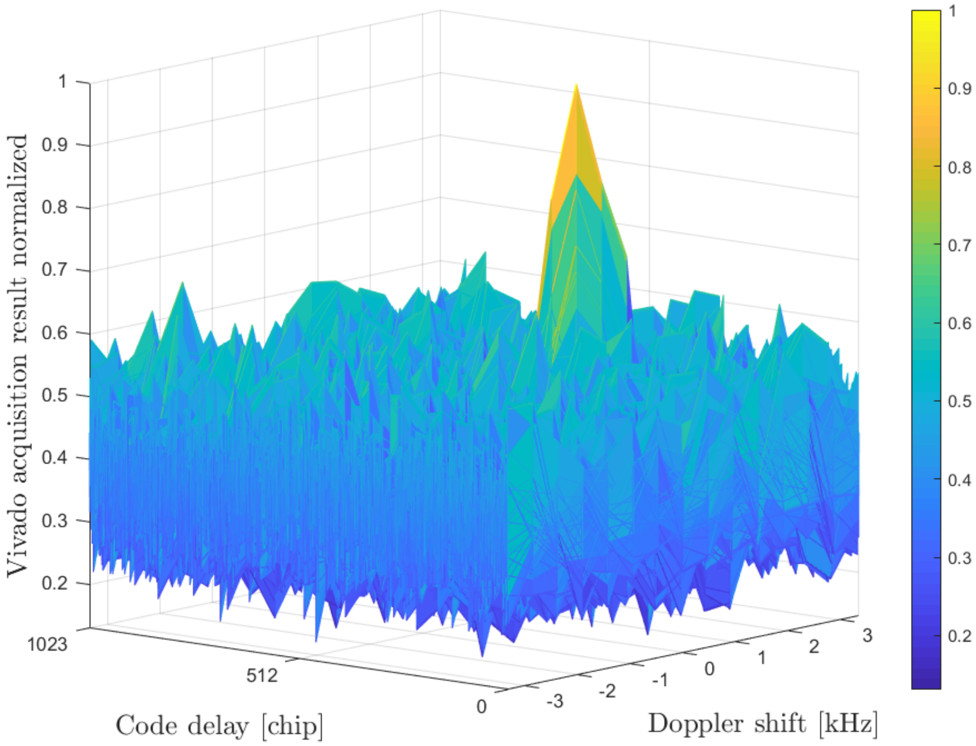

Figure 6.3: Acquisition result of the acquisition channel implemented on Vivado for the detection of the satellite with the PRN code 18, experiencing a code delay of 324 chips and a Doppler shift of 500 Hz. The used parameters were $f_{d,step} = 477$ Hz and $K = 10$. The peak is positioned in $f_d = 477$ Hz and $\tau = 323.5$ chips

Second, the MAX2769C chip is a GNSS specific RF front end targeted for GPS signal processing while the HackRF one was not. Moreover, the data for this test required external resampling and interpolation which is not as efficient as the direct processing of the MAX2769C. A better SNR at its output can be awaited, increasing the confidence in the obtained result.

In addition, multiple parameters can be tuned to obtain an higher SNR, e.g. the number of non-coherent integrations which was here constrained by the simulation time of the design. Improvements paths will be introduced on the conclusion of this master thesis.

Chapter 7

Conclusion

In this master thesis, a study of the GPS L1 C/A signal acquisition has been conducted for the development of a low-Earth orbit GPS receiver using a FPGA-based software defined radio. This led to the implementation of an acquisition channel in VHDL. It ended with a conclusive test on captured GNSS data.

First, the general context has been introduced with a description of the GNSS technology and more precisely of the GPS L1 C/A signal. The modulation scheme and the property of the employed PRN codes have been discussed.

Afterwards, the effects of deploying a GPS receiver in low-Earth orbit have been developed. Indeed, a first analysis has shown that LEO satellites are subject to a Doppler shift of maximum 46 KHz which is ten times higher than for ground receivers. Lastly, the minimum received power on this orbit has been computed to be one dB lower.

Next, the acquisition system, responsible for the satellites detection, has been detailed. The inherent concepts have been introduced such as the acquisition unknowns and the search space. Based on this, the acquisition principle has been explained with a theoretical derivation of its output expression. Three algorithms implementing the acquisition have been detailed and SNR improvements methods were mathematically derived. Finally, FPGA-based SDRs have been discussed with their advantages over different alternatives.

The hardware used for the GPS receiver has then be detailed with its parameters and configurations. From it, the acquisition channel input SNR has been quantified and a theoretical table has been proposed, expressing its output value based on the acquisition parameters.

To validate the feasibility of a FPGA-based GPS receiver, an implementation of a GPS acquisition channel has been made. The general architecture of the channel and its interface have been provided. Its different building blocks were presented with their working principle and specificities.

In the end, the obtained implementation was tested on real GNSS data captured on the ground. The result of the simulation were conclusive, the acquisition channel correctly identifying the Doppler shift and code delay of the satellite with a SNR of 9.9 dB. Figure 6.3 shows this important result of the master thesis.

Multiples improvements could be made to the design to increase its performance and sensitivity :

- A first improvement path would be the implementation of coherent integrations in the acquisition channel, for which the theory has been developed in this master thesis. As for the non-coherent integrations that have been implemented, it would require the addition of a memory to retain the result between integrations. This price to pay in term of resources can provide a significant increase of the output SNR.
- Secondly, a sampling frequency chosen to avoid the usage of zero padding in the processing chain would allow to keep intact the auto-correlation property of the PRN code. For this, an interpolator should be added to resample the generated PRN replica at the new sampling frequency.

Appendices

Appendix A

Mathematical development

A.1 Downconversion expression

This first part gathers the development of the Chapter 3 regarding the downconversion of the GPS L1 C/A signal from its carrier to the intermediate frequency and from this last one to baseband.

A.1.1 Downconversion from carrier to intermediate frequency

The incoming signal at the input of the RF front end can be written as follows:

$$r_{in}(t) = D(t-\tau) * x(t-\tau) * cos(2\pi(f_0 + f_d)t + \theta_0) \tag{A.1}$$

- τ is the delay due to signal propagation.
- $D(t)$ is the navigation message.
- $x(t)$ is PRN code.
- θ_0 is the phase shift accumulated during signal propagation.
- f_d is the Doppler shift experienced by the signal.
- f_0 is the carrier frequency of the GPS L1 C/A signal, 1575.42 MHz

To perform the downconversion, the RF front end first uses local oscillators to create two high frequency signals expressed below, with f_{IF} the intermediate frequency and θ_r the phase shift of the RFFE local oscillators.

$$r_{cos}(t) = cos(2\pi(f_0 - f_{IF})t + \theta_r) \tag{A.2}$$
$$r_{sin}(t) = sin(2\pi(f_0 - f_{IF})t + \theta_r) \tag{A.3}$$

Mixers are then used to multiply those signals with the input data. Afterwards, a low-pass filter of transfer function h_{LPF} rejects the high frequency content. The development is provided below.

$$\begin{aligned}
h_I(t) &= h_{LPF}(t) \circledast [r_{in}(t) * r_{cos}(t)] \\
&= h_{LPF}(t) \circledast [D(t-\tau) * x(t-\tau) * cos(2\pi(f_0 + f_d)t + \theta_0) \\
&\qquad\qquad\qquad\qquad\qquad\qquad * cos(2\pi(f_0 - f_{IF})t + \theta_r] \\
&= h_{LPF}(t) \circledast [D(t-\tau) * x(t-\tau) * \frac{1}{2} * (cos(2\pi(2f_0 - f_{IF} + f_d)t + \theta_0 + \theta_r) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad + cos(2\pi(f_{IF} + f_d)t + \theta_0 - \theta_r)] \\
&= D(t-\tau) * x(t-\tau) * \frac{1}{2} * cos(2\pi(f_{IF} + f_d)t + \theta_0 - \theta_r)
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
h_Q(t) &= h_{LPF}(t) \circledast [r_{in}(t) * r_{sin}(t)] \\
&= h_{LPF}(t) \circledast [D(t-\tau) * x(t-\tau) * cos(2\pi(f_0 + f_d)t + \theta_0) \\
&\qquad\qquad\qquad\qquad\qquad\qquad * sin(2\pi(f_0 - f_{IF})t + \theta_r)] \\
&= h_{LPF}(t) \circledast [D(t-\tau) * x(t-\tau) * \frac{1}{2} * (sin(2\pi(2f_0 - f_{IF} + f_d)t + \theta_0 + \theta_r) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad - sin(2\pi(f_{IF} + f_d)t + \theta_0 - \theta_r)] \\
&= -D(t-\tau) * x(t-\tau) * \frac{1}{2} * sin(2\pi(f_{IF} + f_d)t + \theta_0 - \theta_r)
\end{aligned} \tag{A.5}$$

A.1.2 Doppler wipeoff using IQ demodulation

Here are the developed the expressions for the output of the Doppler wipeoff. It is the downconversion from the intermediate frequency to baseband considering a Doppler shift trial $\hat{f}_d$. It builds on the same principle as Appendix A.1.1 by mixing the incoming signals with a cosine wave and applying a low-pass filter. Figure A.1 depicts this process.

The following notations are used :

- $\hat{f}_d$ is the trial Doppler shift used for Doppler wipeoff.
- $\hat{\theta}$ is the phase shift of the local oscillator used by the Doppler wipeoff.
- $\Delta f_d = f_d - \hat{f}_d$ is the remaining error at the output and thus the residual frequency that will be propagated to the next steps of the acquisition channel.

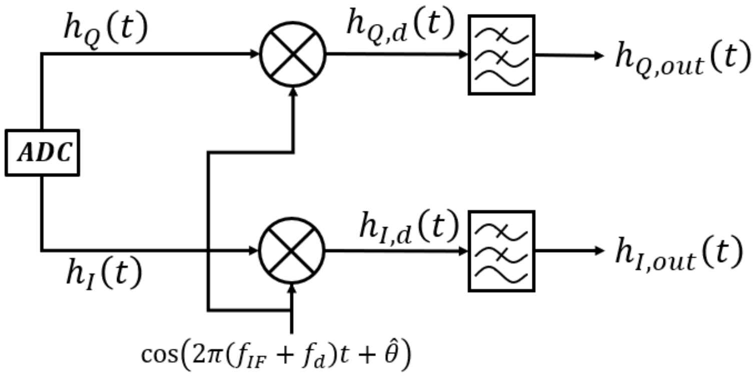

Figure A.1: Doppler wipeoff using IQ demodulation

- $\Delta\theta = \theta - \hat{\theta}$ is the phase shift between the samples and the local oscillator used by the Doppler wipeoff.

$$\begin{aligned} h_{I,d}(t) &= h_I(t) * cos(2\pi(f_{IF} + \hat{f}_d)t + \hat{\theta}) \\ &= D(t-\tau) * x(t-\tau) * cos(2\pi(f_{IF} + f_d)t + \theta) * cos(2\pi(f_{IF} + \hat{f}_d)t + \hat{\theta}) \\ &= D(t-\tau) * x(t-\tau) * \frac{1}{2} * [cos(2\pi(2 * f_{IF} + f_d + \hat{f}_d)t + \theta + \hat{\theta}) \\ &\qquad + cos(2\pi(f_d - \hat{f}_d)t + \theta - \hat{\theta})] \end{aligned} \tag{A.6}$$

$$\begin{aligned} h_{I,out}(t) &= h_{LPF}(t) \circledast h_{I,d}(t) \\ &= D(t-\tau) * x(t-\tau) * \frac{1}{2} * cos(2\pi(f_d - \hat{f}_d)t + \theta - \hat{\theta}) \end{aligned} \tag{A.7}$$

$$\begin{aligned} h_{Q,d}(t) &= h_Q(t) * cos(2\pi(f_{IF} + \hat{f}_d)t + \hat{\theta}) \\ &= -D(t-\tau) * x(t-\tau) * sin(2\pi(f_{IF} + f_d)t + \theta) * cos(2\pi(f_{IF} + \hat{f}_d)t + \hat{\theta}) \\ &= -D(t-\tau) * x(t-\tau) * \frac{1}{2} * [sin(2\pi(2 * f_{IF} + f_d + \hat{f}_d)t + \theta + \hat{\theta}) \\ &\qquad + sin(2\pi(f_d - \hat{f}_d)t + \theta - \hat{\theta})] \end{aligned} \tag{A.8}$$

$$\begin{aligned} h_{Q,out}(t) &= h_{LPF}(t) \circledast h_{Q,d}(t) \\ &= -D(t-\tau) * x(t-\tau) * \frac{1}{2} * sin(2\pi(f_d - \hat{f}_d)t + \theta - \hat{\theta}) \end{aligned} \tag{A.9}$$

A.2 Correlation expression derivation

In this part, the different expressions regarding the correlation will be developed, both for its first mathematical derivation introduced in Section 3.5 and for the PCPS algorithm presented in Section 3.6.3.

A.2.1 Correlation expression for correct code delay trial

Here, the correlation is restricted to the case where the correct code delay is used to perform the correlation, only leaving a dependency on the residual Doppler shift Δf_d. First, the general output of the correlation is expressed below when its modulus is taken. $\hat{\tau}$ and $\Delta\tau = \tau - \hat{\tau}$ are the trial and residual code delay.

$$|h_{corr}(\Delta\tau, \Delta f_d)| = \left| \int_0^{T_{CO}} x(t-\hat{\tau}) * x(t-\tau) * e^{-j2\pi\Delta f_d t}\, dt \right| \tag{A.10}$$

As explained above, the correct code delay is assumed for the correlation here, thus $\Delta\tau = 0$. The PRN code $x(t)$ is a sequence of "+1" and "-1". When multiplied with itself, the result is a constant value of "+1". This allow to perform the development below.

$$\begin{aligned}
|h_{corr}(\Delta\tau = 0, \Delta f_d)|^2 &= \left| \int_0^{T_{CO}} x^2(t) * e^{-j2\pi\Delta f_d t}\, dt \right|^2 \\
&= \left| e^{-j2\pi\Delta f_d \frac{T_{CO}}{2}} \int_{-\frac{T_{CO}}{2}}^{\frac{T_{CO}}{2}} e^{-j2\pi\Delta f_d t}\, dt \right|^2 \\
&= \left| \frac{-1}{2\pi j \Delta f_d} * \left[e^{-j2\pi\Delta f_d t} \right]_{\frac{T_{CO}}{2}}^{-\frac{T_{CO}}{2}} \right|^2 \\
&= \left| \frac{T_{CO}}{\pi\Delta f_d T_{CO}} * \frac{e^{j\pi\Delta f_d T_{CO}} - e^{-j\pi\Delta f_d T_{CO}}}{2j} \right|^2 \\
&= T_{CO}^2 * \left| \frac{sin(\pi\Delta f_d T_{CO})}{\pi\Delta f_d T_{CO}} \right|^2 \\
&= T_{CO}^2 * |sinc(\pi\Delta f_d T_{CO})|^2
\end{aligned} \tag{A.11}$$

A.2.2 Parallel code phase search acquisition

In Section 3.6.3, the PCPS algorithm was introduced. It uses a frequency domain multiplication to perform the correlation. The algorithm is depicted on Figure A.2.

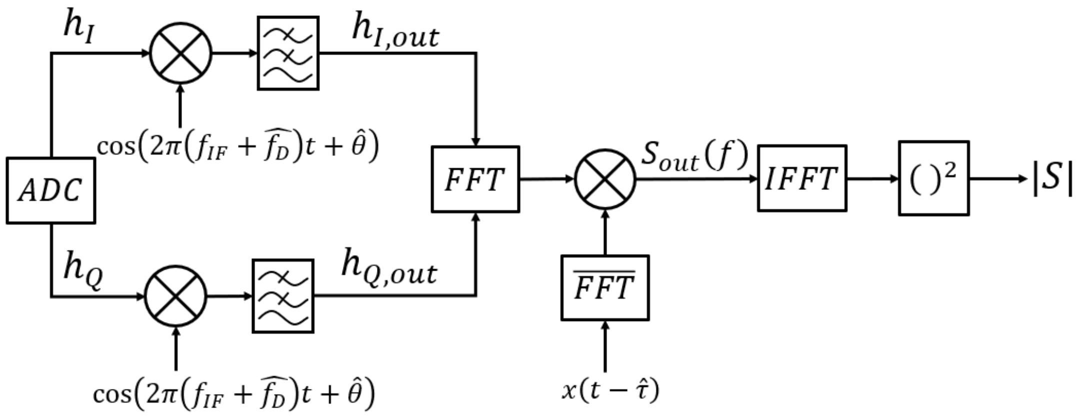

Figure A.2: Parallel code phase search acquisition algorithm

First, the complex signal formed by the output of the Doppler wipeoff, $h_{I,out}$ and $h_{Q,out}$, is calculated. See Appendix A.1.2 for the expressions of those signals. To be noted, the navigation message and amplitude factor $\frac{1}{2}$ have here been neglected for simplicity. The navigation message being at a rate of 50 bps, a bit transition can appear only every 20 PRN codes. Its effect is only major when discussing coherent integrations.

$$\begin{aligned} h_{out}(t) &= h_{I,out}(t) - j * h_{Q,out}(t) \\ &= x(t-\tau) * e^{j2\pi\Delta f_d t} * e^{j\Delta\theta} \end{aligned} \tag{A.12}$$

The Fourier Transform of the signal is then derived as follows, with $X(\omega)$ the Fourier transform of $x(t)$:

$$\begin{aligned} H_{out}(\omega) =& X(\omega - 2\pi\Delta f_d) * e^{-j(\omega-2\pi\Delta f_d)\tau} * e^{j\Delta\theta} \\ =& X(\omega - 2\pi\Delta f_d) * e^{-j\omega\tau} * e^{j2\pi\Delta f_d\tau} * e^{j\Delta\theta} \end{aligned} \tag{A.13}$$

Next, the result is then multiplied by the complex conjugate of $X(\omega)$, the Fourier transform of the PRN replica.

$$S_{out}(\omega) = X(\omega - 2\pi\Delta f_d) * \overline{X(\omega)} * e^{-j\omega\tau} * e^{j2\pi\Delta f_d\tau} * e^{j\Delta\theta} \tag{A.14}$$

The conversion back to time domain using the inverse Fourier transform and the modulus operation are then performed below.

$$
\begin{aligned}
S(\hat{\tau}) =& \int_{-\infty}^{\infty} X(\omega - 2\pi\Delta f_d) * \overline{X(\omega)} * e^{-j\omega\tau} * e^{j2\pi\Delta f_d\tau} * e^{j\Delta\theta} * e^{j\omega\hat{\tau}}\, d\omega \\
=& e^{j(2\pi\Delta f_d\tau + \Delta\theta)} * \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(t) e^{j\omega t}\, dt \right) X(\omega - 2\pi\Delta f_d) * e^{-j\omega\tau} * e^{j\omega\hat{\tau}}\, d\omega \\
=& e^{j(2\pi\Delta f_d\tau + \Delta\theta)} * \int_{-\infty}^{\infty} x(t) \left(\int_{-\infty}^{\infty} X(\omega - 2\pi\Delta f_d) * e^{j\omega(\hat{\tau}+t-\tau)}\, d\omega \right) dt \\
=& e^{j(2\pi\Delta f_d\tau + \Delta\theta)} * \int_{-\infty}^{\infty} x(t) * e^{j2\pi\Delta f_d(\hat{\tau}+t-\tau)} * x(\hat{\tau} + t - \tau)\, dt \\
=& e^{j(2\pi\Delta f_d t + \hat{\tau})} * \int_{0}^{T_{CO}} x(t) * x(\hat{\tau} + t - \tau) * e^{j2\pi\Delta f_d t}\, dt \\
|S(t)|^2 =& \left| \int_{0}^{T_{CO}} x(t) * x(\hat{\tau} + t - \tau) * e^{j2\pi\Delta f_d t}\, dt \right|^2
\end{aligned}
\tag{A.15}
$$

The Fourier Transform of the signal is then derived as follows, with $X(\omega)$ the Fourier transform of $x(t)$:

$$
\begin{aligned}
H_{out}(\omega) =& X(\omega - 2\pi\Delta f_d) * e^{-j(\omega - 2\pi\Delta f_d)\tau} * e^{j\Delta\theta} \\
=& X(\omega - 2\pi\Delta f_d) * e^{-j\omega\tau} * e^{j2\pi\Delta f_d\tau} * e^{j\Delta\theta}
\end{aligned}
\tag{A.16}
$$

Next, the result is then multiplied by the complex conjugate of $X(\omega)$, the Fourier transform of the PRN replica.

$$
S_{out}(\omega) = X(\omega - 2\pi\Delta f_d) * \overline{X(\omega)} * e^{-j\omega\tau} * e^{j2\pi\Delta f_d\tau} * e^{j\Delta\theta}
\tag{A.17}
$$

The conversion back to time domain using the inverse Fourier transform and the modulus operation are then performed below.

$$
\begin{aligned}
S(\hat{\tau}) =& \int_{-\infty}^{\infty} X(\omega - 2\pi\Delta f_d) * \overline{X(\omega)} * e^{-j\omega\tau} * e^{j2\pi\Delta f_d\tau} * e^{j\Delta\theta} * e^{j\omega\hat{\tau}}\, d\omega \\
=& e^{j(2\pi\Delta f_d\tau + \Delta\theta)} * \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(t) e^{j\omega t}\, dt \right) X(\omega - 2\pi\Delta f_d) * e^{-j\omega\tau} * e^{j\omega\hat{\tau}}\, d\omega \\
=& e^{j(2\pi\Delta f_d\tau + \Delta\theta)} * \int_{-\infty}^{\infty} x(t) \left(\int_{-\infty}^{\infty} X(\omega - 2\pi\Delta f_d) * e^{j\omega(\hat{\tau}+t-\tau)}\, d\omega \right) dt \\
=& e^{j(2\pi\Delta f_d\tau + \Delta\theta)} * \int_{-\infty}^{\infty} x(t) * e^{j2\pi\Delta f_d(\hat{\tau}+t-\tau)} * x(\hat{\tau} + t - \tau)\, dt \\
=& e^{j(2\pi\Delta f_d t + \Delta\theta)} * \int_{0}^{T_{CO}} x(t) * x(\hat{\tau} + t - \tau) * e^{j2\pi\Delta f_d t}\, dt \\
|S(t)|^2 =& \left| \int_{0}^{T_{CO}} x(t) * x(\hat{\tau} + t - \tau) * e^{j2\pi\Delta f_d t}\, dt \right|^2
\end{aligned}
\tag{A.18}
$$

Appendix B

Hack-RF one configuration

```
[GNSS-SDR]

;######### GLOBAL OPTIONS ##################
GNSS-SDR.internal_fs_sps=10000000

;######### SIGNAL_SOURCE CONFIG ############
SignalSource.implementation=Osmosdr_Signal_Source
SignalSource.item_type=gr_complex
SignalSource.sampling_frequency=10000000
SignalSource.freq=1575420000
SignalSource.gain=60
SignalSource.rf_gain=40
SignalSource.if_gain=40
SignalSource.AGC_enabled=true
SignalSource.samples=0
SignalSource.repeat=false
;# Next line enables the internal HackRF One bias (3.3 VDC)
SignalSource.osmosdr_args=hackrf,bias=1
SignalSource.enable_throttle_control=false
SignalSource.dump=false
SignalSource.dump_filename=./signal_source.dat

;######### SIGNAL_CONDITIONER CONFIG ############
SignalConditioner.implementation=Signal_Conditioner

;######### DATA_TYPE_ADAPTER CONFIG ############
DataTypeAdapter.implementation=Pass_Through
```

```
;######### INPUT_FILTER CONFIG ############
InputFilter.implementation=Freq_Xlating_Fir_Filter
InputFilter.decimation_factor=1
InputFilter.input_item_type=gr_complex
InputFilter.output_item_type=gr_complex
InputFilter.taps_item_type=float
InputFilter.number_of_taps=5
InputFilter.number_of_bands=2
InputFilter.band1_begin=0.0
InputFilter.band1_end=0.85
InputFilter.band2_begin=0.9
InputFilter.band2_end=1.0
InputFilter.ampl1_begin=1.0
InputFilter.ampl1_end=1.0
InputFilter.ampl2_begin=0.0
InputFilter.ampl2_end=0.0
InputFilter.band1_error=1.0
InputFilter.band2_error=1.0
InputFilter.filter_type=bandpass
InputFilter.grid_density=16
InputFilter.dump=true
InputFilter.dump_filename=../data/input_filter.dat

;######### RESAMPLER CONFIG ############
Resampler.implementation=Pass_Through

;######### CHANNELS GLOBAL CONFIG ############
Channels_1C.count=8
Channels.in_acquisition=1

;######### ACQUISITION GLOBAL CONFIG ############
Acquisition_1C.implementation=GPS_L1_CA_PCPS_Acquisition
Acquisition_1C.item_type=gr_complex
Acquisition_1C.coherent_integration_time_ms=1
Acquisition_1C.pfa=0.01
Acquisition_1C.doppler_max=5000
Acquisition_1C.doppler_step=250
Acquisition_1C.max_dwells=1
Acquisition_1C.dump=true
Acquisition_1C.dump_filename=./acq_dump.dat
```

```
;######### TRACKING GLOBAL CONFIG ############
Tracking_1C.implementation=GPS_L1_CA_DLL_PLL_Tracking
Tracking_1C.item_type=gr_complex
Tracking_1C.extend_correlation_symbols=10
Tracking_1C.early_late_space_chips=0.5
Tracking_1C.early_late_space_narrow_chips=0.15
Tracking_1C.pll_bw_hz=40
Tracking_1C.dll_bw_hz=2.0
Tracking_1C.pll_bw_narrow_hz=5.0
Tracking_1C.dll_bw_narrow_hz=1.50
Tracking_1C.fll_bw_hz=10
Tracking_1C.enable_fll_pull_in=true
Tracking_1C.enable_fll_steady_state=false
Tracking_1C.dump=false
Tracking_1C.dump_filename=tracking_ch_

;######### TELEMETRY DECODER GPS CONFIG ############
TelemetryDecoder_1C.implementation=GPS_L1_CA_Telemetry_Decoder
TelemetryDecoder_1C.dump=false

;######### OBSERVABLES CONFIG ############
Observables.implementation=Hybrid_Observables
Observables.dump=false
Observables.dump_filename=./observables.dat

;######### PVT CONFIG ############
PVT.implementation=RTKLIB_PVT
PVT.positioning_mode=Single
PVT.output_rate_ms=100
PVT.display_rate_ms=500
PVT.iono_model=Broadcast
PVT.trop_model=Saastamoinen
PVT.flag_rtcm_server=true
PVT.flag_rtcm_tty_port=false
PVT.rtcm_dump_devname=/dev/pts/1
PVT.rtcm_tcp_port=2101
PVT.rtcm_MT1019_rate_ms=5000
PVT.rtcm_MT1077_rate_ms=1000
PVT.rinex_version=2
```

```
PVT.enable_monitor=true
PVT.enable_protobuf=true
PVT.monitor_client_addresses=127.0.0.1
PVT.monitor_udp_port=1112

Monitor.enable_monitor=true
PVT.enable_protobuf=true
Monitor.decimation_factor=1
Monitor.client_addresses=127.0.0.1
Monitor.udp_port=1111


GNSS-SDR.telecommand_enabled=true
GNSS-SDR.telecommand_tcp_port=3333
```

Bibliography

[1] *Official U.S. government information about the Global Positioning System (GPS) and related topics*. July 2021. URL: `https://www.gps.gov/systems/gps/`.

[2] *Official Aerospacelab Website*. URL: `https://www.aerospacelab.be/`.

[3] *European Space Agency Website : GNSS signal - Navipedia*. 2014. URL: `https://gssc.esa.int/navipedia/index.php/GNSS_signal`.

[4] Jan Van Sickle. *GPS for land surveyors*. 2008.

[5] Ghafour Amouzad and Ahmad Fauzi. "Advanced Modulation Formats and Multiplexing Techniques for Optical Telecommunication Systems". In: *Trends in Telecommunications Technologies*. Ed. by Christos J. InTech, Mar. 2010.

[6] Kota Solomon Raju. "Digital GPS Signal Generator for L1 Band". In: *Signal & Image Processing : An International Journal* 3.6 (Dec. 2012), pp. 75–86.

[7] Anthony Flores. *IS-GPS-200 : NAVSTAR GPS Space Segment/Navigation User Interfaces*. May 2021.

[8] Peter Steigenberger, Steffen Thoelert, and Oliver Montenbruck. "GNSS satellite transmit power and its impact on orbit determination". In: *Journal of Geodesy* 92.6 (June 2018), pp. 609–624.

[9] Per Enge and Frank van Diggelen. *GPS: An Introduction to Satellite Navigation, with an interactive Worldwide Laboratory using Smartphones*. Stanford University. URL: `https://scpnt.stanford.edu/about/gps-mooc-massive-open-online-course`.

[10] John W. Betz. *Engineering satellite-based navigation and timing: global navigaton satellite systems, signals, and receivers*. Wiley-IEEE Press, 2016.

[11] Pratap Misra and Per Enge. *Global positioning system: signals, measurements, and performance*. 2. ed. Ganga-Jamuna Press, 2006.

[12] Rafiullah Khan et al. "Acquisition strategies of GNSS receiver". In: *International Conference on Computer Networks and Information Technology*. IEEE, July 2011, pp. 119–124.

[13] Maxim Integrated. *Universal GNSS Receiver : MAX2769C*. Tech. rep. Maxim Integrated Products, Inc. URL: https://datasheets.maximintegrated.com/en/ds/MAX2769C.pdf.

[14] Peter J.G. Teunissen and Oliver Montenbruck, eds. *Springer Handbook of Global Navigation Satellite Systems*. Springer International Publishing, 2017.

[15] Myriam Foucras et al. "Assessing the Performance of GNSS Signal Acquisition: New Signals and GPS L1 C/A Code". In: *InsideGNSS* (2014).

[16] Xilinx. *7 Series FPGAs Data Sheet: Overview (DS180)*. Tech. rep. 2020. URL: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf.

[17] Shkelzen Cakaj et al. "Antenna Noise Temperature for Low Earth Orbiting Satellite Ground Stations at L and S Band". In: vol. Conference proceedings. Apr. 2011, pp. 1–6.

[18] Arm. *AMBA 4 AXI4-Stream Protocol Specification*. URL: https://developer.arm.com/documentation/ihi0051/a/Introduction/About-the-AXI4-Stream-protocol (visited on 08/18/2021).

[19] Xilinx. *AXI4-Stream Video IP and System Design Guide (UG934)*. Tech. rep. 2019. URL: https://www.xilinx.com/support/documentation/ip_documentation/axi_videoip/v1_0/ug934_axi_videoIP.pdf.

[20] Xilinx. *CORDIC v6.0 LogiCORE IP Product Guide*. Tech. rep. 2021. URL: https://www.xilinx.com/support/documentation/ip_documentation/cordic/v6_0/pg105-cordic.pdf.

[21] Xilinx. *Fast Fourier Transform v9.1 LogiCORE IP Product Guide*. Tech. rep. 2021. URL: https://www.xilinx.com/support/documentation/ip_documentation/xfft/v9_1/pg109-xfft.pdf.

[22] Great Scott Gadgets. *HackRF One*. Tech. rep. URL: https://greatscottgadgets.com/hackrf/one/.

[23] Mikroe. *GPS Active External Antenna Specification*. URL: https://download.mikroe.com/documents/datasheets/GPS-Active-External-Antenna.pdf.

[24] Nooelec. *HackRF One - 'Tiny TCXO' 10MHz 0.5PPM TCXO Module*. Tech. rep. URL: https://www.nooelec.com/store/tiny-tcxo.html.

[25] Carles Fernández-Prades. *GNSS-SDR documentation*. URL: https://gnss-sdr.org/docs/.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**