Analysis and Development of Algorithms for Fast Acquisition of Modern GNSS Signals

Pedro Afonso Marques Boto

Thesis to obtain the Master of Science Degree in Aerospace Engineering

Supervisor(s): Professor Fernando Nunes and Eng. João Silva

Examination Committee

Chairperson: Prof. João Manuel Lage de Miranda Lemos
Supervisor: Eng. João Silva
Member of the Committee: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino

December 2014
Dedicated to my family
Acknowledgments

I would like to start by thanking everyone that in a way or another helped finish my master's degree, to them my deepest thanks.

I would like to thank Eng. João Silva from Deimos, without him this thesis could not be made and his support and contributions were fundamental to the work made. The same could be said about Professor Fernando Nunes, his knowledge, availability and guidance were fundamental to finish this thesis.

Outside the scholar environment I would like to thank my parents for supporting the “extra” years studying and all their patience. My sister for constantly annoying me. My grandparents for all their words. My friends for the many good moments. My Caroline for everything.
Resumo

O objectivo deste trabalho é a análise de algoritmos para a aquisição de sinais GNSS com a finalidade de implementação no receptor DEIMOS’ GRIP.

Em primeiro lugar é feita uma descrição das novas modulações GNSS, AltBOC(15,10) e CBOC(6,1,1/11). Foi feita a seguir uma descrição dos vários métodos de busca e de processamento do sinal na fase de aquisição de modo a perceber as vantagens e desvantagens dos vários métodos.

De seguida são descritos seis algoritmos de aquisição do sinal CBOC(6,1,1/11) com vista a análise completa de três e uma referência. Os algoritmos escolhidos foram o Double Block Zero Padding Transition Insensitive, Delay and Multiply, Dual Sideband with CBOC as BPSK e a referência é um serial search com matched filter.

Todo o estudo estatístico dos algoritmos é feito, teoricamente e com sinais simulados. Por último é analisado o desempenho dos vários algoritmos com um sinal real.

Palavras-chave: GNSS, Galileo, CBOC, AltBOC, aquisição, DBZPTI, delay and multiply, Dual Sideband
Abstract

The objective of this work is the analysis of various algorithms for the acquisition of GNSS signals taking in account a future implementation of the best one on the DEIMOS’ GRIP receiver.

In first place a description of the new GNSS modulations, AltBOC(15,10) e CBOC(6,1,1/11) is made. Next a full description of the various methods of search and processing of the signal on the acquisition phase is made. With this description it is easier to understand the advantages and disadvantages of all the algorithms presented next.

Six acquisition algorithms of the CBOC(6,1,1/11) are described. After this description three algorithms and a reference are chosen for a full performance analysis. The chosen algorithms are the Double Block Zero Padding Transition Insensitive, Delay and Multiply, Dual Sideband with CBOC as BPSK and the reference is a serial search with matched filter.

All the statistic study is made theoretically and with simulated signals. Finally the performance of the various algorithms is analyzed with a real signal from the receptor.

Keywords: GNSS, Galileo, CBOC, AltBOC, Acquisition, DBZPTI, Delay and Multiply, Dual Sideband
Contents

Acknowledgments ........................................................................................................... v
Resumo .......................................................................................................................... vii
Abstract ......................................................................................................................... ix
List of Tables .................................................................................................................... xv
List of Figures .................................................................................................................. xviii
Glossary ............................................................................................................................ xx

1 Introduction 1
1.1 Motivation .................................................................................................................. 1
1.2 Thesis Objectives ........................................................................................................ 1
1.3 Thesis Contributions ................................................................................................. 2
1.4 Thesis Outline ............................................................................................................ 2

2 GNSS Signals 5
2.1 GPS Signal .................................................................................................................. 5
2.1.1 GPS Signal Overview ........................................................................................... 5
2.1.2 GPS L1 Civil Signal ........................................................................................... 7
2.1.3 Auto-Correlation and Random Sequences ......................................................... 8
2.2 Modulation Types for Satellite Navigation ............................................................... 10
2.2.1 Binary Phase Shift Keying (BPSK) .................................................................... 10
2.2.2 Direct Sequence Spread Spectrum (DSSS) ....................................................... 11
2.2.3 Binary Offset Carrier Signals (BOC) ............................................................... 12
2.3 Galileo Navigation System ....................................................................................... 14
2.3.1 Galileo E1 Band .................................................................................................. 15
2.3.2 Galileo E6 Band .................................................................................................. 16
2.3.3 Galileo E5 Band .................................................................................................. 16
2.4 Galileo Modulations ................................................................................................. 17
2.4.1 CBOC(6,1,1/11) ................................................................................................. 17
2.4.2 AltBOC(15,10) ................................................................................................. 20

3 GNSS Acquisition Techniques and Concepts 25
3.1 Basic Concepts .......................................................................................................... 25
3.1.1 Baseband Signal Processing ............................................. 25
3.1.2 Search Space .............................................................. 28
3.1.3 Detection and False Alarm Probabilities ............................. 29
3.2 Search Strategies ............................................................. 30
  3.2.1 Serial Search ............................................................. 30
  3.2.2 Parallel Search .......................................................... 30
  3.2.3 Assisted Search .......................................................... 31
3.3 Signal Correlation Techniques ........................................... 32
  3.3.1 Matched Filter .......................................................... 32
  3.3.2 Sub-Carrier Demodulation Simplifications .......................... 32
3.4 Detector algorithms ......................................................... 33
  3.4.1 Coherent Combining .................................................... 33
  3.4.2 Non-coherent Combining .............................................. 34
  3.4.3 Differentially Coherent Combining ................................ 36
  3.4.4 Detectors Comparison .................................................. 37
3.5 AltBOC(15,10) Acquisition ............................................... 38
  3.5.1 Single sideband acquisition (SSB) .................................. 38
  3.5.2 Double sideband acquisition (DSB) ................................ 38
  3.5.3 Full-band independent code acquisition (FIC) ..................... 39
  3.5.4 Direct AltBOC method ................................................ 39
  3.5.5 AltBOC search schemes comparison ................................ 39
  3.5.6 Hand-over from the E1 ............................................... 40

4 Acquisition Algorithms Pre-selection .................................... 41
  4.1 Classical Acquisition (Serial Search with Matched Filter) ........ 41
  4.2 Parallel Frequency Space Search ....................................... 42
  4.3 Parallel Code Phase Search ............................................. 43
  4.4 Double Block Zero Padding Transition Insensitive (DBZPTI) ...... 44
  4.5 Dual Sideband, CBOC as BPSK ........................................ 51
  4.6 Delay and Multiply ...................................................... 53
  4.7 Selected Algorithms ..................................................... 54

5 Performance Analysis ....................................................... 55
  5.1 Simulation Parameters .................................................. 55
    5.1.1 Carrier-to-Noise Density Ratio ................................... 55
    5.1.2 Probabilities of False Alarm ....................................... 55
    5.1.3 Integration Time and Doppler Step ................................ 56
    5.1.4 Mean Acquisition Time, MAT ...................................... 56
    5.1.5 Incoming Signal ...................................................... 56
  5.2 Classical acquisition (Serial Search with Matched Filter) ......... 57
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1 Theoretical Performance Evaluation</td>
<td>57</td>
</tr>
<tr>
<td>5.2.2 Simulation Results</td>
<td>58</td>
</tr>
<tr>
<td>5.2.3 Real Signals Simulation Results</td>
<td>60</td>
</tr>
<tr>
<td>5.3 DBZPTI</td>
<td>60</td>
</tr>
<tr>
<td>5.3.1 Theoretical Performance Evaluation</td>
<td>60</td>
</tr>
<tr>
<td>5.3.2 Simulation Results</td>
<td>63</td>
</tr>
<tr>
<td>5.3.3 Real Signals Simulation Results</td>
<td>64</td>
</tr>
<tr>
<td>5.4 Delay and Multiply</td>
<td>65</td>
</tr>
<tr>
<td>5.4.1 Delay and Multiply as a filtering method</td>
<td>65</td>
</tr>
<tr>
<td>5.4.2 Simulation Results</td>
<td>66</td>
</tr>
<tr>
<td>5.4.3 Real Signals Simulation Results</td>
<td>68</td>
</tr>
<tr>
<td>5.5 Dual Sideband, CBOC as BPSK</td>
<td>68</td>
</tr>
<tr>
<td>5.5.1 Theoretical Performance Evaluation</td>
<td>68</td>
</tr>
<tr>
<td>5.5.2 Simulation Results</td>
<td>69</td>
</tr>
<tr>
<td>5.5.3 Real signal Simulation Results</td>
<td>70</td>
</tr>
<tr>
<td>5.6 Results Comparison</td>
<td>71</td>
</tr>
<tr>
<td>6 Conclusions</td>
<td>73</td>
</tr>
<tr>
<td>6.1 Conclusions</td>
<td>73</td>
</tr>
<tr>
<td>6.2 Future Work</td>
<td>74</td>
</tr>
<tr>
<td>Bibliography</td>
<td>77</td>
</tr>
<tr>
<td>A Deimos Grip</td>
<td>79</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Summary of Signals Components .................................................. 5
2.2 GPS Frequency Bands ................................................................. 6
2.3 Galileo Frequency Bands. E5a and E5b signals are part of the E5 signal in its full bandwidth. 15
2.4 Galileo E1 Band Signal Characteristics ......................................... 15
2.5 Galileo E6 Band Signal Characteristics, [1] .................................. 16
2.6 Galileo E5 Band Signal Characteristics ......................................... 16

3.1 Values used to evaluate the different detectors ROC .......................... 37
3.2 Resource usage in AltBOC search based schemes ............................. 39

4.1 Classical Acquisition characteristics .............................................. 41
4.2 Parallel Frequency Acquisition characteristics ................................ 42
4.3 Parallel Code Phase Acquisition characteristics .............................. 43
4.4 DBZPTI Acquisition characteristics ............................................. 44
4.5 Dual Sideband, CBOC as BPSK Acquisition characteristics ............... 51
4.6 Classical Acquisition characteristics ............................................. 53

5.1 Parameters used for the Classical Acquisition ................................. 57
5.2 MAT for all the methods .............................................................. 71
5.3 Acquisition times for the DBZPTI and Parallel Code Phase Search ........ 71
5.4 Peak Magnitude for all the methods ............................................. 71
List of Figures

2.1 GPS signals, [2] ......................................................... 6
2.2 GPS L1 Power Spectra, [3] ........................................... 8
2.3 Auto-Correlation of a simple binary code. Blue line is a signal \( s_1(t) \), Red line is a time shift of that same signal, \( s_2(t - \tau) \), the purple line is the multiplication of both signals and the purple points are the graphic from the auto-correlation function, \( \int s_1(t)s_2(t - \tau) \) ............ 9
2.4 Auto-Correlation properties of the PRN that as the same length as the C/A codes (N=1023), [2] ......................................................... 10
2.5 BPSK Modulation, [4] .................................................. 11
2.6 DSSS Modulation, [4] .................................................. 11
2.7 Amplitude spectrum for the C/A-code and a BOC(1,1) ......................... 12
2.8 Auto-correlation function chip waveform for the C/A-code and a BOC(1,1) ............. 13
2.9 Galileo Constellation of Satellites .................................. 14
2.10 Galileo Frequency Plan [5]. ........................................... 14
2.11 Spectra of Galileo Signals in E1, [1] .................................. 15
2.12 Spectra of Galileo Signals in E6 ..................................... 16
2.13 Spectra of Galileo Signals in E5, [1] .................................. 17
2.14 CBOC(6,1,1/11) data and pilot component pulses ........................ 18
2.15 PSD of CBOC(6,1,1/11) Galileo E1 signal .......................... 19
2.16 Autocorrelation function of CBOC(6,1,1/11) pilot signal ................. 20
2.17 Data and Pilot AltBOC(15,10) subcarriers waveforms, [6]. ............ 21
2.18 PSD of the AltBOC(15,10) signal .................................. 22
2.19 Normalized AltBOC(15,10) ACF ..................................... 23
3.1 Inphase and quadrature sampling, carrier wipeoff .......................... 26
3.2 Baseband signal, carrier wipeoff ..................................... 27
3.3 Inphase and quadrature correlators. Code wipeoff. .......................... 27
3.4 Search Space for a GPS receiver. Signal search area covers Doppler frequency, \( \Delta f_D \) and code phase, \( \Delta \tau \), [4]. ......................................................... 28
3.5 PDF's for binary decision, [7]. ....................................... 30
3.6 ROC of the detection algorithms for \( C/N_0 = 33 \text{ dHz} \) .................. 38
4.1 Galileo E1 signal acquisition with coherent combining, "Classical acquisition" ........ 42
4.2 Block diagram of the Parallel Frequency Search .............................................. 43
4.3 Block diagram of the Parallel Code Phase Search Algorithm .............................. 45
4.4 DBZPTI Step 1 - Processing of the incoming signal. ........................................ 46
4.5 DBZPTI Step 2 - Generation of the local code. ............................................... 47
4.6 DBZPTI Step 3 - Partial Circular Correlation. .................................................. 48
4.7 DBZPTI Step 4 - Shift of the incoming signal. .................................................. 49
4.8 DBZPTI Step 5 - Application of the FFT. ......................................................... 50
4.9 Frequency shift of the CBOC signal ............................................................... 51
4.10 Dual Sideband Acquisition Diagram ........................................................... 52
4.11 Acquisition diagram of the Delay and Multiply method .................................... 53

5.1 Theoretical Performance of the Classical Acquisition Method for, $P_{FA}^{grid} = 0.01$ .... 58
5.2 Simulated PDF for the Classical Acquisition with a $C/N_0 = 44$ dBHz and $P_{fa}^{cell} = 1 \times 10^{-4}$ 59
5.3 Theoretical and Simulated Performance of the Classical Acquisition Method for, $P_{fa}^{cell} = 1 \times 10^{-4}$ ................................................................. 59
5.4 Classical Acquisition with the use of real signals ............................................ 60
5.5 DBZPTI Performance for $K = 1$ .................................................................. 62
5.6 E1 DBZPTI and Reference Theoretical Performance Comparison ......................... 62
5.7 Simulated PDF for the DBZPTI with a $C/N_0 = 44$ dBHz and $P_{fa}^{cell} = 1 \times 10^{-4}$ .... 63
5.8 Theoretical and Simulated Performance of the DBZPTI Acquisition Method for, $P_{fa}^{cell} = 1 \times 10^{-4}$ ................................................................. 63
5.9 DBZPTI acquisition with the use of real signals .............................................. 64
5.10 Output of the Delay and Multiply ................................................................ 65
5.11 Delay and Multiply as a code delay filtering method ...................................... 66
5.12 Cells searched in function of time with the Delay and Multiply filtering methodology for a $C/N_0 = 44$ dBHz ................................................................. 67
5.13 Cells searched in function of time with the Delay and Multiply filtering methodology for a $C/N_0 = 35$ dBHz ................................................................. 67
5.14 Delay and Multiply Real Signals Output ....................................................... 68
5.15 Classical Acquisition theoretical performance with the simulation results for the Dual Sideband, CBOC as BPSK .............................................................. 69
5.16 Empirical E1 Dual Sideband Performance for $P_{fa}^{cell} \approx 1.0611 \times 10^{-8}$ .......... 70
5.17 Dual Sideband, CBOC as BPSK Acquisition with the use of real signals .......... 70
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF</td>
<td>Auto Correlation Function</td>
</tr>
<tr>
<td>AltBOC</td>
<td>Alternative Binary Offset Carrier</td>
</tr>
<tr>
<td>BOC</td>
<td>Binary Offset Carrier</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary Phase Shift-Keying</td>
</tr>
<tr>
<td>C/A</td>
<td>Coarse/Acquisition</td>
</tr>
<tr>
<td>CASM</td>
<td>Coherent Adaptive Subcarrier Modulation</td>
</tr>
<tr>
<td>CBOC</td>
<td>Composite Binary Offset Carrier</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CS</td>
<td>Commercial Service</td>
</tr>
<tr>
<td>DBZPTI</td>
<td>Double Block Zero Padding Transition Insensitive</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DSP</td>
<td>Dual Sideband Processing</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transformation</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>IFT</td>
<td>Inverse Fourier Transform</td>
</tr>
<tr>
<td>MAT</td>
<td>Mean Acquisition Time</td>
</tr>
<tr>
<td>NRZ</td>
<td>Non-return-to-zero</td>
</tr>
<tr>
<td>OS</td>
<td>Open Service</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PRN</td>
<td>Pseudo Random Noise</td>
</tr>
<tr>
<td>PRS</td>
<td>Public Regulated Service</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
</tr>
<tr>
<td>RF</td>
<td>Radio frequency</td>
</tr>
<tr>
<td>RNCP</td>
<td>Right-hand Circularly Polarized</td>
</tr>
<tr>
<td>RNSS</td>
<td>Radio Navigation Satellite Systems</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristics</td>
</tr>
<tr>
<td>SSP</td>
<td>Single Sideband Processing</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------</td>
</tr>
<tr>
<td>SV</td>
<td>Satellite Vehicle</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Galileo is the European global navigation satellite system (GNSS). The main objective of this system is to provide a high precision positioning and at the same time lose the dependency of the Russian GLONASS and the American GPS. Since it is a new system, more complex signals will emerge but at the same time guaranteeing a better positioning. The more challenging signals will be present in bands E1 and E5 of the Galileo system, these are the AltBOC(15,10) and the CBOC(6,1,1/11).

Given the complexity of the Galileo signals the receiving systems need to be updated so they can deal with all this complexity. One of the major challenges is the acquisition time, because in the past the codes were shorter (GPS L1 C/A), now they are much longer (Galileo CBOC(6,1,1/11)) and the acquisition time has to be the same. So there is a need to study new acquisition algorithms, reinvent old ones and invent new ones, so the acquisition of the new signals can be as quick as the old ones.

1.2 Thesis Objectives

The main objective of this thesis is to find an acquisition algorithm that is fast and can be implemented on the Deimos Grip GNSS receptor (Annex A, [8]) and the full study of another algorithm that can be considered state-of-art in terms of speed and performance. Both of these algorithms are aimed for the Galileo E1 band, more precisely for the CBOC(6,1,1/11).

The main limitation of the Deimos receiver is the use of FFT’s, the technique to be implemented can not use FFT’s and has to keep the use of parallel correlators to a minimum, so the main objective is to study an algorithm that is fast despite all the limitations above.

In the future the Deimos receiver might be able to implement a technique that uses FFT’s so a reference was needed for when this time comes. Because of this fact the second purpose of this thesis is the full study of a state-of-the art acquisition algorithm.
1.3 Thesis Contributions

The main contributions of this thesis are:

- The study of the new Galileo signals, CBOC(6,1,1/11) and AltBOC(15,10) in terms of its Power Spectral Densities (PSD) and Auto Correlation Functions (ACF).
- A hand-over technique in which the Doppler frequency and code delay of the AltBOC(15,10) can be obtained from the acquisition of the CBOC(6,1,1/11).
- The full description of 6 acquisition algorithms, 3 using FFT’s and 3 without.
- The full study of 4 algorithms (Classical Acquisition, Delay and Multiply, Double Block Zero Padding Transition Insensitive and Dual Sideband CBOC as BPSK) in terms of:
  - Mean Acquisition Time, MAT.
  - Performance, probability of detection versus the carrier-to-noise ratio.
  - Performance with the use of simulated signals.
  - Performance with the use of a real signal similar to the ones that the receiver will process.
- The use of the Delay and Multiply as a filtering method.

1.4 Thesis Outline

The following thesis is structured as follows:

- First of all chapter 2 gives an introduction about the existing GNSS signals, with a special focus on GPS and Galileo. A description of the full navigation systems is made with emphasis on the transmitting bands, the mathematical description of the signal reaching the receiver and the modulation types used. The various modulations presented in both navigation systems are studied regarding their Power Spectral Densities and Auto Correlation Functions.
- Chapter 3 describes how the acquisition of GNSS signals is processed, starting with some basic concepts to more easily comprehend the acquisition algorithms presented later and the study made on these. Then there are described the different search strategies used for the acquisition of GNSS signals and the different ways that the signal can be processed in the acquisition process. With the knowledge taken from the different search strategies a description of methods to acquire the AltBOC(15,10) is made.
- In chapter 4 there is a full description of all the innovative and traditional acquisition algorithms that were considered for this work. The traditional algorithms are the Classical Acquisition, Parallel Frequency Space Search and the Parallel Code Phase search. The others are the Delay and Multiply, the Double Block Zero Padding Transition Insensitive and the Dual Sideband CBOC as BPSK.
• Chapter 5 presents the full study of all the innovative algorithms comparing them with the reference which was the Classical Acquisition. The study of these algorithms consists on a theoretical analysis of their Mean Acquisition Time and Performance, the performance is based on the computation of probabilities of detection for various carrier-to-noise ratios. After the theoretical analysis, another one is done this time based of results from simulated signals, this was made to validate the theoretical study made before. To finalize a real signal from the Deimos receiver was provided to all these methods so we could get a more conclusive study.

• Finally, chapter 6 summarizes the main results of this work and concludes on the obtained results. Also there is a recommendation for some future work regarding these algorithms.
Chapter 2

GNSS Signals

2.1 GPS Signal

2.1.1 GPS Signal Overview

Each GPS signal consists of three components, [4], [2]:

- **Carrier**: The carrier is a RF signal with a frequency \( f \) which allows the codes and navigation data to travel from the satellite to the receiver.

- **Ranging code**: There are two types of codes:
  
  – **Coarse/Acquisition code**: The C/A code is a unique sequence of 0s and 1s assigned to each satellite. This sequence is called a pseudo-random noise sequence, PRN. These sequences have special properties which allow all satellites to transmit at the same frequency without interfering with each other. Because of the auto-correlation properties of the PRN, the receiver can identify by the code which satellite is sending the message. The C/A code is a sequence of 1023 bits, called *chips*, which is repeated every millisecond, 1 ms, the duration of each chip is about 1 \( \mu \text{s} \) and the *chip width, or wavelength* is about 300 m. The *chipping rate* is 1.023 MHz or Mcps, megachips per second.

  – **Precision code**: The P-code is also a PRN sequence however an extremely long one, about \( 2 \times 10^{14} \) chips. The chipping rate is 10.23 Mcps and the chip width is more or less 30m, this smaller wavelength results in a better precision for range measurements. These codes repeat once a week.

<table>
<thead>
<tr>
<th>Code</th>
<th>Bits</th>
<th>Chip Rate [MHz]</th>
<th>Wavelength [m]</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/A</td>
<td>1023</td>
<td>1.023</td>
<td>300</td>
<td>1 ms</td>
</tr>
<tr>
<td>P-Code</td>
<td>( 2 \times 10^{14} )</td>
<td>10.23</td>
<td>30</td>
<td>1 week</td>
</tr>
</tbody>
</table>

Table 2.1: Summary of Signals Components

5
Navigation Data: This is a binary-coded message consisting of three major components. The first part contains the GPS date and time, plus the satellite’s status and an indication of its health. The second part contains orbital information called ephemeris data and allows the receiver to calculate the position of the satellite. The third part, called the almanac, contains information and status concerning all the satellites; their locations and PRN numbers. The navigation message is transmitted at a leisurely of 50 bits per second with a bit duration of 20 ms and it takes 12.5 minutes for the all message to be sent.

Nowadays each GPS satellite transmits continuously using two RF in the L-band referred to as Link-1(L1) and Link-2(L2). In the following years another band, Link-5 or L5 will be added. Table 2.2 shows all the GPS future and present bands.

<table>
<thead>
<tr>
<th>Frequency Band</th>
<th>Centre Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>1575.42 MHz</td>
</tr>
<tr>
<td>L2</td>
<td>1227.60 MHz</td>
</tr>
<tr>
<td>L5(future)</td>
<td>1176.45 MHz</td>
</tr>
</tbody>
</table>

Table 2.2: GPS Frequency Bands

Only L1 transmits a “civil signal”, a GPS signal that can be used by non-DoD authorised users. The L5 band will too have a civil signal but it is not implemented yet.
2.1.2 GPS L1 Civil Signal

Time Domain Description

The GPS satellite broadcasts the following navigation signal:

\[ s_{L1}(t) = \sqrt{2P_C} D(t) x(t) \cos (2\pi f_{L1} t + \theta_{L1}) \] (2.1a)
\[ + \sqrt{2P_Y} D(t) y(t) \sin (2\pi f_{L1} t + \theta_{L1}) \]
\[ s_{L2}(t) = \sqrt{2P_Y} D(t) y(t) \sin (2\pi f_{L2} t + \theta_{L2}) \] (2.1b)

The most important signal component is in equation 2.1a, the first component is the basis for the vast majority of civil applications and will be the object of more detail in this section. This signal component is in the L1 band just like the other second component, but this corresponds to the military signal. The signal in equation 2.1b is the military signal on the L2 band. Any of these three signals in 2.1 is the product of four terms:

- \( \sqrt{P} \) is the amplitude, with P the average power in a generic signal
- \( D(t) \) is the navigation data
- \( x(t) \) or \( y(t) \) is the spread spectrum code
- \( \cos (2\pi f_d t + \theta) \) or \( \sin (2\pi f_d t + \theta) \) (inphase or quadrature) is the RF carrier.

Both the data and the codes use BPSK to modulate the transmitted carrier. They are composed of sequences of rectangular pulses with amplitude +1 or -1. These pulses are described by:

\[ U(t) = A_p \left( \frac{t - \tau}{T} \right) \] (2.2)

where

\[ p(t) = \begin{cases} 1, & 0 \leq t \leq 1 \\ 0, & \text{otherwise} \end{cases} \] (2.3)

Thus, the pulse \( U(t) \) has amplitude, A, duration, T, and is delayed by \( \tau \) seconds. Now we are in conditions to describe the C/A code, or \( x(t) \) as mentioned earlier. This is a 1023-chip pattern, so one period of this code can be described by:

\[ x_1(t) = \sum_{n=0}^{N-1} x_n p \left( \frac{t - nT_C}{T_C} \right) \] (2.4a)
\[ = p \left( \frac{t}{T_C} \right) * \sum_{n=0}^{N-1} x_n \delta(t - nT_C) \] (2.4b)

N=1023 chips
with $T_C$ standing for the chip duration.

**Amplitude Spectrum**

The code can also be described as a convolution, 2.4b, this enables us to analyze the frequency content of the C/A code signal. Now we can use 2.4b to derive the Fourier transform of one period of the C/A code:

$$X(t) = F\{x(t)\} = F \left\{ p \left( \frac{t}{T_C} \right) * \sum_{n=0}^{N-1} x_n \delta(t - nT_C) \right\}$$

$$= T_C \sqrt{N} \text{sinc}(fT_C) X_{\text{code}}(f)$$

(2.5a)

with $\text{sinc}(x) = \sin(\pi x)/(\pi x)$

Since we have a sinc function we know that the function takes a maximum value of unity at $f = 0$ and falls to 0 at $1/T_C$. Hence, the null-to-null bandwidth is $2/T_C$, which is 2.046 MHz for the C/A code and 20.46 MHz for the P(Y) code.

![Figure 2.2: GPS L1 Power Spectra, [3]](image)

**2.1.3 Auto-Correlation and Random Sequences**

**Auto-Correlation**

The GPS codes were, as mentioned earlier, selected for their auto- and cross-correlation properties. Correlation measures the similarity of two waveforms or sequences. Auto-correlation measures the similarity between any waveform and time shifts of itself and cross-correlation compares a given
waveform with all the time shifts of a second waveform.

The time-average auto-correlation function for the code from the kth satellite, $x^{(k)}(t)$, is:

$$R(\tau) = \frac{1}{T_{\text{code}}} \int_{0}^{T_{\text{code}}} x^{(k)}(t)x^{(k)}(t - \tau)dt$$

(2.6)

where $T_{\text{code}}$ is the code period.

As shown in this equation, auto-correlation multiplies $x^{(k)}(t)$ by a time-shift replica of itself, $x^{(k)}(t - \tau)$ and integrates the product. If $x^{(k)}(t)$ resembles its time-shift replica then $R(\tau)$ will be large.

If we concentrate our attention to time shifts that are integer multiples of the chip duration, $\tau = lT_C$, and combine 2.6 with 2.4a and simplify the equation we get:

$$R(\tau = lT_C) = \frac{1}{T_{\text{code}}} \int_{0}^{T_{\text{code}}} x^{(k)}(t)x^{(k)}(t - lT_C)dt = \frac{1}{N} (\#\text{agreements} - \#\text{disagreements})$$

(2.7)

This is an important result because the auto-correlation can simply be calculated summing the products of the underlying +1’s and -1’s. Note that the auto-correlation takes its maximum when $\tau = 0$, this happens because at this time all the chips are aligned so there are no disagreements, they all agree.

![Figure 2.3: Auto-Correlation of a simple binary code.](image)

As we can see by figure 2.3, the peak occurs when both signals are aligned.

**Random Sequences**

As shown in figure 2.4, the auto-correlation exhibits a peak for the time-shift of 0. This is a very important feature of the PRN codes, but not the only one. The indication of the sidelobes levels is very important too.
Figure 2.4: Auto-Correlation properties of the PRN that as the same length as the C/A codes (N=1023), [2]

\[ \text{Sidelobes} = 20 \log_{10} \frac{1}{\sqrt{N}} = -30 \text{dB}, \text{ for } N = 1023 \]  

For PRN codes of length 1023 the sidelobes are -30dB relative to the main peak, this feature makes the C/A codes much easier to find and are one of the main reasons for the use of PRN. But if the length is 10230, like the P-Code, we have a -40dB sidelobe level relative to the main peak, so we can conclude that the higher is N, the better are the chances of finding our C/A code.

2.2 Modulation Types for Satellite Navigation

In this section we are going to briefly enumerate the modulation types (when describing the Galileo systems these techniques will be more detailed including their auto-correlation properties) that are used for the satellite signals. This modulation process is essential because it lets impress the generated code by the satellite on the carrier. The most used modulation process is the BPSK, binary phase shift keying, but nowadays this method cannot be so much used because the BPSK modulation can not provide a good sharing of the current band spectrum by multiple signals. So new methods had to be studied like the BOC, Binary Offset Carrier, modulation.

2.2.1 Binary Phase Shift Keying (BPSK)

This type of modulation is a digital signaling scheme in which an RF carrier is either transmitted “as is” or with a 180° phase shift over successive intervals depending on whether a digital 0 or 1 is being conveyed. In practical terms, a 0 bit leaves the carrier unchanged and a 1 bit multiplies the carrier by -1, which is equivalent to shifting the phase of the sinusoidal signal by 180°, as exemplified in figure 2.5, where \( T_b = 1/R_b \), being \( R_b \) the data rate in bits per second.
2.2.2 Direct Sequence Spread Spectrum (DSSS)

DSSS is a spread spectrum technology, in this technology the transmitted signal takes up more bandwidth than the information signal that modulates the carrier or broadcast frequency. The name 'spread spectrum' comes from the fact that the carrier signals occur over the full bandwidth (spectrum) of a device's transmitting frequency.

As shown in figure 2.6 DSSS is an extension of BPSK. It just adds a third signal, referred to as spreading, which is similar to the data waveform but at a much higher symbol rate. The spreading is then multiplied by the data being transmitted. The resulting signal resembles white noise. However, this noise-like signal is used to exactly reconstruct the original data at the receiving end, by multiplying it by the same PRN. This process, known as "de-spreading", mathematically constitutes a correlation of the transmitted PRN sequence with the PRN sequence that the receiver already knows the transmitter is using.

There are three primary reasons why DSSS waveforms are employed for satellite navigation. First and most importantly, the frequent phase inversions in the signal introduced by the PRN waveform enable precise ranging by the receiver. Second, the use of different PRN sequences from a well-designed set enables multiple satellites to transmit signals simultaneously at the same frequency. A receiver can distinguish among these signals, based on their different PRN sequences. For this reason, the transmission of multiple DSSS signals having different spreading sequences on a common carrier frequency
is referred to as code division multiple access (CDMA). Finally, DSSS provides significant rejection of narrowband interference.

2.2.3 Binary Offset Carrier Signals (BOC)

Binary Offset Carrier (BOC) modulation offers a simple and effective way of moving signal energy away from band centre and consequently allows the use of a same frequency band by both GPS and Galileo systems. Moreover it offers a high degree of spectral separation with signals centered in the used band and a better robustness against interference.

An Offset Carrier signal results from the product of a NRZ, Non-return-to-zero, spreading code with a subcarrier. BOC modulation is obtained by the product between the spreading code and a squared sub-carrier, which is a NRZ signal, equal to the sign of the sine or cosine waveform.

Because this is the modulation of the future we will study the characteristic of this modulation with more depth.

Amplitude Spectrum

The previous GNSS signals studied used a chip waveform equal to a rectangular pulse, equation 2.2. BOC (sine) signals replace this pulse for:

\[ \text{sign} \left[ \sin \left( \frac{2\pi M t}{T_C} \right) \right], \quad 0 \leq t \leq T_C \] (2.9)

The chip waveform is chopped into M shorter rectangles each with duration \( T_S = T_C / M \). These subchips have alternating sign.

This modulation is represented by BOC(\( \alpha, \beta \)), \( \beta \) is the chipping rate normalized to 1.023 Mcps and \( \alpha \) is the subcarrier frequency normalized to 1.023 Mcps, where the subcarrier frequency is \( 1/2T_S \). Consequently, the number of subchips per chip is \( M = 2\alpha/\beta \).

![Figure 2.7: Amplitude spectrum for the C/A-code and a BOC(1,1)](image)

12
As shown in figure 2.7, BOC(1,1) has no energy at the band center, but pushes its power out to approximately ±1 MHz. However, the subcarrier frequency, $\alpha$, identifies the spectral peak for BOC signals. This spectral separation is an important characteristic of BOC signals, meaning that we can use a larger number of signals on the same band.

**Auto-Correlation**

The auto-correlation function for a BOC signal is, [6]:

$$R_{BOC}(p_n, n)(\tau) = (-1)^k \left[ \frac{k^2 - 2kp - k + p}{p} + (4p - 2k + 1) \frac{|\tau|}{T_C} \right], k = \left\lfloor \frac{2p|\tau|}{T_C} \right\rfloor$$ (2.10)

And for a C/A code is:

$$R_{\text{code}}(i) = \begin{cases} 
1 - \frac{|\tau|}{T_C}, & |\tau| < T_C \\
0, & \text{otherwise}
\end{cases}$$ (2.11)

Figure 2.8 compares equations 2.10 and 2.11. These signals were purposely chosen because of their similarity, the C/A-code has a chipping rate of 1 Mcps just like the BOC code. Despite their similarity by figure 2.8 we can conclude that BOC has a sharper auto-correlation peak which is very good for positioning purposes. However, it has false peaks (secondary lobes) almost as strong as the main peak, which is not a good thing because these can be confused by main peaks leading to wrong acquisition parameters which can result in a bigger ranging error for example.

Figure 2.8: Auto-correlation function chip waveform for the C/A-code and a BOC(1,1)
2.3 Galileo Navigation System

Galileo is the European global navigation satellite system and aims to provide a highly accurate, guaranteed global positioning service under civilian control. It will be interoperable with GPS and Glonass, the American and Russian global navigation satellite systems [9].

Galileo is based on a constellation of 30 satellites (27 operational plus 3 on reserve) and ground stations providing information about the users positioning. The satellites will be on a medium earth orbit (MEO) and distributed in 3 circular orbits planes at an altitude of 23222 km. Each orbit plane will have an inclination of 56° and will contain 9 operational satellites plus one in reserve.

![Figure 2.9: Galileo Constellation of Satellites.](image)

The Galileo navigation Signals are transmitted in the four frequency bands indicated in the next figure, figure 2.10. These four frequency bands are the E5a, E5b, E6 and E1 bands. They provide a wide bandwidth for the transmission of the Galileo Signals.

![Figure 2.10: Galileo Frequency Plan [5].](image)

The band shown in the previous figure have been selected in the allocated spectrum for Radio Navigation Satellite Services (RNSS) and Aeronautical Radio Navigation Services (ARNS), the last one employed by civil aviation users that allows safety critical applications.
Table 2.3: Galileo Frequency Bands. E5a and E5b signals are part of the E5 signal in its full bandwidth.

### 2.3.1 Galileo E1 Band

The E1 band has two different services, one open service which is open to civil users and one PRS or public regulated service. This last one is an encrypted navigation service designed to be more resistant to jamming, involuntary interference and spoofing. The PRS is primarily intended for use by EU Member State government agencies, including emergency services, police, critical infrastructure and networks. Access to the PRS will be controlled through an encryption key system approved by Member States.

E1 signal characteristics are summarized in the table below.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Signal Component</th>
<th>Modulation</th>
<th>Chip Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1B OS</td>
<td>Data</td>
<td>CBOC(6,1,1/11)</td>
<td>1.023 MHz</td>
</tr>
<tr>
<td>E1C OS</td>
<td>Pilot</td>
<td>CBOC(6,1,1/11)</td>
<td>1.023 MHz</td>
</tr>
<tr>
<td>E1A PRS</td>
<td>Data</td>
<td>$BOC_{\cos}(15,2.5)$</td>
<td>2.5575 MHz</td>
</tr>
</tbody>
</table>

Table 2.4: Galileo E1 Band Signal Characteristics

In the table above we have a pilot signal, this signal is data free, it is made only of a ranging code, not modulated by a navigation data stream. This brings some advantages:

- Absence of data bit transitions allows for longer integration times which means better performance in noisy environments;
- Also without data modulation less parameters need to be estimated.

Figure 2.11: Spectra of Galileo Signals in E1, [1]
2.3.2 Galileo E6 Band

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Signal Component</th>
<th>Modulation</th>
<th>Chip Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>E6 CS</td>
<td>Data</td>
<td>BPSK(5)</td>
<td>5.115 MHz</td>
</tr>
<tr>
<td>E1 CS</td>
<td>Pilot</td>
<td>BPSK(5)</td>
<td>5.115 MHz</td>
</tr>
<tr>
<td>E6 PRS</td>
<td>Data</td>
<td>BOC(10,5)</td>
<td>5.115 MHz</td>
</tr>
</tbody>
</table>

Table 2.5: Galileo E6 Band Signal Characteristics, [1]

In this band we have the E6 Commercial Services, or CS as presented in the table above. Encrypted and more accurate than the OS signal, the Commercial Service allows for development of applications for professional or commercial use owing to improved performance and data with greater added value than that obtained through the open service. The Commercial Service (CS) is aimed at market applications requiring higher performance than offered by the Open Service. It provides added value services on payment of a fee. Galileo CS uses combination of two encrypted signals for higher data throughput rate and higher accuracy authenticated data.

![Figure 2.12: Spectra of Galileo Signals in E6](image)

2.3.3 Galileo E5 Band

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Signal Component</th>
<th>Modulation</th>
<th>Chip Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>E5a</td>
<td>Data</td>
<td>AltBOC(15,10)</td>
<td>10.23 MHz</td>
</tr>
<tr>
<td>E5a</td>
<td>Pilot</td>
<td>AltBOC(15,10)</td>
<td>10.23 MHz</td>
</tr>
<tr>
<td>E5b</td>
<td>Data</td>
<td>AltBOC(15,10)</td>
<td>10.23 MHz</td>
</tr>
<tr>
<td>E5b</td>
<td>Pilot</td>
<td>AltBOC(15,10)</td>
<td>10.23 MHz</td>
</tr>
</tbody>
</table>

Table 2.6: Galileo E5 Band Signal Characteristics

All the signals present in this band are AltBOC modulated, this makes, by far the most sophisticated signal among all the signals used for GNSS. Due to its very low code range noise and the even lower multipath influence on the positioning solution the Galileo E5 is able to achieve very accurate positioning results.
2.4 Galileo Modulations

2.4.1 CBOC(6,1,1/11)

Since the main aim of our work is the Galileo E1 band, the modulation that is going to be mostly used in all the methods is the Composite BOC, CBOC(6,1,1/11). As seen before this modulation corresponds to the Galileo E1 band which is the scope of this work.

The CBOC(6,1,1/11) is formed by the sum of $\sqrt{\frac{10}{11}} g_1(t)$ symbols (BOC(1,1)) and $\sqrt{\frac{1}{11}} g_2(t)$ symbols (BOC(6,1)). From [6] the CBOC(6,1,1/11) time function, $g(t)$, is described as:

$$g(t) = \alpha g_1(t) \pm \beta g_2(t)$$  \hspace{1cm} (2.12)

With $\alpha = \sqrt{\frac{10}{11}}$ and $\beta = \sqrt{\frac{1}{11}}$ (the minus sign in 2.12 stands for the pilot signal and the plus sign for the data signal) [5].

Again from [6] we can define $g_1(t)$ (BOC(1,1)) and $g_2(t)$ (BOC(6,1)) as:

$$g_1(t) = \begin{cases} 
\text{sign}[\sin(2\pi t/T_C)] & 0 \leq t \leq T_C \\
0 & \text{otherwise}
\end{cases}$$ \hspace{1cm} (2.13a)

$$g_2(t) = \begin{cases} 
\text{sign}[\sin(12\pi t/T_C)] & 0 \leq t \leq T_C \\
0 & \text{otherwise}
\end{cases}$$ \hspace{1cm} (2.13b)

Applying equation 2.12, the CBOC(6,1,1/11) pilot and data component have the shapes shown in figure 2.14.

The main difference between the data and pilot signals is the $\beta$ value which translates into a $\pm \sqrt{\frac{1}{11}}$ difference as we can see in figure 2.14.
Now that the CBOC(6,1,1/11) time signal was described it is time to find its PSD and ACF. These two parameters are very important because they posibilitate to analyze the signal in terms of it’s frequency (PSD) and find the maximum code delay for a determined degradation (ACF). The frequency analysis is important to determine the necessary bandwidth to be used in the receptor and with the ACF we can determine the code step to be used in the acquisition (more details about the code step when performing the acquisition will be given in the next chapter).

**CBOC PSD**

From [10] it is known that the PSD of CBOC(6,1,1/11) is given by:

\[
G_{CBOC(6,1,1/11)}(f) = \frac{10}{11} G_{BOC(1,1)}(f) + \frac{1}{11} G_{BOC(6,1)}(f)
\]  

(2.14)

But to plot the PSD we are going to use the expressions developed in [6] and equation 2.14:

\[
G_{BOC(pn,n)}(f) = T_C \sin^2(fT_C) \tan^2 \left( \frac{\pi fT_C}{2p} \right)
\]  

(2.15a)

\[
G_{CBOC(6,1,1/11)}(f) = \frac{T_C}{11} \sin^2(fT_C) \left[ 10 \tan^2 \left( \frac{\pi fT_C}{2} \right) + \tan^2 \left( \frac{\pi fT_C}{12} \right) \right]
\]  

(2.15b)

Equation 2.15a is the expression for the PSD of any BOC signal, defined as BOC(pn,n). So, using this in equation 2.14 we obtain 2.15b which is the expression for the PSD of the CBOC(6,1,1/11). From 2.15b it is possible to plot the CBOC PSD, as depicted in figure 2.15.

The figure 2.15 shows that the side lobes are very close in power to the main lobe, which as seen before, can be a disadvantage. Also for all the simulations a bandwidth of 8MHz was used, this value guarantees that almost all the power is going to be used.
To plot the ACF of the CBOC(6,1,1/11) signal the following equations presented in [6] are going to be used:

\[
R_{CBOC(6,1,1/11)} = \frac{10}{11} R_{BOC(1,1)}(\tau) + \frac{1}{11} R_{BOC(6,1)}(\tau) \pm 2 \sqrt{\frac{10}{11}} I_{1,2}(\tau)
\]

\[
I_{1,2}(\tau) = \frac{1}{12} \left[ \Lambda_{TC/12} \left( |\tau| - \frac{TC}{12} \right) + \Lambda_{TC/12} \left( |\tau| - \frac{3TC}{12} \right) \Lambda_{TC/12} \left( |\tau| - \frac{5TC}{12} \right) \right. \\
\left. - \Lambda_{TC/12} \left( |\tau| - \frac{7TC}{12} \right) - \Lambda_{TC/12} \left( |\tau| - \frac{9TC}{12} \right) \right]
\]

\[
\Lambda_L(t) = \begin{cases} 
1 - \frac{|t|}{L} & |t| < L \\
0 & \text{otherwise}
\end{cases}
\]

Equation 2.16a reprents the expression for the ACF of a CBOC(6,1,1/11) signal. In order to fully understand this equation first we need to know the expression for the ACF of a BOC(pn,n) signal with \(0 \leq |\tau| \leq TC\), which is represented in equation 2.10. Also in equation 2.16a there is an unknown function, \(I_{1,2}(\tau)\), this function is described in equation 2.16b with \(\Lambda_L(t)\) representing the triangle function, 2.16c.

The plot of equation 2.16a is displayed in figure 2.16.

From figure 2.16 we can see represented the maximum code delay for a 50% loss of power or a 3dB loss. This is an important value to determine the maximum code step for the acquisition process. If we define the code step as \(0.1728 \times 2 = 0.3456\) chips this means that in the worst case the peak will have a 50% loss of power. With this loss the peak will more easily blend in with the noise leading to a worse acquisition process, but a 50% loss is still an acceptable value.
2.4.2 AltBOC(15,10)

The Alternative BOC modulation (AltBOC) is the most complex modulation present in GNSS. The main scope of this work is to acquire CBOC(6,1,1/11) signals in the E1 band but one of the methods is to acquire the AltBOC signal from the CBOC. This means that the parameters needed for the AltBOC acquisition (Doppler and code phase) are going to be obtained by the CBOC(6,1,1/11) signal. So this section will contain just a brief explanation of the AltBOC, just enough to see why these signals take so much time and resources to be acquired.

The AltBOC is conceptually very similar to the BOC modulation but with an important difference. Contrary to BOC, AltBOC provides high spectral isolation between the two upper main lobes and the two lower main lobes (considering the I and Q phases separately). This is accomplished by using different codes for each main lobe [1].

AltBOC modulation is referred as $\text{AltBOC}(m,n)$, with $f_s = m \times 1.023$ and $f_c = n \times 1.023$. For the Galileo E5 band we are going to work with AltBOC(15,10). The E5 AltBOC(15,10) signal carries two independent data channels and two pilot channels and is defined as [6], [1], [5]:

\[
x(t) = \sum_k A(k)p_d^*(t - kT_C) + B(k)p_d(t - kT_C) + \overline{A}(k)p_p^*(t - kT_C) + \overline{B}(k)p_p(t - kT_C)
\]

(2.17)
With:

\[ A(k) = e_{AI}(k) + je_{AQ}(k), \quad B(k) = e_{BI}(k) + je_{BQ}(k) \]
\[ A(k) = \tau_{AI}(k) + j\tau_{AQ}(k), \quad B(k) = \tau_{BI}(k) + j\tau_{BQ}(k) \]

\[ e_{AI}(k) = c_{AI}(k) \begin{cases} \text{Data Code} \\ d_{AI}(k) = \pm 1 \end{cases}, \quad e_{BI}(k) = c_{BI}(k) d_{BI}(k) = \pm 1 \]

\[ e_{AQ}(k) = c_{AQ}(k) = \pm 1, \quad e_{BQ}(k) = c_{BQ}(k) = \pm 1 \]

\[ \tau_{AI}(k) = e_{BI}(k) \cdot e_{BQ}(k) \cdot e_{AQ}(k), \quad \tau_{AQ}(k) = e_{BI}(k) \cdot e_{BQ}(k) \cdot e_{AI}(k) \]
\[ \tau_{BI}(k) = e_{AI}(k) \cdot e_{BQ}(k) \cdot e_{AQ}(k), \quad \tau_{BQ}(k) = e_{BI}(k) \cdot e_{AI}(k) \cdot e_{AQ}(k) \]

And the complex pulses are defined by:

\[ p_d(t) = sc_d(t) + jsc_d(t - T_s/4) \]
\[ p_p(t) = sc_p(t) + jsc_p(t - T_s/4) \]

\[ sc_d(t) \text{ and } sc_p(t) \text{ have periods } T_s = 2T_c/3 \]

Figure 2.17: Data and Pilot AltBOC(15,10) subcarriers waveforms, [6].

Now that the AltBOC(15,10) time signal was mathematically described it is now time to see its PSD so we can compare with the other modulations and analyze.
AltBOC PSD

From [6] the PSD of the AltBOC(15,10) is given by:

\[
G_{AltBOC(15,10)}(f) = \frac{T_c \cos(\pi f T_c)}{9 \cos \left( \frac{\pi f T_c}{4} \right)} \left[ \frac{4 \sin^2 \left( \frac{f T_c}{6} \right)}{\cos \left( \frac{f T_c}{12} \right)} - \sin^2 \left( \frac{f T_c}{12} \right) \cos \left( \frac{\pi f T_c}{6} \right) \right]
\]  

(2.20)

With equation 2.20 it is now possible to plot the Power Spectral Density of the AltBOC(15,10) signal:

![PSD of the AltBOC(15,10) signal]

Figure 2.18: PSD of the AltBOC(15,10) signal

Using figure 2.18 we can see that the signal spans over 50 MHz; when compared with the CBOC(6,1,1/11) there is a five times increase. This means that the receivers got to have a big bandwidth capable to receive the full signal (this is one of the disadvantages to the acquisition of AltBOC signals). With the method proposed later this enormous bandwidth will not be necessary for the estimate of the Doppler and code phase because the signal to be processed is just the CBOC(6,1,1/11).

AltBOC ACF

Now that the PSD of the signal was analyzed is time to study the AltBOC ACF. For this is needed to use the equations presented in [6]:

\[
R_{AltBOC(15,10)}(\tau) = 8 \Lambda T_c/6 (\tau) - \frac{16}{3} \Lambda T_c/6 (|\tau| - \frac{T_c}{3}) + \frac{8}{3} \Lambda T_c/6 (|\tau| - \frac{2T_c}{3}) - \frac{1}{3} \Lambda T_c/12 (|\tau| - \frac{T_c}{12}) - \frac{1}{3} \Lambda T_c/12 (|\tau| - \frac{3T_c}{12}) + \frac{1}{3} \Lambda T_c/12 (|\tau| - \frac{5T_c}{12}) + \frac{1}{3} \Lambda T_c/12 (|\tau| - \frac{7T_c}{12}) - \frac{1}{3} \Lambda T_c/12 (|\tau| - \frac{9T_c}{12}) - \frac{1}{3} \Lambda T_c/12 (|\tau| - \frac{11T_c}{12})
\]  

(2.21a)
The expression presented in 2.21 has many advantages, one of them being its simplicity. Because it only depends on a triangle function (equation 2.16c) it is very easy to represent its graphic, figure 2.19:

![Normalized AltBOC(15,10) ACF](image)

Figure 2.19: Normalized AltBOC(15,10) ACF

From [5] it is known that the AltBOC has a code length of 10230 chips, more than the double of CBOC(6,1,1/11) code length, and from figure 2.19 the maximum code step in acquisition for a 3dB loss is $2 \times 0.077 = 0.154$ chips, approximately two times the CBOC(6,1,1/11) delay. So the number of cells in the code phase that need to be searched are, $10230 / 0.154 = 66429$. When compared with the CBOC(6,1,1/11) there is 5.6 increase which in terms of acquisition time will not be good. Because of this reason a different acquisition method was thought for this modulation. Acquire the Doppler and code phase of the AltBOC from the CBOC. It is known that both signals come from the same satellite so if we estimate the Doppler and code phase from the CBOC(6,1,1/11) we will be able to find the AltBOC(15,10) Doppler and code phase with a simple calculation.
Chapter 3

GNSS Acquisition Techniques and Concepts

3.1 Basic Concepts

The main objective of the acquisition stage is to determine coarse values of the carrier frequency in the presence of Doppler effect and code phase of the satellite signals to then initialize the tracking stage. The acquisition is a complex process that takes time and can be performed in numerous ways. In the next chapter we are going to describe on what these ways can differ.

3.1.1 Baseband Signal Processing

The received signal after down-conversion can be represented as:

\[
s(t) = \sqrt{P_C}D(t - \tau)x(t - \tau) \cos (2\pi(f_{IF} + f_D)t) + n(t)
\]

where \(f_{IF}\) is the intermediate frequency.

The signal presented above, \(s(t)\), is written as a continuous function in time, even though it is now a discrete time signal, after having suffered an analog to digital conversion so the signal is a sequence defined for a discrete set of times. For this work we will consider that the sample rate is high enough to make the simplification to a continuous signal.

This signal contains some parameters that need to be estimated, such as:

- \(\tau\), is the code delay and contains the basic range and time information
- \(f_D\), is the Doppler shift and contains the pseudorange rate information used to compute the user velocity and clock frequency
- \(n(t)\) is the noise present in the incoming signal.

Now for the estimation of the Doppler frequency the GPS receiver employs two reference signals:
\[
\sqrt{2} \cos \left[ 2\pi (f_{IF} + \hat{f}_D) t \right] \quad (3.2a) \\
\sqrt{2} \sin \left[ 2\pi (f_{IF} + \hat{f}_D) t \right] \quad (3.2b)
\]

where \( \hat{f}_D \) is the estimate of the Doppler frequency assumed by the local oscillator.

These are called the inphase and quadrature reference signals. The process in which these signals are used is shown in the figure below:

As seen in the figure above, the satellite signal shown in equation 3.1 is multiplied by the inphase and quadrature reference signals and then a low-pass filters the result, [2]. In the end there is an inphase and quadrature signal:

Inphase: \( \sqrt{P_C} D(t - \tau) x(t - \tau) \cos (2\pi \Delta f_D t) \) \quad (3.3a)

Quadrature: \( \sqrt{P_C} D(t - \tau) x(t - \tau) \sin (2\pi \Delta f_D t) \) \quad (3.3b)

\( \Delta f_D = f_D - \hat{f}_D \) \quad (3.3c)

This process is referred as \textit{carrier wipeoff} because the signal is no longer modulated by the carrier frequency, the frequency of the resulting signal is the difference between the true Doppler and the receiver’s best estimate of Doppler, equation 3.3c.

The signals that we’ll be working later in this work are in baseband form, so equation 3.1 would have the following form:

\[
s(t) = \sqrt{P_C} D(t - \tau) x(t - \tau) \exp(2\pi f_D t) + n(t) \quad (3.4)
\]

The process is exactly the same as the one described earlier but the inphase and quadrature signals will be one complex signal. Thus figure 3.1 would be modified to figure 3.2:
After the carrier wipeoff there is the code wipeoff, shown in figure 3.3.

As shown in figure 3.3, the inphase and quadrature signals are both multiplied by the local replica, $x(t - \hat{\tau})$, so $\Delta \tau = \tau - \hat{\tau}$, being $\Delta \tau$ the code delay. After the code delay the signal goes through a correlator and the output is in equation 3.5, one function for the inphase signal and another one for the quadrature.

\[
S_{In}(\Delta \tau, \Delta f_D) = \sqrt{P_{CD}} \int_0^{T_{CO}} x(t - \tau) x(t - \hat{\tau}) \cos(2\pi f_D t) dt \quad (3.5a)
\]

\[
S_{Quad}(\Delta \tau, \Delta f_D) = \sqrt{P_{CD}} \int_0^{T_{CO}} x(t - \tau) x(t - \hat{\tau}) \sin(2\pi f_D t) dt \quad (3.5b)
\]

The equations above averages over $T_{CO}$ seconds. This is the coherent averaging time or dwell time. The $T_{CO}$ needs to be shorter than 20 ms, the duration of the GPS L1 C/A code data bit, this is an important value because this way it is harder to integrate across a data bit boundary. Integrations across a data bit boundary complicate the acquisition process and result in a worse sensitivity. In equations 3.5 it is assumed that the data are, $D = +1$ or $-1$ during the entire coherent averaging time.

The inphase and quadrature signal of equation 3.5 could be written in a different manner:
\[
\tilde{S} = S_{In} + jS_{Quad} = \sqrt{P_C}D\tilde{R}(\Delta \tau, \Delta f_D)
\] (3.6a)

\[
\tilde{R}(\Delta \tau, \Delta f_D) = \frac{1}{T_{CO}} \int_{T_{CO}}^{0} x(t - \tau)x(t - \hat{\tau}) \exp(j2\pi \Delta f_D t) dt
\] (3.6b)

With this notation we have obtained a very important function, equation 3.6b, which is called the ambiguity function. With this function it is possible to estimate \((\Delta \tau, \Delta f_D)\). For this estimation it is important to have the magnitude of \(\tilde{S}\), because this function is stripped of the data bit, D and the carrier phase offset, \(\Delta \theta\) and so it is this function that is going to be estimated:

\[
|\tilde{S}|^2 = S_{In}^2 + S_{Quad}^2 = P_C|\tilde{R}|^2
\] (3.7)

### 3.1.2 Search Space

With two parameters to estimate, a search space can be “built”, figure 3.4.

![Search Space for a GPS receiver](image)

**Figure 3.4: Search Space for a GPS receiver.** Signal search area covers Doppler frequency, \(\Delta f_D\) and code phase, \(\Delta \tau\), [4].

In the search process all code phases and Doppler frequencies are looked through. In figure 3.4 each tentative code phase is called a time, code or phase bin and each tentative frequency shift is called a Doppler or frequency bin. Together, a code bin and a Doppler bin compose a cell. The increments of the code bin and Doppler bin are very important. If we consider a Doppler of \(\pm 10\) kHz with increments of 500 Hz and a GPS code with 1023 chips with one chip increment we should have, 1023 \times \left(2^{\frac{10000}{500}} + 1\right) = 41943 combinations.
This is a very large number of combinations and if we consider a Galileo signal which has 10230 or 4092 chips this number becomes 419430 and 167772. This increase in the number of bins is primarily because of the modulation used in the Galileo signals (chapter 2). Besides the length of the code, the size of a code bin is also essential. For a CBOC modulation (Galileo system), the code phase resolution, as seen before in chapter 2 is 0.3456 chips (for a 3dB degradation). For this modulation the Doppler increment has to be different because the integration time has to be more than 4ms, so:

\[
\text{Doppler Increment} = \frac{1}{2 \times T_{\text{int}}} \tag{3.8}
\]

Then by equation 3.8 the Doppler increment has to be 125 Hz. So the CBOC signal with an increment of 125 Hz and an acceptable resolution of 0.3456 chips has

\[
\frac{4092}{0.3456} \times \left( \frac{10000}{500} + 1 \right) \approx 959063
\]

combinations.

Almost a million of combinations is a very large number to search, thus a method that searches every bin (like serial search) may not be the best approach in terms of acquisition time. Therefore, as we have seen before two aspects modulate the size of the space search:

- **Doppler Frequency**, the Doppler frequency alters the size of the search grid in terms of its resolution and range. The resolution is calculated by equation 3.8 and depends on the Integration time and the constant that multiplies by the inverse of the \(T_{\text{int}}\). Usually this number is 1/2 but 3/2 is acceptable too. The range can be ±10 KHz but for this work because of the specs of the receiver and because is static the range can vary from +5KHz to -5KHz, [11].

- **Code Phase**, just like the Doppler frequency the code phase defines the size of the search grid in terms of its resolution and range. The difference is that this resolution and range depend on the signal characteristics, for example, the GPS L1 C/A has 1023 chips but the Galileo CBOC(6,1,1/11) has 4092 chips (range). The GPS can have a code step of 1 chip and the Galileo CBOC(6,1,1/11) for the same resolution has to have a code step of 0.3456 chips (resolution). So the code phase is intrinsically connected to the signal characteristics.

### 3.1.3 Detection and False Alarm Probabilities

In each search space a test statistic is calculated based on the correlation results, if this test exceeds a pre-defined threshold \(\phi\) the signal is considered present and a estimate for the code phase and frequency is achieved. This is the detection stage.

The detection of the signal is a statistical process because each cell either contains noise with the signal absent or noise with the signal present and each case has its own probability density function (PDF), figure 3.5 shows an example.

Each method will have its statistics and they will be analyzed in chapter 5.

29
3.2 Search Strategies

The acquisition of signals can be decomposed in two major areas. The first is the search strategy, which depends on how the search on the grid (figure 3.4) is made. For example, if there is only need to search the frequency space or the code phase or even just search a smaller range of Doppler frequencies (assisted search).

3.2.1 Serial Search

On the serial search strategy every cell of the search grid has to be searched. This is the most basic of the search strategies in terms of algorithm complexity, but because of this simplicity it is also the slowest of all the acquisition strategies. While in other strategies the search grid is smaller, in this one it has the biggest size, for example, for the CBOC(6,1,1/11) with a Doppler step of 125 Hz, range of ±5KHz and a code step of 0.3456 chips, the total number of bins are 41943. So for the serial search technique all the bins have to be searched (the grid is as big as can be), but for other techniques as we will see later the number of bins drastically reduces.

3.2.2 Parallel Search

In the Parallel search all bins of one dimension(frequency or code delay) are searched at the same time. For example, for a determined frequency all the code delay bins are searched at the same time. With this method all the cells are still searched but several cells are tested simultaneously.
Massive Correlators Bank

The easiest way to reduce the acquisition time is implementing a parallel search with a massive numbers of correlators. On every bin a correlation is made between the local replica (with the Doppler frequency and code phase corresponding to that bin) and the incoming signal. For the serial search only one correlator is used so two bins can not be tested at the same time, but for a parallel search, if there are as many correlators as frequency bins it is possible to search at the same time all the frequency bins for a defined code phase. When comparing with the serial search this method is as faster as the number of correlators used. If the bank of correlators is in the order of millions one can even test all the cells at the same time, reducing the acquisition time to a minimum.

Parallel Code Phase Search

The parallel code phase search is made only in the frequency space in the sense that for a specific frequency all the code delays are searched at the same time. Acquisition algorithms that use this method are the fastest because the number of bins to search is very low. Examples of this algorithm are the Delay and Multiply and the use of Circular correlation with FFT.

Parallel Frequency Search

This search is just like the last one, but instead of searching all the code delays simultaneously here the frequency bins are the ones tested at the same time for a specific code delay. It takes more time than the Parallel Code Phase Search but the complexity is lower because not so many bins are searched at the same time. Just like the Parallel Code Phase Search usually the use of FFT is accompanied with this technique.

Double Parallel Search

A double parallel search is a method that simultaneously searches the frequency and code phase. This method in terms of implementation is the hardest one, but in order to search both parameters at the same time the use of FFT’s and circular correlations is necessary. An example of an algorithm using this search is the Double Block Zero Padding. More details about this algorithm will be given in the next chapter.

3.2.3 Assisted Search

The assisted search is when some information is known previous to the acquisition process which allows the search space to be reduced:

- **Doppler Estimate**, sometimes when starting the acquisition process we have some estimation about the Doppler Frequency and with this estimate it is possible to reduce the search grid; instead of having a Doppler range of ±5 KHz now it can be ±1 KHz. An example of this technique is when
the receiver has some data connection, from this connection is possible to have some estimate of
the receiver position and from that a Doppler estimate can be obtained.

- **Hand-Over**, the hand-over technique is using the information of the acquisition of a simpler signal
to accelerate that of a more complex one. For example, acquire the Doppler and code phase from
the CBOC(6,1,1/11) and from this information get the AltBOC(15,10) Doppler and code phase.
This method is possible because both signals come from the same satellite (thus having related
Doppler shifts) and are synchronized (thus having related code phases).

### 3.3 Signal Correlation Techniques

As said earlier the acquisition of signals can be decomposed in two major areas, the first is the search
strategy (related to the way the search space is swept) and now we are going to analyze the second
major area which is the way that the signal is processed in the acquisition process (the way the signal is
treated in the correlation process).

#### 3.3.1 Matched Filter

In the acquisition process there is always a local replica, this replica in the matched filter method is
exactly the same as the incoming signal. For example, if the incoming signal is a CBOC(6,1,1/11) the
local replica is also a CBOC(6,1,1/11), and a correlation between these signals is made. This is the
standard in terms of acquisition and the most basic solution when performing the acquisition.

#### 3.3.2 Sub-Carrier Demodulation Simplifications

An interesting way to reduce the acquisition time is to treat the complex signal as a simpler one. For
example the BPSK modulation is much simpler than the CBOC. So if there was a way to analyze the
CBOC as a BPSK the advantages in terms of acquisition time would be great. As seen in the previous
chapter, the code step for the same loss of 3dB is almost 3 times less for the BPSK modulation, which
means a significant decrease of code delays to be searched.

**Single Sideband and Double Sideband Processing**

The CBOC modulated signal can be seen as a superposition of two BPSK signals located at the
negative and positive subcarrier frequencies. So by using one of this bands (Single Sideband) and shift
it in its frequency domain we can treat it as BPSK signal. If we use both bands (Double Sideband) and
combine them in the end of acquisition, they give a better performance than just using one. With the use
of just one band there is a loss of 3dB regarding the use of the two bands, [12].
3.4 Detector algorithms

The role of the detector is to form a decision statistic based on the received signal samples; this decision is based on a combination of the correlator outputs. The correlator outputs and the way they are combined form the various detector algorithms present today. In this section three algorithms are going to be described and evaluated theoretically:

- Coherent Combining
- Non-coherent Combining
- Differentially Coherent Combining

3.4.1 Coherent Combining

The coherent combining is a very important algorithm when processing a signal. In fact this kind of integration provides the best performance in terms of noise variance reduction. With the increase of a single trial integration time the sensitivity is also increased, so our best interest is to keep this value as high as possible, but the coherent integration time as some limitations. Although this technique reduces noise, its performance is limited by the bit transitions provided by secondary codes and data navigation messages. So in order to avoid this problem usually the same integration time as the primary code period is used. Because of the challenges from the coherent integration time other integration techniques can be used to improve sensitivity like the non-coherent integration or differentially coherent combining.

Statistical Analysis

Considering the decision statistic described in equation 3.6b, for the coherent combining, from [13], this decision takes the form:

\[
D(\hat{\theta}) = \left| \sum_{m=0}^{M_N-1} r(mT_s) \exp(-j\hat{\omega}DM_t)c(mT_s - \hat{\zeta}T_{chip}) \right| \tag{3.9}
\]

In equation 3.9, the parameter to estimate is \( \hat{\theta} = \{ \hat{\zeta}, \hat{\omega} \} \) with \( \hat{\zeta} \) has the code phase and \( \hat{\omega} \) as the Doppler frequency. \( r(mT_s) \) is the sample that was correlated with \( T_s = \frac{1}{F_s} \) as the sampling period. \( M_c \) is the number of periods coherently summed and \( N_s = \frac{T_{chip}}{T_s} \) is the number of samples per code period. Taking in account all the variables described above the integration time can be described by:

\[
T_{int} = M_cT_{code} \tag{3.10}
\]

Recalling section 3.3.1, we can have two situations in the correlators: only noise is present (which is referred as \( H_0 \) condition) or signal and noise is present (referred as \( H_1 \) conditions). When only noise is present the distribution of the decision variable can be represented as central \( \chi^2 \) random variable with two degrees of freedom and variance:
\[ \sigma_Y^2 = \frac{M_c N_s B_i f N_0}{2} \]  

(3.11)

with the bandwidth, \( B_i f \) typically as half the sampling frequency and \( N_0 \) as the noise power spectral density.

The probability density function of the decision variable when signal is present, \( f_{H_0} \), and when signal and noise is present, \( f_{H_1} \), are given by:

\[
f_{H_0}(x) = \frac{1}{2\sigma_Y^2} \exp \left( -\frac{x}{2\sigma_Y^2} \right) \]

(3.12a)

\[
f_{H_1}(x, \lambda) = \frac{1}{2\sigma_Y^2} \exp \left( -\frac{x + \lambda}{2\sigma_Y^2} \right) I_0 \left( \frac{\sqrt{x\lambda}}{\sigma_Y} \right) \]

(3.12b)

with \( I_n(\cdot) \) standing for the \( n^{th} \) order modified Bessel function of the first kind.

We know that only when noise is present the distribution of the decision variable can be represented as central \( \chi^2 \) random variable with two degrees of freedom, but when both noise and the signal are present, the distribution of the decision variable is a non-central \( \chi^2 \) random variable with two degrees of freedom with non-centrality parameter:

\[
\lambda = \frac{(M_c N_s)^2 C}{2} \]

(3.13)

This non-centrality parameter, \( \lambda \), is given by the squared-magnitude of the expected value of the coherent correlator output, \( \mathbb{E} \left[ \sum_{m=0}^{M_c N_s - 1} r(mT_s) \exp\left( -j\hat{\omega}_D mT_s \right) c(mT_s - \hat{\zeta}_C T_i) \right] \).

If a single decision statistic is compared to a threshold, \( V_{th} \), to determine whether or not the signal is present and synchronized, then the probabilities of false alarm, \( P_{fa} \), and detection, \( P_d \), are given by:

\[
P_d = \int_{V_{th}}^{\infty} f_{H_1}(x, \lambda) dx = Q_1 \left( \frac{\sqrt{\lambda}}{\sigma_Y}, \frac{\sqrt{V_{th}}}{\sigma_Y} \right) \]  

(3.14a)

\[
P_{fa} = \int_{V_{th}}^{\infty} f_{H_0}(x) dx = \exp \left( -\frac{V_{th}}{2\sigma_Y^2} \right) \]  

(3.14b)

where \( Q_k(a,b) \) is the \( k^{th} \) order generalized Marcum Q-function which is defined as, ??:

\[
Q_k(a,b) = \int_b^{\infty} x \left( \frac{x}{a} \right)^{m-1} \exp \left( -\frac{x^2 + a^2}{2} \right) I_{m-1}(ax) dx \]

(3.15)

Equations 3.14 are going to be used when comparing the sensitivity of all the detectors for a general Galileo E1 signal.

### 3.4.2 Non-coherent Combining

The non-coherent combining does not suffer so much with bit transitions as the coherent integration. This technique consists in simply summing \( K \) instances of the output of the basic acquisition block. This
is called non-coherent combining and results in better detection for the same signal, because of its higher sensitivity. The downside of this technique is the increase in terms of acquisition time because of the number of summations made. We need to process the signal as many times as the necessary K summations.

Statistical Analysis

As stated earlier the decision statistic of the non-coherent combining is formed by non-coherently summing an integer number of times, K, the coherently combining outputs, so the decision variable is, [13]:

$$D(\hat{\theta}) = \sum_{k=0}^{K-1} \left| \sum_{m=kM_cN_s}^{(k+1)M_cN_s-1} r(mT_s) \exp(-j\tilde{\omega}_D mT_s) c(mT_s - \tilde{\zeta}_T g) \right|^2$$  \hspace{1cm} (3.16)

In this case the integration time has an increase, as big as the number of non-summation so:

$$T_{int}^{noncoh} = KM_cT_{code}$$  \hspace{1cm} (3.17)

In this method, when the signal is absent and there is only noise (hypothesis $H_0$) the distribution of the decision variable is the sum of K independent $\chi^2$ random variables with zero mean, variance $\sigma_Y^2$ and two degrees of freedom. This way it can be represented as a central $\chi^2$ distribution with variance $\sigma_Y^2$ and 2K degrees of freedom. When signal and noise are present (hypothesis $H_1$) the decision statistic is distributed as a non-central $\chi^2$ random variables with variance $\sigma_Y^2$, 2K degrees of freedom and non-centrality parameter:

$$\lambda = \frac{(M_cN_s)^2CK}{2}$$  \hspace{1cm} (3.18)

The PDF of both hypotheses is given by:

$$f_{H_0}(x) = \frac{1}{2\sigma_Y^2} \frac{1}{\Gamma(K)} \left( \frac{x}{2\sigma_Y^2} \right)^{K-1} \exp\left( -\frac{x}{2\sigma_Y^2} \right)$$  \hspace{1cm} (3.19a)

$$f_{H_1}(x, \lambda) = \frac{1}{2\sigma_Y^2} \left( \frac{x}{\lambda} \right)^{\frac{K-1}{2}} \exp\left( -\frac{x+\lambda}{2\sigma_Y^2} \right) I_{K-1} \left( \frac{\sqrt{x}\lambda}{\sigma_Y^2} \right)$$  \hspace{1cm} (3.19b)

with $\Gamma(K)$ as the Gamma function, defined as, [14]:

$$\Gamma(p) = \int_0^\infty t^{p-1} e^{-t} dt, \quad p > 0$$  \hspace{1cm} (3.20a)

$$\Gamma(p) = (p-1)!, \quad p \text{ an integer, } p > 0$$

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}, \quad \Gamma\left(\frac{3}{2}\right) = \frac{\sqrt{\pi}}{2}$$
Finally the probabilities of detection and false alarm for a threshold, $V_{th}$, are:

$$P_d = \int_{V_{th}}^\infty f_{H_1}(x)dx = Q_K \left( \frac{\sqrt{\lambda}}{\sigma_Y}, \frac{\sqrt{V_{th}}}{\sigma_Y} \right)$$ \hspace{1cm} (3.21a)

$$P_{fa} = \int_{V_{th}}^\infty f_{H_0}(x)dx = \frac{\Gamma_K \left( -\frac{V_{th}}{2\sigma_Y^2} \right)}{\Gamma(K)}$$ \hspace{1cm} (3.21b)

with $\Gamma_K(x)$ as the incomplete gamma function, [15]:

$$\Gamma_a(x) = \int_x^\infty t^{a-1}e^{-t}dt$$ \hspace{1cm} (3.22)

### 3.4.3 Differentially Coherent Combining

The Differentially Coherent Combining calculates the decision statistic using the product of the coherent correlator output and the complex conjugate of the previous coherent correlator output. An integer number, $R$, of such products are summed together and the square magnitude of the result is taken as the decision statistic.

#### Statistical Analysis

The $i^{th}$ complex coherent correlator output is given by [13]:

$$Y_i = \sum_{m=iM_cN_s}^{(i+1)M_cN_s-1} r(mT_s) \exp(-j\hat{\omega}DmT_s)c(mT_s - \hat{\zeta}T_C)$$ \hspace{1cm} (3.23)

so the decision statistic is:

$$D(\hat{\theta}) = \left| \sum_{r=1}^R Y_r Y_r^* \right|^2$$ \hspace{1cm} (3.24)

By [16] the PDF of the decision variable for both hypotheses, $H_0$ and $H_1$ is:

$$f_{H_0}(x) = \frac{1}{2\sigma_Y^2 \sqrt{1+\rho^2}} \exp \left( -\frac{z}{4\sigma_Y^2} \right) I_0 \left( \frac{z}{4\sigma_Y^2} \sqrt{1+\rho^2} \right)$$ \hspace{1cm} (3.25a)

$$f_{H_1}(x) \approx \frac{1}{2\pi} \int_0^\infty \frac{1}{t} \exp \left( -\frac{1}{2} \left( \frac{(m-x)^2}{t} + (m-t)^2 \right) \right) dt$$ \hspace{1cm} (3.25b)

where $\rho = \sqrt{\frac{C}{\sigma_Y^2}}$

For this study we are only going to analyze the case where $R=1$, because for a larger $R$ the expressions are very difficult to be theoretically represented and would require a very intensive study on this matter. Since the objective of this section is an introduction on the different detectors existing today there is no need for such a deep analysis.
The probabilities of false alarm and detection are given by, [16]:

\[
P_d \approx \Psi(V_{T_n}) = \frac{1 + erf\left(\frac{m}{\sqrt{2}}\right)}{4} - \frac{1}{\sqrt{8\pi}} \int_{0}^{\infty} \exp\left(-\frac{(y - m)^2}{2}\right) erf\left(\frac{-m + V_{T_n}}{\sqrt{2}}\right) dy\]

(3.26a)

\[
P_{fa} \approx \exp\left(-\frac{V_{T_n}}{2}\right)
\]

(3.26b)

where \(m = \sqrt{2 \frac{C}{N_0}} T_{int}, V_{T_n} = \frac{V_{th}}{\sigma Y}\) and, [14]:

\[
erf(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt
\]

(3.27)

is the error function.

### 3.4.4 Detectors Comparison

With all the expressions deduced in the previous sections we are going to evaluate the sensitivity of the various detectors in terms of their probabilities of detection and false alarm. This study is going to be made for a \(C/N_0 = 33\) dBHz, because with this value the different sensitivities are clearly visible.

The receiver operating characteristics, ROC, are going to be evaluated for a E1 Galileo signal, CBOC(6,1,1/11), so the parameters used are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C/N_0)</td>
<td>33 dBHz</td>
</tr>
<tr>
<td>Coherent Integration Time</td>
<td>4 ms</td>
</tr>
<tr>
<td>Sampling Frequency</td>
<td>20 MHz</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>10 MHz</td>
</tr>
<tr>
<td>(R)</td>
<td>1</td>
</tr>
<tr>
<td>(K)</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3.1: Values used to evaluate the different detectors ROC

Now we are in conditions to compute the ROC for the different detectors, figure 3.6:

By figure 3.6 the method with the best sensitivity is the coherent combining over two periods. This is an expected result, as stated earlier. This method guarantees the best results, the only downside are the bit transitions (for figure 3.6 we are assuming that there are not any transitions) and because the Galileo codes have secondary codes this result will not correspond to reality. So to overcome this nuisance there are the non-coherent combining and the differentially coherent combining, the last method is theoretically better than the non-coherent combining. Despite being more complex it has a better sensitivity than the non-coherent, being very close to the coherent combining over two periods which is an excellent result.

Another interesting fact from figure 3.6 is the very large increase in sensitivity when using two periods instead of one. For a false alarm probability of \(10^{-3}\) the difference in terms of probability of detection is approximately 30%.
When applying all the Search Strategies, Signal Correlation Techniques and Detector Algorithms stated before we are left with numerous possibilities to acquire the AltBOC signal. The most effective methods to acquire the AltBOC using the information on the previous sections will be described next [17]. Also for all the methods we are assuming that the secondary code phase is unknown and the aim of the acquisition is only to acquire the primary code.

3.5 AltBOC(15,10) Acquisition

When applying all the Search Strategies, Signal Correlation Techniques and Detector Algorithms stated before we are left with numerous possibilities to acquire the AltBOC signal. The most effective methods to acquire the AltBOC using the information on the previous sections will be described next [17]. Also for all the methods we are assuming that the secondary code phase is unknown and the aim of the acquisition is only to acquire the primary code.

3.5.1 Single sideband acquisition (SSB)

The AltBOC method can be obtained by the processing of only one of its bands. The input signal is mixed with the local carrier centred at one of the sidebands and the result is then mixed with the local signal void of any subcarrier. This method can be used in two ways:

- Using any of the E5aQ, E5bQ, E5aI and E5bI channels with coherent integration limit of one millisecond (secondary code chip duration).
- Non-coherent combining of E5a or E5b. Integration duration not constrained by the spreading codes or the data.

3.5.2 Double sideband acquisition (DSB)

As seen before in this method the correlation results of both bands are used together. When acquiring the AltBOC this methodology can be used in two ways:

- Non-coherent combination of the Quadrature (E5aQ, E5bQ) or Inphase bands (E5aI, E5bI).
- Non-coherent combining of E5a and E5b.
3.5.3 Full-band independent code acquisition (FIC)

In this method the local carrier is centred at the centre of E5 band. The reference signal contains the spreading code and the subcarrier corresponding to the signal component of interest. The four variants of this method are:

- Processing of any of the E5aQ, E5bQ, E5aI and E5bI bands with integration limit of 1 milisecond (secondary code chip duration).
- Non-coherent combining of E5a or E5b. Integration duration not constrained by the spreading codes or the data.
- Non-coherent combining of E5I or E5Q, combination of pilot or data channels.
- Non-coherent combining of E5a and E5b.

3.5.4 Direct AltBOC method

In this method the reference signal generator employs a look-up table so as to combine all the four signal components, then the correlation with these four components is made and the maximum correlation value among the four is selected. Taking into account the search strategies listed before this can be considered as a matched filter.

3.5.5 AltBOC search schemes comparison

From [17]:

<table>
<thead>
<tr>
<th>Method</th>
<th>Code / Subcarriers Generators</th>
<th>Correlation Waveform</th>
<th>Correlation Power (% of Direct AltBOC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSB, any one code</td>
<td>1 / 0</td>
<td>BPSK(10)</td>
<td>21.34</td>
</tr>
<tr>
<td>SSB, one sideband</td>
<td>2 / 0</td>
<td>BPSK(10)</td>
<td>42.68</td>
</tr>
<tr>
<td>DSB, both sidebands</td>
<td>4 / 0</td>
<td>BPSK(10)</td>
<td>85.36</td>
</tr>
<tr>
<td>FIC, any one code</td>
<td>1 / 1</td>
<td>BPSK(10)</td>
<td>21.34</td>
</tr>
<tr>
<td>FIC, pilots or data</td>
<td>2 / 2</td>
<td>AltBOC(15,10)</td>
<td>42.68</td>
</tr>
<tr>
<td>FIC, pilots and data</td>
<td>4 / 2</td>
<td>AltBOC(15,10)</td>
<td>85.36</td>
</tr>
<tr>
<td>Direct AltBOC</td>
<td>4 codes with a LUT</td>
<td>AltBOC(15,10)</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3.2: Resource usage in AltBOC search based schemes

As seen in table 3.2 the best method is the matched filter/direct AltBOC. This technique is always the best because we are correlating the full incoming signal with an equal replica so it has to present the best results. But this method presents some disadvantages in terms of acquisition time, so alternatives are welcome, and the Dual sideband seems to be a very good alternative due to its correlation power and faster acquisition due to the correlation waveform (with this waveform the code step can be larger which results in less cells to search).
3.5.6 Hand-over from the E1

It is known that both the E1 signal and the E5 are transmitted from the same satellites and their
dynamic will be the same (same range, range rate and clock error). So the Doppler dynamics from the
E1 will be the same as the E5 and the start of both codes is aligned. This alignment is very important
because it allows to acquire the code phase delay of the E5 signal from the E1.

We know that the start of an E1 code always corresponds to the start of an E5 code (the opposite
is not always true so this method only works from the E1 to the E5 and not the contrary) and a delay of
$\frac{1}{4}T_{E1}$, $\frac{3}{2}T_{E1}$ and $\frac{3}{4}T_{E1}$ also corresponds to the beginning of an E5 code, because the period of the E1
code, $T_{E1}$, is 4ms and the period of the E5 code, $T_{E5}$, is 1ms. So if the delay of the E1, $\tau_{E1}$, is known
the E5 delay, $\tau_{E5}$, can also be known by the least positive remainder of the following division:

$$\tau_{E5} = \text{remainder of } \frac{\tau_{E1}}{T_{E5}}$$  \hspace{1cm} (3.28)

If we have $\tau_{E1} = 2.5\text{ms}$ the remainder of the division $\frac{\tau_{E1}}{T_{E5}} = \frac{0.0025}{0.001}$ is 0.5µs which corresponds
approximately to the delay we want to find.

The downside of this method is the difference between AltBOC and CBOC modulations. As seen
before the AltBOC signal needs a much higher resolution when finding the right code phase, so after
this method it is necessary a small search in the vicinity of the delay obtained (from the E1 signal) with
the AltBOC signal.

Moreover, the Doppler frequency is obtained by:

$$f_{D}^{E5} = f_{D}^{E1} \times \left( \frac{f_{0}^{E5}}{f_{0}^{E1}} \right)$$  \hspace{1cm} (3.29)

with $f_{0}^{E5}$ and $f_{0}^{E1}$ as the carrier RF frequency (1575.42MHz to E1 and 1176.45MHz to E5).
Chapter 4

Acquisition Algorithms Pre-selection

4.1 Classical Acquisition (Serial Search with Matched Filter)

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>Signal Processing Technique</th>
<th>Combining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial Search</td>
<td>Matched Filter</td>
<td>Coherent</td>
</tr>
</tbody>
</table>

Table 4.1: Classical Acquisition characteristics

The “classical acquisition” presented in [18] will be the reference when evaluating the performance of all the studied methods. The paper considers the problem of acquiring the OS signals (\(B = \) Data channel and \(C = \) Pilot Channel) of Galileo E1 band. This band carries three signals combined in a CASM (Coherent Adaptive Sub-Carrier Modulation) scheme. Channel A is used for public regulated service (PRS), whereas the latter two are used for open service (OS). Channel B and C play different roles: The first is the data channel that carries the navigation message, whereas the second, the pilot channel, is using for determining the pseudorange.

At least two solutions can be envisaged for the acquisition of the OS signals. One of them consists of processing only the pilot signal. However, only half of the useful power will be utilized and the receiver may fail to acquire and track weak signals. Alternatively, pilot and data channels may be combined in order to provide a more reliable signal detection. In [18], two strategies were conceived for the efficient combining of data and pilot channels. In the first strategy (non-coherent combining), the received signal is correlated separately with the pilot and data local replicas. The correlation outputs are then squared and summed. The second strategy (coherent combining) consists in correlating the received signal by two new signals, resulting from the summing and subtraction of the data and pilot local replicas (data + pilot and data - pilot). Than, the maximum of the two correlations is adopted as the decision variable.

The RF Galileo E1 band signal is [19], [20].

\[
y(t) = \sqrt{2} \left[ e_B(t) - e_C(t) \right] \cos(2\pi f_{RF}t) - \frac{1}{3} \left[ 2e_A(t) + e_A(t)e_B(t)e_C(t) \right] \sin(2\pi f_{RF}t) \tag{4.1}
\]

where \(e_A(t), e_B(t)\) and \(e_C(t)\) are real binary sequences with values ±1. These signals are combined
using a CASM scheme to produce a constant envelope signal.

For this thesis the reference is the acquisition with coherent combining because it outperforms the acquisition with noncoherent combining.

\[
\text{Incoming Signal} \times \text{Baseband} \times \text{Frequency Generator} \exp(-j2\pi f_D t) \times \text{Local Code (Data+Pilot)} \times \text{Local Code (Data-Pilot)} \times \frac{1}{TCO} \int_0^{TCO} \cdots dt \times \frac{1}{TCO} \int_0^{TCO} \cdots dt
\]

\[|.|^2 \quad |.|^2 \quad \text{MAX} \quad \text{Output Decision Variable} \]

Figure 4.1: Galileo E1 signal acquisition with coherent combining, “Classical acquisition”

The algorithm of figure 4.1 is our reference because of its simplicity. This method of acquisition is similar to the GPS acquisition described earlier just with a difference in the local replicas. Because of the complexity of the Galileo E1 signal the local replica has to be divided in two codes, one with the Data + Pilot and another with the Data - Pilot. These two codes are necessary because of the secondary code on the Pilot signal. Having these two codes is a strategy to eliminate the ambiguity that can exist in the secondary code of the E1 Pilot signal. We know that only those two equivalent spreading sequences are possible ([21]) so when correlating with both, the equivalent code that maximizes the cross-correlation is likely to be the correct one. After the coherent combining of both correlators we are in the presence of the decision variable.

4.2 Parallel Frequency Space Search

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>Signal Processing Technique</th>
<th>Combining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Search</td>
<td>Matched Filter</td>
<td>Coherent</td>
</tr>
</tbody>
</table>

Table 4.2: Parallel Frequency Acquisition characteristics

The serial search method can be very time-consuming, but if one of the two parameters (Doppler frequency or code phase) could be eliminated from the search or implemented in parallel, the time of search would drastically improve.

The Parallel Frequency Space Search Algorithm parallelizes the search for one parameter using a Fourier transform.

As seen in figure 4.2 the incoming signal (just as the serial search algorithm) is multiplied by the locally generated PRN code. It is after this multiplication that this method diverges from the serial search one. The resulting signal is transformed into the frequency domain by a Fourier transform.
Figure 4.2: Block diagram of the Parallel Frequency Search

The squared output of the Fourier transformation will show a distinct peak in magnitude if the locally generated PRN code is perfectly aligned. The peak will be located at the frequency index corresponding to the frequency of the continuous-wave signal and thereby the frequency of the carrier wave signal.

The accuracy of this method depends on the length of the Fourier transformation. For example, if a 1ms of data is analysed, the number of samples is \( N = \frac{1}{1000} \times f_s \), with \( f_s \) as the sampling frequency. The frequency resolution of the output is:

\[
\Delta f = \frac{f_s}{N}
\]  

(4.2)

When comparing this method with the serial search, the advantage in terms of time is evident. Where the serial search algorithm steps through all possible code phases and frequency carriers, this algorithm only steps through the different code phases; for the Galileo system 4092/code step or 10230/code step. However this comes with the cost of a frequency domain transformation with each code phase. Depending on the implementation of the Fourier Transform this method can be much faster than the serial search one. Also with this method the complexity increases very much because of the many correlators needed.

### 4.3 Parallel Code Phase Search

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>Signal Processing Technique</th>
<th>Combining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Search</td>
<td>Matched Filter</td>
<td>Coherent</td>
</tr>
</tbody>
</table>

Table 4.3: Parallel Code Phase Acquisition characteristics

As seen before in chapter 3 the main contributor for the search space size is the code phase, so this method parallelizes the code phase dimension and only the frequency steps will be performed.

For this method a circular cross correlation will be made between the input and the PRN. Let the DFT of the finite lengths sequences \( x(n) \) and \( y(n) \), both with length \( N \), be:
\[ X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad (4.3a) \]
\[ Y(k) = \sum_{n=0}^{N-1} y(n)e^{-j2\pi kn/N} \quad (4.3b) \]

The circular cross-correlation between the two sequences and its discrete N-point Fourier transform can be expressed as:

\[ z(n) = \frac{1}{N} \sum_{m=0}^{N-1} x(-m)y(m - n) \quad (4.4a) \]
\[ Z(k) = \sum_{m=0}^{N-1} x(m)e^{j2\pi km/N} \sum_{n=0}^{N-1} y(m + n)e^{-j2\pi k(m+n)/N} = X^*(k)Y(k) \quad (4.4b) \]

The process in the receiver for the Parallel Code Phase Search method is in the block diagram of figure 4.3. The incoming signal is multiplied by a locally generated carrier signal and then a Fourier transformation is applied. The result of this transformation is multiplied with the complex conjugate of a locally generated PRN code that was transformed into the frequency domain. The result of the multiplication suffers an inverse Fourier transform and is transformed into the time domain. The absolute value of this output is the correlation between the input and the PRN code. Like all the other methods this value is our decision variable.

The efficiency of this method depends on the implementation of the FFT and IFT, because for every frequency step, both transformations are performed.

### 4.4 Double Block Zero Padding Transition Insensitive (DBZPTI)

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>Signal Processing Technique</th>
<th>Combining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double Parallel Search</td>
<td>Matched Filter</td>
<td>No summations</td>
</tr>
</tbody>
</table>

Table 4.4: DBZPTI Acquisition characteristics

One of the methods to be implemented is the Double Block Zero Padding Transition Insensitive Improved, presented in [22]. This method is an improvement from the one proposed in [23] by the same authors. The improved one is better in terms of degradation of the peak in function of the Doppler frequency.
Figure 4.3: Block diagram of the Parallel Code Phase Search Algorithm
This acquisition algorithm is explained in detail in [23] and [22]. Here is going to be briefly explained in 5 steps:

1. The incoming signal in baseband is split in $2M$ blocks of equal length with each block containing $N$ samples. This $M$ is defined by the Doppler range and the integration time, $M = \Delta f_D \times T_{int}$. After the splitting we are left with $2M$ blocks and every two adjacent blocks are combined to form a larger block containing $2N$ samples, figure 4.4.

Figure 4.4: DBZPTI Step 1 - Processing of the incoming signal.
2. M blocks (corresponding to the defined $T_{int}$) of local code are generated. Instead of combining two adjacent blocks, this time, every block is zero padded, so we still have $2N$ samples like the incoming signal, figure 4.5.

![Local Code Diagram](image)

2. Local Code

![Diagram](image)

Figure 4.5: DBZPTI Step 2 - Generation of the local code.

3. As illustrated in figure 4.6, each combination of 2 blocks of the incoming signal is circularly correlated with each combination of blocks of the local code. Because of the zero padding of the local code blocks only half of this circular correlation is kept. This leads to a matrix containing M rows and each row has N columns. The N points represent a partial correlation of length $T_{int}/M$ ms at N possible delays.
3 Circular Correlation

Figure 4.6: DBZPTI Step 3 - Partial Circular Correlation.

4. The last step is made for all possible delays. This means we are going to shift the incoming signal by one block and apply the circular correlation with the local code unchanged, figure 4.7. This step is made M times, for all the incoming signal combination of two blocks. When this is complete we are left with M matrices each one corresponding to some N code delays. These matrices are concatenated generating one matrix of $M \times N$ columns and M rows, $\underbrace{M \times N \times M}_{\text{Rows} \times \text{Columns}}$, figure 4.7.
Circular Correlation for all Delays

5. The matrix $M \times N \times M$ generated before is zero-padded by a value of our choice. According to [22], the best value is $ZP = 4M$, leading to a matrix $5M \times N \times M$, figure 4.8. The last step consists of applying an FFT to all columns (every delay), one by one.
5 Zero-Padding and FFT Application

Figure 4.8: DBZPTI Step 5 - Application of the FFT.
4.5 Dual Sideband, CBOC as BPSK

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>Signal Processing Technique</th>
<th>Combining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial Search</td>
<td>Dual Sideband</td>
<td>Coherent and Non-coherent</td>
</tr>
</tbody>
</table>

Table 4.5: Dual Sideband, CBOC as BPSK Acquisition characteristics

An interesting method to study and analyze is the acquisition of a CBOC signal as a BPSK. This method can be very advantageous, because as seen before, the maximum code phase step for a BPSK signal is larger when comparing with the CBOC (for the same 3 dB loss of power the BPSK delay is approximately 3 times bigger). This means that the number of bins to be searched decreases and the acquisition process is quicker. The downside of this method is the loss of power when using one or two bands instead of all the signal and the deterioration of the correlation function properties.

If we want to transform the CBOC signal into a BPSK a frequency shift is needed, figure 4.9. This technique is called Dual Sideband Processing, DSP.

![Frequency shift of the CBOC signal](image)

Figure 4.9: Frequency shift of the CBOC signal

By the power spectral density of the BPSK modulation, figure 2.7, it is clear that the signal power is all concentrated in the middle but by the PSD of the CBOC, figure 2.15, the main lobe is not situated in the middle but in its subcarrier frequencies 1.023 MHz and -1.023 MHz. So we need to shift these two lobes for the middle, first the upper lobe and then the lower. To achieve this we have to multiply
the incoming signal by $\exp(-j2\pi f_{\text{shift}}t)$ for the upper lobe and $\exp(j2\pi f_{\text{shift}}t)$ for the lower lobe, figure 4.10. Now we are in the presence of two BPSK signals, these two signals are going to be acquired by a classical acquisition and in the end a non-coherent combining of both is going to be performed.

Figure 4.10: Dual Sideband Acquisition Diagram
4.6 Delay and Multiply

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>Signal Processing Technique</th>
<th>Combining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Search</td>
<td>Matched Filter</td>
<td>Coherent</td>
</tr>
</tbody>
</table>

Table 4.6: Classical Acquisition characteristics

The advantage of the Delay and Multiply is the elimination of the Doppler frequency before correlation. This way it is only necessary to search the code phase bins one time and not for every Doppler frequency. In the end, when the code delay is found there is only need for a quick search on the frequency bins of that code delay to determine the Doppler frequency. The elimination of the Doppler frequency is accomplished by multiplying the incoming signal with a delayed version of itself:

\[
s_{new}(t) = s_{incoming}(t) \times s_{delayed}(t - \tau)
= A(t)A^*(t - \tau) \exp(j2\pi f_{doppler}t) \exp(-j2\pi f_{doppler}(t - \tau))
= A(t)A^*(t - \tau) \exp(j2\pi f_{doppler}\tau)
\approx A(t)A^*(t - \tau) \times 1, \quad \text{if } \tau \approx 0
\] (4.5)

As seen by equation 4.5 the new signal no longer has a carrier, so it does not depend on the Doppler frequency. With a single multiplication the Doppler frequency has been eliminated. The delay, \( \tau \), has to be as small as possible so \( \exp(j2\pi f_{doppler}\tau) \approx 1 \), but it cannot be smaller than one chip, otherwise the new code loses its correlation properties. So the minimum delay is defined as \( \tau = 1/(1.023 \times 10^6) = 9.7752 \times 10^{-7} \) seconds.

With this last multiplication we have gained the loss of the dependency on the Doppler frequency but the noise now is squared and obviously the output will pay the price. After the multiplication the acquisition process is similar to the classical acquisition, figure 4.11.

![Figure 4.11: Acquisition diagram of the Delay and Multiply method](image-url)
4.7 Selected Algorithms

The goal of this work is to select an algorithm which can accelerate the acquisition process in DEIMOS’ GRIP receiver, but there are some limitations regarding the acquisition process. The first one is the use of FFT’s or IFFT’s. This means that any method that uses these two operations unfortunately can not be implemented without major architectural changes to the receiver. Secondly the method has to use less than the maximum number of correlators present on the DEIMOS’ GRIP receiver (80).

Taking into account all the conditions above the methods that fulfill the criteria are the Dual Sideband CBOC as BPSK and the Delay and Multiply which, although implying changes in the receiver architecture, they are considered minor when compared to those required by FFT based techniques. In addition, one other goal was to analyze a method which, although not being eligible for easy implementation on DEIMOS’ receiver, would be interesting for software receivers. So the study for the DBZPTI is also made for future reference. The methods that are going to be analyzed in the next chapter are:

- Classical Acquisition, as the reference
- Double Block Zero Padding Transition Insensitive, DBZPTI
- Delay and Multiply
- Dual Sideband, CBOC as BPSK
Chapter 5

Performance Analysis

5.1 Simulation Parameters

5.1.1 Carrier-to-Noise Density Ratio

As stated earlier, the goal of this work is to select an algorithm eligible for implementation on DEIMOS’ GRIP receiver (appendix A, [8]) (with minor modifications) which accelerates acquisition for strong signals (as those present on an open-sky scenario) when compared with the currently implemented approach (serial search with matched filter). According to [5] the minimum received power on ground for Galileo signals at the output of an ideally matched RHCP 0 dBi polarized user receiving antenna when the SV elevation angle is higher than 10 degrees is -157 dBW. Considering that the power of the thermal noise at the receiver is typically $7.81 \times 10^{-21} W/Hz$ or -201 dBW/Hz, [24], the carrier-to-noise density ratio is given by:

$$\frac{C}{N_0} = -157 dBW - (-201 dBW/Hz) = 44 dBHz \quad (5.1)$$

This value will be our reference when addressing the performance of the algorithms. However, we will extend the range until a $C/N_0 = 35$ dBHz.

5.1.2 Probabilities of False Alarm

For the following simulations we are going to consider two probabilities of false alarm, one is the probability of false alarm of the cell, $P_{fa}^{cell}$, as mentioned earlier in chapter 3 and the other one is the probability of false alarm of the search grid, $P_{FA}^{grid}$. This probability was presented in [21] and is given by:

$$P_{FA}^{grid} (\beta) = 1 - (1 - P_{fa}^{cell} (\beta))^M \quad \text{With M the total number of cells in the search grid} \quad (5.2)$$

The probability of false alarm of the cell can be defined by the probability of false alarm of the grid. For the theoretical studies we are going to consider the reference $P_{FA}^{grid} = 0.01$, which means that
after doing a full sweep of the search space, there is a 1% chance of reaching a false alarm. For the simulation tests we are going to consider a $P_{f_a}^{cell} = 1 \times 10^{-4}$. To achieve somewhat reliable results using a $P_{FA}^{grid} = 0.01$ would take a very large amount of simulations (because the resolution would have to be very low, something only achieved with millions of simulations) in which it would take weeks just to simulate a single method. This way we are going to validate the theoretical data for a $P_{f_a}^{cell} = 1 \times 10^{-4}$ but keep the theoretical study for a $P_{FA}^{grid} = 0.01$ as a reference. Also values in the range of $P_{f_a}^{cell} = 1 \times 10^{-4}$ are typical and present in numerous studies, [23].

5.1.3 Integration Time and Doppler Step

The integration time is made equal to $T_{int} = 4\text{ms}$, because that is the E1 primary code period; so this value usually is not smaller and we are going to consider the same for all the methods simulated.

As a consequence, the Integration time defined the Doppler step given by 3.8 is going to be equal to 125 Hz.

5.1.4 Mean Acquisition Time, MAT

In all the methods the MAT will be primarily given by the integration time and the number of cells (correlations that are made). The only exceptions is the DBZPTI, because this is an algorithm for a software receiver and not a hardware one. Instead of computing the time by the MAT it will be evaluated based on the number of operations and compared with our reference, the classical acquisition. In all methods we are assuming that the correlation of the Data+pilot and the Data-Pilot is made in parallel; for this to be possible, 2 correlators are always used.

5.1.5 Incoming Signal

The incoming signal used in all the simulations is the same, defined as the one in [5]:

$$r_{incoming\text{signal}}(t) = \sqrt{C/2} (e_B(t) - e_C(t)) \exp(2\pi f_D t) + n_{complex}(t) \quad (5.3a)$$

$$e_B(t) = c_B(t) D_B(t) s_{cB}(t) \quad (5.3b)$$

$$e_C(t) = c_C(t) s_{cC}(t) \quad (5.3c)$$

$$n_{complex}(t) = n(t) + jn(t) \quad (5.3d)$$

In equation 5.3a C is the power of the incoming signal. This value is calculated versus the carrier to noise ratio, $C/N_0$. The $e_B(t)$ and $e_C(t)$ are the B and C components of the Galileo E1 signal, the B component has a Data code, equation 5.3b, whereas the C component does not, but this component has a primary and secondary code. All these codes except the Data were taken from [5]. The noise, equation 5.3d, is assumed to be complex white Gaussian and the Data is a random binary sequence.
5.2 Classical acquisition (Serial Search with Matched Filter)

5.2.1 Theoretical Performance Evaluation

MAT

The mean acquisition time is given by:

$$MAT_{\text{classical}} = \text{Total number of bins} \times \text{Coherent Integration time} \times \frac{1}{2}$$

In equation 5.4 when calculating the MAT we are considering half of the total cells, because we are interested in the mean time and not the total time, that is why the 1/2 factor is there. The total number of bins to be searched is given by:

$$\text{Total number of cells} = \text{Doppler Bins} \times \text{Code phase Bins}$$

$$= \frac{\text{Frequency range}}{\text{Doppler Step}} \times \frac{\text{Total number of chips}}{\text{Maximum Code delay}}$$

(5.4a)

With:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration Time</td>
<td>$NT_\tau = 4\text{ms}$</td>
</tr>
<tr>
<td>Code Step</td>
<td>0.3456 chips</td>
</tr>
<tr>
<td>Doppler Step</td>
<td>125 Hz</td>
</tr>
<tr>
<td>Frequency Range</td>
<td>$\pm 5\text{kHz}$</td>
</tr>
<tr>
<td>Total number of chips</td>
<td>4092</td>
</tr>
</tbody>
</table>

Table 5.1: Parameters used for the Classical Acquisition

Taking into account table 5.1:

$$\text{Total Number of cells} = 947200$$

$$MAT_{\text{Classical}} \approx 1894\text{seconds} \approx 32\text{minutes}$$

(5.5a)

(5.5b)

Statistical Analysis

To statistically evaluate this method, the equations presented in [18] are used:
\[ \sigma_n^2 = \frac{N_0 f_s}{4N} = \frac{N_0 B_{IF}}{2N} \]  
\[ P_{fa}^{cell}(V_{th}) = 1 - \left( 1 - \exp \left( -\frac{V_{th}}{4\sigma_n^2} \right) \right)^2 \]  
\[ \lambda \approx C, \text{ as the noncentrality parameter} \]  
\[ P_d(V_{th}) = 1 - \left[ 1 - \exp \left( -\frac{V_{th}}{4\sigma_n^2} \right) \right] \left[ 1 - Q_1 \left( \frac{\lambda}{2\sigma_n^2}, \sqrt{\frac{V_{th}}{2\sigma_n^2}} \right) \right] \]

where \( Q_1(\ldots) \) is the Marcum Q function of order 1

The equations above allow us to evaluate the classical acquisition in terms of its probability of detection \( 5.6d \). In equation \( 5.6a \) it is presented the noise variance in the end of the acquisition process. This value will be used to compute the probability of false alarm of a cell, \( 5.6b \) and the probability of detection \( 5.6d \). The \( P_{fa}^{cell} \) is given by equation \( 5.2 \) \( (P_{grid}^{FA} = 0.01) \) and is, \( \lambda \approx 1.0611 \times 10^{-8} \). From this value and equation \( 5.6b \) it is possible to get the threshold value. Having the noncentrality parameter, equation \( 5.6c \), and the threshold defined we can obtain the probability of detection by equation \( 5.6d \). The above process is made for different values of \( C/N_0 \) yielding the following figure:

![Figure 5.1: Theoretical Performance of the Classical Acquisition Method for, \( P_{grid}^{FA} = 0.01 \)](image)

Using figure 5.1 it is possible to see that for our reference \( C/N_0 \) the probability of detection is approximately 1.

### 5.2.2 Simulation Results

To validate our theoretical analysis a Matlab simulation is made based on the diagram of figure 4.1 and the incoming signal from equation \( 5.3a \).

To achieve somewhat reliable numbers \( 5 \times 10^3 \) results were taken and a histogram was plotted for
some values of $C/N_0$. The most important value is the reference, $\frac{C}{N_0} = 44dBHz$, the PDF for this value is shown in figure 5.2.

![Simulated PDF for the Classical Acquisition with a $C/N_0 = 44dBHz$ and $P_{fa} = 1 \times 10^{-4}$](image)

Using figure 5.2 it is possible to see that the probability of detection is very close to one, because the distance between the two distributions is very large and for the threshold chosen the distribution of the probability of detection is almost all to the right of the threshold (see also figure 3.5).

The probability of detection was simulated for a range of $C/N_0$ between 35 and 45 dBHz and the result is shown in figure 5.3:

![Theoretical and Simulated Performance of the Classical Acquisition Method for, $P_{fa} = 1 \times 10^{-4}$](image)

Analyzing figure 5.3 it is possible to see that the theoretical values are not very far from the simulated ones, and with exception of the value for $C/N_0 = 35$ dBHz, the maximum deviation is less than 1 %. These are good results because they validate the theoretical propositions.
5.2.3 Real Signals Simulation Results

To really put to the test our algorithms, a real signal from the receiver was processed using all the algorithms. This way it is possible to really validate the algorithms in representative and realistic scenarios and also draw more conclusive results about the algorithm to implement on the receiver.

When using real signals with the classical acquisition the result is in figure 5.4:

![Figure 5.4: Classical Acquisition with the use of real signals](image)

5.3 DBZPTI

5.3.1 Theoretical Performance Evaluation

MAT

As this method relies on the parallel search using FFT’s and is more targeted at software implementation, the calculation of the MAT cannot be based on the time it takes to test a single cell (which is usually the coherent integration time for classical methods) since the time to test a batch of cells may not be directly or mostly related with the coherent integration time, being instead related with the time to perform the FFT computation. This way the MAT is calculated in terms of computational time, computing the number of operations and how much time it takes the cpu to make every operation. According to [22] for $f_s = 20.48$ MHz without any coherent combining the time it takes to acquire a CBOC(6,1,1/11) signal is:

\[
MAT_{DBZPTI} \approx 6\text{seconds}
\] (5.7a)

As seen by the result of equation 5.7 it is not even fair to compare this method with the classical acquisition because of the use of FFT’s. In fact we are comparing 1894 seconds for the classical acquisi-
sition and 6 seconds for the DBZPTI; the DBZPTI is 315 faster. But when comparing with the parallel
code phase search, the authors claim that for the same results this method is 1.5 faster.

**Statistical Analysis**

Using the equations presented in [22] we can compute the probabilities of detection and false alarm:

\[
\sigma_{DBZPTI}^2 = \frac{N_0 M^2}{4T_{int}} \quad (5.8a)
\]

\[
\lambda_{DBZPTI} \approx \frac{A^2}{4} M^2 \quad (5.8b)
\]

Equation 5.8a represents the variance of the decision variable when the signal is not present in the
DBZPTI and equation 5.8b is the noncentrality parameter, necessary for the computation of the non-
central chi square distribution of the probability of detection of the DBZPTI. This method is defined by
the \(\chi^2\) variable with 2K degrees of freedom, with K denoting the number of non-coherent summations.

Using the equations presented in [10]:

\[
\lambda_K = K \lambda_{DBZP} ; V_{th}' = \frac{\beta}{\sigma_{DBZP}} ; \rho_c = \frac{\lambda}{\sigma_{DBZP}} \quad (5.9a)
\]

\[
P_{fa,K}(V_{th}') = \exp \left\{ -\frac{V_{th}'}{2} \right\} \sum_{i=0}^{K-1} \frac{1}{i!} \left( \frac{\beta'}{2} \right)^i \quad (5.9b)
\]

\[
P_{d,K}(V_{th}') = Q_K \left( \sqrt{K \rho_c} ; \sqrt{V_{th}'} \right) \quad (5.9c)
\]

According to equations 5.9 is possible to compute the probability of detection of the DBZPTI method.
Since the degrees of freedom of the \(\chi^2\) distribution are 2K, a new noncentrality parameter, \(\lambda_K\) is created
along with some new auxiliary parameters like \(\beta'\) and \(\rho_c\). Just like the classical acquisition, first the
threshold, \(V_{th}'\) is computed by equation 5.9b where \(P_{fa,K}(V_{th}') = 1.0611 \times 10^{-8}\). Then the probability of
detection given by equation 5.9c is computed by a Marcum Q function of order K. This process is made
for a range of \(C/N_0\) between 35 and 45 dBHz and the result is shown in figure 5.5:

Using exactly the same parameters in 5.1 we can see that the probability of detection (for a \(C/N_0 = 44\) dBHz) is very similar to the classical acquisition, \(P_d = 1\). This means that this method for our reference
has the same performance of the classical acquisition but is many times faster. Actually the performance
for this range of \(C/N_0\) values is very similar as we can see in figure 5.6. So this method in terms of
performance cannot be considered worse than the classical acquisition, but in terms of speed is many
times faster.
Since this method is considered for a future implementation, for reference this method is also extremely fast for the acquisition of weak signals. For a signal with a $C/N_0 = 27 \text{ dBHz}$, the probability of detection is 0.93 for a $K = 38$ and $P_{fa} = 10^{-3}$ [22]. For these parameters it takes 250 seconds for the DBZPTI acquisition of the signal. So this method is also very useful for applications with weak signals.
5.3.2 Simulation Results

To validate our theoretical analysis a Matlab simulation is made based on the algorithm described in section 4.4.

To achieve somewhat reliable numbers $5 \times 10^3$ results were taken and a histogram was plotted for some values of $C/N_0$. The most important value is the reference, $C/N_0 = 44$ dBHz, the PDF for this value is shown in figure 5.7.

![Figure 5.7: Simulated PDF for the DBZPTI with a $C/N_0 = 44$ dBHz and $P_{cell}^{fa} = 1 \times 10^{-4}$](image)

Using figure 5.7 it is possible to see that the probability of detection is very close to one, just like the classical acquisition and the theoretical predictions.

The probability of detection was simulated for a range of $C/N_0$ between 35 and 45 dBHz and the result is shown in figure 5.8:

![Figure 5.8: Theoretical and Simulated Performance of the DBZPTI Acquisition Method for, $P_{cell}^{fa} = 1 \times 10^{-4}$](image)

63
Just like the classical acquisition the theoretical results are very near the simulated ones. The maximum deviation is of 2\% which brings trust for the theoretical analysis made.

Also with this figure we see that the DBZPTI is as reliable as the classical acquisition (but many times faster).

### 5.3.3 Real Signals Simulation Results

To test the DBZPTI algorithm real signals acquired with the receiver are going to be used. The process is the same as the simulations above, the only thing that changes is the incoming signal and the sampling frequency from 16 MHz to 26 MHz. This increase in the sampling frequency is negligible when analyzing the peaks.

![DBZPTI acquisition with the use of real signals](image)

**Figure 5.9:** DBZPTI acquisition with the use of real signals

Figure 5.9 shows that the DBZPTI real signal has a stronger peak than the classical acquisition (figure 5.4), 80\% stronger. A more detailed analysis is going to be made in section 5.6.

Since this algorithm is mainly targeted for a software receiver it is relatively important to know how much time took the algorithm implemented in Matlab to acquire the signal. On a fourth generation Intel Core i7-4700MQ with 2.0 Intel Turbo Boost and Clock Speed of 2.40 / 3.40 Turbo GHz the total time to acquire the signal was 1.388 seconds. In contrast, the classical acquisition (considering a hardware implementation) took 32 minutes.
5.4 Delay and Multiply

5.4.1 Delay and Multiply as a filtering method

As seen in the previous chapter the delay and multiply has a big drawback relatively to its fast MAT which is the multiplication by a delayed version of the incoming signal. This multiplication introduces a squared noise component which reduces drastically the performance, as seen in [25].

For a $C/N_0 = 44$ dBHz the peak is not clearly visible amid the noise, so there is no way that this method can compete with any of the others, taking in account that all present a probability of detection of 1 and the peak amplitude is much stronger than the noise. Because of these limitations a strategy was created where this method would be useful.

Despite the fact that the peak is not visible in the output we now have some information regarding all the code delays searched.

A first look at the graphic of figure 5.10 does not make sense because no peak is visible but it can be seen as a help to choose in what order the code delays are to be searched. In the classical acquisition method the code delays were searched from the first to the last delay, but with this information the first code delay to be searched is the one with the highest magnitude in figure 5.10, the second is the second with the highest magnitude and so on until the decision variable from the classical acquisition exceeds the defined threshold, which means we are in the presence of the right delay for the correct Doppler frequency.

With this methodology the delay and multiply is used just to filter the order in which the code delays are chosen in the classical acquisition. If the code delays to be searched before finding the right one are few then it is worth using this method.
Now we can compute how much time is saved by using the delay and multiply filtering methodology described in figure 5.11.

\[ MAT_{DM} = \text{Code Bins} \times \text{Coherent Integration time} + \text{Delays searched} \times \text{Frequency Bins} \times \text{Coherent Integration time} \]  

(5.10)

From equation 5.10 we have the time of the delay and multiply method plus the time of the classical acquisition. The delay and multiply takes a fixed time, for the values we have been using it takes, \( \frac{4992}{0.3456} \times 4 \times 10^{-3} \approx 47 \) seconds. Now the total time varies depending on how many code delays are searched in the classical acquisition. Some simulations are going to be made so we can have an idea whether this method is worth of not.

### 5.4.2 Simulation Results

For our reference \( C/N_0 \), \( 5 \times 10^3 \) tests were made. All the number of cells searched before finding the code phase and Doppler frequency were stored and an histogram was made so we can have a better idea of how many cells are going to be searched before finding the real peak.

Using figure 5.12 we can see that half the time approximately 2000 cells are searched before finding the right values, this means by equation 5.10 a MAT of 47 seconds + \( 2000 \times 80 \times 4 \times 10^{-3} = 687 \) seconds \( \approx 11.5 \) minutes. With the previous result we have a decrease of 1/3 comparing to the classical acquisition search.
Figure 5.12: Cells searched in function of time with the Delay and Multiply filtering methodology for a $C/N_0 = 44$ dBHz

For a $C/N_0 = 35$ dBHz, which is the worst case in our range we have the following distribution in terms of cells searched:

Figure 5.13: Cells searched in function of time with the Delay and Multiply filtering methodology for a $C/N_0 = 35$ dBHz

Using figure 5.13 it is possible to see that 50 % of the time 2320 cells are searched which means approximately a 15 % increase in comparison with the $C/N_0 = 44$ dBHz. This means an acquisition time of 13 minutes instead of 11 which is still a very good result. So for our range of $C/N_0$ the acquisition time varies from 11 to 13 minutes for half the time.

After the delay and multiply filtering the process is the same as the classical acquisition, so the study has already been made in section 5.2.

The great advantage of this filtering is the classical acquisition performance where the time is reduced to a third. Regarding the classical acquisition, when we are considering the mean acquisition time we are assuming that only half the cells are searched, so the code phase bins are 5920, the delay and multiply filtering reaches this number of cells searched only 12 % of the time for our reference, $C/N_0 = 44$ dBHz.
5.4.3 Real Signals Simulation Results

The delay and multiply filtering was applied to the real signals and the results are very good. The number of cells to be searched before finding the right Doppler and code delay are 418, which results in an acquisition time of:

\[ \text{Acquisition Time DM Real Signals} = 181 \text{ seconds} \approx 3 \text{ minutes} \quad (5.11) \]

The acquisition output is the same as the one in section 5.2, because the Delay and Multiply is just a filtering method to the classical acquisition.

5.5 Dual Sideband, CBOC as BPSK

5.5.1 Theoretical Performance Evaluation

MAT

For the calculations of the MAT of this method we are going to consider that both the upper and lower sidebands are being correlated at the same time, so instead of needing two correlators we are going to use four.

Since we are treating the CBOC(6,1,1/11) as a BPSK(1), as seen before in chapter 2, the maximum delay can be 1 chip instead of 0.3456, so the MAT will be the same as the classical acquisition but with the difference regarding the maximum code delay:

\[ \text{Total number of code bins} = 4092 \quad (5.12a) \]
\[ \text{MAT}_{\text{Dual}} = 655 \text{ seconds} \approx 11 \text{ minutes} \quad (5.12b) \]
Comparing with the classical acquisition we see a 1/3 reduction; the only thing left is to evaluate its performance.

Performance

The performance of this method can be approximated by the equations used to evaluate our reference acquisition, but with a little degradation (approximately 1dB). This degradation comes from not using all the band and the code may not be all aligned. So for our reference $\frac{C}{N_0}$ we still expect a probability of detection very close to 1.

First we are going to see what the real degradation is comparing our simulated results and then evaluate the performance of this method.

5.5.2 Simulation Results

![Figure 5.15: Classical Acquisition theoretical performance with the simulation results for the Dual Sideband, CBOC as BPSK](image)

When comparing the classical acquisition theoretical performance (already validated by simulations in section 5.2) with the simulated results for the Dual Sideband, CBOC as BPSK we can see a decrease in performance but not significant. The mean deviation is of 4% in terms of probability of detection, this number corresponds to approximately a 0.5 dB degradation when comparing with the classical acquisition. So this method too gives a probability of detection very close to one for our reference $\frac{C}{N_0}$.

Taking into account the previous conclusion the performance for a $P_{fa}^{ref} \approx 1.0611 \times 10^{-8}$ should be very similar to the classical acquisition with a 1dB degradation:
5.5.3 Real signal Simulation Results

By figure 5.17 the peak is very distinct from the noise just like all the others methods, so it is also a good alternative in terms of performance. Besides the peak is slightly lower in terms of magnitude when comparing with the reference. This result agrees with our theoretical evaluations, since we are not using all the band and there may be a misalignment in the correlation.
5.6 Results Comparison

When analyzing the different MAT it is possible to verify that all the methods are quicker than our reference which can be considered good results taking into account all the existing limitations:

<table>
<thead>
<tr>
<th>Method</th>
<th>MAT (Minutes)</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical Acquisition</td>
<td>32</td>
<td>-</td>
</tr>
<tr>
<td>Delay and Multiply</td>
<td>11</td>
<td>34 %</td>
</tr>
<tr>
<td>Dual Sideband</td>
<td>11</td>
<td>34 %</td>
</tr>
</tbody>
</table>

Table 5.2: MAT for all the methods

According to table 5.2 both the new methods guarantee a 34% reduction of acquisition time when comparing with the reference. But since the MAT of the delay and multiply varies with the $C/N_0$, maybe for stronger signals its MAT can be inferior. For the real signal the result is very good, the acquisition time for the real signal is 3 minutes which means a 90% reduction. Only with much further testing this result can be definitive but it opens a very interesting door.

Also for all these methods the performance is exactly the same taking into account our $C/N_0$ reference, so we can just evaluate the algorithms in terms of their MAT.

It is not fair to place the MAT of the DBZPTI in table 5.2 because they serve different purposes. So for a fair comparison the reference for the DBZPTI is the Parallel Code Phase Search presented in section 4.3. The acquisition times presented are the total and not mean and are the results presented by Matlab when acquiring the real signal:

<table>
<thead>
<tr>
<th>Method</th>
<th>AT (Seconds)</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Code Phase Search</td>
<td>12.75</td>
<td>-</td>
</tr>
<tr>
<td>DBZPTI</td>
<td>2.2</td>
<td>91 %</td>
</tr>
</tbody>
</table>

Table 5.3: Acquisition times for the DBZPTI and Parallel Code Phase Search

Again the results of the studied methods are better than the reference. As we were expecting the DBZPTI is a very fast method with good performance, comparable to the classical acquisition. This proves that without any limitation the DBZPTI seems to be the best candidate for a GNSS receiver.

Also for the real signal the peaks had different magnitudes:

<table>
<thead>
<tr>
<th>Method</th>
<th>Peak Magnitude(for similar noise floors)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical Acquisition</td>
<td>$2.5 \times 10^4$</td>
</tr>
<tr>
<td>DBZPTI</td>
<td>$4.37 \times 10^4$</td>
</tr>
<tr>
<td>Dual Sideband</td>
<td>$2.2 \times 10^4$</td>
</tr>
</tbody>
</table>

Table 5.4: Peak Magnitude for all the methods

Taking in account 5.4 the magnitude of the DBZPTI is the biggest by almost two times. This fact did not occur in any of the simulations and was an unexpected result. Unexpected until a certain point, it is known that the resolution (in terms of code phases) for the DBZPTI is much higher than the classical one which results in an increased sensitivity and for this signal this increased resolution was significant.
In the simulations, the delay introduced to the incoming signal was of 1.2ms. Thus, both resolutions of the techniques were enough to search the right cell, but for a more specific delay (ex: 1.238948ms) the method with a better resolution would give much better results. Also there may be the case of a secondary bit transition in which the classical acquisition is more sensitive. As expected, the peak magnitude of the Dual sideband in slightly lower than the classical acquisition. This can be explained by the use of non-coherent combination, so some power of the signal was not used.
Chapter 6

Conclusions

This chapter presents a summary of the conclusions reached in each chapter and makes some recommendations for future work.

6.1 Conclusions

The main objective of this thesis, stated in chapter 1, was to study one algorithm eligible to accelerate the acquisition process in the DEIMOS GRIP receiver without major modifications, which means that the use of FFT’s in the acquisition process was not possible. From here a full study of the acquisition process was made so this objective could be accomplished.

Chapter 2 presented the study of the GNSS signals with focus on the GPS and Galileo signals. From here we could conclude that the Galileo signals bring many advantages but present many challenges in the acquisition process, more specifically, because of their complexity and length, their acquisition in a short time proves to constitute a very difficult challenge, furthermore when the use of FFT’s is not an option.

To overcome the problems presented in the previous chapter, chapter 3 presents a full description of the acquisition process and all the methods we could use. From this chapter the conclusion is that the acquisition process is a very complex process with many variables and with an infinite range of options in terms of the algorithm design; all options bringing different results. So from all these options some choices were made and the study of the (considered) most promising algorithms is made on the next chapter.

Chapter 4 introduces a description of all the considered algorithms and in the end the ones that are going to be evaluated were chosen:

- **Classical Acquisition**, as the reference
- **Double Block Zero Padding Transition Insensitive, DBZPTI**
- **Delay and Multiply**
- **Dual Sideband, CBOC as BPSK**
In the above list there is a FFT based algorithm, besides the fact that the DBZPTI cannot be implemented in the DEIMOS GRIP receiver without some major changes. Nevertheless, it is full studied so it could serve as a reference for the future. From all the algorithms described this seems to be the most interesting of them all.

In chapter 5 the full study of all the algorithms is carried out, a theoretical study is made in terms of the mean acquisition time and its probability of detection for different $\frac{C}{N_0}$. Tests with simulated signals were applied to all the algorithms to validate the theoretical assumptions. And in the end the use of a real signal from the DEIMOS GRIP receiver was applied to all the methods. From this chapter we could conclude that:

- The delay and multiply by itself is not a good solution due to its low sensitivity even in the presence of strong signals. Because of this fact this algorithm was "recycled" as a filtering method for the classical acquisition.

- The DBZPTI is by far the best method in terms of speed outperforming all the other ones. It was compared with another FFT based method (parallel code phase search) and it is still 12 times faster. In terms of sensitivity it is very similar to our reference.

- The delay and multiply as a filtering method has the same sensitivity as our reference but the dual sideband algorithm has also a very similar sensitivity. The difference between both methods regarding sensitivity is negligible. With our simulated signals they both present the same mean acquisition time, but the delay and multiply MAT is somewhat variable where the MAT of the dual sideband is not; so one can say that the dual sideband method barely outperforms the delay and multiply.

- With the use of a real signal the acquisition time of the delay and multiply filtering method was 3 minutes whereas the MAT of the Dual Sideband was 11 minutes. This is just a single test and the result does not have much strength but is still a result. So with this result maybe it was worth trying the delay and multiply on the receiver and draw some more conclusions.

From all the results one can consider the Dual Sideband method as a more stable candidate (because the MAT is constant does not vary with the reduction of the $\frac{C}{N_0}$ and the delay and multiply is a more unpredictable candidate. Nonetheless the delay and multiply seems like a method worth trying.

6.2 Future Work

For the future it would be good to explore more the Delay and Multiply Filtering method presented. This method was just an idea but taking into account the results it would be worth to exploit it with real data and put it to the test. Only with further testing one can really evaluate the real strength of this methodology. Also it is a method that is not deeply analyzed and it would be worth studying, because without the use the FFT’s it is the only one that eliminates the frequency dependency making it really fast.
Bibliography


Appendix A

Deimos Grip

This appendix briefly describes the GNSS receiver that was used to obtain the real signals that were processed in chapter 5.
GRIP is a new concept of GNSS Receiver that allows full access to internal registers on a powerful and modular board supporting advanced signal tracking and optimised interfaces.

GRIP consists of a main board including the receiver core, FLASH and SDRAM, local oscillator and various interfaces, as Ethernet, PCIe, JTAG and general purpose I/O (GPIO) pins (FMC connector, header pins), and a second plug-in board which includes the RF core and CAN bus interface.

GRIP contains up to 160 real correlators and 16 fully flexible channels for dual-system and single-frequency configurations, supporting real-time Galileo E1 CBOC, Galileo E5 AltBOC, GPS L1 C/A, GPS L1C and GPS L5 signals. Dual-frequency also supported with use of external front-end.

GRIP has two companion software applications for receiver monitoring and control: GRIP-CENTER, a multi-platform Java-based application which allows local/remote output visualization and analysis as well basic receiver control; and GRIP-MON, which allows the recording of internal receiver signals and provides read/write access to the GRIP’s hardware registers for advanced control and receiver monitoring, through a MATLAB extension and toolbox.

GRIP is supplied with the register data sheet, allowing users to develop software and uploading it to the receiver. Such capability allows building an entire receiver and its application software from scratch.

**Specifications**
- Fully configurable FPGA-based (Xilinx Virtex-6) 16-channel Galileo/GPS receiver
- Integrated single frequency RF front-end and support for external dual-frequency RF front-end
- Digital IF/baseband signals with up to 4-bits
- Embedded dual-processor (MicroBlaze)
- BPSK/BOC/CBOC/AltBOC signal processing
- Access to internal receiver observables and configuration parameters
- Advanced software package (receiver monitoring and control, measurement post-processing, navigation)
- Interfaces: PCIe, Ethernet, CAN, UART, JTAG, GPIO (FMC and header pins), SMA
- Robust manufacturing and modular design

**Applications**
- Applications requiring hybridization with external sensors (accelerometers, gyroscopes, barometer, magnetometer, Wi-Fi, LIDAR, etc.) with various integration approaches (up to deep integration)
- Performance assessment of high precision applications using BOC/CBOC/AltBOC signals
- R&D projects
- Validation and testing
- Education: go inside a hardware receiver!
- Signal-in-Space understanding and verification
- Post-processing of GNSS measurements for multipath and interference analysis
The GRIP-CENTER software interfaces the GRIP receiver to download, in real-time, elementary data and displaying it on a dedicated graphical user interface. It allows visualisation and analysis of receiver observables as well as display of the navigation solution on a map. The software includes binary to RINEX conversion capability.

**Specifications**

- Interface over Ethernet using either a local or remote connection (LAN or internet);
- Records basic receiver observables (pseudoranges, carrier phase, navigation data, position);
- Visualise user position in map, receiver observables and their statistics, $C/N_0$, and satellite skyplots;
- On-line and post-processing modes: display in real-time or replay stored data.
- RINEX v3 file generation.

**Example**

The user connects GRIP to GRIP-CENTER installed on a PC via Ethernet. For debugging purposes, the user can check if the receiver is working as expected by inspecting main observables in real-time as well as issuing basic commands.

Receiver data can be recorded for later use. For example, different files from different recording sessions can be visualised on the same figure, allowing easy comparison of trends or other unique events. For each session the user may generate separate RINEX files.

The figures below show in real-time the overlay of the position computed by GRIP over a map as well as a carrier to noise density ratio histogram.

**Requirements**

- GRIP
- PC running Microsoft® Windows XP OS with 2GB RAM, Intel Pentium 4 @3GHz or better
- Available Ethernet port
- Internet connection for map visualisation
GRIP-MON is an advanced receiver monitoring and control software application for real-time access to receiver registers via PCIe interface. It includes a MATLAB® extension which allows real-time monitoring and control of internal receiver signals and registers.

GRIP-MON provides a user with direct memory access to GRIP memory areas. It supports two monitoring methods: Data Recording to file and MATLAB Requests (read/write).

The user is supplied with a list of all accessible GRIP registers and their read/write permissions, allowing the modification of many configuration parameters (integration times, loop discriminators, loop bandwidths, among others).

**Specifications**
- Data recording of GRIP observables to file
- Synchronisation with GRIP events such as integration epochs, measurement epochs, PPS
- Read/Write access to GRIP registers
- Complete extension set to allow receiver monitoring via MATLAB
- Intuitive interface

**Requirements**

- **GRIP**
  - PC running Microsoft® Windows XP OS with 2GB RAM, Intel Pentium 4 @3GHz or better and power supply with at least 450Watt
  - MATLAB, version R2009b or higher
  - Available PCIe slot

**Example**

When GRIP-MON is run, it first loads its kernel driver and starts listening for events.

Using the GRIP-MON MATLAB extension, it is possible to directly access GRIP memory/registers using read/write/queue requests. A list of commands can be written in MATLAB language with corresponding replies sent to the caller functions (in MATLAB).

The following screenshot shows an example of GRIP-MON and its MATLAB® extension being used to monitor GRIP’s internal data (signal level histogram, Doppler estimates, etc.).

The tool also allows, for example, the use of a navigation filter integrated with inertial sensors, running in MATLAB environment and in real-time. Given the command nature of the tool and the flexibility of MATLAB, the possibilities for the use of the tool are countless.