Comparative Performance Analysis of Scalar and Vector GNSS Receiver Architectures

António Pedro Bretes Negrinho

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. Fernando Duarte Nunes

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. Fernando Duarte Nunes
Member of the Committee: Prof. Fernando Manuel Gomes de Sousa

November 2019

In loving memory of my late grandfather José Cardoso Bretes

We miss you dearly with each passing day

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I would like to thank first and foremost my thesis supervisor Prof. Fernando Nunes, for whom I have the greatest respect and admiration, for sparking my interest on satellite navigation. Thank you for always lending a helping hand when I most needed it and making sure I stayed on track during these past 8 months. I really enjoyed all our conversations at your office and hope that, perhaps in the future, we can work together again.

I would like to thank my mum and dad for their unconditional love that has carried me through miles and miles. I can't thank you both enough for everything you have done for me and for making me the man I am today (even though I will always be your little kid). I am blessed to be your son and hope I can make you both proud. I thank the rest of my family as well: João and Laura, for the blast I had growing up with you; my aunt Isabel who is like a second mother to me; both my grandmothers, for all their wisdom and love and Marta, aunt Teresa and uncle Bernardo for their unyielding support and interest on my scientific education. To my grandfather... I'll see you around old man, may the Force be with you.

A very special thank you goes out to my wonderful girlfriend Sara, who I love so much. Thank you for being my safe place, my best friend and for always caring and supporting me in a way only you know how. This wouldn't have been possible without you. Of course, I extend my gratitude to Paula and Luís for receiving me into their home with open arms and always treating me as kin.

I also thank my flatmates Maria, Pedro and Filipe for being naive enough to let me live in the same apartment as them and then resisting the urge to stab me once they realized the mistake they had made. I love you three so much.

To all my friends back in my hometown (you know who you are), I am so thankful for everything we've lived together and for knowing we'll always lookout for one another. I am so proud of the men and women we're becoming. You guys are the best.

I would like to thank all the wonderful people I got to meet at Instituto Superior Técnico, in particular my teammates from Fener, for all the camaraderie, the great memories made on the pitch and the close bonds we have formed. A very special mention goes out to Pedro Trindade with whom I became close friends since day one at university.

Last but not least, I would like to thank Prof. Paulo Quintino. I am not sure you realize how much of an impact you had on me, in shaping my interest for science. I do not know where my journey in engineering will take me but it certainly started with a 15 year old me in one of your classes. And for that, I am extremely thankful.

Resumo

A presente tese visa comparar o desempenho entre arquiteturas escalares e vetoriais de recetores GNSS, pricipalmente em relação ao seguimento de código. Primeiramente, é efetuada uma revisão bibliográfica, acompanhada de uma descrição matemática do processo de seguimento de código e do efeito de multi-percurso. As principais diferenças entre arquiteturas escalares e vetoriais são enfatizadas. São descritos dois algoritmos de estimação para navegação: o algoritmo de mínimos-quadrados e o filtro de Kalman extendido (EKF), assim como a sua integração nos recetores escalares e vetoriais. Todas as arquiteturas e algoritmos dos recetores são implementados em MATLAB e simulados em vários cenários e com várias dinâmicas: pedestre, carro, aeronave. Utilizam-se diferentes modulações: BPSK, BOCs(1,1) e CBOC(6,1,1/11), sendo que as duas últimas são utilizadas no sistema Galileo. Um novo discriminador de código, baseado num banco de correladores, foi, também, implementado para processamento do sinal CBOC(6,1,1/11), algo impossível com um discriminador tradicional. Os resultados das simulações mostram que as arquiteturas vetoriais apresentam melhor desempenho que as escalares em praticamente todas as simulações, especialmente em cenários de forte atenuação, independentemente do algoritmo de estimação considerado. Por fim, foi implementado um algoritmo vetorial com seguimento de código e frequência. Obtiveram-se parâmetros para desempenho ótimo e analisou-se a robustez relativamente a baixa potência de receção.

Palavras-chave: Seguimento vetorial, Seguimento escalar, VDFLL, VDLL, GNSS

Abstract

This thesis addresses the performance comparison between scalar and vector architectures for GNSS receivers, regarding mainly code tracking. First, a review on the important literature is undertaken, alongside a mathematical description of the code tracking process and multipath effect. The main differences between scalar and vector receiver architectures are also highlighted. Two navigation estimation algorithms are described: the least-squares algorithm and the extended Kalman filter (EKF), as well as their integration in scalar and vector receivers. All the receiver architectures, and algorithms are implemented in MATLAB and simulated in a variety of signal reception scenarios and dynamics: pedestrian, car and aircraft. Different signal modulations are utilized: BPSK, BOCs(1,1) and CBOC(6,1,1/11), these last two being used by the Galileo system. Additionally, a new code discriminator based on a bank of correlators was devised for processing of the CBOC(6,1,1/11) signal, which was not possible with the traditional one. Simulation results demonstrate that vector architectures outperform scalar ones in almost all simulations, especially in scenarios of severe attenuation, regardless of the estimation algorithm used. Additionally, a vector algorithm, with simultaneous code tracking and frequency tracking, was implemented. Some simulations where undertaken, culminating in a choice of tuning values for optimal performance, as well as an analysis of robustness to poor signal power.

Keywords: Vector tracking, Scalar tracking, VDFLL, VDLL, GNSS

Contents

Acknowledgments . . . vii

Resumo . . . ix

Abstract . . . xi

List of Tables . . . xvii

List of Figures . . . xix

Nomenclature . . . xx

Glossary . . . xxiii

1 Introduction 1

1.1 Motivation . . . 1

1.2 Objectives . . . 2

1.3 Thesis Outline . . . 2

2 Theoretical Background 5

2.1 The principle of satellite navigation . . . 5

2.2 Code tracking . . . 6

2.3 Multipath . . . 11

2.4 Vector tracking architectures . . . 13

3 Scalar and vector receiver algorithms analysis 17

3.1 Least-Squares algorithm . . . 17

3.1.1 Integration of the Least-Squares algorithm in a scalar receiver . . . 19

3.1.2 Integration of the Least-Squares algorithm in a vector receiver . . . 19

3.2 Extended Kalman filter algorithm . . . 21

3.2.1 Dynamics and observations models for the EKF algorithm . . . 24

3.2.2 Integration of the EKF algorithm in a scalar receiver . . . 28

3.2.3 Integration of the EKF algorithm in a vector receiver . . . 28

4 Scalar receiver simulation results 31

4.1 Scalar receiver with least-squares algorithm . . . 31

4.1.1 Standard scenario simulation results . . . 31

4.1.2 Shadowing scenario simulation results . . . 35

4.1.3 Multipath scenario simulation results 37
4.1.4 Signal comparison simulation results 38
4.2 Scalar receiver with extended Kalman filter algorithm 40
4.2.1 Standard scenario simulation results 40
4.2.2 Shadowing scenario simulation results 43
4.2.3 Multipath scenario simulation results 44
4.2.4 Signal comparison simulation results 45

5 Vector receiver simulation results **47**
5.1 Vector receiver with least-squares algorithm 47
5.1.1 Standard scenario simulation results 47
5.1.2 Shadowing scenario simulation results 51
5.1.3 Multipath scenario simulation results 53
5.1.4 Signal comparison simulation results 54
5.2 Vector receiver with extended Kalman filter algorithm 55
5.2.1 Standard scenario simulation results 55
5.2.2 Shadowing scenario simulation results 57
5.2.3 Multipath scenario simulation results 59
5.2.4 Signal comparison simulation results 60
5.3 Overall comparison 61

6 Vector delay/frequency lock loop characterization and results **65**
6.1 Code and carrier tracking 65
6.2 VDFLL characterization 67
6.3 VDFLL results 70

7 Conclusions **73**
7.1 Achievements 73
7.2 Future Work 74

Bibliography **77**

A GNSS signals characterization **81**
A.1 **BPSK** signal 81
A.2 **BOCs(1,1)** signal 82
A.3 Galileo **CBOC(6,1,1/11)** signal 83

B Generation of random correlated Gaussian noise vectors **85**
B.1 Generation process 85
B.2 Early-Prompt-Late correlators 86

C Generation of a normalized code discriminator from a bank of correlators **89**

D Characterization of an imperfect oscillator **95**

List of Tables

5.1 Scalar vs Vector receiver performance with varying number of satellites (least-squares). . 49

5.2 Scalar vs Vector receiver performance with C/N$_0$ degradation (least-squares). 50

5.3 Scalar vs Vector receiver performance in a shadowing scenario (least-squares). 53

5.4 Scalar vs Vector receiver performance in a multipath scenario (least-squares). 54

5.5 Scalar vs Vector receiver performance for several modulations and code discriminators (least-squares). 55

5.6 Scalar vs Vector receiver performance with varying number of used satellites (EKF). . . . 56

5.7 Scalar vs Vector receiver performance with C/N$_0$ degradation (EKF). 57

5.8 Scalar vs Vector receiver performance in a shadowing scenario (EKF). 59

5.9 Scalar vs Vector receiver performance in a multipath scenario (EKF). 59

5.10 Scalar vs Vector receiver performance for several modulations and code discriminators (EKF). 60

5.11 Global Vector vs Scalar receiver architecture comparison, with the least-squares and EKF algorithms, using BPSK modulation (GPS C/A). 62

5.12 Global Vector vs Scalar receiver architecture comparison, with the least-squares and EKF algorithms, using different modulations. 63

List of Figures

2.1 Block diagram of one channel of the conventional GNSS receiver. 6

2.2 Carrier frequency wipe-off scheme [8]. 7

2.3 Generation of the early-late inphase and quadrature components [8]. 8

2.4 NELP discriminator response for a BPSK signal. 10

2.5 NELP response for various early-late spacings and different modulations. 11

2.6 Multipath envelopes for several modulations. 13

2.7 Comparison of the scalar and vector tracking architectures [5]. 14

3.1 Architecture of a scalar receiver [26]. 19

3.2 Architecture of a vector receiver with ideal carrier tracking [26]. 21

3.3 Kalman filter operation [31]. 23

3.4 PV model using double integrated Brownian motion [31]. 25

3.5 Receiver's clock state model [34]. 26

4.1 Results for the scalar receiver with varying number of satellites (least-squares). 32

4.2 Results for the scalar receiver with varying correlation interval T (least-squares). 33

4.3 Results for the scalar receiver with varying C/N$_0$ in all satellites (least-squares). 34

4.4 Illustration of an urban shadowing scenario (urban canyon) [38]. 35

4.5 Shadowing scenario results for the scalar receiver (least-squares). 36

4.6 Multipath scenario results for the scalar receiver (least-squares). 38

4.7 Signal performance comparison for a scalar receiver (least-squares). 39

4.8 Results for the scalar receiver with varying number of satellites (EKF). 40

4.9 Results for the scalar receiver with varying correlation interval T (EKF). 41

4.10 Results for the scalar receiver with varying C/N$_0$ in all satellites (EKF). 42

4.11 Shadowing scenario results for the scalar receiver (EKF). 43

4.12 Multipath scenario results for the scalar receiver (EKF). 44

4.13 Signal performance comparison for a scalar receiver (EKF). 45

5.1 Set of simulation configurations. 47

5.2 Results for the vector receiver with varying number of used satellites (least-squares). . . . 48

5.3 Results for the vector receiver with varying correlation interval T (least-squares). 49

5.4 Results for the vector receiver with varying C/N$_0$ in all satellites (least-squares). 51

5.5 Shadowing scenario results for the vector receiver (least-squares). 52
5.6 Multipath scenario results for the scalar receiver (least-squares). 53
5.7 Signal performance comparison for a vector receiver (least-squares). 54
5.8 Results for the vector receiver with varying number of used satellites (EKF). 55
5.9 Results for the vector receiver with varying correlation interval T (EKF). 56
5.10 Results for the vector receiver with varying C/N$_0$ in all satellites (EKF). 57
5.11 Shadowing scenario results for the vector receiver (EKF). 58
5.12 Multipath scenario results for the vector receiver (EKF). 59
5.13 Signal performance comparison for a scalar receiver (EKF). 60

6.1 VDLL and VDFLL architectures [24]. 65
6.2 VDFLL rms error variation as a function of γ_c and γ_f. 70
6.3 Time evolution of the VDFLL position vector errors with varying C/N_0. 71

A.1 GPS C/A (BPSK) signal characterization. 82
A.2 BOCs(1,1) signal characterization. 83
A.3 BOCs(1,1) signal characterization. 84

C.1 Target S-curve. 89
C.2 Bank of inphase/quadrature correlators [45]. 90
C.3 S-curves obtained with a bank of correlators. 93
C.4 NELP *vs* bank of correlators performance for an initial code delay of $0.4T_c$. 94

D.1 Allan variance for TCXO (temperature-compensated XTAL oscillator), OCXO (ovenized XTAL oscillator) and rubidium oscillators. 96

Nomenclature

Greek symbols

Δ	DLL early-Late spacing
ϵ	Code delay error
γ_c	Code discriminator feedback gain
γ_f	Frequency discriminator feedback gain
ω_d	Doppler frequency (in radians)
Φ	State transition matrix
ρ	Pseudorange
τ	Time delay
τ_{code}	Spreading signal code delay
θ, ϕ	Phase offset angles
$\tilde{\omega}$	VDFLL frequency error (in radians)

Roman symbols

C/N$_0$	Carrier-to-noise ratio
$\widetilde{R}_k$	Observations noise covariance matrix (at instant k)
A_n	Amplitude (of signal n)
B	Bandwidth
$D_c(\epsilon)$	Code discriminator output function
$D_f(\tilde{\omega})$	Frequency discriminator output function
E	Expected value
G	Geometry matrix
$G_X(f)$	Power specrtal density (of signal X)

H_k	Observations matrix (at instant k)
K_k	Kalman gain (at instant k)
Q_k	Dynamics noise covariance matrix
$R_X(t)$	Autocorrelation function (of signal X)
T	Correlation interval
T_c	Chip time
$X(t)$	Encoded data signal
$\tilde{f}_n$	VDFLL frequency error (in Hz)
c	Speed of light in vacuum
t	Time

Superscripts

$\cdot$	First order time derivative
$\wedge$	Estimate
-1	Inverse
T	Transpose

Glossary

AWGN	Additive White Gaussian Noise
BOCs	Binary Offset Carrier (sine-shaped)
BPSK	Binary Phase Shift Keying
C/A	Coarse/Acquisition
CBOC	Composite BOC
DLL	Delay Lock-Loop
ECEF	Earth-Centered Earth-Fixed
EKF	Extended Kalman Filter
FLL	Frequency Lock-Loop
GDOP	Geometric Dilution of Precision
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IST	Instituto Superior Técnico
LPF	Low-Pass Filter
NCO	Numerically-Controlled Oscillator
NELP	Normalized Non-Coherent Early-Late Power
PLL	Phase Lock-Loop
PSD	Power Spectral Density
PVT	Position, Velocity and Time
RMS	Root Mean Square
VDFLL	Vector Delay/Frequency Lock-Loop
VDLL	Vector Delay Lock-Loop.

Chapter 1

Introduction

This chapter serves as an introduction to the present thesis, on the research topic ”Comparative Performance Analysis of Scalar and Vector GNSS Receiver Architectures”. The motivation behind the presented topic is addressed, followed by a listing of the proposed objectives and an outline of this thesis' structure.

1.1 Motivation

Originally devised for military applications, the Global Positioning System (GPS) was the first Global Navigation Satellite System (GNSS) to be put in operation by the United States of America in the 1970's [1], [2]. Since its inception, other similar systems have been put in operation, such as the Glonass (Russia), the BeiDou (China) and the Galileo (European Union). These systems have been utilized for civilian ends for several years now, ranging from user positioning, precision agriculture, traffic management and civil aviation, among others. This was made possible through the integration of GNSS receivers in almost any electronic device, such as cellphones, personal computers, tablets and even cars, such that today we consider access to satellite navigation as an essential asset to our everyday lives [3].

Because of the widespread usage of GNSS, its shortcomings must be tackled in order to ensure that these type of systems are robust and trustworthy so that other services and activities can rely on them. The first shortcoming is the power of the received signals, which can be as low as 10^{-16} watts [4]. This leads to easy interference of signal reception, either intentional or not, implying receivers must be able to operate with very attenuated signals. Another problem in signal reception is the blockage of incoming signals by buildings, trees or any other sort of object, natural or man-made. A GNSS receiver needs to maintain line-of-sight with the satellites whose signals it is tracking and the loss of one or more incoming signals negatively affects the performance of a receiver in a drastic way. This prolonged loss of signal tracking will usually lead receivers to re-do the time consuming task of re-acquiring the lost signals [2], [3]. Signal reception can also be affected by other phenomena such as ionospheric scintillation, multipath or high receiver dynamics, all of which degrade the receiver's performance.

A method to overcome these hindrances in signal reception is the utilization of vector architectures in GNSS receivers. There is some literature that shows vector receivers have been able to carry out the task of tracking severely attenuated signals, high receiver dynamics and recover from the momentary blockage of incoming signals [1], [3], [5], [6], [7]. This thesis provides an extensive comparative performance analysis between a traditional scalar receiver and a vector receiver, in several reception scenarios, with different signal modulations and with different positioning algorithms.

1.2 Objectives

The present thesis' main objective is the performance comparison between traditional (scalar) and vector receiver architectures for satellite navigation, in several simulation scenarios and using different navigation signals, where the code tracking operation was implemented and the frequency tracking operation was considered perfect. For a vector receiver, this type of architecture is called the Vector Delay Lock Loop (VDLL), where we only consider the code tracking operation as a vector structure. In the final part of the document, the condition of perfect frequency tracking was relaxed, by implementing an architecture - the Vector Delay/Frequency Lock Loop (VDFLL) - where both the code and carrier were tracked. The initial proposed objectives of this thesis are now enumerated:

1. Generate both GPS and Galileo navigation signals and a satellite constellation;
2. Implement a scalar and vector receiver architecture for code tracking, with two different estimation algorithms;
3. Devise several simulation scenarios to test the receiver architectures;
4. Compare the obtained results between both the scalar and vector architectures and draw conclusions on that comparison.

All these tasks were carried out using the MATLAB software for implementation, testing and validation. All graphs and plots of the obtained results, presented in this thesis, were also produced using the same software.

Other objectives were drawn during the course of the present thesis, either resulting from issues encountered during development or surplus of available time for research on this thesis' topic. These objectives were the development of a new code discriminator for Galileo signals and the implementation of the aforementioned VDFLL architecture. Both objectives were accomplished.

1.3 Thesis Outline

This final section presents the contents of the chapters that compose this thesis:

- Chapter 2 provides a theoretical background on the proposed research topic, addressing the principle of satellite navigation, signal code tracking, multipath and an analysis on vector tracking receiver architectures, including a review of relevant literature.

- Chapter 3, contains an analysis on the receiver estimation algorithms utilized with both a scalar and a vector receiver. The integration of these algorithms in both receivers is addressed in detail, specifying the needed adaptations for each receiver architecture.

- The obtained results for the scalar receiver, with two estimation algorithms, are addressed in chapter 4. In this chapter, the several simulation scenarios devised are also described, as well as the performance metrics to be used. Results for several signal modulations are also presented.

- Similarly to the previous chapter, in chapter 5, the results obtained for the vector receiver (VDLL) are presented. These results are then compared to the ones obtained in the previous chapter, for each simulation scenario and estimation algorithm. A final overall comparison of the obtained results is displayed at the end, in two tables, for easy visualization.

- In chapter 6, a description and characterization of the VDFLL is provided, alongside some performance results for this algorithm.

- Finally, in chapter 7, the conclusions drawn from this thesis are reported, alongside some suggestions for future work on the topic.

- There are also four appendices in the present thesis, where other secondary, but nonetheless, important topics are addressed. Appendix A presents a characterization of the several GNSS signals utilized throughout this thesis; appendix B addresses the method with which we can generate vectors of correlated noise; appendix C describes a new code discriminator, based on a bank of correlators; and appendix D provides a mathematical characterization of the oscillators found usually in GNSS receivers' clocks.

Chapter 2

Theoretical Background

2.1 The principle of satellite navigation

Global navigation satellite systems (GNSS) operate under the premise that a given receiver, located on or above the Earth's surface, is fed by a set of broadcast signals from a satellite constellation being able to extract its position from those signals. In order to perform that task, the receiver marks the time at which the signals are received and identifies the time at which they were broadcast, through information encoded in them [2]. Knowing the time difference between broadcast and arrival, it is possible to calculate the distance between satellite and receiver and extract the x, y, z coordinates of the latter, in a three-dimensional reference frame. As such, three signals (one for each coordinate) are needed to obtain the values of the three coordinates, assuming that receiver clock is in synchronism with the clocks in all satellites. This is, however, not true as the receiver clock is usually of poorer quality than the satellite clocks, causing a Δt bias in the time measurements at the receiver. This translates into a fourth coordinate $u = c\Delta t$ (c is the speed of light in vacuum) that represents a distance error. That distance must be obtained as well, now being needed four satellites to do so. This u variable is common to all measured distances as all signals' arrival time is marked with the same receiver clock. This distance between a satellite and the receiver, affected by the u coordinate, is called the pseudorange.

Considering the earth-centered earth-fixed (ECEF) coordinate system, we can define the pseudorange ρ_i between a given satellite i with coordinates (X_i, Y_i, Z_i) and a receiver with coordinates (x, y, z) near the Earth's surface as:

$$\rho_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} + u \tag{2.1}$$

where $u < 0$ corresponds to having a retarded receiver clock, regarding the satellite clock, and $u > 0$ corresponds to the opposite case, where the receiver clock is advanced relative to the satellite clock. Another source of error that affects the pseudorange is due to the ionospheric and tropospheric propagation delays. In order to achieve an accurate positioning solution, these errors must be eliminated or minimized.

A traditional (scalar architecture) receiver tracks each satellite's received signal independently from one another, allocating one channel for each of them without any sharing of information [2]. The tracking portion of a receiver is composed of a code tracking block (addressed in section 2.2), a frequency tracking block and a navigation algorithm (addressed in chapter 3).

2.2 Code tracking

Figure 2.1 describes the tracking part of a conventional GNSS receiver. It is constituted by a carrier frequency wipe-off block, three complex correlators (E = early, P = prompt and L = late), being complemented with a code and carrier discriminator. The synchronization errors generated at the discriminators feed low-pass filters (LPF) which provide feedback signals to the carrier and spreading signal numerically-controlled oscillators (NCO).

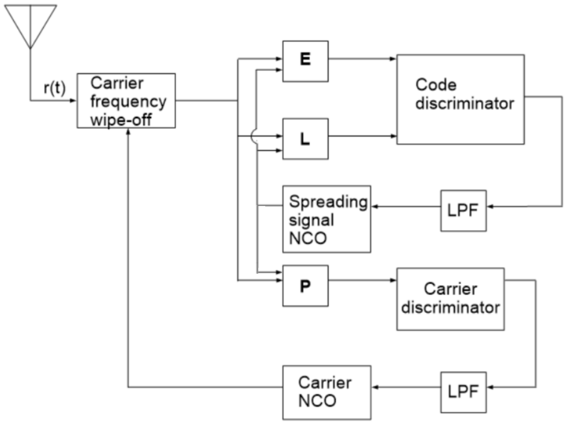


Figure 2.1: Block diagram of one channel of the conventional GNSS receiver.

The early and late correlators, alongside the code discriminator, are responsible for tracking the incoming signal's code delay - code tracking - forming a feedback structure called the Delay Lock Loop (DLL). On the other hand, the prompt correlator and the carrier discriminator form a feedback structure, through the carrier frequency wipe-off block, responsible for tracking the Doppler frequency shift of the incoming signal - carrier tracking . This structure is called a Frequency Lock Loop (FLL) if the carrier discriminator output depends on the frequency error or a phase lock loop (PLL) if it depends on the phase error. In this section, we will analyze the code tracking process assuming that the frequency tracking is carried out in a perfect manner (i.e. the receiver always estimates the incoming signals' Doppler frequency without error). The results presented in chapters 4 and 5 were obtained under the same assumption.

Let us consider the conventional GNSS receiver is fed by an incoming signal, generated from N satellites in view. The received signal is of the following type:

$$r(t) = \sum_{i=1}^{N} A_i X_i(t) \cos(\omega_0 t + \omega_{d_i} t + \theta_i) + w(t) \tag{2.2}$$

with A_i and $X_i(t)$ being, respectively, the amplitude of the signal from satellite i and encoded data signal; ω_0 and ω_{d_n} being, respectively, the carrier wave frequency and the Doppler frequency shift, in rad/s, and $w(t)$ being Additive White Gaussian Noise (AWGN), with power spectral density of $G_w(f) = N_0/2$. We can split equation (2.2) into its inphase and quadrature components, after low-pass filtering:

$$\begin{bmatrix} r_I(t) \\ r_Q(t) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} A_i X_i(t) \cos(\omega_{d_i} t + \theta_i) \\ \sum_{i=1}^{N} A_i X_i(t) \sin(\omega_{d_i} t + \theta_i) \end{bmatrix} + \begin{bmatrix} n_I(t) \\ n_Q(t) \end{bmatrix} \tag{2.3}$$

where $X_i(t)$ encompasses the spreading sequence of satellite i and the digital modulation ($X_i(t)$ will be denoted henceforth as spreading signal); $n_I(t)$ and $n_Q(t)$ are low-pass Gaussian white noises, with power spectral densities $G_{n_I} = G_{n_Q} = N_0 \Pi(f/2B)$, with B representing the bandwidth of the spreading signal $X_i(t)$ (assumed to the same as the receiver's) and $\Pi(f/2B)$ denoting a rectangle of duration $2B$ centered at the origin.

Let us assume that equation (2.3) is subject to a perfect Doppler wipe-off scheme for each signal i. According to figure 2.2, for a given signal n (with $n \in [1, \ldots, N]$), we obtain:

$$\begin{aligned} \begin{bmatrix} \tilde{r}_{I_n}(t) \\ \tilde{r}_{Q_n}(t) \end{bmatrix} &= \begin{bmatrix} \cos(\omega_{d_n} t) & \sin(\omega_{d_n} t) \\ -\sin(\omega_{d_n} t) & \cos(\omega_{d_n} t) \end{bmatrix} \begin{bmatrix} r_I(t) \\ r_Q(t) \end{bmatrix} \\ &= A_n X_n(t) \begin{bmatrix} \cos(\theta_n) \\ \sin(\theta_n) \end{bmatrix} + \begin{bmatrix} \sum_{i=1,\, i\neq n}^{N} A_i X_i(t) \cos\left[(\omega_{d_i} - \omega_{d_n})t + \theta_i\right] \\ \sum_{i=1,\, i\neq n}^{N} A_i X_i(t) \sin\left[(\omega_{d_i} - \omega_{d_n})t + \theta_i\right] \end{bmatrix} + \begin{bmatrix} \tilde{n}_{I_n}(t) \\ \tilde{n}_{Q_n}(t) \end{bmatrix} \end{aligned} \tag{2.4}$$

In equation (2.4), the middle term represents the inter-satellite interference, which, for simplicity's sake will, from now on, be disregarded in further mathematical relations. The first and third term, of the same equation, are, respectively, the useful part of the signal and the one resulting from channel noise. This last part can be expressed as:

$$\begin{bmatrix} \tilde{n}_{I_n}(t) \\ \tilde{n}_{Q_n}(t) \end{bmatrix} = \begin{bmatrix} \cos(\omega_{d_n} t) & \sin(\omega_{d_n} t) \\ -\sin(\omega_{d_n} t) & \cos(\omega_{d_n} t) \end{bmatrix} \begin{bmatrix} n_I(t) \\ n_Q(t) \end{bmatrix} \tag{2.5}$$

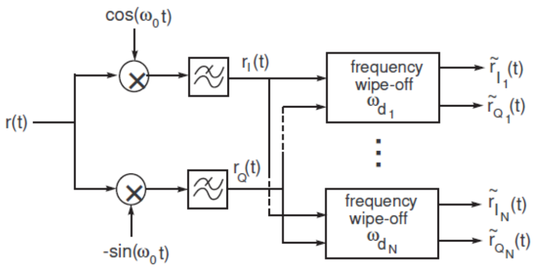


Figure 2.2: Carrier frequency wipe-off scheme [8].

Let us now characterize the noise vector (2.5). The autocorrelation function of $\tilde{n}_{I_n}(t)$ is:

$$R_{\tilde{n}_I}(\tau) = E\{\tilde{n}_{I_n}(t)\tilde{n}_{I_n}(t-\tau)\} = R_{n_I}(\tau)\cos(\omega_{d_n}\tau) \tag{2.6}$$

where

$$R_{n_I}(\tau) = 2N_0B\,\text{sinc}(2B\tau) \approx N_0\delta(\tau) \tag{2.7}$$

since $B >> 1$. The term $\delta(\tau)$ represents the Dirac delta function at instant τ.

We have $R_{n_I}(\tau) = R_{n_Q}(\tau)$ thus, $R_{\tilde{n}_I}(\tau) = R_{\tilde{n}_Q}(\tau)$. The cross correlation of $\tilde{n}_{I_n}(t)$ and $\tilde{n}_{Q_n}(t)$ is defined as follows:

$$R_{\tilde{n}_I\tilde{n}_Q}(\tau) = E\{\tilde{n}_{I_n}(t)\,\tilde{n}_{Q_n}(t-\tau)\} = R_{n_I}(\tau)\sin(\omega_{d_n}\tau) \approx 0 \tag{2.8}$$

Thus, the vectors $[n_I(t)\,n_Q(t)]^T$ and $[\tilde{n}_{I_n}(t)\,\tilde{n}_{Q_n}(t)]^T$ are statistically identical.

Let us now correlate the $[\tilde{r}_{I_n}(t)\,\tilde{r}_{Q_n}(t)]^T$ vector with two locally (in the receiver) generated copies of the spreading signal, one advanced, $X_n(t+\Delta/2-\epsilon_n)$, and one delayed, $X_n(t-\Delta/2-\epsilon_n)$, where Δ represents the DLL early-late spacing and ϵ_n represents the delay error between the locally-generated spreading code signal replica and the spreading code signal of the incoming signal. We now have four different signals: an early inphase and an early quadrature signal, as well as a late inphase and a late quadrature signal. This process is depicted in figure 2.3.

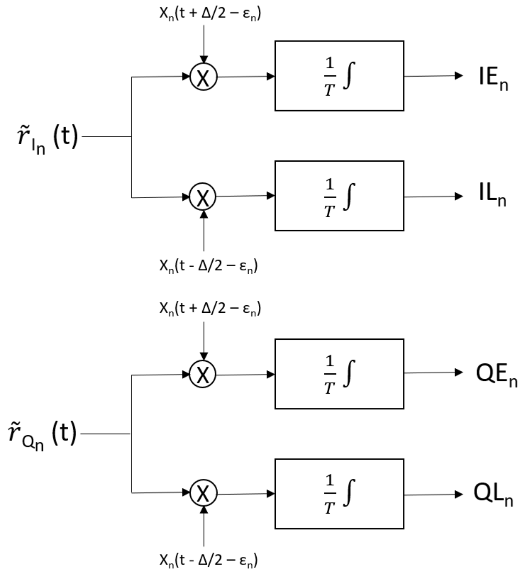


Figure 2.3: Generation of the early-late inphase and quadrature components [8].

Let us analyze, for example, the early inphase correlator output:

$$\begin{aligned} IE_n &= \frac{1}{T}\int_0^T \tilde{r}_{I_n}(t)\,X_n\left(t+\frac{\Delta}{2}-\epsilon_n\right)\,dt \\ &= A_nR_X\left(\epsilon_n-\frac{\Delta}{2}\right)\cos(\theta_n)+N_{IE_n} \end{aligned} \tag{2.9}$$

where T is the DLL correlation interval and:

$$R_X(\tau) = \frac{1}{T_p}\int_0^{T_p} X(t)X(t-\tau)\,d\tau \tag{2.10}$$

being the autocorrelation function of the spreading code signal $X_n(t)$ and T_p being its period. The autocorrelation expressions of the spreading signals, for several types of GNSS are indicated in appendix A.

Let us address the noise component of (2.9). It can be described as:

$$N_{IE_n} = \frac{1}{T}\int_0^T \tilde{n}_{I_n}(t)\, X_n\left(t + \frac{\Delta}{2} - \epsilon_n\right)\, dt \tag{2.11}$$

with zero mean and variance defined as:

$$\begin{aligned} E\{N_{IE_n}^2\} &= \frac{1}{T^2}\int_0^T\int_0^T E\{\tilde{n}_{I_n}(t)\, \tilde{n}_{I_n}(\lambda)\}\, X_n\left(t + \frac{\Delta}{2} - \epsilon_n\right)\, X_n\left(\lambda + \frac{\Delta}{2} - \epsilon_n\right)\}\, dt\, d\lambda \\ &= \frac{N_0}{T^2}\int_0^T X_n^2\left(\lambda + \frac{\Delta}{2} - \epsilon_n\right)\, d\lambda = \frac{N_0}{T} \end{aligned} \tag{2.12}$$

The other correlator outputs are:

$$\begin{aligned} IL_n &= A_n R_X\left(\epsilon_n + \frac{\Delta}{2}\right)\cos(\theta_n) + N_{IL_n} \\ QE_n &= A_n R_X\left(\epsilon_n - \frac{\Delta}{2}\right)\sin(\theta_n) + N_{QE_n} \\ QL_n &= A_n R_X\left(\epsilon_n + \frac{\Delta}{2}\right)\sin(\theta_n) + N_{QL_n} \end{aligned} \tag{2.13}$$

The noise components are all zero-mean, Gaussian variables with variance given by (2.12). The inphase components $[N_{IE_n}\, N_{IL_n}]^T$ and quadrature components $[N_{QE_n}\, N_{QL_n}]^T$ are independent from one another; however, the elements of each of those vectors are correlated:

$$\begin{aligned} E\{N_{IE_n}\, N_{IL_n}\} &= \frac{1}{T^2}\int_0^T\int_0^T E\{\tilde{n}_{I_n}(t)\, \tilde{n}_{I_n}(\lambda)\, X_n\left(t + \frac{\Delta}{2} - \epsilon_n\right)\, X_n\left(\lambda - \frac{\Delta}{2} - \epsilon_n\right)\}\, dt\, d\lambda \\ &= \frac{N_0}{T^2}\int_0^T X_n\left(\lambda + \frac{\Delta}{2} - \epsilon_n\right)\, X_n\left(\lambda - \frac{\Delta}{2} - \epsilon_n\right) = \frac{N_0}{T} R_X(\Delta) \end{aligned} \tag{2.14}$$

and $E\{N_{IE_n}\, N_{IL_n}\} = E\{N_{QE_n}\, N_{QL_n}\}$.

Let us now define function $D_c(\epsilon_n)$ as the response of a normalized non-coherent early-late power discriminator (NELP), with the following expression:

$$D_c(\epsilon_n) = \frac{(IE_n^2 + QE_n^2) - (IL_n^2 + QL_n^2)}{IE_n^2 + QE_n^2 + IL_n^2 + QL_n^2} \tag{2.15}$$

The fact that the discriminator is normalized means that the output of $D_c(\epsilon_n)$ will not depend on the amplitude A_n of the incoming signals. Also, by using a non-coherent discriminator, the output will be also independent of the carrier phase θ. If we disregard the effects of noise, we can analyze the NELP's ideal response. Without noise, equation (2.15) becomes:

$$D_c(\epsilon_n) = \frac{R_X^2\left(\epsilon_n - \frac{\Delta}{2}\right) - R_X^2\left(\epsilon_n + \frac{\Delta}{2}\right)}{R_X^2\left(\epsilon_n - \frac{\Delta}{2}\right) + R_X^2\left(\epsilon_n + \frac{\Delta}{2}\right)} \tag{2.16}$$

The NELP's response (S-curve), for a BPSK signal, with various values of early-late spacing is sketched in figure 2.4.

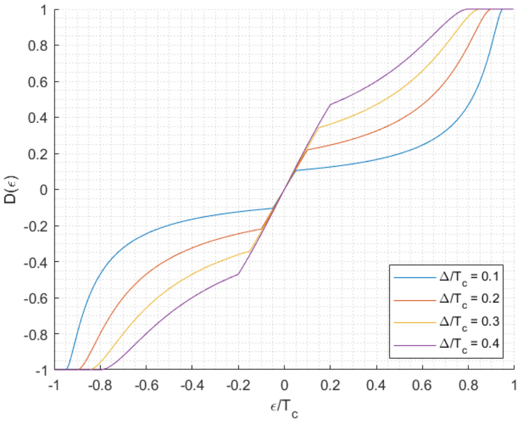


Figure 2.4: NELP discriminator response for a BPSK signal.

The DLL's purpose is to provide an estimate $\hat{\tau}_{\text{code}}$ of the delay τ_{code} of the spreading signal $X(t)$. As such, the DLL's estimate of the code delay is updated, at instant $(k+1)T$, for satellite n, through:

$$\hat{\tau}_{\text{code}_n}(k+1) = \hat{\tau}_{\text{code}_n}(k) - \gamma_c D_c(\epsilon_n) \tag{2.17}$$

where γ_c represents the code discriminator feedback gain and $\epsilon_n = \tau_{\text{code}_n} - \hat{\tau}_{\text{code}_n}$.

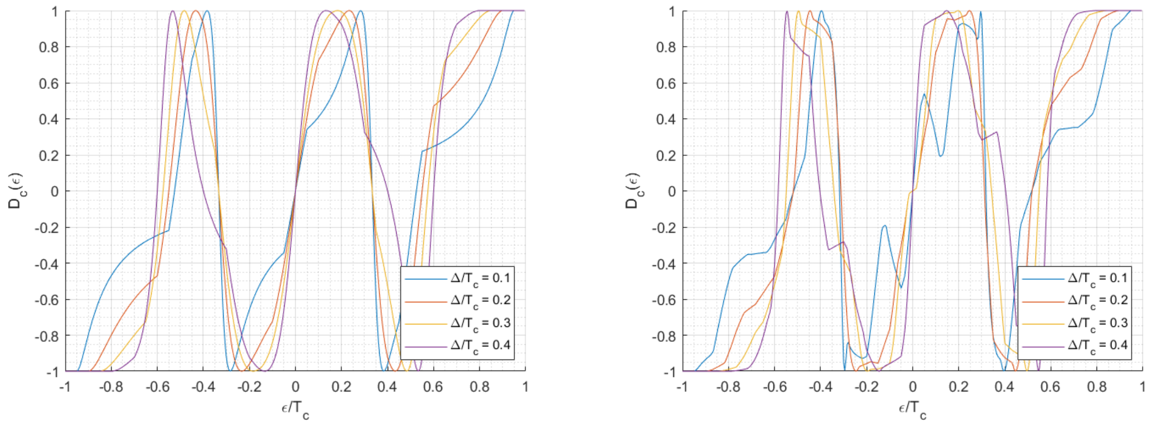
Code tracking is an essential task of a GNSS receiver as the incoming spreading code signal delay is related to the pseudorange between them through the following relation:

$$\rho_i = c \cdot \tau_{\text{code}_i} \tag{2.18}$$

where ρ_i represents the pseudorange between a satellite i and the receiver - defined in equation (2.1) - and c represents the speed of light in vacuum.

Utilizing a NELP discriminator with the BOCs(1,1) and CBOC(6,1,1/11) signals yields poor results, when compared to the ones obtained using a BPSK signal. Let us first analyze the S-curves (discriminator response curves) of a NELP discriminator for both the BOCs(1,1) and CBOC(6,1,1/11) signals, with several normalized early-late spacings (Δ/T_c), where T_c represents the spreading code's chip time, in figure 2.5.

The ideal S-curve for a given signal is always positive for positive values of the error ϵ/T_c and always negative for negative values of the error. Otherwise, $D_c(\epsilon_n)$ would have the wrong signal causing the DLL's code delay estimates to diverge from the correct value stabilizing near a new point where the S-curve crosses the 0 value for the $D_c(\epsilon)$ axis with positive slope, that does not correspond to having $\epsilon/T_c = 0$. This phenomenon is called false-code lock [2], [9]. In figure 2.5 we can observe that the desired behavior does not occur, with several inflections below 0 in the positive portion of the ϵ/T_c axis and the opposite in the negative portion. As such, the NELP discriminator does not behave in an acceptable way for both the BOCs(1,1) and CBOC(6,1,1/11) signals and thus, an alternative must be devised. This alternative code discriminator is called the bank of correlators and is described in full detail in appendix C.



(a) NELP S-curve for a BOCs(1,1) signal

(b) NELP S-curve for a CBOC(6,1,1/11) pilot signal

Figure 2.5: NELP response for various early-late spacings and different modulations.

2.3 Multipath

A multipath scenario occurs when, besides the signal in line-of-sight (LOS), replicas of that same signal reach a GNSS receiver, with a delay in time τ. Thus, considering that only one of these replicas reaches the receiver and $N = 1$ from equation (2.2), the received signal can be expressed as [10]:

$$r(t) = AX(t)\cos(\omega_0 t + \theta) + \alpha AX(t - \tau)\cos(\omega_0 t + \theta + \phi) + w(t) \tag{2.19}$$

where $X(t)$ represents the encoded data signal, τ represents the additional delay of the reflected ray, ϕ represents a phase offset due to the extra delay and α represents the attenuation of the multipath signal.

If a NELP discriminator is in use in the receiver's DLL, then from (2.19) we will obtain two inphase and two quadrature components, one advanced: the early one; and one delayed in time: the late one, analogously to (2.9) and (2.13).

$$\begin{aligned}
I_E =& AR_X\left(\epsilon - \frac{\Delta}{2}\right)\cos(\theta) + \alpha AR_X\left(\epsilon - \frac{\Delta}{2} - \tau\right)\cos(\theta + \phi) + N_{i,E} \\
I_L =& AR_X\left(\epsilon + \frac{\Delta}{2}\right)\cos(\theta) + \alpha AR_X\left(\epsilon + \frac{\Delta}{2} - \tau\right)\cos(\theta + \phi) + N_{i,L} \\
Q_E =& AR_X\left(\epsilon - \frac{\Delta}{2}\right)\sin(\theta) + \alpha AR_X\left(\epsilon - \frac{\Delta}{2} - \tau\right)\sin(\theta + \phi) + N_{q,E} \\
Q_L =& AR_X\left(\epsilon + \frac{\Delta}{2}\right)\sin(\theta) + \alpha AR_X\left(\epsilon + \frac{\Delta}{2} - \tau\right)\sin(\theta + \phi) + N_{q,L}
\end{aligned} \tag{2.20}$$

For the sake of simplicity, we will consider a non-normalized early-late power discriminator is in use from here onward, as the results obtained would be the same with a normalized version. For the same reason, the presence of noise will be disregarded.

The response of the discriminator is now described by:

$$\begin{aligned}
D_c(\epsilon,\tau,\phi) &= (I_E^2+Q_E^2)-(I_L^2+Q_L^2)\\
&= A^2R_X^2\left(\epsilon-\frac{\Delta}{2}\right)+\alpha^2A^2R_X^2\left(\epsilon-\frac{\Delta}{2}-\tau\right)+2\alpha A^2R_X\left(\epsilon-\frac{\Delta}{2}\right)R_X\left(\epsilon-\frac{\Delta}{2}-\tau\right)\cos(\phi)\\
&\quad -A^2R_X^2\left(\frac{\epsilon+\Delta}{2}\right)-\alpha^2A^2R_X^2\left(\epsilon+\frac{\Delta}{2}-\tau\right)-2\alpha A^2R_X\left(\frac{\epsilon+\Delta}{2}\right)R_X\left(\epsilon+\frac{\Delta}{2}-\tau\right)\cos(\phi)
\end{aligned} \tag{2.21}$$

We can now define two curves called *multipath envelopes*, which represent the worst case scenarios for measured pseudorange errors, in a DLL, with the presence of noise being disregarded. To obtain these curves, the following equations must be solved in order to ϵ:

$$\begin{aligned}
(i) &\qquad D_c(\epsilon,\tau,0)=0\\
(ii) &\qquad D_c(\epsilon,\tau,\pi)=0
\end{aligned} \tag{2.22}$$

These equations depict the situation where the DLL feedback loop is not being updated (equilibrium condition), according to (2.17), and the multipath reflected ray is either in phase ($\phi=0$) or in opposition of phase ($\phi=\pi$) with the direct ray. From equation (i) we obtain:

$$R_X\left(\epsilon-\frac{\Delta}{2}\right)+\alpha R_X\left(\epsilon-\tau-\frac{\Delta}{2}\right)=\pm\left[R_X\left(\epsilon+\frac{\Delta}{2}\right)+\alpha R_X\left(\epsilon-\tau+\frac{\Delta}{2}\right)\right] \tag{2.23}$$

and from equation (ii) we obtain:

$$R_X\left(\epsilon-\frac{\Delta}{2}\right)-\alpha R_X\left(\epsilon-\tau-\frac{\Delta}{2}\right)=\pm\left[R_X\left(\epsilon+\frac{\Delta}{2}\right)-\alpha R_X\left(\epsilon-\tau+\frac{\Delta}{2}\right)\right] \tag{2.24}$$

It is important to notice that in both (2.23) and (2.24), only the solution with the $+$ sign, in the right member, is valid. This is illustrated by the fact that if we consider $\tau=0$, we would obtain:

$$R_X\left(\epsilon-\frac{\Delta}{2}\right)=\pm R_X\left(\epsilon+\frac{\Delta}{2}\right) \tag{2.25}$$

where, due to the fact that GNSS signals' autocorrelation functions are symmetrical, $\epsilon=0$ would be the only solution to the equation above, thus rendering the $+$ sign the only valid solution.

Now, we can merge equations (2.23) and (2.24) to obtain:

$$R_X\left(\epsilon-\frac{\Delta}{2}\right)-R_X\left(\epsilon+\frac{\Delta}{2}\right)=\pm\alpha\left[R_X\left(\epsilon-\tau-\frac{\Delta}{2}\right)-R_X\left(\epsilon-\tau+\frac{\Delta}{2}\right)\right] \tag{2.26}$$

Equation (2.26) represents the solution for obtaining the multipath envelopes, which can be seen in figure 2.6, for the GPS C/A signal and the Galileo BOCs(1,1) and CBOC(6,1,1/11) pilot signals, using a NELP discriminator, with $\Delta/T_c=0.1$ and $\alpha=0.5$. We also express the delay τ in meters (multiplied by c).

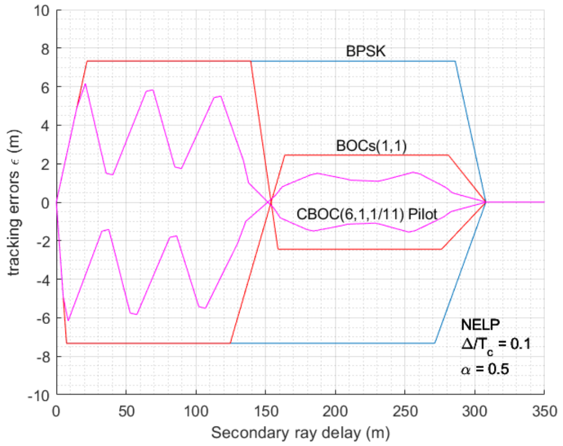


Figure 2.6: Multipath envelopes for several modulations.

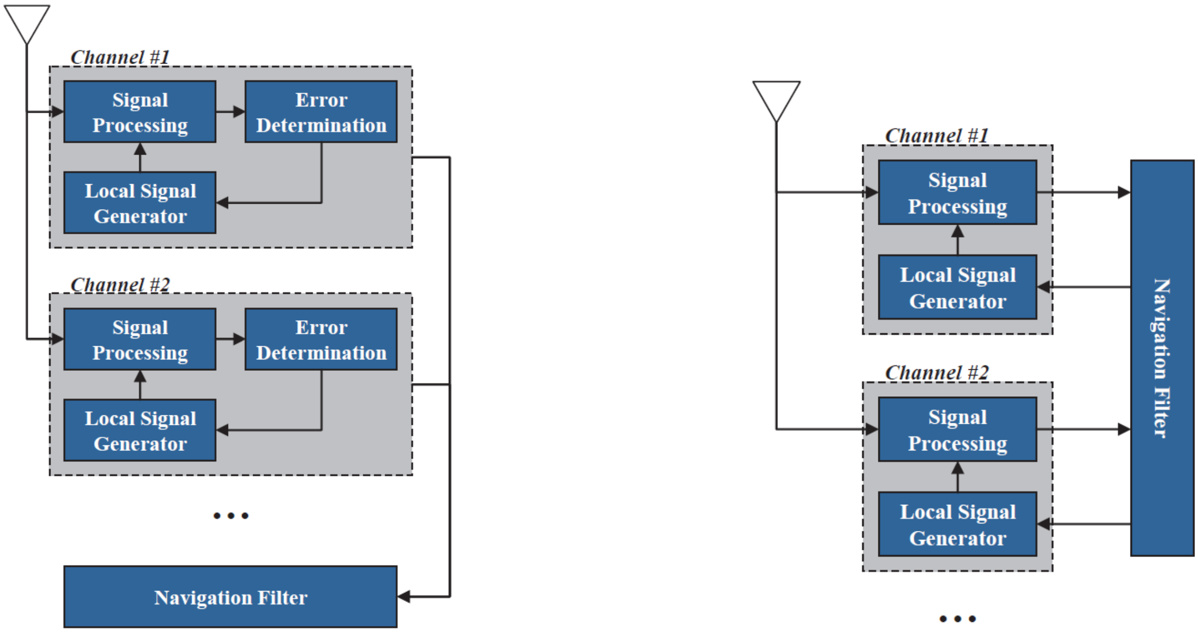
Notice that for small secondary ray delays $(< 20\,\mathrm{m})$ the various solutions in figure 2.6 are equivalent. However, for longer delays, the BOCs(1,1) and CBOC(6,1,1/11) pilot signals exhibit better performance. This is especially true for delays larger than approximately 150 meters.

2.4 Vector tracking architectures

The traditional GNSS receiver architecture, a scalar tracking architecture, tracks the received GNSS signals (with four being the needed minimum) independently from one another, using a set of tracking loops, for code and Doppler phase or frequency. These tracking loops then feed an estimation algorithm, in order to obtain an estimate of the receiver's position, velocity, and time (PVT). Vector tracking architectures are an advanced and more complex form of processing the several GNSS signals that reach a given receiver. They are capable of operating at lower carrier-to-noise ratios (C/N$_0$) and tolerating higher receiver dynamics. The vector architecture, contrarily to a traditional scalar architecture, processes all signals jointly through one big loop of code and frequency discriminators, coupled through an estimation algorithm (Least-Squares, extended Kalman filter, etc.) [1], [3], [11]. Figure 2.7 illustrates the differences between the scalar and vector architectures.

Several advantages exist from tracking the satellite signals jointly. For example, the tracking of the receiver's dynamics is assured by the whole set of received signals, whereas in a scalar receiver, it must be done individually by each channel, which usually leads to a poorer overall performance. Also, in a vector architecture, in case of signal blockage (from on or more satellites) or in a case of poor C/N$_0$ conditions, the cumulative power of the satellite signals allows for the receiver to continue to operate with acceptable performance [1], [12]. In other words, the several satellite signals, in a vector receiver, aid one another when subject to abnormal conditions.

Vector receivers can present three types of architecture [1], [3]: the Vector Delay Lock Loop (VDLL), where the code tracking portion of the receiver forms a vector structure coupled with the estimation algorithm, while the carrier tracking portion is separated as in the scalar architecture; the Vector Frequency Lock Loop (VFLL), with the carrier tracking portion being coupled with the estimation algorithm and the



(a) Basic scalar tracking architecture

(b) Basic vector tracking architecture

Figure 2.7: Comparison of the scalar and vector tracking architectures [5].

code tracking part being separated; and, finally, the Vector Delay/Frequency Lock Loop (VDFLL) where both the code and carrier tracking portions of the receiver are coupled, through an estimation algorithm, in a single feedback loop.

The first concepts of vector tracking architectures date back to the decade of 1980 (Copps et al. [13]), where an architecture combining GPS tracking loops and an *inertial measurements unit*, coupled through an estimation algorithm, was proposed. Later, in the decade of 1990, Sennot and Senffner [14], [15] and [16] presented some architectures for coupled GPS processors. In the year of 1996, the concept of the VDLL was coined by Parkinson and Spilker [12], becoming one of the most cited references for vector tracking architectures [1], although with only faint details on its implementation being presented. Two years later, in 1998, Zhodzishsky et al. [17] introduced the concept of *Co-Op* tracking which borrows heavily from [12] but also allows for carrier phase vector tracking, whereas the original VDLL described by Spilker could only achieve code vector tracking.

With the turn of the millennium, several papers have continued to be published on the subject, delving deeper into the implementation, testing and validation of vector architectures for GNSS receivers. At the university of Calgary, Canada, Petovello and Lachapelle [5] implemented and tested three different Kalman filter implementations for estimating signal tracking errors, with emphasis on carrier phase. The authors managed to obtain a 15 dB improvement in performance, over existing scalar architectures. In 2006, Pany and Eissfeller [6] attested to the VDLL's robustness to severe variations of C/N$_0$ as well as managing to track signals with power below 10dB-Hz using a vector architecture. Research on the benefits of the VDLL subject to interference, in comparison to traditional receiver architectures, was conducted by Benson [7]. More recently, in 2010, performance analysis of several Kalman filter approaches for vector tracking loops, have been undertaken by Won et al. [18]. Implementation of vector receivers with proprietary software has also been studied by Won et al. [19]. The same year, methodologies for evaluating the comparative performance of vector receiver architectures, against scalar ones, have been presented by Bhattacharyya and Gebre-Egziabher [20], as well as, criteria for Kalman filter tuning, in vector tracking loops by Zhao and D.Akos [21].

Research on vector tracking algorithms as also recently birthed some PhD theses on the subject, such as Lashley [1] in 2009, Bhattacharyya [3] in 2012 and Brewer [22] in 2014. A particularly complete list of published work on vector tracking algorithms can be found in [22].

In Portugal, in particular, research on vector tracking algorithms and its comparative performance in regard to scalar algorithms, in several scenarios, has been conducted mainly by Fernando D. Nunes and Fernando M. G. Sousa, at the *Instituto de Telecomunicações*, located in *Instituto Superior Técnico* (IST), in Lisbon. Some important papers published on the subject, by these researchers, are [23], [24] and [25], published between 2012 and 2016.

Chapter 3

Scalar and vector receiver algorithms analysis

In this chapter, we will analyze the estimation algorithms to be utilized by both the scalar and vector receivers, as well as their integration in the architecture of those receivers. Section 3.1 performs this analysis on the least-squares algorithm, while section 3.2 does the same for the extended Kalman filter (EKF) algorithm. We consider, in this chapter, that the estimation algorithms are only fed by information provided by the DLL's and we also consider that carrier tracking is carried out perfectly by the receiver. The algorithms and line of thought presented in this chapter are the ones adopted to produce the results presented in chapters 4 and 5.

3.1 Least-Squares algorithm

Let us start by recalling the definition of pseudorange, stated in chapter 2, section 2.1, and defined mathematically in equation (2.1). We rewrite that same equation here, in order to more easily follow the line of thought of this algorithm:

$$\rho_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} + u \tag{3.1}$$

where $u = c\Delta t$, c is the speed of light in vacuum and Δt is the delay of the receiver's clock relative to the GNSS clock.

Recall that a set of $N \geq 4$ satellites is required to acquire both the position and the error resulting from the clock difference u. It is very difficult to solve a system of four or more of equations of the type of (3.1) in an exact way. As such, utilizing the Taylor series expansion, while only keeping the linear part, proves to be a much better method, as it allows for the system to be solved iteratively. Using the linear Taylor series expansion on (3.1) we obtain the following increments equation:

$$\Delta\rho_i = \frac{\partial\rho_i}{\partial x}\Delta x + \frac{\partial\rho_i}{\partial y}\Delta y + \frac{\partial\rho_i}{\partial z}\Delta z + \frac{\partial\rho_i}{\partial u}\Delta u \tag{3.2}$$

This equation can also be expressed in matrix form, resulting in:

$$\underbrace{\begin{bmatrix} \Delta\rho_1 \\ \vdots \\ \Delta\rho_N \end{bmatrix}}_{\Delta\rho} = \underbrace{\begin{bmatrix} \frac{\partial\rho_1}{\partial x} & \frac{\partial\rho_1}{\partial y} & \frac{\partial\rho_1}{\partial z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial\rho_N}{\partial x} & \frac{\partial\rho_N}{\partial y} & \frac{\partial\rho_N}{\partial z} & 1 \end{bmatrix}}_{G} \cdot \underbrace{\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta u \end{bmatrix}}_{\Delta X} \tag{3.3}$$

with G being the geometry matrix. Since G can be a non-square matrix, a general least-squares solution for equation (3.3) can be written using the pseudo-inverse of G:

$$\Delta X = (G^T G)^{-1} G^T \Delta\rho \tag{3.4}$$

By using equation (3.4) one can achieve an estimated solution for the receiver's position $(\hat{x}, \hat{y}, \hat{z})$ and clock delay $(\hat{u})$, provided an initial estimate $(\hat{x}_0, \hat{y}_0, \hat{z}_0, \hat{u}_0)$ is given. Several iterations of the described process are needed in order to reduce the effects of the linearization undertaken in equation (3.2), utilizing the initial estimate to produce a new estimate, which is, in turn, used as the starting estimate for a new iteration step. This process is repeated as many times as needed until a desired convergence value is achieved. The stop criterion may be based on a pre-defined difference between the results of the current and the previous estimate.

The steps of the the least-squares algorithm are now described:

1. Assume the positions of N satellites $[X_i\, Y_i\, Z_i]^T$ are known and the pseudoranges ρ_i are measured.

2. Assume an initial estimate for the receiver's state (position and clock error) $[\hat{x}_0\, \hat{y}_0\, \hat{z}_0\, \hat{u}_0]^T$.

3. Determine the approximate geometry matrix

$$\begin{aligned} \hat{G} &= \begin{bmatrix} \frac{\partial\rho_1}{\partial x} & \frac{\partial\rho_1}{\partial y} & \frac{\partial\rho_1}{\partial z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial\rho_N}{\partial x} & \frac{\partial\rho_N}{\partial y} & \frac{\partial\rho_N}{\partial z} & 1 \end{bmatrix}_{x=\hat{x_0},y=\hat{y_0},z=\hat{z_0}} \\ &= \begin{bmatrix} -\frac{X_1-\hat{x_0}}{D_1} & -\frac{Y_1-\hat{y_0}}{D_1} & -\frac{Z_1-\hat{z_0}}{D_1} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{X_N-\hat{x_0}}{D_N} & -\frac{Y_N-\hat{y_0}}{D_N} & -\frac{Z_N-\hat{z_0}}{D_N} & 1 \end{bmatrix} \end{aligned} \tag{3.5}$$

where

$$D_i = \sqrt{(X_i - \hat{x}_0)^2 + (Y_i - \hat{y}_0)^2 + (Z_i - \hat{z}_0)^2} \tag{3.6}$$

are the estimated receiver-satellite distances (ranges).

4. Obtain the pseudorange error vector

$$\Delta\rho = \rho_{measured} - \rho_{estimated} \tag{3.7}$$

where $\rho_{measured}$ is obtained during step 1 and $\rho_{estimated}$ can be expressed as:

$$\rho_{estimated} = \sqrt{(X_i - \hat{x}_0)^2 + (X_i - \hat{y}_0)^2 + (Z_i - \hat{z}_0)^2} + \hat{u}_0 \tag{3.8}$$

5. Utilize equation (3.4) to obtain the position error vector ΔX and update the initial estimate of step 2

$$[\hat{x}_1\, \hat{y}_1\, \hat{z}_1\, \hat{u}_1]^T = [\hat{x}_0\, \hat{y}_0\, \hat{z}_0\, \hat{u}_0]^T + \underbrace{[\Delta x\, \Delta y\, \Delta x\, \Delta u]^T}_{\Delta X} \tag{3.9}$$

6. Repeat steps 2 through 5, with the estimate obtained in step 5 replacing the one assumed in step 2, for the new iteration cycle. The number of repetitions of those steps will depend on the speed of convergence of the algorithm, as well as the desired accuracy for the results it produces.

With the steps of a conventional least-squares algorithm explained, an overview on how to include them in a scalar and vector receiver architecture ensues.

3.1.1 Integration of the Least-Squares algorithm in a scalar receiver

In a scalar receiver, the least-squares algorithm described earlier is an ”off the shelf” solution as it can be used immediately, receiving the measured pseudoranges ρ_i from the several DLLs assigned to track each satellite in use. As such, the scalar receiver processes information in a waterfall type of processing, with it flowing just one way, as we can see in figure 3.1:

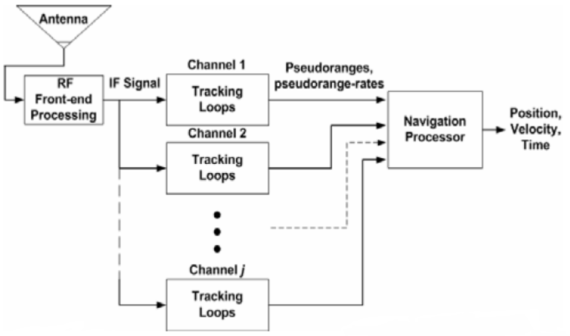


Figure 3.1: Architecture of a scalar receiver [26].

Notice that each satellite is tracked independently of the others, with each DLL feeding a measured pseudorange to the least-squares algorithm with no information being fed back.

3.1.2 Integration of the Least-Squares algorithm in a vector receiver

In a vector receiver, the least-squares algorithm works in a slightly different way than in the scalar case, presented in section (3.1), as it is the element that closes a feedback loop that characterizes this type of receiver.

The least-squares algorithm is fed with a pseudorange error vector $\Delta\rho$ that results from subtracting a vector of measured pseudoranges, obtained from a series of code discriminators, from a vector of predicted pseudoranges, that are initialized on the first iteration of the algorithm. This $\Delta\rho$ vector is related to the vector of position errors ΔX through the following relation:

$$\Delta\rho = G\,\Delta X \tag{3.10}$$

with G being the geometry matrix presented in (3.3).

The pseudorange errors can be expressed, in the linearity region, through the code discriminator outputs multiplied by a matrix of discriminator gains:

$$\Delta\rho = \Gamma \begin{bmatrix} D(\epsilon_1) \\ \vdots \\ D(\epsilon_N) \end{bmatrix} \tag{3.11}$$

with Γ being the diagonal matrix of discriminator gains:

$$\Gamma = \begin{bmatrix} \gamma_{c_1} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \gamma_{c_N} \end{bmatrix}_{(N\times N)} \tag{3.12}$$

We can now write the least-squares solution of (3.10) as:

$$\begin{aligned} \Delta X &= (G^TG)^{-1}G^T\Delta\rho \\ &= (G^TG)^{-1}G^T\Gamma \begin{bmatrix} D(\epsilon_1) \\ \vdots \\ D(\epsilon_N) \end{bmatrix} \end{aligned} \tag{3.13}$$

using ΔX, at instant k, to update the estimate for the position vector of the receiver, in instant $k+1$:

$$\hat{X}(k+1|k) = \hat{X}(k|k-1) - \Delta X(k) \tag{3.14}$$

Now, with $\hat{X}(k+1|k)$ obtained, it is possible to create a predicted pseudorange between satellite i and the receiver at instant $k+1$, assuming the position for that satellite, at that instant, is known $(X_i(k+1),\, Y_i(k+1),\, Z_i(k+1))$:

$$\begin{aligned} \hat{\rho}_i(k+1|k) \;&= \sqrt{(X_i(k+1)-\hat{x}(k+1|k))^2 + (Y_i(k+1)-\hat{y}(k+1|k))^2 + (Z_i(k+1)-\hat{z}(k+1|k))^2} \\ &\quad +\; \hat{u}(k+1|k) \end{aligned} \tag{3.15}$$

These predicted pseudoranges $\hat{\rho}_i(k+1|k)_{(i=1,\dots,N)}$ will be fed back to the code discriminators, in order to be compared with the pseudoranges measured by them. That comparison will result in a new $\Delta\rho$, beginning the explained loop anew. A representation of the vector receiver with the least-squares algorithm is now presented in figure 3.2.

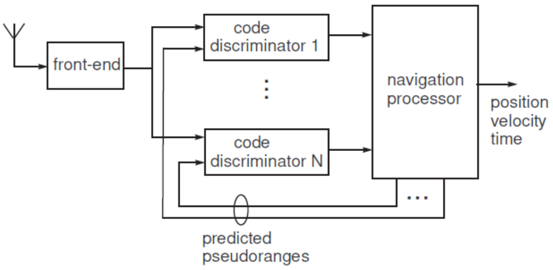


Figure 3.2: Architecture of a vector receiver with ideal carrier tracking [26].

Notice that now, unlike the scalar receiver, it is not possible to consider the tracking loops and the navigation processor as separate blocks. In a vector receiver they form a single block which we call the VDLL, since we consider the code tracking block coupled with the navigation processor.

In order to initialize the algorithm, a first estimate of the N pseudoranges $\hat{\rho}$ is needed, as well as a first estimate of the receivers position vector $\hat{X}$.

3.2 Extended Kalman filter algorithm

The Kalman filter, first proposed by Rudolph E. Kalman in 1960 [27], is a recursive linear estimator, meaning it computes the current estimate of a state vector x considering the information of all past estimates, while only storing the previous estimate [28], [29].

The Kalman filter is characterized by two steps, at each iteration (instant k in time): a prediction step and a filtering step [28], [29]. In the prediction step, $\hat{x}_k$ is estimated based on the set of past observations $Z_i = \{z_0, \cdots, z_{k-1}\}$ and in the filtering step, we correct the predicted estimate $\hat{x}_k$ using the current observation z_k (the subscripts indicate time instants).

In order to apply the Kalman filter, we need to define a model of the random process that is to be estimated: the *dynamics' model*.

$$x_{k+1} = \Phi_k x_k + w_k \tag{3.16}$$

where $x_{k_{(N\times1)}}$ is the state vector at instant k; $\Phi_{k_{(N\times N)}}$ is the state transition matrix from instant k to $k+1$ and $w_{k_{(N\times1)}}$ is the dynamics noise vector at instant k.

We also need a model for the measurements of the random process, which occur at discrete time intervals: the *observations' model*.

$$z_k = H_k x_k + v_k \tag{3.17}$$

where $z_{k_{(M\times1)}}$ is the observations vector at instant k; $H_{k_{(M\times N)}}$ is the observations matrix and $v_{k_{(N\times1)}}$ is the observations noise vector.

If we assume that both w_k and v_k are white Gaussian sequences with a known covariance matrix and null cross correlation between one another, we can write the covariance matrix of both these sequences as:

$$E\{w_k w_i^T\} = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases}$$

$$E\{v_k v_i^T\} = \begin{cases} \widetilde{R}_k, & i = k \\ 0, & i \neq k \end{cases} \tag{3.18}$$

$$E\{w_k v_i^T\} = 0, \qquad \forall i, k$$

It can be shown that the filtering step equations are [28]:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K_k \left[z_k - H_k \hat{x}(k|k-1)\right]$$

$$P(k|k) = \left[I - K_k H_k\right] P(k|k-1) \left[I - K_k H_k\right]^T + K_k \widetilde{R}_k K_k^T \tag{3.19}$$

where $\hat{x}(k|k)$ represents the filtered estimate of x at instant k, $\hat{x}(k|k-1)$ represents the predicted estimate of x at instant k, and $P(k|k-1)$ and $P(k|k)$ represent respectively the prediction and filtering error covariance matrices. The term K_k is the Kalman gain and can be computed as:

$$K_k = P(k|k-1) H_k^T \left(H_k P(k|k-1) H_k^T + \widetilde{R}_k\right)^{-1} \tag{3.20}$$

The term $z_k - H_k \hat{x}(k|k-1)$ in (3.19) is called the innovations process [30] and it is a white Gaussian sequence, when the Kalman filter is operating correctly. The innovations process will be of extreme importance when we analyze the Kalman filter algorithm in a vector receiver.

According to [28], the prediction step equations can be expressed as:

$$\hat{x}(k+1|k) = \Phi_k \hat{x}(k|k)$$

$$P(k+1|k) = \Phi_k P(k|k) \Phi_k^T + Q_k \tag{3.21}$$

Matrices $P(k|k)$ and $P(k+1|k)$ provide information on the quality of the filtering and prediction step state vector estimates, respectively. Those matrices do not depend on the observations and, therefore, can be computed prior to the filter's operation. The Kalman filter's operation can be seen in the schematic of figure 3.3 .

The Kalman filter, as was just characterized, assumes that both the model for the dynamics and observations is linear, which in a real-world scenario is almost never true. Thus, a solution for this issue is to use the Taylor series expansion on both the dynamics and observations models, to linearize its non-linearities, and keep only the linear terms. However, this linearization must be computed at each

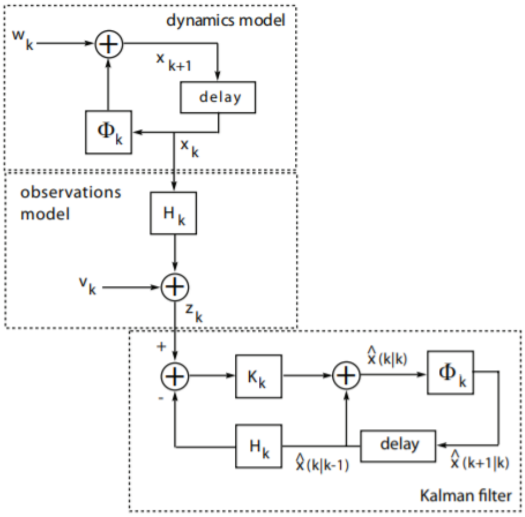


Figure 3.3: Kalman filter operation [31].

iteration step to avoid the filter's divergence [31]. This approach leads to the extended Kalman filter (EKF).

Let us assume that we have the following dynamics' and observations' models [31]:

$$\begin{aligned} x_{k+1} &= f(x_k) + w_k \\ z_k &= h(x_k) + v_k \end{aligned} \tag{3.22}$$

where w_k and v_k are defined by (3.18) and $f(x_k)$ and $h(x_k)$ are non-linear functions. We can expand these functions in Taylor's series, assuming they vary slowly in time, around the conditional means $\hat{x}(k|k)$ and $\hat{x}(k|k-1)$. Thus we have:

$$\begin{aligned} f(x_k) &= f(\hat{x}(k|k)) + F_k(x_k - \hat{x}(k|k)) + \dots \\ h(x_k) &= h(\hat{x}(k|k)) + H_k(x_k - \hat{x}(k|k-1)) + \dots \end{aligned} \tag{3.23}$$

where F_k and H_k are the Jacobian matrices of $f(x_k)$ and $h(x_k)$, respectively. The Jacobian matrices can be defined as:

$$\begin{aligned} F_k &= \left[\frac{\partial f_i(x)}{\partial x_j}\right]_{x=\hat{x}(k|k)} \qquad (N \times N) \\ H_k &= \left[\frac{\partial h_i(x)}{\partial x_j}\right]_{x=\hat{x}(k|k-1)} \qquad (M \times N) \end{aligned} \tag{3.24}$$

If we disregard the higher-order terms of (3.23), we can express (3.22) as :

$$\begin{aligned} x_{k+1} &= F_k\, x_k + w_k + u_k \\ z_k &= H_k\, x_k + v_k + y_k \end{aligned} \tag{3.25}$$

with $u_k = f(\hat{x}(k|k)) - F_k(\hat{x}(k|k))$ and $y_k = h(\hat{x}(k|k-1)) - H_k\hat{x}(k|k-1)$ being small quantities.

The equations in (3.25) define a linear Kalman filtering problem. Thus we can write, for the filtering step [30]:

$$\begin{aligned} \hat{x}(k|k) &= \hat{x}(k|k-1) + K_k\left(z_k - h\left[\hat{x}(k|k-1)\right]\right) \\ P(k|k) &= \left[I - K_kH_k\right]P(k|k-1)\left[I - K_kH_k\right]^T + K_k\widetilde{R}_kK_k^T \end{aligned} \tag{3.26}$$

with the Kalman gain being:

$$K_k = P(k|k-1)H_k^T\left(H_kP(k|k-1)H_k^T + \widetilde{R}_k\right)^{-1} \tag{3.27}$$

If the dynamics model in (3.22) is linear, the prediction step can be expressed as follows [30]:

$$\begin{aligned} \hat{x}(k+1|k) &= \Phi_k\hat{x}(k|k) \\ P(k+1|k) &= \Phi_kP(k|k)\Phi_k^T + Q_k \end{aligned} \tag{3.28}$$

It is to be noted that if (3.22) is linear, then it is transformed in equations (3.16) and (3.17) and the EKF is transformed into a linear Kalman filter. Also, the EKF does not need both the dynamics' and observations' models to be non-linear: if one is, then the EKF algorithm must be used instead of the linear Kalman filter. Both the linear Kalman filter and the EKF require initializations, in order to start the algorithms, which can be expressed as: $\hat{x}(0|-1) = \hat{x}_0$ and $P(0|-1) = \hat{P}_0$, assuming $\hat{x}_0$ and $\hat{P}_0$ are known.

The EKF algorithm behaves well in most non-linear estimation problems, if the initial error is not very large, and both the dynamics' and observations' noises are also small. This, allied to the fact that the algorithm is relatively simple from a computational point of view, means the EKF is a good first approach to a non-linear estimation problem [31]. However, the extended Kalman filter is not "bullet-proof" and can diverge from the true value of the state vector it is trying to estimate. This is especially true for situations where dynamics or observations have a very non-linear behavior, coupled with the way the EKF linearizes that behavior [32]. In order to better understand the limitations of the EKF, and gain some insight on a proposed solution, the unscented Kalman filter, it is recommended the reading of [32] for a detailed explanation. Also, in [33], a thorough description of the Kalman filter, its derivatives and more complex estimators can be found.

3.2.1 Dynamics and observations models for the EKF algorithm

In order to use the EKF algorithm, as detailed in 3.2, it is necessary to describe both a dynamics and an observations model, to be used in the algorithm. In the case of the present thesis, the dynamics model used is linear and the observations model is nonlinear.

Dynamics model

The dynamics model utilized was the PVT model (position, velocity and time), where we have a state vector x_k with eight components: three for position in the x_u, y_u, z_u axes plus three for the respective velocities $\dot{x}_u$, $\dot{y}_u$, $\dot{z}_u$) and two for the receiver's clock state [34]. We define the state vector, at instant k, as $[x_{1,k}, \cdots, x_{8,k}]^T$, where $x_{1,k} = x_{u,k}$, $x_{2,k} = \dot{x}_{u,k}$, $x_{3,k} = y_{u,k}$, $x_{4,k} = \dot{y}_{u,k}$, $x_{5,k} = z_{u,k}$, $x_{6,k} = \dot{z}_{u,k}$, $x_{7,k} = c\,x_{\phi,k} = u_u$ and $x_{8,k} = c\,x_{f,k}$. The $x_{7,k}$ and $x_{8,k}$ represent the receiver's clock states. In a PV model, we model the receiver position along each coordinate x, y, z, in the ECEF reference frame, as double integrated Brownian motion, as can be seen in figure 3.4.

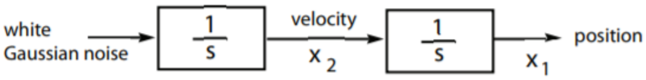


Figure 3.4: PV model using double integrated Brownian motion [31].

Let us define the continuous-time state model corresponding to the receiver coordinate x_u:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ u_v(t) \end{bmatrix} \tag{3.29}$$

with the covariance of the noise vector being $E\{[0\,u_v(t)]^T\,[0\,u_v(\tau)]\} = Q\delta(t-\tau)$ and, as a result, the covariance matrix being:

$$Q_k = q_v \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.30}$$

Since GNSS receivers operate in discrete-time, with sampling interval Δt, the model in (3.29) must be expressed in discrete-time as well. Thus, we have:

$$\begin{bmatrix} \dot{x}_{1,k+1} \\ \dot{x}_{2,k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{a} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} u_{1,k} \\ u_{2,k} \end{bmatrix} \tag{3.31}$$

with $\Delta t = T$ (correlation interval). The dynamics noise covariance matrix becomes:

$$Q'_k = E\{[u_{1,k}\,u_{2,k}]^T\,[u_{1,k}\,u_{2,k}]\} = q_v \Delta t \begin{bmatrix} \frac{(\Delta t)^2}{3} & \frac{\Delta t}{2} \\ \frac{\Delta t}{2} & 1 \end{bmatrix} \tag{3.32}$$

For the y and z coordinates, the discrete-time state model is the same. However, the receiver's clock state model is slightly different and, thus, will be described now. In figure 3.5 is a representation of the clock's state model, where $u_\phi(t)$ and $u_f(t)$ are independent zero-mean white Gaussian noises characterized by the following covariance matrix:

$$Q_u = \begin{bmatrix} q_\phi & 0 \\ 0 & q_f \end{bmatrix} \tag{3.33}$$

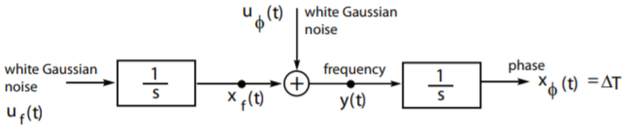


Figure 3.5: Receiver's clock state model [34].

with the variances q_ϕ and q_f being the power spectral densities (PSD) of $u_\phi(t)$ and $u_f(t)$ respectively.

The receiver's clock state model equation, for continuous-time is

$$\begin{bmatrix} \dot{x}_\phi(t) \\ \dot{x}_f(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_\phi(t) \\ x_f(t) \end{bmatrix} + \begin{bmatrix} u_\phi(t) \\ u_f(t) \end{bmatrix} \tag{3.34}$$

and the corresponding discrete-time version

$$\begin{bmatrix} x_{\phi,\,k+1} \\ x_{f,\,k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x_{\phi,\,k} \\ x_{f,\,k} \end{bmatrix} + \begin{bmatrix} u_{\phi,\,k} \\ u_{f,\,k} \end{bmatrix} \tag{3.35}$$

The noise covariance matrix, in discrete time, can be expressed as:

$$\begin{aligned} \widetilde{Q}_k &= \int_{t_k}^{t_{k+1}} A(t_{k+1},\tau) Q_u A(t_{k+1},\tau)^T \, d\tau \\ &= \begin{bmatrix} q_\phi \Delta t + \frac{q_f (\Delta t)^3}{3} & \frac{q_f (\Delta t)^2}{2} \\ \frac{q_f (\Delta t)^2}{2} & q_f \Delta t \end{bmatrix} \end{aligned} \tag{3.36}$$

The variances q_ϕ and q_f can be described as [34]:

$$\begin{aligned} q_\phi &\approx \frac{h_0}{2} \\ q_f &\approx 2\pi^2 h_{-2} \end{aligned} \tag{3.37}$$

with h_0 and h_{-2} being the Allan variance parameters [35]. In [34], several values for the Allan variance parameters can be found. In the simulations undertaken during the course of this thesis, it was assumed that the receiver's clock oscillator was a temperature compensated crystal and, therefore, according to [34] we have $h_0 = 2 \times 10^{-19}$ and $h_{-2} = 2 \times 10^{-20}$. It should be noted that, when treating clock deviations as errors in meters, the Allan variance parameters must be multiplied by the speed of light squared ($c^2 = 9 \times 10^{16}$).

Now, we can write the full dynamics' model for the receiver. The state transition matrix is $\Phi =$ diag$\{a\,a\,a\,A\}$ and the dynamics noise covariance matrix is $Q_k =$ diag$\{Q'_k\,Q'_k\,Q'_k\,\widetilde{Q}_k\}$. Thus, we have:

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \\ x_{4,k+1} \\ x_{5,k+1} \\ x_{6,k+1} \\ x_{7,k+1} \\ x_{8,k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \Delta t & & & & & & \\ & 1 & & & & & \mathbf{0} & \\ & & 1 & \Delta t & & & & \\ & & & 1 & & & & \\ & & & & 1 & \Delta t & & \\ & & & & & 1 & & \\ & \mathbf{0} & & & & & 1 & \Delta t \\ & & & & & & & 1 \end{bmatrix}}_{\Phi} \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \\ x_{4,k} \\ x_{5,k} \\ x_{6,k} \\ x_{7,k} \\ x_{8,k} \end{bmatrix} + \begin{bmatrix} u_{1,k} \\ u_{2,k} \\ u_{3,k} \\ u_{4,k} \\ u_{5,k} \\ u_{6,k} \\ u_{7,k} \\ u_{8,k} \end{bmatrix} \tag{3.38}$$

with the dynamics noise covariance matrix being:

$$Q_k = \begin{bmatrix} \frac{q_v(\Delta t)^3}{3} & \frac{q_v(\Delta t)^2}{2} & & & & & & \\ \frac{q_v(\Delta t)^2}{2} & q_v\Delta t & & & & \mathbf{0} & & \\ & & \frac{q_v(\Delta t)^3}{3} & \frac{q_v(\Delta t)^2}{2} & & & & \\ & & \frac{q_v(\Delta t)^2}{2} & q_v\Delta t & & & & \\ & & & & \frac{q_v(\Delta t)^3}{3} & \frac{q_v(\Delta t)^2}{2} & & \\ & & & & \frac{q_v(\Delta t)^2}{2} & q_v\Delta t & & \\ & \mathbf{0} & & & & & q_\phi\Delta t + \frac{q_f(\Delta t)^3}{3} & \frac{q_f(\Delta t)^2}{2} \\ & & & & & & \frac{q_f(\Delta t)^2}{2} & q_f\Delta t \end{bmatrix} \tag{3.39}$$

Observations' model

The observations' model equation, presented in (3.22), is non-linear. We have that $z_k = [\rho_{1,k}, \cdots, \rho_{N,k}]$, where $\rho_{i,\, i=1,\cdots,N}$ represents the measured pseudorange from satellite i, with $N \geq 4$; and $h(x_k)$ is defined as:

$$h(x_k) = \begin{bmatrix} \sqrt{(X_1 - x_u)^2 + (Y_1 - y_u)^2 + (Z_1 - z_u)^2} + u_u \\ \vdots \\ \sqrt{(X_N - x_u)^2 + (Y_N - y_u)^2 + (Z_N - z_u)^2} + u_u \end{bmatrix} \tag{3.40}$$

where X_i, Y_i and Z_i represent the i^{th} satellite's coordinates in the ECEF frame. The observations noise covariance matrix is expressed as [31]:

$$\widetilde{R}_k = \begin{bmatrix} \sigma^2_{1,UERE} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma^2_{N,UERE} \end{bmatrix} \tag{3.41}$$

In this thesis, we assume that all satellites share the same $\sigma^2_{i,UERE} = 1.4$ m (user equivalent range error) value, which can be defined using the parameters present in [36].

As was stated in 3.2, we can express the observations matrix according to (3.24):

$$H_k = \left[\frac{\partial h_i[\hat{x}(k|k-1)]}{\partial x_j}\right]_{N\times l} \tag{3.42}$$

with n being the number of satellites used and l being the size of the state vector, which in the case of the PVT model is $l = 8$. For the PVT model we obtain:

$$H_k = -\begin{bmatrix} a_{x_1} & 0 & a_{y_1} & 0 & a_{z_1} & 0 & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x_N} & 0 & a_{y_N} & 0 & a_{z_N} & 0 & -1 & 0 \end{bmatrix} \tag{3.43}$$

with:

$$\begin{aligned} a_{x_i} &= \frac{X_i - \hat{x}_u}{r_i} \\ a_{y_i} &= \frac{Y_i - \hat{y}_u}{r_i} \\ a_{z_i} &= \frac{Z_i - \hat{z}_u}{r_i} \\ r_i &= \sqrt{(X_i - \hat{x}_u)^2 + (Y_i - \hat{y}_u)^2 + (Z_i - \hat{z}_u)^2} \end{aligned} \tag{3.44}$$

where:

$$[\hat{x}_u \, \hat{y}_u \, \hat{z}_u]^T = [\hat{x}_1(k|k-1) \, \hat{x}_3(k|k-1) \, \hat{x}_5(k|k-1)]^T \tag{3.45}$$

3.2.2 Integration of the EKF algorithm in a scalar receiver

In a scalar receiver, similarly to the least-squares algorithm, the EKF can also be used in an ”off the shelf” approach. It just needs to be fed with a set of pseudoranges, at each t_k iteration from a set of DLL’s. As in 3.1.1, there is no feedback of information between the estimation algorithm and the DLL’s that feed it.

It should be noted that, unlike the Least-Squares algorithm, the EKF algorithm with the PVT model is capable of estimating the receiver’s velocity in all three axes and the receiver’s clock drift, without needing an observation that contains that information (pseudoranges only possess information on the receiver’s position and clock bias). Information for those receiver’s states could be obtained through the usage of FLL’s, as described in [23]. The structure of a scalar receiver with the EKF algorithm is the same as in figure 3.1.

3.2.3 Integration of the EKF algorithm in a vector receiver

The EKF algorithm analyzed in 3.2 cannot be directly applied to a vector receiver, as in the scalar case. This is due to the fact that the estimation algorithm, in a vector structure, must close a feedback loop, with regard to a set of code discriminators (and possibly a set of phase discriminators). As such, the algorithm must be modified. In this analysis, we will follow an adaptation of the model first proposed in

[23] for a VDFLL and further utilized in [24] and [25].

The vector tracking algorithm that is herein implemented is the VDLL, meaning only the code tracking loops are coupled with the EKF estimation algorithm. Therefore, we assume perfect carrier phase synchronization. We consider the same filtering and prediction step equations presented in 3.2, which are (3.26) and (3.28), respectively. We also consider equation (3.27) for the extended Kalman filter gain. The core difference between the scalar and vector variants resides in the innovations process $z_k - h\left[\hat{x}(k|k-1)\right]$. In the vector application of the EKF algorithm, there is no direct access to the two terms that compose the innovations process. Instead, it is described through the outputs of the several code discriminators:

$$z_k - h\left[\hat{x}(k|k-1)\right] = \begin{bmatrix} \gamma_{c_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \gamma_{c_N} \end{bmatrix} \begin{bmatrix} D_{c_1} \\ \vdots \\ D_{c_N} \end{bmatrix} \tag{3.46}$$

where $\left[D_{c_1}, \cdots, D_{c_N}\right]^T$ is the vector containing the outputs of the code discriminators and γ_{c_i} represents the code discriminator gain so that [23]:

$$\gamma_{c_i} D_{c_i} \approx c \cdot \epsilon_i \tag{3.47}$$

where c is the speed of light in vacuum and ϵ_i represents the code synchronization error. If the value of γ_{c_i} is shared among all satellites, equation (3.46) becomes:

$$z_k - h\left[\hat{x}(k|k-1)\right] = \gamma_c \begin{bmatrix} D_{c_1} \\ \vdots \\ D_{c_N} \end{bmatrix} \tag{3.48}$$

The right-hand members of equations (3.46) or (3.48) constitute the innovations process (or residuals) $z_k - h\left[\hat{x}(k|k-1)\right]$ in the update of the state vector in (3.26).

Another difference, regarding the EKF's implementation in a scalar receiver, is the way the $\widetilde{R}_k$ matrix is defined. In a vector receiver, with EKF, the matrix $\widetilde{R}_k$ is diagonal with its entries expressed as [23]:

$$r_{ii} = -\frac{c^2 \Delta}{2p\left(\frac{C}{N_0}\right)_i T}, \qquad i = 1, \cdots, N \tag{3.49}$$

where Δ is the code discriminator early-late spacing, T is the correlation interval and p takes the following values:

$$\begin{aligned} p &= -\frac{1}{T_c}, && \text{for BPSK signals} \\ p &= -\frac{3}{T_c}, && \text{for BOCs(1,1) signals} \\ p &= -\frac{(53 + 2\sqrt{10})}{11 T_c}, && \text{for CBOC(6,1,1/11) signals} \end{aligned} \tag{3.50}$$

with T_c being the chip time of the respective signal.

The feedback process in the vector receiver consists of computing $\hat{x}(k+1|k)$, utilizing the filtered estimate $\hat{x}(k|k)$, and rebuild a set of estimated pseudoranges, at the instant $k+1$, using equation (3.15). These estimated pseudoranges will be compared to a set of measured ones, with the difference between them - the residuals - being expressed through the code discriminator outputs D_{c_i}, as those outputs depend on the code delay error ϵ. The structure of a vector receiver with the EKF algorithm is the same as in figure 3.2.

Chapter 4

Scalar receiver simulation results

In this chapter, we analyze the performance of the scalar receiver, implemented with both the least-squares algorithm and the EKF algorithm, in various scenarios. Section 4.1 presents the obtained results with a scalar receiver with the least-squares algorithm and also describes the simulation scenarios utilized to test the receiver performance throughout this thesis. Section 4.2 presents the obtained performance results for a scalar receiver with the EKF algorithm.

4.1 Scalar receiver with least-squares algorithm

4.1.1 Standard scenario simulation results

The first scenario in which the performance of both the scalar and vector receivers is analyzed, is a *Standard Scenario*, with the following characteristics:

- Receiver initially located at the IST Alameda campus, in Lisbon, Portugal ($38^\circ\ 44'\,12.276''$N, $\quad 9^\circ\ 8'\,18.212''$W).
- Constellation of $N \geq 4$ satellites, that minimizes the GDOP parameter, computed as GDOP $=$ $\sqrt{\text{tr}\{(G^T G)^{-1}\}}$, where $G.$ is the matrix defined in (3.3).
- Same value of carrier-to-noise ratio C/N$_0$ shared across all satellites, kept constant throughout the duration of the simulation.
- Initial estimate of the receiver's position equal to its actual position
- No initial positioning errors due to receiver's clock delay
- Estimated and real code delay equal to one another at the start of the simulation
- No inter-satellite interference
- No atmospheric influence (such as rain, clouds, ionospheric scintillation or tropospheric attenuation, etc) over the signal quality
- BPSK signal and infinite receiver bandwidth

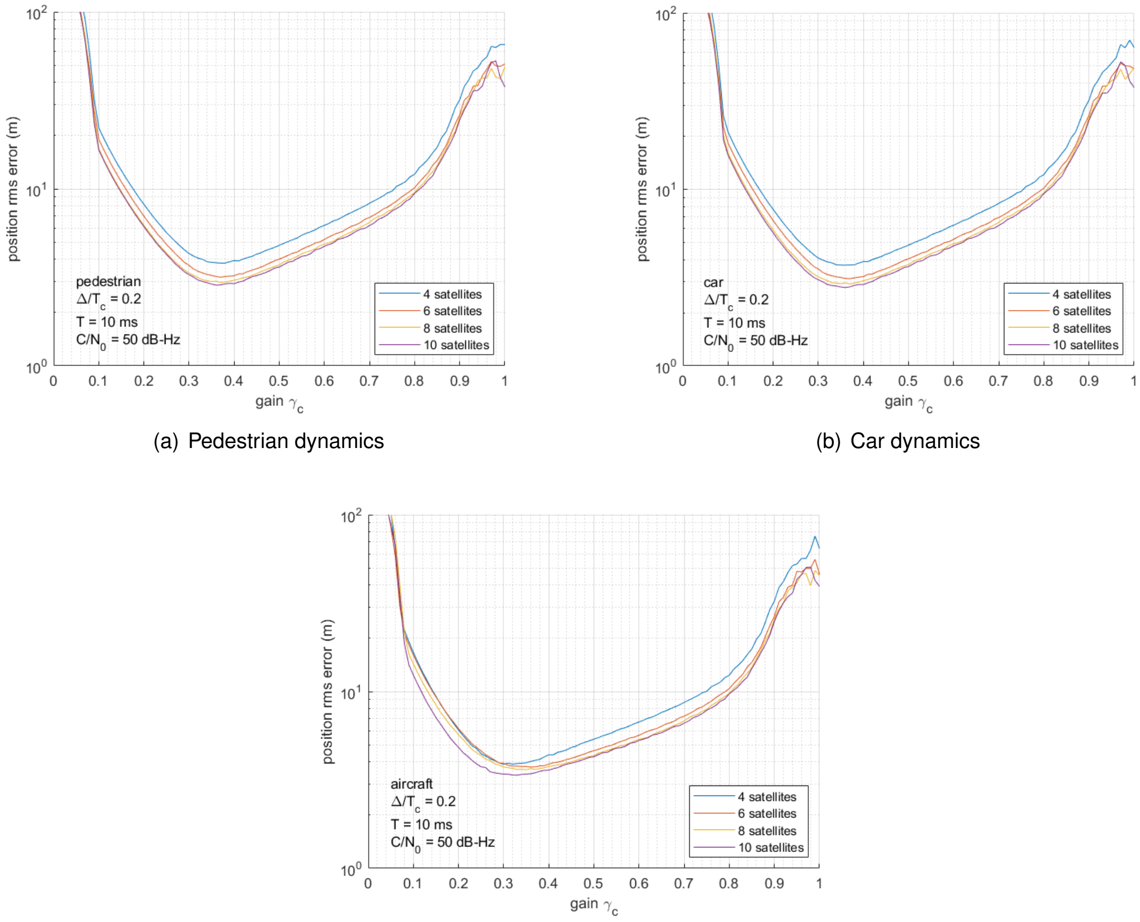
- Simulation interval of 100 seconds.

- All positions of satellites and receiver are computed in the ECEF coordinate system.

The aim of this simulation is to measure the root mean square (rms) error of the generalized position vector ΔX (including the component of the receiver clock bias), versus the γ_{c_i} values of the DLL code discriminator feedback gain. This rms error e_{rms} is be defined as:

$$e_{rms}(\gamma_{c_i}) = \sqrt{\frac{\sum_{k=1}^{n}(\hat{x}_{u,k}(\gamma_{c_i}) - x_{u,k})^2 + (\hat{y}_{u,k}(\gamma_{c_i}) - y_{u,k})^2 + (\hat{z}_{u,k}(\gamma_{c_i}) - z_{u,k})^2 + (\hat{u}_{u,k}(\gamma_{c_i}) - u_{u,k})^2}{n}} \tag{4.1}$$

where $k = 1, \ldots, n$ is the index of time and n is the number of samples used, for each run of index i (different value of γ_{c_i}). The number n varies with the chosen DLL correlation interval T (as such the sampling interval is T), so that multiplying one by the other, we obtain a simulation interval of 100 seconds. For the scalar receiver, the value of γ_{c_i} was swept between 0 and 1 in steps of 0.01, which leads to $i = 1, \ldots, 100$. In figure 4.1 we can see the results obtained for constellations with different number of satellites, all visible from the starting point of the receiver: the IST Alameda campus in Lisbon.



(a) Pedestrian dynamics

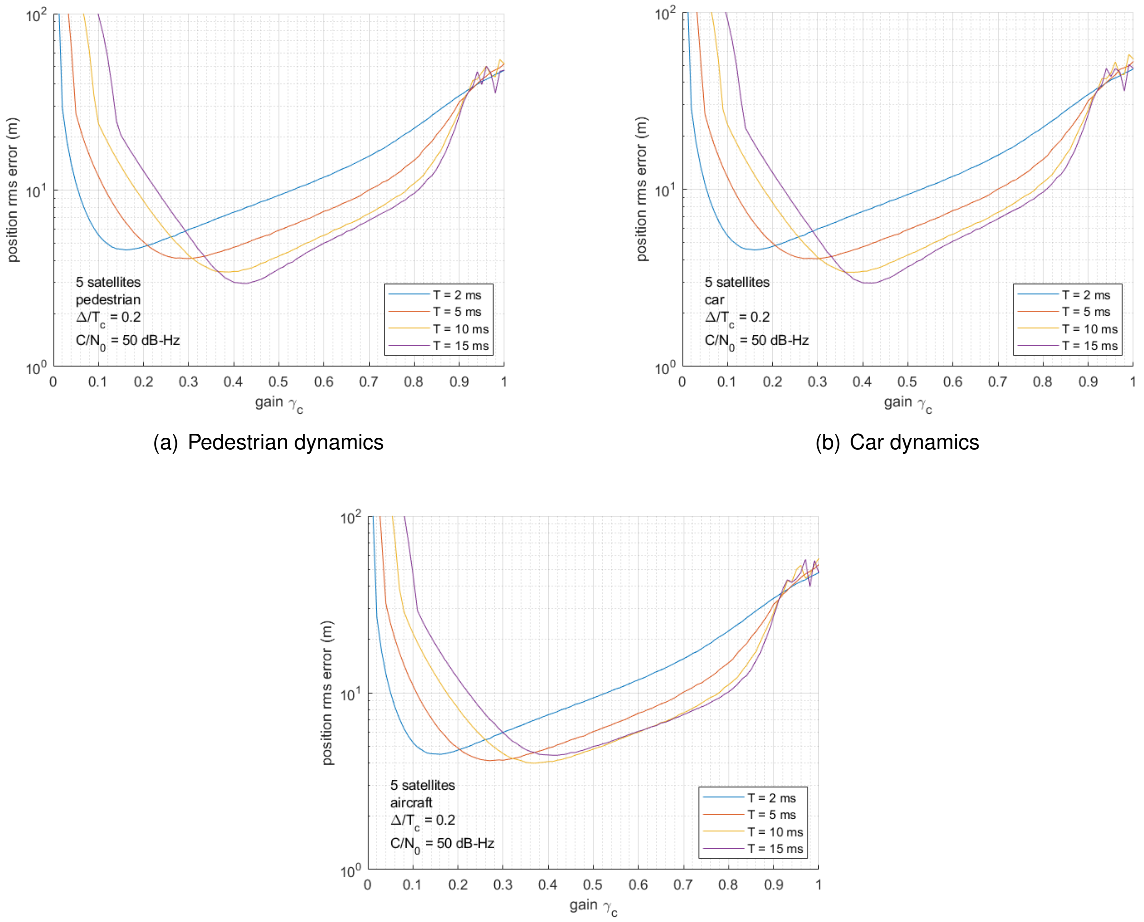
(b) Car dynamics

(c) Aircraft dynamics

Figure 4.1: Results for the scalar receiver with varying number of satellites (least-squares).

Three different receiver dynamics were tested: pedestrian, car and aircraft. In the pedestrian case, the receiver motion is characterized, in the ECEF coordinate system, by $v_x = v_y = v_z = 1$ m/s; in the car case, the receiver presents $v_x = v_y = v_z = 20$ m/s; and finally in the aircraft case, the receiver presents $v_x = v_y = v_z = 130$ m/s. These scenarios correspond, respectively, to velocities of 6.2 Km/h, 125 Km/h and 811 Km/h. The GPS satellite trajectories were calculated using the SEM almanacs (2019-02-19) available at [37].

It is clear that, as the number of satellites in the constellation increases, the rms error diminishes, with the effects of increasing from 6 to 10 satellites being of little relevance. There is one exception in the aircraft dynamics case, where there is a noticeable performance increase, going from 8 to 10 satellites in use. It is also clear that the scalar receiver has little sensibility towards the adopted receiver dynamics, with negligible differences between the pedestrian and car scenarios. For the aircraft case, performance degrades slightly, never reaching a difference of 1 m. Also, regardless of the number of satellites in use, the scalar receiver presents an optimal value for the gain γ_c, which produces the smallest rms error. This value of rms error ranges from 3 m to 4 m. With the same correlation interval T, the optimum value of γ_c is practically the same in the pedestrian and car cases, being around $\gamma_c = 0.4$, changing slightly for the aircraft case, where it leans more towards $\gamma_c = 0.3$. However, varying the value of T changes drastically the optimum code discriminator gain, implying the γ_c/T relation is constant. The results for this simulation can be seen in figure 4.2.



(a) Pedestrian dynamics

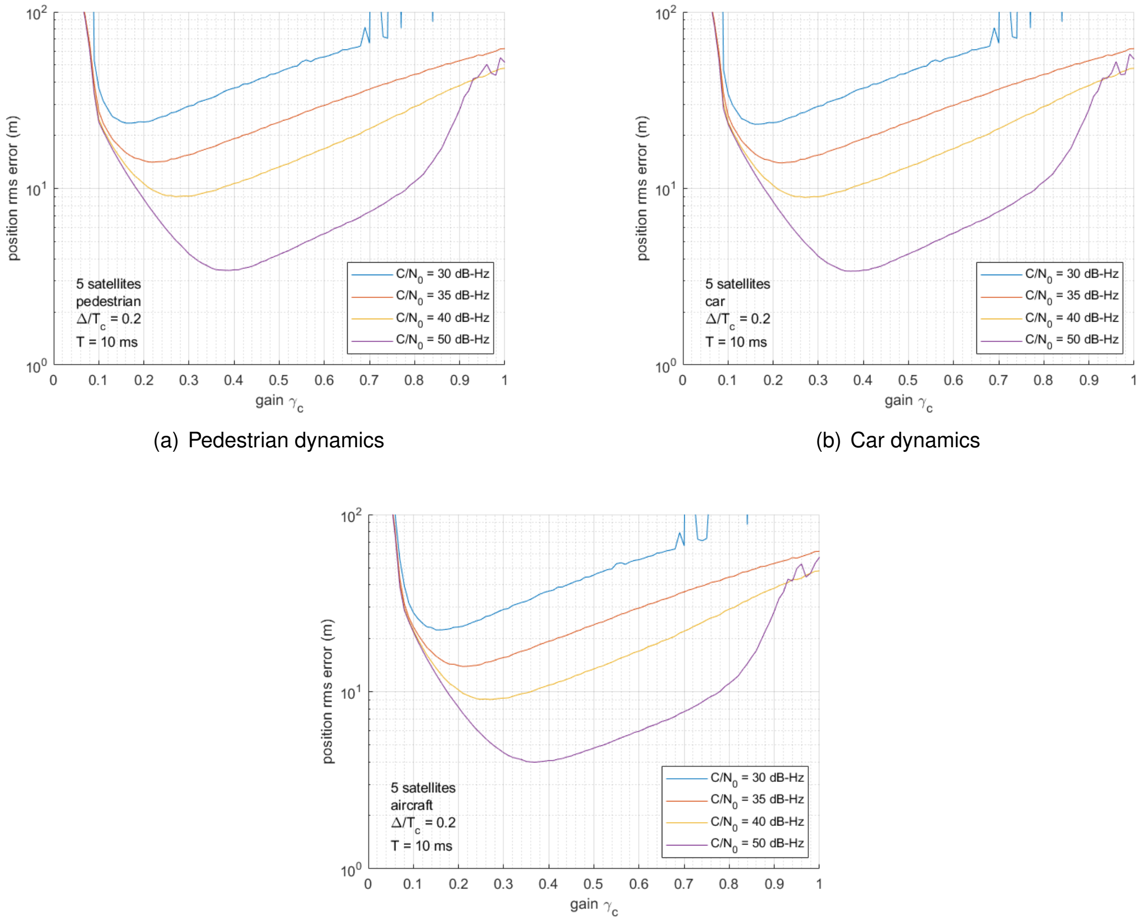
(b) Car dynamics

(c) Aircraft dynamics

Figure 4.2: Results for the scalar receiver with varying correlation interval T (least-squares).

Once again, the receiver dynamics has little influence on the optimal value of γ_c, although performance, for the aircraft case, is somewhat degraded, especially for higher values of T. The rms error, for the optimal γ_c, reaches values as low as 3 m for $T = 15$ ms and $\gamma_c = 0.41$ and as high as around 5 m for $T = 2$ ms and $\gamma_c = 0.16$ for the pedestrian and car cases. For the aircraft dynamics case, the lowest rms error obtained is 5 m for $T = 10$ ms and $\gamma_c = 0.37$ which differs from the previous scenarios, where increasing the correlation interval always produced better performance results.

Let us now analyze what happens if the carrier-to-noise ratio C/N$_0$ is degraded in all satellites in use. When we degrade the C/N$_0$ from 50 dB-Hz to 40 dB-Hz, in figure 4.3, it becomes clear that the performance of the scalar receiver, independently of receiver dynamics, becomes reasonably worse, with the rms error reaching a minimum of just 9 m. This value is very poor and as such, renders the receiver quite useless for accurate positioning. Further degrading C/N$_0$ yields even worse results, with actual loss of receiver tracking for C/N$_0$ $= 30$ dB-Hz, represented by the spikes in its respective curve in figure 4.3. This scenario, where all satellites in use suddenly share a low value for the carrier-to-noise ratio, is a very unlikely one and is only being presented for the sake of a performance comparison study between the scalar and vector receivers. More realistic scenarios will be simulated in section 4.1.2, for example.



(a) Pedestrian dynamics

(b) Car dynamics

(c) Aircraft dynamics

Figure 4.3: Results for the scalar receiver with varying C/N$_0$ in all satellites (least-squares).

4.1.2 Shadowing scenario simulation results

The shadowing scenario aims to portray a very common event in signal reception, which is the temporary loss or severe attenuation of signal from one or more satellites. Such events can happen due to obstacles in the reception path, both natural (trees, hills, etc) and man-made (such as tall buildings, walls, etc).

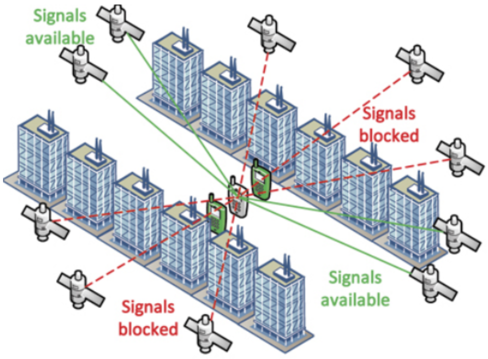


Figure 4.4: Illustration of an urban shadowing scenario (urban canyon) [38].

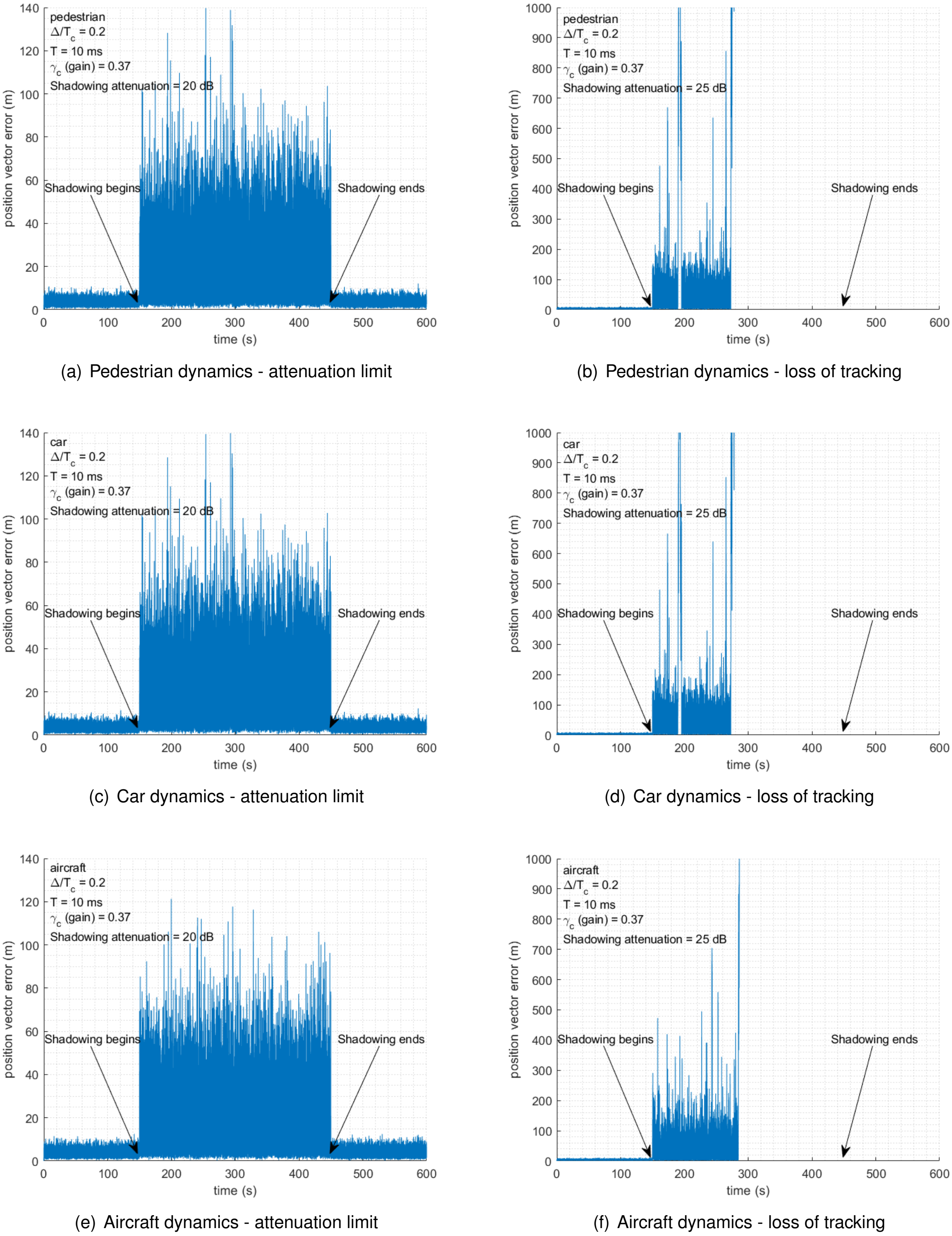
The results depicted below, for the scalar receiver, were obtained with the characteristics similar to the ones in section 4.1.1, with the following exceptions:

- Constellation of $N = 4$ satellites, in order to simulate the worst possible conditions in terms of available satellites.
- Value of C/N$_0$ $= 50$ dB-Hz shared across all satellites, with exception of the one whose signal is being blocked, which will be varied in several simulations, in increments of -5 dB. The blocked signal originates from the same satellite in each simulation.
- Fixed value of the code discriminator gain γ_c was chosen as the optimum for each receiver dynamics.
- Simulation interval of 10 minutes, with 5 minutes of shadowing.

We consider now an instantaneous position vector error at each instant kT, defined as:

$$e_k = \sqrt{(\hat{x}_{u,k} - x_{u,k})^2 + (\hat{y}_{u,k} - y_{u,k})^2 + (\hat{z}_{u,k} - z_{u,k})^2 + (\hat{u}_{u,k} - u_{u,k})^2} \tag{4.2}$$

instead of the rms error, since we assume a fixed γ_c value and present the positioning errors as a function of time. In figure 4.5 we can observe the results for a scalar receiver in the pedestrian, car and aircraft cases. In order to avoid the inclusion of several graphs for each value of simulated shadowing attenuation, in each scenario, only the graph with the value of C/N$_0$ for the shadowed satellite that caused failure in tracking, $(\widetilde{\text{C/N}_0})$ dB-Hz, will be shown, alongside the C/N$_0$ tested immediately before, $(\widetilde{\text{C/N}_0})$ dB-Hz $+\ 5$ dB. Also, it was decided to limit the y-axis of the loss of tracking plots to 1000 m in terms of rms error, as to better visualize the receiver's behavior immediately after the beginning of the shadowing period.



(a) Pedestrian dynamics - attenuation limit

(b) Pedestrian dynamics - loss of tracking

(c) Car dynamics - attenuation limit

(d) Car dynamics - loss of tracking

(e) Aircraft dynamics - attenuation limit

(f) Aircraft dynamics - loss of tracking

Figure 4.5: Shadowing scenario results for the scalar receiver (least-squares).

We can clearly see that for a constellation of 4 satellites, where one is obstructed during 5 minutes (represented in the simulation by having a lower C/N$_0$), the performance quickly degrades for all three receiver dynamics, with no perceptible difference in behavior. During the shadowing period, with 4 satellites, the scalar receiver is quite useless in terms of accurate positioning, with position errors, for power attenuations of 20 dB, reaching as high as ≈ 140 m for pedestrian, car and aircraft receivers.

The receiver experiences loss of tracking for an attenuation in the shadowed satellite of 25 dB, for the three receiver dynamics considered, with errors increasing drastically during the shadowing period. The definitive loss of tracking is visible through the inability of the receiver in lowering the value of the position error after the shadowing period, with that value continuing to increase. It can be, therefore, concluded that receiver dynamics has little impact on a scalar receiver's performance in a shadowing scenario, with similar behavior registered across the pedestrian, car and aircraft cases.

This loss of tracking can be said to occur, indubitably, after about $150\,\mathrm{s}$ (two and a half minutes) into the shadowing period (near $300\,\mathrm{s}$). It should also be noted that in a case where the shadowed satellite's signal has zero power, the navigation equation system cannot be solved, as there would be 3 equations for 4 variables. As such, under those conditions, loss of tracking should be immediate, with the receiver being unable to operate at all (not even being able to present the very poor results depicted in the right column of figure 4.5).

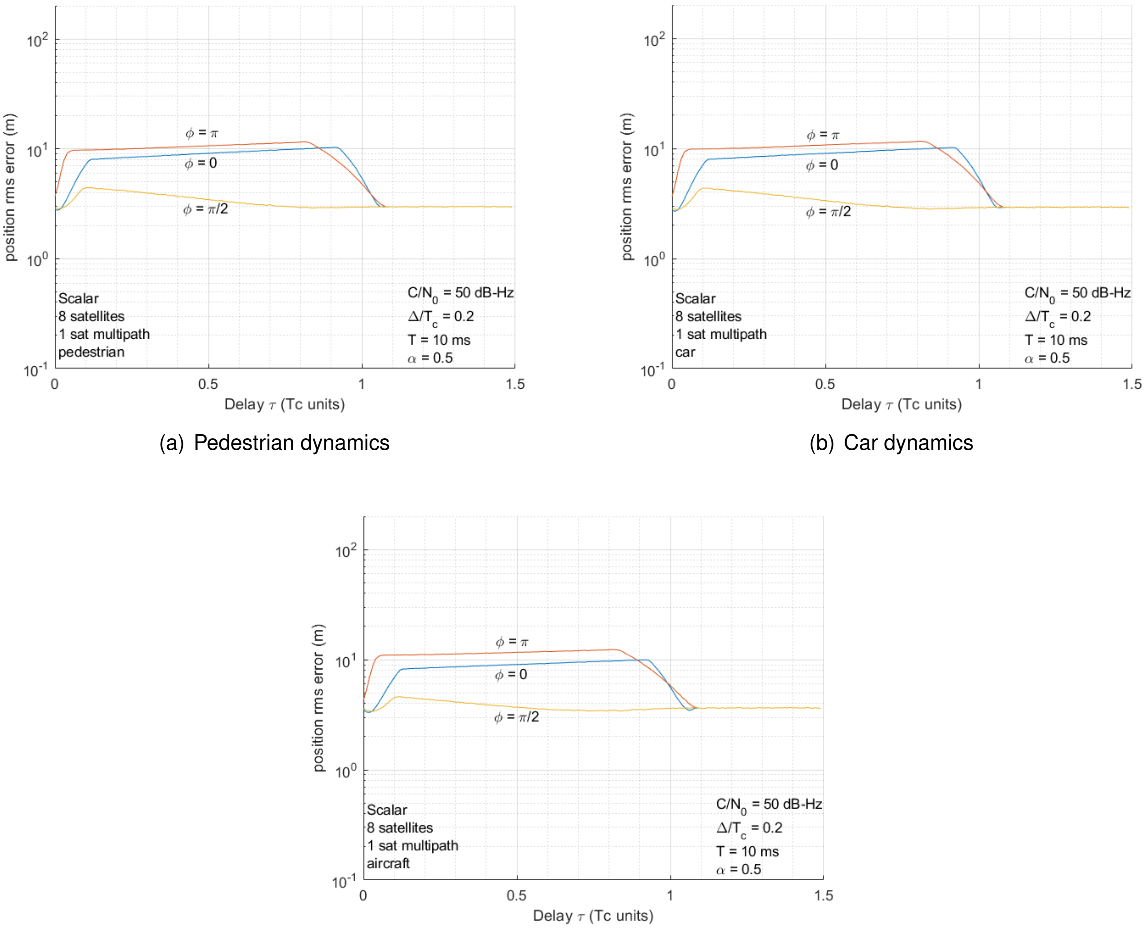
4.1.3 Multipath scenario simulation results

As discussed in 2.3, multipath scenarios occur when several signals, apart from the one in line-of-sight, reach the receiver, with attenuated amplitudes and slight time delays. In this simulation scenario, we will test the scalar receiver in a constellation of 8 satellites, where one satellite suffers from multipath effects, for pedestrian, car and aircraft dynamics. This simulation follows the same conditions present in the standard scenario, in section 4.1.1, albeit with the following exceptions:

- The satellite that suffers from multipath has its signal described, at the DLL, by equation (2.20), presented in section 2.
- It is the same satellite that suffers the multipath effects, across all simulations.
- The delayed signal has half the amplitude of the signal received in line-of-sight ($\alpha = 0.5$) and a phase difference of ϕ regarding that same signal. Several values of ϕ will be tested and kept constant during their respective simulations.
- Optimal performance values of γ_c were chosen for each receiver dynamics (pedestrian, car and aircraft) and the value for the delay τ of the delayed signal was swept between 0 and $1.5\,T_c$ units.

The results of the multipath scenario simulations for the scalar receiver, with the least-squares algorithm, are presented in figure 4.6. Several values for ϕ were simulated, with $\phi = 0$, $\phi = \pi/2$ and $\phi = \pi$ (radians) producing the most interesting results and, therefore, being the ones whose curves are displayed here. The values 0 and π represent, respectively, the delayed signal being inphase and in opposition of phase with the signal in line-of-sight. These two values of ϕ are the ones that originate the multipath envelopes seen in figure 2.6. The value $\pi/2$ is of particular interest as it is the value that nullifies the terms that depend on that variable in equation (2.21).

Let us analyze the obtained results. We can see that the worst performance results are obtained with $\phi = \pi$ and the best with $\phi = \pi/2$, for all three receiver dynamics. A quick analysis also shows that receiver dynamics seem to, once again, have little impact in the obtained results. For $\phi = \pi$, we



(a) Pedestrian dynamics

(b) Car dynamics

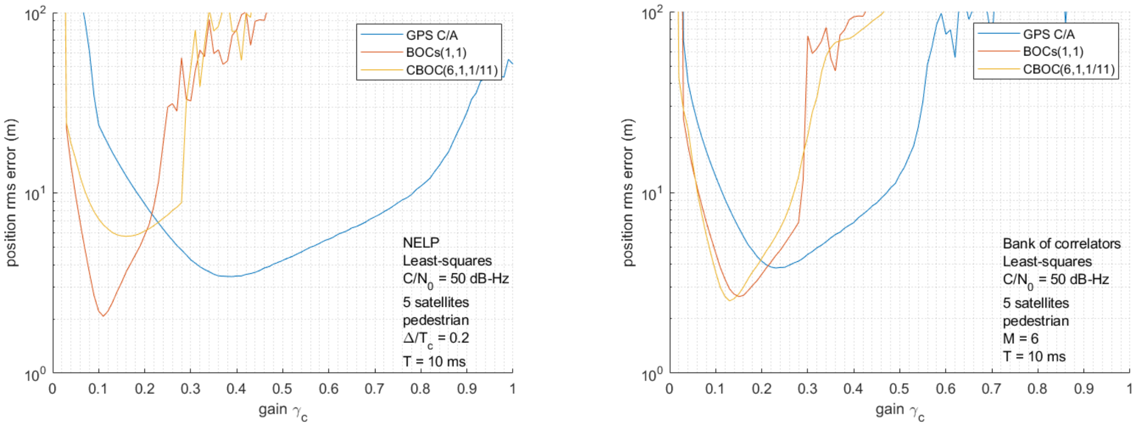
(c) Aircraft dynamics

Figure 4.6: Multipath scenario results for the scalar receiver (least-squares).

obtain the highest value registered for the rms error, with it slightly surpassing $10\,$m, for a delay of the secondary ray of $\tau \approx 0.85\,T_c$ units. After that peak, the rms error quickly decays to $\approx 3\,$m, which is the best performance value for this receiver with the least-squares algorithm (as can be seen in figure 4.3), with C/N$_0$ $= 50$ dB-Hz. For $\phi = 0$, for all three receiver dynamics, we observe a similar behavior to the $\phi = \pi$ curve. However, it presents a lower rms error than the red curve until approximately $0.87\,T_c$ units, where it exceeds the $\phi = \pi$ curve, reaching an error of around $10\,$m. Lastly, for $\phi = \pi/2$, we obtain the curve that presents best performance in a multipath scenario, for all three receiver dynamics. This curve presents a rms error peak, for $\tau \approx 0.1\,T_c$ units, of around $4\,$m, that decays to the already mentioned best value of $3\,$m, reaching it at $\tau \approx 0.7\,T_c$ units, for the three receiver dynamics. As a final note, it should be noted that multipath has no effect in the performance of the receiver for multipath delays bigger than $\approx 1,1\,T_c$ units, presenting a behavior that attempts to mimic the curves in figure 2.6, in chapter 2.

4.1.4 Signal comparison simulation results

Here, the results for a comparative analysis of the GPS C/A signal (BPSK), the BOCs(1,1) and the Galileo CBOC(6,1,1/11) are presented. The simulation scenario utilized was the same as in 4.1.1, while keeping a constellation of 5 satellites, $T = 10\,\mathrm{ms}$ and C/N$_0$ $= 50$ dB-Hz and considering pedestrian dynamics and an optimal value of γ_c. The obtained results can be viewed in figure 4.7



(a) NELP discriminator

(b) Bank of correlators

Figure 4.7: Signal performance comparison for a scalar receiver (least-squares).

It was chosen to only include the results for pedestrian dynamics as it was the one that produced the best results (in terms of rms error) and for the sake of brevity. For the presented results, two identical receivers, with exception of their code discriminator, were utilized. One uses the NELP discriminator, which has been the standard for all simulations so far; and the other uses a newly devised bank of correlators. One of the advantages of the bank of correlators is that we may synthesize any desired S-curve for the code discriminator, thus avoiding the occurence of the false-code lock effect [39]. A full breakdown on the bank of correlators can be found in appendix C, explaining its operation and benefits over the NELP discriminator. In the following simulations, we used a bank with $2\,\mathrm{M} = 12$ correlators.

Let us now analyze the obtained results. For the NELP discriminator, we obtain minimum rms errors of $\approx 3.5\,\mathrm{m}$ for the C/A signal, $2\,\mathrm{m}$ for the BOCs(1,1) and $\approx 6\,\mathrm{m}$ for the CBOC(6,1,1/11). Both the BOCs(1,1) and CBOC(6,1,1/11) present high rms errors for $\gamma_c > 0.3$, while presenting a curve particularly narrow around their best obtained value for rms errors. For the bank of correlators we obtain rms errors of $4\,\mathrm{m}$ for the C/A signal and less than $3\,\mathrm{m}$ for both the BOCs(1,1) and CBOC(6,1,1/11) signals, with these last two signals exhibiting a quite similar behavior. It is visible that with the bank of correlators, the C/A signal presents worse performance, when compared to utilizing a NELP discriminator, whereas the CBOC(6,1,1/11) signal's performance improves drastically. The BOCs(1,1) signal presents the best performance registered with the NELP discriminator, degrading slightly to match the CBOC's one with the bank of correlators. However, the performance values presented in figure 4.7 should not constitute the sole deciding factor over signal and code discriminator, as the multipath envelopes presented in figure 2.6 and the analysis in appendix C also bear significant weight in the overall comparison.

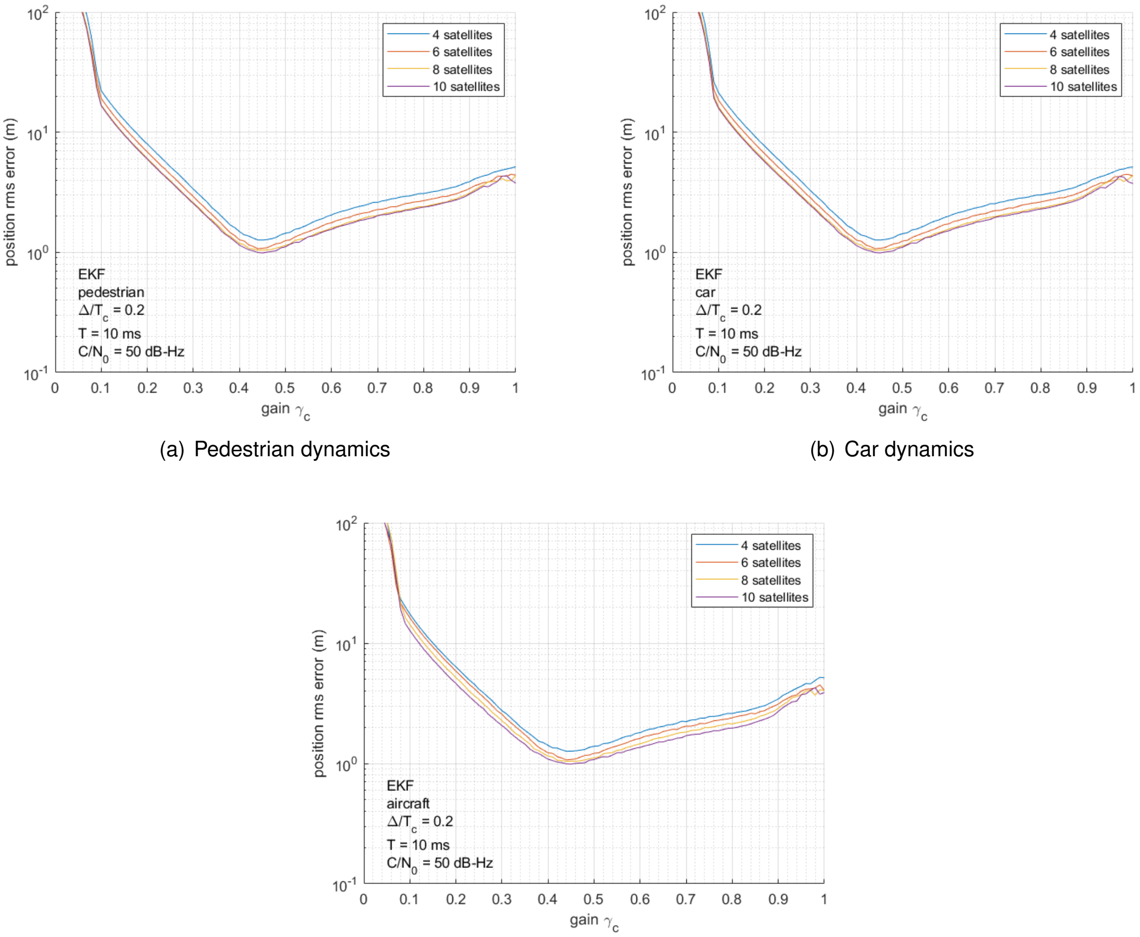
On a side note, it was demonstrated that with the bank of correlators, the CBOC(6,1,1/11) signal, utilized in the *Galileo* system could offer better performance than the GPS C/A signal with a NELP discriminator, which are the standard signal and code discriminator utilized in GNSS systems.

4.2 Scalar receiver with extended Kalman filter algorithm

Even though the main objective of this thesis is to compare the performance between scalar and vector receiver architectures and not between estimation algorithms, we can expect that, for the scalar receiver, the EKF algorithm will outperform the least-squares algorithm with visible improvements. Some comments on the comparative performance of these two algorithms, for the same receiver, will be made nonetheless.

4.2.1 Standard scenario simulation results

Let us analyze the results for the scalar receiver with the EKF algorithm, with varying number of satellites, displayed in figure 4.8.



(a) Pedestrian dynamics

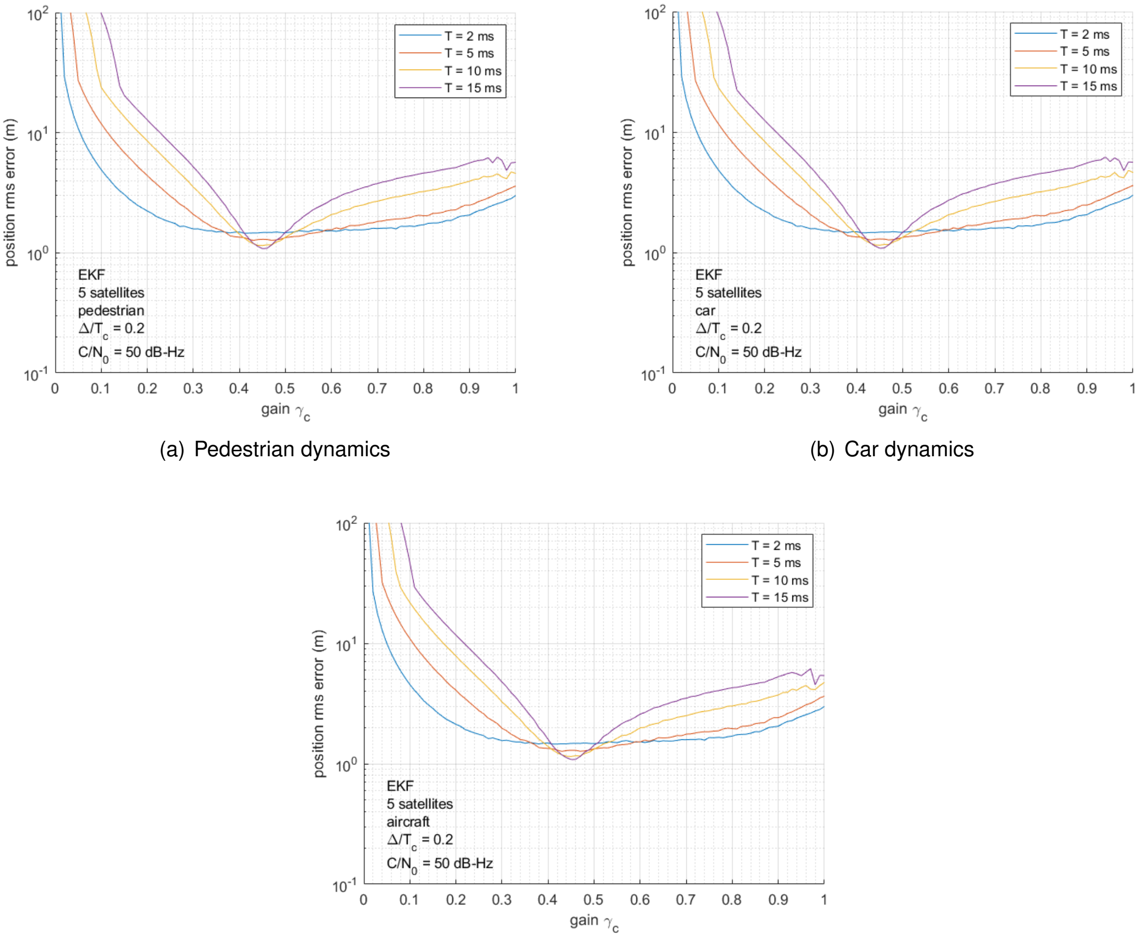
(b) Car dynamics

(c) Aircraft dynamics

Figure 4.8: Results for the scalar receiver with varying number of satellites (EKF).

Similarly to the results obtained by using the least-squares algorithm, in figure 4.1, the scalar receiver with the EKF algorithm seems to presents an optimal value for the γ_c gain. This value, in this simulation, appears to be the same across the three receiver dynamics, being $\gamma_c \approx 0.45$. In fact, the effect of the receiver dynamics seem to be almost negligible, with the three sets of curves presented being virtually similar. Performance improves as the number of satellites in use increases, but with very little changes when that number increases beyond 6. The lowest value for the rms error registered is of ≈ 1 m, obtained with the aforementioned value of γ_c, with that error being the same for the three receiver dynamics considered. As such, it can be concluded that the EKF algorithm is in fact an improvement over the least-squares one.

Let us now analyze what happens, in terms of performance, when the value for the correlation interval T is altered, in figure 4.9.



(a) Pedestrian dynamics

(b) Car dynamics

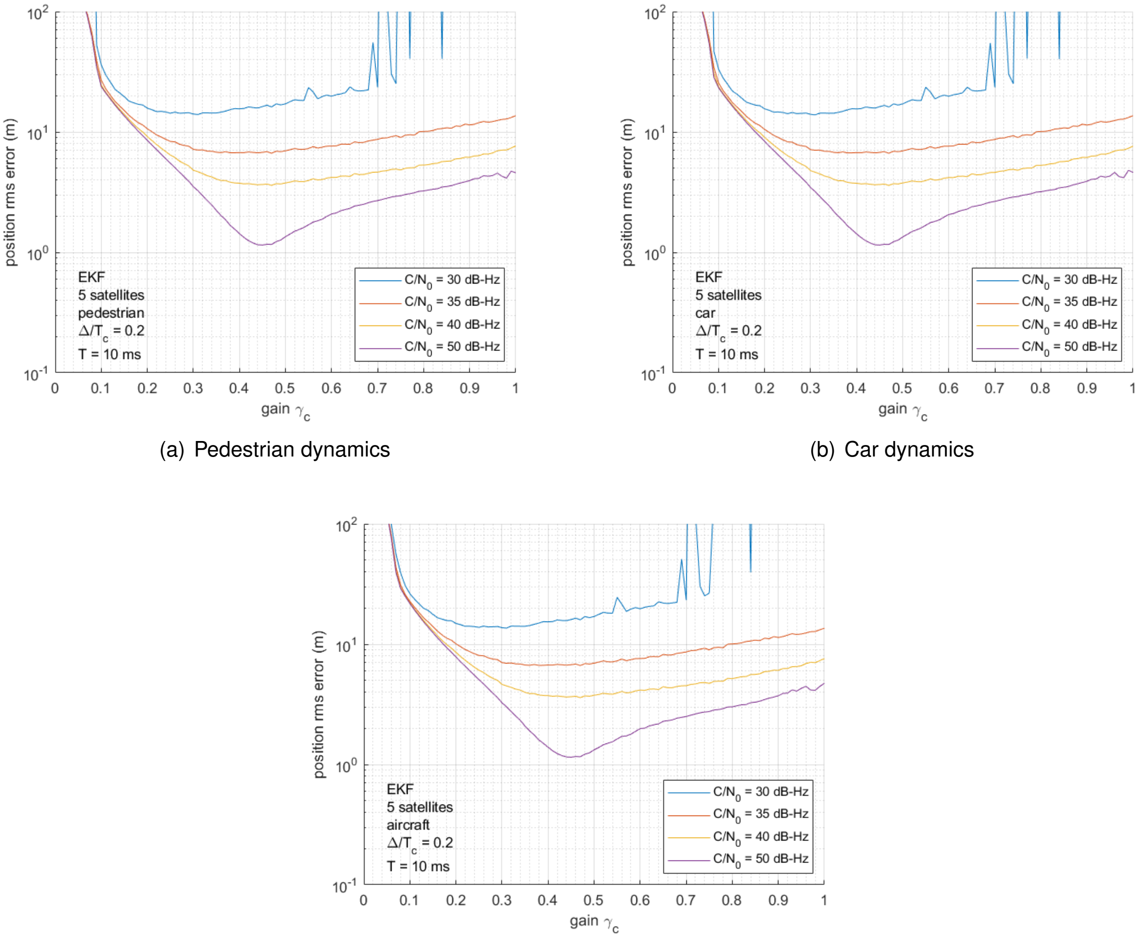
(c) Aircraft dynamics

Figure 4.9: Results for the scalar receiver with varying correlation interval T (EKF).

When compared to the least-squares algorithm (see figure 4.2), the results in varying T obtained with the EKF are very different, presenting a quite interesting behavior. Firstly, as has been pointed so far, the receiver dynamics does not influence the results obtained. Secondly, we can observe that across the whole spectrum of code discriminator gains γ_c, a correlation interval of $T = 2$ ms produces the best results with exception of values in the vicinity of $\gamma_c \approx 0.45$ (the optimal code discriminator gain, identified previously in figure 4.8). For that value of γ_c, $T = 15$ ms yields the best performance results (rms error of ≈ 1 m), with the recommended course of action being increasing the correlation interval.

Contrarily to what was observed with the least-squares algorithm, altering the value of T does not produce different optimal code discriminator gains, instead just changing the performance obtained with a fixed single optimal value of γ_c. That performance increases with an increase of the correlation interval, resulting in a curve that gradually becomes more sharpened around the optimal value of γ_c, as T increases. Only the curves for $T = 2$ ms don't exhibit this behavior, stabilizing in a constant value for rms error between $\gamma_c \approx 0.3$ and $\gamma_c \approx 0.75$.

Finally, let us analyze the curves by changing the C/N$_0$ values for all the satellites in the constellation in use. The results can be viewed in figure 4.10.



(a) Pedestrian dynamics

(b) Car dynamics

(c) Aircraft dynamics

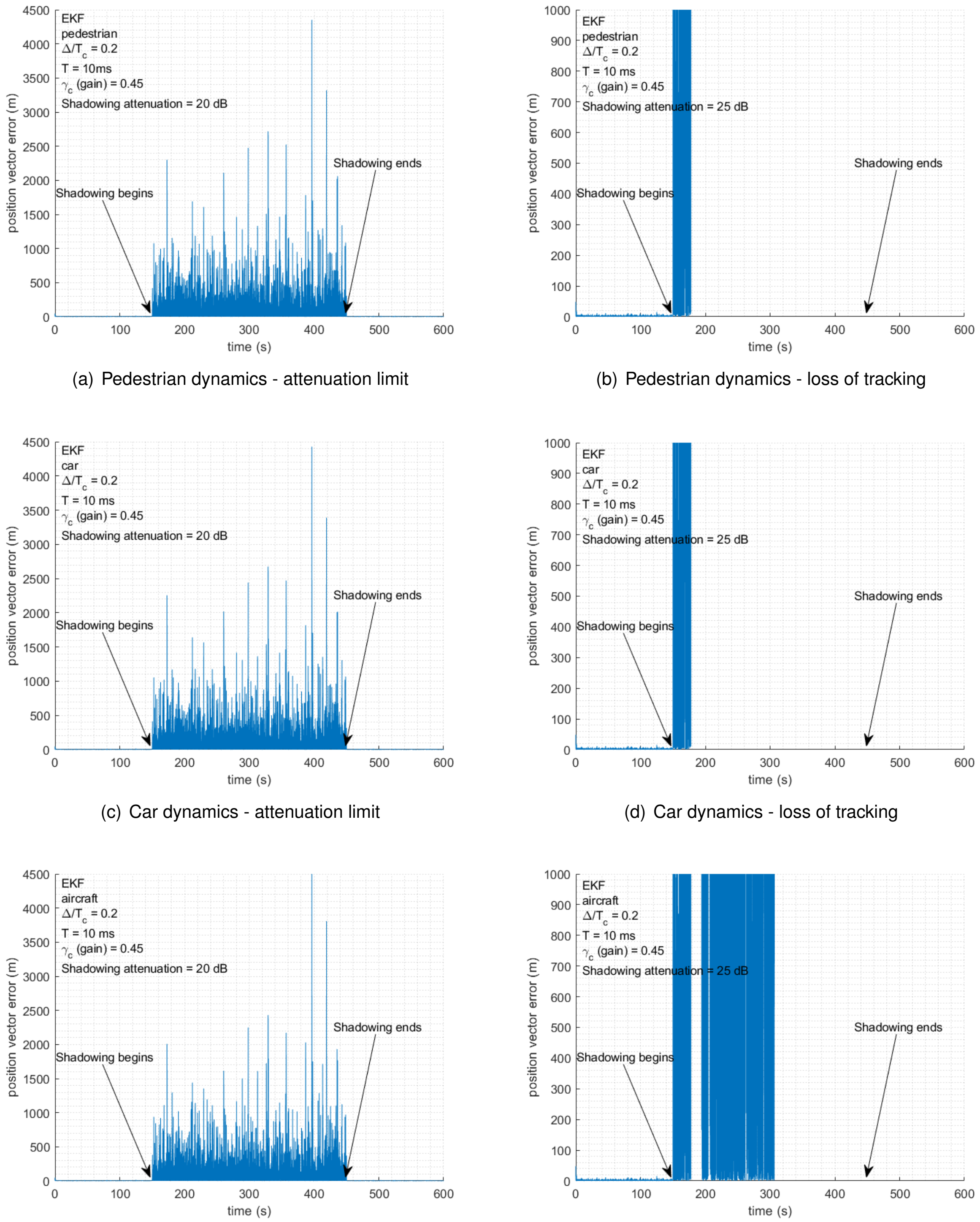
Figure 4.10: Results for the scalar receiver with varying C/N$_0$ in all satellites (EKF).

A first analysis of the presented curves shows that performance degrades gradually with a decrease in C/N$_0$, with loss of tracking occurring for C/N$_0 = 30$ dB-Hz. Receiver dynamics continue to bear little relevance for the obtained results, with the three sets of curves almost presenting the same shape. For C/N$_0 = 50$dB-Hz we obtain the best performance results, for the three dynamics, with rms errors nearing 1 m. As we degrade the carrier-to-noise ratio, performance degrades as well, with rms errors of ≈ 4 m obtained for C/N$_0 = 40$dB-Hz and rms errors of ≈ 7 m for C/N$_0 = 35$dB-Hz. Unlike with the least squares algorithm, the rms errors with the EKF can be kept below 10 m for C/N$_0 < 40$ dB-Hz, presenting, in fact, very acceptable results for 40 dB-Hz.

For C/N$_0 = 30$ dB-Hz the poorest results are obtained, with the lowest rms error obtained being ≈ 15 m. For values of $\gamma_c \geq 0.7$ the scalar receiver experiences loss of tracking, as the rms errors quickly increase, producing a series of spikes, as can be seen in figure 4.10.

4.2.2 Shadowing scenario simulation results

Let us analyze now the results obtained for a shadowing scenario similar to the one portrayed in section 4.1.2, as seen in figure 4.11.



(a) Pedestrian dynamics - attenuation limit

(b) Pedestrian dynamics - loss of tracking

(c) Car dynamics - attenuation limit

(d) Car dynamics - loss of tracking

(e) Aircraft dynamics - attenuation limit

(f) Aircraft dynamics - loss of tracking

Figure 4.11: Shadowing scenario results for the scalar receiver (EKF).

If we compare these results with the ones obtained for the least-squares algorithm, in figure 4.5, we clearly see that there is no difference whatsoever in the value of C/N$_0$ attenuation of the shadowed satellite that triggers the loss of lock. For all three receiver dynamics, tracking is lost for a power attenuation of 25 dB, while considering, once again, the optimal value of γ_c. Tracking is lost short after entering the shadowing period. During the shadowing period, we register position errors that reach the 4000 m thus making it impossible to maintain accurate receiver positioning. After the shadowing period (which ends at 450 s), the scalar receiver is unable to recover tracking, with the errors continuing to increase.

Even though it is not the purpose of this simulation to compare the performance of the receiver during the shadowing period, but rather its ability to resume tracking after that period, it should be noted that the EKF presents errors much bigger than the ones presented by the least-squares algorithm. The least-squares algorithm presents a maximum position error, during the shadowing period of ≈ 140 m, whereas the EKF presents one of ≈ 4000 m, in the case where tracking can be resumed. This disparity in in rms errors is most likely caused by the fact $\widetilde{R}_k$ is kept constant throughout all the EKF simulations and not tuned to optimally to each receiver dynamics.

4.2.3 Multipath scenario simulation results

The results for the multipath scenario simulations, for a scalar receiver with the EKF can now be seen in figure 4.12.

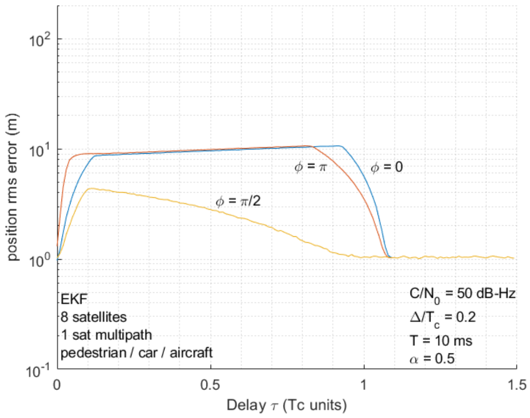


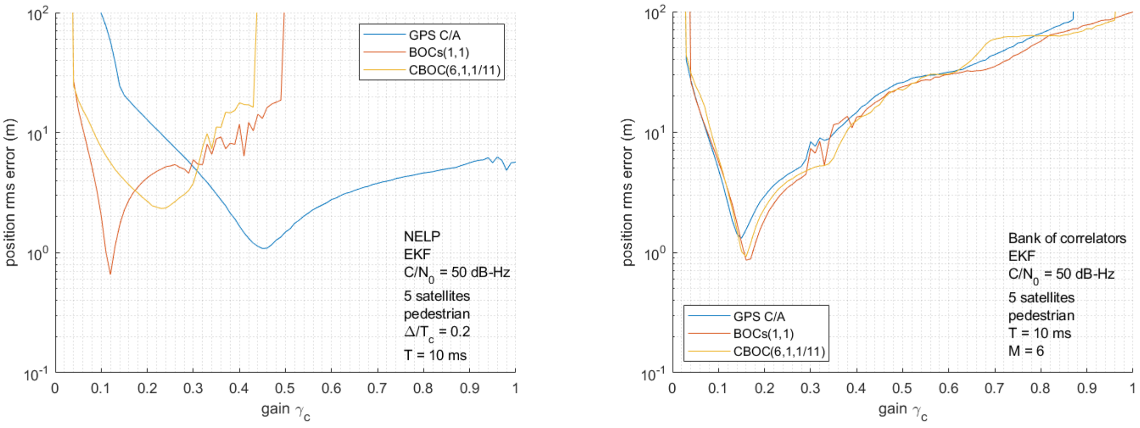
Figure 4.12: Multipath scenario results for the scalar receiver (EKF).

The obtained results for the scalar receiver with the EKF did not present any differences regarding receiver dynamics whatsoever. Therefore, for the sake of simplicity, it was chosen to only present one graph, demonstrating the results for the three considered receiver dynamics. Comparing these results with the ones presented in figure 4.6, we can see that they are very much the same, with very subtle differences. Therefore, using a more sophisticated estimation algorithm seems to have no impact whatsoever on receiver performance subject to multipath effects.

By analyzing figure 4.12, we can observe that the $\phi = 0$ and $\phi = \pi$ curves overlap for most of the simulated τ values. They present a maximum rms error of just little over 10 m, that then decays to 1 m, starting from the $\tau = 0.85\,T_c$ units and the $\tau = 0.95\,T_c$ units for $\phi = 0$ and $\phi = \pi$ respectively. For $\phi = \pi/2$ we observe a peak in rms error of 4 m for $\tau \approx 0.15\,T_c$ units, that will degrade gradually to 1 m.

4.2.4 Signal comparison simulation results

Let us now analyze the results obtained by comparing the performance of the GPS C/A signal (BPSK) with BOCs(1,1) and CBOC(6,1,1/11) signals, in figure 4.13.



(a) NELP discriminator

(b) Bank of correlators

Figure 4.13: Signal performance comparison for a scalar receiver (EKF).

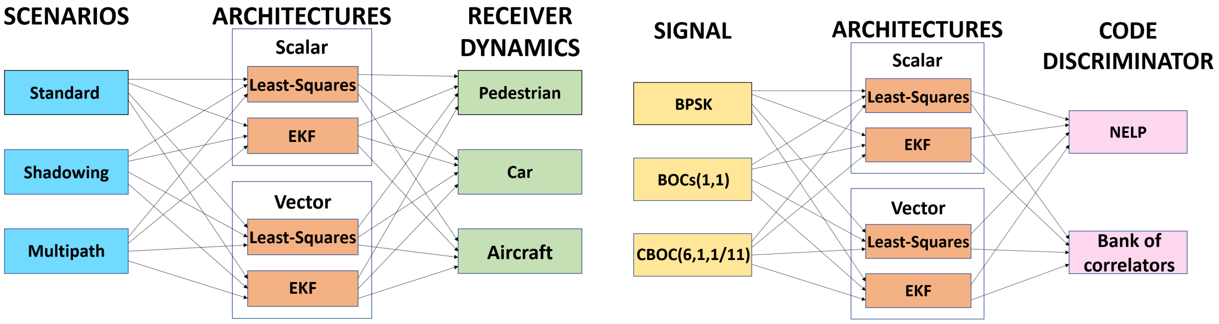
Once again, as expected, the results improve by utilizing the EKF estimation algorithm, instead of the least-squares. With the NELP discriminator, we obtain the following best performance results, in terms of rms errors: 1 m for the C/A signal, ≈ 0.65 m for the BOCs(1,1) signal and ≈ 2.5 m for the CBOC(6,1,1/11). For the bank of correlators we obtain ≈ 1.5 m for the C/A signal, ≈ 0.85 m for the BOCs(1,1) signal and 0.9 m for the CBOC(6,1,1/11) signal.

When using the NELP discriminator, the three curves exhibit distinct behaviors with loss of lock occurring with the BOCs(1,1) signal for $\gamma_c \approx 0.5$ and with the CBOC(6,1,1/11) signal for $\gamma_c \approx 0.4$. The three curves practically overlap when using the bank of correlators, with only the already mentioned differences in the best achieved values of rms errors. Like with the results obtained for the scalar receiver with the least-squares algorithm, there is no advantage in utilizing the bank of correlators for reception of the C/A signal, with the opposite being true for the CBOC signal, with the obtained results improving greatly. The BOCs(1,1) once again presents the best performance results in the simulated scenario but, as mentioned in 4.1.4, other factors must be taken into consideration when evaluating global performance.

Chapter 5

Vector receiver simulation results

In this chapter we discuss the obtained results for the simulation scenarios described in the previous chapter, but now with a vector receiver. Section 5.1 presents the obtained results with a vector receiver and the least-squares algorithm and compare them to the ones obtained with the scalar receiver and the same estimation algorithm. Section 5.2 presents the same results and analysis but for a vector receiver with the EKF algorithm. A full comparison of receiver algorithms and receiver architectures is presented at the end of this chapter, in section 5.3. The results displayed in this chapter were obtained with a VDLL. The block diagram of figure 5.1 summarizes the simulation configurations considered in this thesis.



(a) General configuration

(b) Configuration for signal analysis

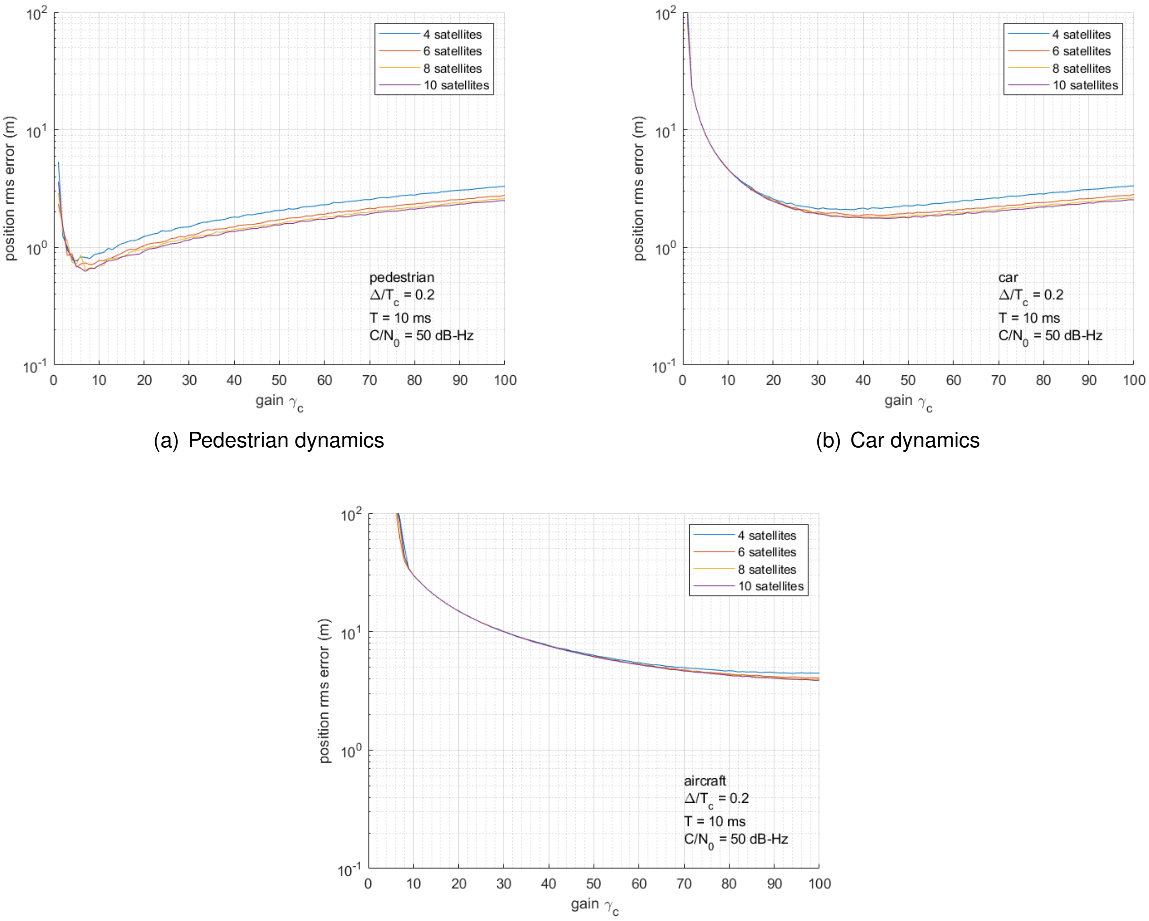
Figure 5.1: Set of simulation configurations.

5.1 Vector receiver with least-squares algorithm

5.1.1 Standard scenario simulation results

Herein we analyze the behavior of a vector receiver, undertaking the same type of simulations as with the scalar counterpart, in 4.1.1. In figure 5.2, the results for pedestrian, car and aircraft scenarios are shown with varying number of satellites.

Firstly, it is to be noted the range of γ_c in the vector receiver case: from 0 to 100; being needed higher values for the DLL feedback. The curves for the vector receiver exhibit a rather different behavior from the ones belonging to its scalar counterpart. The pedestrian and car scenarios present optimal γ_c



(a) Pedestrian dynamics

(b) Car dynamics

(c) Aircraft dynamics

Figure 5.2: Results for the vector receiver with varying number of used satellites (least-squares).

values for $\gamma_c \approx 7$ and $\gamma_c \approx 40$, respectively. On the other hand, the receiver for the aircraft scenario exhibits a constantly decreasing behavior relative to its rms error, with its optimum value of γ_c being the highest possible, for the range represented in figure 5.2. Analogously to the scalar receiver, increasing the number of satellites in use beyond 6 proves to be quite ineffective, as results barely improve.

For the pedestrian receiver, with 10 satellites in use, it is possible to achieve rms errors of approximately 0.6 m for $\gamma_c = 7$. Using only 4 satellites, the rms error can still be maintained below 1 m, something impossible for a scalar receiver. In the car scenario, again with 10 satellites, the rms error can be as low as almost 2 m, for values of γ_c between 30 and 50. If only 4 satellites are in use, one can achieve a minimum of rms error slightly higher than 2 m. Finally, in the aircraft scenario, with 10 satellites, an rms error of 4 m is obtained, for the highest value of gain simulated: $\gamma_c = 100$. If the constellation is reduced to 4 satellites, the error increases to around 4.5 m, for the same value of γ_c.

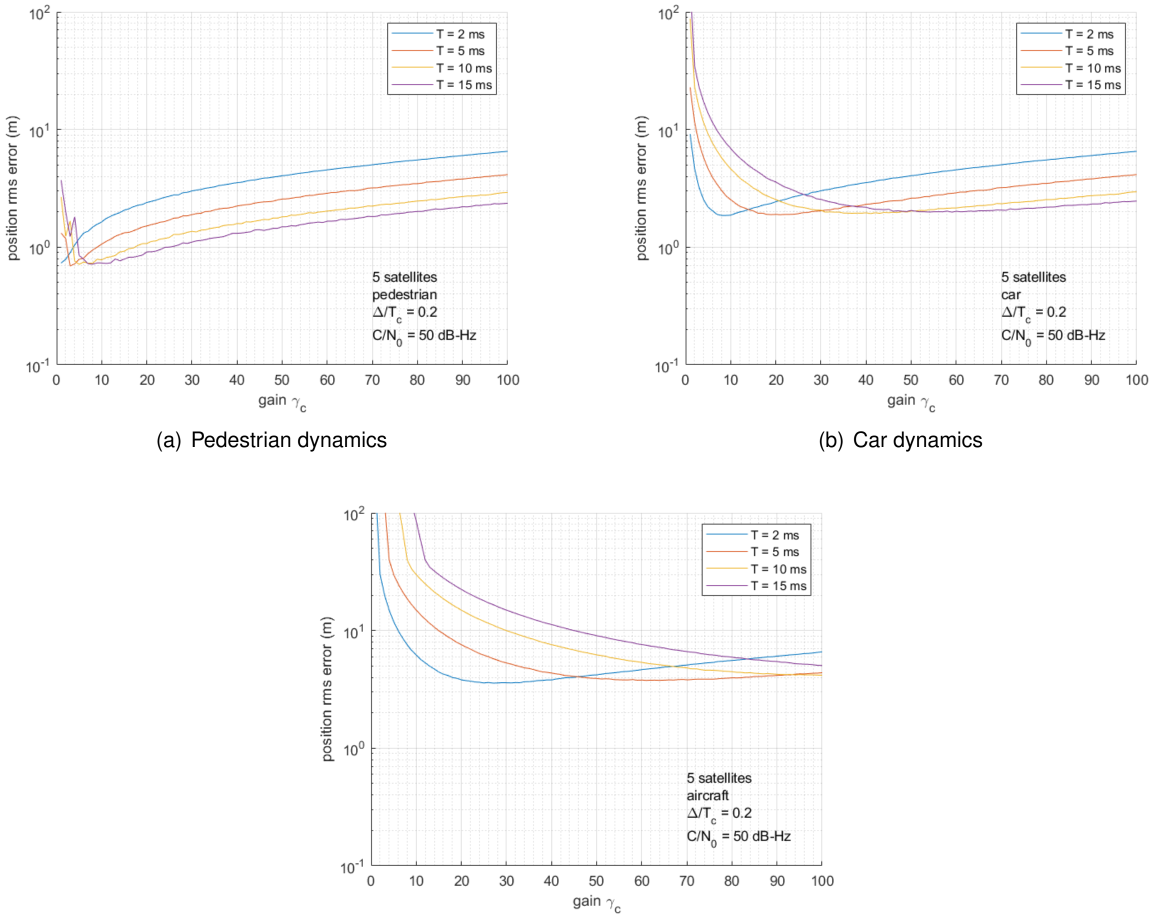
Comparing these results with the ones of the scalar receiver, the first thing that is clear is that a vector receiver needs to adjust its DLL feedback gain to its dynamics, in order to obtain the best performance. In table 5.1, a comparison of the best performance results obtained for both receivers, with 4 and 10 satellites, is shown.

We can conclude that results, for the vector receiver, with pedestrian and car dynamics, are clearly

	Minimum rms error (m)	
4 satellites	Scalar	Vector
Pedestrian	$\approx$ 4	$\approx$ 0.8
Car	$\approx$ 4	$\approx$ 2
Aircraft	4	$\approx$ 4.5
10 satellites	Scalar	Vector
Pedestrian	$\approx$ 3	$\approx$ 0.6
Car	$\approx$ 3	$\approx$ 1.8
Aircraft	$\approx$ 3.5	4

Table 5.1: Scalar vs Vector receiver performance with varying number of satellites (least-squares).

better than those of the scalar receiver, suffering little degradation when the number of satellites in use diminishes. In the aircraft case, however, the scalar receiver proves to be slightly better, with performance degradation being quite similar in both receivers. Despite this last small shortcoming, the vector receiver performs significantly better than the scalar variant, providing visible improvements in two out of the three dynamics tested. Next we analyze what happens when the correlation interval T is varied, as was done for the scalar receiver in figure 4.2.



(a) Pedestrian dynamics

(b) Car dynamics

(c) Aircraft dynamics

Figure 5.3: Results for the vector receiver with varying correlation interval T (least-squares).

All three receiver dynamics, for a certain range of γ_c produce better results for $T = 2$ ms, with the pedestrian receiver having the smallest range and the aircraft receiver the biggest. Increasing the value of γ_c, while maintaining $T = 2$ ms, will eventually stop producing the best possible performance results, inverting the order of the best value for T - for the highest γ_c, $T = 15$ ms produces the best results and $T = 2$ ms the worse (the aircraft case is an exception, where for the simulated γ_c range, $T = 10$ ms eventually produces the best results).

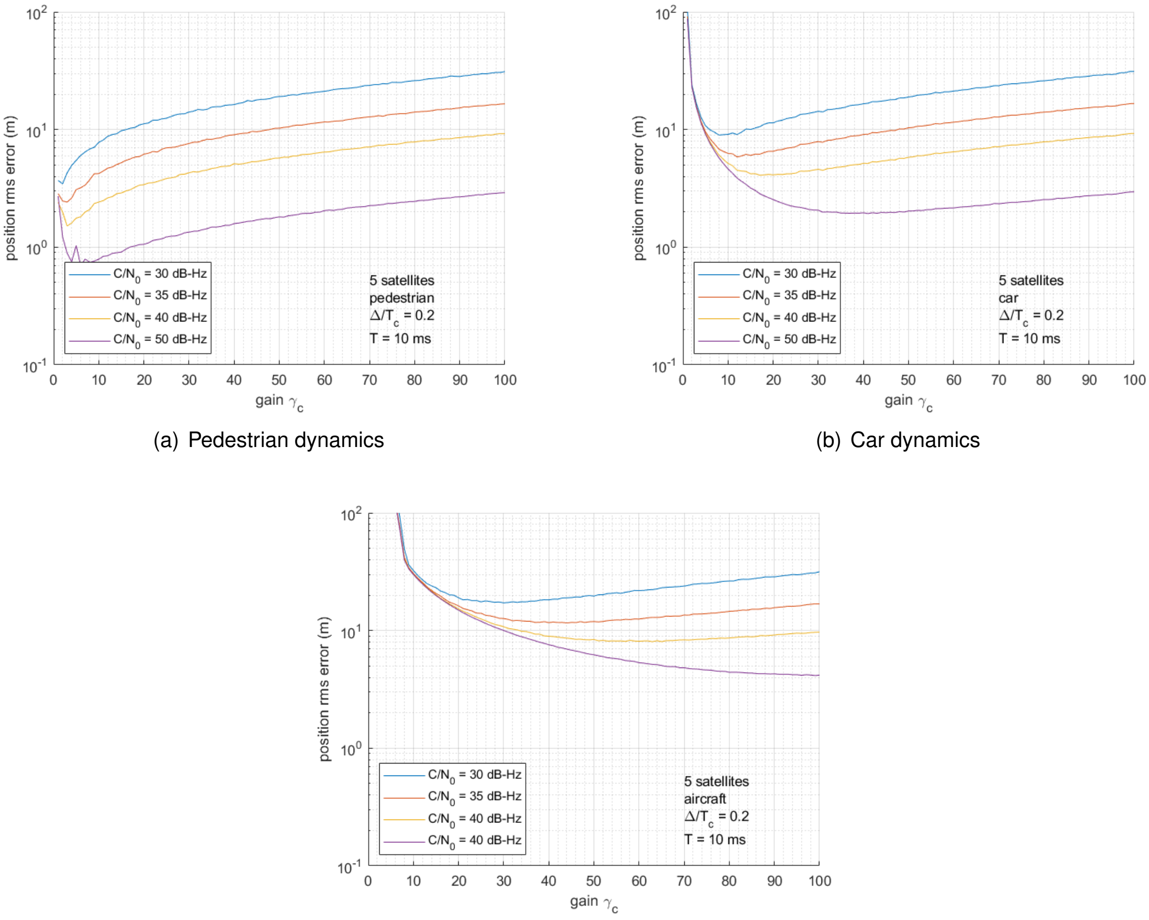
For the pedestrian receiver, the lowest rms error registered is of ≈ 0.7 m, achievable with all simulated values of T; for the car receiver, the lowest rms error obtained is around 2 m, once again with all represented values of T; and finally, for the aircraft case, the lowest rms error registered is of around 3.5 m, for $T = 2$ ms. This comes to show that the shortcoming presented in Table 5.1, where the vector receiver produced slightly worse results than its scalar counterpart, can be overcome, if one reduces the correlation interval in the DLL. More so, it is possible to obtain these results with a constellation with half the satellites, with a vector receiver, when compared to a scalar one (for aircraft dynamics, with 10 satellites and a scalar receiver, an rms error of ≈ 3.5 m was obtained, whereas with 5 satellites and a vector receiver, one could obtain a similar value for the rms error). This comes to show that besides γ_c gain tweaking, the vector receiver also benefits from adaptations of its DLL's correlation interval to the situation it is inserted in; where in the scalar case a ”set and forget” approach seems to be optimal. Finally, the results for varying the C/N$_0$ of all satellites in use, are presented in figure 5.4.

As with the scalar receiver, visible degradation in performance is registered as the C/N$_0$ decreases. This degradation is most dramatic when changing from 50 dB-Hz to 40 dB-Hz, with this gap in carrier-to-noise ratio being the most important, as it depicts passing from a very good scenario for signal reception quality to one much more plausible. As such, let us analyze, in table 5.2, the differences in minimum rms error registered for both receiver architectures, with the three receiver dynamics, and all values of C/N$_0$.

	Minimum rms error (m)			
Scalar	C/N$_0 = 50$ dB-Hz	C/N$_0 = 40$ dB-Hz	C/N$_0 = 35$ dB-Hz	C/N$_0 = 30$ dB-Hz
Pedestrian	$\approx$ 3.5	$\approx$ 9	$\approx$ 15	$\approx$ 25
Car	$\approx$ 3.5	$\approx$ 9	$\approx$ 15	$\approx$ 25
Aircraft	4	9	$\approx$ 15	$\approx$ 25
Vector	C/N$_0 = 50$ dB-Hz	C/N$_0 = 40$ dB-Hz	C/N$_0 = 35$ dB-Hz	C/N$_0 = 30$ dB-Hz
Pedestrian	$\approx$ 0.7	$\approx$ 1.5	$\approx$ 2.5	$\approx$ 3.5
Car	2	4	6	9
Aircraft	4	8	$\approx$ 10	$\approx$ 20

Table 5.2: Scalar vs Vector receiver performance with C/N$_0$ degradation (least-squares).

Comparing the results for both scenarios, it is evident the vector receiver presents dramatically better results in both the pedestrian and car cases, with much less performance degradation from applying a 10 dB attenuation to all satellites. In fact, for pedestrian dynamics, with C/N$_0 = 30$ dB-Hz, the vector receiver equals the performance of the scalar one, with C/N$_0 = 50$ dB-Hz. For aircraft dynamics, the



(a) Pedestrian dynamics

(b) Car dynamics

(c) Aircraft dynamics

Figure 5.4: Results for the vector receiver with varying C/N$_0$ in all satellites (least-squares).

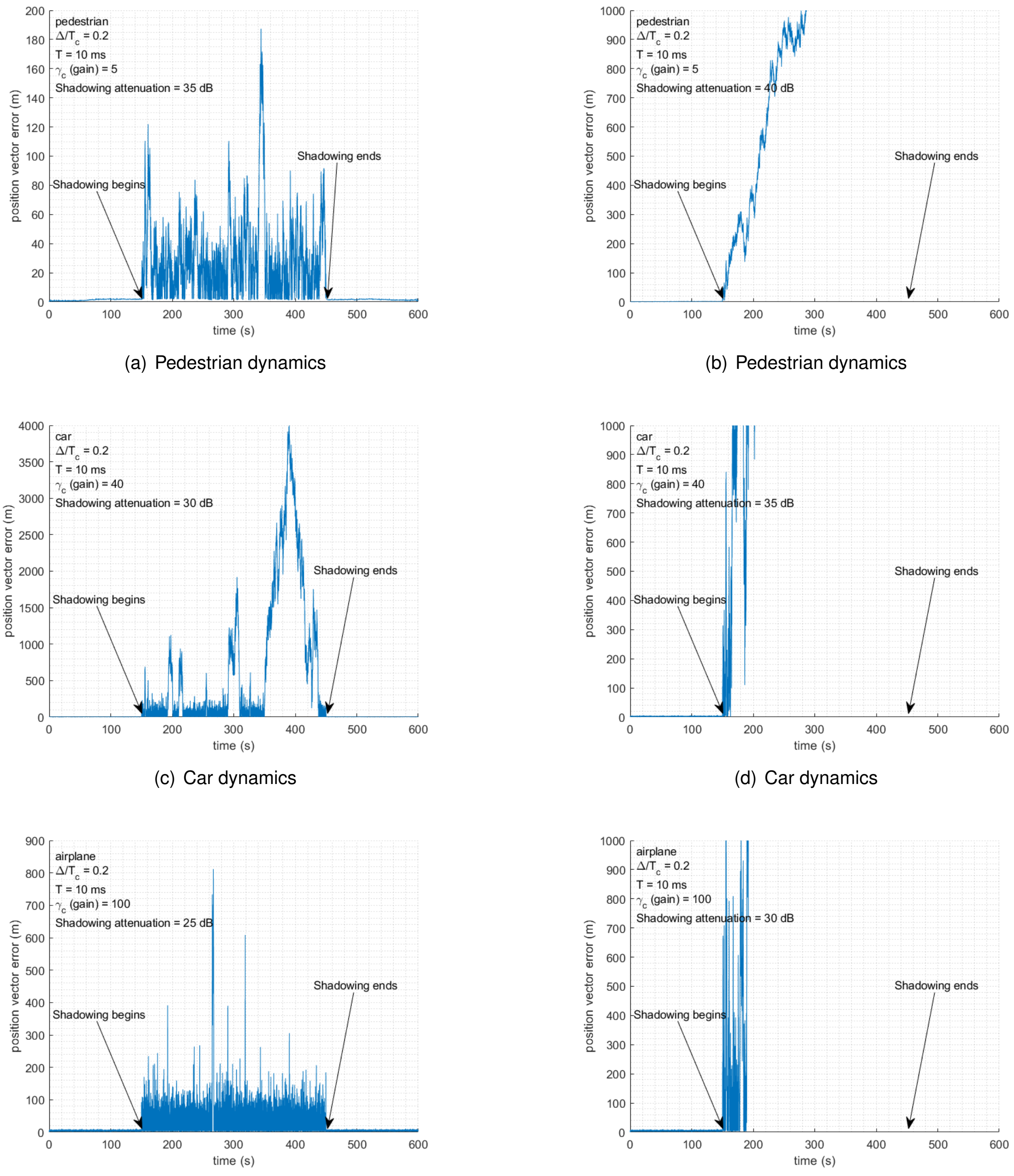
performance difference between these two types of receivers is more subtle, with the vector receiver suffering slightly less with the performance degradation.

With a scalar receiver one can only obtain average rms errors for C/N$_0$ $= 50$ dB-Hz, for all dynamics; whereas with a vector receiver, the rms error for a pedestrian receiver is very acceptable down to C/N$_0$ $= 30$ dB-Hz and for a car receiver down to C/N$_0$ $= 35$ dB-Hz. For the aircraft receiver, although the vector receiver presents better results, the only acceptable rms error occurs for C/N$_0$ $= 50$ dB-Hz for both receiver architectures. This comes to show that for aircraft dynamics, the vector receiver does not present significant improvements, considering the same estimation algorithm.

5.1.2 Shadowing scenario simulation results

The simulation presented in 4.1.2 is now repeated with a vector receiver. Once again, only the plot with the value of C/N$_0$ for the shadowed satellite that caused failure in tracking will be shown, alongside the C/N$_0$ tested immediately before. The results are presented in figure 5.5.

In table 5.3, we can see a comparison between the scalar and vector receivers' performance in shadowing scenarios. The vector receiver presents better performance across all three scenarios, withstanding more severe attenuation than its scalar counterpart. The differences between both receivers are most pronounced in the pedestrian case, where the vector receiver withstands an attenuation of 35 dB (15 dB more than a scalar receiver), and least pronounced in the aircraft case, with the vector receiver functioning correctly for an attenuation of 25 dB, while the scalar one can only operate properly with 20 dB of attenuation. Thus, we can conclude that, like in 5.1.1, the vector receiver outperforms the scalar one undoubtedly for pedestrian and car dynamics and is marginally better for aircraft dynamics. It is to be noted that the ”marginal” improvement in 5 dB for the tolerated attenuation translates into the vector receiver tolerating around 3 times less signal power than the scalar one.



(a) Pedestrian dynamics

(b) Pedestrian dynamics

(c) Car dynamics

(d) Car dynamics

(e) Aircraft dynamics

(f) Aircraft dynamics

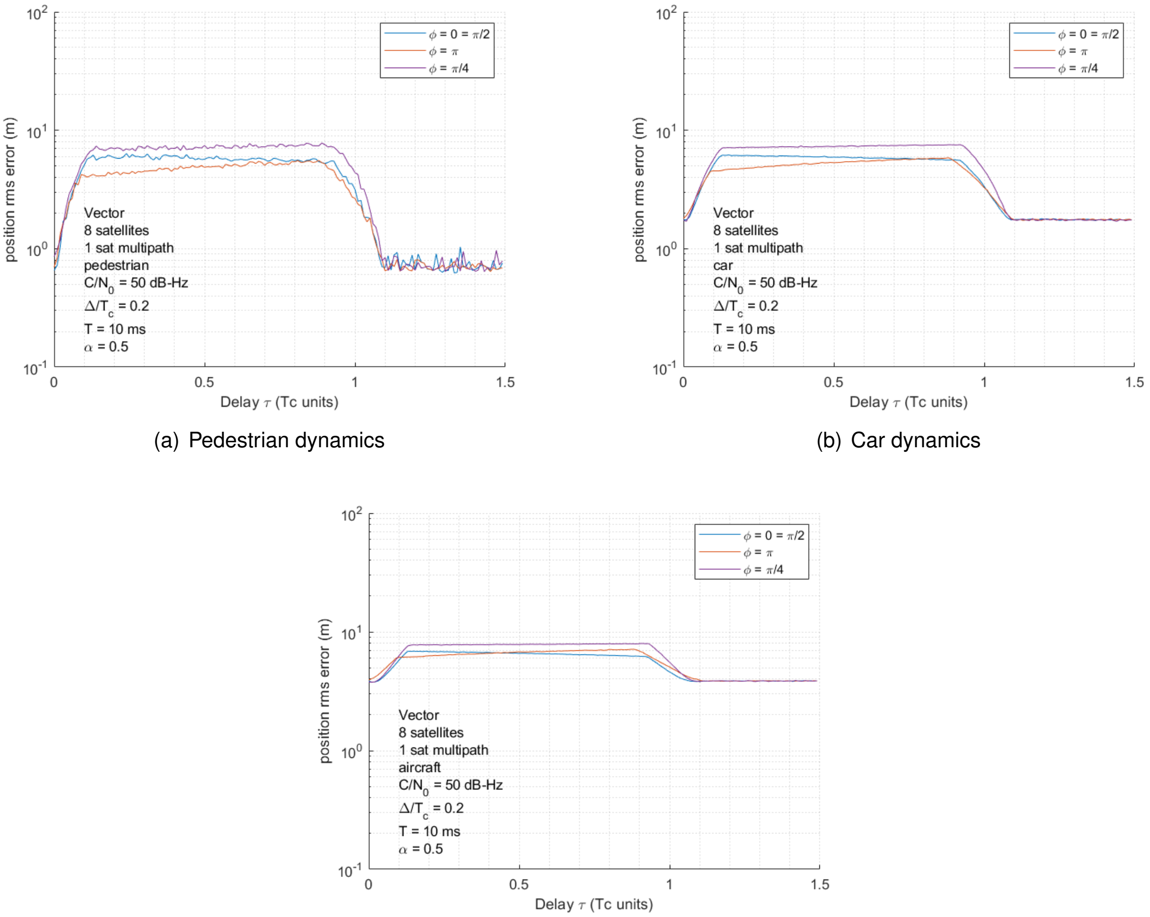
Figure 5.5: Shadowing scenario results for the vector receiver (least-squares).

	Max. Attenuation before loss of tracking (dB)	
	Scalar	Vector
Pedestrian	20	35
Car	20	30
Aircraft	20	25

Table 5.3: Scalar vs Vector receiver performance in a shadowing scenario (least-squares).

5.1.3 Multipath scenario simulation results

In figure 5.6 the results for the multipath scenario simulations, for a vector receiver with the least-squares algorithm, are displayed. In figure 5.6 we can see that, unlike in chapter 4, we now have displayed a fourth simulated value of $\phi = \pi/4$. This is due to the fact that, for a vector receiver, the $\phi = 0$ and $\phi = \pi/2$ curves overlap and the $\phi = \pi/4$ curve presents the poorest results obtained. The highest rms errors registered were of $8\,$m for $\phi = \pi/4$, for the three receiver dynamics. The $\phi = \pi$ curve presents the lowest errors, with exception of the aircraft receiver, where it surpasses the $\phi = 0 = \pi/2$ one. The lowest rms error peaks registered were of $5\,$m for $\phi = \pi$ (pedestrian and car cases) and for $\phi = 0 = \pi/2$ (aircraft case). No multipath effects are recorded after $1.1\,T_c$ units, as the three curves converge to



(a) Pedestrian dynamics

(b) Car dynamics

(c) Aircraft dynamics

Figure 5.6: Multipath scenario results for the scalar receiver (least-squares).

the normal rms error registered with this combination of used satellites and C/N$_0$. While in terms of

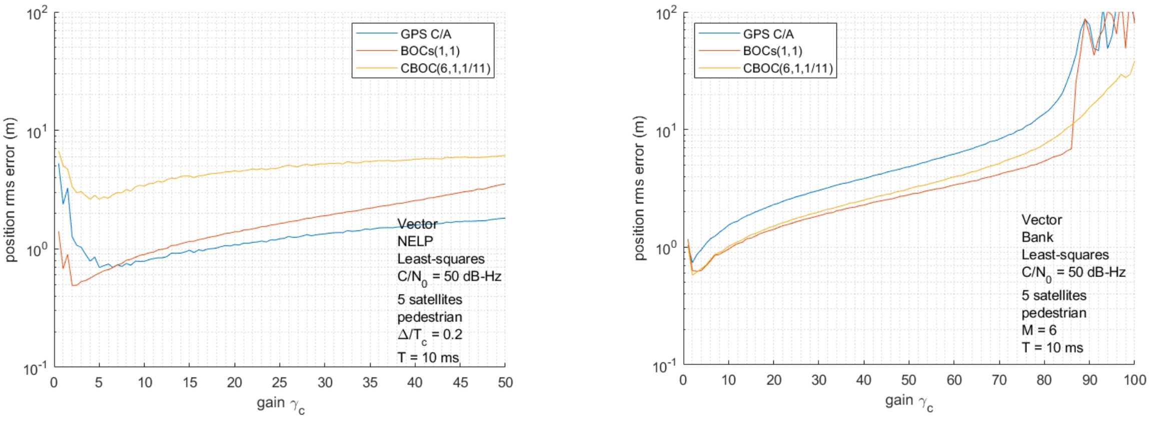
	Maximum rms error (m)	
1 satellite multipath	Scalar	Vector
Pedestrian	$\approx$ 12	8
Car	$\approx$ 12	8
Aircraft	$\approx$ 12	8

Table 5.4: Scalar vs Vector receiver performance in a multipath scenario (least-squares).

maximum rms error registered the vector receiver outperforms the scalar one, an analysis of figure 4.6, in chapter 4, shows that the scalar receiver, for $\phi = \pi/2$ we obtain a faster convergence to the rms error that corresponds to no multipath effects, something that does not happen with the vector receiver, with any of the displayed curves. However this is a very special case, as multipath is quite an unpredictable phenomenon and the vector receiver does present, in general terms, a lower maximum value for rms error.

5.1.4 Signal comparison simulation results

In figure 5.7 we can see the results from the signal comparison between GPS C/A, BOCs(1,1) and CBOC(6,1,1/11) pilot. The BOCs(1,1) presents the best results with both the NELP and bank of correlators, achieving minimum values of rms error of 0.5 m and 0.6 m respectively. Once again, the GPS C/A signal does not benefit from the bank of correlators, with its minimum rms error being 0.8 m, where with the NELP discriminator we have 0.7 m. The CBOC(6,1,1/11) pilot signal, on the other hand, benefits greatly from the bank of correlators, with the minimum rms error decreasing from ≈ 3 m to 0.6 m. Let us compare these results with the ones from the scalar receiver, in table 5.5.



(a) NELP discriminator

(b) Bank of correlators

Figure 5.7: Signal performance comparison for a vector receiver (least-squares).

As we can see from table 5.5, the results improve greatly by using a vector receiver regardless of using a NELP discriminator or a bank of correlators, with the vector architecture reducing the minimum rms errors in 1.5 m to 3.2 m. There is, therefore, great advantage in using a vector receiver regardless of

	Minimum rms error (m)	
NELP discriminator	Scalar	Vector
GPS C/A	$\approx$ 3.5	0.7
BOCs(1,1)	2	0.5
CBOC(6,1,1/11) pilot	$\approx$ 6	$\approx$ 3
Bank of correlators	Scalar	Vector
GPS C/A	4	0.8
BOCs(1,1)	$\approx$ 2.5	0.6
CBOC(6,1,1/11) pilot	$\approx$ 2.5	0.6

Table 5.5: Scalar vs Vector receiver performance for several modulations and code discriminators (least-squares).

modulation or code discriminator. It should also be noted the benefits of using a bank of correlators for reception of the CBOC(6,1,1/11) signal.

5.2 Vector receiver with extended Kalman filter algorithm

5.2.1 Standard scenario simulation results

Herein we analyze the results obtained in the standard simulation scenario, as defined in section 4.1.1. In figure 5.8 we can observe the results with a varying number of satellites. The plots obtained were identical for all three receiver dynamics, meaning we can disregard it when analyzing the obtained results. We can also immediately conclude that there is little or no improvement in performance when increasing the number of satellites from 4 to 10, with the best recorded values of rms error being 0.6 m and 0.5 m. These values were obtained with $\gamma_c = 100$. For $\gamma_c \leq 60$ it is very hard to discern between the four curves, with them being quite ”noisy”. Therefore, we can assume that the average performance is the same no matter the number of satellites utilized, with the separation occurring as γ_c is increased. In table 5.6 we can see the comparison between these results and the ones obtained for a scalar receiver

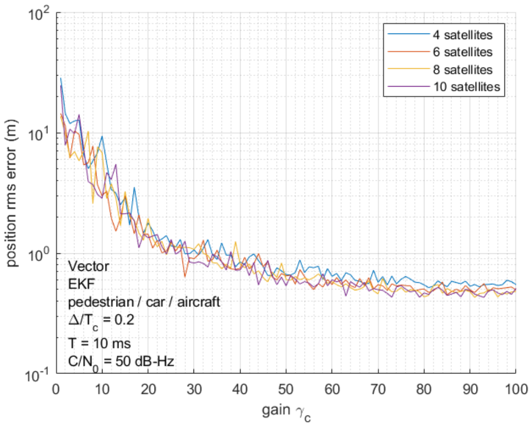


Figure 5.8: Results for the vector receiver with varying number of used satellites (EKF).

with the EKF algorithm (presented in 4.2.1). There is a considerable performance improvement when

	Minimum rms error (m)	
4 satellites	Scalar	Vector
Pedestrian	$\approx$ 1.5	$\approx$ 0.6
Car	$\approx$ 1.5	$\approx$ 0.6
Aircraft	$\approx$ 1.5	$\approx$ 0.6
10 satellites	Scalar	Vector
Pedestrian	1	$\approx$ 0.5
Car	1	$\approx$ 0.5
Aircraft	1	$\approx$ 0.5

Table 5.6: Scalar vs Vector receiver performance with varying number of used satellites (EKF).

utilizing a vector receiver architecture, with errors decreasing in around 1 m, for 4 satellites and 0.5 m for 10 satellites. A vector architecture therefore outperforms the traditional scalar architecture, especially in the case where a limited amount of satellites are present, which constitutes a difficult scenario of operation. Let us now analyze what happens when the correlation interval T is varied, in figure 5.9.

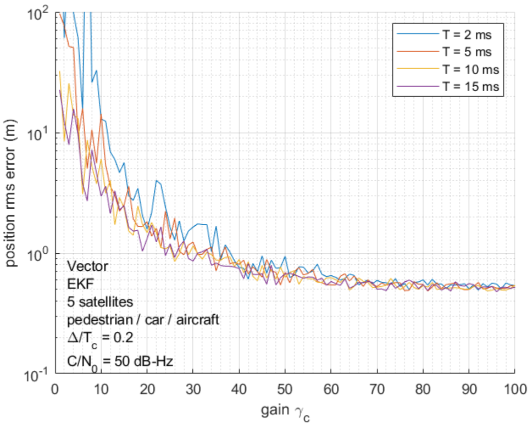


Figure 5.9: Results for the vector receiver with varying correlation interval T (EKF).

Once more, receiver dynamics have no weight in the obtained results, with the three dynamics producing the same results. We can see that changing the correlation interval produces no visible changes in the obtained results for $\gamma_c > 50$, with the curves overlapping and converging to 0.55 m in rms error. For $\gamma_c < 50$ we can see that the $T = 2$ ms produces slightly worse results than the other curves. However, all the curves are quite "noisy", with several rough oscillations, making it difficult to quantify how worse the $T = 2$ ms curve is.

The curves for varying values of C/N$_0$ are now presented in figure 5.10, where receiver dynamics, again, do not influence the obtained results. As expected, lowering the value of C/N$_0$ also lowers the receiver's performance, with the degradation being gradual, as we lower the C/N$_0$ value in use. Let us observe a comparison of these results with the ones obtained with a scalar receiver (in figure 4.10) in table 5.7.

The results comparison in table 5.7 shows without any doubt that there is a clear benefit in using a

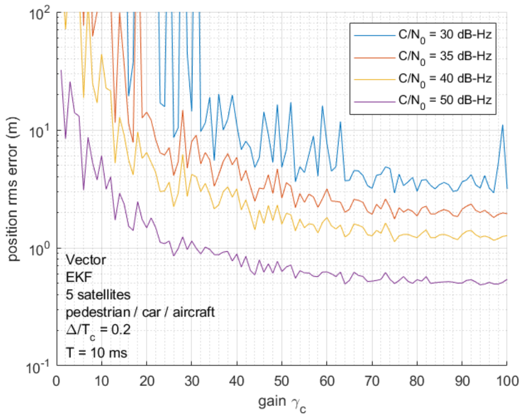


Figure 5.10: Results for the vector receiver with varying C/N$_0$ in all satellites (EKF).

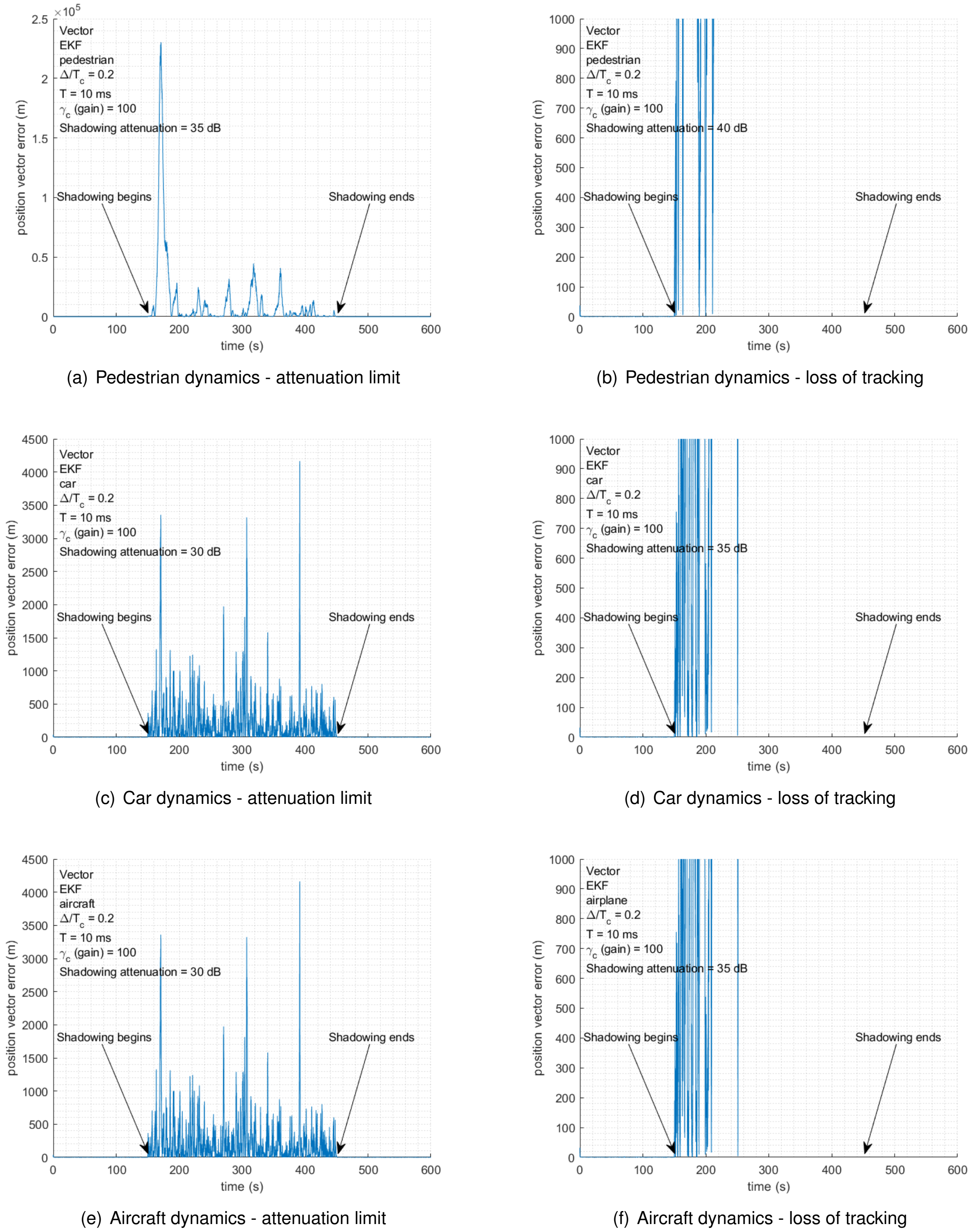
	Minimum rms error (m)			
Scalar	C/N$_0 = 50$ dB-Hz	C/N$_0 = 40$ dB-Hz	C/N$_0 = 35$ dB-Hz	C/N$_0 = 30$ dB-Hz
Pedestrian	$\approx$ 1	$\approx$ 4	7	$\approx$ 15
Car	$\approx$ 1	$\approx$ 4	7	$\approx$ 15
Aircraft	$\approx$ 1	$\approx$ 4	7	$\approx$ 15
Vector	C/N$_0 = 50$ dB-Hz	C/N$_0 = 40$ dB-Hz	C/N$_0 = 35$ dB-Hz	C/N$_0 = 30$ dB-Hz
Pedestrian	$\approx$ 0.5	$\approx$ 1	$\approx$ 2	$\approx$ 3
Car	$\approx$ 0.5	$\approx$ 1	$\approx$ 2	$\approx$ 3
Aircraft	$\approx$ 0.5	$\approx$ 1	$\approx$ 2	$\approx$ 3

Table 5.7: Scalar vs Vector receiver performance with C/N$_0$ degradation (EKF).

vector receiver architecture as the rms errors it presents are clearly lower than the ones exhibited by a traditional receiver. This is especially true for lower values of C/N$_0$ (with C/N$_0 = 30$ dB-Hz the vector receiver achieves an rms error of just 3 m), where the vector receiver still provides usable performance while the scalar one doesn't. For C/N$_0 = 50$ dB-Hz the vector receiver reduces the registered rms error by half, with this reduction becoming more pronounced as C/N$_0$ decreases, with C/N$_0 = 30$ dB-Hz registering a rms error reduction by a factor of 5. It should also be noted that the EKF algorithm produces better results than the least-squares (for the same receiver architecture) and ensures receiver dynamics do not influence the receiver's performance.

5.2.2 Shadowing scenario simulation results

The results for the shadowing scenario, described in section 4.1.2, with a vector receiver and the EKF algorithm are now displayed in figure 5.11. We can compare the obtained results with the ones from a scalar receiver (displayed in figure 4.11) in table 5.8. We can observe that the vector receiver is able to sustain higher levels of signal attenuation, for the shadowed satellite, than the scalar receiver. For the pedestrian case, the vector receiver bears an attenuation of 15 dB more than the scalar one, with that value being 10 dB and for the car and aircraft cases. These are considerable improvements over the traditional receiver as those values translate to $\times 31.6$ and $\times 10$ less signal power, in linear units.



(a) Pedestrian dynamics - attenuation limit

(b) Pedestrian dynamics - loss of tracking

(c) Car dynamics - attenuation limit

(d) Car dynamics - loss of tracking

(e) Aircraft dynamics - attenuation limit

(f) Aircraft dynamics - loss of tracking

Figure 5.11: Shadowing scenario results for the vector receiver (EKF).

Like with all the obtained results for the shadowing scenario, loss of lock can be said to occur before the 300 s mark. These results are very similar to the ones presented in table 5.3 (with exception to the aircraft case), suggesting that the estimation algorithm has little influence over receiver performance in a shadowing scenario, with the main differentiating factor being the receiver architecture.

	Max. Attenuation before loss of tracking (dB)	
	Scalar	Vector
Pedestrian	20	35
Car	20	30
Aircraft	20	30

Table 5.8: Scalar vs Vector receiver performance in a shadowing scenario (EKF).

5.2.3 Multipath scenario simulation results

The results obtained with the multipath scenario, for a vector receiver with the EKF, are displayed in figure 5.12. A quick analysis shows that there is no difference in results regarding the receiver dynamics. Like with the results presented in figure 5.6 (vector receiver with least-squares) the curve presenting the worse performance is the one that corresponds to having $\phi = \pi/4$ and the one presenting the best corresponds to having $\phi = \pi$, with the third curve standing in between these two. The $\phi = 0$ and $\phi = \pi/2$ overlap, as was the case in figure 5.6. The maximum rms error registered is of barely 8 m for $\tau = 0.92\,T_c$ units, obtained with the $\phi = \pi/4$ curve. The $\phi = \pi$ curve presents a peak of 5 m, corresponding to the lowest peak registered. The $\phi = \pi/4$ curve always produced the worst results and the $\phi = \pi$ curve produced the best ones. Let us compare these performance results with the ones obtained in figure 4.12 for a scalar receiver, in table 5.9.

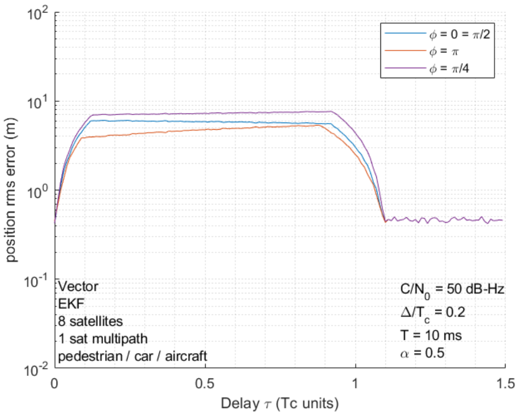


Figure 5.12: Multipath scenario results for the vector receiver (EKF).

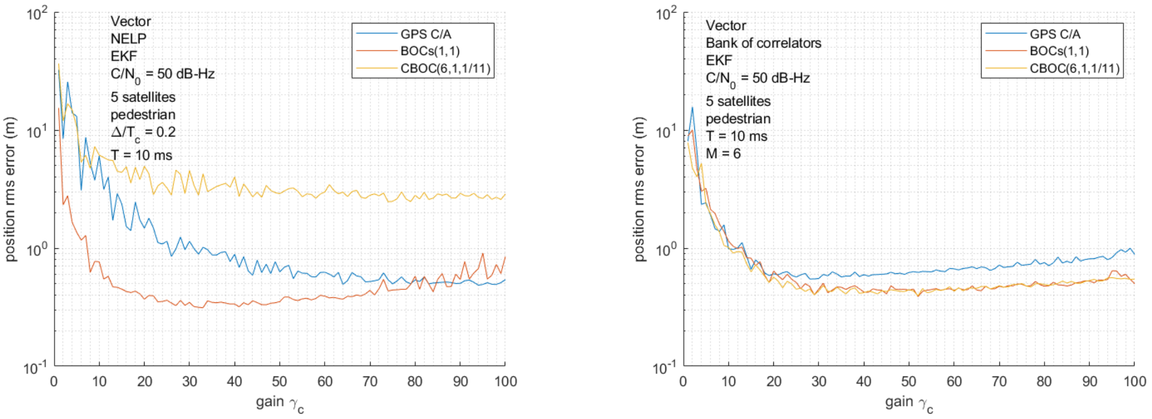
	Maximum rms error (m)	
1 satellite multipath	Scalar	Vector
Pedestrian	10	8
Car	10	8
Aircraft	10	8

Table 5.9: Scalar vs Vector receiver performance in a multipath scenario (EKF).

Regarding the maximum rms errors obtained, the vector receiver outperforms slightly the scalar one for the case where one satellite suffers from multipath. However, the maximum rms errors obtained imply poor performance from both receivers, with the EKF algorithm producing the same performance results as the least-squares (whose results are present in table 5.4), for the vector receiver. Still, the vector architecture outperforms the scalar one, with both estimation algorithms.

5.2.4 Signal comparison simulation results

Finally, we perform a signal performance comparison for the vector receiver with the EKF algorithm, using both the NELP and the bank of correlators as code discriminators. The results can be seen in figure 5.13. As has been the norm, the BOCs(1,1) presents the best performance for both code discriminators; the GPS C/A signal does not benefit from the usage of a bank of correlators and the CBOC(6,1,1/11) benefits greatly from it. The BOCs(1,1) presents minimum rms errors of 0.3 m and 0.4 m for the NELP and bank of correlators respectively; the GPS C/A signal presents 0.5 m and 0.6 m as minimum rms errors, for the same discriminators (in the same order) and the CBOC(6,1,1/11) signal presents 3 m and 0.4 m minimum errors, with the largest value being obtained with the NELP discriminator and the lowest with the bank of correlators. In table 5.10 a comparison with the scalar receiver (figure 4.13) is drawn.



(a) NELP discriminator

(b) Bank of correlators

Figure 5.13: Signal performance comparison for a scalar receiver (EKF).

	Minimum rms error (m)	
NELP discriminator	Scalar	Vector
GPS C/A	$\approx$ 1	0.5
BOCs(1,1)	$\approx$ 0.7	0.3
CBOC(6,1,1/11) pilot	$\approx$ 2.5	3
Bank of correlators	Scalar	Vector
GPS C/A	$\approx$ 1.5	0.6
BOCs(1,1)	$\approx$ 0.9	0.4
CBOC(6,1,1/11) pilot	$\approx$ 0.9	0.4

Table 5.10: Scalar vs Vector receiver performance for several modulations and code discriminators (EKF).

The table shows that, with exception of the CBOC(6,1,1/11) signal with a NELP discriminator, the receiver benefits greatly from having a vector architecture, with errors being reduced in up to 0.9 m. The CBOC(6,1,1/11) exception can be disregarded as it has been shown in the several signal performance comparisons, presented in this thesis, that it should only be utilized with a bank of correlators, despite the receiver architecture in use. Doing so, greatly increases the performance obtained with the CBOC(6,1,1/11) signal.

5.3 Overall comparison

In this section, we draw a global comparison between the scalar and vector receiver architectures, with both the least-squares and EKF estimation algorithms. This comparison is presented in tables 5.11 and 5.12, where a selection of the most important obtained results, in the simulated scenarios, are displayed. All results were obtained with 5 satellites (with exception the 4 sat and 10 sat columns, the shadowing column, where 4 satellites were used, and the multipath results, where 8 satellites were used) and C/N$_0$ $= 50$ dB-Hz (with exception of the C/N$_0$ $= 30$ dB-Hz and shadowing columns). Also, an early-late spacing of $\Delta/T_c = 0.2$ and integration time of $T = 10$ ms were considered.

An analysis on table 5.11, performed with BPSK modulation (GPS C/A), shows that the worse performance results were generally obtained with a scalar receiver with the least-squares algorithm and the best were obtained with a vector receiver with the EKF algorithm. Also, when considering the same estimation algorithm, the vector receiver almost always outperforms the scalar one. However, it is interesting to note that better performance can be achieved with a vector receiver with the least-squares algorithm, than with a scalar receiver with the EKF algorithm, for pedestrian dynamics in the standard scenario. For higher receiver dynamics, in the same scenario, the same does not happen, with the estimation algorithm (EKF vs least-squares) being more important than the receiver architecture (vector vs scalar). The performance difference between vector and scalar receivers is particularly visible in the standard simulation results for C/N$_0$ $= 30$ dB-Hz, where improvements in terms of rms error are quite visible. With exception of the aircraft vector receiver with the least-squares algorithm, all vector receiver results for C/N$_0$ $= 30$ dB-Hz are below 10 m, something unattainable with a scalar receiver. Moreover, for the vector receiver with the EKF, with that value of C/N$_0$, the registered rms error, for all dynamics, is of 3 m, which is a very acceptable value.

When analyzing the results from the shadowing scenario, the registered values are very much self-explanatory. There is great benefit form utilizing a vector receiver in such a scenario, with improvements, in tolerated attenuation, being as high as 15 dB (pedestrian dynamics). Also, these simulations demonstrated that estimation algorithms have little influence on the obtained performance, with the receiver architecture being the deciding factor.

For the multipath results, we can conclude that receiver architecture influences the obtained results and that the chosen estimation algorithm only affects the scalar receiver's performance. The best performance results are registered with a vector receiver, regardless of the adopted estimation algorithm (rms error of 8 m) and the worst with a scalar receiver with the least-squares algorithm (rms error of 12 m). It can also be concluded that for the simulated scenario, where one satellite suffers from multipath, a vector

	Minimum rms error in standard scenario (m)					
Scalar (least-squares)	4 sat	10 sat	50 dB-Hz	30 dB-Hz	Max. shadowing attenuation (dB)	Maximum rms error in multipath (m)
Pedestrian	$\approx$ 4	$\approx$ 3	$\approx$ 3.5	$\approx$ 25	20	$\approx$ 12
Car	$\approx$ 4	$\approx$ 3	$\approx$ 3.5	$\approx$ 25	20	$\approx$ 12
Aircraft	$\approx$ 4	$\approx$ 3.5	4	$\approx$ 25	20	$\approx$12
Scalar (EKF)	4 sat	10 sat	50 dB-Hz	30 dB-Hz	Max. shadowing attenuation (dB)	Maximum rms error in multipath (m)
Pedestrian	$\approx$ 1.5	1	$\approx$ 1	$\approx$ 15	20	10
Car	$\approx$ 1.5	1	$\approx$ 1	$\approx$ 15	20	10
Aircraft	$\approx$ 1.5	1	$\approx$ 1	$\approx$ 15	20	10
Vector (least-squares)	4 sat	10 sat	50 dB-Hz	30 dB-Hz	Max. shadowing attenuation (dB)	Maximum rms error in multipath (m)
Pedestrian	$\approx$ 0.8	$\approx$ 0.6	$\approx$ 0.7	$\approx$ 3.5	35	8
Car	$\approx$ 2	$\approx$ 1.8	2	9	30	8
Aircraft	$\approx$ 4.5	4	4	20	25	8
Vector (EKF)	4 sat	10 sat	50 dB-Hz	30 dB-Hz	Max. shadowing attenuation (dB)	Maximum rms error in multipath (m)
Pedestrian	$\approx$ 0.6	$\approx$ 0.5	$\approx$ 0.5	$\approx$ 3	35	8
Car	$\approx$ 0.6	$\approx$ 0.5	$\approx$ 0.5	$\approx$ 3	30	8
Aircraft	$\approx$ 0.6	$\approx$ 0.5	$\approx$ 0.5	$\approx$ 3	30	8

Table 5.11: Global Vector vs Scalar receiver architecture comparison, with the least-squares and EKF algorithms, using BPSK modulation (GPS C/A).

	Minimum rms error for several signals (m)	
Scalar (least-squares)	NELP	Bank
GPS C/A	≈ 3.5	4
BOCs(1,1)	2	≈ 2.5
CBOC(6,1,1/11)	≈ 6	≈ 2.5
Scalar (EKF)	NELP	Bank
GPS C/A	≈ 1	≈ 1.5
BOCs(1,1)	≈ 0.7	≈ 0.9
CBOC(6,1,1/11)	≈ 2.5	≈ 0.9
Vector (least-squares)	NELP	Bank
GPS C/A	0.7	0.8
BOCs(1,1)	0.5	0.6
CBOC(6,1,1/11)	≈ 2.5	0.6
Vector (EKF)	NELP	Bank
GPS C/A	0.5	0.6
BOCs(1,1)	0.3	0.4
CBOC(6,1,1/11)	3	0.4

Table 5.12: Global Vector vs Scalar receiver architecture comparison, with the least-squares and EKF algorithms, using different modulations.

receiver will always outperform a scalar receiver. Despite the fact all obtained results, for the multipath scenario, display a quite elevated maximum rms errors, an analysis of the shape of the curves should also be taken into consideration (figures 4.6, 4.12, 5.6 and 5.12), with the scalar receiver presenting a bigger range of registered rms errors. However, since multipath is a very unpredictable phenomenon, the peak rms error registered should always be considered as a term of comparison. Another important observation is how much multipath degrades the quality of signal reception, even in the case where only one satellite's signal suffers from that effect, within a considerable constellation of 8 satellites.

Moving on to table 5.12, which compares the performance obtained with different modulations, it can be immediately concluded that CBOC(6,1,1/11) benefits greatly from the bank of correlators and should only be used with it, and the GPS C/A signal does not benefit from using such code discriminator, with the NELP being the logical choice. For the signal comparison analysis it was shown that a vector receiver outperforms a scalar one regardless of estimation algorithm and code discriminator, where the only exception is the CBOC(6,1,1/11) signal used with a NELP discriminator. We can however disregard this exception due to the aforementioned reasons. It is interesting to observe that for a vector receiver and a bank of correlators (regardless of estimation algorithm) we obtain better performance from a GPS C/A signal than with the same signal for a scalar receiver and the optimal code discriminator for this signal: the NELP. This observation further attests to the benefits of using a vector architecture. It can be assumed that the same situation (vector receiver with sub-optimal discriminator outperforming a scalar receiver with the optimal one) does not happen for the CBOC(6,1,1/11) as it is a much more complex signal, demanding a more specific discriminator block.

Regarding the results obtained for the BOCs(1,1) signal, we observe that the performance registered with this signal increases if a vector architecture is utilized, instead of a scalar one. Contrarily to what was observed for the CBOC(6,1,1/11), the BOCs(1,1) signal presents better results when a NELP discriminator is being used. These results can be somewhat misleading, as using this modulation with the NELP discriminator can lead to false-code lock, as demonstrated in appendix C. As such, the bank of correlators presents itself as an alternative where that phenomenon does not occur and with very little difference in the minimum rms error values achieved, in comparison to the NELP. On a last note, it should be acknowledged that even if BOCs(1,1) and CBOC(6,1,1/11) present the same performance results with the bank of correlators (regardless of architecture and estimation algorithm), the latter presents more robustness to multipath as demonstrated through the multipath envelopes in figure 2.6.

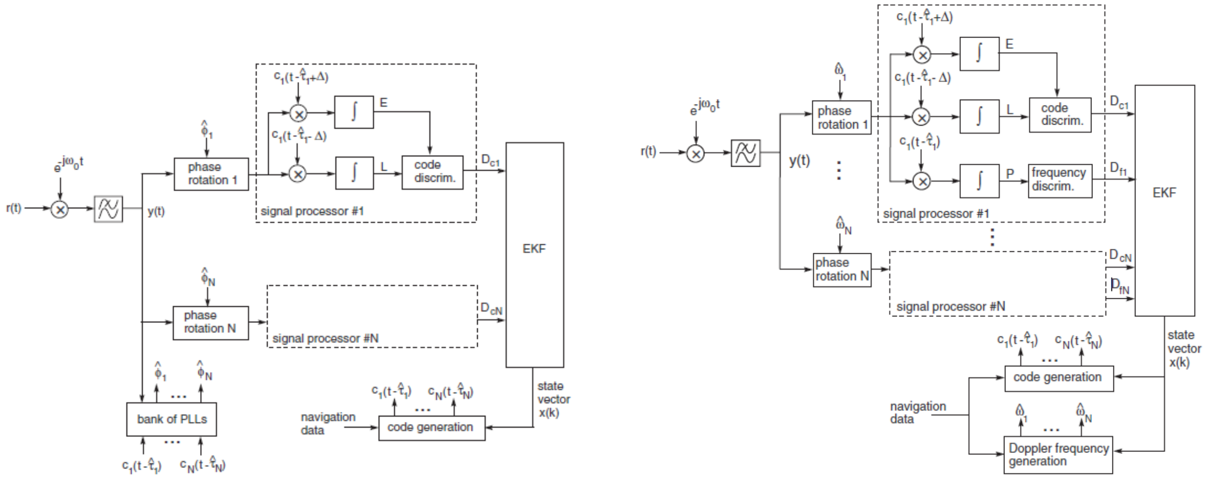
Chapter 6

Vector delay/frequency lock loop characterization and results

The results presented in chapter 5 were obtained with a VDLL and assumed that the carrier tracking section of the receiver operated without any flaws. In this his appendix we analyze the implementation of the VDFLL, a vector receiver architecture that includes both the code and carrier frequency tracking. The characterization and results presented in this appendix assume the usage of a BPSK signal (GPS C/A), a NELP code discriminator and a prompt Doppler frequency discriminator.

6.1 Code and carrier tracking

Figure 6.1 ilustrates the differences between the VDLL and VDFLL.



(a) Vector delay lock loop

(b) Vector delay/frequency lock loop

Figure 6.1: VDLL and VDFLL architectures [24].

Notice how on figure 6.1(a), representing the VDLL, only the code discriminators feed the EKF and there is feedback information only through the code generation block. Carrier tracking is being carried out by the bank of PLLs, separately from the feedback loop just described. This type of architecture - the VDLL - was the one considered to obtain the results in chapter 5 (considering perfect carrier tracking). In figure 6.1(b), the VDFLL, both the code and frequency discriminators form a single feedback loop with the EKF, existing feedback information through the code and Doppler frequency generation blocks. In figure 6.1(b) we consider that the receiver's local oscillator is a non-ideal oscillator subject to frequency variations relative to the nominal frequency $\omega_0 = 2\pi f_0$. This behavior is described in appendix D. We will now describe and analyze the VDFLL algorithm.

In order to simulate the VDFLL, we must also implement the carrier frequency tracking part of the receiver. As in chapter 2, signal $r(t)$, from equation (2.2), is heterodyned to baseband by multiplying it by the local oscillator output and lowpass filtering the result. Assume that the local oscillator (typically a crystal oscillator) is $\cos(\omega_c t + \theta(t))$, where $\theta(t)$ is an additional time-varying phase generated by the receiver's non-optimal clock. The instantaneous frequency deviation is

$$\omega_e(t) = \frac{d\theta(t)}{dt} \tag{6.1}$$

Assuming that $r(t)$ is expressed through its inphase and quadrature components and subject to a non-perfect Doppler wipe-off scheme for each satellite, we obtain a result analogous to equation (2.4), where ω_{d_n} is the true Doppler frequency (for satellite n), $\hat{\omega}_{d_n}$ is the estimate of ω_{d_n} and $\hat{\omega}_e$ is the estimate of the frequency deviation:

$$\begin{aligned}
\begin{bmatrix} \tilde{r}_{I_n}(t) \\ \tilde{r}_{Q_n}(t) \end{bmatrix} &= \begin{bmatrix} \cos\left[\left(\hat{\omega}_{d_n} - \hat{\omega}_e\right)t\right] & \sin\left[\left(\hat{\omega}_{d_n} - \hat{\omega}_e\right)t\right] \\ -\sin\left[\left(\hat{\omega}_{d_n} - \hat{\omega}_e\right)t\right] & \cos\left[\left(\hat{\omega}_{d_n} - \hat{\omega}_e\right)t\right] \end{bmatrix} \begin{bmatrix} r_I(t) \\ r_Q(t) \end{bmatrix} \\
&= A_n X_n(t) \begin{bmatrix} \cos\left[\left(\left(\omega_{d_n} - \omega_e\right) - \left(\hat{\omega}_{d_n} - \hat{\omega}_e\right)\right)t + \theta_n\right] \\ \sin\left[\left(\left(\omega_{d_n} - \omega_e\right) - \left(\hat{\omega}_{d_n} - \hat{\omega}_e\right)\right)t + \theta_n\right] \end{bmatrix} \\
&\quad + \begin{bmatrix} \sum_{i=1,\, i\neq n}^{N} A_i X_i(t) \cos\left[\left(\left(\omega_{d_i} - \omega_e\right) - \left(\hat{\omega}_{d_n} - \hat{\omega}_e\right)\right)t + \theta_i\right] \\ \sum_{i=1,\, i\neq n}^{N} A_i X_i(t) \sin\left[\left(\left(\omega_{d_i} - \omega_e\right) - \left(\hat{\omega}_{d_n} - \hat{\omega}_e\right)\right)t + \theta_i\right] \end{bmatrix} \\
&\quad + \begin{bmatrix} \tilde{n}_{I_n}(t) \\ \tilde{n}_{Q_n}(t) \end{bmatrix}
\end{aligned} \tag{6.2}$$

where both ω_{d_n} and ω_e are slowly-varying time functions.

Once again, we can disregard the inter-satellite interference term (the one with the sums), for simplicity's sake. The noise vector $\left[\tilde{n}_{I_n}(t)\, \tilde{n}_{Q_n}(t)\right]^T$ is statistically equal to the homonymous one presented in section 2.2, in chapter 2. If we correlate the obtained signals with an early, $X_n(t+\Delta/2-\epsilon_n)$, a prompt, $X_n(t-\epsilon_n)$, and a late, $X_n(t-\Delta/2-\epsilon_n)$, locally generated copies of the spreading signal, we will obtain six different resulting signals: an inphase and quadrature early signal, an inphase and quadrature prompt signal and an inphase and quadrature late signal. Let us analyze the early inphase signal:

$$IE_n = \frac{1}{T}\int_0^T \tilde{r}_{I_n}(t)\, X_n\left(t + \frac{\Delta}{2} - \epsilon_n\right)\, dt \tag{6.3}$$

In order to solve the integral in (6.3), we must divide the integration interval (correlation interval T) into

M sub-intervals of duration D, such that $T = MD$ and $\tilde{\omega}_n D \ll 1$, considering $\tilde{\omega}_n = (\omega_{d_n} - \omega_e) - (\hat{\omega}_{d_n} - \hat{\omega}_e)$ is the VDFLL frequency error in rad/s. Thus, we obtain:

$$IE_n = \frac{A_n}{M} R_X \left(\epsilon_n - \frac{\Delta}{2}\right) \cos\left(\frac{\tilde{\omega}_n T}{2} + \theta_n\right) \sum_{i=0}^{M-1} \exp\left(j\tilde{\omega}_n i D\right) + N_{IE_n} \tag{6.4}$$

which can be simplified into:

$$IE_n = A_n R_X \left(\epsilon_n - \frac{\Delta}{2}\right) \cos\left(\frac{\tilde{\omega}_n T}{2} + \theta_n\right) \text{sinc}\left(\tilde{f}_n T\right) + N_{IE_n} \tag{6.5}$$

The other resulting signals' expressions are:

$$\begin{aligned} IP_n =& A_n R_X \left(\epsilon_n\right) \cos\left(\frac{\tilde{\omega}_n T}{2} + \theta_n\right) \text{sinc}\left(\tilde{f}_n T\right) + N_{IP_n} \\ IL_n =& A_n R_X \left(\epsilon_n + \frac{\Delta}{2}\right) \cos\left(\frac{\tilde{\omega}_n T}{2} + \theta_n\right) \text{sinc}\left(\tilde{f}_n T\right) + N_{IL_n} \\ QE_n =& A_n R_X \left(\epsilon_n - \frac{\Delta}{2}\right) \sin\left(\frac{\tilde{\omega}_n T}{2} + \theta_n\right) \text{sinc}\left(\tilde{f}_n T\right) + N_{QE_n} \\ QP_n =& A_n R_X \left(\epsilon_n\right) \sin\left(\frac{\tilde{\omega}_n T}{2} + \theta_n\right) \text{sinc}\left(\tilde{f}_n T\right) + N_{QP_n} \\ QL_n =& A_n R_X \left(\epsilon_n + \frac{\Delta}{2}\right) \sin\left(\frac{\tilde{\omega}_n T}{2} + \theta_n\right) \text{sinc}\left(\tilde{f}_n T\right) + N_{QL_n} \end{aligned} \tag{6.6}$$

The noise vector $[N_{IE_n}\, N_{IP_n}\, N_{IL_n}\, N_{QE_n}\, N_{QP_n}\, N_{QL_n}]^T$ can be generated through the methods described in appendix B.

With the signals described in (6.5) and (6.6) we can define expressions for both the code and frequency discriminators. For the VDFLL we consider that the normalized code discriminator is described as in equation (2.15) and that we can define the normalized frequency discriminator function $D_f(\tilde{\omega}_n)$ as:

$$D_f(\tilde{\omega}_n) = \frac{IP_n(k-1)QP_n(k) - IP_n(k)QP_n(k-1)}{IP_n(k)^2 + QP_n(k)^2} \tag{6.7}$$

Notice that the frequency discriminator function depends on signals generated at two different instants [2]: k and $k{-}1$. Both the code and frequency discriminators share the same integration time of $T = 10\,\text{ms}$

6.2 VDFLL characterization

We can describe the VDFLL by expanding on the VDLL characterization presented in chapter 3, more specifically in subsection 3.2.3, as it was implemented using the EKF estimation algorithm. Since we now have N code discriminators and N frequency discriminators, equation (3.46) can be re-written as:

$$z_k - h\left[\hat{x}(k|k-1)\right] = \begin{bmatrix} \gamma_{c_1} & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & \gamma_{c_N} & & & \\ & & & \gamma_{f_1} & & \\ & \mathbf{0} & & & \ddots & \\ & & & & & \gamma_{f_N} \end{bmatrix} \cdot \begin{bmatrix} D_{c_1} \\ \vdots \\ D_{c_N} \\ D_{f_1} \\ \vdots \\ D_{f_N} \end{bmatrix} \tag{6.8}$$

where $[D_{c_1}, \cdots, D_{c_N}, D_{f_1}, \cdots, D_{f_N}]^T$ is the vector containing both the code and frequency discriminator outputs. The terms γ_{c_i} and γ_{f_i} represent the code and frequency discriminator gains so that [23]:

$$\begin{aligned} \gamma_{c_i} D_{c_i} &\approx c \cdot \epsilon_i \\ \gamma_{f_i} D_{f_i} &\approx \tilde{\omega}_i \end{aligned} \tag{6.9}$$

where c represents the speed of light in vacuum and ϵ_i and $\tilde{\omega}_i$ represent the code synchronization error and VDFLL frequency error of satellite i, respectively. Notice that the matrix containing the γ terms now has $(2N \times 2N)$ dimension, as opposed to the one presented in equation (3.46) which has $(N \times N)$ dimension.

Alterations must also be made to some parts of the EKF algorithm, namely the observations matrix H_k, presented in 3.2.1, and the entries of the observations noise covariance matrix $\widetilde{R}_k$ presented in 3.2.3.

Starting with the H_k matrix, we must expand the one presented in (3.43), in order to accommodate the terms resulting from a new type of observations: the i^{th} satellite's signal Doppler frequency. Let us express the observations matrix in the following way:

$$H_k = \left[\frac{\partial h_i[\hat{x}(k|k-1)]}{\partial x_j}\right]_{2N \times l} \tag{6.10}$$

with typically $l = 8$. Notice that now we have twice the number of rows of the matrix presented in (3.42). We can express the $h(x_k)$ matrix as [23], [24], [25]:

$$h(x_k) = \begin{bmatrix} U(k) \\ \Omega(k) \end{bmatrix}_{2N \times 1} \tag{6.11}$$

where $h(x_k)$ stands for $h\left[x(t_k)\right]$ with $t_k = kT$. We have that:

$$U(k) = \begin{bmatrix} \sqrt{(X_1 - x_u)^2 + (Y_1 - y_u)^2 + (Z_1 - z_u)^2} + u_u \\ \vdots \\ \sqrt{(X_N - x_u)^2 + (Y_N - y_u)^2 + (Z_N - z_u)^2} + u_u \end{bmatrix}_{N \times 1} \tag{6.12}$$

is a matrix containing the pseudoranges between the N satellites and the receiver and:

$$\begin{aligned}\Omega(k) &= -\,\frac{2\pi f_c}{c}\cdot\frac{d}{dt}U(k)\\ &= -\,\frac{2\pi f_c}{c}\begin{bmatrix}\frac{(X_1-x_u)}{d_1}\cdot\left(\dot{X}_1-\dot{x}_u\right)+\frac{(Y_1-x_u)}{d_1}\cdot\left(\dot{Y}_1-\dot{y}_u\right)+\frac{(Z_1-x_u)}{d_1}\cdot\left(\dot{Z}_1-\dot{z}_u\right)+\dot{u}_u\\ \vdots\\ \frac{(X_N-x_u)}{d_N}\cdot\left(\dot{X}_N-\dot{x}_u\right)+\frac{(Y_N-x_u)}{d_N}\cdot\left(\dot{Y}_N-\dot{y}_u\right)+\frac{(Z_N-x_u)}{d_N}\cdot\left(\dot{Z}_N-\dot{z}_u\right)+\dot{u}_u\end{bmatrix}_{N\times 1}\\ d_i &= \sqrt{(X_i-x_u)^2+(Y_i-y_u)^2+(Z_i-z_u)^2}\qquad i=1\,,\ldots,\,N\end{aligned}\tag{6.13}$$

is a matrix containing the Doppler frequencies associated with the transmitted signals from the N satellites, where f_c represents the signal's carrier frequency in Hz. The terms x_u, $\dot{x}_u$, y_u, $\dot{y}_u$, z_u and $\dot{z}_u$ represent the receiver's position and velocity, respectively, in each of the axis of the ECEF reference frame; whereas the $\dot{u}_u$ represents the receiver's clock drift which is the variation of the clock bias u_u.

Thus, we can write the new H_k matrix as:

$$H_k = -\begin{bmatrix} a_{x_1} & 0 & a_{y_1} & 0 & a_{z_1} & 0 & -1 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ a_{x_N} & 0 & a_{y_N} & 0 & a_{z_N} & 0 & -1 & 0\\ \xi_{x_1}\frac{2\pi f_c}{c} & a_{x_1}\frac{2\pi f_c}{c} & \xi_{y_1}\frac{2\pi f_c}{c} & a_{y_1}\frac{2\pi f_c}{c} & \xi_{z_1}\frac{2\pi f_c}{c} & a_{z_1}\frac{2\pi f_c}{c} & 0 & \frac{2\pi f_c}{c}\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \xi_{x_N}\frac{2\pi f_c}{c} & a_{x_N}\frac{2\pi f_c}{c} & \xi_{y_N}\frac{2\pi f_c}{c} & a_{y_N}\frac{2\pi f_c}{c} & \xi_{z_N}\frac{2\pi f_c}{c} & a_{z_N}\frac{2\pi f_c}{c} & 0 & \frac{2\pi f_c}{c}\end{bmatrix}\tag{6.14}$$

where a_{x_i}, a_{y_i} and a_{z_i} terms are defined in (3.44) and we can express the ξ_{x_i}, ξ_{y_i} and ξ_{z_i} terms as:

$$\begin{aligned}\xi_{x_i} &= \frac{1}{r_i}\left(\dot{X}_i-\hat{v}_{x_u}\right)-\frac{s_i}{r_i^3}\left(X_i-\hat{x}_u\right)\\ \xi_{y_i} &= \frac{1}{r_i}\left(\dot{Y}_i-\hat{v}_{y_u}\right)-\frac{s_i}{r_i^3}\left(Y_i-\hat{y}_u\right)\\ \xi_{z_i} &= \frac{1}{r_i}\left(\dot{Z}_i-\hat{v}_{z_u}\right)-\frac{s_i}{r_i^3}\left(Z_i-\hat{z}_u\right)\\ s_i &= (X_i-\hat{x}_u)\left(\dot{X}_i-\hat{v}_{x_u}\right)+(Y_i-\hat{y}_u)\left(\dot{Y}_i-\hat{v}_{y_u}\right)+(Z_i-\hat{z}_u)\left(\dot{Z}_i-\hat{v}_{z_u}\right)\\ r_i &= \sqrt{(X_i-\hat{x}_u)^2+(Y_i-\hat{y}_u)^2+(Z_i-\hat{z}_u)^2}\end{aligned}\tag{6.15}$$

where $\hat{v}_{x_u}$, $\hat{v}_{y_u}$ and $\hat{v}_{z_u}$ are the estimates of $\dot{x}_u$, $\dot{y}_u$ and $\dot{z}_u$ respectively. Also, we consider:

$$[\hat{x}_u\,\hat{v}_{x_u}\,\hat{y}_u\,\hat{v}_{y_u}\,\hat{z}_u\,\hat{v}_{z_u}]^T = [\hat{x}_1(k|k-1)\,\hat{x}_2(k|k-1)\,\hat{x}_3(k|k-1)\,\hat{x}_4(k|k-1)\,\hat{x}_5(k|k-1)\,\hat{x}_6(k|k-1)]^T\tag{6.16}$$

according to the EKF PVT model's state vector, presented in 3.2.

Regarding the $\widetilde{R}_k$ matrix, it is still diagonal but now of $(2N\times 2N)$ dimension, with its r_{ii} entries defined as [23]:

$$r_{ii} = \begin{cases} -\frac{c^2\Delta}{2p\left(\frac{C}{N_0}\right)_i T}, & i = 1, \ldots, N \\ \frac{1}{\left(\frac{C}{N_0}\right)_i T^3}, & i = N+1, \ldots, 2N \end{cases} \tag{6.17}$$

where c represents the speed of light in vacuum, T represents the VDFLL correlation interval, Δ represents the DLL early-late spacing and p can be described as in (3.50). In practical terms, when producing the results presented in 6.3, the entries of matrix $\widetilde{R}_k$, represented in (6.17), had to be manually adjusted, on a trial and error basis, in order to improve the obtained results. Also, the receiver's clock was now modeled as double integrated brownian motion.

6.3 VDFLL results

The results presented in this section for the VDFLL were obtained while simulating the vector receiver in the standard scenario described in chapter 4, considering car dynamics. All satellites shared the same code discriminator gain γ_c and frequency discriminator gain γ_f. Also, we considered a constellation of 8 satellites with C/N$_0$ $= 50$ dB-Hz and ω_e varying sinusoidaly between -11.59 and $+11.59$ rad/s. In figure 6.2, can observe a 3-D plot of the rms error (defined in chapter 4), as a function of both γ_c and γ_f.

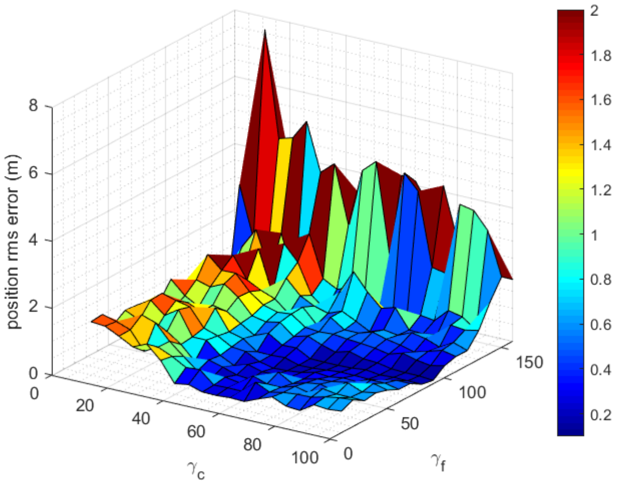


Figure 6.2: VDFLL rms error variation as a function of γ_c and γ_f.

The figure shows that the results are best in the $\gamma_c > 50$ and $\gamma_f < 130$ region, with performance worsening outside of those boundaries. Within that optimum region, the best results appear to be obtained with $80 \leq \gamma_c \leq 100$ and $\gamma_f = 100$, where the rms error reaches values as low as 0.2 m. We actually have that the optimal γ_f is $\gamma_f = 1/T$, since we have a correlation interval of $T = 0.01$ s, which mimics the VDFLL theoretical analysis described in [23], [24] and [25]. Let us analyze the way the position vector error (instantaneous error) evolves in time, for the VDFLL. As such, let us consider we have $\gamma_c = 100$ and $\gamma_f = 100$ as well and assume we have the same conditions as the ones used to obtain the results in figure 6.2. In figure 6.3 we can see how the VDFLL position vector error, described in equation (4.2), varies according to the value of C/N$_0$ shared among all used 8 satellites.

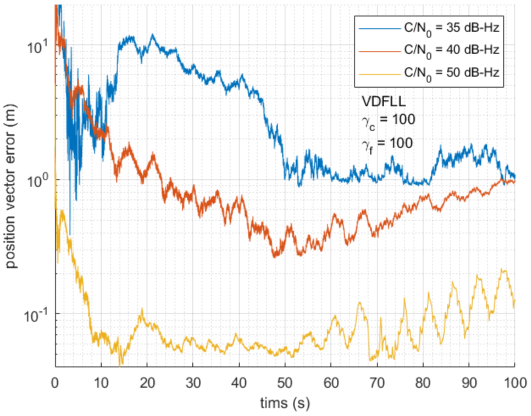


Figure 6.3: Time evolution of the VDFLL position vector errors with varying C/N_0.

For C/N$_0$ $= 50$ dB-Hz we register outstanding performance for the VDFLL, achieving errors below 0.01 m, between 10 s and 80 s. After 70 s, the position vector error exhibits a somewhat oscillatory behavior, reaching 0.2 m at the end of the simulation interval. It should be taken into account that the results regarding the curve just described, were obtained under extremely good reception conditions, where 8 satellites share a very high value for C/N$_0$, resulting in possibly unrealistic results.

When we consider C/N$_0$ $= 40$ dB-Hz, we obtain slightly worse results, while still being exceptional. Between 20 s and 95 s we observe position vector errors below 1 m. There is a constant accentuated decrease in error up until 50 s into the simulation, followed by a milder increase in it, until it converges to 1 m at 100 s. This curve demonstrates more plausible results, as the value considered for C/N$_0$ is much more realistic. As such, the performance results registered are very satisfactory, demonstrating that the VDFLL is a potent algorithm for navigation purposes.

At last, we simulate the VDFLL for C/N$_0$ $= 35$ dB-Hz. For this final curve, we notice that the position vector errors are somewhat noisy between 0 s and 20 s, with them quickly increasing to around 10 m and then decaying to about 1 m at 50 s and remaining there until the end of the simulation interval. This attests even further to how powerful the VDFLL algorithm can be, as lowering the signal-to-noise power ratio by 5 dB on all satellites - approximately having three times less signal power at reception, assuming noise power remains unaffected - practically has no effect on navigation performance, as after 100 s we achieve the same position vector error as the one obtained with C/N$_0$ $= 40$ dB-Hz.

On a final remark, implementing a functional VDFLL can be a challenge in itself as the algorithm's stability depends on various factors and can be quite elusive. Tuning the VDFLL to obtain the results presented in this chapter proved to be a difficult task, requiring several trial and error adjustments to not only the feedback gains γ_c and γ_f but also to the $\widetilde{R}_k$ matrix of the EKF used to close the feedback loop that characterizes vector receivers and the adopted clock dynamics. This was due to the fact all the aforementioned parameters now influence the whole functioning of the receiver, instead of their original blocks - i.e. the DLL feedback gain and the $\widetilde{R}_k$ matrix now influence the whole algorithm instead of just the code tracking portion or the EKF, respectively, for example.

Chapter 7

Conclusions

This chapter presents, in section 7.1, the conclusions drawn from the present thesis and, in section 7.2, guidelines for future work.

7.1 Achievements

The present thesis started, in chapter 2, with a brief enunciation of the satellite navigation principle, followed by a mathematical analysis of the code tracking operation and the effects of multipath on signal reception. The fundamental differences between a scalar and vector receiver were also laid out, alongside a comprehensive review on the published literature on the topic, highlighting the major achieved milestones and the most influential authors.

Two different estimation algorithms, the least-squares algorithm and the extended Kalman filter, were described in chapter 3. Their integration on scalar and vector receiver architectures was also addressed, pointing out the necessary modifications and adjustments for that integration.

In chapter 4, simulation results for the scalar receiver, with both the least-squares algorithm and the extended Kalman filter, were presented, together with the description of the scenarios in which to simulate the receivers' performance. Three different simulation scenarios were devised: a standard one, where signal reception would be near perfect, depending solely on the C/N_0 ratio or constellation properties such as the number of used satellites or the correlation intervals considered; a shadowing scenario, where one of the satellite's C/N_0 would be very poor compared to the other ones'; and a multipath scenario, where one of the used satellites would suffer the effects of multipath. Also, a performance comparison between the three different modulations was undertaken, considering two different code discriminators: the NELP and a bank of correlators (described in appendix C). The NELP produced better results for the GPS C/A signal and the BOCs(1,1) signal, whereas the bank of correlators outperformed the former for the CBOC(6,1,1/11) signal (used in the Galileo system).

It was shown in chapter 4 that the extended Kalman filter produced far better results than the least-squares algorithm in all tests, unsurprisingly.

In chapter 5 the simulation results for the vector receiver were presented, obtained in the same conditions as the ones in chapter 4. A global comparison between both the scalar and vector receiver (considering both estimation algorithms for both receiver architectures) was presented. Regarding the standard scenario, it was concluded that the vector receiver with the extended Kalman filter presented the best overall results, whereas the scalar receiver with the least-squares algorithm exhibited the poorest performance of the simulated receivers. The vector receiver with the least-squares algorithm managed to outperform the scalar one with the extended Kalman filter for low receiver dynamics, but was outperformed by the latter for more demanding dynamics. This analysis was true whether we considered a constellation of 4 or 10 satellites. Also, for low values of C/N_0, across all satellites in use, the vector receiver proved to be much more robust than the scalar receiver, with the obtained results implying the performance hierarchy mentioned previously.

In the shadowing scenario, performance could be analyzed regardless of the adopted estimation algorithm. It was shown that the vector receiver was able to sustain much more attenuation to one of the 4 satellites in use, in this simulation, before losing tracking abilities, than the scalar receiver.

For the multipath scenario the results demonstrated that the vector receiver was marginally better than the scalar one, albeit still producing poor positioning results. Utilizing an extended Kalman filter over a least-squares algorithm also slightly improved the results obtained with the scalar receiver, whereas the vector receiver appeared unaffected regarding the choice of estimation algorithm.

Regarding the signal comparison analysis, the vector receiver always produced better results than the scalar one, with the NELP discriminator being the best choice for the GPS C/A signal and the bank of correlators only being useful for the CBOC(6,1,1/11) signal, for which the NELP performs very poorly, leading to false-code locking. For the BOCs(1,1) signal, the NELP produced better performance results but could lead to false-code lock, whereas the bank of correlators would solve this issue, for this modulation.

In short, we can conclude from chapter 5 that a vector receiver, regardless of the adopted estimation algorithm, outperforms a scalar one, especially in situations of poor signal reception (low overall C/N_0 or shadowing), with exception of the case described a few paragraphs earlier, in the standard scenario.

In chapter 6 a thorough description of the VDFLL was presented, analyzing both code and frequency tracking operations and the alterations to be made to the extended Kalman filter. Some performance results were also presented, regarding the VDFLL's dependence on the feedback gains of the code and frequency tracking loops and the evolution in time of the positioning error of the algorithm. Optimal feedback gains were identified and the algorithm's performance under different values of C/N_0 was also analyzed. The implementation of the VDFLL proved to be a considerable achievement as this has revealed to be a difficult task, with the algorithm being very hard to tune.

7.2 Future Work

There are several paths to improving the work presented in this thesis. Some of the most relevant ones are now highlighted:

- **Implementation of the acquisition block**: The inclusion of the acquisition block, considered to function perfectly in this thesis, would allow for more realistic simulation results and better modeling of a real GNSS receiver.

- **Further research on the VDFLL**: The VDFLL presented in chapter 6 was implemented considering previous literature but some parameters had to be adjusted manually under a trial and error basis, as there was either no formal description of them or the existing one was unusable. This comes to show that a lot of ground still needs to be threaded on the topic of the VDFLL. Also, more and more varied simulations of the algorithm will most certainly be needed.

- **Devising new simulation scenarios**: Keeping in line with the last point, the creation of new simulation scenarios is paramount to obtaining a more incisive view on the performance of vector architectures for GNSS receivers. Among these scenarios we emphasize the reception of signals in the presence of ionospheric scintillation which occurs in polar and equatorial regions and may prevent the operation of conventional GNSS receivers [40] Another important topic is the performance assessment of vector versus scalar architectures in presence of jamming which is the intentional generation of interference signals with the purpose of degrading or denying the operation of the GNSS receiver [41].

- **Further simulations with the Galileo CBOC(6,1,1/11) signal**: With the dawn of the Galileo system, it is important to gather as much information as possible on the behavior of the CBOC(6,1,1/11) signal, as it is to be the one most widely used, due to its open-access nature. A similar modulation, called Time-Multiplexed BOC (TMBOC), is also being deployed in the new GPS satellites.

- **Development of a target s-curve choice method for the bank of correlators**: Developing a scientific method to choosing an optimal s-curve for the bank of correlators (described in appendix C) would be a very interesting and relevant topic of research. It would allow for optimization of the bank of correlators for reception of GNSS signals (both GPS and Galileo) in several reception conditions.

Bibliography

[1] M. Lashley. *Modeling and Performance Analysis of GPS Vector Tracking Algorithms*. PhD thesis, Graduate Faculty, Auburn University, Alabama, 2009.

[2] J. W. Betz. *Engineering Satellite-Based Navigation and Timing. Global Navigation Satellite Systems, Signals and Receivers*. John Wiley and Sons, INC., Hoboken, New York, First edition, 2016. ISBN:978-1-118-61597-3.

[3] S. Bhattacharyya. *Performance Integrity Analysis of the Vector Tracking Architecture of GNSS Receivers*. PhD thesis, Graduate School, University of Minnesota, Minnesota, 2012.

[4] P. Misra and P. Enge. *Global Positioning System - Signals, Measurement and Performance*. Ganga-Jamuna Press, Lincoln, Massachusetts, Second edition, 2004. ISBN:0-9709544-1-7.

[5] M. G. Petovello and G. Lachapelle. Comparison of Vector-Based Software Receiver Implementations With Application to Ultra-Tight GPS/INS Integration. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, pages 1790 – 1799, September 2006. Fort Worth, TX.

[6] T. Pany and B. Eissfeller. Use of a Vector Delay Lock Loop Receiver for GNSS Signal Power Analysis in Bad Signal Conditions. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, pages 893–903, April 2006. Coronado, CA.

[7] D. Benson. Interference Benefits of a Vector Delay Lock Loop (VDLL) GPS Receiver. In *Proceedings of the 63rd Annual Technical Meeting of The Institute of Navigation (2007)*, pages 749 – 756, April 2007. Cambridge, MA.

[8] F. D. Nunes. Scalar and vector architecture analysis. Technical report, Instituto Superior Técnico, Lisboa, 2018.

[9] S. Jin. *Global Navigation Satellite Systems - Signal, Theory and Applications*. Intech, Rijeka, Croatia, First edition, 2012. ISBN:978-953-307-843-4.

[10] F. D. Nunes. NELP and HRC discriminators. Technical report, Instituto Superior Técnico, Lisboa, 2015.

[11] M. Lashley and D. Bevly. GNSS Solutions: what are vector tracking loops, and what are their benefits and drawbacks? *Inside GNSS*, 4(3):16 – 21, May/June 2009.

[12] B. W. Parkinson and J. J. Spilker. *Global Positioning System: Theory and Applications Volume 1*. American Institute of Aeronautics and Astronautics Inc., First edition, 1996. ISBN:1-56347-106-X.

[13] E. M. Copps, G. J. Geier, W. C. Fidler, and P. A. Grundy. Optimal Processing of GPS Signals. *NAVIGATION: Journal of the Institute of Navigation*, 27(3):171 – 182, Fall 1980.

[14] J. W. Sennot and D. Senffner. The Use of Satellite Geometry for Prevention of Cycle Slips in a GPS Processor. *NAVIGATION: Journal of the Institute of Navigation*, 39(2):217 – 236, Summer 1992.

[15] J. W. Sennot and D. Senffner. Comparison of Continuity and Integrity Characteristics for Integrated and Decoupled Demodulation/Navigation Receivers. In *Proceedings of the 8th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1995)*, pages 1531–1537, September 1995. Palm Springs, CA.

[16] J. W. Sennot and D. Senffner. A GPS Carrier Phase Processor for Real-Time High Dynamics Tracking. In *Proceedings of the 53rd Annual Meeting of The Institute of Navigation (1997)*, pages 299–308, June 1997. Albuquerque, NM.

[17] M. Zhodzishsky, S. Yudanov, V. Veitsel, and J. Ashjaee. Co-Op Tracking for Carrier Phase. In *Proceedings of the 11th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1998)*, pages 653 – 664, September 1998. Nashville, TN.

[18] J.-H. Won, D. Dotterbock, and B. Eissfeller. Performance Comparison of Different Forms of Kalman Filter Approaches for a Vector-Based GNSS Signal Tracking Loop. *NAVIGATION: Journal of the Institute of Navigation*, 57(3):185 – 199, Fall 2010.

[19] J. H. Won, B. Eissfeller, and T. Pany. Implementation, Test and Validation of a Vector-Tracking-Loop with the ipex Software Receiver. In *Proceedings of the 24th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2011)*, pages 795 – 802, September 2011. Portland, OR.

[20] S. Bhattacharyya and D. Gebre-Egziabher. Development and Validation of Parametric Models for Vector Tracking Loops. *NAVIGATION: Journal of the Institute of Navigation*, 57(4):275 – 295, Winter 2010-2011.

[21] S. Zhao and D.Akos. An Open Source GPS/GNSS Vector Tracking Loop - Implementation, Filter Tuning and Results. In *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*, pages 1293 – 1305, January 2011. San Diego, CA.

[22] J. J. Brewer. *The Differential Vector Phase-Locked Loop for Global Navigation Satellite System Signal Tracking*. PhD thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2014.

[23] F. M. G. Sousa and F. D. Nunes. Characterization and Performance Analysis of a VDFLL GNSS Receiver Architecture. In *NAVITEC 2012: 6th ESA Workshop on Satellite Navigation Technologies*

and European Workshop on GNSS Signals and Signal Processing, December 2012. Noordwijk, Netherlands.

[24] F. M. G. Sousa and F. D. Nunes. Performance comparison of a VDFLL versus VDLL and Scalar GNSS Receiver Architectures in Harsh Scenarios. In *NAVITEC 2014: 7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing*, December 2014. Noordwijk, Netherlands.

[25] F. M. G. Sousa and F. D. Nunes. Performance of a VDFLL GNSS Receiver Architectures in Presence of Interference. In *NAVITEC 2016: 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing*, December 2016. Noordwijk, Netherlands.

[26] M. Lashley and D. M. Bevly. Vector Delay/Frequency Lock Loop Implementation and Analysis. In *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, pages 1073 – 1086, January 2009. Anaheim, CA.

[27] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *ASME - Journal of Basic Engineering*, 82(Series D):35–45, March 1960.

[28] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, First edition, 1974. ISBN:0-262-57048-3.

[29] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, INC, First edition, 1970.

[30] B. Anderson and J. Moore. *Optimal Filtering*. Prentice-Hall, First edition, 1979. ISBN:0-262-57048-3.

[31] F. D. Nunes. Air traffic control systems. Lecture notes, Instituto Superior Técnico, Lisboa, 2017.

[32] E. A. Wan and R. van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *Proceedings of Adaptative Systems for Signal Processing, Communications and Control Symposium*, pages 153–158. IEEE, October 2000.

[33] D. Simon. *Optimal State Estimation*. John Wiley and Sons, INC., Hoboken, New York, First edition, 2006. ISBN-13: 978-0-471-70858-2.

[34] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, Third edition, 1997. ISBN:0-471-12839-2.

[35] F. L. Walls and D. W. Allan. Measurements of Frequency Stability. In *Proceedings of the IEEE Special Issue*, volume 4, pages 162 – 168. IEEE, January 1986.

[36] E. D. Kaplan and C. J. Hegarty. *Understanding GPS. Principles and Applications*. Artech House, INC, Second edition, 2006. ISBN:1-58053-894-0.

[37] SEM Almanacs. URL `https://www.celestrak.com`, . Accessed: 2019-02-19.

[38] Shadow Matching: Improved GNSS Accuracy in Urban Canyons. URL `https://www.gpsworld.com/wirelesspersonal-navigationshadow-matching-12550/`, . Accessed: 2019-07-17.

[39] P. Fine and W. Wilson. Tracking Algorithm for GPS Offset Carrier Signals. In *Proceedings of the 1999 National Technical Meeting of The Institute of Navigation*, pages 671 – 676, January 1999. San Diego, CA.

[40] R. Conker, M. El-Arini, C. Hegarty, and T. Tsiao. Modelling the Effects of Ionospheric Scintillation on GPS/Satellite-Based Augmentation System Availability. *Rado Science*, 38(1):1–23, January 2003.

[41] F. Dovis, editor. *GNSS Interference Threats and Countermeasures*. Artech House, Boston, First edition, 2015. ISBN-13: 978-1-60807-810-3.

[42] *European GNSS (Galileo) open service Signal In Space Interface Control Document*. European Global Navigation Satellite Systems Agency, December 2016.

[43] *BeiDou Navigation Satellite System Signal In Space Interface Control Document*. China Satellite Navigation Office, August 2017.

[44] F. M. G. Sousa and F. D. Nunes. New Expressions for the Autocorrelation Function of BOC GNSS Signals. *NAVIGATION: Journal of the Institute of Navigation*, 60(1):1 – 9, Spring 2013.

[45] F. D. Nunes. Performance of a normalized code discriminator based on a bank of correlators. Technical report, Instituto Superior Técnico, Lisboa, 2019.

[46] C. W. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall, First edition, 1992. ISBN:0-13-852112-3.

[47] T. Pany, M. Irsigler, and B. Eissfeller. S-Curve Shaping: A New Method for Optimum Discriminator Based Code Multipath Mitigation. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, pages 2139 – 2154, September 2005. Long Beach, CA.

[48] H. Meyr and G. Ascheid. *Synchronization in Digital Communications. Phase- Frequency-Locked Loops, and Amplitude Control*. Wiley, New York, First edition, 1990. ISBN:0-471-50193-X.

Appendix A

GNSS signals characterization

In this appendix we will present expressions for the autocorrelation functions of the GNSS signals used in this thesis, alongside their respective power spectral densities.

A.1 BPSK signal

Assuming a truly random code sequence ± 1V, the autocorrelation function for a binary phase-shift keying (BPSK) signal is described as a triangle function, centered at $\tau = 0$. The C/A signal belonging to the GPS is well approximated by the following triangular autocorrelation.

$$R_{BPSK}(\tau) = \Lambda_{T_c}(\tau) \tag{A.1}$$

where the triangle function is expressed as the following branch function:

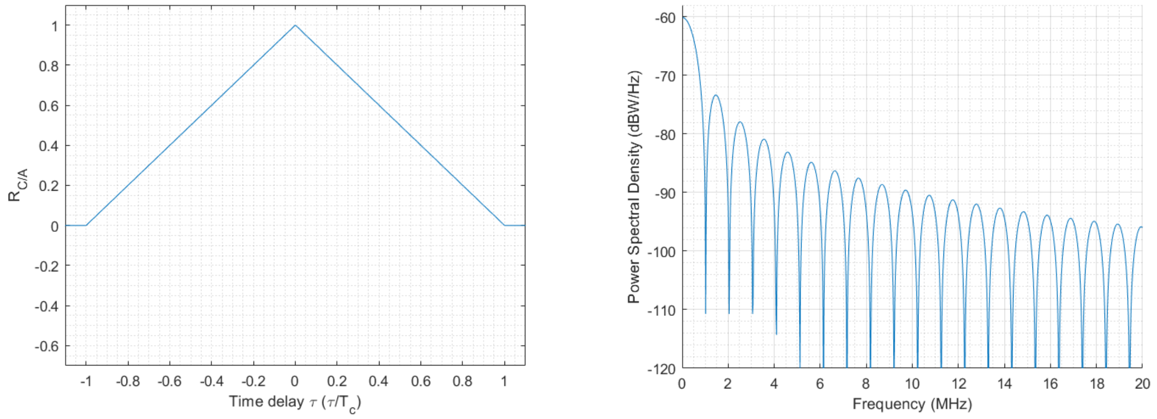
$$\Lambda_{T_c}(\tau) = \begin{cases} 1 - \frac{|\tau|}{T_c}, & |\tau| < T_c \\ 0, & \text{otherwise} \end{cases} \tag{A.2}$$

with the index of Λ indicating half duration of the triangle and $T_c = 10^{-3}/1023\,$s representing the C/A code's chip time in seconds. A representation of this autocorrelation function can be found in figure A.1.

The power spectral density (PSD) of the GPS C/A signal, also depicted in figure A.1, is expressed as follows:

$$G_{BPSK}(f) = T_c\,\text{sinc}^2(f\,T_c) \tag{A.3}$$

where f represents the frequency in Hz and sinc$(x) \equiv \sin(\pi x)/(\pi x)$.



(a) Autocorrelation function of a GPS C/A signal.

(b) Power spectral density of a GPS C/A signal.

Figure A.1: GPS C/A (BPSK) signal characterization.

A.2 BOCs(1,1) signal

The binary offset carrier (BOC) signal results from multiplying a random sequence ± 1V of rate $n \times 1.023$ Mchips/s by a digital subcarrier sgn $[\sin(2\pi f_s t + \theta_{BOC})]$ where $f_s = m \times 1.023$MHz is the subcarrier frequency and $\theta_{BOC} = 0$ for sine-shaped BOC or $\theta_{BOC} = \pi/2$ for cosine-shaped BOC [2].

The sine-shaped binary offset carrier BOCs(m, n) signal is used in the Galileo system [42], to form the composite BOC (CBOC) signal and in the BeiDou system directly [43]. According to [44], the autocorrelation function of the general BOCs(m, n) signal can be expressed as a sum of triangle functions, as follows:

$$R_{BOCs(pn,n)}(\tau) = \Lambda_{T_c/(2p)}(\tau) + \sum_{k=1}^{2p-1} (-1)^k \left(1 - \frac{k}{2p}\right) \Lambda_{T_c/(2p)} \left(|\tau| - k\frac{T_c}{2p}\right) \tag{A.4}$$

where $T_c = 10^{-3}/(1023n)$ s. In order to obtain the autocorrelation function of the BOCs(1,1) signal, the coefficients p and n must be equal to 1. Simplifying (A.4), with $p = n = 1$, we obtain:

$$R_{BOCs(1,1)}(\tau) = \Lambda_{T_c/2}(\tau) - \frac{1}{2}\Lambda_{T_c/2}\left(|\tau| - \frac{T_c}{2}\right) \tag{A.5}$$

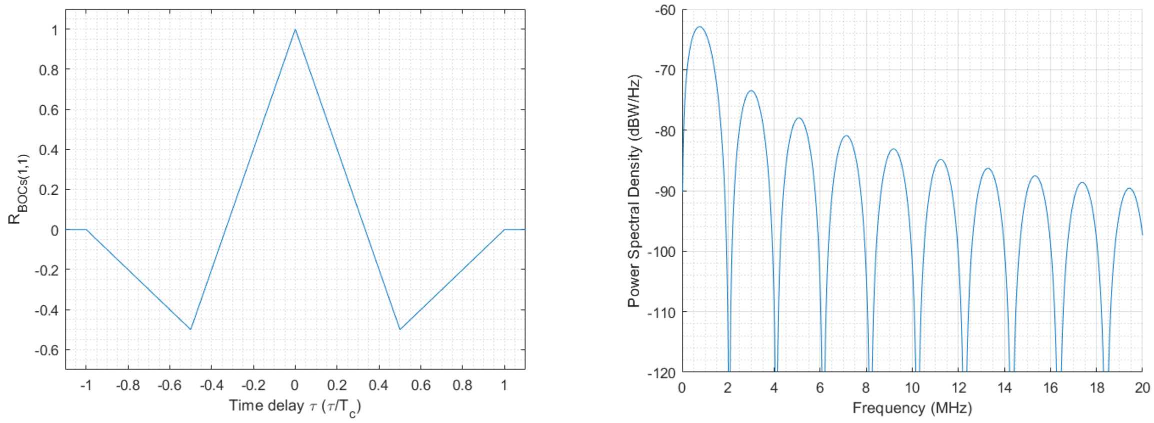
The PSD of the general BOC signal can be expressed as follows [44]:

$$G_{BOCs(pn,n)}(f) = T_c \,\text{sinc}^2(f\,T_c)\,\tan^2\left(\frac{\pi f T_c}{2p}\right) \qquad p = 1, 2, \ldots \tag{A.6}$$

For BOCs(1,1) we obtain:

$$G_{BOCs(1,1)}(f) = T_c \,\text{sinc}^2(f\,T_c)\,\tan^2\left(\frac{\pi f T_c}{2}\right) \tag{A.7}$$

with the chip duration being equal to the one of the C/A signal. A graphical representation of the autocorrelation function and PSD of the BOCs(1,1) is sketched in figure A.2.



(a) Autocorrelation function of the Galileo BOCs(1,1) signal.

(b) Power spectral density of the Galileo BOCs(1,1) signal.

Figure A.2: BOCs(1,1) signal characterization.

A.3 Galileo CBOC(6,1,1/11) signal

We can determine the autocorrelation function of the Galileo CBOC(6,1,1/11) pilot and data signals from [44]:

$$R_{CBOC(6,1,1/11)} = \frac{10}{11}R_{BOCs(1,1)}(\tau) + \frac{1}{11}R_{BOCs(6,1)}(\tau) \pm 2\frac{\sqrt{11}}{10}I_{1,2}(\tau) \tag{A.8}$$

with the $+$ sign standing for the data signal and and the $-$ sign standing for the pilot signal. The terms $R_{BOCs(1,1)}(\tau)$ and $R_{BOCs(6,1)}(\tau)$ can be obtained from (A.4) and the term $I_{1,2}(\tau)$ can be expressed as:

$$\begin{aligned} I_{1,2}(\tau) =& \frac{1}{12}\Bigg[\Lambda_{T_c/12}\left(|\tau| - \frac{T_c}{12}\right) + \Lambda_{T_c/12}\left(|\tau| - \frac{3T_c}{12}\right) \\ &+ \Lambda_{T_c/12}\left(|\tau| - \frac{5T_c}{12}\right) - \Lambda_{T_c/12}\left(|\tau| - \frac{7T_c}{12}\right) \\ &- \Lambda_{T_c/12}\left(|\tau| - \frac{9T_c}{12}\right) - \Lambda_{T_c/12}\left(|\tau| - \frac{11T_c}{12}\right)\Bigg] \end{aligned} \tag{A.9}$$

The PSD of the Galileo CBOC(6,1,1/11) can be expressed as [44]:

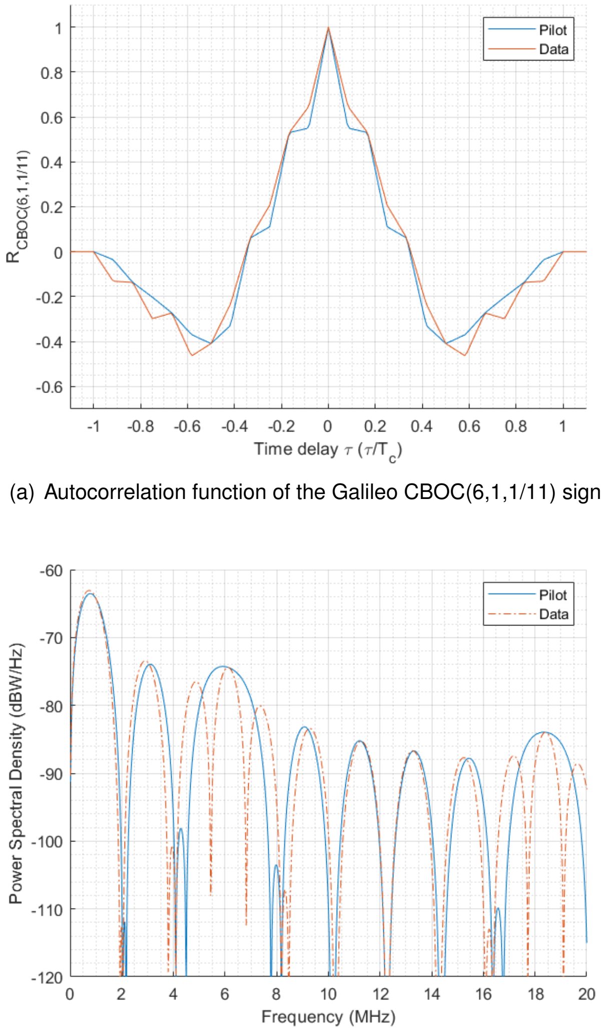
$$G_{CBOC(6,1,1/11)}(f) = \frac{10}{11}G_{BOCs(1,1)}(f) + \frac{1}{11}G_{BOCs(6,1)}(f) \pm \frac{2\sqrt{10}}{11}G_{1,2}(f) \tag{A.10}$$

where, once more, the $+$ sign stands for the data signal and the $-$ stands for the pilot signal.

The expressions for $G_{BOCs(1,1)}(f)$ and $G_{BOCs(6,1)}(f)$ are obtained from (A.6) and $G_{1,2}(f)$ is given by:

$$G_{1,2}(f) = \frac{T_c}{36}\text{sinc}^2\left(\frac{fT_c}{12}\right)\frac{\sin^2\left(\frac{\pi f T_c}{2}\right)\sin(\pi f T_c)}{\sin\left(\frac{\pi f T_c}{6}\right)} \tag{A.11}$$

In figure A.3 we can observe theboth the plots for the autocorrelation funtion and the power spectral density of the CBOC(6,1,1/11) pilot and data signals.



(a) Autocorrelation function of the Galileo CBOC(6,1,1/11) signal.

(b) Power spectral density of the Galileo CBOC(6,1,1/11) signal.

Figure A.3: BOCs(1,1) signal characterization.

Appendix B

Generation of random correlated Gaussian noise vectors

In the present thesis, it is required to generate vectors of correlated noise, namely, inphase and quadrature noise vectors, for use in the various simulations undertaken. This annex will address the method used for generating those vectors, with any desired dimension, as well as present some important specific cases. The methods here presented are described in further detail in [8] and [45].

B.1 Generation process

Consider the problem of generating a zero-mean Gaussian random vector $X_{(n\times 1)}$ and covariance matrix $\Xi_{(n\times n)}$, using a zero-mean Gaussian vector $U_{(n\times 1)}$ with independent components of unity variance, that is, $E\{UU^T\} = I_{(n\times n)}$. Specifically we are looking for a square matrix $\widetilde{G}_{(n\times n)}$ such that $X = \widetilde{G}U$. Thus,

$$\Xi = E\{\widetilde{G}U(\widetilde{G}U)^T\} = \widetilde{G}\widetilde{G}^T \tag{B.1}$$

According to [46], we can write the covariance matrix in terms of its eigenvalues and eigenvectors as

$$\Xi = VLV^T \tag{B.2}$$

with V being a matrix whose columns are the eigenvalues of Ξ and L is a diagonal matrix whose main diagonal is composed by the $\lambda_{i=1,\cdots,n}$ eigenvalues of Ξ. By comparing (B.1) and (B.2) we can obtain the expression for $\widetilde{G}$:

$$\widetilde{G} = VL^{1/2} = V\text{diag}\{\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}\} \tag{B.3}$$

B.2 Early-Prompt-Late correlators

A special case of interest of the process just described is the case where we have correlation of the incoming $\tilde{r}_{I_k}$ and $\tilde{r}_{Q_k}$ signals with three locally generated code replicas: early, prompt and late. In this case, a more direct approach to generating a random correlated Gaussian noise vector can be taken. Let us suppose we want to generate a generic $N = [N_1\, N_2\, N_3]^T$ noise vector with zero-mean and the following covariance matrix:

$$E\{N\,N^T\} = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\kappa_{12} & \sigma_1\sigma_3\kappa_{13} \\ \sigma_1\sigma_2\kappa_{12} & \sigma_2^2 & \sigma_2\sigma_3\kappa_{23} \\ \sigma_1\sigma_3\kappa_{13} & \sigma_2\sigma_3\kappa_{23} & \sigma_3^2 \end{bmatrix} \tag{B.4}$$

where σ_i^2 represents the variance of N_i and $\kappa_{ij} = E\{N_i N_j\}/\sigma_i\sigma_j$ being the correlation coefficient of N_i and N_j. We can obtain vector N from a zero-mean U vector, with unity variance Gaussian independent random variables, through the following method:

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} \tag{B.5}$$

It can be shown that the components α_{ij} can be expressed as:

$$\begin{aligned} \alpha_{11} &= \sigma_1 \\ \alpha_{21} &= \sigma_2\kappa_{12} \\ \alpha_{31} &= \sigma_3\kappa_{13} \\ \alpha_{22} &= \pm\sigma_2\sqrt{1-\kappa_{12}^2} \\ \alpha_{32} &= \pm\sigma_3\frac{\kappa_{23}-\kappa_{12}\kappa_{13}}{\sqrt{1-\kappa_{12}^2}} \\ \alpha_{33} &= \pm\sigma_3\sqrt{1-\kappa_{13}^2-\frac{(\kappa_{23}-\kappa_{12}\kappa_{13})^2}{1-\kappa_{12}^2}} \end{aligned} \tag{B.6}$$

Another case of particular interest is when there is correlation of the incoming signal with only two code replicas: an early and a late one; which is the case for a NELP discriminator, that sees widespread usage across GNSS receivers. In that case, we have $N = [N_1\, N_2]^T$ with the following covariance matrix:

$$E\{N\,N^T\} = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\kappa_{12} \\ \sigma_1\sigma_2\kappa_{12} & \sigma_2^2 \end{bmatrix} \tag{B.7}$$

We can obtain N from $U = [U_1\, U_2]^T$ if we make $\alpha_{31} = \alpha_{32} = \alpha_{33} = 0$. The result is:

$$\begin{bmatrix} N_1 \\ N_2 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & 0 \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \tag{B.8}$$

with the coefficients being obtained from (B.6), as:

$$\begin{aligned}\alpha_{11} &= \sigma_1 \\ \alpha_{21} &= \sigma_2 \rho_{12} \\ \alpha_{22} &= \pm\sigma_2 \sqrt{1-\kappa_{12}^2}\end{aligned} \tag{B.9}$$

Appendix C

Generation of a normalized code discriminator from a bank of correlators

The main objective of the bank of correlators is trying to approximate its response to an ideal S-curve previously devised, such as proposed in [47]. Let us assume we wish that the bank of correlators produces the following S-curve:

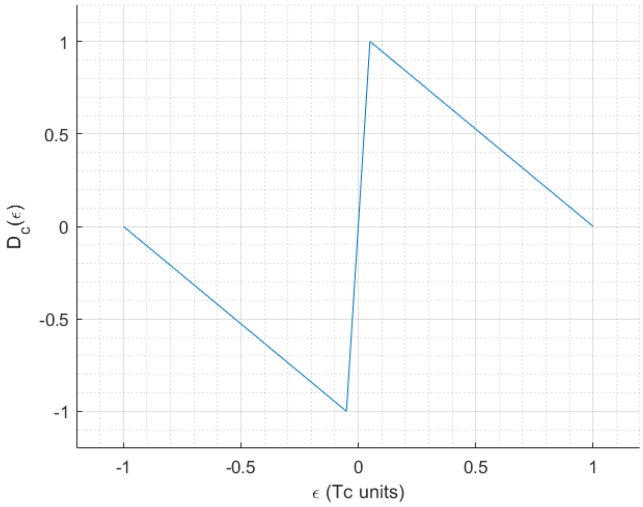


Figure C.1: Target S-curve.

which is defined by:

$$\widetilde{D}(\epsilon) = \begin{cases} \epsilon/e, & |\epsilon| < e \\ [(T_c - |\epsilon|)/(T_c - e)]\,\mathsf{sign}(\epsilon), & e \leq |\epsilon| \leq T_c \\ 0, & |\epsilon| > T_c \end{cases} \tag{C.1}$$

with $e/T_c = 0.05$. The discriminator response curve shown in figure C.1 is robust to the effect of false-code lock, while exhibiting, at the same time, a linear region near the origin followed by a slow linear decay to zero.

Let us assume our discriminator is composed of $2M+1$ correlators that correlate the vector $[\tilde{r}_{I_n}(t)\,\tilde{r}_{Q_n}(t)]^T$ with $2M+1$ early and late locally generated code replicas of the type $X_n(t + iT_c/M - \epsilon_k)$ with $i = -M, -M+1, \ldots, M$, T_c representing the code chip time and ϵ_k representing the delay error of the local code generator, regarding the incoming signal [45]. This process, which is similar to the one presented in chapter 2 (figure 2.3), is now presented in figure C.2.

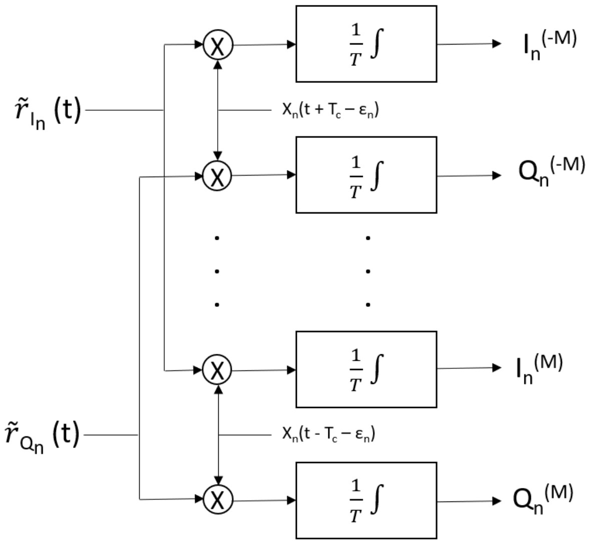


Figure C.2: Bank of inphase/quadrature correlators [45].

For the inphase/quadrature correlators outputs, we obtain:

$$\begin{aligned} I_n^{(i)} &= \frac{1}{T}\int_0^T \tilde{r}_{I_n}(t)\,X_n\left(t + \frac{iT_c}{M} - \epsilon_n\right)\,dt \\ &= A_n R_X\left(\epsilon_n - \frac{iT_c}{M}\right)\cos(\theta_n) + N_{I_n}^{(i)}, \qquad i = -M, \ldots, M \\ Q_n^{(i)} &= \frac{1}{T}\int_0^T \tilde{r}_{Q_n}(t)\,X_n\left(t + \frac{iT_c}{M} - \epsilon_n\right)\,dt \\ &= A_n R_X\left(\epsilon_n - \frac{iT_c}{M}\right)\sin(\theta_n) + N_{Q_n}^{(i)}, \qquad i = -M, \ldots, M \end{aligned} \tag{C.2}$$

where A_n is the signal amplitude and θ_n is the signal phase. The autocorrelation function of the code X_n is given by:

$$R_X(\tau) = \frac{1}{T_p}\int_0^{T_p} X(t)X(t-\tau)\,d\tau \tag{C.3}$$

where T_p is the code period. The noise terms $N_{I_n}^{(i)}$ and $N_{Q_n}^{(i)}$ are determined using the method presented in Appendix B.

Assume we have the following vector $N_{I_n} = \left[N_{I_n}^{(-M)}, \cdots, N_{I_n}^{(M)}\right]^T$, that contains the noise components resulting from the correlation of the $\tilde{n}_{I_n}$ noise part of the $\tilde{r}_{I_n}$ signal with the $2M+1$ early and late code replicas. Each component of N_{I_n} can be expressed as follows:

$$N_{I_n}^{(i)} = \frac{1}{T}\int_0^T \tilde{n}_{I_n}(t)\, X_n\left(t + \frac{iT_c}{M} + \epsilon_n\right) dt \qquad i = -M, -M+1, \cdots, M-1, M \tag{C.4}$$

having zero-mean and variance given by:

$$\begin{aligned} E\{(N_{I_n}^{(i)})^2\} &= \frac{1}{T^2}\int_0^T\int_0^T E\{\tilde{n}_{I_n}(t)\,\tilde{n}_{I_n}(\lambda)\}\, X_n\left(t + \frac{iT_c}{M} - \epsilon_n\right)\, X_n\left(\lambda + \frac{iT_c}{M} - \epsilon_n\right)\}\, dt\, d\lambda \\ &= \frac{N_0}{T^2}\int_0^T d\lambda = \frac{N_0}{T} \end{aligned} \tag{C.5}$$

All the components of N_{I_n} are correlated but are independent from the components of its quadrature counterpart N_{Q_n}. The noise covariance matrix of both these vectors is:

$$E\{N_{I_n}^{(i)} N_{I_n}^{(j)}\} = E\{N_{Q_n}^{(i)} N_{Q_n}^{(j)}\} = \Xi \tag{C.6}$$

with the matrix's components being defined as:

$$\begin{aligned} \Xi(i,j) = E\{N_{I_n}^{(i)} N_{I_n}^{(j)}\} &= \frac{1}{T^2}\int_0^T\int_0^T E\{\tilde{n}_{I_n}(t)\,\tilde{n}_{I_n}(\lambda)\}\, X_n\left(t + \frac{iT_c}{M} - \epsilon_n\right)\, X_n\left(\lambda + \frac{jT_c}{M} - \epsilon_n\right)\}\, dt\, d\lambda \\ &= \frac{N_0}{T} R_X\left(\frac{(i-j)T_c}{M}\right) \end{aligned} \tag{C.7}$$

We can generate vector N_{I_n} or N_{Q_n} from a general U vector, composed of $2M+1$ independent Gaussian random variables with zero-mean and unity variance through $N_{I_n} = \widetilde{G}U$, where $\widetilde{G}$ is a matrix that has to be determined. This allows for the following relation:

$$\Xi = \widetilde{G}\widetilde{G}^T \tag{C.8}$$

Once more, according to [46], we can write the covariance matrix in terms of its eigenvalues and eigenvectors:

$$\Xi = VLV^T \tag{C.9}$$

with V being a matrix whose columns are the eigenvectors of Ξ and L is a diagonal matrix, whose main diagonal is composed of the $\lambda_{i=1,\cdots,2M+1}$ eigenvalues of Ξ. By analyzing (C.8) and (C.9) we finally obtain the expression for $\widetilde{G}$:

$$\widetilde{G} = VL^{1/2} = V\,\text{diag}\{\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_{2M+1}}\} \tag{C.10}$$

Now, let us consider that the response of the proposed normalized code discriminator is expressed by [45]:

$$D_c(\epsilon) = \frac{\sum_{i=-M}^{M} \mu_i \left[\left(I_n^{(i)}\right)^2 + \left(Q_n^{(i)}\right)^2\right]}{\sum_{i=-M}^{M} \left(I_n^{(i)}\right)^2 + \left(Q_n^{(i)}\right)^2} \tag{C.11}$$

where the term μ_i represents the weight coefficient of each pair of inphase and quadrature correlators. If we assume that there is absence of noise in equation (C.11), it can be re-written as:

$$D_c(\epsilon) = \frac{1}{S(\epsilon)} \sum_{i=-M}^{M} \mu_i R_X^2 \left(\epsilon - \frac{iT_c}{M}\right) \tag{C.12}$$

with:

$$S(\epsilon) = \sum_{i=-M}^{M} R_X^2 \left(\epsilon - \frac{iT_c}{M}\right) \tag{C.13}$$

Equation (C.12) allows for an S-curve that is independent of the amplitude A_n, meaning the discriminator is normalized. Now, it is needed to synthesize the vector of weight coefficients $C = [\mu_{-M}, \ldots, \mu_M]^T$ that best approximates $D_c(\epsilon_n)$ to $\widetilde{D}_c(\epsilon_n)$ (given in equation (C.1)) for a set of delays ϵ_j, where $-T_c \leq \epsilon_j \leq T_c$, $j = 1, \ldots, J$ and $J \geq 2M+1$. The weight coefficients can be expressed as the solution to a least squares problem, where the starting point is the desired S-curve $\widetilde{D}_k(\epsilon)$ [45]:

$$\begin{bmatrix} S(\epsilon_1)\widetilde{D}_c(\epsilon_1) \\ S(\epsilon_2)\widetilde{D}_c(\epsilon_2) \\ \vdots \\ S(\epsilon_J)\widetilde{D}_c(\epsilon_J) \end{bmatrix} = \underbrace{\begin{bmatrix} R_X^2\left(\epsilon_1 + T_c\right) & R_X^2\left(\epsilon_1 + \frac{(M-1)T_c}{M}\right) & \cdots & R_X^2\left(\epsilon_1 - T_c\right) \\ R_X^2\left(\epsilon_2 + T_c\right) & R_X^2\left(\epsilon_2 + \frac{(M-1)T_c}{M}\right) & \cdots & R_X^2\left(\epsilon_2 - T_c\right) \\ \vdots & \vdots & \ddots & \vdots \\ R_X^2\left(\epsilon_J + T_c\right) & R_X^2\left(\epsilon_J + \frac{(M-1)T_c}{M}\right) & \cdots & R_X^2\left(\epsilon_J - T_c\right) \end{bmatrix}}_{R} \begin{bmatrix} \mu_{-M} \\ \mu_{-M+1} \\ \vdots \\ \mu_M \end{bmatrix} \tag{C.14}$$

Equation (C.14) can be re-written in a more compact way, as:

$$S\widetilde{D} = RC \tag{C.15}$$

where:

$$S = \begin{bmatrix} S(\epsilon_1) & & & \mathbf{0} \\ & S(\epsilon_2) & & \\ & & \ddots & \\ \mathbf{0} & & & S(\epsilon_J) \end{bmatrix}, \quad \widetilde{D} = \begin{bmatrix} \widetilde{D}_c(\epsilon_1) \\ \widetilde{D}_c(\epsilon_2) \\ \vdots \\ \widetilde{D}_c(\epsilon_J) \end{bmatrix}, \quad C = \begin{bmatrix} \mu_{-M} \\ \mu_{-M+1} \\ \vdots \\ \mu_M \end{bmatrix} \tag{C.16}$$

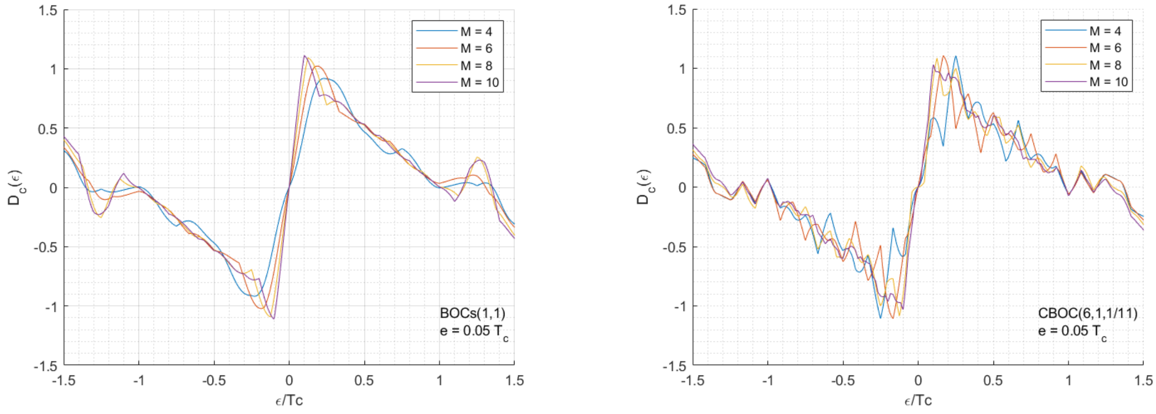
Finally, the least-squares solution of (C.14) is:

$$C = (R^T R)^{-1} R^T S \widetilde{D} \tag{C.17}$$

In figure C.3, the S-curves obtained using this method, for BOCs(1,1) and CBOC(6,1,1/11) signals, are presented. They attempt to mimic the curve shown in figure C.1, therefore considering $e/T_c = 0.05$. The μ_i coefficients, for the displayed curves, were obtained with a set of $J = 1000$ uniform delays, defined by:

$$\epsilon_j = -1 + (j-1)\frac{2}{J-1}, \qquad j = 1, \ldots, J \tag{C.18}$$

Both sets of curves presented in figure C.3 appear to be a good approximation of the ideal S-curve described by equation (C.1), albeit with some undesired behavior for $\epsilon/T_c > 1$, where for positive values



(a) S-curve for a BOCs(1,1) signal

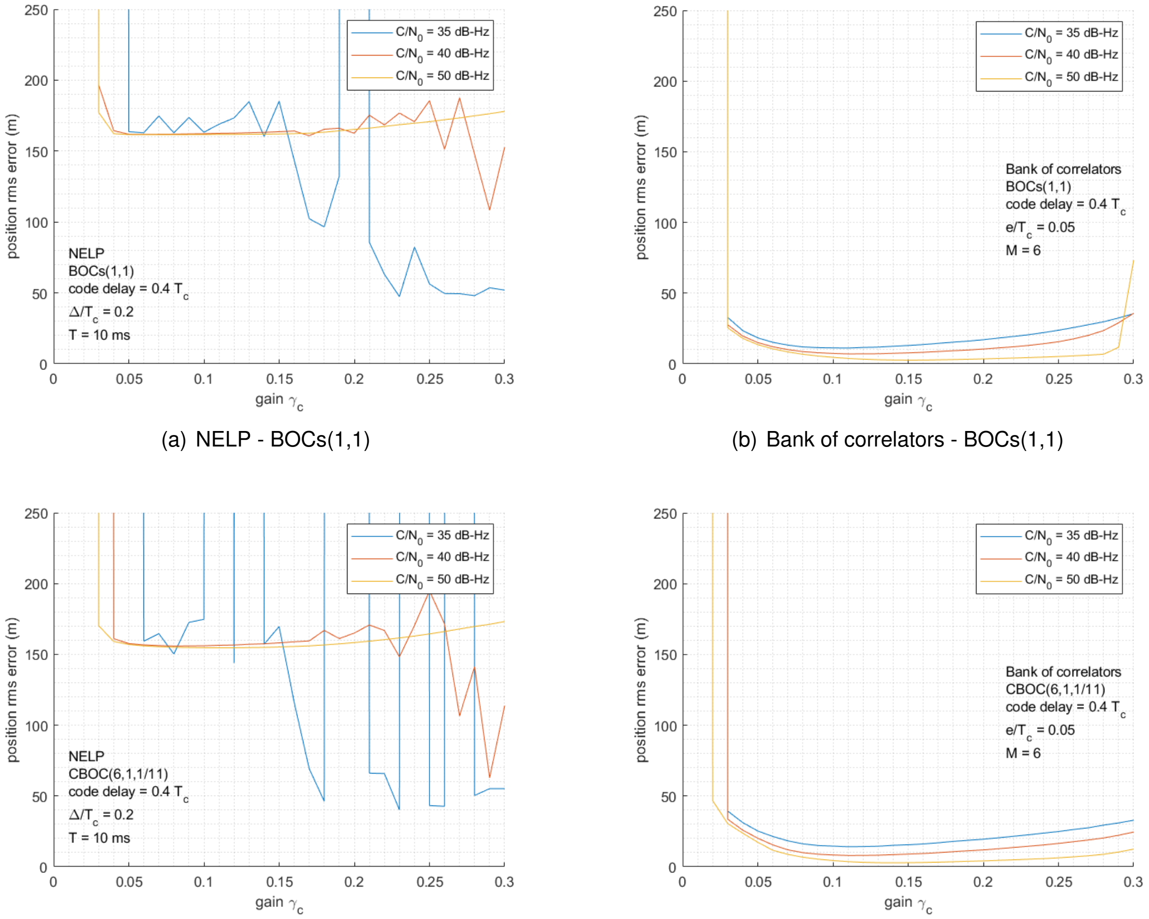
(b) S-curve for a CBOC(6,1,1/11) signal

Figure C.3: S-curves obtained with a bank of correlators.

of ϵ/T_c correspond negative values of $D(\epsilon)$ (and the opposite for negative values of ϵ/T_c smaller than -1). This phenomenon, however, can be disregarded as the amount of noise necessary to reach that value of ϵ/T_c implies that the receiver is already unable to provide a good estimate of its position. Thus, we are much more interested in the discriminator's response near $\epsilon/T_c = 0$, where it should present a linear behavior, and up to $\epsilon/T_c = 0.6$.

If we take a step back to analyze the NELP discriminator's response for both the BOCs(1,1) and CBOC(6,1,1/11) signals, in figure 2.5, we can observe that, for around $\epsilon/T_c \approx -0.6$ or $\epsilon/T_c \approx 0.5$, the curve crosses the horizontal axis with a positive slope. This means that a false-code lock can already occur, something that by using the bank of correlators, is not possible. Let us now compare the performance obtained with the bank of correlators, for an initial code delay error of $0.4T_c$, against the performance obtained with the NELP discriminator, in figure C.4.

We can clearly see from that image that for an initial code delay of $0.4\,T_c$, for both the BOCs(1,1) and CBOC(6,1,1/11) signals, the DLL with the NELP discriminator converges to a false-code lock around $\epsilon = 0.5\,T_c$ (as shown in figure 2.5, for $\Delta/T_c = 0.2$), for $C/N_0 \geq 40$ dB-Hz (for $C/N_0 = 35$ dB-Hz the NELP discriminator appears to diverge from a solution). This false-code lock convergence to $\epsilon \approx 0.5\,T_c$ would translate to a rms error of ≈ 160 m, thus preventing accurate positioning of the receiver. In contrast to these results, the DLL with the discriminator formed by the bank of correlators does not reveal whatsoever any convergence to false-code lock, for both signals. This is corroborated by the rms errors presented by the bank of correlators, which can reach values as low as 2.5 m.



(a) NELP - BOCs(1,1)

(b) Bank of correlators - BOCs(1,1)

(c) NELP - CBOC(6,1,1/11)

(d) Bank of correlators - CBOC(6,1,1/11)

Figure C.4: NELP *vs* bank of correlators performance for an initial code delay of $0.4T_c$.

Appendix D

Characterization of an imperfect oscillator

In a GNSS receiver, the clock is perhaps the most important component as it affects pseudorange and Doppler frequency measurements, as well as carrier frequency and code phase synchronization. As such, we will present, in this appendix, a concise analysis on a general imperfect oscillator, which is used by a GNSS receiver to measure time (making it effectively its clock).

We can express the output of a real oscillator as:

$$s(t) = A\cos\left(\Phi(t)\right) \tag{D.1}$$

with A representing the oscillation amplitude and $\Phi(t)$ the total phase. We can also define the instantaneous frequency of oscillation, in Hz, as:

$$f_i(t) = \frac{1}{2\pi} \cdot \frac{d}{dt}\Phi(t) \tag{D.2}$$

which can be modeled as [48]:

$$f_i(t) = f_0 + \Delta f + \dot{\Delta f}\, t + \frac{1}{2\pi}\dot{\Psi}(t) \tag{D.3}$$

Thus, we have:

$$\Phi(t) = 2\pi(f_0 + \Delta f)t + \pi\dot{\Delta f}\, t^2 + \Psi(t) + \theta_0 \tag{D.4}$$

A perfect oscillator will have just one constant frequency component f_0. However, imperfect (real) oscillators have the f_0 component distorted by deterministic phenomena, such as the frequency drift Δf (Hz) and drift-rate $\dot{\Delta f}$ (Hz/s); and the random phase noise process $\Psi(t)$ (rad). Let us describe the random fluctuations of the instantaneous frequency, caused by the phase noise, through the normalized frequency deviation $y(t)$:

$$y(t) = \frac{1}{2\pi f_0} \cdot \frac{d}{dt}\Psi(t) \tag{D.5}$$

We can quantify the the normalized instantaneous frequency drift caused by the oscillator phase noise $\Psi(t)$, in a given time interval τ, through the Allan variance:

$$\sigma_y^2(\tau) = \frac{E\{[\Psi(t+2\tau) - \Psi(t+\tau)] - [\Psi(t+\tau) - \Psi(t)]\}^2}{(2\tau)^2(2\pi f_0)^2} \tag{D.6}$$

which can be computed in terms of h_α parameters, that characterize a given type of oscillator [35]:

$$\sigma_y^2(\tau) = \frac{(2\pi)^2}{6}h_{-2} + 2\ln(2)h_{-1} + \frac{h_0}{2\tau} + \frac{1.038 + 3\ln(2\pi f_h \tau)}{(2\pi)^2\tau^2}h_1 + \frac{3f_h}{(2\pi)^2\tau^2}h_2 \tag{D.7}$$

where f_h represents the upper cutoff frequency of the measuring equipment being used and the parameters h_{-2} to h_2 represent, respectively, the following types of noise: random walk frequency, flicker frequency, white frequency, flicker phase and white phase.

In figure D.1 we can see the plots of the Allan standard deviation ($\sigma_y(t)$) for quartz oscillators (TCXO and OCXO) and rubidium oscillators, using the h_α parameters, available in [34]. The quartz oscillators present good frequency stability for small intervals, such as $0 < \tau < 100$ s with $\sigma_y^2(\tau) \approx 10^{-20}$ but, due to the ageing effect, exhibit considerable degradation for longer intervals ($\sigma_y^2(\tau) \approx 10^{-17}$ for 1 day). On the other hand, rubidium oscillators present very good frequency stability over large periods of time, with $\sigma_y^2(\tau) < 10^{-22}$ for intervals of 1 day.

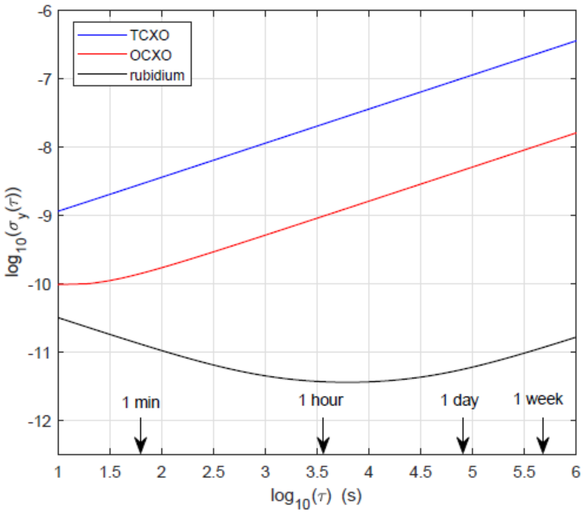


Figure D.1: Allan variance for TCXO (temperature-compensated XTAL oscillator), OCXO (ovenized XTAL oscillator) and rubidium oscillators.

If we consider that an imperfect oscillator is used as a clock, we can write its time function as [48]:

$$T(t) = \frac{1}{2\pi f_0}\Phi(t) \tag{D.8}$$

which can be re-written, using (D.4), as:

$$T(t) = \left(1 + \frac{\Delta f}{f_0}\right)t + \frac{\dot{\Delta f}}{2f_0}t^2 + \frac{\Psi(t) - \Psi(t_0)}{2\pi f_0} - t_0 \tag{D.9}$$

where $\Delta f/f_0$ represents the normalized frequency drift and t_0 represents a constant delay. This means that $\Delta f > 0$ yields a time compression and $\Delta f < 0$ leads to a time dilation. A perfect oscillator would generate the time function $T(t) = t - t_0$.