UNIVERSITÀ DI PALERMO

SCUOLA DI DOTTORATO
Dottorato in Ingegneria Elettronica e delle Telecomunicazioni
XXII Ciclo

Tesi di Dottorato

Kalman Filters for GNSS Applications

Marco Rao

Tutore
Ch.mo Prof. Giovanni Garbo

Coordinatore del corso di dottorato
Ch.mo Prof. Stefano Riva Sanseverino

Gennaio 2011

Le idee valide si esprimono con poche parole.

Summary

The knowledge of the kinematic characteristics of a vehicle or of a pedestrian is at the moment a main focus for the development of new applications. This is possible thanks to the availability of positioning services that are based on the existence of Global Navigation Satellite Systems (GNSSs), that play a fundamental role in this context.

At the moment the Global Positioning System (GPS), which was initially developed as a military system and subsequently evolved for civil applications, is the only functioning and widely available satellite based navigation system. Nontheless, many other satellites systems are being developed. Among them there are the European Galileo, the Chinese BeiDou-Compass, the Indian Regional Navigational Satellite System (IRNSS) and the GLObal NAvigation Satellite System (GLONASS).

All these systems will be useful for several different applications of ordinary life, like transports (terrestrial, maritime, and avionic), tracking of the vehicles fleets (trains, trucks, vessels, etc), the synchronization of telecommunication (DVB-T) and energy distribution networks, the precise mapping of the territory and the integration with other navigation systems for the automatic guide of agricultural machines, just to consider only a few of the most important civil applications. This is happening thanks to the development, the improvement and the ease of access to the technology related to the GNSS.

A GNSS is devoted to broadcast a signal which is received and processed by a receiver. These devices are demanded to run a large number of procedures, whose final aim is the computation of the navigation solution, which is usually called the Position Velocity Time (PVT) solution. The Kalman filter represents a fundamental block which is usually implemented in a GNSS receiver, since it can be used for tracking, positioning, integration and to develop more robust receivers.

This thesis resumes the Ph. D. program in the GNSS field in the period 2008-2010. The main topic of the Ph. D. activity is related to the analysis of the Kalman Filter (KF) and

of its variants, in particular the Extended Kalman Filter (EKF), in order to understand why it is a valuable tool in the navigation field. The whole activity can be summarized as follows:

- *First year of the Ph. D. program.* The activity was focused on the study of possible applications of the Kalman filter to the telecommunications area. A bibliographic research has been held and experiments related to the equalization of a Multiple Input Multiple Output (MIMO) channel have been conducted, in order to get familiar with this family of filters.

- *Second year of the Ph. D. program.* The research was related to the first main topic of the Ph. D. program: the study of the integration between a GNSS receiver and an Inertial Navigation System (INS). I led my research activity in the NavSAS lab in Politecnico di Torino, where I was under the tutoring of Prof. Letizia Lo Presti and where I had the chance to use different kinds of receivers and Inertial Measurement Units (IMUs). During this stay, the research activity followed two different paths. The first one went through a deeper analysis of the Kalman filter, since the task of integration requires the exploitation of the extended Kalman filter. The second path went through the analysis of the architecture of the GNSS receivers.

- *Third year of the Ph. D. program.* Finally, the research activity was related to the Peer to Peer (P2P) positioning project sponsored by the European Space Agency (ESA). The integration of the GNSS measurements with altitude measurements, available thanks to the presence of a P2P network, was investigated, bringing to the development of new and improved positioning algorithms.

A chi mi è stato più vicino.

Contents

List of Tables

List of Figures

List of Acronyms

AA Altitude Aided

ADC Analog to Digital Converter

AGC Adaptive Gain Control

AWGN Additive White Gaussian Noise

CAF Complex-Ambiguity Function

CDMA Code Division Multiple Access

COTS commercial off the shelf

DCM Direct Cosine Matrix

DoD Department of Defense

DLL Delay Lock Loop

DVB Digital Video Brodcasting

ECEF Earth-Centered Earth-fixed

EKF Extended Kalman Filter

ENB Equivalent Noise Bandwidth

ENU East-North-Up

ESA European Space Agency

FFT Fast Fourier Transform

FIR Finite Impulse Response

FLL Frequency Lock Loop

GLONASS GLObal NAvigation Satellite System

GM Gauss Markov

GNSS Global Navigation Satellite System

GPS Global Positioning System

IAA Iterative Altitude Aided

IF Intermediate Frequency

IIR Infinite Impulse Response

IMU Inertial Measurement Unit

INS Inertial Navigation System

IRNSS Indian Regional Navigational Satellite System

JPO Joint Program Office

KF Kalman Filter

LKF Linearized Kalman Filter

LOS Line of Sight

LPF Low Pass Filter

LS Least Squares

MAP Maximum A Posteriori

MEMS Micro Electro-Mechanical Systems

MIMO Multiple Input Multiple Output

ML Maximum Likelihood

MSE Mean Squared Error

MMSE Minimum Mean Squared Error

NCO Numerically Controlled Oscillator

NED North-East-Down

P2P Peer to Peer

PD Phase Detector

PLL Phase Lock Loop

PRN Pseudo Random Noise

PSD Power Spectral Density

PVT Position Velocity Time

RF Radio Frequency

RFI Radio Frequency Interference

SDR Software Defined Radio

TOA Time of Arrival

UAV Unmanned Air Vehicles

UWB Ultra Wide Band

WAVE Wireless Access in Vehicular Environment

WGS84 World Geodetic System 1984

Chapter 1

Introduction

1.1 Navigation

The art of navigation has very old origins and it changed very much with respect to its first forms. Despite this evolution, it is still related to the need of moving from a place to another one. This task can be achieved in many different ways.

The most simple form of navigation one can think about is the one based on waypoints. It relies on the existence of recognizable features and objects that characterize the environment. Of course, this method is reliable as long as these references exist.

A more evolved kind of navigation is the one based on cartography. The possibility to use a map allows the navigator to estimate its position on the basis of the observed environment and to determine its position in a reference frame. For example, once the user has located himself on a map, he can determine its approximate latitude and longitude with respect to the equator and to the Greenwich meridian, respectively.

Different needs arose with the birth of maritime navigation. At the beginning, ancient civilizations traveled using the coastal line as a reference point, otherwise they would have been lost in the sea.

Open sky navigation was made possible when navigators observed that stars rotates around a fixed point. Nowadays, this fixed point is close to the Polar Star, but due to equinox precession and other astronomical phenomena it changes in time, as shown in Figure 1.1.

This technique was particularly useful to determine maritime routes, since it did not rely on the coastal line, but it only implied the determination of the height of known

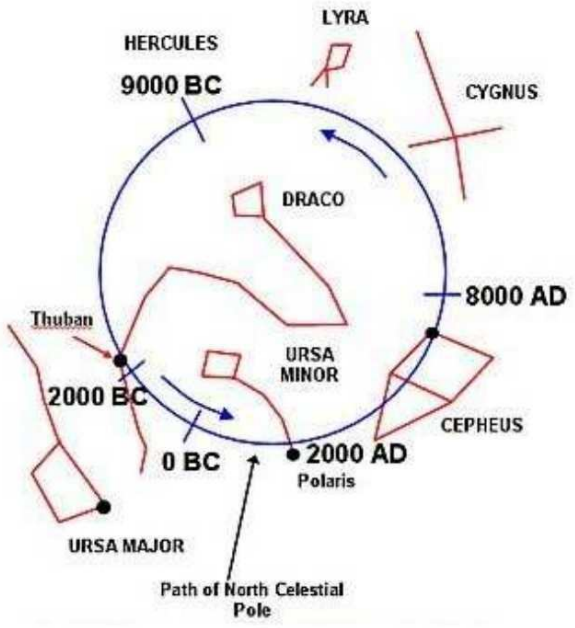

Figure 1.1. Celestial North Pole in time.

stars above the horizon, using a sextant. Of course, other problems arose: the precision of the route was strictly dependant on the time reference, that was not available from the beginning, and other reference had to be used at daytime.

In the last decades many evolved navigation systems were developed. After millennia, we still look at the sky to get oriented, not only on the sea, but also in the air and in the ground. In fact nowadays, the satellite based positioning and navigation is one of the most used service to navigate and it is familiar to a growing number of people. Thanks to the recent advance in technology, GNSS receivers are available for a few dollars and are integrated in many devices, allowing to estimate the position of a user equipped with a simple GNSS handheld receiver.

The navigation technique based on a GNSS relies on the existence of a constellations of satellites, as shown in Figure 1.2, whose position is known with precision. By means of triangulation, a user equipped with a GNSS receiver is enabled to estimate his own position.

The first project related to satellite based navigation emerged in the Sixties: the U.S. Department of Defense (DoD) needed a global, all-weather conditions, continuously available and accurate positioning and navigation system. In those years, the U.S. Navy

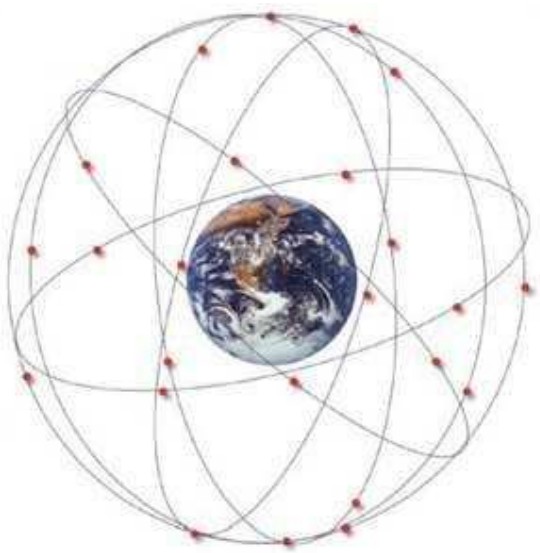

Figure 1.2. Satellites constellation.

developed the first two navigation systems based on satellites, the Transit and the Timation.

In December 1973, the DoD decided to create a Joint Program Office (JPO) which was in charge of coordinating the development of a new global positioning system based on satellites, to improve the performance of the previous systems. In this way, the NAVSTAR-GPS was born [1]. In the Nineties, the GPS became fully operational, both on military and civil side.

Despite many GNSSs are being developed at the moment, the GPS is the only system that is available to the most of the people, despite also GLONASS exists.

One of the main strengths of the GPS is that it provides absolute navigation information, that has no dependency on the considered initial point and whose error is not subject to drifts, since it is bounded in time.

The main disadvantage of the GPS is instead its inability to guarantee a continuous and reliable solution. In fact, every GNSS measurement is reliable if the Line of Sight (LOS) between the satellites and the user is not obstructed. This is not guaranteed in an urban scenario, where buildings may interrupt the LOS. Even dense foliage areas may be detrimental for a correct operation.

Moreover, the GPS is sensitive to many kind of disturbances, starting from multipath and going through jamming, that may be voluntary or not, and spoofing, which is usually

voluntary.

Due to these weaknesses, the reliability of the GPS should be improved using navigation devices that show different and complementary characteristics with respect to the GPS.

One of the best suited modern navigation devices to play this role is represented by INSs. These are self-contained devices that allow for dead reckoning. In this case an estimate of the position is computed from knowledge of the initial position and measurements of acceleration and angular rates. On the contrary of the GPS and of other GNSSs, an INS is unsensitive to the interference of external electromagnetic fields and it operates independently from the environment.

An INS is capable to provide a navigation solution with continuity, but the quality of this solution degrades in time, compromising the long term accuracy. This drift is considerable when using a low quality Micro Electro-Mechanical Systems (MEMS), while better IMUs, like the tactical or navigation ones, guarantee better stability in time.

It is then necessary to periodically refresh the estimate of an INS, to avoid large drifts and to bound the errors. Thanks to the existence of low cost MEMS and GPS receiver, in the last years these two systems have been coupled to compensate the respective lacks and to design more reliable and complex systems.

At the end of this short history of navigation, it should appear clear that one of the main differences between old navigation techniques and new navigation techniques has to be found in the complexity of the navigation system, that is strictly linked to performance. An improvement is possible thanks to the continuously growing available computational power, that allows for better and more complex data processing. While old navigators could only rely on stars, simple instruments and their own ability to exploit their observations, modern systems are capable to process data automatically and to provide accurate estimates of the position of the user.

Nowadays, the main issue is to find reliable algorithms to process the available measurements.

1.2 Data Affected by Uncertainty

Since every navigation system operates in the real world, all the collected data is affected by a certain degree of uncertainty. If one considers the measurements related to a GNSS, an IMU or to an older navigation system, they are always affected by a certain amount of

noise.

Since noise is present, two main issues arise:

- is it possible to compute an estimate of the user position that mitigates or minimizes the impact of noise?
- if the first task is accomplished, is it possible to quantify the reliability of the obtained estimate?

The obtained estimate is not equal to the real position, but it should be the best computable guess, i.e. the one that minimizes the effects of noise and that is as close as possible to the actual position. Probability theory provides the tools to model the real world taking into account this uncertainty, by means of a probabilistic inference process.

Probabilistic inference can be stated in terms of a state-space model: our aim is to estimate some hidden variables that, at least in part, we cannot access directly, e.g. the position of the user, by means of some noisy observations, that are in some way related to the state of the system. This scenario is depicted in Figure 1.3.

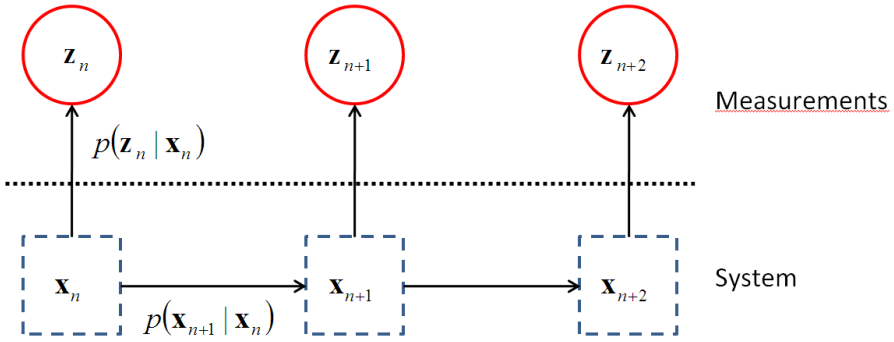

Figure 1.3. Dynamic state-space model.

The graph shown in Figure 1.3 belongs to the family of Bayesian networks. These networks do not necessarily model also the time flow, but they usually model the connection between the nodes (in this case states and observations) of the network using edges.

In our case the necessity for a time dependency arises, so edges are replaced by arrows. Moreover, arrows are also used to point out the cause-effect relation that exists between the state and the observation at a given time.

Considering the vector of observation from time 1 up to n $\mathbf{z}_{1:n}$, a complete solution to the probabilistic inference problem is given by the a posteriori probability $p(\mathbf{x}|\mathbf{z}_{1:n})$ function, that models the probability of the state at time n $\mathbf{x}_n$ given the vector of observations

up to time n. This probability allows to define the optimal state as the mean of the state computed exploiting this probability distribution

$$\hat{\mathbf{x}}_n = E[\mathbf{x}|\mathbf{z}_{1:n}] = \int_{-\infty}^{+\infty} \mathbf{x}_n p(\mathbf{x}|\mathbf{z}_{1:n}) d\mathbf{x}_n \tag{1.1}$$

Once an optimal estimate of the state is defined, a new problem arise: is it possible to efficiently compute the posterior density $p(\mathbf{x}|\mathbf{z}_{1:n})$?

Of course, an optimal solution is obtained using the recursive Bayesian estimation algorithm [2, 3]. This solution is computationally feasible in the case of linear Gaussian system but it must be approximated in the case of more complex non linear systems.

In both the circumstances, the Kalman filter can be used to approach this problem. It is in fact the optimal solution in the Minimum Mean Squared Error (MMSE), Maximum Likelihood (ML) and Maximum A Posteriori (MAP) sense [4] for the recursive Bayesian estimation in the case of linear systems. Concerning the non linear problems, the extended Kalman filter is a useful tool and it has been very popular in the last decades, becoming almost a standard when a non linear problem must be solved.

By the way, it must be noticed that the extended Kalman filter is just a suboptimal implementation of the recursive Bayesian estimation and this characteristic may lead to divergence issues. To solve this issues, other algorithms have been described in the literature, such the unscented Kalman filter [5].

The Kalman filter is a widely known tool [6,7]. It has been invented in 1960 by Rudolf E. Kalman, despite earlier papers of different authors exists. It is an adaptive filter that has been used in many fields through the years and it has found a tailored field of application in navigation.

1.3 Motivation of the Work

The main objective of this thesis is to discuss the Kalman filter theory and its applications in the field of navigation. In particular, we will consider its application to the fields of satellites based and inertial navigation.

Nowadays, the satellite based positioning and navigation is a service which is available to a large number of people and it is useful in many different fields related to navigation. The estimation of the PVT related to the user is based on pseudorange and pseudorange rate measurements, that are properly processed by means of positioning algorithm.

Of course, since these measurements comes from the real world, they are affected by disturbances and they should be properly processed.

Many algorithms were proposed in literature [5, 6, 8], starting from the simple Least Squares (LS) filter, up to the recent particle filters, unscented Kalman filers and so on. Despite the large number of available choices, the Kalman filter can be considered the de facto standard for to the computation of the position and other tasks related to navigation [9].

The Kalman filter has its strengths in its high flexibility and simplicity, that are characteristic of this filter. It allows good performance with a reasonable computational load.

Despite a large number of books and paper related to the application of the Kalman filter to the navigation exists [10–13], it is hard to find a comprehensive coverage of the topic, since many obscure details remain. For this reason, one of the aims of this thesis is to discuss the use of the Kalman filter in a GNSS receiver, highlighting the key points that must be understood in order to achieve a correct implementation.

The application of the Kalman filter to the field of navigation are not only related to the stand alone GNSS receivers. In fact, the Kalman filter is a useful tool to realize integrated systems, i.e. a combination of at least two different navigation systems, addressed to obtain a more robust and precise estimate of the position of the user.

One of the possible solutions is to design a system that integrates an INS with a GNSS receiver [14]. There are many reasons to design such a system:

- higher solution rates
- improved position estimates, obtained as a proper weighting of the available solutions
- robustness to GNSS interference, jamming and spoofing

An integrated system is useful due to the vulnerabilities of a GNSS receiver. It is in fact well known that the reception of GNSS signals can be affected by Radio Frequency Interference (RFI) that can partially or totally compromise the correct behavior of the receiver. Unintentional interfering signals such as harmonics generated by Digital Video Brodcasting (DVB) transmitters or intentional one, as intentional high power signals generated by jammers, can dramatically compromise the navigation data access control mechanism. Intentional disturbs can also be used to inhibit the tracking stage in order to force the receiver status in the re-acquisition mode for spoofing purposes.

In these cases, the INS provides the system with a solution that cannot be affected by external sources and whose degree of reliability can be compared with the one of the GNSS measurements, in order to provide an estimate of the position that is a properly weighted mean of the two available estimates. The Kalman filter is the tool that allows for an easy weighting of the input solutions.

In this thesis, integrated systems are discussed and analyzed by means of a mathematical descriptions, simulations and real data analysis.

Another objective of this thesis is to describe a new application of the Kalman filter. In fact, the integration with INS is not the only possible way to exploit external measurement. Many different kinds of measurements, obtained by different sensors, can be integrated using a Kalman filter.

One of the topic that is attracting a growing attention in the recent years is represented by the cooperative positioning [15, 16]. Many communication technologies such as Wireless Access in Vehicular Environment (WAVE) and Ultra Wide Band (UWB) were developed in order to allow vehicles and pedestrians to communicate using radio interfaces. A huge number of services may be implemented exploiting these link.

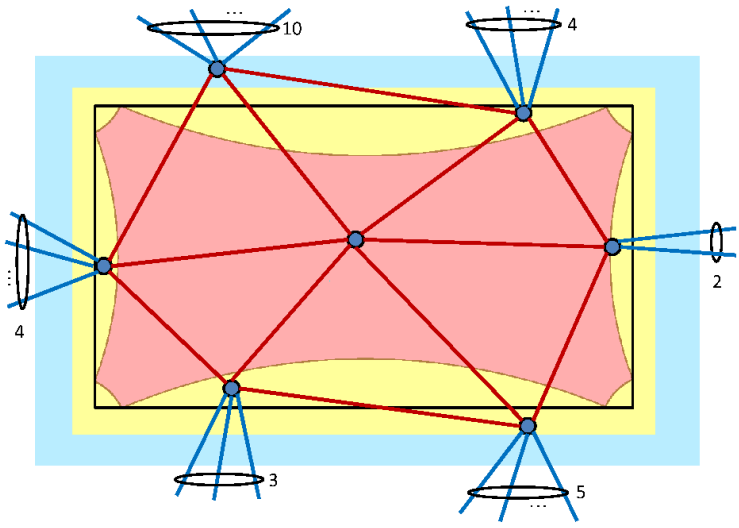

Figure 1.4. P2P network and satellite visibility of the peers.

In this thesis, we are interested in a cooperative positioning system that relies on the existence of these connections and on the positioning capabilities offered by a GNSS, as

shown in Figure 1.4. In particular, thanks to the support of the ESA, we had the chance to develop two algorithms that exploits external measurements available thanks to a P2P network.

These algorithms are based on the availability of an external altitude measurement provided by a certain number of near peers. This measurement is processed together with GNSS measurements in order to obtain a better solution than the one obtained by the stand-alone GNSS receiver. Of course, also these algorithms are based on a Kalman filter and their strong point has to be found in the small modifications that must be made to the stand-alone algorithm.

1.4 Dissertation Outline

This dissertation contains seven chapters, organized as described below.

Chapter 1 presents the motivation, the objectives and the contributions of this dissertation to the theory of the Kalman filter applied to the field of satellite based navigation.

Chapter 2 introduces the reader to the topic of satellite based navigation, with a particular reference to the already existent GPS, which is the system considered in every data collection, since it is the only one available.

Chapter 3 is devoted to the description of the INS architecture and to introduce the navigation equations, that are used to obtain the position of the user starting from the measured accelerations and angular rates.

Chapter 4 is focused on the discussion of the Kalman filter theory, introducing the idea of state space equation and Kalman gain. Moreover, particular filters useful to deal with non linear problems are introduced.

Chapter 5 deals with the GNSS data processing, considering the GPS. The Kalman filter is here used to estimate the PVT of the user starting from the pseudorange and pseudorange rate measurements, provided by the tracking loops. Then the integration between INS and GNSS is considered, showing that it is an evolution of the Kalman filter used in the stand alone GNSS receiver.

Chapter 6 introduces the possibility to integrate the measurement available thanks to a GNSS receiver with other measurement that are available thanks to the presence of a P2P network. In particular, the possibility of an external altitude aiding is discussed and analyzed in details. Moreover, the possibility to exploit external pseudorange measurements is described.

Finally, Chapter 7 addresses the conclusions of this work, summarizes the achieved results and analyzes the possible future development of the considered topics.

Chapter 2

Global Navigation Satellite Systems

The satellites that belong to a GNSS are demanded to broadcast signals to allow the users equipped with a GNSS receiver to compute an estimate of their own PVT. For a matter of ease of computation, this estimate is usually computed in the Earth-Centered Earth-fixed (ECEF) reference system.

In a satellite systems like the GPS, the user is enabled to compute its own PVT by means of Time of Arrival (TOA) measurements: the receiver determines the position of the user evaluating the time interval between the transmission of the signal and its reception.

Such a system sets two different problems: the time synchronization at the system level and the time synchronization at the user level. We will briefly analyze these problems in the following.

Since the TOA measurements are related to time, it is required that the satellites in the system must be accurately synchronized with respect to a common time scale. It is then necessary to design an infrastructure that satisfies this request and to guarantee a precise measurement of time. In the case of the GPS, this issues has been solved as follows:

- a control segment made of a master control station, an alternate master control station and a host of dedicated and shared ground antennas and monitor stations exists;

- each satellite is equipped with a very accurate atomic clock. This solution was already adopted in Timation.

The adoption of the atomic clock on board of the satellites guaranties very low clock

drifts. Nonetheless, the clocks of the satellites may be biased with respect to the time scale and this bias affects the TOA measurements. For this reason, the ground network is demanded to upload to the satellites the correction for their offsets, which are then broadcast to the user in the navigation message.

On the user side, if the clock of the receiver were synchronized with the GNSS time, an estimate of the position is obtained by means of a triangulation:

- the satellite embeds the transmission time in the transmitted signal;
- the receiver compares the transmission time with the TOA and computes the propagation time;
- the propagation time is linked to the geometrical distance between the user and the satellites by a factor equal to the speed of light

$$r = c(T_{rx} - T_{tx}) \tag{2.1}$$

where T_{rx} is the system time at which the signal is received by the user, T_{tx} is the system time at which the signal was broadcast by the satellite and c is the speed of light.

In a three dimensional space, the range r_i between the user and the i-th satellites describe a sphere centered on the known position $\mathbf{p}_{s,i} = [x_{s,i}, y_{s,i}, z_{s,i}]^T$ of the i-th satellite and on whose surface the user lies. It is possible to compute a very precise point that represents the true position of the user $\mathbf{p}_u = [x_u, y_u, z_u]$, setting a system of at least three equations of this kind:

$$r_i = ||\mathbf{p}_{s,i} - \mathbf{p}_u|| \tag{2.2}$$

In a GNSS, the clock of the receiver is not synchronized to the system time. Synchronization can be performed only when the signal is acquired and tracked and the navigation message demodulated. A perfect synchronization with the system time is unfeasible, so that the range measurement is always affected by an error in time, i.e. a bias.

It is then clear that in a GNSS system the TOA measurements are related to a quantity similar to geometrical ranges and named *pseudorange*, which is affected by the misalignment of the clock of the receiver. This new measurement ρ_i between the user and the i-th satellite can be defined as the sum of the true range r_i and of an error related to the misalignment τ that affects the clock of the receiver:

$$\rho_i = r_i + c\tau \tag{2.3}$$

Since the clock misalignment τ that affects every TOA measurement is the same, it is possible to estimate it using the GNSS signal. Therefore, in a system where the receiver clock is not synchronized with system time, the number of unknowns is given by the three user position coordinates and by τ, leading to a four dimensional vector of unknowns. At least four satellites are then necessary to obtain an estimate of the position and time of the user, as shown in Figure 2.1.

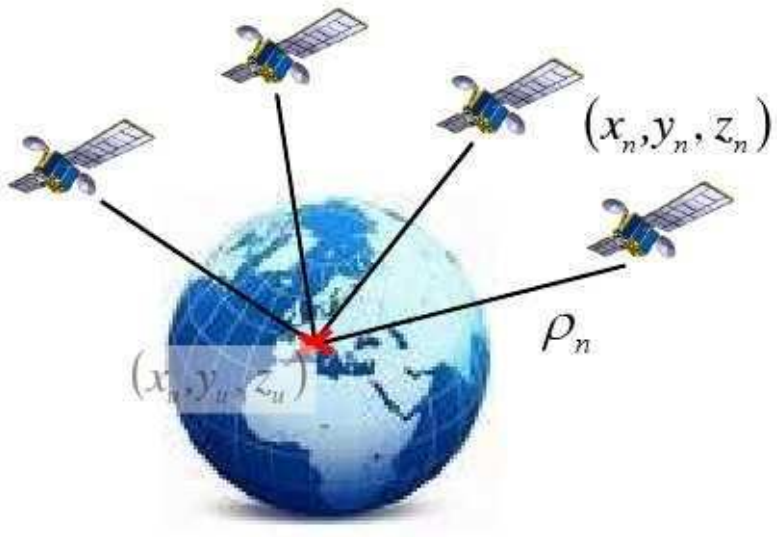

Figure 2.1. Satellites based positioning.

A system with the minimum number of equations assumes the form given in (2.4) [17].

$$\begin{cases} |\rho_1| = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + c\tau \\ |\rho_2| = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + c\tau \\ |\rho_3| = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + c\tau \\ |\rho_4| = \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + c\tau \end{cases} \tag{2.4}$$

where x_u, y_u, z_u are the user coordinates in the ECEF frame. Since the errors due to propagation of the signal are not considered at this time, the solution of the system of equations (2.4) provides the exact position of the user [8].

It must be noticed that the equations in (2.4) are non linear. The solution of such a problem requires the adoption of and iterative algorithm and the linearization of the equations around the best available estimate of the position of the user.

2.1 Structure of the GPS signal

An overview of the GPS signal is necessary for a clean understanding of this work. The signals that are transmitted by the satellites of the GPS constellation and that are actually used for positioning nowadays are the C/A and P codes, that are transmitted on the L1 and L2 bands. These are considered as "current" signals.

On the other side, there is a group of signals belonging to the modernization phase, namely the L2C, L5 and M code, classified as "in evolution". In fact, though there are some satellites that are already broadcasting L2C and M code signals, the modernization phase is still under development and its completion is foreseen for the next years. Moreover there is the L1C signal, which is under design and standardization.

The GPS and these new signals will share the spectrum with the other GNSSs. Table 2.1 outlines the frequency plan for the GPS and Galileo.

Table 2.1. GPS and Galileo bands

System	Band	Bandwidth (MHz)	Center Frequency (MHz)
GPS	L5	24 [1164-1188]	1176.45
	L2	20 [1217-1230]	1227.60
	L1	24 [1563-1587]	1575.42
Galileo	E5a	27 [1164-1191.795]	1176.45
	E5b	25 [1191.795-1217]	1207.14
	E6	40 [1260-1300]	1278.75
	E1	32 [15559-1591]	1575.42

Three bands are allocated for each system: L1, L2, and L5 are for GPS, while, E1, E6, and E5 (E5a and E5b) are for Galileo. Portions of bands are shared between the two systems.

Within the different phases of the GPS system, different satellite blocks have been developed [8]:

- the initial concept validation satellites were called Block I. Ten prototype satellites have been launched between 1978 and 1985;

- 24 satellites of Block II/IIA have been launched between 1989 and 1995, when the system was declared operational. Block II satellites are the initial production satellites, while Block IIA refers to upgraded production satellites;

- block IIR satellites (R stands for replenishment) entered in service in 1997;
- modified Block IIR versions, denoted as Block IIR-M, started in 2005;
- block IIF (F stands for follow-on) is now in production and the first launches started in 2010;
- block III satellites are in the planning stage for a post-2010 deployment. Since these satellites are launched only as replacement for a satellite failure, their schedule is difficult to predict.

The current GPS constellation consists of 31 satellites, distributed ad indicated by Table 2.2.

Table 2.2. Current GPS constellation

	Block IIA	Block IIR	Block IIR-M	Total
n° satellites	14	12	5	31

Figure 2.2 provides a sketch of the current and foreseen GPS signals spectra, pointing out the satellites block they belong to.

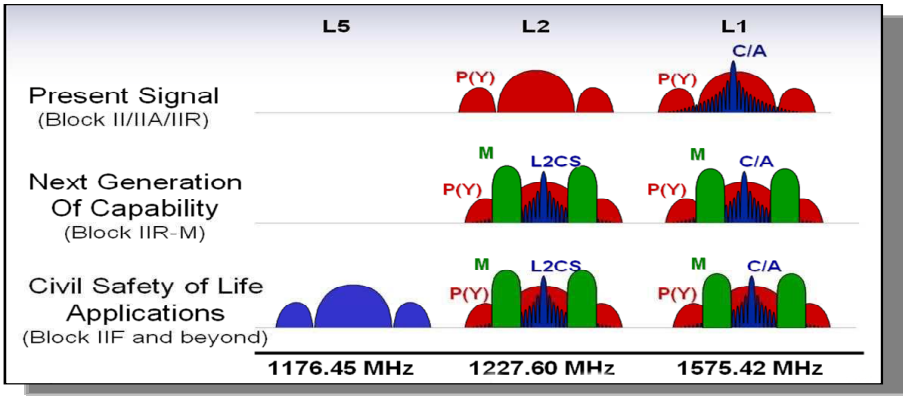

Figure 2.2. Spectra of the current and planned GPS signals.

Currently each GPS satellite transmits three signal on two different bands. A civil signal is transmitted on the L1 carrier, while a military signal is transmitted both on the L1 and L2 bands and it is reserved to DoD-authorized users [18].

The modernization of the GPS signal started in the late 1990s and it is still going on, also driven by the potential competition of the other satellites systems that are being developed. The plans for the modernization of the GPS cover both civil and military fields.

Table 2.3 summarizes all the time-domain and spectral characteristics of the current and future GPS signals.

Table 2.3. Current and future GPS signals (from [19] [20] [21] [18] [22] [23])

Band/Signal	Service	Modulation Scheme	Code Rate [Mcps]	Data Rate [bps]	Central Frequency	Spectral Occupation$^{(*)}$ [MHz]	Min Rx Power [dBW]
L1	SPS	BPSK	C/A 1.023	50	1575.42	2.046	-158.5
	PPS	BPSK	P 10.23	50	1575.42	10.23	-161.5
L2	SPS	BPSK	C/A 0.5115	50	1227.60	1.023	-158.5
	PPS	BPSK	P 5.115	50	1227.60	10.23	-161.5
L5	SPS	BPSK	10.23	25	1176.45	24	-157.9
L2/L2C	Civil	BPSK	CM 511.5E3	50$^{(**)}$	1227.60	2.046	-160
			CL 511.5E3	no data			
L5	Civil	QPSK	I5 10.23E6	50	1176.45	20.46	-154.5
			Q5 10.23E6	no data			
L1-L2/ M code	Military	BOC(10,5)	c.g.$^{(***)}$	N/A	1575.42 1227.60	30.69	-158
L1/L1C	Civil	BOC(1,1) or TMBOC	C_P 1.023E6	no data	1575.42	4.092	-157
			C_D 1.023E6	50 or 75			
			C_O 100	no data			

$^{(*)}$ referring to the main lobe/lobes
$^{(**)}$ the data rate includes the FEC
$^{(***)}$ cryptographically generated

Two new civil signals are defined: a signal on L2, called L2C, and a signal on L5, called L5 [23]. The M code is the military code, properly designed to have sufficient isolation to prevent mutual interference. In addition the United States is planning to add a modernized civil signal upon the L1 frequency within the Block III time frame. The design of this new signal, referred to as L1C, is still under development. As far as the two civil signals L2C and L5, which belong to the Block IIR-M and IIF respectively, they are foreseen to be fully operational in 2011 (L2C) and 2015 (L5) as it is evaluated in [22].

2.2 General Architecture for a GNSS Receiver

A generic GNSS signal is received by means of an antenna and it is then processed by the different stages of a GNSS receiver. This Section aims at describing the basic architecture of a GNSS receives, considering the main operations. The literature related to this field is wide and several books are devoted to this topic. Among them, detailed explanations on the receiver operation can be found in [8] [17] [18] [24]. In Figure 2.3 a simplified scheme of the first stages of a GNSS receiver is shown.

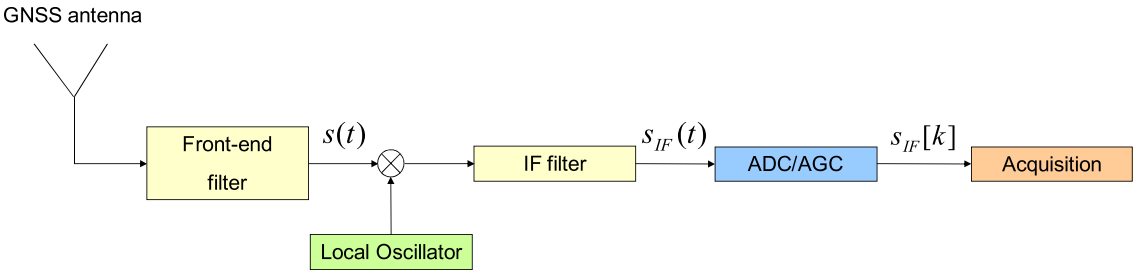

Figure 2.3. First functional blocks of a GNSS receiver.

2.2.1 Front-end Stage

The received signal at the GNSS antenna is composed by the sum of all the received waveforms broadcast by each one of the N_s satellites in view at the time of measurement and it is affected by noise and other disturbing signals.

The front-end is the first stage demanded to process the signal. It is made up of four different functional blocks, namely the Radio Frequency (RF) stages, a local oscillator emitting an Intermediate Frequency (IF), an IF filter characterized by the bandwidth B_{IF}, which is related to the bandwidth of the considered GNSS signal and an Analog to Digital Converter (ADC).

The main aim of this stage is to obtain a signal characterized by a frequency which is lower than the one that characterize the GNSS signal (i.e. for the GPS L1 C/A code $f = 1575.42$MHz) and to filter out all the components that are outside of the bandwidth of the signal. The filtered signal $y_{IF}(t)$ is still composed by the sum of N_s waveforms and it can be modeled as in [18]

$$s_{IF}(t) = \sum_{g=1}^{N_s} d_g(t) + \eta_{IF}(t) \tag{2.5}$$

where $d_g(t)$ is the useful GNSS signal associated to the g-th satellite and $\eta_{IF}(t)$ is the additive Gaussian noise, characterized by a flat $N_0/2$ valued Power Spectral Density (PSD) inside the front-end bandwidth B_{IF}. In particular, it is supposed to be obtained by an independent and identically distributed (i.i.d) random process convoluted with the impulse response of the front-end filter.

The signal $d_g(t)$ can be written in the following form

$$d_g(t) = \sqrt{2C_g}c_g(t-\tau_g)b_g(t-\tau_g)\cos(2\pi(f_{IF}+f_{D_g})t+\phi_g) \tag{2.6}$$

where

- C_g is the power of the g-th signal;
- $c_g(t)$ is the spreading code emitted by the g-th satellite. Each satellite is characterized by a Gold code, in order to obtain a Code Division Multiple Access (CDMA) system where each signal is orthogonal to the others;
- τ_g is the delay introduced by the communication channel;
- $b_g(t)$ is the bitstream of the navigation message.
- f_{IF} is the intermediate frequency;
- f_{D_g} is the unknown Doppler frequency of the received signal;
- ϕ_g is a carrier phase, generally modeled as a random variable uniformly distributed in the range $[-\pi,\pi]$.

The signal in (2.6) is still in the continuous time domain. The ADC is demanded to transform the IF signal in a sequence of samples at frequency f_s. For the A/D conversion, both single-bit and multi-bit architectures are currently in use. Most low-cost commercial receivers employ 1 bit sampling in narrow (i.e. 2MHz) bandwidths. High-end receivers typically use anywhere from 1.5 bit (i.e. three levels) to 3 bit (eight levels) sampling in bandwidth range from 2 to 20MHz.

At the output of the ADC, the signal (2.5) assumes the following form

$$s_{IF}[n] = s_{IF}(nT_s) = \sum_{g=1}^{N_s} r_{IF}[n] + \eta_{IF}[n] \tag{2.7}$$

where $\eta_{IF}[n]$ is the sampled noise, $T_s = 1/f_s$ is the sampling period and n is the discrete time. In (2.7) the quantization effect is neglected. The term $r_{IF}[n]$ can be written as

$$r_{IF}[n] = \sqrt{2C_g}c_g[nT_s - \tau_g]b_g[nT_s - \tau_g]\cos(2\pi(f_{IF} + f_{D_g})nT_s + \phi_g) \tag{2.8}$$

Notice that both the phase ϕ_g and the delay τ_g include an additional contribution of phase and delay, due to the front-end filters.

Due to the orthogonality property of the GNSS codes the N_s received signals can be analyzed separately. Hence, without loss of generality, the sum in (2.7) is considered for $N_s = 1$ hereinafter, and the subscript g will be omitted for simplicity, when possible.

At the output of the ADC stage the variance of the filtered IF Gaussian noise $\eta_{IF}[n]$ can be written as

$$\sigma^2_{IF} = E[\eta^2_{IF}[n]] = E[\eta^2_{IF}(nT_s)] = N_0 B_{IF} \tag{2.9}$$

In most cases $f_s = 2B_{IF}$, therefore the variance can be written as

$$\sigma^2_{IF} = \frac{N_0 f_s F_{sh}}{2} \tag{2.10}$$

where a term F_{sh} is included to take into account the shaping of the filter transfer function. In this case the output noise will result colored.

The output of the ADC is then fed to a certain number of channels that identifies the component of the signal related to a specific satellites and tracks it. The first operation executed by each channel is the acquisition.

2.2.2 Acquisition Stage

All the operations described in Subsection 2.2.1 must be performed in hardware. After the ADC, the digital samples can be processed by dedicated hardware or by means of a Software Defined Radio (SDR) GNSS receiver.

Independently from this implementation choice, the digital samples must be fed to the acquisition block. In Figure 2.4 a serial acquisition scheme is shown.

Modern acquisition strategies are based on the Fast Fourier Transform (FFT), which might speed up the acquisition phase for some receiver architectures. Using the FFT approach when the match between the local and the incoming codes is found, the whole frequency range of the search space is scanned and the spectrum shows a spike corresponding to frequency of the incoming carrier. The use of modern frequency domain search techniques for rapid acquisition is described in [25].

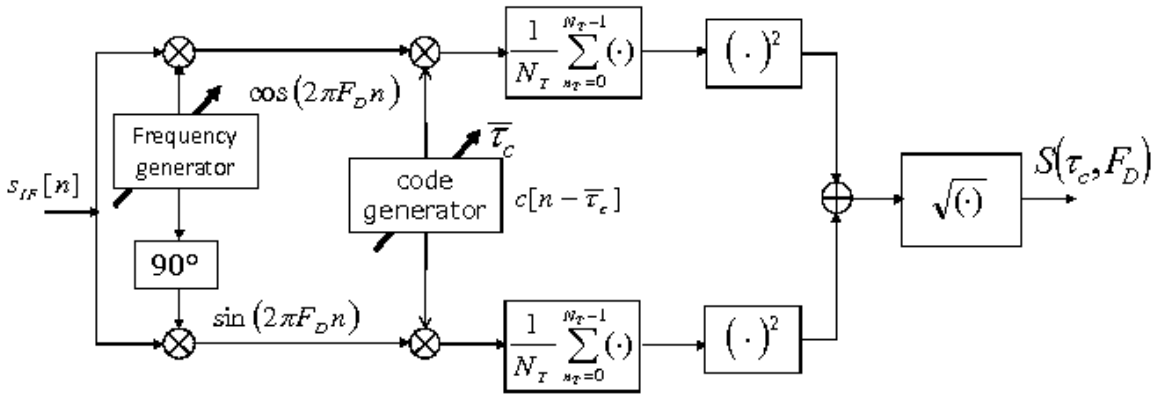

Figure 2.4. Operative scheme of the acquisition unit. The low-pass filters after the cosine/sine multiplications are omitted, since the coherent integration blocks already act like low-pass filters.

The main objective of the acquisition is to provide the tracking stages with a coarse estimate of the Doppler frequency and of the code delay that affect the received signal. To this purpose, a two dimensional function $S(\tau, f_D)$ of code delay and Doppler frequency, usually called Complex-Ambiguity Function (CAF), is computed for a discrete set of frequency and delay values, namely the search space. The spacing between the frequency points (bins) is usually in the order of the hundreds of Hertz, while the code delay is considered with a spacing of one chip.

The first operation is the conversion of the signal from IF to baseband: the signal is projected on two orthogonal sinusoids characterized by the frequency $F_d = (f_{IF} + \bar{f}_D)T_s$, where $\bar{f}_D$ is a test Doppler frequency, which spans a frequency range which depends on the specific application. For example, the range $\pm$5kHz is used in terrestrial receivers.

Then the signal is coherently correlated by a local GNSS code delayed by a test delay $\bar{\tau}_c$, which spans the code period (the Pseudo-Random Noise (PRN) code of the g-th satellite is here considered).

The in-phase and quadrature branches are then squared, summed and square-rooted, to obtain the modulus of the incoming signal, which is now independent from its phase component and is equal to the CAF.

Due to the orthogonality of the Gold sequences and of the sinusoidal carriers, it is evident that in the noise-free case the CAF reaches a maximum peak in the 2D plane when $F_d = (f_{IF} + f_{D_g})T_s$ and $\bar{\tau}_c = \tau_g$. In the noisy case, the position of this maximum is used as a first rough estimation of the Doppler and time delay of the received signal. This information is used by the subsequent receiver stages to maintain the signal locked

to the local code during the data demodulation.

2.2.3 Tracking Stage

In the acquisition block, estimates of the Doppler frequency $\widehat{f_d}$ and code phase $\widehat{\tau_c}$ are provided by correlations among the in-phase and quadrature components of the incoming signal and of a GNSS code local replica. These estimates are not refined: in fact the estimate of the frequency is affected by an error which is not larger than half of the size of the frequency bean, while the code delay is affected by an error which is less than one chip. Due to these errors, a refinement stage, that is also demanded to keep the receiver in the tracking state, is required.

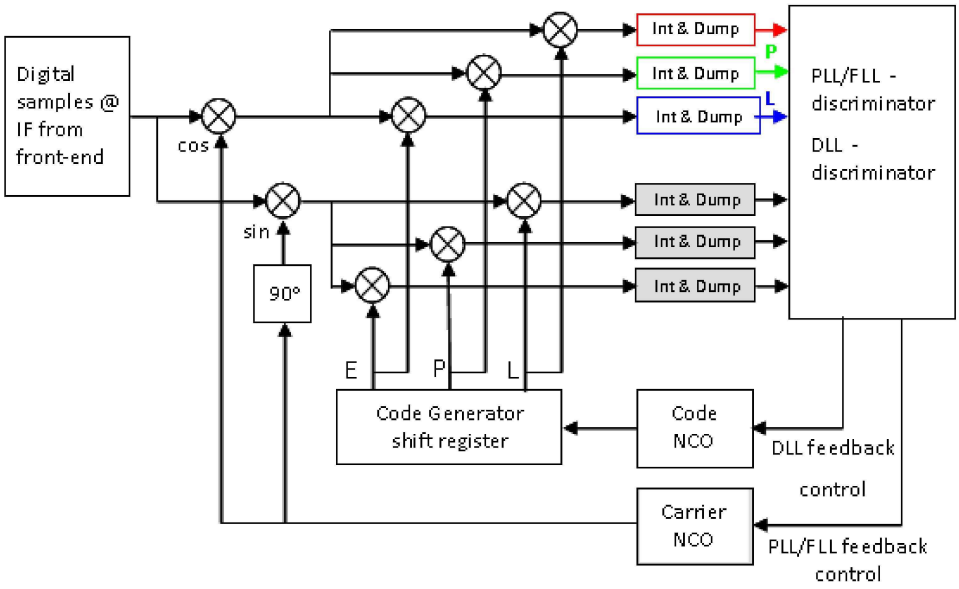

Figure 2.5. Interaction between a DLL and a PLL.

Two different blocks are devoted to this tasks: the Delay Lock Loop (DLL) tracks the code phase, while the Phase Lock Loop (PLL) tracks the total phase of the signal. These blocks are analyzed in the following and a general scheme is shown in Figure 2.5, to underline the strong interaction between the two devices.

Delay Lock Loops

In Figure 2.6 a tracking stage featuring a DLL is represented.

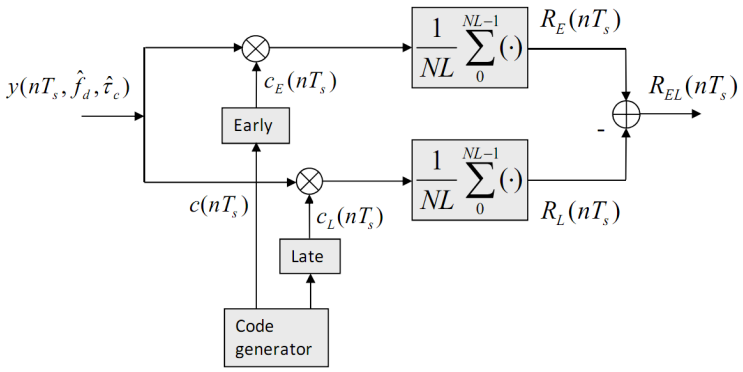

Figure 2.6. Delay Lock Loop in open loop configuration.

The input of the DLL block can be modeled as

$$y(nT_s) = \sum_{g}^{N_s} x_g(nT_s) + \eta(nT_s) \tag{2.11}$$

where $x_g(nT_s)$ is the useful signal received from the g-th satellite after the acquisition stage, and $\eta(nT_s)$ represents the additive Gaussian noise with PSD equal to $N_0/2$, assumed flat inside the front-end bandwidth B_{IF}. The signal $x_g(nT_s)$ can be defined as [18]

$$x_g(nT_s) = \sqrt{2C_g}c_g(nT_s - \widehat{\tau_g})b_g(nT_s - \widehat{\tau_g})\mathrm{e}^{j(2\pi\Delta f_{d_g} nT_s + \phi_g)} \tag{2.12}$$

where

- C_g is the received power of the g-th satellite
- $c_g(nT_s)$ is the spreading code of the g-th satellite
- $b_g(nT_s)$ represents the bit stream of the navigation message
- $\widehat{\tau_g}$ is the estimation of the delay τ_g obtained by the acquisition stage of the g-th communication channel
- ϕ_g is the carrier phase
- Δf_{d_g} includes the residual difference between the unknown Doppler frequency f_{d_g} and the estimated one $\widehat{f}_{d_g}$ performed by the acquisition stage or by the PLL

$$\Delta f_{d_g} = f_d - \widehat{f}_{d_g} \tag{2.13}$$

As for the acquisition model, due to the orthogonal properties of the GNSS codes (negligible cross-correlation values), each signal can be analyzed separately. For this reason, hereinafter the case of a single satellite is considered and the g subscript will be omitted.

In a traditional GNSS receiver architecture, the DLL refines the estimation of the code delay comparing the Early, Prompt and Late replicas of the considered code. Moreover, the output of the Prompt branch is fed to the PLL.

For a matter of simplicity, the DLL block scheme considered in Figure 2.6 is the one in the open configuration, where only the discriminator function is evaluated. In this configuration, The DLL is not linked to the PLL and a perfect estimate of the phase $\widehat{\phi}$ is assumed.

The main goal of the open DLL block is to provide the discrimination function $R_{EL}(nT_s)$ obtained by the correlations calculated over the coherent time $T_{co} = NLT_s$, where N is the number of code periods to be integrated and L is the code length in samples. The incoming signal is correlated with the code $c(nT_s)$ and the delayed (Late) $c_L(nT_s)$ and the anticipated (Early) $c_E(nT_s)$ replicas, defined as

$$c_E(nT_s) = c\left(nT_s + \frac{\Delta}{2}\right) \tag{2.14}$$

$$c_L(nT_s) = c\left(nT_s - \frac{\Delta}{2}\right) \tag{2.15}$$

where Δ is the spacing of the discriminator.

Phase Lock Loops

A PLL is a negative feedback control system and its aim is to track the total phase $\theta[n]$ of a carrier, i.e. the frequency f_n and the instantaneous phase ϕ_n of a sinusoid. It is made of three main blocks, namely the Phase Detector (PD), the Low Pass Filter (LPF) and the Numerically Controlled Oscillator (NCO). In the following we will analyze these blocks, referring to Figure 2.7.

The output signal of the PD has to be related to the difference between the input phase $\theta[n] = 2\pi f_n nT_s + \phi_n$ and the estimated phase $\hat{\theta}[n] = 2\pi \hat{f}_n nT_s + \hat{\phi}_n$. At first the PLL computes the correlation of the input signal with a local sinusoidal carrier and its $\pi/2$ back shifted version, getting the in-phase and quadrature components, respectively.

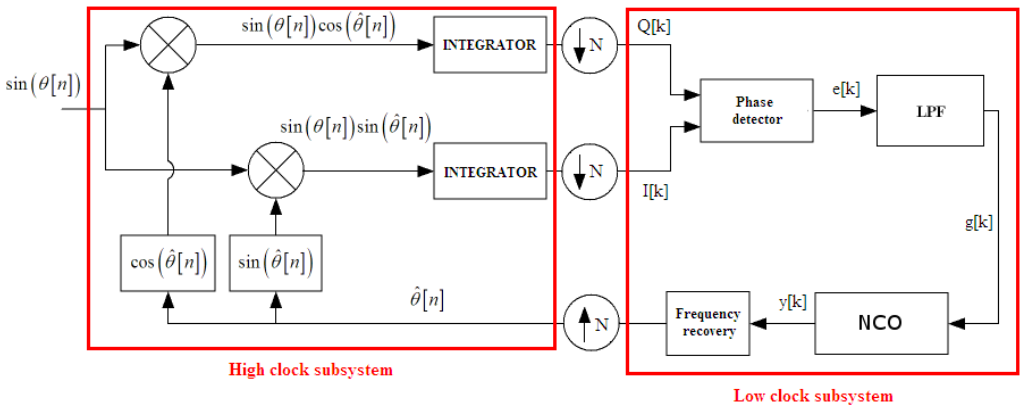

Figure 2.7. Phase Lock Loop in open loop configuration.

Using Werner formulas, for the quadrature branch one obtains

$$\sin\theta[n]\cos\hat{\theta}[n] = \frac{1}{2}[\sin(\theta[n]-\hat{\theta}[n]) + \sin(\theta[n]+\hat{\theta}[n])] \tag{2.16}$$
$$= \frac{1}{2}[\sin(2\pi(f_n-\hat{f}_n)nT_s + \phi_n - \hat{\phi}_n) - \sin(2\pi(f_n+\hat{f}_n)nT_s + \phi_n + \hat{\phi}_n)]$$

and similarly for the in-phase branch

$$\sin\theta[n]\sin\hat{\theta}[n] = \frac{1}{2}[\cos(\theta[n]-\hat{\theta}[n]) + \cos(\theta[n]+\hat{\theta}[n])] \tag{2.17}$$
$$= \frac{1}{2}[\cos(2\pi(f_n-\hat{f}_n)nT_s + \phi_n - \hat{\phi}_n) - \cos(2\pi(f_n+\hat{f}_n)nT_s + \phi_n + \hat{\phi}_n)]$$

Both the in-phase and the quadrature components are both characterized by a low frequency component and a high frequency component, given by the difference and the sum of the input and local carrier total phases. These values are accumulated in the quadrature and in-phase integrators for a whole integration window $T_f = NT_s$, where N is the number of samples included in an integration interval. At this stage the outputted values are low pass filtered due to the action of the integrators. Considering a tight gap between the estimated frequency and the actual frequency and if the high frequency component has been filtered out by the integrator, the quadrature and in-phase components can be described as follows

$$Q[k] \approx N^2 2\pi\Delta f[k]\frac{T_s}{2} \tag{2.18}$$

$$I[k] \approx N \tag{2.19}$$

where $\Delta f[k] = f_n - \hat{f}_n$, with $k = \lfloor n/N \rfloor$ and $\Delta\phi_n = \phi_n - \hat{\phi}_n \approx 0$. This notation implies that $\Delta f[k]$ is fairly constant in the integration window T_f.

The computed correlation values are the inputs of the PD, that has to output the error signal

$$e[k] = \tan^{-1} \frac{Q}{I} \tag{2.20}$$

which is a function of the phase difference between the quadrature and the in-phase components. It can be shown that the arc tangent discriminator represents is the ML estimator [26]. Nonetheless there are other choices besides the arc tangent discriminator.

The LPF is then fed with the error signal $e[k]$. Its role is to suppress the high frequency component in the error signal due to noise and to output the signal $g[k]$, which is the control signal for the NCO.

The NCO is demanded to tune the oscillator phase to be synchronous with the input carrier phase. The z domain model for the NCO is the one of an accumulator

$$H(z) = \frac{z^{-1}}{(1 - z^{-1})} \tag{2.21}$$

so that the output is $y[k] = y[k-1] + g[k-1]$. $H(z)$ has a delay z^{-1} for the system to be causal, so the output $y[k]$ is the phase difference that has to be compensated at time k and that has been computed using the date of the integration interval at time $k-1$. The difference quotient of the output of the NCO is the value needed to perform the frequency correction around the centre frequency of the oscillator.

Working Ranges of PLLs and False Lock

The operation of a PLL is characterized by the existence of different states. These states are determined by the difference between the input frequency f_n and the frequency $\hat{f}_n$ estimated by the NCO.

When a PLL is operating correctly, it is phase locked. If one considers the beginning of the operation, there is a maximum tolerable mismatch between the input and estimated frequency $|f_n - \hat{f}_n|$ for the PLL to get in the phase lock state. This range of frequency is called pull-in range. There is also a narrower range of frequencies where the PLL manages to get in the phase lock state without any cycle slip. In this case, we will talk about the lock-in range of the PLL.

Once the PLL is locked, the NCO must track the input frequency with a certain precision. The maximum allowed difference between the input and the generated frequency to keep the device in the tracking state is called hold-in range. It is quite different from

the pull-out, which is that frequency step which causes a PLL to fall out lock if applied to the reference input of the PLL.

Finally, when the initial frequency of the NCO is set to a frequency value not belonging to the pull-in range, the PLL may lock to a wrong frequency, generating a false-lock event [27]. In this case, the message associated to the signal may appear valid, as shown in Figure 2.8

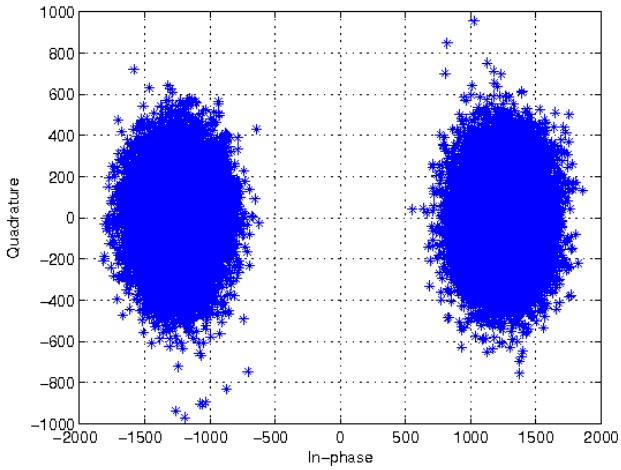

Figure 2.8. Scatter plot for a false-lock event in the presence of a real GPS signal.

By the way, a more accurate analysis in time would show the result in Figure 2.9.

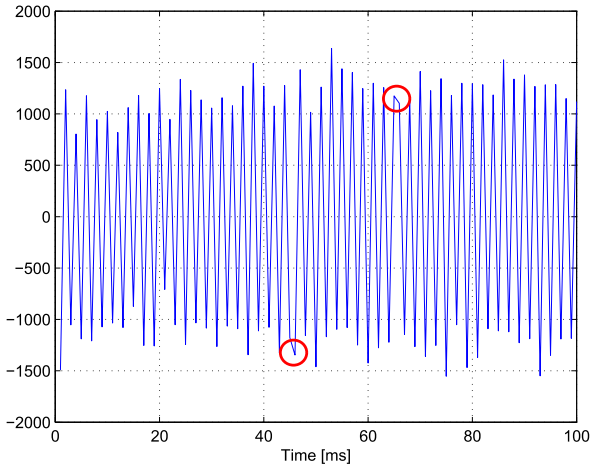

Figure 2.9. Navigation message in the presence of a false lock event.

Since at each millisecond the estimate of the bit is rotated by $\pi/2$, it is evident that the low frequency component in (2.16) and (2.17) is not close to zero, but a residual frequency must be present. In particular, referring to Figure 2.9, the two red circles point out two constant bits that are 20ms far, a time equal to the duration of a bit of the navigation message in the GPS.

In the case of correct operation, the residual total phase would be almost zero and the summation blocks would integrate a sinusoid and a cosinusoid featuring a frequency of a few Hz, ideally zero. The outputs of the correlators would show an almost constant non-zero value in the in-phase branch and an almost constant and close to zero value in the quadrature branch. In this scenario the discriminator output is close to zero and the systems locks to the actual frequency.

Now, let us analyze the case shown in Figures 2.8 and 2.9. In this case the Frequency Lock Loop (FLL) provides the PLL with a frequency 500 Hz apart from the right one, due to a wrong acquisition estimate. The residual phase yields two 500 Hz waves with a 2 ms period, as shown in Figure 2.10.

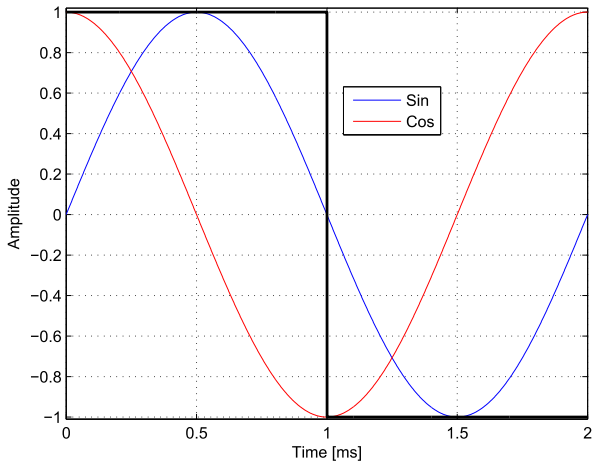

Figure 2.10. Low frequency sinusoids in the case of a 500Hz false lock with 1ms integration time.

The PLL is aware of the whole phase and not only of the frequency, so it will try to make the quadrature branch equal to zero.

The quadrature and in-phase are computed integrating the waves for $T_f = 1$ ms. These two scalar values will be input to the discriminator. In the following we will consider the continuous time domain, but the construction has to be considered true both in

the case of a sampled wave and of a squared wave.

Let us assume the worst case, i.e. when the whole energy is in the quadrature branch. To compute the output values of the correlators, we remember that

$$\frac{1}{T}\int_0^T \sin(2\pi\frac{t}{2T})dt = \frac{2}{\pi}, \tag{2.22}$$

where $2T$ is the wave period and T is the integration time. The quadrature and in-phase values are the input of the phase discriminator block. The output $e\,[k]$ of this device is computed as the arc tangent of the quadrature-in-phase ratio, as follows

$$e\,[k] = \tan^{-1}\frac{Q\,[k]}{I\,[k]} \tag{2.23}$$

In this case, if we consider the low frequency component only and $T/2 = T_f = 1$ ms, the discriminator would behave as follows

$$Q\,[k] = \pm\frac{2}{\pi}, I\,[k] = 0 \rightarrow e\,[k] = \pm\frac{\pi}{2}, \tag{2.24}$$

where the sign of $Q\,[k]$ depends upon the phase of the sinusoid, involving a $\pi/2$ instantaneous phase steering in the NCO generated waves. This correction implies what follows

$$Q\,[k] = \frac{1}{T}\int_0^T \sin(2\pi\frac{t}{T} \pm \frac{\pi}{2})dt = 0, \tag{2.25}$$

$$I\,[k] = \frac{1}{T}\int_0^T \cos(2\pi\frac{t}{T} \pm \frac{\pi}{2})dt = \frac{2}{\pi}, \tag{2.26}$$

being Q and I the quadrature and in-phase values.

It is clear that scenarios where the energy is distributed between the two branches would generate smaller instantaneous phase correction, but the effect would be the same.

Now, in the peculiar case of a 500 Hz frequency shift and of a 1 ms integration period, we can state that the PLL has an harmful degree of freedom in the instantaneous phase, which can be used to null the quadrature branch output and to put the system in a false lock state. The only difference with respect to the "true" lock system is represented by the halved energy of the in-phase branch, due to the residual sinusoid in the signal.

The phenomenon can be easily generalized. The main issue to be noticed is the presence of sinusoids which are integrated for an integer number of periods plus an half of their period, i.e. sinusoids with period $(2T_f)/(2n+1)$, where n is an integer, would behave in the same way of the one we analyzed. In fact in this case the phase steering of

the NCO generated carrier produces a null value in the quadrature branch. For example, if $T_f = 1$ms, we would notice a false lock for residual frequencies equal to ± 500Hz, ± 1500Hz, ± 2500Hz and so on.

We must notice that shorter wave periods would imply lower values for the in-phase branch in case of false lock. In fact, the energy is computed integrating the sinusoid for T_f seconds, but only the uncomplete period of the wave is not nulled by the summation blocks.

It also has to be noticed that sinusoids with shorter periods have to be considered rare, because only a very inaccurate acquisition or peculiar signals might generate such a scenario.

2.3 PLLs: an Insight on Loop Filters and Adaptive Techniques

PLLs are usually designed to use a constant bandwidth. Since this value is chosen considering the maximum dynamic of the user, it is of course non optimal. A better choice would be to adapt the bandwidth on the actual dynamics of the user, in real time [28,29]. In the following a second order loop filter of the Infinite Impulse Response (IIR) type, characterized by two degrees of freedom, is discussed. The filter is shown in Figure 2.11.

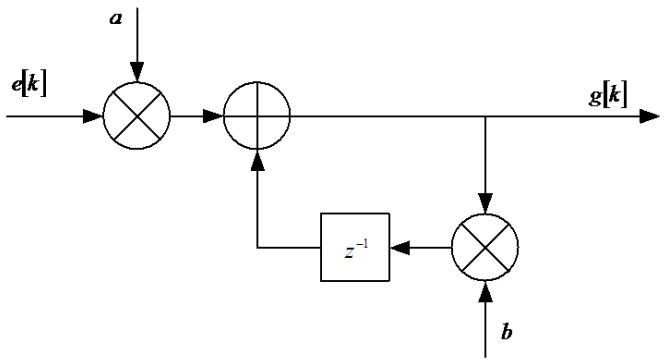

Figure 2.11. IIR loop filter.

Nonetheless, it is possible to implement a Finite Impulse Response (FIR) loop filter to obtain a faster transient response, but in that case it would be necessary to compute and store a certain amount of filter coefficients values in order to obtain a set of desired Equivalent Noise Bandwidth (ENB) values that minimize the transient time. The choice of the IIR loop filter is justified by the simple expressions for the coefficients of the filter

with respect to a desired ENB.

The ENB of a system that is characterized by a closed loop transfer function $H_c(z)$ can be expressed in terms of its transfer function as follows [26]

$$B_{eq} = \frac{1}{2j\pi} \oint_{||z||=1} H_c(z) H_c(z^{-1}) \frac{dz}{z} \tag{2.27}$$

In the case of the adopted IIR loop filter, the closed form to evaluate the equivalent bandwidth is the following

$$B_{eq} T_f = \frac{2(a-b) + b(a+b)}{2b[4-a-b]} \tag{2.28}$$

The value of ENB is crucial, because it is responsible for the error caused by the thermal noise, which is proportional to the ENB, and for the dynamic stress error, which is conversely proportional to the ENB. For a constant bandwidth PLL the ENB is chosen in order to fit the maximum dynamic of the Doppler shift, but such a value might be too conservative for the most of the time, especially when the Doppler shift dynamic is quite slow.

By the way, in the design of a PLL it is usually required to fit the bandwidth requirements and to compute the filter coefficients in order to satisfy this constraint. Nonetheless, there are two degrees of freedom, represented by the two coefficients a and b, so another constraint is needed. This one is represented by the damping factor ζ. In order to obtain the shortest step response, it is quite common to set $\zeta = 0.707$ [30], but other values around this one may fit the system scenario.

Following [31], an auxiliary parameter δ is now defined as follows

$$\delta = \frac{4 B_{eq} T_f}{1 + 4\zeta^2 + 8\zeta^2 B_{eq} T_f} \tag{2.29}$$

so that the two filter coefficients can now be defined as

$$b = 4\zeta^2 \delta a = b(1+\delta) \tag{2.30}$$

In a constant bandwidth PLL, the two loop filter coefficients are kept constant, in order to keep constant the system bandwidth. An adaptive bandwidth PLL instead tracks one or more of the blocks outputs to modify the ENB. The algorithm presented in this paper tracks the estimated variance of the PD output $e[k]$ by means of a circular buffer capable of storing L_{buffer} items. Nominally $\sigma_e^2[k]$ =var$\{E[k]\}$, where $E[k]$ is the set of values that $e[k]$ assumed between $k - L_{buffer} + 1$ and k.

When the PLL is first fed with the frequency value provided by the FLL, it has to recover the error of the FLL. The system starts with a bandwidth $B_{eq,init}$. It will show a transient and then the estimate of the frequency will settle around the actual frequency value with an error which depends on the noise spectral density. During the transient, if the frequency is changing rapidly, the value of the output of the discriminator will change, its variance will be high and its mean will be non zero. Conversely, once the frequency estimate is good and the frequency variations are slow, the variance of the output of the discriminator will be lower and the mean will be near to zero, if the system bandwidth is wide enough to track Doppler variations.

The buffer keeps track of the values of the error signal. Once the buffer is full, the algorithm compares the estimated variance $\sigma_e^2[k]$ and the estimated mean $\mu_e[k]$ of the error signal with two different threshold, namely $T_{\sigma_e^2}$ and T_μ. Variance control is needed in order to make the bandwidth narrower, while mean control is needed in order to widen the bandwidth. Nonetheless, a frequency correction control is implemented, in order to allow the system to recover quick and sudden Doppler velocity variations.

Both the thresholds depend on the current value of the carrier to noise ratio C/N_0, that is continuously estimated by the system and it is used to dynamically set the threshold values. If the estimated variance drops below the variance threshold, the algorithm recognizes that the system is in a steady state and multiply the actual bandwidth by a positive factor b_{down} smaller than one.

$$\sigma_e^2[k] < T_{\sigma_e^2} \rightarrow B_{eq}[k+1] = b_{down} B_{eq}[k] \tag{2.31}$$

The new bandwidth value will be used to compute the new filter coefficients. This choice is justified by the fact that the system manages to track the frequency variations with ease, because the Doppler velocity is very low. Such a scenario does not require a very reactive system and allows to make the bandwidth narrower, in order to cut off a greater amount of noise. If the estimated variance is lower than the threshold, the bandwidth is decreased each time the buffer is full, down to a bottom bandwidth limit that has been set by the user or that can be set in accordance with the estimated carrier to noise ratio.

Conversely, if the estimated error signal mean is higher than the mean threshold, the algorithm recognizes that the system is not in a steady state and that the bandwidth has to be widened by a factor b_{up} greater than one, in order to better track the variations. This control has a higher priority with respect to the variance control, in order to keep the PLL

in its tracking state.

$$\mu_e[k] < T_{\mu_e} \rightarrow B_{eq}[k+1] = b_{up} B_{eq}[k] \tag{2.32}$$

Finally, the highest priority control is the one related to frequency corrections. If the NCO correction at time k, $\Delta f[k]$, differs from the one at time $k-1$, $\Delta f[k-1]$, for a value greater than the ENB, the system bandwidth is reset to the initial value $B_{eq,init}$ and the old frequency correction $\Delta f[k-1]$ is applied to the NCO.

$$\Delta f[k] > B_{eq}[k] \rightarrow B_{eq}[k+1] = B_{eq,init}, \quad \Delta f[k] = \Delta f[k-1] \tag{2.33}$$

This control detects when the system is losing lock and tries to avoid it, resetting the bandwidth of the PLL and the buffer memory quickly. The operational flowchart is shown in Figure 2.12.

2.3.1 Simulations

In order to test the constant bandwidth PLLs and its adaptive counterpart, some simulations were run. The input signal is a pure sinusoid.. The hypothesis is that the DLL is functioning properly, wiping off the Pseudo Random Noise (PRN) code from the signal. The setting is similar to the one that can be found in [26,32]. In the first half of simulation, the signal is characterized by a 50Hz/s Doppler rate, that is almost the Doppler velocity of a user that moves toward a satellite with an acceleration of 1g [17]. Then the frequency variations are kept slower.

Such a test is intended to show that the proposed algorithm outperforms the constant bandwidth PLL at low dynamics, while it performs as well as the traditional PLL for higher dynamics. The signal is characterized by $C/N_0 = 45$dB/Hz, which is a very good carrier to noise ratio.

In Figure 2.13, the trend of the two different estimated frequencies compared to the actual frequencies is shown. During the first half of the simulation time the two PLL behave almost in the same way. In the second half, the actual frequency varies very slowly and the adaptive PLL bandwidth gets smaller, in order to obtain a more accurate estimate of the frequency. The good behavior of the adaptive loop filter will be evident analyzing the estimation error curves.

In Figure 2.14, the trend of the error estimate is shown. While during the high Doppler velocity the estimate errors performed by the two systems are very similar, in the final

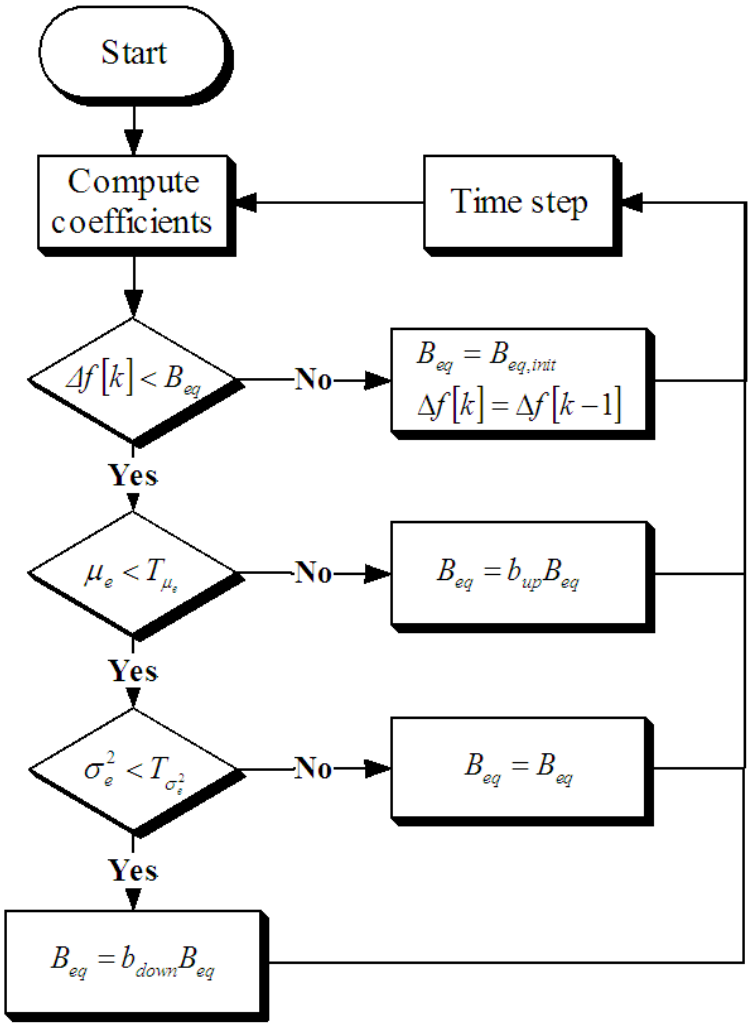

Figure 2.12. Flowchart of the adaptive PLL algorithm.

part of the simulation the adaptive PLL manages to get an error ten times smaller than the constant bandwidth counterpart.

Finally, in Figure 2.15 the trend of the total phase estimate is shown. The total phase accuracy is almost the same, but the notable fact is that the adaptive PLL is affected by low frequency noise, while the constant bandwidth PLL is affected by high frequency noise. This is one of the results of the adoption of narrower bandwidth.

For the sake of completeness, another simulation with a worse carrier to noise ratio is reported. All the parameters are kept constant with respect to the other experiment, except the carrier to noise ratio that characterize signal, set to $C/N_0 = 37$dB/Hz, which

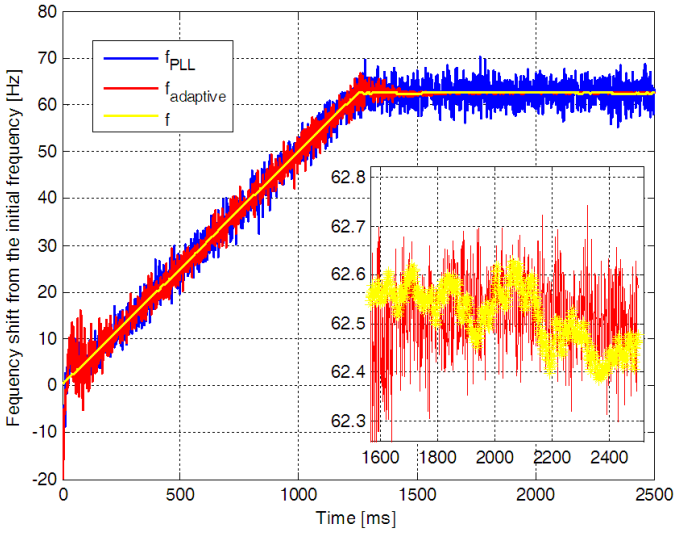

Figure 2.13. Estimated and actual frequencies - high C/N_0.

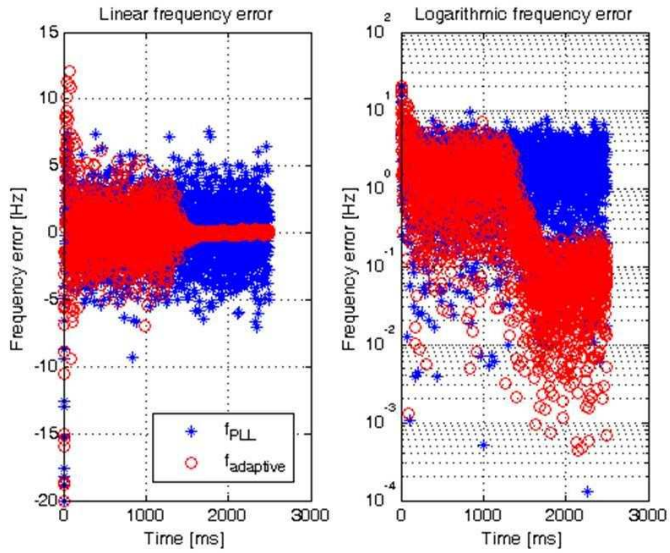

Figure 2.14. Estimated frequency error - high C/N_0.

is a low carrier to noise ratio.

Looking at the frequency estimation errors plots in Figure 2.16 and 2.17, it is evident that the worsening of the signal strength has an impact on the system behavior. The adaptive PLL outperforms anyway the constant bandwidth PLL, but this time the

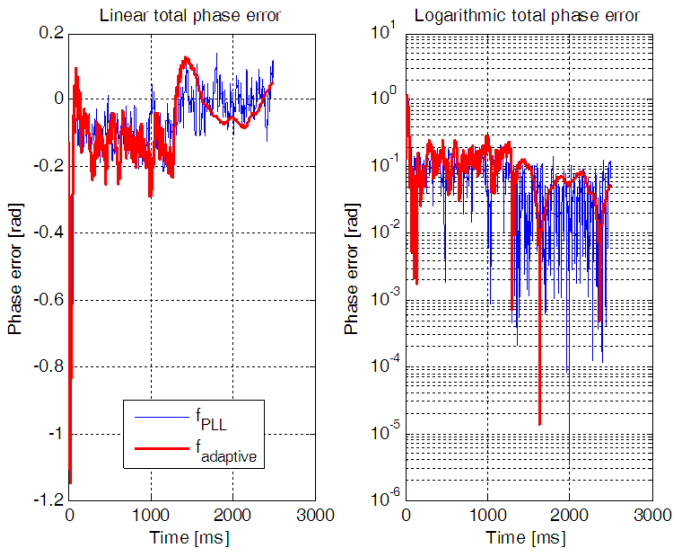

Figure 2.15. Estimated phase error - high C/N_0.

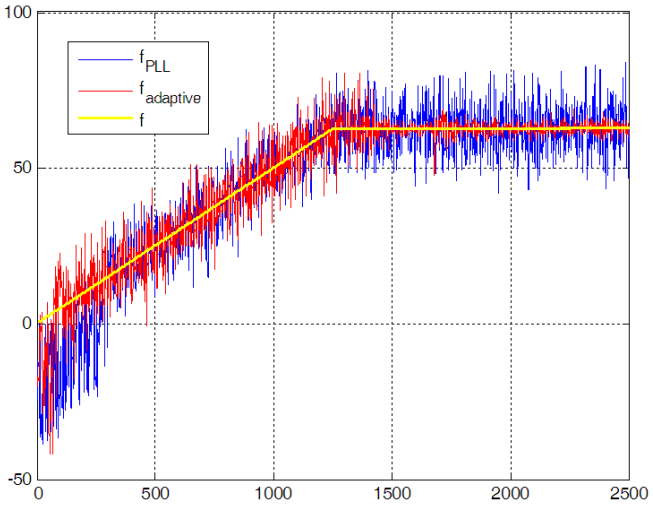

Figure 2.16. Estimated and actual frequencies - low C/N_0.

improvement is less evident, because the adaptive PLL has to limit his bandwidth to a value that is higher than the one of the first experiment.

Moreover, the adaptive PLL manages to reach the steady state faster than the constant bandwidth PLL, thanks to a higher initial bandwidth. This is evident looking at Figure

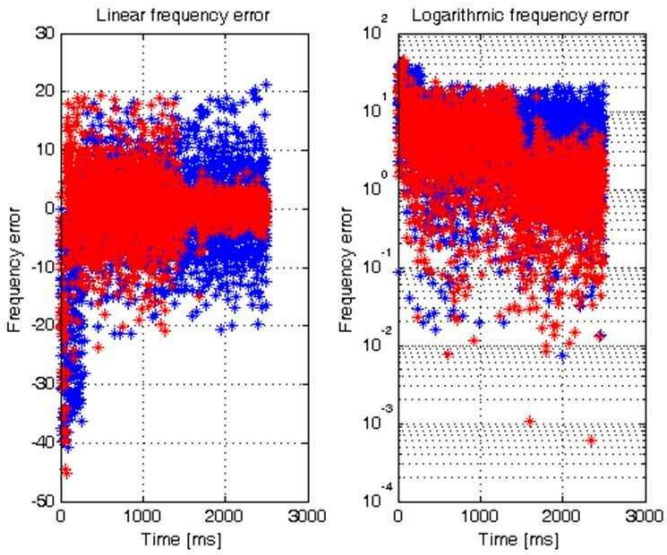

Figure 2.17. Estimated frequency error - low C/N_0.

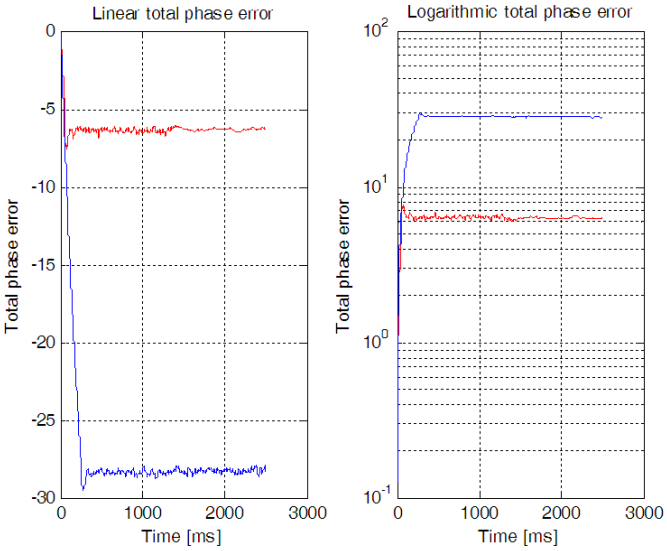

Figure 2.18. Estimated phase - low C/N_0.

2.18. If the system gets in the tracking state faster, the accumulated phase error is lower, due to a smaller number of cycles lost during the transient.

Chapter 3

Inertial Navigation

This Chapter is devoted to the description of INSs and IMUs. These are devices whose operation is related to the laws of classical mechanics as formulated by Isaac Newton:

- **First law**: every body remains in a state of rest or uniform motion unless it is acted upon by an external unbalanced force;
- **Second law**: a body of mass m subject to a force F undergoes an acceleration a that has the same direction as the force and a magnitude that is directly proportional to the force and inversely proportional to the mass, i.e., F = ma;
- **Third law**: the mutual forces of action and reaction between two bodies are equal, opposite and collinear.

If the acceleration vector can be measured, it is possible to compute the change in velocity and position integrating the measurements in time. To this purpose, IMUs are equipped with 3 accelerometers, mounted along mutually perpendicular axes, capable of measuring the acceleration along their own axis.

In order to navigate with respect to a given reference frame, the displacement of the accelerometers with respect to the frame axes must be tracked. IMUs are provided with gyroscopes to measure rotational motion, to solve the accelerations into the reference frame, before integrating velocity and position.

Finally, INSs are devices that, briefly, are made up of an IMU and a processor that, computing the so-called *mechanization equations*, provides an estimate of the position of the user.

Since the whole system is local and it does not depend upon the external environment, it is then clear that INSs are *self-contained* devices.

3.1 Classification of Inertial Measurement Units

The arrangement of the accelerometers and gyroscopes is generally the same in all INSs used for navigation. The common approach is to use three single axis accelerometers and gyroscopes, with an accelerometer and gyroscope placed with its input axis along each of three mutually orthogonal axes. The collection of sensors used is referred to as the instrument cluster. The surface on which the instrument cluster is placed is referred to as the platform.

The platform is normally either suspended in a set of gimbals, referred to as a gimballed or stabilized platform system, or placed directly upon the body to which it is to reference, known as a strapdown system.

The platform system and its housing are referred to as the IMU. Each IMU has different methods of sensing rotations and correspondingly the computation process is different in each circumstance and the function of gyroscopes is different in gimballed and strapdown configurations. In the following, a brief description of the two different types of IMU is provided.

3.1.1 Gimballed Systems

A gimbal is a pivoted support that allows the rotation of an object about a single axis. Gimbals can be assembled on orthogonal axes, which is useful when there is the need to obtain a stable platform mounted on the innermost gimbal, regardless of the motion of its support. This is the case of a gimballed IMU, shown in Figure 3.1.

Gimballed IMUs are space consuming, but they allows for a greater accuracy with respect to strapdown devices. For this reason, they are mainly used on board of ships, submarines and aircrafts.

In a gimballed IMU the system tries to keep the platform orthogonal to the gravity vector. To this purpose, a minimum of three gimbals is needed to allow an inertial navigation system to remain fixed in inertial space, compensating for changes in the vehicle's yaw, pitch, and roll. In this application, the IMU is equipped with three orthogonally

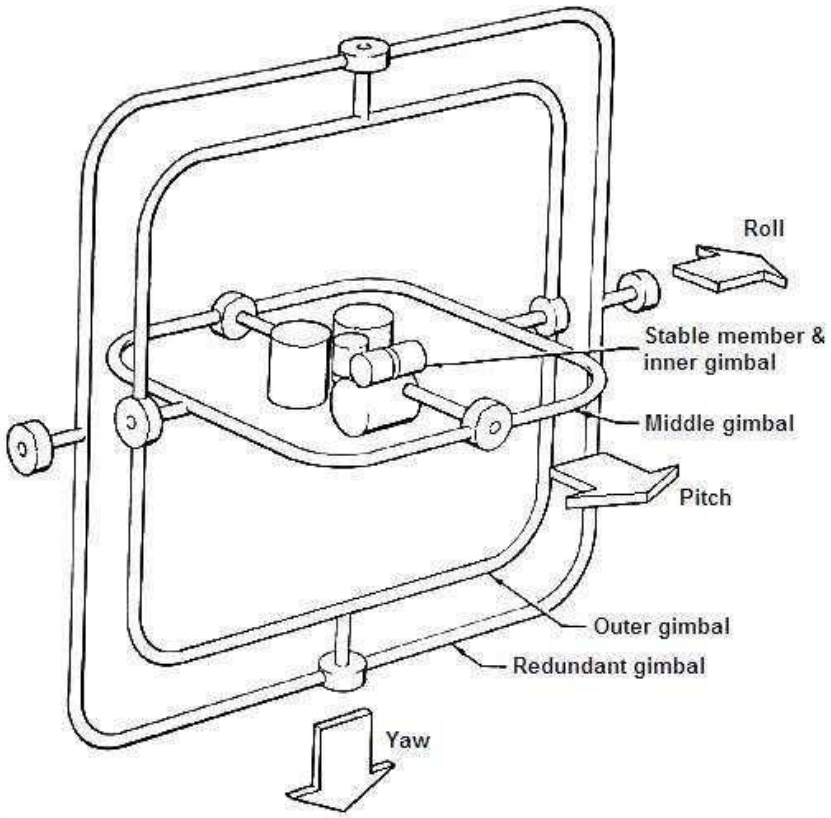

Figure 3.1. Sketch for a gimballed IMU.

mounted gyros to sense rotation about all axes in three-dimensional space. The gyro outputs drive motors controlling the orientation of the three gimbals as required to maintain the orientation of the IMU. In turn, angular measurement devices called resolvers and mounted on the three gimbals provide the nine cosine values for the direction cosine matrix needed to orient the vehicle.

In inertial navigation systems, gimbal lock may occur when vehicle rotation causes two of the three gimbal rings to align with their pivot axes in a single plane. When this occurs, it is no longer possible to maintain the orientation of the sensing platform. For this reason, a redundant gimbal can be used. This expedient is sufficient to solve the gimbal lock issue.

Gimballed platforms require numerous parts in order to maintain the platform in the correct orientation. The major components required include:

- The instrument cluster, held in a fixed orientation;

- The gimbals, used as the divider between the instrument cluster and the case to which motion is referenced. The gimbal has two orthogonal pivot axes which attach to outward and inward pivot axes;
- The bearings, used to allow the motion of the gimbals, found at the pivot axes of all gimbals;
- The gimbal motors, used to help maintaining the orientation of the platform. There exists one motor for each gimbal and it is located next to the bearing;
- The pickoffs, used to measure the angular rates of the gimbals. The pickoffs are located at the opposite pivot point to the gimbal motor.

3.1.2 Strapdown System

Modern systems are characterized by a more simple architecture, since the cluster is rigidly mounted on the body of the vehicle. The absence of the gimbals allows for the implementation of smaller and cheaper devices at the cost of a different mathematical approach to the problem. A Microstrain 3DM-GX3 device is shown in Figure 3.2.

Figure 3.2. Strapdown IMU.

A strapdown system does not attempt to preserve a fixed alignment like gimballed systems, since the platform is attached to the body of the vehicle to which it is to reference. The IMU in strapdown configurations is normally placed so that its axes are parallel to the axes of the vehicle. Whereas a gimballed system provides the defined axes in a mechanical method, maintaining an orientation, a strapdown system computes the

axes orientation mathematically, calculating the separation of the current state from the original defined axes, replacing the mechanical gimbal set with a mathematical one.

Nowadays, strapdown systems are getting far more common than gimballed ones, because of the simplicity of their mechanical realization.

3.2 Accelometer Technologies

Accelerometers are the devices devoted to the measurement of the *proper acceleration*, i.e. the acceleration experienced in a free fall. Its functioning is well described by a spring-mass system, where a mass elongates the spring up to the equilibrium between the applied force and the reaction of the spring.

Many different types of accelerometers exist. A short description of some accelerometer technologies is given in the following. They have been chosen since they are the most well suited for silicon implementation.

Piezoelectric

This family of accelerometers exploits the piezoelectric effect to measure accelerations. This is an effect typical of the materials that show a charge, i.e. an electric potential, when they are subject to a pressure.

These devices are made up by a mass which is in a direct contact with a piezoelectric element. When the mass experience a force, an electric charge proportional to the piezoelectric coefficient appears across the piezoelectric element.

High piezoelectric coefficients allow to realize systems comparable to spring-mass systems with small masses and high spring constants. These kind of devices are useful for high frequency applications.

Strain Gage

This family of accelerometers exploits the piezoresistance property of electrical conductors. In fact, if a conductor that is stretched or compressed, it experiences a variation of its resistivity. This effect is different from the piezoelectric effect, since no electric potential appears.

There exist many different realizations of strain gage accelerometers, but the most

common are based on bonded and unbonded metal wire gages. Also bonded semiconductor gages exist and they are mostly used in MEMS devices.

Capacitive

Capacitive accelerometers are usually made up by three parallel plates built in silicon: each one of the external plates forms a capacitor with the middle plate.

In these devices, the middle one represents the mass and four tethers constitute the spring. Since the middle plate can move, the experienced accelerations can be translated in mass displacements. When the middle plate changes its position, the device experience a change of capacitance between the middle and the external plates. This change is transduced into an electrical signal, that represents the measurement of acceleration.

3.3 Gyroscope Technologies

Gyroscopes can be realized in many different ways. In the following, we briefly analyze three different implementation, characterized by different precision, dimension and cost.

Vibrating Structure

A vibrating structure gyroscope is a device whose functioning is based on the measurement of the Coriolis force generated from the rotation of the device. In fact, from a physical point of view, a vibrating object tends to continue vibrating in the same plane as its support rotates, generating a force.

A vibrating motor keeps the oscillations of a vibrating elements constant, while a sensing capacitor measures the Coriolis force due to rotation [33].

Vibrating structure gyroscopes are the cheapest devices that we consider. They are gaining a great importance in the last years, since MEMS IMU generally implements this kind of gyroscopes. Possible implementations are based on piezoelectric resonators or tuning forks, lithographically constructed in the package.

Sagnac Effect Based Gyroscopes

When a beam of light is split and the two beams are made to follow a trajectory in opposite directions, the Sagnac effect may appear. This effect is due to the fact that the beam is

propagating in a plane that is rotating around its orthogonal axis and it can be measured by means of an interferometer.

The most precise implementations of gyroscopes rely on this principle.

Fibre Optic

A fibre optic gyroscope uses the interference of light to detect mechanical rotation. The sensor is made up by a long optical fibre wrapped around a coil. A beam splitter is used to split the beam of a laser diode in two beams. This light beams travel the optical fiber in two opposite directions and they are recombined at the end.

If the sensor undergoes a rotation, the beam traveling against the rotation covers a shorter path than the other beam, involving a phase shift. When the two beams are recombined, their phases are different and it is possible to determine the angular velocity of the device measuring the intensity of the combined beam.

Ring Laser

A ring laser gyroscope is a self-contained interferometer. A laser generator is put in the path of the light. As the laser excitation is started, a brief transient characterized by the emission of a range of frequencies is present. This transient ends when a particular frequency outcompetes the others and the light is very close to monochromatic.

The number of cycles of the light emitted by the laser is the same in both directions. This property is preserved when the ring laser setup is rotating, thanks to a wavelength shift that allows the same number of cycles in both directions of propagation. If the two received frequencies of laserlight interfere, a new frequency, given by the difference between the two frequencies, is obtained and the period of this beat frequency is linearly proportional to the angular velocity of the ring laser with respect to inertial space.

3.4 Structure of an Inertial Measurement Unit

Inertial navigation is based upon the measurement of the *external force* applied to the body, mentioned in Newton's First Law. Every force applied to the body generates an acceleration, that can be integrated in order to obtain the body velocity and, after one more integration, position, both with respect to an initial condition. The device demanded to

the measurement of the total applied acceleration is the *accelerometer*. An accelerometer measures the acceleration that the sensor is subject to, along a predefined direction.

A component of the acceleration measured by these devices is represented by the gravity acceleration $\mathbf{g}$. Accelerometers do not sense directly the gravity acceleration, but the reaction to this acceleration. This phenomenon can be easily understood if one considers a spring-mass system whose elongation axis is parallel to the gravity direction. The mass will be attracted towards the centre of the Earth, but the system measure a force equal to the weight of the mass but opposite in direction. The same criterion applies to the accelerometers, whatever technology they rely on.

In order to analyze the motion of the body, the gravity component must be removed from the total acceleration acting on the body. In fact, a steady body would record an acceleration equal to the reaction to the gravity acceleration, whose integration would make the body position and velocity to diverge. The gravity component is also present when the body is in motion and, in order to remove its contribution, the inertial device needs other sensors to measure the body attitude.

Gyroscopes are the devices demanded to the computation of the attitude. They measure angular speed (turn rate) the sensor is subject to, along a predefined axis. The angular speed, once integrated, provides for angular displacement (orientation). Thus, the angular orientation of the body can be calculated by integrating the angular rate measurements, provided that an initial orientation of the sensor axis with respect to a reference is given.

As long as the accelerometers are rigidly attached to the body in movement, their reference frame is the body frame (i.e. a frame integral with the body and centered in the center of the accelerometer system), which is usually insufficient for providing the body movement with respect to an external reference frame. In fact, the rotation (angular displacement) of the body with respect to the external reference frame, i.e. the body attitude, must be computed from the instantaneous angular orientation of the body, obtained from three gyroscopes rigidly mounted on the body along the orthogonal accelerometers axes. Thereby, the attitude information is used to resolve the accelerometer measurements into the reference frame.

The knowledge of the body attitude allows to compute the total vehicle acceleration $\mathbf{a}$ starting from the *specific force* $\mathbf{f}$, which is the acceleration of the body affected by the gravity component:

$$\mathbf{f}(t) = \mathbf{a}(t) - \mathbf{g}(t) \tag{3.1}$$

3.5 Coordinate Frames

Coordinate frames are orthogonal reference systems used to point out the displacement of the user. Considering the topic of this thesis, three coordinate frames should be discussed.

3.5.1 Inertial Frame (i-frame)

The inertial frame is considered to be non-rotating and non-accelerating with respect to far-off galaxies (Schwarz, 1996). This frame is important since the Newton's laws are defined in this frame and every acceleration measured by the IMU has to be related to this frame.

The axes of this reference system are defined as follows

- origin in the centre of mass of the earth.
- z axis along the instantaneous spin axis of the earth.
- x axis towards the mean equinoctial colore in the equatorial plane.
- y axis orthogonal to the plane identified by the x and z axes, to complete a right-handed frame.

3.5.2 Earth Centred Earth Fixed Frame (ECEF or e-frame)

The ECEF frame is used for navigation operations, since the GPS solution is easily obtained in this frame and the IMU measurements can be easily reported in this frame. It is defined as follows

- origin in the centre of mass of the earth.
- z axis along the instantaneous spin axis of the earth.
- x axis pointing towards the intersection of meridian of Greenwich with the Equator.
- y axis orthogonal to the plane identified by the x and z axes, to complete a right-handed frame.

3.5.3 Body Frame (b-frame)

The body frame is related to the three axes that characterize the disposal of the IMU components. While in gimballed IMUs this frame can be kept aligned with a given reference frame (e.g. with the East-North-Up (ENU) frame, to keep it leveled), in a strapdown inertial system IMU is rigidly mounted to the vehicle, so that the orientation of its axes will change during the navigation. It is a common choice to orient the axes of the IMU with the ones of the vehicle, even if it is not mandatory

- Origin : Centre of IMU
- x axis pointing towards the right of the vehicle.
- y axis pointing towards the front of the vehicle.
- z axis orthogonal to the plane identified by the x and z axes, to complete a right-handed frame.

3.6 The Coriolis Theorem

The Coriolis theorem relates the velocity of the body with respect to the Earth measured in an inertial frame $\mathbf{v}^i(t)$ to the velocity expressed in the rotating frame $\mathbf{v}^e(t)$

$$\mathbf{v}^e(t) = \dot{\mathbf{p}}^e(t) = \mathbf{v}^i(t) - \boldsymbol{\omega}_{ie} \times \mathbf{p}(t) \tag{3.2}$$

where $\mathbf{v}^e(t)$ is the ground speed, $\mathbf{v}^i(t)$ is the speed with respect to the inertial frame, $\boldsymbol{\omega}_{ie}$ is the turning rate of the ECEF frame with respect to the inertial frame, $\times$ denotes vector cross product and $\mathbf{p}(t)$ is the position of the vehicle on the earth [34]. The derivative of (3.2) gives [34]

$$\begin{aligned}
\frac{d\mathbf{v}^e(t)}{dt}\Big|_i = \dot{\mathbf{v}}^i(t) &= \ddot{\mathbf{p}}^i(t) - \boldsymbol{\omega}^e_{ie} \times \dot{\mathbf{p}}^i(t) = \\
&= \ddot{\mathbf{p}}^i(t) - \boldsymbol{\omega}^e_{ie} \times \mathbf{v}^i(t) = \\
&= \ddot{\mathbf{p}}^i(t) - \boldsymbol{\omega}^e_{ie} \times \left[\mathbf{v}^e(t) + \boldsymbol{\omega}^e_{ie} \times \mathbf{p}^i(t)\right] = \\
&= \ddot{\mathbf{p}}^i(t) - \boldsymbol{\omega}^e_{ie} \times \mathbf{v}^e(t) - \boldsymbol{\omega}^e_{ie} \times \left[\boldsymbol{\omega}^e_{ie} \times \mathbf{p}^i(t)\right]
\end{aligned} \tag{3.3}$$

where the Earth turn rate has been assumed constant, i.e. $\dot{\boldsymbol{\omega}^e_{ie}} = \mathbf{0}$. In (3.3), the term $\boldsymbol{\omega}^e_{ie} \times \mathbf{v}^e(t)$ is known as *Coriolis acceleration* and represents the acceleration caused by

the body velocity over the surface of a rotating Earth, while the term $\boldsymbol{\omega}_{ie}^{e} \times \left[\boldsymbol{\omega}_{ie}^{e} \times \mathbf{p}^{i}(t)\right]$ defines the *centripetal acceleration* experienced by the body owing to the rotation of the Earth.

3.7 Mechanization Equations

The procedure used to compute the trajectory of the body (i.e. the instantaneous position, velocity and attitude) in the selected reference frame starting from the inertial sensors measurements is called *mechanization*.

These equations are presented hereafter in the order they are used in an INS:

1. Computation of the current attitude in the proper reference frame;
2. Computation of the current velocity in the proper reference frame;
3. Computation of the current position in the proper reference frame.

The considered reference frame is the ECEF one, whose axes have their origin in the centre of the Earth. The x axis is oriented towards the intersection between the equator and the Greenwich meridian, the z axis along the Earth rotation axis and the y axis is oriented in order to obtain a right-handed triplet of orthogonal axes.

3.7.1 Computation of the Attitude

The output of the gyroscopes is a vector of three angular rates

$$\boldsymbol{\omega}^{b} = [\omega_{bx}, \omega_{by}, \omega_{bz}]^{T} \tag{3.4}$$

from which the body attitude can be formally obtained via time integration. In order to express the angular rate vector measured by the gyros in the e-frame, the following transformation must be computed [34]

$$\boldsymbol{\omega}_{eb}^{b}(t) = \boldsymbol{\omega}^{b}(t) - \mathbf{C}_{b}^{eT}(t)\boldsymbol{\omega}_{ie}^{e} \tag{3.5}$$

where $\boldsymbol{\omega}_{eb}^{b}(t)$ represents the turn rate of the body with respect to the e frame (subscript eb) as measured by the gyros (superscript b), $\boldsymbol{\omega}_{ie}^{e}$ is the Earth rotation rate expressed in body axes and $\mathbf{C}_{b}^{e}(t)$ is the rotation matrix (Direct Cosine Matrix (DCM)) from the body

frame to the Earth frame. In particular, the DCM may be calculated from $\boldsymbol{\omega}_{eb}^{b}(t)$ using the relationship [34]

$$\dot{\mathbf{C}}_{b}^{e}(t) = \mathbf{C}_{b}^{e}(t)\boldsymbol{\Omega}_{eb}^{b}(t) \tag{3.6}$$

where $\dot{\mathbf{C}}_{b}^{e}(t)$ is the time derivative of $\mathbf{C}_{b}^{e}(t)$ and $\boldsymbol{\Omega}_{eb}^{b}(t)$ is the skew-symmetric matrix derived from $\boldsymbol{\omega}_{eb}^{b}(t)$.[1]

Discrete-time propagation of the DCM can be obtained by means of the Taylor approximation

$$\mathbf{C}_{b}^{e}[n+1] = \left(\mathbf{I} + T_c\boldsymbol{\Omega}_{eb}^{b}[n]\right)\mathbf{C}_{b}^{e}[n] \tag{3.12}$$

3.7.2 Time Propagation of the Velocity

The effect of the gravity on the measurement must be removed at first analyzing the measurement of the IMU while the body is still. Such analysis allows the INS to estimate the attitude of the body, that is represented by the DCM, and to remove the gravity effect simply adding a vector that nulls the output to the measurement. Such a vector can be easily computed using a gravity model.

After the steady state alignment, the body starts to move and to change its attitude, therefore also the gravity vector to be subtracted rotates accordingly to the body. The DCM update equation is necessary to correctly remove the gravity acceleration, because it tracks the body attitude variations and, consequently, the variations of the vector that nullifies the gravity effect.

Two other forces that must be removed are represented by the Coriolis acceleration and by the centripetal force due to earth rotation. Starting from the specific force $\mathbf{f}^{b}$ that is measured by the accelerometers and subtracting the Coriolis, the centripetal and the

[1] A *skew-symmetric matrix* is a square matrix whose transpose is also its negative; that is, it satisfies the equation $\mathbf{A}^{T} = -\mathbf{A}$. Given a vector

$$\mathbf{a} = [a_1\ a_2\ a_3]^{T}, \tag{3.7}$$

it is possible to build the skew-symmetric matrix associated to the vector

$$\mathbf{A} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{3.8}$$

The following relationships hold between vector cross-product and vector-matrix product:

$$\mathbf{a}\times\mathbf{b} = \mathbf{A}\mathbf{b}, \tag{3.9}$$

$$\mathbf{a}\times\mathbf{b} = -\mathbf{b}\times\mathbf{a} = -\mathbf{B}\mathbf{a}, \tag{3.10}$$

$$\mathbf{a}\times[\mathbf{a}\times\mathbf{b}] = \mathbf{A}\mathbf{A}\mathbf{b}. \tag{3.11}$$

gravity accelerations, it is possible to obtain an equation to describe the velocity variations in time, as follows:

$$\dot{\mathbf{v}}^e(t) = \mathbf{C}_b^e(t)\mathbf{f}^b(t) - 2\boldsymbol{\Omega}_{ie}^e\mathbf{v}^e(t) + \mathbf{g}_\ell^e(t) \tag{3.13}$$

where $\mathbf{g}_\ell^e$ is the sum of the gravity and of the centripetal acceleration in the e-frame.

3.7.3 Time Propagation of the Position

The propagation of the position is described by the well known relationship between space and velocity:

$$\dot{\mathbf{p}}^e(t) = \mathbf{v}^e(t) \tag{3.14}$$

where $\mathbf{p}^e(t)$ is the 3-dimensional position of the body in the e-frame.

3.8 Initial Alignment

The mechanization equations presented in Section 3.7 allow to compute the variations of the angles, velocity and position that characterize a vehicle. These variations must be applied to an initial point, in order to keep the estimate of the position of the vehicle up to date. For this reason, the initial position, velocity and attitude of the vehicle must be known.

While the position and the velocity of the user may be easily determined, the estimation of the initial attitude of the vehicle requires an alignment procedure. This operation is usually performed in two steps, namely the coarse alignment and the fine alignment. Moreover, in the following the vehicle will be considered still.

Coarse Alignment

The coarse alignment provides a first coarse estimation of the roll, pitch and yaw angles of the IMU. These angles are related to the ENU frame, so that a further rotation, involving the latitude and longitude of the user, is required to obtain the rotation matrix from the body frame to the ECEF frame.

The first quantity that must be considered in order to perform the horizontal alignment of the IMU is the gravity vector. Under static conditions, so that the output of the

accelerometers is constituted only by the reaction to gravity, the roll angle can be computed as

$$\hat{\phi} = -\sin^{-1}\left(\frac{\overline{a}_y}{\sqrt{\overline{a}_y^2 + \overline{a}_z^2}}\right) \tag{3.15}$$

The term $\overline{a}_i$ points out a time average on the output on the i-th axis. The time requested for coarse alignment operation depends upon the quality of the IMU. A reasonable averaging time for our IMU time is between five and ten minutes.

Once the roll angle is computed, the pitch angle can be computed as follows

$$\hat{\theta} = \sin^{-1}\left(\frac{\overline{a}_x}{\sqrt{\overline{a}_x^2 + \overline{a}_y^2 + \overline{a}_z^2}}\right) \tag{3.16}$$

The roll and pitch angles can be used to perform the horizontal alignment, left-multiplying the output vector with the appropriate rotation matrix $\mathbf{C}^h$

$$\mathbf{C}_b^h = \mathbf{C}_y(-\theta)\mathbf{C}_x(-\phi) \tag{3.17}$$

where $\mathbf{C}_x$ and $\mathbf{C}_y$ represent the rotations around the x and y axis, respectively.

The $\mathbf{C}^h$ matrix can be used to rotate in the horizontal plane the output of the magnetometers. An average of the horizontal components $m_{h,x}$ and $m_{h,y}$ can then be used to perform the estimation of the heading. The horizontal component of this vector must be rotated by an angle ψ, so that the y axis points north

$$\hat{\psi} = -\tan^{-1}\left(\frac{\overline{m}_{h,x}}{\overline{m}_{h,y}}\right) \tag{3.18}$$

Once the roll, pitch and azimuth angles are determined, it is possible to compute the rotation matrix from the b-frame to the e-frame as

$$\mathbf{C}_b^e = \mathbf{C}_z(-\pi/2 - \nu)\mathbf{C}_y(\lambda - \pi/2)\mathbf{C}_z(\psi)\mathbf{C}_y(-\theta)\mathbf{C}_x(-\phi) \tag{3.19}$$

where λ and ν are the latitude and longitude of the IMU respectively.

Fine Alignment

The fine alignment process is required to refine the estimates of the initial roll, pitch and yaw angles provided by the coarse alignment operation.

This process exploits an extended Kalman filter to estimate the error states. The main idea that allows to set up the model associated to the Kalman filter is that the horizontal acceleration must be equal to zero, since the IMU has been approximately aligned.

3.9 IMU Errors

Different kinds of errors affect the operation of the INS. They are mostly related to the characteristics and the assembling of the sensors of the IMU. The most relevant errors can be imputed to noise, biases, scale factors and non perfect orthogonality between sensors. We will briefly analyze these errors in the following.

Noise characterizes and affects the operation of every electronic equipment, interfering with the output signal. Since it is not a deterministic process, it cannot be removed from the signal, but its impact must be quantified.

In Kalman filter based systems, it is possible to stochastically model noise by means of a noise covariance matrix that points out the amount of uncertainty associated to the INS only solution due to the presence of sensor noise. This solution is based on the assumption that noise is a zero mean white Gaussian process.

Different techniques can be adopted to estimate the power associated to this stochastic process. The one used in this work consists in the analysis of the log of a static IMU and in the computation of the variance associated to this data. More advanced techniques are instead based on wavelet decomposition, in order to separate the low frequency component, associated to different error processes, from the high frequency components due to noise.

The bias is related to the mean output of the sensor in a time where it is not experiencing forces or torques. If an IMU is kept still and the effect of gravity is removed from the output, the bias is equal to the mean of the output of the device.

It is usual to consider different contributions to this error: a turn on component and a drift component. The turn on bias is different each time the IMU is powered and it requires a proper calibration. Since this initial bias is not stable in time, variations must be taken into account. The drift models how the bias changes in time. If this component is slow and it is relevant with respect to the sensor noise, a proper stochastic modeling is useful to improve the performance of the INS.

Scale factor errors are due to a ratio between the output and the input which is different from the unit. It is a deterministic error related to the sensors and it can be measured during the calibration process.

Finally, there exists non-orthogonality errors due to a non perfect assembly of the sensors. The accelerometers and gyroscopes that forms an IMU must be assembled along

three mutually orthogonal axes, so that each measurement is not influenced by the measurements along the other axes. Of course, an assembly affected by misalignment implies that the measurement of a sensor is affected by the forces and torques along the other two axes.

Chapter 4

The Kalman Filter

When a set of measured data is affected by errors, a typical problem is to try to process the data with an algorithm able to perform some denoising. A very famous algorithm of this kind is the Kalman Filter (KF).

The KF is a set of recursive equations that efficiently estimates the state of a system in order to minimize the Mean Squared Error (MSE) between the real value and the estimated one. In GNSS applications the system is generally an object moving on or flying over the surface of the Earth and its state may include position, velocity, acceleration, attitude and also nuisance variables, for example due to noise sources. The equations involved can be defined in continuous-time or by means of discrete-time formulations. In general, since usual real-time applications of the recursive solution are implemented to process discrete data, the latter definition is preferred to describe the main operations of the KF.

Here, the discrete KF is used to estimate the states of a discrete time process created by linear equations. The discrete-time formulation is a simple derivation of the continuous-time one. The variables of interest are quantities, called *states*, that the KF aims to estimate exploiting the recursive procedure. The states are described by means of a theoretical model that defines how their values change with time and by direct or indirect measurements. The information generated by the model and measurements is combined by the KF to estimate the states.

4.1 State-Space Models

To construct a suitable model characterizing the relationships among the states of a system, it is common to consider the state as a time domain signal, at the output of a known shaping filter. This is a classical approach that can be easily found in literature [6]. The autocorrelation output of the system represented in Figure 4.1 can be obtained by

$$r_x(t) = r_\eta(t) * r_g(t) \tag{4.1}$$

where $r_\eta(t)$ is the autocorrelation of the input noise and $r_g(t)$ is the temporal autocorrelation of $g(t)$, impulse response of the shaping filter.

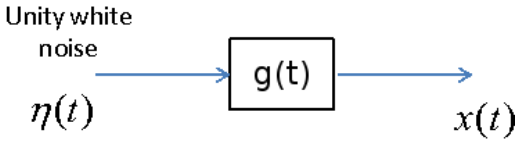

Figure 4.1. Shaping filter [6].

If the problem is analyzed in the Laplace domain s, the function $G(s)$, obtained as the Laplace transform of the function $g(t)$ in Figure 4.1, is the transformation to be applied to the white noise signal given as input, typically having unit variance. The design of $G(s)$ is strictly related to the target power spectral density function $S_x(s)$ of the output $x(t)$. If the function $G(s)$ have a finite number of poles and zeros, all included in the left half-plane of the S domain, the $S_x(s)$ can written by the following factorization

$$S_x(s) = \sigma^2 \cdot G(s)G(-s) \tag{4.2}$$

where σ is the standard deviation of the input white noise.

The part of the spectral factorization with poles and zeros in the left half-plan of the S domain provides the minimum phase filter, named shaping filter [6]. This allows to obtain an output $x(t)$ with known spectral characteristics by a driving input white noise.

In the continuous-time domain there exists a large class of physical processes with a time evolution well described by means of stochastic differential equations. One of the most common model example relates states by simple integrals.

Notice that this kind of transformations leads to a limit condition in terms of stability, due to the poles of the system, but it allows to easily define the equation model, setting an intuitive physical relationship among the states. As an example to clarify the concept,

three states, such as acceleration, velocity and position, can be considered. The relations shown in Figure 4.2 can be applied to obtain these state.

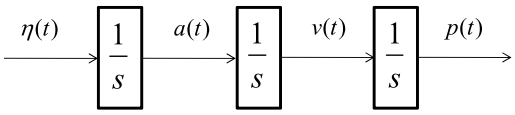

Figure 4.2. State model in Laplace domain.

In Figure 4.2, $\eta(t)$ represents the white noise input function, $a(t)$ the acceleration, $v(t)$ the velocity and $p(t)$ the position, obtained by means of subsequent integrations, represented by the integrator blocks. The first shaping filter is applied to the input to obtain an acceleration. When a white input process is integrated, the resulting output is called random walk. The continuous time-invariant state-space model can be easily defined as

$$\underbrace{\begin{bmatrix} \dot{p}(t) \\ \dot{v}(t) \\ \dot{a}(t) \end{bmatrix}}_{\dot{x(t)}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} p(t) \\ v(t) \\ a(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{\Gamma}} \eta(t) \tag{4.3}$$

$$\dot{x(t)} = \mathbf{F}x(t) + \mathbf{w}(t) \tag{4.4}$$

where

$$\mathbf{w}(t) = \mathbf{\Gamma}\eta(t) \tag{4.5}$$

A general system of differential equations governing the state-space model on which the KF works can be written as:[1]

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t) + \mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{h}(\mathbf{x}(t),t) + \boldsymbol{\nu}(t) \end{aligned} \tag{4.6}$$

where $\mathbf{x}(t)$ is the *state vector* as a function of the time t, $\mathbf{u}(t)$ is the *deterministic forcing function*, $\mathbf{w}(t)$ is the stochastic forcing function or *model noise* or *driving noise*, $\mathbf{z}(t)$ is the *measurement vector* (or *observation vector*), $\boldsymbol{\nu}(t)$ is the *measurement noise* and $\mathbf{f}$ and $\mathbf{h}$ two linear or non-linear functions. These two equations are referred as the *state equation* and the *measurement equation* (or *observation equation*), respectively. Assuming a linear KF, (4.6)

[1]Note that in the following analysis, every vector will be considered as a column, unless differently specified.

can be written in the following form

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{H}(t)\mathbf{x}(t) + \boldsymbol{\nu}(t) \end{aligned} \tag{4.7}$$

where $\mathbf{F}(t)$ is the *continuous-time state transition matrix*, $\mathbf{B}(t)$ is the matrix that relates the deterministic input to the states and $\mathbf{H}(t)$ is the linear relationship between the states and the observations (*measurement matrix*). The Extended Kalman Filter (EKF) or the Linearized Kalman Filter (LKF) are used if the functions $\mathbf{f}$ and/or $\mathbf{h}$ in (4.6) are not linear. The non-linear problem, which appears in all the applications related to GNSS based positioning, is discussed in Section 4.4.

Here, we focus on the discrete time version of the system in (4.7). Considering the state variable at some discrete time epochs $t_n = nT_s$, where T_s is the chosen sampling interval, each epoch can be seen as a discrete-time sequence $x_i[n]$, and all the variables can be represented by a *state vector* of the type

$$\mathbf{x}[n] = \begin{bmatrix} x_1[n] & x_2[n] & \cdots & x_N[n] \end{bmatrix}^T \tag{4.8}$$

where *N* is the number of state variables characterizing the system.

An important element of the Kalman filter theory is the state *propagation model*. This is the element which takes into account the dynamic of the physical processes governing the evolution of the evolution of the system state. In our GNSS applications we are mainly interested in the evolution of the state in the discrete-time domain. The *discrete-time state-space equation* in the absence of deterministic forcing functions corresponds to

$$\mathbf{x}[n+1] = \mathbf{\Phi}[n]\mathbf{x}[n] + \mathbf{w}[n] \tag{4.9}$$

where $\mathbf{x}[n+1]$ is the state vector at time $n+1$, $\mathbf{\Phi}[n]$ is the *discrete-time state transition matrix*, which relates the states at the time n to the states at the time $n+1$ in the absence of forcing functions, and $\mathbf{w}[n]$ is a zero-mean white Gaussian noise vector of length *N* named *discrete-time model noise*, with known covariance

$$E[\mathbf{w}[n]\mathbf{w}[i]^T] = \begin{cases} \mathbf{Q}[n] & n = i \\ 0 & n \neq i \end{cases} \tag{4.10}$$

where $\mathbf{Q}[n]$ is the covariance matrix of the process noise.

At each epoch *n* a vector $\mathbf{z}[n]$ of *M* measured data can be defined as

$$\mathbf{z}[n] = \begin{bmatrix} z_1[n] & z_2[n] & \cdots & z_M[n] \end{bmatrix}^T \tag{4.11}$$

The measured vector is a linear combination of the state vector $\mathbf{x}[n]$, plus additive noise, that is

$$\mathbf{z}[n] = \mathbf{H}[n]\mathbf{x}[n] + \boldsymbol{\nu}[n], \tag{4.12}$$

where $\mathbf{z}[n]$ is the discrete-time measurement and $\mathbf{H}[n]$ is the *discrete-time observation matrix*, that relates the state $\mathbf{x}[n]$ with the measurement $\mathbf{z}[n]$. The term $\boldsymbol{\nu}[n]$ is an additive noise component (of length M) with know statistical properties. Usually $\boldsymbol{\nu}[n]$ is supposed to be a zero-mean white Gaussian process, with known covariance $\mathbf{R}[n]$, defined by

$$E[\boldsymbol{\nu}[n]\boldsymbol{\nu}[i]^T] = \begin{cases} \mathbf{R}[n] & n = i \\ 0 & n \neq i \end{cases} \tag{4.13}$$

The term measurement refers to the outputs of sensors that are used in estimating the system state, which is the unknown of the Kalman problem. The case with N = M and $\mathbf{H}[n] = \mathbf{I}$, where $\boldsymbol{I}$ is the identity matrix, corresponds to the simplest case, where all the state variables to be estimated are directly available from the measurement. More generically $\mathbf{H}[n]$ is the $M \times N$ matrix representing a linear transformation that the measuring system (or a transmission channel) operates on the data to be estimated. Note that the model noise and the observation noise are independent vectors, then

$$E[\mathbf{w}[n]\boldsymbol{\nu}[i]^T] = 0 \tag{4.14}$$

Notice that the vector $\mathbf{w}[n]$ can be written also as

$$\mathbf{w}[n] = \boldsymbol{\Gamma}[n]\boldsymbol{\eta}[n] \tag{4.15}$$

where $\boldsymbol{\eta}[n]$ is an N length vector of sampled zero-mean white sequence and $\boldsymbol{\Gamma}[n]$ is the $N \times N$ *input matrix*. Because of the linearity of the operation, $\mathbf{w}[n]$ is a zero-mean white process too, as requested by the KF theory (see Chapter 21).

It is evident that, given a system and its state, the first step is to verify if (4.9) holds. If the evolution of the state is not governed by (4.9), the standard KF cannot be applied. A possible solution is to adopt a complementary Kalman filter (see Section 4.4).

In the presence of a deterministic forcing function (i.e., when the state evolution is driven *also* by a deterministic signal, whose temporal evolution is known[2]) the system

[2] An example of such a situation is a Position-Velocity state model, with an Acceleration component assumed known in average at each instant. In this case the evolution of state Velocity is driven by a non-zero-mean, time-varying noise process, whose mean value is deterministic.

evolution and the measurements relation can be described by the more general forms:

$$\mathbf{x}[n+1] = \mathbf{\Phi}[n]\mathbf{x}[n] + \mathbf{B}[n]\mathbf{u}[n] + \mathbf{\Gamma}[n]\boldsymbol{\eta}[n] \tag{4.16}$$

$$\mathbf{z}[n] = \mathbf{H}[n]\mathbf{x}[n] + \mathbf{D}[n]\mathbf{u}[n] + \boldsymbol{\nu}[n] \tag{4.17}$$

where

- $\boldsymbol{\eta}[n]$ is the discrete-time random input system noise (discrete-time stochastic forcing function). It is considered white and Gaussian.
- $\mathbf{u}[n]$ is the discrete-time deterministic input;
- $\mathbf{B}[n]$ is the matrix that relates the deterministic input to the states;
- $\mathbf{\Gamma}[n]$ is the matrix that relates the forcing function to the states;
- $\mathbf{D}[n]$ is a matrix giving the connection between the measurement and the deterministic input.

Equation (4.16) can be obtained by the superimposition principle and decomposed in the sum of a linear deterministic system

$$\mathbf{x}_u[n+1] = \mathbf{\Phi}[n]\mathbf{x}_u[n] + \mathbf{B}[n]\mathbf{u}[n] \tag{4.18}$$

and a linear stochastic system given by

$$\mathbf{x}_\eta[n+1] = \mathbf{\Phi}[n]\mathbf{x}_\eta[n] + \mathbf{\Gamma}[n]\boldsymbol{\eta}[n] \tag{4.19}$$

where the subscripts u and η are used to for the contribute due to the deterministic input and forcing function respectively.

Also the measurement equation, as the state one, can be decomposed considering the sum of deterministic input and noise contribution (subscript ν).

$$\mathbf{z}[n] = \mathbf{z}_u[n] + \mathbf{z}_\nu[n] \tag{4.20}$$

where

$$\mathbf{z}_u[n] = \mathbf{H}[n]\mathbf{x}_u[n] + \mathbf{D}[n]u[n] \tag{4.21}$$

and

$$\mathbf{z}_\nu[n] = \mathbf{H}[n]\mathbf{x}_\eta[n] + \boldsymbol{\nu}[n] \tag{4.22}$$

The block diagram of Figure 4.3 is a graphical representation of the system described by equations (4.16) and (4.17).

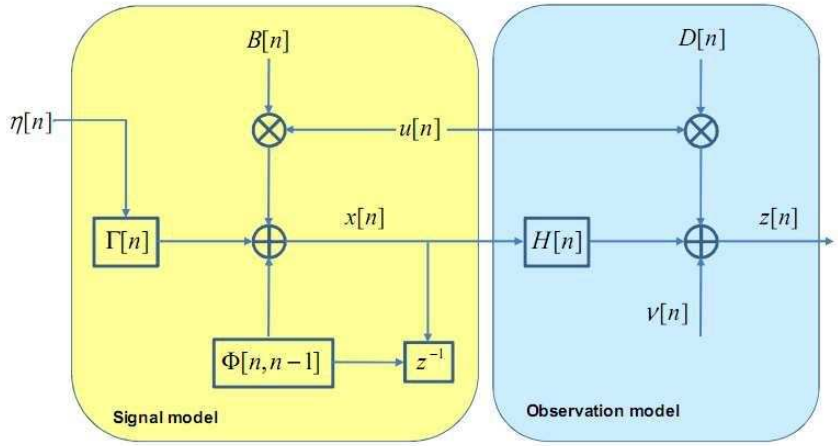

Figure 4.3. State model.

4.2 Continuous Time to Discrete Time Transformation

The equations of motion typically involved in navigation problems are usually written and handled in the continuous-time domain. However, the automatic computation of the navigation solution needs their translation in a discrete-time form. For this reason this Section is devoted to go through the transformation process of the differential problem (4.7).

Expressions (4.9) and (4.12) can be derived from (4.7) in several ways. The most common approach is described in this Section [6, 35], despite other techniques exist [36].

Starting from the continuous-time state equation (4.7) in absence of forcing functions (invoking the superposition principle, the noise contribution will be discussed later on in this Section), i.e.

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t), \tag{4.23}$$

it is possible to suppose that the transition matrix $\mathbf{F}(t)$ is constant during the sampling interval T_c. In this case, using a Taylor series expansion, it is possible to determine the discrete time state transition matrix at time n $\mathbf{\Phi}[n]$ as follows [35]

$$\mathbf{\Phi}[n] = e^{\mathbf{F}(t_{n+1},t_n)T_c} = [\mathbf{I} + \mathbf{F}(t_{n+1},t_n)T_c + \dots] \tag{4.24}$$

At time t_n Equation (4.24) can be substituted into Equation (4.23), in order to compute

the value of the state at time $t_{n+1} = t_n + T_c$, obtaining

$$\mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) \cdot e^{\mathbf{F}_{n+1,n}T_c} = \mathbf{x}(t_n) \cdot [\mathbf{I} + \mathbf{F}_{n+1,n}T_c + \dots] \tag{4.25}$$

where $\mathbf{F}_{n+1,n}$ is the value assumed by $\mathbf{F}(t)$ in the interval (t_n,t_{n+1}).

We stress that such an approximation is valid if the continuous time transition matrix is constant during a sampling interval T_c. Thereby, from (4.24)-(4.25) the discrete-time formulation of the state-space equation in the absence of forcing functions is

$$\mathbf{x}[n+1] = \mathbf{\Phi}[n]\mathbf{x}[n] \tag{4.26}$$

Equation (4.24) shows that different approximations are possible for the discrete-time state transition matrix. For instance, an approximation to the first order from time n to time $n+1$, $\forall n \geq 0$, is

$$\mathbf{\Phi}[n] = \mathbf{I} + \mathbf{F}_{n+1,1}T_c \tag{4.27}$$

In order to get a better approximation, a second order approximation might be considered, obtaining

$$\mathbf{\Phi}[n] = \mathbf{I} + \mathbf{F}_{n+1,1}T_c + \frac{(\mathbf{F}_{n+1,1}T_c)^2}{2!} \tag{4.28}$$

It is worth noticing that (4.27) and (4.28) are two different but possible formulations for the discrete-time state transition matrix, whose choice depends on the accuracy needed to represent the state model. The procedure explained in this paragraph to define the discrete-time matrix $\mathbf{\Phi}[n]$ is the most common one, but other methods can be applied, obtaining similar results.

While obtaining the discrete time version of the transition matrix is an easy task, the definition of the covariance matrix of the sampled noise term $\mathbf{w}[n] = \mathbf{\Gamma}[n]\boldsymbol{\eta}[n]$ in (4.9) may be non trivial. The (approximate) analytical solution of (4.7) in the presence of the model noise $\mathbf{w}(t)$ only can be written as

$$\mathbf{x}(t_{n+1}) = \mathbf{\Phi}[n]\mathbf{x}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{F}(t_{n+1},\tau)\mathbf{w}(\tau)\, d\tau \tag{4.29}$$

where $\mathbf{F}(t_{n+1},\tau)$ is the continuous time transition matrix from time τ to time t_{n+1}. It is then possible to relate $\mathbf{w}[n]$ to $\mathbf{w}(t)$ as follows

$$\mathbf{w}[n] = \int_{t_n}^{t_{n+1}} \mathbf{F}(t_{n+1},\tau)\mathbf{w}(\tau)\, d\tau. \tag{4.30}$$

Note that the discrete time noise vector $\boldsymbol{\eta}[n]$ is the result of the accumulation of the continuous time noise $\mathbf{w}(t)$ over the sampling interval T_c. Therefore it has the same kind of statistical distribution of $\mathbf{w}(t)$ but different variance.

As a consequence, the model noise covariance matrix that characterizes $\mathbf{w}[n]$ can be obtained as

$$\begin{aligned}\mathbf{Q}[n] &= \mathbb{E}\left\{\mathbf{w}[n]\mathbf{w}^H[n]\right\} \\ &= \mathbb{E}\left\{\left[\int_{t_n}^{t_{n+1}} \mathbf{F}(t_{n+1},\tau)\mathbf{w}(\xi)d\xi\right]\left[\int_{t_n}^{t_{n+1}} \mathbf{F}(t_{n+1},\tau)\mathbf{w}(\tau)d\tau\right]^H\right\} \\ &= \int_{t_n}^{t_{n+1}}\int_{t_n}^{t_{n+1}} \mathbf{F}(t_{n+1},\tau)\mathbb{E}\left\{\mathbf{w}(\xi)\mathbf{w}^H(\tau)\right\}\mathbf{F}^H(t_{n+1},\tau)d\xi d\tau\}\end{aligned} \tag{4.31}$$

Equation (4.31) cannot be evaluated easily. If the transition matrix $\mathbf{F}(t_{n+1},\tau)$ can be considered constant during the sampling interval, that is our case, then (4.31) can be drastically simplified, exploiting (4.28) and obtaining

$$\mathbf{Q}[n] = \boldsymbol{\Phi}[n]\mathbf{Q}_w\boldsymbol{\Phi}^H[n]T_c. \tag{4.32}$$

where the matrix $\mathbf{Q}_w = \mathbb{E}\left\{\mathbf{w}(\xi)\mathbf{w}^H(\tau)\right\}$ is a diagonal matrix, due to the nature of the model noise, assumed uncorrelated in time.

Finally, the discrete-time version of the measurement equation is simply defined sampling the continuous time expression at the instants $t_n = nT_c$:

$$\mathbf{z}[n] = \mathbf{H}[n]\mathbf{x}[n] + \boldsymbol{\nu}[n]. \tag{4.33}$$

4.3 Recursive Estimation and Initial Conditions

In the Kalman filter approach, the state variables are estimated by means of a recursive iterative process, based on two steps. The state estimate $\widehat{\mathbf{x}}[n]$ at each discrete time n is a mixture of two main contributions:

1. a predicted estimate $\widehat{\mathbf{x}}[n+1]^-$, obtained by applying the state transition matrix, in the first step on the iterative procedure;

2. a contribution given by the measurement vector called correction step;

The basic idea is to correct the prediction using a fresh measurement at each epoch n. The prediction of the state at the $n+1$-th instant is obtained ignoring the disturbance noise, such as

$$\widehat{\mathbf{x}}[n]^{-} = \mathbf{\Phi}[n-1]\widehat{\mathbf{x}}[n-1] \tag{4.34}$$

The symbol $\widehat{\mathbf{x}}[n]$ specifies that such vector does not represent the state estimate at time n but only its prediction, carried out using both the old estimation and the matrix $\mathbf{\Phi}[n]$. The vector describing the error between the current state $\mathbf{x}[n]$ and its predicted estimate $\widehat{\mathbf{x}}[n]^{-}$ can be defined as

$$\mathbf{e}[n]^{-} = \mathbf{x}[n] - \widehat{\mathbf{x}}[n]^{-} \tag{4.35}$$

with covariance matrix

$$\mathbf{P}[n]^{-} = E\{\mathbf{e}[n]^{-}\mathbf{e}[n]^{-,T}\} \tag{4.36}$$

Similarly, defining $\mathbf{e}[n]$ as the error between the state $\mathbf{x}[n]$ and the estimate $\widehat{\mathbf{x}}[n]$, it can be defined

$$\mathbf{P}[n] = E\{\mathbf{e}[n]\mathbf{e}[n]^{T}\} \tag{4.37}$$

The predicted state $\widehat{\mathbf{x}}[n]^{-}$ is not the final estimation at epoch n, but only the starting point of the estimate $\widehat{\mathbf{x}}[n]$ provided by the Kalman filter, which can be written as

$$\widehat{\mathbf{x}}[n] = \widehat{\mathbf{x}}[n]^{-} + \mathbf{K}[n](\mathbf{z}[n] - \mathbf{H}[n]\widehat{\mathbf{x}}[n]^{-}) \tag{4.38}$$

The term $(\mathbf{z}[n] - \mathbf{H}(n)\widehat{\mathbf{x}}[n]^{-})$ is called innovation or residual vector. The equation (4.38) specifies the linear structure of the formula giving the Kalman estimate, but does not clarify the value of the factor $\mathbf{K}[n]$. In [37] the matrix $\mathbf{K}[n]$ is mentioned as the *crown jewel of the Kalman filtering*. This is because $\mathbf{K}[n]$ is the weighting factor which allows the mixing between the measured data $\mathbf{z}[n]$ and the predicted values based on the state evolution laws. An important method to evaluate $\mathbf{K}[n]$ is based on the minimization of the mean squared error. It can be demonstrated [6] that the minimization procedure leads to the following expression, also called *Kalman gain*

$$\mathbf{K}[n] = \mathbf{P}[n]^{-}\mathbf{H}[n]^{T}(\mathbf{H}[n]\mathbf{P}[n]^{-}\mathbf{H}[n]^{T} + \mathbf{R}[n])^{-1} \tag{4.39}$$

and to the associated formula for updating the $\mathbf{P}[n]$ matrix

$$\mathbf{P}[n] = [\mathbf{I} - \mathbf{K}[n]\mathbf{H}[n]]\mathbf{P}[n]^{-} \tag{4.40}$$

It must be noticed that the linear structure of (4.38) offers the optimal solution when dealing with Gaussian noise process [35]. In all the other cases such theory leads to the optimum *linear* filter. Finally, in order to start the iterative procedure, the Kalman filter is initialized with some initial conditions that may be arbitrary. Considering $n = 0$ as the initial epoch, both $\widehat{\mathbf{x}}[-1]$ and $\mathbf{P}(-1)$ quantities must be known. The initialization of the $\mathbf{P}[-1]$ elements must take into account the uncertainty on the knowledge of the states before the measuring process starts. A possible choice can be:

$$\widehat{\mathbf{x}}[-1] = \mathbb{E}\{\mathbf{x}[-1]\} \tag{4.41}$$

$$\mathbf{P}[-1] = Var\{\mathbf{x}[-1]\} \tag{4.42}$$

The entire recursive Kalman algorithm procedure in the discrete time domain is depicted in Figure 4.4.

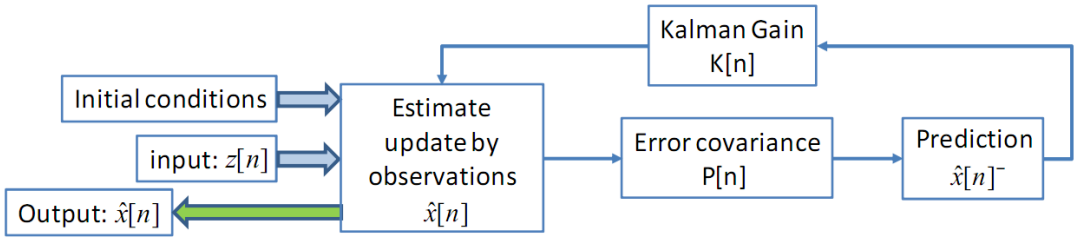

Figure 4.4. Kalman filter algorithm [6].

4.4 Complementary Kalman Filter

A complementary KF allows to use a KF to solve non-linear problems. This implies the approximation of the non-linear equations by a first order Taylor series computed about a "point" that belongs to the so-called *nominal trajectory*[3].

[3]Trajectory has to be intended in this moment as the generic evolution in time of the state vector. In practice, for a positioning problem like that addressed later on in this chapter, it is the time series of the information (whatever estimated, known or measured) about the *position* and *velocity* of a body (considered as a "punctual body") in a certain reference frame. Sometimes the trajectory may also contain the *acceleration*, the *time* (the latter is considered in GNSS receivers, where the bias and drift of the local clock with respect to the satellites clock are parameters of interest), or the body *attitude*.

The nominal trajectory has to be intended as a known trajectory that represents an approximation of the real trajectory. It can be a known approximated trajectory, like the planned path of an aeroplane that flights between two cities, or it can be provided by a theoretical motion model, such as the velocity random walk or the acceleration Gauss Markov (GM) process, or by a navigation system, like an INS.

The continuous-time differential equations governing the state-space model are written in (4.6) and reported here for clarity

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t) + \mathbf{w}(t)\\ \mathbf{z}(t) &= \mathbf{h}(\mathbf{x}(t),t) + \boldsymbol{\nu}(t)\end{aligned}$$

As introduced in Section 4.1, it is necessary to resort to a complementary KF every time at least one between $\mathbf{f}$ and $\mathbf{h}$ is non-linear.

Assuming that a nominal trajectory $\breve{\mathbf{x}}(t)$ is available, it is possible to compute the actual trajectory as follows

$$\mathbf{x}(t) = \breve{\mathbf{x}}(t) + \Delta\mathbf{x}(t) \tag{4.43}$$

where $\Delta\mathbf{x}(t)$ is the *increment* between the nominal and the actual trajectory.

Substituting (4.43) in the system (4.6), it results:

$$\begin{aligned}\dot{\breve{\mathbf{x}}}(t) + \Delta\dot{\mathbf{x}}(t) &= \mathbf{f}(\breve{\mathbf{x}}(t) + \Delta\mathbf{x}(t),\mathbf{u}(t),t) + \mathbf{w}(t)\\ \mathbf{z}(t) &= \mathbf{h}(\breve{\mathbf{x}}(t) + \Delta\mathbf{x}(t),t) + \boldsymbol{\nu}(t)\end{aligned} \tag{4.44}$$

while for the nominal trajectory the state-space equations are

$$\begin{aligned}\dot{\breve{\mathbf{x}}}(t) &= \mathbf{f}(\breve{\mathbf{x}}(t),\mathbf{u}(t),t)\\ \breve{\mathbf{z}}(t) &= \mathbf{h}(\breve{\mathbf{x}}(t),t)\end{aligned} \tag{4.45}$$

If the nominal trajectory is a good approximation of the actual trajectory, then the increment is small. In this case the two non-linear equations $\mathbf{f}$ and $\mathbf{h}$ can be approximated by means of a first order Taylor series

$$\begin{aligned}\dot{\breve{\mathbf{x}}}(t) + \Delta\dot{\mathbf{x}}(t) &\approx \mathbf{f}(\breve{\mathbf{x}}(t),\mathbf{u}(t),t) + \left[\frac{\partial\mathbf{f}(t)}{\partial\mathbf{x}}\right]_{\mathbf{x}=\breve{\mathbf{x}}}\Delta\mathbf{x}(t) + o\left(\Delta\mathbf{x}(t)\right) + \mathbf{w}(t)\\ \mathbf{z}(t) &\approx \mathbf{h}(\breve{\mathbf{x}}(t),t) + \left[\frac{\partial\mathbf{h}(t)}{\partial\mathbf{x}}\right]_{\mathbf{x}=\breve{\mathbf{x}}}\Delta\mathbf{x}(t) + o\left(\Delta\mathbf{x}(t)\right) + \boldsymbol{\nu}(t)\end{aligned} \tag{4.46}$$

where the Jacobians of the non-linear functions $\mathbf{f}$ ed $\mathbf{h}$, namely $\left[\frac{\partial \mathbf{f}(t)}{\partial \mathbf{x}}\right]$ and $\left[\frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}}\right]$, are employed. Then, using (4.45), the system (4.46) becomes

$$\begin{aligned} \Delta\dot{\mathbf{x}}(t) &= \left[\frac{\partial \mathbf{f}(t)}{\partial \mathbf{x}}\right]_{\mathbf{x}=\breve{\mathbf{x}}} \Delta\mathbf{x}(t) + \mathbf{w}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{w}(t) \\ \Delta\mathbf{z}(t) &\triangleq \mathbf{z}(t) - \breve{\mathbf{z}}(t) = \left[\frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}}\right]_{\mathbf{x}=\breve{\mathbf{x}}} \Delta\mathbf{x}(t) + \boldsymbol{\nu}(t) = \mathbf{H}(t)\Delta\mathbf{x}(t) + \boldsymbol{\nu}(t) \end{aligned} \tag{4.47}$$

which is *linear* in the *incremental states* $\Delta\mathbf{x}(t)$, provided that the following definitions hold

$$\mathbf{F}(t) = \left[\frac{\partial \mathbf{f}(t)}{\partial \mathbf{x}}\right]_{\mathbf{x}=\breve{\mathbf{x}}} \tag{4.48}$$

$$\mathbf{H}(t) = \left[\frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}}\right]_{\mathbf{x}=\breve{\mathbf{x}}} \tag{4.49}$$

$$\breve{\mathbf{z}}(t) = \mathbf{h}\left(\breve{\mathbf{x}}(t)\right). \tag{4.50}$$

The analogy of (4.47) with (4.7) is evident and allows us to apply all the results discussed in the previous sections to this problem, provided that the incremental state $\Delta\mathbf{x}(t)$ is considered. As a consequence, the KF applied to the discrete-time version of problem (4.47) corresponds to

1. *Prediction:* the *a-priori state* is estimated on the basis of the state-space model

$$\Delta\mathbf{x}^{-}[n] = \boldsymbol{\Phi}[n]\Delta\mathbf{x}[n-1] \tag{4.51}$$

and the *a-priori error covariance matrix* is computed

$$\mathbf{P}^{-}[n] = \boldsymbol{\Phi}[n]\mathbf{P}[n-1]\boldsymbol{\Phi}[n]^{T} + \mathbf{Q}[n-1] \tag{4.52}$$

where $\mathbf{P}[n] = \mathbb{E}\left\{\boldsymbol{\epsilon}[n]\boldsymbol{\epsilon}^{H}[n]\right\}$ is the *state estimation error covariance matrix*, $\boldsymbol{\epsilon}[n]$ is the *estimation error* on $\Delta\mathbf{x}[n]$ and $\mathbf{Q}[n] = \mathbb{E}\left\{\boldsymbol{\eta}[n]\boldsymbol{\eta}^{H}[n]\right\}$ is the *model noise covariance matrix.*

2. *Update:* the *innovation* (4.53) due to a new measurement is computed and — through the Kalman gain (4.54) — is applied to the predicted state to obtain its *a-posteriori estimate* (4.55)

$$\boldsymbol{\alpha}[n] = \mathbf{z}[n] - \mathbf{h}(\breve{\mathbf{x}}[n]) - \mathbf{H}[n]\Delta\mathbf{x}^{-}[n] \tag{4.53}$$

$$\mathbf{K}[n] = \mathbf{P}^{-}[n]\mathbf{H}^{T}[n]\left(\mathbf{H}[n]\mathbf{P}^{-}[n]\mathbf{H}^{T}[n] + \mathbf{R}[n]\right)^{-1} \tag{4.54}$$

$$\Delta\mathbf{x}[n] = \Delta\mathbf{x}^{-}[n] + \mathbf{K}[n]\boldsymbol{\alpha}[n] \tag{4.55}$$

where the matrix $\mathbf{R}[n] = \mathbb{E}\left\{\boldsymbol{\nu}[n]\boldsymbol{\nu}^{H}[n]\right\}$ in (4.54) is the *observation noise covariance matrix.*

The covariance matrix is also updated using the Kalman gain

$$\mathbf{P}[n] = (\mathbf{I} - \mathbf{K}[n]\mathbf{H}[n])\,\mathbf{P}^{-}[n] \tag{4.56}$$

It is worth noticing that the term $\mathbf{H}[n]\Delta\mathbf{x}^{-}[n]$ in (4.53) is an a-priori estimate $\Delta\mathbf{z}^{-}[n]$ of the increment $\Delta\mathbf{z}[n]$, computed without any information about the current measurement. At time n it is possible to consider the innovation as the residual error due to the information stored in the new observation, that is in general unpredictable using the measurement from time 0 to time $n-1$.

Defining $\mathbf{x}^{+}[n]$ as the estimated trajectory vector at time n and recalling that $\breve{\mathbf{x}}[n]$ is the known nominal trajectory at the same instant, it is possible to correct the latter adding the a posteriori state

$$\mathbf{x}^{+}[n] = \breve{\mathbf{x}}[n] + \Delta\mathbf{x}[n] \tag{4.57}$$

Thus, (4.57) concludes the n-th iteration of the algorithm, reporting the incremental states to the estimated trajectory.

The system (4.47) describes a so-called *complementary* Kalman filter. This kind of filter is used to estimate the difference between the reference trajectory and an external measurement of the actual trajectory. The error estimate $\Delta\mathbf{x}(t)$ is then used to correct the reference trajectory, as in (4.57). Complementary Kalman filters represent a valid solution for the design of multi sensor navigation architectures [6].

4.4.1 Linearized and Extended Architectures

As expressed before, a complementary Kalman filter, characterized by incremental error state, is a linear filter that can also be applied to non linear problems. In this case, the designer has to linearize the problem around a point on a known reference trajectory. Thereby the linearization represents an approximation and is affected by errors. A *corrected trajectory* is then computed by means of (4.57).

Whenever the corrected trajectory is kept away from the feedback loop, the system runs a *linearized Kalman filter.* Such a filter would perform well if the reference trajectory is reliable, because the linearization is taken about the reference trajectory.

If linearization takes place about the filter's estimated trajectory, the corrected trajectory must be periodically input in the feedback loop so that an *extended Kalman filter* is realized. The corrected trajectory is a function of the estimate of the state, that is function

of the measurements. This filter gets unstable in scenarios where initial uncertainties and measurement errors are large, but it is a good choice whenever the corrected trajectory is reasonably better than the reference trajectory.

Chapter 5

Kalman Filter for GNSS Applications

GNSS receivers rely on the Extended Kalman Filter (EKF) to compute the PVT solution, since it allows to estimate a smooth trajectory at a low computational cost, thanks to the presence of the model. Moreover, the EKF is also used to integrate inertial systems with GNSS receivers, since the INS can easily substitute the Kalman model.

In this Chapter we consider these two applications of the Kalman filter, discussing the theoretical approach, the modeling and the obtained results. Both simulations and real data analysis are considered, to provide an overall view of this topic.

5.1 Stand-Alone GNSS Receiver: PVT Computation Using an EKF

The EKF can be adopted in a stand-alone GPS receiver to compute the PVT solution. This Section presents the necessary measurements, the state variables, their initialization and their role in the computation of the PVT. We introduce this structure because it is fundamental to thoroughly understand the hybrid architectures discussed in the rest of the chapter.

5.1.1 State-Space Model

The considered Kalman filter model is characterized by the following incremental states (corrections to be applied to the nominal quantities)

$$\Delta\mathbf{x} = [\Delta x, \Delta y, \Delta z, \Delta\tau, \Delta v_x, \Delta v_y, \Delta v_z, \Delta v_\tau]^T \tag{5.1}$$

where Δx, Δy and Δz are the corrections to be applied to the predicted ECEF coordinates, $\Delta\tau$ is the correction to be applied to the predicted GNSS receiver clock bias, Δv_x, Δv_y and Δv_z are the corrections to be appliced to the predicted ECEF velocities and Δv_τ is the correction to be applied to the predicted GNSS receiver clock drift.

The state vector is then an eight dimensional vector, where the first four components represent position corrections in meters and the last four components represent velocity corrections in meters per second. This choice is fair because the considered reference system has three spatial dimensions and one temporal dimension, that can be anyway expressed in meters once it has been multiplied by the speed of light. A block scheme of the system is shown in Fig. 5.1.

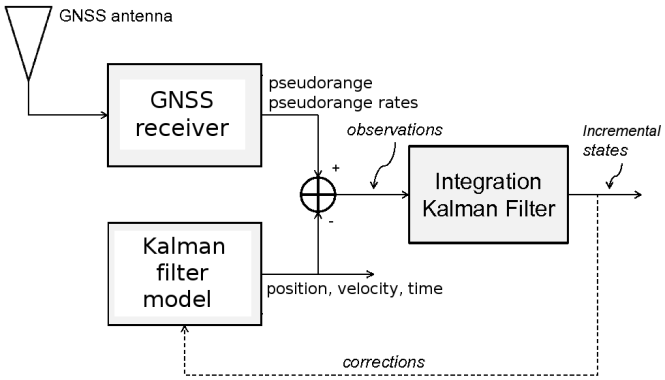

Figure 5.1. Block scheme for a stand alone GNSS receiver.

The state-space model usually considered in such a PVT estimation problem is a simple linear system written as

$$\begin{cases} \dot{x}(t) &= v_x(t) \\ \dot{v}_x(t) &= w_{v_x}(t) \end{cases} \tag{5.2}$$

for each ECEF coordinate x, y, z and

$$\begin{cases} \dot{\tau}(t) &= v_\tau(t) + w_\tau(t) \\ \dot{v}_\tau(t) &= w_{v_\tau}(t) \end{cases} \tag{5.3}$$

for the clock components. The driving noises are assumed zero-mean random processes with variances σ_p^2 for the three acceleration components $w_{v_x}(t)$, $w_{v_y}(t)$, $w_{v_z}(t)$, variance σ_τ^2 for the component $w_\tau(t)$ and variance σ_d^2 for the component $w_{v_\tau}(t)$. In the discrete

time domain, the (time-invariant) state transition matrix is then

$$\mathbf{\Phi}[n] = \begin{bmatrix} 1 & 0 & 0 & 0 & T_c & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & T_c & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & T_c & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & T_c \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

where T_c is the sampling interval, while the model noise vector is

$$\boldsymbol{\eta}[n] = [0, 0, 0, \eta_4[n], \eta_5[n], \eta_6[n], \eta_7[n], \eta_8[n]]^T \tag{5.5}$$

with a (time invariant) model noise covariance matrix structured as

$$\mathbf{Q}[n] = \mathbb{E}\left\{\boldsymbol{\eta}[n]\boldsymbol{\eta}^H[n]\right\} = \tag{5.6}$$
$$\begin{bmatrix} \sigma_p^2\frac{T_c^3}{3} & 0 & 0 & 0 & \sigma_p^2\frac{T_c^2}{2} & 0 & 0 & 0 \\ 0 & \sigma_p^2\frac{T_c^3}{3} & 0 & 0 & 0 & \sigma_p^2\frac{T_c^2}{2} & 0 & 0 \\ 0 & 0 & \sigma_p^2\frac{T_c^3}{3} & 0 & 0 & 0 & \sigma_p^2\frac{T_c^2}{2} & 0 \\ 0 & 0 & 0 & \sigma_\tau^2 T_c + \sigma_d^2\frac{T_c^3}{3} & 0 & 0 & 0 & \sigma_d^2\frac{T_c^2}{2} \\ \sigma_p^2\frac{T_c^2}{2} & 0 & 0 & 0 & \sigma_p^2 T_c & 0 & 0 & 0 \\ 0 & \sigma_p^2\frac{T_c^2}{2} & 0 & 0 & 0 & \sigma_p^2 T_c & 0 & 0 \\ 0 & 0 & \sigma_p^2\frac{T_c^2}{2} & 0 & 0 & 0 & \sigma_p^2 T_c & 0 \\ 0 & 0 & 0 & \sigma_d^2\frac{T_c^2}{2} & 0 & 0 & 0 & \sigma_d^2 T_c \end{bmatrix}$$

obtained as described in Section 4.2.

5.1.2 Linearization of the Measurement Equation

In the stand-alone PVT estimation problem based on the EKF, the measurement vector is defined as the column vector that stores the *pseudorange meaurements* ρ in meters and the *pseudorange rates measurements* r in meters per second

$$\mathbf{z}[n] = [\rho_1[n], \ \ldots, \rho_{N_{sat}}[n], r_1[n], \ \ldots, r_{N_{sat}}[n]]^T \tag{5.7}$$

where N_{sat} is the number of visible satellites.

Pseudoranges are related to the Euclidean distance between the satellites and the user and are defined as follows

$$\rho_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - \tau \tag{5.8}$$

where the triplet $[x_i,y_i,z_i]$ is the *i*-th satellite position while the triplet $[x,y,z]$ points out the position of the user. These are not *actual* ranges but *pseudo*-ranges, because they are distances that also accounts for the receiver clock misalignment τ, expressed in meters.

Unfortunately, the *geometric pseudorange* defined in (5.8) significantly differs from the measured pseudorange. This happens because the geometric pseudorange does not account for satellites clocks corrections and ionospheric and tropospheric errors that affect the received GNSS signal. Thus the measured *raw* pseudorange can be defined as follows [8]

$$\rho_i^{raw} = \rho_i + c\tau_i^{sat_{clock}} + e_i^{tropo} + e_i^{iono} + e^{loc} \tag{5.9}$$

where c is the speed of light, $\tau_i^{sat_{clock}}$ is the i-th satellite clock misalignment with respect to the GPS time in seconds, e_i^{tropo} is the delay due to the troposphere propagation in meters, e_i^{iono} is the delay due to the ionosphere propagation in meters and e^{loc} is a nuisance due to other local sources of error (e.g., multipath). These three different kinds of errors must be predicted and corrected to obtain ρ_i. It must be noticed that it is not possible to obtain a perfect prediction of the measured pseudorange values, because of a random uncorrelated error component and imprecise error models [8,38].

According to the EKF theory, the non-linear pseudorange equation (5.8) requires to be linearized using a first order Taylor series. The Jacobian of the non-linear relationship $\mathbf{h}[n]$ (5.8) between the user position and clock $\mathbf{x}^{(p)} \triangleq [x,y,z,\tau]^T$ and the N_{sat} pseudoranges $\rho_1, \dots \rho_{N_{sat}}$ happens to be

$$\frac{\partial \mathbf{h}[n]}{\partial \mathbf{x}^{(p)}} = \begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial z} & \frac{\partial h_1}{\partial \tau} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_1}{\partial z} & \frac{\partial h_1}{\partial \tau} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \frac{x-x_1}{d_1} & \frac{y-y_1}{d_1} & \frac{z-z_1}{d_1} & -1 \\ \frac{x-x_2}{d_2} & \frac{y-y_2}{d_2} & \frac{z-z_2}{d_2} & -1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{5.10}$$

where d_j is the norm of the vector $[x - x_j, y - y_j, z - z_j]^T$ (distance between the j-th satellite and the user). It is then possible to define the *measurement matrix*

$$\mathbf{H}_1[n] \triangleq \left[\frac{\partial \mathbf{h}[n]}{\partial \mathbf{x}^{(p)}}\right]_{\mathbf{x}=\check{\mathbf{x}}} = \begin{bmatrix} \frac{\check{x}-x_1}{d_1} & \frac{\check{y}-y_1}{d_1} & \frac{\check{z}-z_1}{d_1} & -1 \\ \frac{\check{x}-x_2}{d_2} & \frac{\check{y}-y_2}{d_2} & \frac{\check{z}-z_2}{d_2} & -1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{5.11}$$

which can be also written as

$$\mathbf{H}_1[n] = \begin{bmatrix} \breve{\mathbf{a}}_1[n] & \breve{\mathbf{a}}_2[n] & ... & \breve{\mathbf{a}}_{N_{sat}[n]} \\ -1 & -1 & ... & -1 \end{bmatrix}^T_{\mathbf{x}=\breve{\mathbf{x}}} \tag{5.12}$$

where $\breve{\mathbf{a}}_j[n] \triangleq \frac{1}{d_j[n]}[\breve{x}[n] - x_j[n], \breve{y}[n] - y_j[n], \breve{z}[n] - z_j[n]]^T$ is the unitary vector applied to the position of the user and pointing towards the j-th satellite in the time instant n. As a consequence, the measurement equation (4.47) referring to pseudoranges at the time instant n can be linearized as

$$\Delta\mathbf{z}^{(p)}[n] = \begin{bmatrix} \rho_1[n] - \breve{\rho}_1[n] \\ \vdots \\ \rho_{N_{sat}}[n] - \breve{\rho}_{N_{sat}}[n] \end{bmatrix} = \mathbf{H}_1[n]\Delta\mathbf{x}^{(p)}[n] + \boldsymbol{\nu}^{(p)}[n] \tag{5.13}$$

where $\breve{\rho}_j[n]$, $\forall j = 1, \dots N_{sat}$ are the nominal pseudoranges corresponding to the nominal trajectory where the linearization takes places, $\rho_j[n]$ are the measured pseudoranges after the correction (5.9) and $\boldsymbol{\nu}^{(p)}[n]$ is the component of the observation noise referred to positions. The prediction of the nominal pseudoranges $\breve{\rho}_1[n]$ is discussed in the next Subsection.

Pseudorange rates model the variations in time of the pseudoranges and they are strictly related to the relative motion between satellites and user and to the Doppler shift in the carrier frequency that this motion implies. The formula that connects the Doppler shift to the radial velocity between the user and the j-th satellite is

$$\frac{c(f_j - f_{T_j})}{f_{T_j}} + \mathbf{v}_j[n] \cdot \mathbf{a}_j[n] = \mathbf{v}[n] \cdot \mathbf{a}_j[n] - \frac{f_j v_\tau[n]}{f_{T_j}} \tag{5.14}$$

where $\mathbf{v}_j[n]$ is the three dimensional vector related to the velocity of the j-th satellite, $\mathbf{v}[n]$ is the three dimensional vector related to the velocity of the user, $v_\tau[n]$ is the clock drift in meters per second, f_{T_j} is the frequency transmitted by the j-th satellite and f_j is the Doppler-affected frequency of the j-th satellite that is observed by the user.

Observing that the ratio $f_j/f_{T_j} \approx 1$, (5.14) can be simplified into

$$\delta_j[n] = \mathbf{v}[n] \cdot \breve{\mathbf{a}}_j[n] - v_\tau[n] \tag{5.15}$$

where $\delta_j[n] = c(f_j - f_{T_j})/f_{T_j} + \mathbf{v}_j[n] \cdot \breve{\mathbf{a}}_j[n]$ is known and the nominal user position is assumed in $\breve{\mathbf{a}}_j[n]$.

If the velocity of the user is known, it is possible to estimate the pseudorange rate of every single satellite, which is equal to the radial component of the difference between the velocity of the *j*-th satellite and velocity of the user

$$r_j[n] = \breve{\mathbf{a}}_j[n]^T \cdot (\mathbf{v}[n] - \mathbf{v}_j[n]) - v_\tau[n]. \tag{5.16}$$

Considering the nominal pseudorange rate corresponding to the nominal trajectory as

$$\breve{r}_j[n] = \breve{\mathbf{a}}_j[n]^T \cdot (\breve{\mathbf{v}}[n] - \mathbf{v}_j[n]) - \breve{v_\tau}[n] \tag{5.17}$$

it is then possible to compute the incremental measurement on the pseudorange rate related to the *j*-th satellite as

$$r_j[n] - \breve{r}_j[n] = \breve{\mathbf{a}}_j[n]^T \cdot (\mathbf{v}[n] - \breve{\mathbf{v}}[n]) - (v_\tau[n] - \breve{v_\tau}[n]) \tag{5.18}$$

Then, the measurement equation (4.47) referring to pseudorange rates at the time instant n can be linearized as

$$\Delta\mathbf{z}^{(v)}[n] = \begin{bmatrix} r_1[n] - \breve{r}_1[n] \\ \vdots \\ r_{N_{sat}}[n] - \breve{r}_{N_{sat}}[n] \end{bmatrix} = \mathbf{H}_1[n]\Delta\mathbf{x}^{(v)}[n] + \boldsymbol{\nu}^{(v)}[n] \tag{5.19}$$

where $\Delta\mathbf{x}^{(v)}[n] = [\Delta v_x[n], \Delta v_y[n], \Delta v_z[n], \Delta v_\tau[n]]^T$ and $\boldsymbol{\nu}^{(v)}[n]$ is the component of the measurement noise referred to velocities.

As a consequence of the above considerations, the complete *system of linearized observation equation* can be written putting together (5.13) and (5.19) as

$$\Delta\mathbf{z}[n] = \begin{bmatrix} \mathbf{H}_1[n] & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1[n] \end{bmatrix} \Delta\mathbf{x}[n] + \boldsymbol{\nu}[n] \tag{5.20}$$

where it is possible to define the complete *observation matrix* for this problem

$$\mathbf{H}[n] \triangleq \begin{bmatrix} \mathbf{H}_1[n] & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1[n] \end{bmatrix} \tag{5.21}$$

where $\mathbf{H}_1[n]$ is defined in (5.11).

5.1.3 Pseudorange and Pseudorange Rate Prediction

The nominal pseudorange can be predicted starting from the nominal position of the user, namely $\breve{x}[n]$, $\breve{y}[n]$, $\breve{z}[n]$, $\breve{\tau}[n]$, to compute a nominal pseudorange free from propagation errors

$$\breve{\rho}_i^{ef}[n] = \sqrt{(x_i[n] - \breve{x}[n])^2 + (y_i[n] - \breve{y}[n])^2 + (z_i[n] - \breve{z}[n])^2} - \breve{\tau}[n] \tag{5.22}$$

which is then corrected with the estimates of the errors considered in (5.9), in order to obtain a prediction of the nominal raw pseudoranges. These errors can be estimated using the data stored in the ephemerides and resorting to ionospheric and tropospheric error models, so that

$$\breve{\rho}_i[n] = \breve{\rho}_i^{ef}[n] + \hat{\tau}_i^{sat_{clock}}[n] + \hat{e}_i^{tropo}[n] + \hat{e}_i^{iono}[n], \quad \forall i = 1, \dots N_{sat} \tag{5.23}$$

where the hat $\hat{\cdot}$ points out an estimated quantity.

Similarly, the nominal pseudorange rates have to be predicted, resorting to the Doppler shift measurements provided by the receiver and the nominal trajectory, as stated in (5.17).

5.1.4 Error Covariance Matrices

The model used in the EKF structure requires the initialization of the PVT solution and of the covariance matrices of the model noise, $\mathbf{Q}[n]$, of the observation noise, $\mathbf{R}[n]$, and of the state error, $\mathbf{P}[n]$.

The *model noise covariance matrix* $\mathbf{Q}[n]$ is time invariant and has been written in (5.6).

The *observation noise covariance matrix* $\mathbf{R}[n] = \mathbb{E}\left\{\boldsymbol{\nu}[n]\boldsymbol{\nu}^H[n]\right\}$ can be assumed as diagonal and it is time varying. The diagonal structure indicates that every pseudorange and pseudorange rate measurement error is considered independent from the other errors. The temporal variation accounts for the possibility of modifying during time the "confidence" associated to the observation error of each satellite, depending on the current condition of the satellite. For instance, it is a common choice to weight each satellite using the inverse of the sine of its elevation, but other weights might be defined. A good choice is to consider a balanced contribution of elevation and C/N_0.

Finally, the *state estimation error covariance matrix* $\mathbf{P}[n] = \mathbb{E}\left\{\boldsymbol{\epsilon}[n]\boldsymbol{\epsilon}^H[n]\right\}$, where $\boldsymbol{\epsilon}[n]$ is the estimation error on $\Delta\mathbf{x}[n]$, as determined by the KF, is a time varying matrix that represents an index of correctness of the state. It must be initialized together with the state and its entries depend upon the level of knowledge of the initial state. It is a reasonable and conservative choice to initialize $\mathbf{P}[0]$ slightly greater than its expected value, as the KF algorithm will tune it while running.

5.2 INS/GNSS Integrated Systems

Despite GNSSs such as GPS provide the user with accurate estimates of his own position and velocity with a rate as high as 20 Hz, high buildings, tunnels, foliage, interference and many other obstacles and disturbances may alter the GPS signal, making the estimated position highly noisy (and therefore erroneous) or even unavailable.

To limit these detrimental effects, it is possible to rely upon other navigation systems that can be used together with satellite navigation receivers. INSs represent perfect candidates to implement an integrated system together with the GPS (in general, GNSS), thanks to their complementary characteristic [39,40], which are resumed in Table 5.1.

Table 5.1. Pro's and con's of INSs and GNSSs

System	Pro's	Con's
INS	Self-contained Insensitive to external sources of errors (intentional or non-intentional, interference, environment) High rate Small errors in a short time	Unlimited error growth (drifts)
GNSS	Bounded error in any instant	Not self-contained Sensitive to jamming Environment dependent

Three conceptual approaches can be identified for integrating INS-based positioning and GNSS-based positioning (*system hybridization*):

1. *loose integration;*
2. *tight integration;*
3. *ultra-tight integration.*

These solutions differ for the degree of integration of the two systems, i.e. for the nature of the information extracted from the two systems and used in the hybridization process, as well as for the architecture of the interactions between the two systems. Furthermore we believe that it is worth considering the terms integration and coupling as different, in order to better define different kinds of systems. In the following, with *integration* we define a system where the INS measurements are fed to an "hybridization

engine" (usually, a KF) and corrected by means of different kinds of GNSS measurements. On the other hand, with the term *coupling* we point out a peculiar feature of the system, which is the employment of the corrected PVT solution to drive the NCOs that are present in the PLL and in the DLL.

In the following sections, the algorithmic scheme that realizes every integrated system will be analyzed in details. The mathematical tool that enables such an integration is the Kalman Filter (KF), thanks to its ability to blend different sources of noisy measurements in a single state-space description (*model*) of the evolution of the system. Although other mathematical tools for the system hybridization can be envisaged [41–43], the KF is historically and conceptually the principal approach.

The three hybridization architectures between GNSS and INS considered can share in principle the same basic state-space model of the involved quantities. Minor differences can be foreseen in the number of the involved states and in the forcing functions models; on the other hand, substantial differences determine the three observation models, which lead to substantially different integration strategies.

The following Section 5.3 offers a detailed description of the commonly adopted state-space model in the specific context of the loose integration architecture, but it is important to understand that the same model will be adopted also in the tight and ultra-tight architectures with minor differences. On the contrary, each observation model is peculiar for the specific architecture and will be properly discussed in dedicated subsections.

5.3 General Architecture for the Loose Integration

A loosely-integrated system uses the *trajectory* measured from the GNSS receiver to compute the *corrections* to be applied to the trajectory estimated by the INS computer and to estimate, if necessary, the biases that affect the accelerometers and the gyroscopes. Loose integration is based on the definition of a state-space model of the hybrid system and the application of an EKF to compute the corrections necessary to refine the INS-based trajectory.

It is worth noticing that *in the loose coupling case the GNSS information is used as a refinement of the INS information*: the GNSS information is used to counteract the intrinsic derivation of the INS solution, correcting the INS trajectory. Such corrections are computed via an EKF, whose state-space model is described hereafter.

The architecture of a loose integration can be represented as in Figure 5.2: the IMU

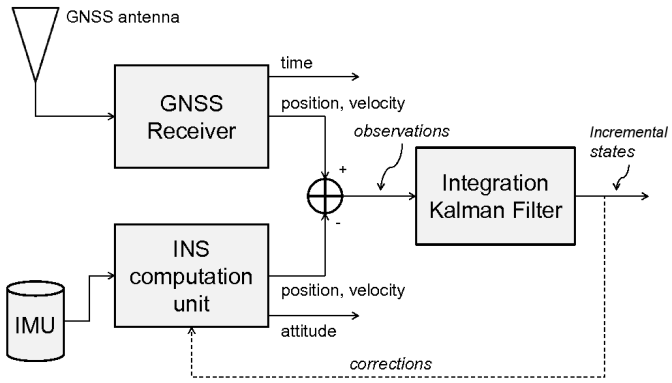

Figure 5.2. Block scheme for a loosely coupled GPS/INS system.

measurements are used to predict the position, velocity and attitude by means of the navigation equations in the INS block. The INS-predicted trajectory is then employed to feed the KF, that uses the differences in trajectory (GPS-measured minus INS-predicted) as its observations.

In the following, the body trajectory will be expressed in the ECEF frame and every vector in this frame will be labeled with the supercript $\cdot^e$. Other references such ENU, North-East-Down (NED) or wander azimuth can be adopted. The output of the IMU is given in the body frame, labeled by the superscript ^{b}.

5.3.1 Loose Integration: State-Space Model

The definition of the state-space model needed to implement a loosely integrated system for GNSS-INS hybridization is shown hereafter. As discusssed before, this model is common for *all* the integration architectures, therefore it will be recalled with minor changes in Section 5.4 and 5.5. The equations describing the model are initially written in the continuous-time domain, then they are translated in the discrete-time domain.

In order to implement a KF-based integration architecture, the set of the system states (*incremental states*) has to be identified at first. The structure adopted here is the following

$$\Delta\mathbf{x}(t) = \left[\Delta\mathbf{p}^e(t)^T, \Delta\mathbf{v}^e(t)^T, \Delta\boldsymbol{\psi}^e(t)^T, \mathbf{b_a}^b(t)^T, \mathbf{b_g}^b(t)^T\right]^T \in \mathbb{R}^{15,1} \tag{5.24}$$

where the superscripts ^{e} and ^{b} point out *Earth frame* and *body frame*, respectively. The state vector stores the following components

$\Delta\mathbf{p}^e(t) \in \mathbb{R}^{3,1}$ is the *corrections vector* to be applied to the *nominal position of the user* at the time instant t, expressed in the Earth frame;

$\Delta\mathbf{v}^e(t) \in \mathbb{R}^{3,1}$ is the *corrections vector* to be applied to the *nominal velocity of the user* at the time instant t, expressed in the Earth frame;

$\Delta\boldsymbol{\psi}^e(t) \in \mathbb{R}^{3,1}$ is the vector of misalignment around each axis (*attitude corrections*) at the time instant t, expressed in the Earth frame;

$\mathbf{b_a}^b(t) \in \mathbb{R}^{3,1}$ is the vector of the biases of the accelerometers at the time instant t, expressed in the body frame;

$\mathbf{b_g}^b(t) \in \mathbb{R}^{3,1}$ is the vector of the biases of the gyroscopes at the time instant t, expressed in the body frame;

The evolution in time of the elements stored in the state vector is ruled by a set of differential equations, namely the *state-transition model,* that are discussed in the following.

Space Equation

The time evolution of the position error $\Delta\mathbf{p}^e(t)$ is ruled by the following equation

$$\Delta\dot{\mathbf{p}}^e(t) = \Delta\mathbf{v}^e(t) \tag{5.25}$$

which is the canonical differential equation that rules the dependency between velocity and space.

Velocity Equation

The description of the time evolution of the three dimensional velocity vector requires to resort to the inertial model. Referring to the Coriolis theorem discussed in Section 3.6, it is possible to relate the inertial frame and the ECEF frame velocity variations by means of the following equation, obtained differentiating (3.2) with respect to time [34]

$$\dot{\mathbf{v}}^e(t) = \dot{\mathbf{v}}^i(t) - \boldsymbol{\omega}^e_{ie} \times \mathbf{v}^e(t) \tag{5.26}$$

The velocity variation $\dot{\mathbf{v}}^i(t)$ in the inertial frame can be written as in (3.1)

$$\ddot{\mathbf{p}}^i(t) = \mathbf{f}(t) + \mathbf{g}(t) \tag{5.27}$$

which, substituted into (3.3), gives

$$\dot{\mathbf{v}}^i(t) = \mathbf{f}(t) - \boldsymbol{\omega}_{ie}^e \times \mathbf{v}^e(t) + \mathbf{g}_\ell(t) \tag{5.28}$$

where the term $\mathbf{f}(t)$ represents the applied specific force, $\boldsymbol{\omega}_{ie}^e \times \mathbf{v}^e(t)$ is the Coriolis acceleration and

$$\mathbf{g}_\ell(t) \triangleq \mathbf{g}(t) - \boldsymbol{\omega}_{ie}^e \times [\boldsymbol{\omega}_{ie}^e \times \mathbf{p}^e(t)] = \mathbf{g}(t) - \boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{p}^e(t) \tag{5.29}$$

is the so-called *local gravity vector,* representing the sum of the accelerations caused by the mass attraction force ($\mathbf{g}(t)$, the gravitational acceleration vector) and the centripetal acceleration due to Earth rotation.

Equation (5.28) can be substituted in (5.26), obtaining the differential equation describing the velocity variations in the earth frame

$$\begin{aligned} \dot{\mathbf{v}}^e(t) &= [\mathbf{f}(t) - \boldsymbol{\omega}_{ie}^e \times \mathbf{v}^e(t) + \mathbf{g}_\ell(t)] - \boldsymbol{\omega}_{ie}^e \times \mathbf{v}^e(t) = \\ &= \mathbf{f}(t) - 2\boldsymbol{\omega}_{ie}^e \times \mathbf{v}^e(t) + \mathbf{g}_\ell(t) \end{aligned} \tag{5.30}$$

Defining the vector storing the corrections of velocity $\Delta\mathbf{v}^e(t)$ as the difference between the true velocity and the nominal velocity obtained from the INS, i.e. $\Delta\mathbf{v}^e(t) = \dot{\mathbf{v}}^e(t) - \breve{\dot{\mathbf{v}}}^e(t)$, then (5.30) allows to express the evolution in time of the velocity correction as

$$\Delta\dot{\mathbf{v}}^e(t) = \left[\mathbf{C}_b^e \mathbf{f}^b - \breve{\mathbf{C}}_b^e \breve{\mathbf{f}}^b\right] - \left[2\boldsymbol{\Omega}_{ie}^e \mathbf{v}^e(t) - 2\breve{\boldsymbol{\Omega}}_{ie}^e \breve{\mathbf{v}}^e(t)\right] + [\mathbf{g}_\ell - \breve{\mathbf{g}}_\ell] \tag{5.31}$$

where the symbol $\breve{\cdot}$ refers to estimates provided by the INS, i.e. the nominal trajectory. The misalignment $\Delta\boldsymbol{\psi}^e(t)$ that affects the measured DCM $\breve{\mathbf{C}}_b^e$ with respect to the actual DCM $\mathbf{C}_b^e$ relates the two as follows [34]:

$$\breve{\mathbf{C}}_b^e = [\mathbf{I}_{3\times3} - \Delta\boldsymbol{\psi}^e(t)\times]\, \mathbf{C}_b^e \tag{5.32}$$

Substituting (5.32) in (5.31) one obtains

$$\begin{aligned} \Delta\dot{\mathbf{v}}^e(t) &= \mathbf{C}_b^e \mathbf{f}^b - \mathbf{C}_b^e \breve{\mathbf{f}}^b + \Delta\boldsymbol{\psi}^e(t) \times \mathbf{C}_b^e \breve{\mathbf{f}}^b - 2\boldsymbol{\Omega}_{ie}^e \Delta\mathbf{v}^e(t) + \Delta\mathbf{g}_\ell(t) \\ &= \mathbf{C}_b^e \left(\mathbf{f}^b - \breve{\mathbf{f}}^b\right) + \Delta\boldsymbol{\psi}^e(t) \times \breve{\mathbf{f}}^e - 2\boldsymbol{\Omega}_{ie}^e \Delta\mathbf{v}^e(t) + \Delta\mathbf{g}_\ell(t) \\ &= \mathbf{C}_b^e \mathbf{b_a}^b(t) - \mathbf{F}\Delta\boldsymbol{\psi}^e(t) - 2\boldsymbol{\Omega}_{ie}^e \Delta\mathbf{v}^e(t) + \Delta\mathbf{g}_\ell(t) \end{aligned} \tag{5.33}$$

where the bias on the accelerometers has been expressed as $\mathbf{b_a}^b(t) = \mathbf{f}^b - \breve{\mathbf{f}}^b$, the difference $\Delta\boldsymbol{\Omega}_{ie}^e = \boldsymbol{\Omega}_{ie}^e - \breve{\boldsymbol{\Omega}}_{ie}^e = 0$ due to the knowledge of the earth rotation rate and the mathematical equality (3.10) has been used to express $\Delta\boldsymbol{\psi}^e(t) \times \breve{\mathbf{f}} = -\mathbf{F}\Delta\boldsymbol{\psi}^e(t)$.

As far as the term $\Delta\mathbf{g}_\ell(t) = \mathbf{g}_\ell(t) - \breve{\mathbf{g}}_\ell(t)$ is concerned, it can be written explicitly resorting to the definition of the local gravity component (5.29). First, the dependence of the gravity vector $\mathbf{g}(t)$ on the position of the body with respect to the Earth can be expressed through the *tensor of the gravitational field* $\mathbf{\Gamma}$, namely

$$\mathbf{g}(t) = \mathbf{\Gamma}\mathbf{p}^e(t) \tag{5.34}$$

where $\mathbf{\Gamma}$ is assumed constant in time. Then, from (5.29) it is possible to write

$$\Delta\mathbf{g}_\ell(t) = \mathbf{\Gamma}\Delta\mathbf{p}^e(t) - \mathbf{\Omega}^e_{ie}\mathbf{\Omega}^e_{ie}\Delta\mathbf{p}(t) = \mathbf{N}^e\Delta\mathbf{p}^e(t) \tag{5.35}$$

The term $\mathbf{N}^e \triangleq \mathbf{\Gamma} - \mathbf{\Omega}^e_{ie}\mathbf{\Omega}^e_{ie}$ is called *tensor of gravity gradients* [14] and it includes the gravitational and centripetal components of the acceleration, so that $\mathbf{N}^e\Delta\mathbf{p}^e(t)$ is the position-dependent *local gravity component perturbation* affecting the acceleration. An expression for $\mathbf{N}^e$ can be found in [14].

This way, the differential equation governing the evolution of the incremental velocity $\Delta\mathbf{v}^e(t)$ can be written as

$$\Delta\dot{\mathbf{v}}^e(t) = \mathbf{N}^e\Delta\mathbf{p}^e(t) - 2\mathbf{\Omega}^e_{ie}\Delta\mathbf{v}^e(t) - \mathbf{F}\Delta\boldsymbol{\psi}^e(t) + \mathbf{C}^e_b\mathbf{b_a}^b(t) + \mathbf{C}^e_b\mathbf{w}_a(t) \tag{5.36}$$

where $\mathbf{w}_a(t)$ is a driving noise term acting on the accelerometers in the body frame.

Attitude Misalignment Equation

Equation (5.32) is the starting point for the derivation of the attitude misalignment equation and it can be rearranged as follows

$$\Delta\boldsymbol{\psi}^e(t)\times = \mathbf{I}_{3\times3} - \breve{\mathbf{C}}^e_b\mathbf{C}^{eT}_b \tag{5.37}$$

where it has been considered that every DCM is an orthonormal matrix. Differentiating (5.37) yields:

$$\Delta\dot{\boldsymbol{\psi}}^e(t)\times = -\breve{\mathbf{C}}^e_b\dot{\mathbf{C}}^{eT}_b - \dot{\breve{\mathbf{C}}}^e_b\mathbf{C}^{eT}_b \tag{5.38}$$

Recalling (3.6) and substituting the appropriate quantities in (5.38), we write

$$\begin{aligned}
\Delta\dot{\boldsymbol{\psi}}^e(t)\times &= -\breve{\mathbf{C}}^e_b\left[\mathbf{C}^e_b\mathbf{\Omega}^b_{eb}\right]^T - \left[\breve{\mathbf{C}}^e_b\breve{\mathbf{\Omega}}^b_{eb}\right]\mathbf{C}^{eT}_b \\
&= \hat{\mathbf{C}}^e_b\left[\mathbf{\Omega}^b_{eb} - \breve{\mathbf{\Omega}}^b_{eb}\right]\mathbf{C}^{eT}_b \\
&= \hat{\mathbf{C}}^e_b\Delta\mathbf{\Omega}^b_{eb}\mathbf{C}^{eT}_b
\end{aligned} \tag{5.39}$$

where the skew symmetric matrix property $\mathbf{A}^T = -\mathbf{A}$ has been exploited. Inserting (5.32) into the last row of (5.39) yields

$$\begin{aligned} \Delta\dot{\boldsymbol{\psi}}^e(t)\times &= [\mathbf{I}_{3\times 3} - \Delta\boldsymbol{\psi}^e(t)\times]\,\mathbf{C}_b^e\Delta\boldsymbol{\Omega}_{eb}^b\mathbf{C}_b^{eT} \\ &= \mathbf{C}_b^e\Delta\boldsymbol{\Omega}_{eb}^b\mathbf{C}_b^{eT} - \Delta\boldsymbol{\psi}^e(t)\times\mathbf{C}_b^e\Delta\boldsymbol{\Omega}_{eb}^b\mathbf{C}_b^{eT} \\ &\approx \mathbf{C}_b^e\Delta\boldsymbol{\Omega}_{eb}^b\mathbf{C}_b^{eT} \end{aligned} \tag{5.40}$$

since the products between incremental corrections result in small second order terms, which can be ignored. It is possible to observe that (5.40) can be expressed in vector form as [44]

$$\Delta\dot{\boldsymbol{\psi}}^e(t) = \mathbf{C}_b^e\Delta\boldsymbol{\omega}_{eb}^b \tag{5.41}$$

The rotation between the earth and the body frame can be regarded as the sum of two contributions namely the rotation of the body with respect to the inertial frame and the rotation of the ECEF with respect to the inertial frame, leading to

$$\boldsymbol{\omega}_{eb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_e^b\boldsymbol{\omega}_{ie}^e \tag{5.42}$$

Using (5.42) for both the true and the nominal rotations and recalling (5.32), it is easy to write

$$\begin{aligned} \Delta\boldsymbol{\omega}_{eb}^b &= \Delta\boldsymbol{\omega}_{ib}^b - (\mathbf{C}_e^b - \breve{\mathbf{C}}_e^b)\boldsymbol{\omega}_{ie}^e \\ &= \Delta\boldsymbol{\omega}_{ib}^b - (\Delta\boldsymbol{\psi}^e\times\mathbf{C}_b^e)\boldsymbol{\omega}_{ie}^e \end{aligned} \tag{5.43}$$

where the Earth rotation term $\boldsymbol{\omega}_{ie}^e$ has been considered perfectly known. Equation (5.43) can now be substituted into (5.41). Exploiting the properties of the skew-symmetric matrices, it is possible to obtain

$$\Delta\dot{\boldsymbol{\psi}}^e(t) = \mathbf{C}_b^e\left[\Delta\boldsymbol{\omega}_{ib}^b - (\Delta\boldsymbol{\psi}^e\times\mathbf{C}_b^e)\boldsymbol{\omega}_{ie}^e\right] \tag{5.44}$$

$$= \mathbf{C}_b^e\left[\Delta\boldsymbol{\omega}_{ib}^b + \mathbf{C}_e^b\Delta\boldsymbol{\psi}^e\times\boldsymbol{\omega}_{ie}^e\right] \tag{5.45}$$

$$= \mathbf{C}_b^e\Delta\boldsymbol{\omega}_{ib}^b + \Delta\boldsymbol{\psi}^e\times\boldsymbol{\omega}_{ie}^e \tag{5.46}$$

$$= \mathbf{C}_b^e\Delta\boldsymbol{\omega}_{ib}^b - \boldsymbol{\Omega}_{ie}^e\Delta\boldsymbol{\psi}^e \tag{5.47}$$

Now, expressing the bias on the gyroscopes as $\mathbf{b_g}^b(t) = -\Delta\boldsymbol{\omega}_{ib}^b$ and considering the presence of a driving noise $\mathbf{w}_g(t)$ on the gyros themselves, the differential equation governing the attitude misalignment is written as

$$\Delta\dot{\boldsymbol{\psi}}^e(t) = -\boldsymbol{\Omega}_{ie}^e\Delta\boldsymbol{\psi}^e(t) - \mathbf{C}_b^e\mathbf{b_g}^b(t) - \mathbf{C}_b^e\mathbf{w}_g(t) \tag{5.48}$$

Accelerometers Bias Equation

The accelerometers bias vector $\mathbf{b_a}^b(t)$ is modeled as a continuous time Gauss-Markov process

$$\dot{\mathbf{b}}_{\mathbf{a}}^b(t) = \mathbf{D}_a \mathbf{b_a}^b(t) + \mathbf{w}_{aa}(t) \tag{5.49}$$

where $\mathbf{D}_a$ is the time-constant diagonal matrix that defines a first-state Gauss-Markov model and $\mathbf{w}_{aa}(t)$ is the driving noise for the gyro biases, as represented in Figure 5.3 [14]. The matrix $\mathbf{D}_a$ stores in its diagonal the time constants γ_a of the Gauss-Markov model

$$[\mathbf{D}_a]_{ij} = \begin{cases} e^{-\gamma_a} - 1 \approx -\gamma_a, & i = j \\ 0, & i \neq j \end{cases} \tag{5.50}$$

The noise $\mathbf{w}_{aa}(t)$ is an Additive White Gaussian Noise (AWGN) process with zero mean and variance $\sigma_{aa}^2 = 2\sigma_{ba}^2\gamma_a$, where σ_{ba}^2 is the variance of the Gauss-Markov process.

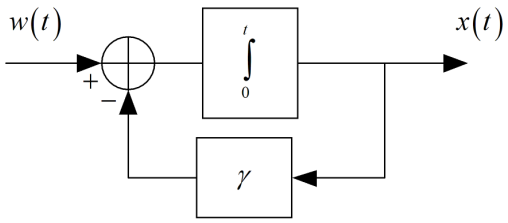

Figure 5.3. Block scheme for a Gauss-Markov continuous time system.

Gyroscopes Bias Equation

Analogously to $\mathbf{b_a}^b(t)$, the gyroscopes bias vector $\mathbf{b_g}^b(t)$ is modeled as a continuous time Gauss-Markov process

$$\dot{\mathbf{b}}_{\mathbf{g}}^b(t) = \mathbf{D}_g \mathbf{b_g}^b(t) + \mathbf{w}_{gg}(t) \tag{5.51}$$

where the matrix $\mathbf{D}_g$ stores in its diagonal the time constants $-\gamma_g$ of the Gauss-Markov model. The noise $\mathbf{w}_{gg}(t)$ is an AWGN process with zero mean and variance $\sigma_{gg}^2 = 2\sigma_{bg}^2\gamma_g$, being σ_{bg}^2 the variance of the Gauss-Markov process.

5.3.2 Loose Integration: State Transition Matrix

Once the equations that describe the system are determined, it is necessary to provide for the definition of the equivalent problem in the discrete time domain, to implement in software the discrete-time version of the complementary KF discussed in Chapter 4.

The differential equations discussed in Section 5.3.1 must be translated in the discrete time domain, in order to define the state transition matrix $\mathbf{\Phi}[n]$ of the discrete-time state-space model. Thus, the discrete-time approximation of the system (5.25)-(5.51) becomes

$$\Delta\mathbf{p}^e[n+1] = \Delta\mathbf{p}^e[n] + T_c\Delta\mathbf{v}^e[n] \tag{5.52}$$

$$\Delta\mathbf{v}^e[n+1] = \mathbf{N}^e\Delta\mathbf{p}^e[n] + (\mathbf{I}_3 - 2T_c\mathbf{\Omega}_{ie}^e)\,\Delta\mathbf{v}^e[n] - T_c\mathbf{F}[n]\Delta\boldsymbol{\psi}^e[n] + T_c\mathbf{C}_b^e[n]{\mathbf{b_a}}^b[n] + T_c\mathbf{C}_b^e[n]\mathbf{w}_a[n] \tag{5.53}$$

$$\Delta\boldsymbol{\psi}^e[n+1] = (\mathbf{I}_3 - T_c\mathbf{\Omega}_{ie}^e)\,\Delta\boldsymbol{\psi}^e[n] - T_c\mathbf{C}_b^e[n]{\mathbf{b_g}}^b[n] - T_c\mathbf{C}_b^e[n]\mathbf{w}_g[n] \tag{5.54}$$

$${\mathbf{b_a}}^b[n+1] = (\mathbf{I}_3 + T_c\mathbf{D}_a)\,{\mathbf{b_a}}^b[n] + T_c\mathbf{w}_{aa}[n] \tag{5.55}$$

$${\mathbf{b_g}}^b[n+1] = (\mathbf{I}_3 + T_c\mathbf{D}_g)\,{\mathbf{b_g}}^b[n] + T_c\mathbf{w}_{gg}[n] \tag{5.56}$$

so that the discrete-time state-space model is written as

$$\Delta\mathbf{x}[n+1] = \mathbf{\Phi}[n]\Delta\mathbf{x}[n] + \mathbf{G}[n]\mathbf{w}[n], \tag{5.57}$$

where

$$\mathbf{w}[n] = \left[\mathbf{w}_a(nT_c)^T, \mathbf{w}_g(nT_c)^T, \mathbf{w}_{aa}(nT_c)^T, \mathbf{w}_{gg}(nT_c)^T\right]^T \in \mathbb{R}^{12,1} \tag{5.58}$$

$$\mathbf{\Phi}[n] = \begin{bmatrix} \mathbf{I}_3 & T_c\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{N}^e & \mathbf{I}_3 - 2T_c\mathbf{\Omega}_{ie}^e & -T_c\mathbf{F}[n] & T_c\mathbf{C}_b^e[n] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 - T_c\mathbf{\Omega}_{ie}^e & \mathbf{0} & -T_c\mathbf{C}_b^e[n] \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 + T_c\mathbf{D}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 + T_c\mathbf{D}_g \end{bmatrix} \in \mathbb{R}^{15,15} \tag{5.59}$$

$$\mathbf{G}[n] = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ T_c\mathbf{C}_b^e[n] & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -T_c\mathbf{C}_b^e[n] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & T_c\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & T_c\mathbf{I}_3 \end{bmatrix} \in \mathbb{R}^{15,12} \tag{5.60}$$

Note that, comparing (5.57) with (4.9), here the stochastic forcing function is defined as $\boldsymbol{\eta}[n] = \mathbf{G}[n]\mathbf{w}[n]$, so that the model noise covariance matrix $\mathbf{Q}[n]$ in (4.31)-(4.32) must be modified accordingly:

$$\mathbf{Q}[n] = \mathbf{\Phi}[n]\mathbf{G}[n]\mathbf{Q}_w\mathbf{G}^T[n]\mathbf{\Phi}^T[n]T_c.$$

5.3.3 Loose Integration: Measurement Equation

On the contrary of the state transition model derived in the previous subsections, the observation model defined for the loosely integrated architecture is peculiar for this specific approach.

The incremental observation vector $\Delta\mathbf{z}[n] = \mathbf{z}[n] - \breve{\mathbf{z}}[n]$ in (4.47) for the complementary KF used to implement the loose integration is defined through the terms

$\mathbf{z}[n] = \left[\mathbf{p}^e[n]^T \mathbf{v}^e[n]^T\right]^T \in \mathbb{R}^{6,1}$, vector of the GPS-measured trajectory at time n;

$\breve{\mathbf{z}}[n] = \left[\breve{\mathbf{p}}^e[n]^T \breve{\mathbf{v}}^e[n]^T\right]^T \in \mathbb{R}^{6,1}$, INS-estimated trajectory at time n.

The predicted observation $\Delta\mathbf{z}^-[n]$ related to the state space model is obtained from the a-priori incremental state $\Delta\mathbf{x}^-[n] = \mathbf{\Phi}[n]\Delta\mathbf{x}[n-1]$ as

$$\Delta\mathbf{z}^-[n] = \mathbf{H}_{lo}[n]\Delta\mathbf{x}^-[n] \tag{5.61}$$

In the case of a loose integrated architecture, the observation matrix $\mathbf{H}_{lo}[n]$ is constant in time and evidently equal to

$$\mathbf{H}_{lo} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 6} \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 6} \end{bmatrix} \in \mathbb{R}^{6,15} \tag{5.62}$$

Therefore it is possible to compute the innovation $\boldsymbol{\alpha}[n]$ in (4.53) as

$$\boldsymbol{\alpha}[n] = (\mathbf{z}[n] - \breve{\mathbf{z}}[n]) - \mathbf{H}_{lo}\Delta\mathbf{x}^-[n] \tag{5.63}$$

With the above definitions, the complementary KF iterations outlined in Section 4.4 can be used to implement the loose integration algorithm.

5.4 General Architecture for the Tight Integration

A tightly-coupled system uses the *pseudorange and pseudorange rate information* extracted from the GNSS receiver to compute the *corrections* to be applied to the trajectory estimated by the INS computer and to estimate, if necessary, the biases that affect the accelerometers and the gyroscopes. Tight coupling is based on the definition of a state-space model of the hybrid system and on the application of an EKF to compute the corrections necessary to refine the INS-based trajectory.

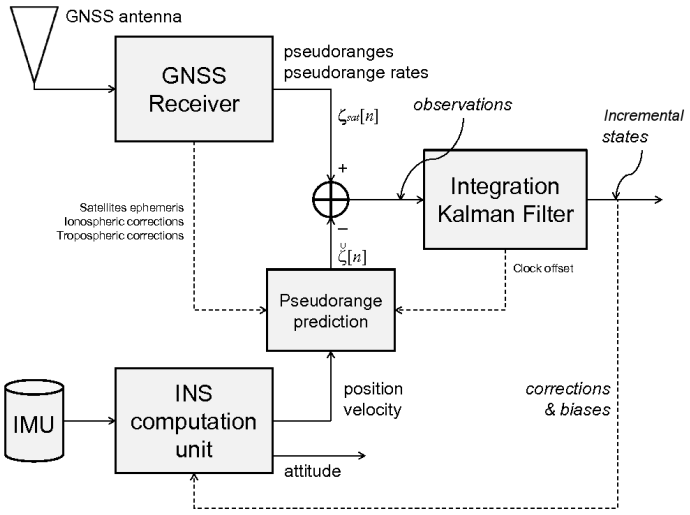

Figure 5.4. Block scheme for a tightly coupled GPS/INS system.

It is worth noticing that the GNSS information is used as a refinement of the INS information: the GNSS information is used to counteract the intrinsic derivation of the INS solution, correcting the INS trajectory. Such corrections are computed via an EKF, whose state-space model is described hereafter.

The architecture of a tight integration can be represented as in Figure 5.5: the measurements of the IMU are used to predict the position, velocity and attitude by means of the navigation equations in the INS block. The INS-predicted trajectory is then employed in the feedback branch to predict the pseudoranges of all visible satellites. The Kalman filter uses the differences in pseudoranges (GPS-measured minus INS-predicted) as its observations.

5.4.1 Tight Integration: State-Space Model

As done in the previous Section, in order to implement a tightly coupled architecture, the set of the system states (*incremental states*) has to be firstly identified. The structure considered here is the following

$$\Delta\mathbf{x}(t) = \left[\Delta\mathbf{p}^e(t)^T, \Delta\tau(t), \Delta\mathbf{v}^e(t)^T, \Delta v_\tau(t), \Delta\boldsymbol{\psi}^e(t)^T, \mathbf{b_a}^b(t)^T, \mathbf{b_g}^b(t)^T\right]^T \in \mathbb{R}^{17,1} \quad (5.64)$$

It can be easily noticed that the state vector (5.64) for tightly coupled systems is identical to (5.24), previously set for the loosely coupled system, plus two additional components

$\Delta\tau(t) \in \mathbb{R}^{1,1}$ is the receiver clock bias expressed in meters;

$\Delta v_\tau(t) \in \mathbb{R}^{1,1}$ is the receiver clock drift expressed in meters per second.

The evolution in time of the new elements stored in the state vector is ruled by the differential equations presented hereafter.

Clock Misalignment Equation

The time evolution of the clock misalignment error $\Delta\tau(t)$ is ruled by

$$\dot{\Delta\tau}(t) = \Delta v_\tau(t) \tag{5.65}$$

whose differential structure is consistent with (5.25).

Clock Drift Equation

The clock drift error $\Delta v_\tau(t)$ evolution is described by the following equation

$$\dot{\Delta v_\tau}(t) = v_\tau(t) \tag{5.66}$$

where $v_\tau(t)$ is a white noise. Equation (5.66) defines therefore a random walk upon clock drift.

5.4.2 Tight Integration: State Transition Matrix

The tightly coupled system is characterized by the same transition matrix of the loosely coupled system plus two more line to characterize the clock bias and drift. The equations in the discrete time domain are defined starting from (5.65) and (5.66)

$$\Delta\tau[n+1] = \Delta\tau[n] + \Delta v_\tau[n] T_c \tag{5.67}$$

$$\Delta v_\tau[n+1] = \Delta v_\tau[n] + v_\tau T_c \tag{5.68}$$

It is now easy to determine the matrices we are interested in

$$\mathbf{\Phi}[n] = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times1} & T_c\mathbf{I}_3 & \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_{1\times3} & 1 & \mathbf{0}_{1\times3} & T_c & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{N}^e & \mathbf{0}_{3\times1} & \mathbf{I}_3 - 2T_c\mathbf{\Omega}^e_{ie} & \mathbf{0}_{3\times1} & -T_c\mathbf{F}[n] & T_c\mathbf{C}^e_b[n] & \mathbf{0}_3 \\ \mathbf{0}_{1\times3} & 0 & \mathbf{0}_{1\times3} & 1 & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_3 & \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_{3\times1} & \mathbf{I}_3 - T_c\mathbf{\Omega}^e_{ie} & \mathbf{0} & -T_c\mathbf{C}^e_b[n] \\ \mathbf{0}_3 & \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{I}_3 + T_c\mathbf{D}_a & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 + T_c\mathbf{D}_g \end{bmatrix} \tag{5.69}$$

where $\mathbf{\Phi}[n] \in \mathbb{R}^{17,17}$ and

$$\mathbf{G}[n] = \begin{bmatrix} \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ 0 & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{3\times1} & T_c\mathbf{C}_b^e[n] & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ T_c & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{3\times1} & \mathbf{0}_3 & -T_c\mathbf{C}_b^e[n] & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_3 & T_c\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & T_c\mathbf{I}_3 \\ \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_{3\times1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \in \mathbb{R}^{17,13} \tag{5.70}$$

with the definition of the model noise vector $\mathbf{w}[n] = \left[\upsilon_\tau[n], \mathbf{w}_a[n]^T, \mathbf{w}_g[n]^T, \mathbf{w}_{aa}[n]^T, \mathbf{w}_{gg}[n]^T\right]^T$, $\mathbf{w}[n] \in \mathbb{R}^{13,1}$.

5.4.3 Tight Integration: Measurement Equation

The incremental observation vector of the complementary KF, $\Delta\mathbf{z}[n]$, is defined as

$$\Delta\mathbf{z}[n] = \boldsymbol{\zeta}_{sat}[n] - \hat{\boldsymbol{\zeta}}[n] \tag{5.71}$$

where

$\boldsymbol{\zeta}_{sat}[n] = \left[\boldsymbol{\rho}[n]^T\ \mathbf{r}[n]^T\right]^T \in \mathbb{R}^{2N_{sat},1}$ is the vector of the pre-corrected ([38]) GPS-measured pseudoranges $\boldsymbol{\rho}[n]$ and pseudorange rates $\mathbf{r}[n]$ at the time instant n;

$\hat{\boldsymbol{\zeta}}[n] = \left[\hat{\boldsymbol{\rho}}[n]^T\ \hat{\mathbf{r}}[n]^T\right]^T$ is the predicted pseudorange and pseudorange rate vector computed from the current estimate of the target trajectory.

The predicted incremental observation, dependent upon the state-space model only, is written from the a-priori state $\Delta\mathbf{x}^-[n] = \mathbf{\Phi}[n]\Delta\mathbf{x}[n-1]$ as follows

$$\Delta\mathbf{z}^-[n] = \mathbf{H}_{ti}[n]\Delta\mathbf{x}^-[n] \tag{5.72}$$

where the observation matrix $\mathbf{H}_{ti}[n]$ is time varying and depends upon the GNSS only observation matrix $\mathbf{H}[n]$ in (5.21), being equal to

$$\mathbf{H}_{ti}[n] = \begin{bmatrix} \mathbf{H}[n] & \mathbf{0}_{N_{sat}\times3} & \mathbf{0}_{N_{sat}\times3} & \mathbf{0}_{N_{sat}\times8} \\ \mathbf{0}_{N_{sat}\times3} & \mathbf{H}[n] & \mathbf{0}_{N_{sat}\times3} & \mathbf{0}_{N_{sat}\times8} \end{bmatrix} \in \mathbb{R}^{2N_{sat},17} \tag{5.73}$$

Therefore, combining (5.75) and (5.76), it is possible to compute the innovation $\boldsymbol{\alpha}[n]$

$$\boldsymbol{\alpha}[n] = \Delta\mathbf{z}[n] - \mathbf{H}_{ti}[n]\Delta\mathbf{x}^{-}[n] \tag{5.74}$$

As in the case of loose coupling, with the above definitions the complementary KF iterations outlined in Section 4.3 can be used to solve the system of equations related to the tight coupling integration.

5.5 General Architecture for the Ultra-Tight Integration

It must be stated that the term ultra-tight coupling (or integration) may point out different kind of architectures.

In the following, an ultra-tightly-coupled system is considered as a system that uses the *quadrature* and *in-phase* information extracted from the GNSS receiver PLLs to compute the *corrections* to be applied to the trajectory estimated by the INS computer, as well as to estimate, if necessary, the biases that affect the accelerometers and the gyroscopes.

Also ultra-tight coupling is based on the definition of a state-space model of the hybrid system and on the application of an EKF to compute the corrections necessary to refine the INS-based trajectory.

It is worth noticing that in the ultra-tight coupling case the GNSS information is used as a refinement of the INS information: the GNSS information is used to counteract the intrinsic derivation of the INS solution, correcting the INS trajectory. Such corrections are computed via an EKF, whose state-space model is described hereafter.

The considered ultra-tight architecture can be represented as in Figure 5.5: the measurements provided by the IMU are used to predict the position, velocity and attitude by means of the navigation equations in the INS block. The INS-predicted trajectory is then employed in the feedback branch to predict the quadrature and in-phase measurements of the PLLs tracking each channel. The EKF uses the differences (incremental states) between the measurements and the predicted values (GPS-measured minus INS-predicted) as its observations.

Nonetheless, we stress that it is possible to consider as an ultra tightly coupled system any kind of GNSS and INS integration that exploits the INS corrected solution to drive the PLL oscillators.

This solution allows the GNSS receiver to remove part of the Doppler shift due to the user dynamics from the received signal, so that the PLLs' bandwidth can be narrowed to

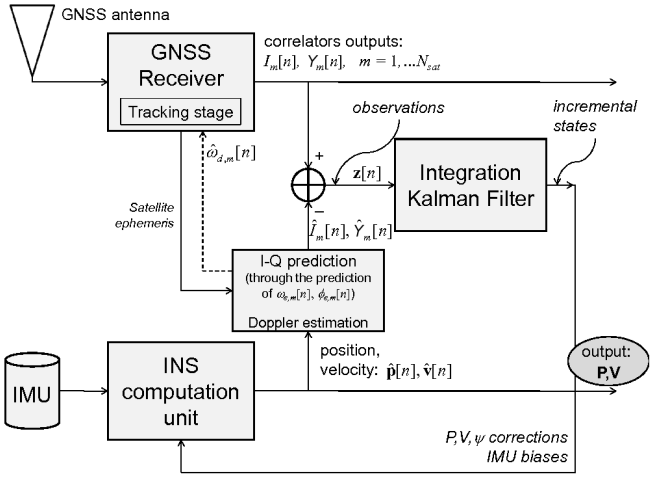

Figure 5.5. Block scheme for an ultra-tightly coupled GPS/INS system.

a couple of Hz, providing a more accurate GNSS solution.

This approach allows to consider as an ultra-tightly coupled system even a loosely-coupled GNSS-INS system, if the INS solution is fed back to the PLLs' oscillators. However, this type of architecture will not be considered in the following.

5.5.1 Ultra-Tight Integration: State-Space Model

For the state-space model of the ultra tight coupled system, the reader can refer to the one of the loosely coupled system. The two state-space models are in fact the same.

5.5.2 Ultra-Tight Integration: State Transition Matrix

For the state transition matrix of the ultra tight coupled system, the reader can refer to the one of the loosely coupled system. The two state transition matrices are in fact the same.

5.5.3 Ultra-Tight Integration: Measurement Equation

The incremental observation vector of the EKF, $\Delta\mathbf{z}[n]$, is defined as

$$\Delta\mathbf{z}[n] = \boldsymbol{\zeta}_{sat}[n] - \hat{\boldsymbol{\zeta}}[n] \tag{5.75}$$

where

$\boldsymbol{\zeta}_{sat}[n] = \left[\mathbf{Y}[n]^T\ \mathbf{I}[n]^T\right]^T \in \mathbb{R}^{2N_{sat},1}$ is the vector of the GPS-measured quadrature $\mathbf{Y}[n]$ and in-phase $\mathbf{I}[n]$ signals at the output of the correlators at the time instant n;

$\hat{\boldsymbol{\zeta}}[n] = \left[\hat{\mathbf{Y}}[n]^T\ \hat{\mathbf{I}}[n]^T\right]^T$ is the predicted quadrature and in-phase signal vector at the output of the correlators, computed from the current estimate of the trajectory of the target.

The measurement equation that determines the relationship between the state-error vector and the measurement prediction error vector is then

$$\Delta\mathbf{z}[n]^- = \mathbf{H}_{ul}[n]\Delta\mathbf{x}^-[n] \tag{5.76}$$

where the observation matrix $\mathbf{H}_{ul}[n]$ is time varying and must be computed at each time step. A possible way to define the entries of such a matrix can be found in [45], yielding to the following form:

$$\mathbf{H}_{ul}[n] = \begin{bmatrix} \mathbf{h}_1^T[n] \\ \vdots \\ \mathbf{h}_{N_{sat}}^T[n] \\ \mathbf{k}_1^T[n] \\ \vdots \\ \mathbf{k}_{N_{sat}}^T[n] \end{bmatrix} \in \mathbb{R}^{2N_{sat},15} \tag{5.77}$$

where, for the m-th satellite

$$\mathbf{h}_m[n] = \frac{\partial\mathbb{E}\{Y_m[n]\}}{\partial\phi_{e,m}}\frac{\partial\phi_{e,m}}{\partial\mathbf{x}} + \frac{\partial\mathbb{E}\{Y_m[n]\}}{\partial\omega_{e,m}}\frac{\partial\omega_{e,m}}{\partial\mathbf{x}} \tag{5.78}$$

$$\mathbf{k}_m[n] = \frac{\partial\mathbb{E}\{I_m[n]\}}{\partial\phi_{e,m}}\frac{\partial\phi_{e,m}}{\partial\mathbf{x}} + \frac{\partial\mathbb{E}\{I_m[n]\}}{\partial\omega_{e,m}}\frac{\partial\omega_{e,m}}{\partial\mathbf{x}} \tag{5.79}$$

$$\tag{5.80}$$

$\phi_{e,m}$ is the estimation error related to the received carrier phase and $\omega_{e,m}$ is the estimation error related to the Doppler angular rate (2π times the Doppler frequency).

Therefore, combining (5.75) and (5.76), it is possible to compute the innovation $\boldsymbol{\alpha}[n]$

$$\boldsymbol{\alpha}[n] = \Delta\mathbf{z}[n] - \mathbf{H}_{ul}[n]\Delta\mathbf{x}^-[n] \tag{5.81}$$

As in the case of loose coupling, with the above definitions the complementary KF iterations outlined in Section 4.3 solves the ultra-tight coupling integration problem.

5.6 Simulations and Results

In the following we will analyze the results related to the discussed topics.

5.6.1 Stand-Alone GNSS Kalman Based Receiver

The first experiment that we consider is related to the stand-alone GNSS receiver. We show two different scenarios, to underline the difference between a receiver based on a Kalman filter and another one based on a LS filter.

Figure 5.6. LS and Kalman based GNSS receivers - Low velocity.

The results for the first scenario are shown in Figure 5.6. The data collection is performed while moving with a car in the city of Turin, so that the speed is quite slow. The main advantage of the Kalman filter with respect to the LS filter is the presence of memory. Memory allows for a better filtering of the data, so that the present position depends upon the past position. The result is a smoother trajectory, as it can be seen in Figure 5.6.

Figure 5.7. LS and Kalman based GNSS receivers - High velocity.

For the sake of completeness, we also tested the algorithm in the presence of an high

dynamics vehicular scenario. In Figure 5.7 the results of the processing of the data collected on the highway from Bergamo to Milan are shown. The vehicle is traveling at an almost constant speed, around 130 Km/h. Also in this case, the Kalman filter allows for a smoother trajectory.

5.6.2 Simulated Integration Schemes

This Section is devoted to the discussion of some results obtained using MATLAB simulators. Every scenario will consider a two dimensional space where the user is moving. The navigation system is a simple GNSS-like system, where only the measurements of interest are simulated.

5.6.3 Loosely Coupling Simulator

A loosely-coupled system uses the reference trajectory measured from the GNSS receiver to compute the *corrections* to be applied to the trajectory estimated by the INS computer and to estimate, if necessary, the biases that affect the accelerometers and the gyroscopes. Loose coupling is based on the definition of a state-space model of the hybrid system and the application of an EKF to compute the corrections necessary to refine the INS-based trajectory. It is worth noticing that in the loose coupling case the GNSS information is used as a refinement of the INS information: the GNSS information is used to counteract the intrinsic derivation of the INS solution, correcting the INS trajectory.

The results of a two dimensional simulation are shown to compare the performance of a loosely coupled integrated system based on a Linearized Kalman Filter (LKF) and on an Extended Kalman Filter (EKF) , starting from the models described in the previous sections. The considered frame is inertial like.

Two accelerometers are simulated to collect the body translation while only a gyroscope is simulated to sense the rotational movements. The INS is simulated defining the acceleration and the angular rate, both affected by Gaussian random noise.

A normally distributed bias is added to simulate the drifting effects that affect the accelerometers. No bias has been considered for the output of the gyroscope. The output errors of the gyroscope are modeled as a Gauss Markov process, with a driving noise of 10deg/s and a time constant equal to 500s $(1-\sigma)$. The standard deviation of the bias random noise is equal to 0.001m/s^2.

The output of the GNSS receiver is simulated considering the actual position of the

user and adding white noise with a different standard deviation $(1-\sigma)$ on the x and z components (7 m and 10 m respectively). The velocity output is simulated considering a standard deviation $(1-\sigma)$ of 0.5m/s for the x component and 0.7m/s for the z component. The output rate is 1Hz, which is the usual output rate of most commercial receivers.

This simulation is useful to point out the different behavior of the two implementations:

- LKF uses the reference trajectory provided by the INS to linearize the navigation equations and it computes the corrections to be applied.
- EKF linearizes the navigation equation around the corrected trajectory, so that the reference trajectory is constantly updated by the corrections.

Figure 5.8 shows both the components of the trajectory to create the two dimensional representation. On the right of Figure 5.8, a zoomed picture allows to better analyze the estimated trajectories during the GNSS outage, which starts after 120s and lasts for 40s.

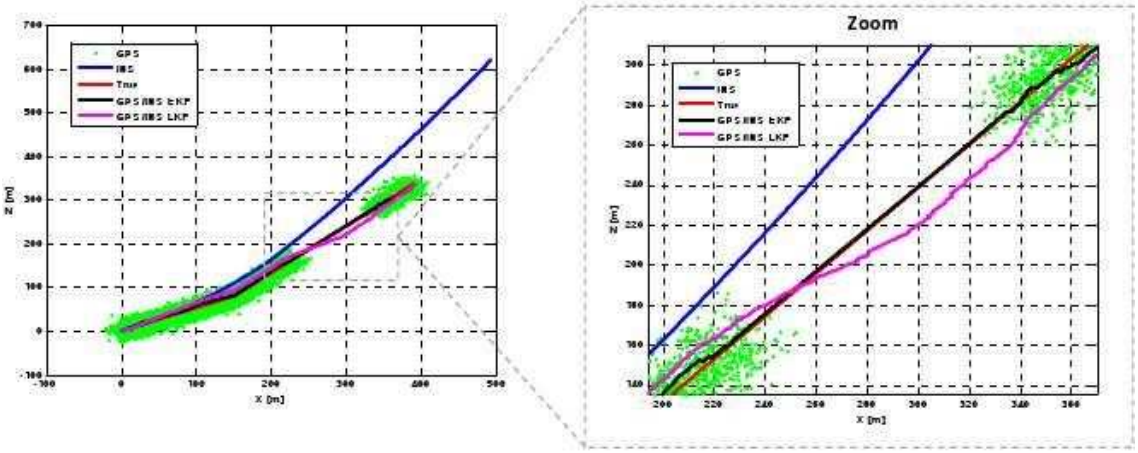

Figure 5.8. Loose integration: free INS, EKF, LKF and true trajectories.

Both filters show good smoothing performance and both the loose integration schemes manage to correct the drift of the INS. The errors of the two estimated trajectories with respect to the true trajectory are shown component by component in Figure 5.9.

Both LKF and EKF perform well when GNSS measurements are available, compensating the drift of the free running INS. In the event of a GNSS outage, the LKF relies only on the model to compute the corrections, while the EKF only accumulates the uncertainty on the model.

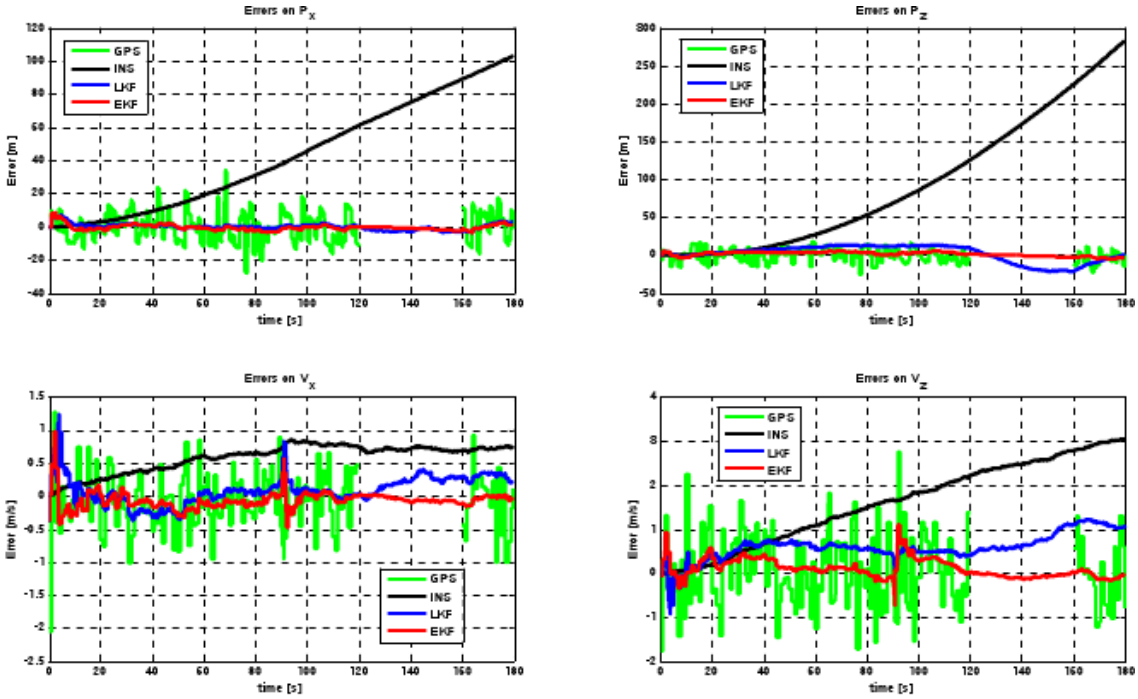

Figure 5.9. Position and velocity errors for the loose integration.

The EKF approach performs better than the LKF during a GNSS outage event. Also when the GNSS measurements are available, differences between the estimates of the EKF and LKF can be observed, corroborating the better performance of the EKF. This difference can be justified by the different approach adopted in the linearization of the navigation equations.

Figure 5.10 shows the correction estimated by the two filters. It is possible to notice how the correction applied to the INS output by the LKF grows with time. This phenomenon points out that the linearization point is far from the real trajectory and in this case higher order terms of the Taylor approximation are relevant.

The corrections computed by the EKF are instead restrained, meaning that the linearization point is close to the actual trajectory and only a first order Taylor approximation is sufficient to obtain a good estimate of the actual trajectory.

5.6.4 Tightly Coupling Simulator

The simulated system is based on measurements of pseudorange and pseudorange rate and on their estimation. These measurement are simulated considering the Euclidean

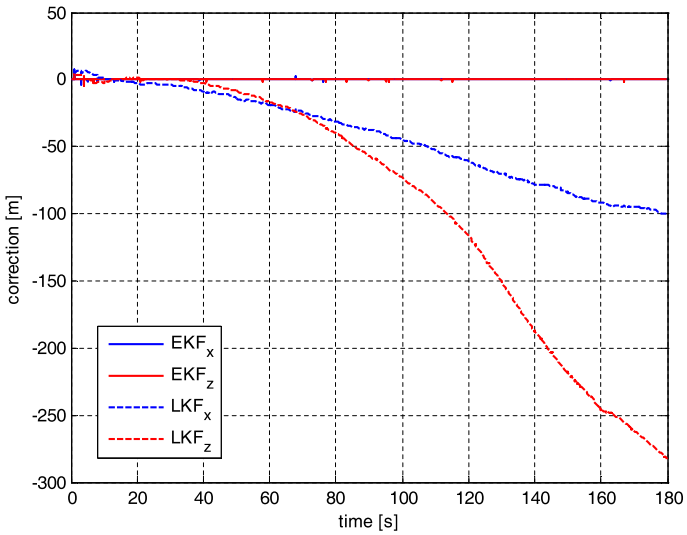

Figure 5.10. Corrections computed by the EKF and the LKF.

distance between the user and the satellite, the Doppler effect and the receiver clock bias and drift.

The measured pseudorange and pseudorange rate values must be related to the simulated GNSS receiver. They are computed starting from the actual trajectory corrupted by a 7m noise in position and 0.2m/s noise in velocity (1-σ). Other values can be chosen, referring to GNSS receivers data sheet. The output rate is again equal to 1Hz.

The estimation of the measurements is computed starting from the INS corrected trajectory. The INS is characterized by a Gauss Markov driving noise of 3m/s^3 with a time constant equal to 500s for the accelerometers, by a Gauss Markov driving noise of 8deg/s^2 with a time constant equal to 500s for the gyroscope (1-σ).

Moreover, a white noise of 0.5m/s^2 is added to the accelerometers Gauss Markov noise process and a white noise of 0.5deg/s is added to the gyroscopes Gauss Markov noise process (1-σ).

Once the measurements and their estimations are computed, the Kalman filter uses them to compute the innovation. The filter runs each time a GNSS measurement is available, so that the filter rate is equal to 1Hz. It must be noted that in simulation it is easy to synchronize the satellite based and the inertial based systems, while synchronization becomes a relevant issue when a real time system must be designed.

The simulated trajectory is a 5 minutes run. In Figure 5.11 the position of the user and of the satellites is shown.

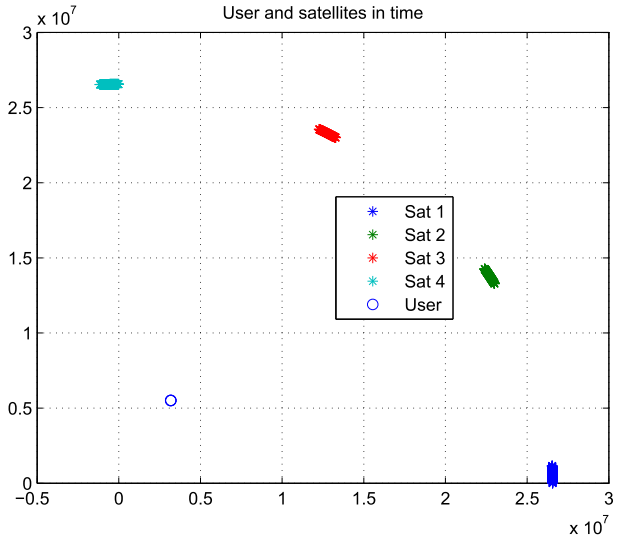

Figure 5.11. Satellites and user positions for the tight coupling.

The satellite trajectory is based on the Kepler laws. Of course, a two dimensional simulation limits the satellite motion along a meridian. The distances between the user and the satellites resemble the ones typical of GPS. In Figure 5.12 the user trajectory is shown.

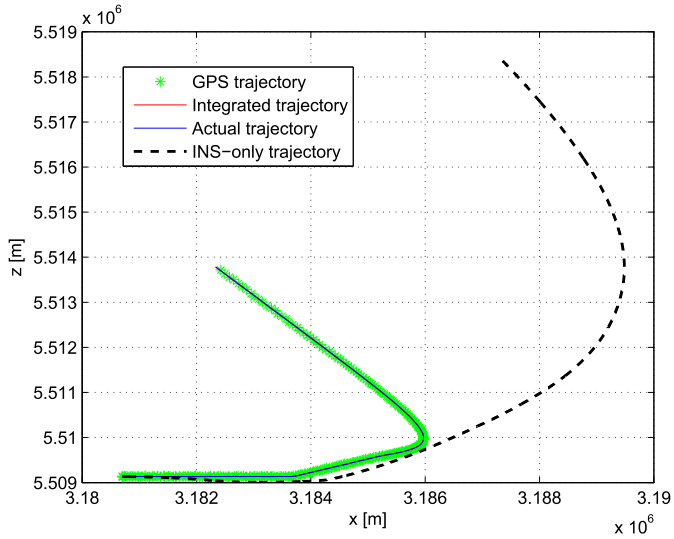

Figure 5.12. Considered trajectory for the tight coupling.

Also in this case, the user motion is constrained to a two dimensional plane, but it is

irrelevant when the algorithm must be tested. It is evident at a first glance that the error performed by the integrated system is bounded to a few meters, while an INS in dead reckoning performs an unbounded error. The shown trajectory is obtained integrating the acceleration shown in Figure 5.13

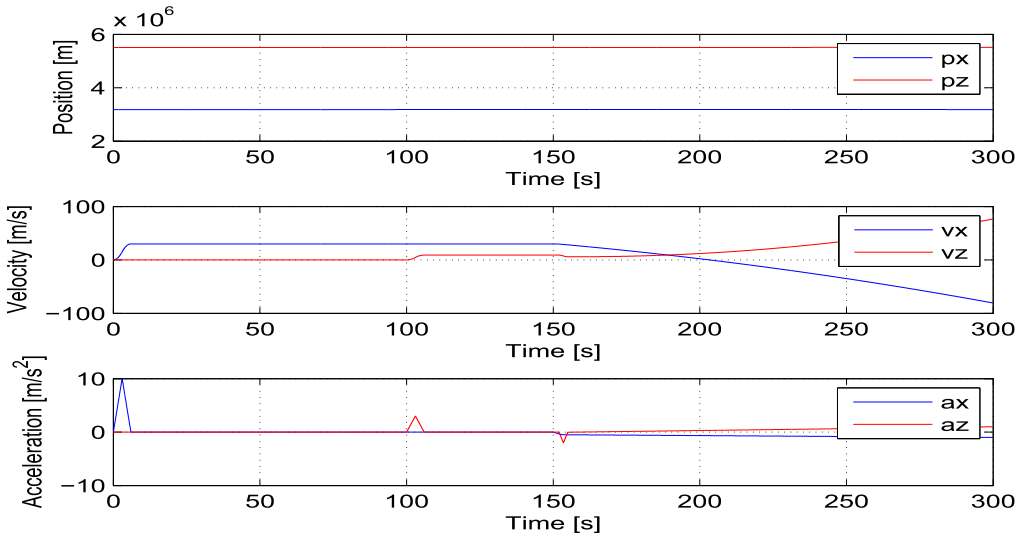

Figure 5.13. Position, velocity and acceleration for the tight coupling.

The x component of the acceleration grows up to 10m/s^2 in 3 seconds and then decreases to 0m/s^2 in 3 seconds. It then quickly grows to -0.5m/s^2 and finally it slowly grows to -1m/s^2. Once the first 100 seconds expired, the z component of the acceleration grows up to 3m/s^2 in 3 seconds and then decreases to 0m/s^2 in 3 seconds. Once 152 seconds passed from the beginning, the acceleration shows a spike of -2m/s^2 and finally it slowly grows to 1m/s^2. Velocity and position are computed starting from their initial value and integrating the accelerometers output. The trajectory estimation error is shown for both components in Figure 5.14.

From Figure 5.14 it is possible to appreciate how the integration of the two system by means of a Kalman filter allows to obtain a smooth trajectory. At the same time, the noisy GNSS trajectory is filtered and the INS trajectory, which is prone to diverge, is bounded. The velocity estimation error is shown for both components in Figure 5.15.

In this case, the velocity estimation provided from the two devices is similar, so that no improvements are obtained, so that the INS velocity estimation results bounded but no peculiar filtering effect can be noticed. This result is different from the one obtained for

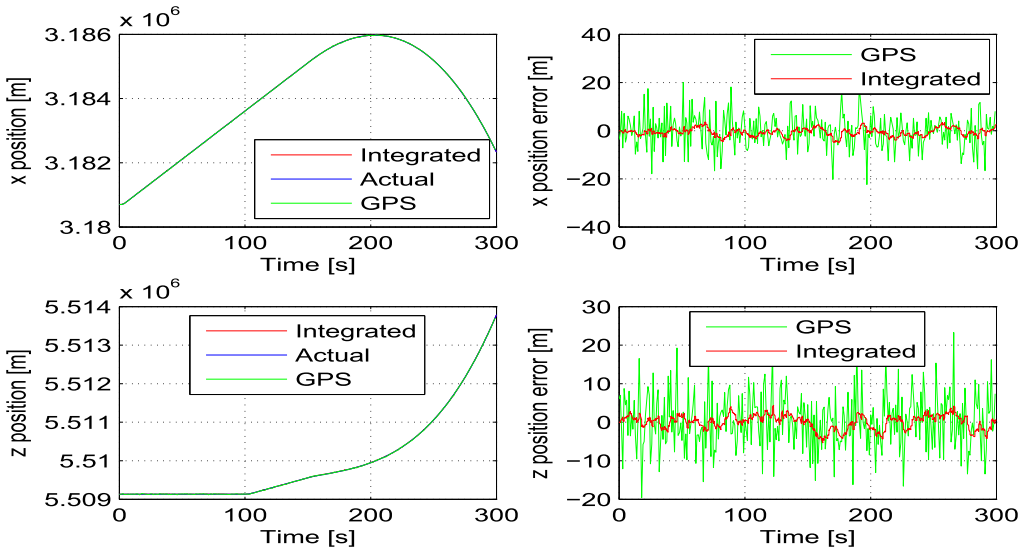

Figure 5.14. Position and related errors for the tight coupling.

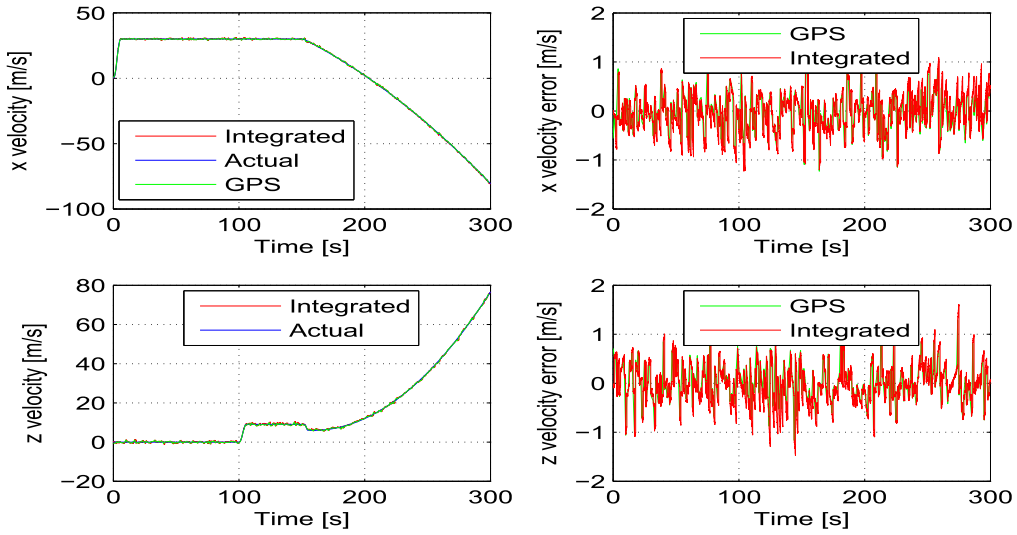

Figure 5.15. Velocity and related errors for the tight coupling.

the loose coupling scenario. The difference can be imputed to the different characteristic of the simulated GNSS and INS.

5.6.5 Ultra-Tightly Coupling Simulator

The simulated system is based on the measurement of $\mathbf{Y}$ and $\mathbf{I}$ and on their estimation. No CA-code is simulated.

It must be noticed that in an real GNSS receiver the subsystem implied in the $\mathbf{Y}$ and $\mathbf{I}$ computation is far more complex that the one considered here. By the way, for the sake of simplicity and to allow a simple discussion of the ultra-tight integration method, the measured $\mathbf{Y}$ and $\mathbf{I}$ values are computed starting from the actual trajectory, once it has been corrupted by a 7m noise in position and 0.2m/s noise in velocity (1-σ). The measurement rate is equal to 1KHz.

The estimation of the measurements is computed starting from the INS corrected trajectory. The INS is characterized by a Gauss Markov driving noise of 3 m/s^3 with a time constant equal to 500s for the accelerometers, by a Gauss Markov driving noise of 8deg/s^2 with a time constant equal to 500s for the gyroscope (1-σ).

Moreover, a white noise of 0.5m/s^2 is added to the accelerometers Gauss Markov noise process and a white noise of 0.5deg/s is added to the gyroscopes Gauss Markov noise process (1-σ).

The Kalman filter uses the estimate of the measurements to compute the innovation. The filter runs each 100ms, so that the filter rate is equal to 10Hz.

The simulated trajectory is a 100 seconds run. In Figure 5.16 the position of the user and of the satellites is shown.

All the considerations made in the Section related to the simulator for the tight integration hold. In Figure 5.17 the trajectory of the user is shown.

The final error performed by the integrated system is equal to 0.55m, while the INS in dead reckoning performs a 2006.72m final error. Also in this case, the trajectory estimated by the INS is characterized by an unbounded error. The uncertainty related to the GNSS trajectory is plotted starting from the points that generates the current $\mathbf{Y}$ and $\mathbf{I}$ measurements. The actual trajectory is obtained integrating the acceleration shown in Figure 5.18

The x component of the acceleration grows up to 10m/s^2 in 3s and then decreases to 0m/s^2 in 3s. It then quickly grows to -0.5m/s^2 and finally it slowly grows to -1m/s^2. Once the first 5s expired, the z component of the acceleration grows up to 5m/s^2 in 3s and then decreases to 0m/s^2 in 3s. Once 52s passed from the beginning, the acceleration shows a spike of -2m/s^2 and finally it slowly grows to 1m/s^2. Velocity and position are

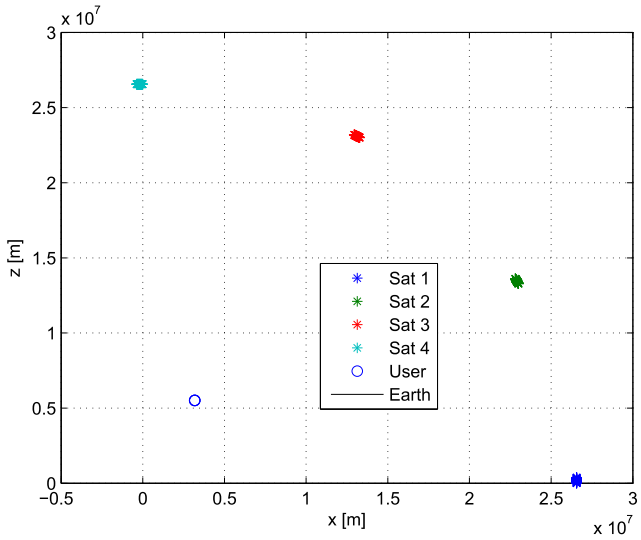

Figure 5.16. Satellites and user positions for the ultra-tight coupling.

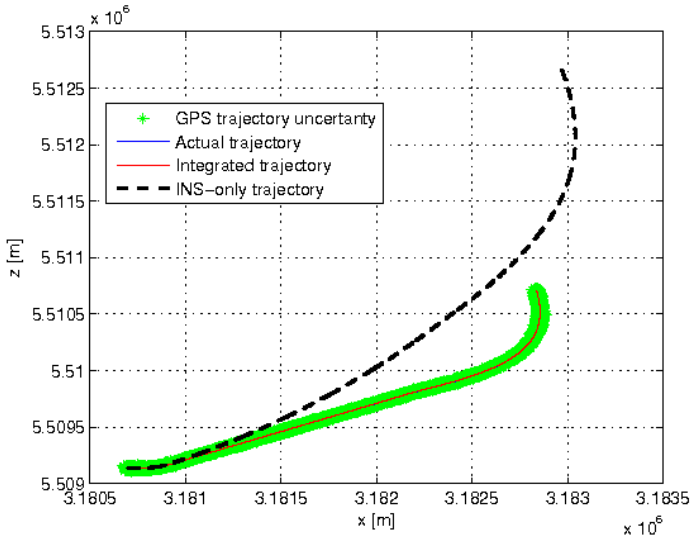

Figure 5.17. Considered trajectory for the ultra-tight coupling.

computed starting from their initial value and integrating the accelerometers output. The trajectory estimation error is shown for both components in Figure 5.19.

The error is small, if compared with the previous integration solutions. This is due to the precision of the measurements and of their estimates. Since the measurements rate is 100 times the one of the filter rate, the measurement considered in the Kalman filter is a

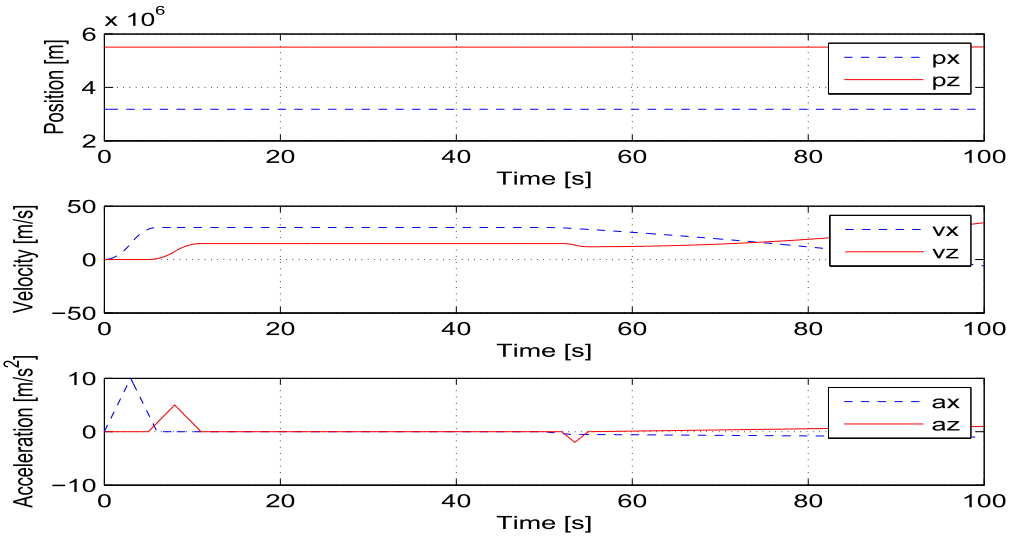

Figure 5.18. Position, velocity and acceleration for the ultra-tight coupling.

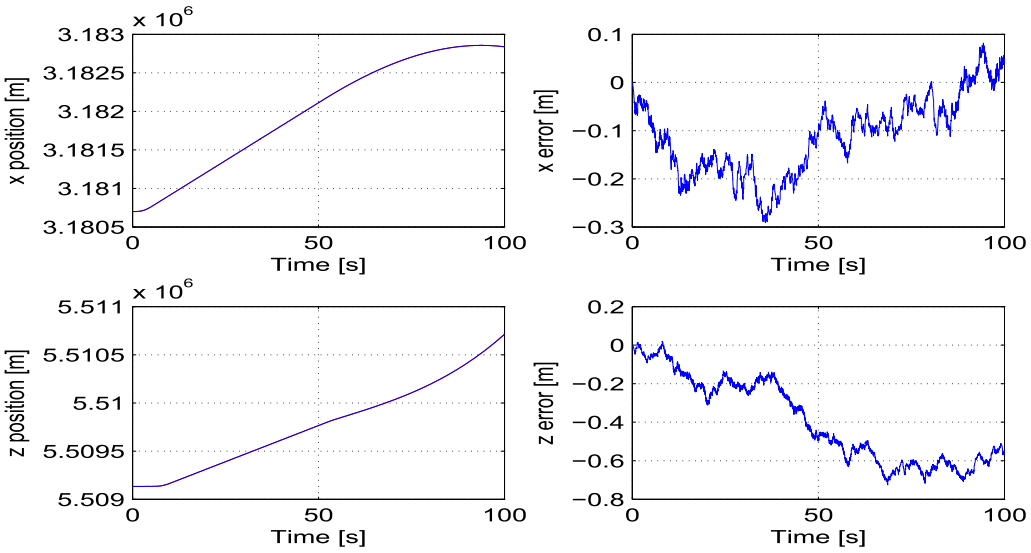

Figure 5.19. Position and related errors for the ultra-tight coupling.

mean of the last 100 measurements, so that the noise effect is mitigated in time. Moreover, the Kalman filter rate is 10 times the rate of the two formerly considered Kalman filters, so that the INS trajectory is corrected more often. A realistic implementation of the PLL and of the DLL would lead to different and worse results, comparable to the ones obtained

with the loose and tight integration schemes. The error related to the estimate of the velocity is shown in Figure 5.20, considering its components separately.

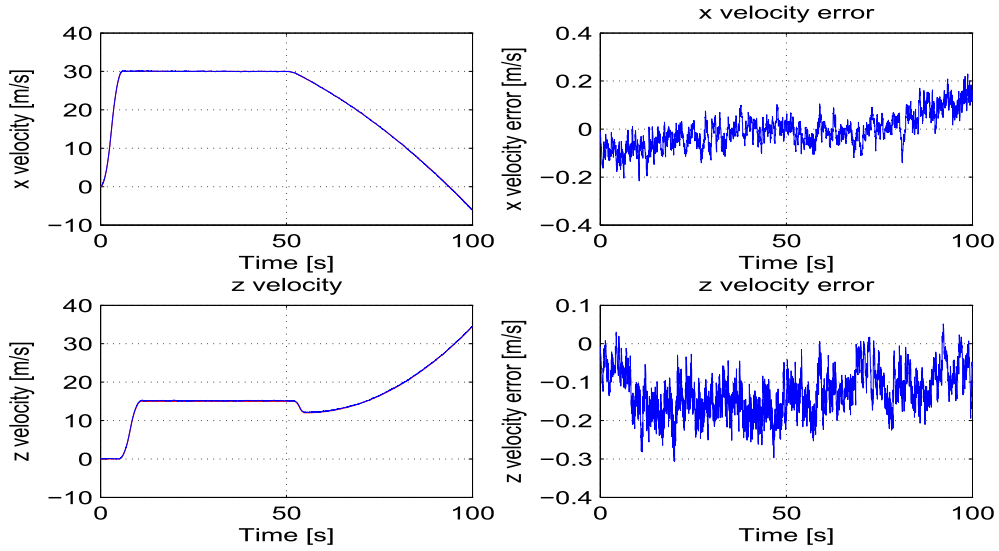

Figure 5.20. Velocity and related errors for the ultra-tight coupling.

The same considerations made for the position error are still valid and explain why the velocity estimations is reliable.

5.6.6 Real Data Loosely and Tightly Integrated Systems

A further trial that was performed was related to the integration of a 3DM-GX3 IMU with a GPS receiver. While the integration in the presence of the GPS signal was reliable and stable, allowing for a higher output rate, the results obtained in the absence of the GPS signal were less reliable. This is due to the low quality of the available IMU [46].

By the way, even a low quality IMU allows to estimate the position of the user when no GPS signal is present. In the following, we consider three different outages. They were simulated removing the pseudoranges from the data collected for a certain amount of time.

Roundabout

The first scenario considers data collected while driving a roundabout. The GPS outage begins a few meter before the turning and ends once the roundabout is driven and a short

straight road, after 30s. The main aim of this run is to test the performance of the system during a left-hand bend and its stability at the end of the bend.

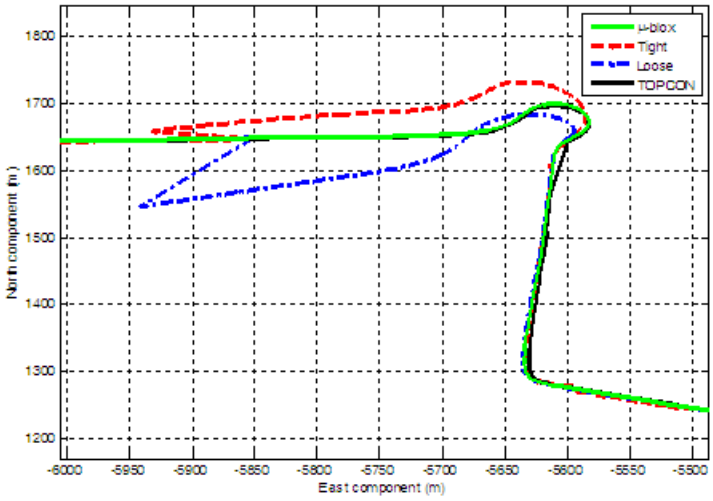

Figure 5.21. Trajectory for loose and tight integration - Case 1.

In Figure 5.21 it is possible to notice how the two integrated systems show similar trajectory errors with respect to the nominal path. In this case the GPS outage is simulated just before the beginning of the roundabout. In this case, both the loose and the tight integrations are sensitive to the trajectory, but in a slightly different way. In both the circumstances, the bend is clearly visible but the loose integration is globally more accurate to the turning variations than the tight strategy but it diverges more rapidly after the end of the roundabout. The differences can be ascribed to the two different receivers on which the integration schemes are based.

These results are confirmed by the errors reported in Figure 5.22, where the vertical error is the main contribution for the 3D error curve related to the tight integration: a substantial difference between the 3D and the 2D errors curve is clearly visible for the tight case rather than for the loose case. Notice that in this scenario no relevant variations in the height of the user are detected by the reference outputs. The main motivation of the better performance in the vertical direction experienced by the loose integration is due to the fact that it is based on a trajectory provided by a GPS receiver that compensates the error related to the vertical component in some way. The tight integration, instead, uses a stand-alone Kalman filter that does not take into account any compensation strategy.

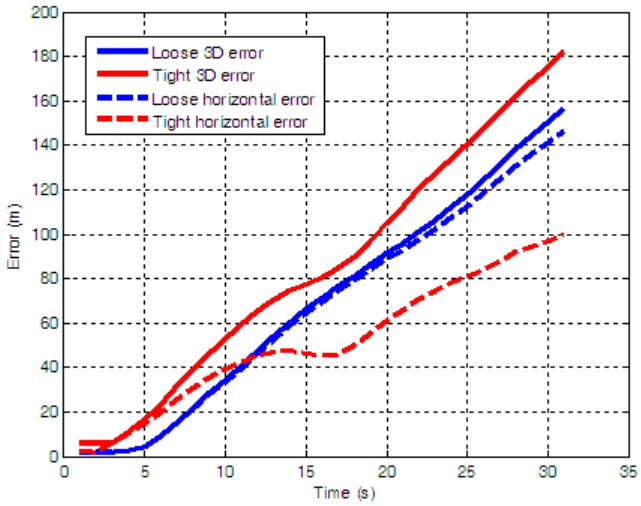

Figure 5.22. Error for loose and tight integration - Case 1.

Approaching a Flyover

The second scenario considers a shorter bend before entering a flyover and the outage lasts for 30s. The main aim of this run is to test the system stability while turning and increasing the height of the user.

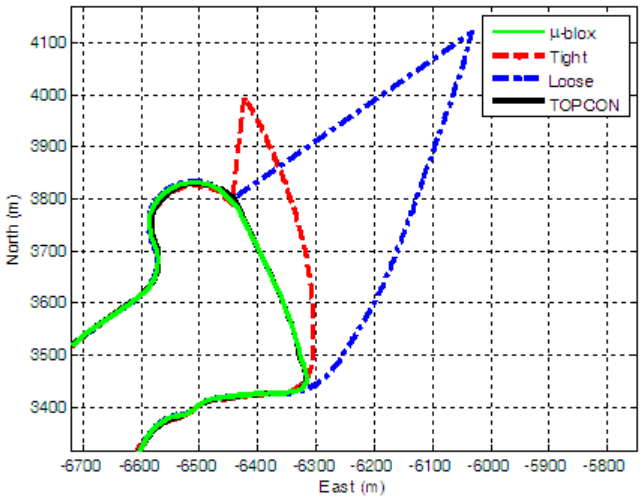

Figure 5.23. Trajectory for loose and tight integration - Case 2.

Figure 5.23 shows how the tight integration is in this case more accurate than the loosely coupled approach to detectthe variations related to the direction of the vehicle.

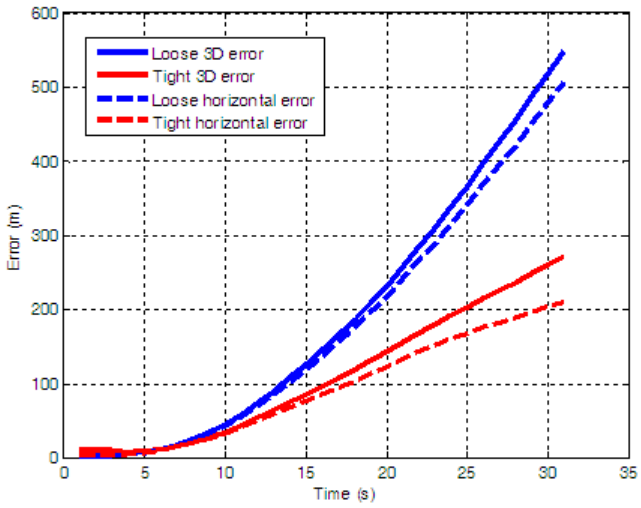

Figure 5.24. Error for loose and tight integration - Case 2.

Also in this case, as shown in Figure 5.24, the vertical component results worse in the case of the tightly integrated system. Nonetheless, both in the horizontal and in the 3D evaluation, the tightly coupled system outperforms the loosely coupled system.

Motorway

The third and final scenario is a 50s long GPS outage on a motorway. The main aim of the run is to test the system performance at high speed and with small changes in attitude.

In this case the two approaches results similar (almost symmetrical around the true positions) and relevant errors can be observed in Figure 5.25 for both tight and loose integrations.

Also in this case the errors observed in Figure 5.26 show that the tightly integrated system is more sensitive to vertical errors.

Conclusions on the Real Data Analysis

On the basis of the observed results, it is possible to understand that the two approaches considered in the former subsections are very similar and that the performance of an integrated system is dependent both on the quality of the GPS receiver and of the IMU.

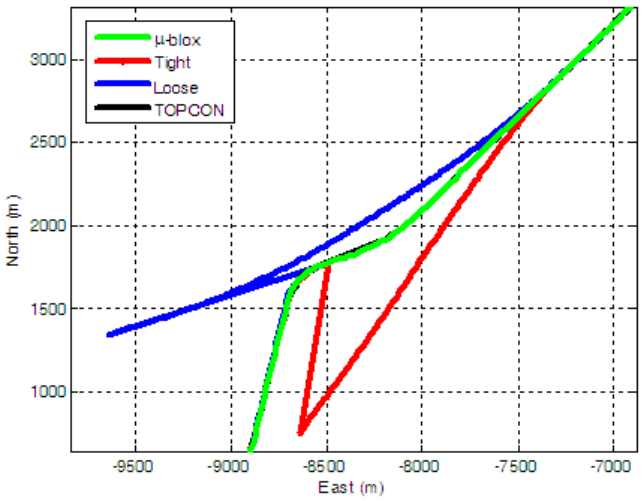

Figure 5.25. Trajectory for loose and tight integration - Case 3.

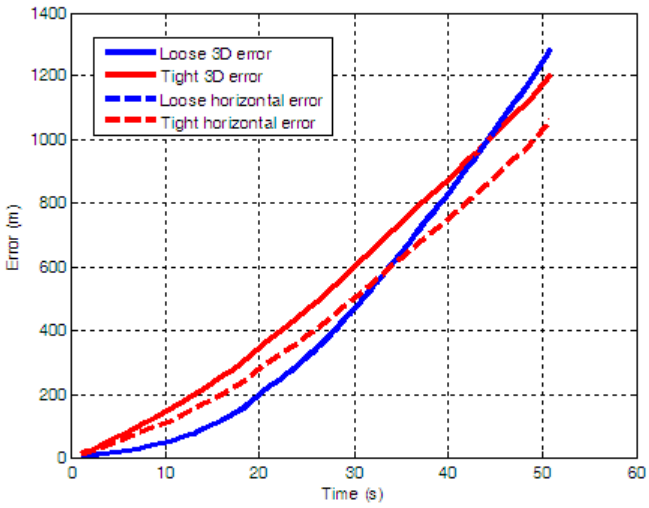

Figure 5.26. Error for loose and tight integration - Case 3.

Commercial GPS receivers may implement techniques for a better estimate of the satellite-based solution. In the considered scenarios this characteristic is evident when the vertical channel is considered and a difference between the commercial receiver and the MATLAB receiver is evident.

Moreover, commercial receivers may provide a solution even when less than four

satellites are in view, so that the main advantage of the tight integration upon the loose integration may be found in the fact that a tightly integrated system adopts a single KF.

Finally, to improve the performance of an integrated system and to slow down the drift of the INS-only solution, it is possible to design more complex systems, taking into account the constraints on the dynamic of the vehicle. This approach has already been considered in the literature [47].

Chapter 6

External Aiding in a Peer to Peer Network

The WAVE [48] [49] [50] is a wireless technology that is capturing the attention of the scientific community in the field of vehicular wireless communications. This is an environment where the properties of the physical layer are rapidly changing and where very short-duration communications are required [51].

This technology allows every vehicle to exchange data with the other users around by means of a wireless connection in a P2P fashion [16]. This opportunity allows the users in the network, namely the peers, to share every information in their possession and, potentially, to cooperate.

Exploiting the existence of direct communication links among nodes of a network, which are aware of their own position since they are equipped with a GNSS receiver, it is possible to implement a cooperative positioning service [15]. A GNSS provide a reliable position when at least four satellites are visible at the antenna of the receiver [8], which is generally guaranteed in open sky outdoor environments. Nonetheless, in peculiar conditions the satellite based localization may be inaccurate or fail [52].

A cooperative system gives the users the chance to exploit the information received from other peers to improve the performance of the stand alone receiver, both in open sky and in light indoor environments. In this chapter, the attention is focused on a possible solution for the cooperative positioning problem.

6.1 The Considered Scenario

Both the considered algorithms are applied to the same scenario, which is introduced in the following.

$N_{pr}+1$ peers populate the environment. One aided user, pointed out by the subscript "0" requests the aiding data to the N_{pr} aiding peers in its coverage area. The aiding peers are displaced in a limited and narrow range of altitudes. The aided user manages to get the aiding measurement from each peer in the coverage area using the WAVE based P2P network. Finally, there might be peers outside the coverage area of the aided user, but they cannot be reached. This scenario is schematically represented in Figure 6.1.

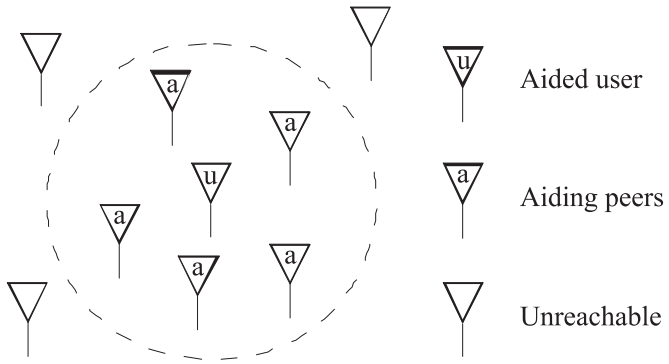

Figure 6.1. Coverage area associated to the aided peer.

Without any loss of generality, no more than one aided user will be considered in the following analysis, due to the fact that the performance of the designed system does not improve when two aided users are able to communicate. Each aided peer is then independent from the other aided peers.

6.2 Altitude Aiding Algorithms

Two algorithms based on the transmission of the altitude computed by the aiding peers are described in this Section.

6.2.1 Definition of the Aiding Data and Reliability Index

Considering the scenario described in Section 6.1, each aiding peer is provided with a GNSS receiver that is capable to provide an estimation $\hat{u}_m$ of its true altitude u_m. The

true and the estimated altitudes can be related as follows

$$\hat{u}_m = u_m + \xi_m \tag{6.1}$$

where ξ_m is a random variable that models the error that affects the estimate, imputed to various error sources that may affect the GNSS receiver measurement.

Here, the N_{pr} altitude values provided by the aiding peers represent the aiding measurements. These measurements can be processed in order to obtain only one aiding data $\overline{u}$ that is here obtained as follows

$$\overline{u} = \sum_{m=1}^{N_{pr}} \beta_m \hat{u}_m \tag{6.2}$$

where the coefficients β_m, used to weight the aiding data, are constrained to

$$\sum_{m=1}^{N_{pr}} \beta_m = 1 \tag{6.3}$$

Many strategies can be adopted to set the value of the coefficients β_m adequately. If the peers are considered equally reliable, the weights β_m will be the same independently from m

$$\beta_m = \frac{1}{N_{pr}}, \forall m \in \{1, \ldots, N_{pr}\} \tag{6.4}$$

The aiding data $\overline{u}$ can be similar to the true altitude of the aided user u_0. This similarity is strictly dependent on the disposition of the aiding peers around the aided user. In fact, if the weights are chosen according to (6.4), the mean in (6.2) mitigates the effect of the noise ξ_m that affects each one of the aiding measurements in (6.1). However this does not guarantee the required similarity. An example is when the peers altitude distribution $\{u_m\}$ is biased with respect to the altitude $\overline{u}$ of the aided user. A typical biased scenario is represented by the one of an aided peer on a flyover, with the aiding peers on the road underneath.

It is then necessary to evaluate the reliability of the aiding data $\overline{u}$ considering the distribution of the altitudes of the aiding peers and, possibly, comparing these altitudes with the one of the aided peer.

If only three satellites are in view, the computed aiding altitude $\overline{u}$ can be used to compute the PVT of the user and a reliability index of this parameter can be defined only on the basis of the estimated standard deviation of the received altitudes $\hat{u}_m$

$$\hat{\sigma}_{\hat{u}_m} = \sqrt{\sum_{m=1}^{N_{pr}} \beta_m^2 \left(\hat{u}_m - \overline{u}\right)^2} \tag{6.5}$$

However this index is not useful to identify a potential bias affecting the aiding measurements with respect to the true altitude of the user.

If four or more satellites are available, the user can compute the PVT solution. The aiding measurements can be then exploited to improve the precision of the GNSS solution. Moreover, the distribution of the aiding measurements $\{u_m\}$ can now be compared with an estimation $\hat{u}_0$ of the altitude of the user, in order to obtain the second order moment of the aiding measurements with respect to the GNSS-only computed altitude

$$\hat{\gamma}_{\hat{u}_m} = \sqrt{\sum_{m=1}^{N_{pr}} \beta_m^2 \left(\hat{u}_m - \hat{u}_0\right)^2} \tag{6.6}$$

With these definitions in mind, it is possible to define a parameter that states the reliability of the aiding measurement and that can be easily used in a Kalman filter to set the weight of the aiding measurement in the computation of the solution.

In the case that only three satellites are in view, the reliability index $r_{\overline{u}}$ is defined as

$$r_{\overline{u}} = \frac{1}{\hat{\sigma}_{\hat{u}_m}} \tag{6.7}$$

It can be perceived, and it will be shown in the simulations, that the reliability index in (6.7) performs well if a large set of unbiased aiding altitudes is available, but it is not capable to detect a biased set of aiding measurements.

If four or more satellites are in view, it is possible to exploit the second order moment defined in (6.6) to define a different reliability index

$$r_{\overline{u}} = \frac{1}{(N_{sat} - 2)\hat{\gamma}_{\hat{u}_m}} \tag{6.8}$$

In the reliability index in (6.8), the distribution of the received altitudes is computed with respect to the GNSS only computed altitude, so that a bias in the aiding may be detected. Moreover, since four or more satellites are available, the reliability index in (6.8) accounts for the number of satellites used in the computation of the solution thanks to the factor $N_{sat} - 2$, so that the reliability of the aiding $\overline{u}$ is conversely proportional to the number of satellites. The idea is to make the altitude aid increasingly irrelevant as the quality of the estimated position improves, thanks to the availability of an increasing number of visible satellites.

The value $N_{sat} - 2$ is chosen to make the reliability index in (6.8) compatible with the one in (6.7) (as it is evident substituting $N_{sat} = 3$) and the simulations allow to assert that it is a sound choice in terms of the performance of the system.

6.2.2 Aided Solution

A GNSS receiver can compute the position of the user starting from pseudorange measurements. The pseudorange ρ_i related to the i-th satellite is defined as follows

$$\rho_i = \sqrt{(x - x_{si})^2 + (y - y_{si})^2 + (z - z_{si})^2} - c\tau \tag{6.9}$$

where the triplet $[x,y,z]$ points out the considered position of the user, $[x_{si},y_{si},z_{si}]$ points out the i-th satellite position, c is the speed of light and τ is the receiver clock misalignment of the considered user.

If the pseudoranges related to at least four different satellites are available to the user, it is possible to solve a system of equations in the form of (6.9) to obtain the position of the user and the clock misalignment. By the way, if less than four satellites are in view, it is possible to replace the pseudorange measurements with other measurements. However, since the vector of the unknowns $\mathbf{p} = [x,y,z,\tau]$ includes τ, at least one pseudorange measurement is necessary. The other pseudoranges can be substituted with other independent observations.

We consider a single equation based on an estimate of the altitude of the user, so that at least three pseudoranges are necessary to perform the computation of the PVT of the user. The main purpose of the following discussion is to show the effectiveness of this aiding parameter, if the peers belonging to the P2P network are enabled to exchange it.

The idea is to exploit the altitude aiding data defined in (6.2) to determine a new equation to be considered together with (6.9). Therefore a first step is to relate the altitude measurement with the ECEF coordinates $[x,y,z]$ of the user, as described in [8] and in [53].

The system of equations that considers at least three pseudorange ρ_i and the altitude aiding $\overline{u}$ can be easily solved adopting an EKF [54]. Moreover, this method allows to exploit the parameter defined in (6.7) to balance the impact of the altitude aiding in the computation of the position of the user. This procedure is named Altitude Aided (AA) algorithm [53] and the main differences with respect to a traditional Kalman based PVT algorithm are briefly described in the following.

The model adopted in the EKF is a constant velocity one. Any details about the structure of the model and measurement matrices can be found in [6] or [53]. The adoption of the AA method only requires the insertion of a new line of the measurement matrix, that is the line of the altitude aiding data, and the corresponding entry in the measurement covariance matrix.

From the perturbation of the equations that relates the ECEF with the geodetic coordinates, it is possible to derive a matrix $\mathbf{G}$ that relates the errors in the two coordinate systems [47] of the form

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \underbrace{\begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & 0 & g_{33} \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} \Delta\phi \\ \Delta\lambda \\ \Delta h \end{bmatrix} \tag{6.10}$$

where Δ points out an error variable (state and measurement) used in the EKF approach, ϕ is the latitude of the user, λ is the longitude of the user, and h is the altitude of the user. The entries g_{ij} can be written in terms of ϕ, λ, h, and other parameters of the World Geodetic System 1984 (WGS84) [55] parameters, as it can be found in [53]

$$\begin{aligned}
g_{11} &= A\cos\phi\cos\lambda - (N+h)\sin\phi\cos\lambda && (6.11)\\
g_{12} &= -(N+h)\cos\phi\sin\lambda && (6.12)\\
g_{13} &= \cos\lambda\cos\phi && (6.13)\\
g_{21} &= A\cos\phi\sin\lambda - (N+h)\sin\phi\sin\lambda && (6.14)\\
g_{22} &= (N+h)\cos\phi\cos\lambda && (6.15)\\
g_{23} &= \sin\lambda\cos\phi && (6.16)\\
g_{31} &= A(1-e^2)\sin\phi + N(1-e^2)\cos\phi + h\cos\phi && (6.17)\\
g_{33} &= \sin\phi && (6.18)\\
A &= -ae^2\sin\phi\cos\phi/\sqrt{(1-e^2\sin^2\phi)^3} && (6.19)
\end{aligned}$$

where N is the Prime Vertical Radius, e is the eccentricity and a is the semi-major axis.

The only value of interest is Δh , so that the last row of the measurement matrix $\mathbf{H}$ is

$$\mathbf{H}_{N_{sat}+1,:} = \left[(\mathbf{G})^{-1}_{3,1},(\mathbf{G})^{-1}_{3,2},(\mathbf{G})^{-1}_{3,3},0\right] \tag{6.20}$$

where 0 is a null element required since the matrix $\mathbf{H}$ has four columns, but only the first three columns, related to position, are implied in the computation of Δh, while the fourth one is related to the receiver clock misalignment.

Finally, one of the requirements of a Kalman filter is a measure of the uncertainty related to the available measurement, stored in the measurement error covariance matrix

$\mathbf{R}[k]$, which is usually diagonal [6]

$$\mathbf{R}[k] = E\{\boldsymbol{\nu}[k]\boldsymbol{\nu}^T[k]\} \tag{6.21}$$

The last entry σ_u^2 in $\mathbf{R}[k]$ is the one related to the altitude aiding and in the AA algorithm it is set equal to the square of the inverse of the reliability index, defined in (6.7) and (6.8)

$$\sigma_u^2 = \left(\frac{1}{r_{\overline{u}}}\right)^2 \tag{6.22}$$

6.2.3 Iterative Aiding Algorithm

In Section 6.2.1 it is pointed out that the reliability index defined in (6.8) is capable to detect a bias in the received measurements. This capability is useful since there is the chance that the aided user is displaced at a different altitude with respect to the aiding peers. In this case, the user is provided with measurements that differ from the actual altitude even by some meters.

When only three satellites are in view, only the reliability index defined in (6.7) is available, which is not capable to detect any bias that affects the aiding measurements. In this case, the solution provided by the AA algorithm might be wrong. To overcome this issue, an iterative technique, named Iterative Altitude Aided (IAA)-PVT, capable to deal with biased aiding measurements, has been successfully adopted [56,57], as shown in the following.

Not only the IAA algorithm is prone to the detection of a bias that affects the received aiding measurements, but it is capable to compute the PVT solution even if only three satellites are available. In this condition a traditional receiver can compute its position only if it is already synchronized with the clock of the satellite, but this situation is not generally realistic, especially at the first fix. Therefore in these cases the IAA method represents an improvement with respect to a receiver that exploits only pseudoranges.

The algorithm can be summarized in the following steps:

1. Initialization:

   (a) Set the initial position of the aided peer $\mathbf{p}_0[0] = [x_0,y_0,z_0]^T$ as a mean of the positions $\mathbf{p}_m[0] = [x_m,y_m,z_m]^T$ received from the aiding peers;

   (b) Estimate $\tau[0]$ using the initial position $\mathbf{p}_0[0]$ and the available pseudorange measurements;

(c) Set $\overline{u}[0]$ as the mean of the received altitude aidings;

(d) Set an altitude measurement error variance σ_u^2, which is the last element of the matrix $\mathbf{R}$ in (6.21), greater than the pseudorange measurement error variances;

(e) Initialize the recursion by setting $\overline{u}[1] = \overline{u}[0]$;

2. Recursion:

(a) Compute the PVT $[\mathbf{p}_0^T[k],\tau_0[k]]^T$ using three satellites and the aiding altitude measurement $\overline{u}[k]$;

(b) Use $\mathbf{p}_0[k]$ to estimate the altitude of the aided peer $\hat{u}^+[k]$;

(c) Compute the second order moment $\hat{\gamma}_{\hat{u}_m}[k+1]$ of the aiding measurements $\hat{u}_m[1]$ with respect to the estimated altitude $\hat{u}^+[k]$, by setting $\hat{u}_0 = \hat{u}^+[k]$ in (6.6);

(d) Compute the new aiding data $\overline{u}[k+1]$ as a weighted mean of the computed GNSS $\hat{u}^+[k]$ altitude and of the present aiding altitude $\overline{u}[k]$ using a weight w;

(e) Update the measurement error covariance matrix $\mathbf{R}[k+1]$ related to the altitude measurement, using the previously computed value $\hat{\gamma}_{\hat{u}_m}[k+1]$;

(f) Iterate the recursion in time, as long as data is available and as long as the algorithm must be run.

Given this brief summary, the rationale behind this recursive scheme is presented hereafter, analyzing the algorithm step by step.

In the initialization phase of the IAA algorithm, the initial position $\mathbf{p}_0[0]$ of the Kalman filter is set considering an arithmetic mean of the coordinates $\mathbf{p}_m[0] = [x_m[0],y_m[0],z_m[0]]^T$ of the peers in the coverage area of the aided user

$$[x_0[0],y_0[0],z_0[0]]^T = \frac{1}{N_{pr}} \sum_{m=1}^{N_{pr}} \mathbf{p}_m[0] \tag{6.23}$$

Once the position of the user is initialized, it is possible to estimate an initial value of the clock misalignment $\tau_0[0]$ exploiting (6.9), the initial position estimate and the available pseudorange measurements.

In the case of a biased disposition of the aiding peers with respect to the aided user, the aided user will be initially placed in a slightly wrong position and, in particular, its

initial altitude will be affected by a bias. It is possible to define the biased aiding data as follows

$$\overline{u}[0] = u_0[0] + b[0] \tag{6.24}$$

where $b[0]$ is the altitude bias that affects the aiding data.

Once the initialization procedure is completed, the Kalman filter computes the PVT solution solving the system of equations made up by the three equations related to the available pseudorange measurements $\rho_1[1]$, $\rho_2[1]$, $\rho_3[1]$ and by the equation related to the received altitude aiding $\overline{u}[1] = \overline{u}[0]$.

Only at the first step, in order to rely on the pseudorange measurements more than on the altitude aiding measurement, the altitude measurement error variance in the $\mathbf{R}[1]$ matrix $\hat{\sigma}^2_{\hat{u}_m}[1]$ should be high.

At the first step of the recursion loop, a new PVT solution is available

$$\mathbf{p}_0[1] = [x_0[1], y_0[1], z_0[1], \tau_0[1]]^T \tag{6.25}$$

The solution computed in (6.25) can be used to estimate a new altitude value $\hat{u}^+[1]$, which is different from the altitude aiding $\overline{u}[0]$

$$\hat{u}^+[1] = f(x_0[1], y_0[1], z_0[1]) \tag{6.26}$$

where $f(\cdot)$ is the function used to evaluate the altitude associated to ECEF triplet given as input [53]. It must be noticed that $\mathbf{p}_0[1]$ is an estimate and it can be affected by a relevant bias.

Once the GNSS estimated altitude defined in (6.26) is available, it is possible to update the variance σ^2_u of the altitude measurement error to be set in $\mathbf{R}[2]$. Despite only three satellites are in view, the availability of an estimate of the altitude of the user allows to set σ^2_u equal to the second order moment of the aiding measurements $\{\hat{u}_m\}$ with respect to $\hat{u}^+[1]$, similarly to (6.6)

$$\hat{\gamma}_{\hat{u}_m}[2] = \sqrt{\frac{1}{N^2_{pr}} \sum_{m=1}^{N_{pr}} (\hat{u}_m[1] - \hat{u}^+[1])^2} \tag{6.27}$$

A new aiding altitude $\overline{u}[2]$ is then computed as a weighted mean of $\hat{u}^+[1]$ and $\overline{u}[1]$

$$\overline{u}[2] = (1 - w)\hat{u}^+[1] + w\overline{u}[1] \tag{6.28}$$

where w is the weight used to determine the contribution of the GNSS computed altitude and of the former aiding altitude.

Once the first iteration is completed, the new aiding data, computed in (6.28), is used together with the second order moment of the altitude, determined in (6.27), in the next iteration. It is then possible to describe a generic iteration by means of the following equations

$$\hat{\gamma}_{\hat{u}_m}[k] = \sqrt{\frac{1}{N_{pr}^2}\sum_{m=1}^{N_{pr}}\left(\hat{u}_m[1]-\hat{u}^+[k-1]\right)^2} \tag{6.29}$$

$$\overline{u}[k] = (1-w)\cdot\hat{u}^+[k-1]+w\cdot\overline{u}[k-1] \tag{6.30}$$

$$\mathbf{p}_0[k] = [x_0[k],y_0[k],z_0[k]]^T \tag{6.31}$$

$$\hat{u}^+[k] = f(x_0[k],y_0[k],z_0[k]) \tag{6.32}$$

The aiding data $\overline{u}[1]$ is computed using the aiding measurements received at the first step $\{\hat{u}_m[1]\}$. The IAA algorithm iteratively refines it only on the basis of the aiding data available at the previous step and of the GNSS altitude computed in the current step. Moreover, the entry of the covariance matrix $\mathbf{R}[k]$ related to the aid $\overline{u}[k]$ is set on the basis of the altitude values received from the aiding peers $\{\hat{u}_m[1]\}$.

This is a conservative choice that allows the algorithm to converge quickly in the case of a reliable aiding and to avoid that an initial biased aiding has a remarkable impact on the computed altitude.

Convergence of the IAA algorithm, a simulative approach

The IAA algorithm has been developed in response to a bigger project sponsored by the ESA. During this project a simulator of a network of peers equipped with GNSS receivers has been developed, to test the performance of the developed algorithms. The measurements are generated in accordance with [58] and the simulator considers asynchronous transmission.

In order to test if the IAA algorithm is convergent or not, a convergence curve was obtained by means of simulation. 50s of simulated data were considered for 100 different runs. In Figure 6.2 the result of this study is shown.

In this simulation, four peers are considered in open sky, while the aided peer is in light indoor, it has 3 satellites in view and it is placed 5m above the other peers. The initial position of the aided peer is obtained as a mean of the already received aiding peers positions. Since the algorithm has been developed for the static case, the peers are considered static.

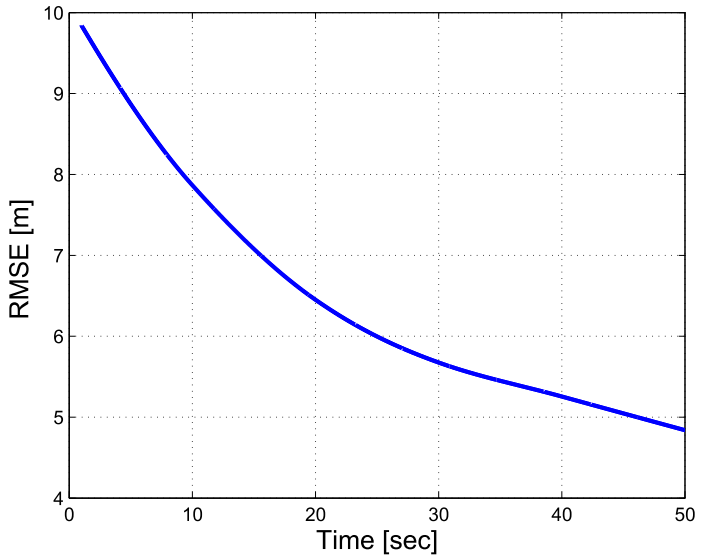

Figure 6.2. IAA algorithm convergence, 50 seconds, 100 repetitions, 3 satellites, 4 peers $w = 0.85$.

It can be seen that despite the initial error is significant, the IAA algorithm is capable to reduce it, bringing it below the initial value of the bias. Moreover, the error is constantly decreasing, showing that the algorithm is stable and improves the estimate in time.

6.3 Pseudorange Aiding Algorithm

In this Section we describe a different approach to the cooperative positioning problem. This time, an algorithm based on the transmission of the ranges computed by the aiding peers will be described.

6.3.1 Definition of the Aiding Data

Considering the scenario described in Section 6.1, the generic m-th aiding peer has at least four satellites in view and it is characterized by a true position $\mathbf{p}_m = [x_m, y_m, z_m]^T$, that is estimated as $\hat{\mathbf{p}}_m = [\hat{x}_m, \hat{y}_m, \hat{z}_m]^T$ by the GNSS receiver installed on the vehicle. The true and the estimated positions are related as follows

$$\hat{\mathbf{p}}_m = \mathbf{p}_m + \boldsymbol{\xi}_m \tag{6.33}$$

where $\boldsymbol{\xi}_m$ is a three component random vector that models the error that affects the estimate.

Each one of the aiding peers estimates its own position and knows the position of the satellites used in the estimation. Upon the request of the aided user, an aiding peer m can send the list of the distances between its estimated position and a satellite j that is not in the LOS of the aided user, namely the estimated range

$$\hat{r}_{m,j} = \sqrt{(x_j - \hat{x}_m)^2 + (y_j - \hat{y}_m)^2 + (z_j - \hat{z}_m)^2} \tag{6.34}$$

It must be noticed that the choice to transmit the range measurement instead of the pseudorange measurement is explained considering that a pseudorange measurement is affected by a clock bias which is not the same among the receivers in the network. Moreover, it can be observed that the estimation of the $\hat{r}_{m,j}$ is characterized by a low computational effort for the aiding peer, as it can be easily obtained from the computed PVT solution.

The *j*-th satellite is not in the line of sight of the aided user, so that the following range is not available

$$r_{0,j} = \sqrt{(x_j - x_0)^2 + (y_j - y_0)^2 + (z_j - z_0)^2} \tag{6.35}$$

If only three pseudorange measurements are available, the range in (6.35) would enable the aided user to compute its own position. Since $r_{0,j}$ cannot be measured directly by the user, a possible solution is to compute an approximation $\hat{r}_{0,j}$ of $r_{0,j}$, relying upon the data provided by the aided peers. In fact, the ranges provided by the aiding peers are available to the user requesting the aid. For the considered *j*-th satellite, the aided peer can compute an approximation of the aiding range as a weighted mean of the received ranges

$$\hat{r}_{0,j} \cong \sum_{m=1}^{N_{pr}} \beta_m \hat{r}_{m,j} \tag{6.36}$$

where β_m are the coefficients used in the computation of the mean, so that

$$\sum_{m=1}^{N_{pr}} \beta_m = 1 \tag{6.37}$$

The choice to compute a mean of the received ranges can be understood considering a distribution of peers in the coverage area of the aided user: none of these ranges

can exactly replace the missing range (except some particular cases, shown in the Section devoted to the simulations), but a mean of a large enough number of ranges is an approximation of the range that the aided user is missing.

By the way, due to the geometrical nature of the range measurement, it might be a good choice to weight the aiding provided by a close peer more than the one of a far peer. If the peers in the network are capable of ranging measurement between each other, it is possible to determine the weights β_m as a function of the distance d_m between the user and the m-th peer. If no ranging capability is available, the ranging can be done a posteriori using the computed solution, in an iterative fashion.

It is evident that the range in (6.36) is not equal to the true range in (6.35), but it represents an approximation of the true range and enables the determination of the position of the aided peer when less than four satellites are in view.

6.3.2 Augmented Navigation Equations

The aiding range computed previously must be used together with the usual navigation equations. In this section, the system of navigation equations, augmented with the range equations, is described.

A GNSS receiver manages to compute the position of the user starting from the measured pseudorange or carrier phase. In the following only the pseudorange measurements related to the aided user will be considered.

The pseudorange $\rho_{0,j}$ is related to the geometrical distance between the aided user and the j-th satellite and to the receiver clock misalignment with respect to the GNSS time.

A set of pseudoranges can be used to define a system of equations to obtain the position of the user and the clock misalignment. Despite at least four satellites are required to be in view in order to have enough pseudorange measurements to solve the system, when range aidings are available, the receiver can compute the position of the user even if less than four satellites are in view.

The range in (6.34) stores all the information contained in the pseudorange measurement available to the m-th aiding peer, except his own clock bias τ_m. As a consequence, the aiding measurement in (6.36) is a function of the mean of positions of the peers in the coverage area of the aided user, and at least one pseudorange measurement is needed to the aided user to compute his own τ_0, which is the only clock misalignment involved in

the computation of the position of the aided user.

It must be noticed that the positions implied in the computation of (6.34) are related to the aiding users and not to the aided user, so that the position estimated by the aided user exploiting the aiding range in (6.36) will be affected by a bias, since a difference exists between the range in (6.36) and the range in (6.35), being the latter related to the distance between the aided user and the considered satellite. By the way, when less than four satellites are available, such a bias might be accepted as a tradeoff to be paid to obtain a solution, which would be unavailable whereas a standard GNSS receiver would not be capable to compute any position at all.

Pseudorange measurements and range measurements can be used together to compute the position of the user when three or less pseudorange measurements are available. These equations can be easily linearized by means of a Taylor first order approximation around a given linearization point, in order to set up a linear system of equations, as shown in [8], to be iteratively solved. The linearization point is in fact updated iteratively, exploiting the errors related to the prediction of the measurement to evaluate the errors related to the PVT of the user. Considering that a range measurement is not dependent upon the receiver clock bias, it is possible to obtain the following linear system

$$\begin{bmatrix} \boldsymbol{\Delta\rho} \\ \boldsymbol{\Delta r} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{a}_x & \mathbf{a}_y & \mathbf{a}_z & \mathbf{g} \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} \Delta x_0 \\ \Delta y_0 \\ \Delta z_0 \\ -c\Delta\tau_0 \end{bmatrix}}_{\boldsymbol{\Delta}} \tag{6.38}$$

where $\boldsymbol{\Delta\rho}$ is the error related to the prediction of the pseudorange, $\boldsymbol{\Delta r}$ is the error related to the prediction of the range, $\mathbf{H}$ is the matrix that stores the components $\mathbf{a}_i$ of the unitary vectors that point from the current estimated position of the user to the satellites, $\mathbf{g}$ is a vector whose components are equal to one if related to a pseudorange measurement and equal to zero if related to a range measurement, reflecting the definitions of the pseudorange and of the range, and $\boldsymbol{\Delta}$ is the vector of corrections to be applied to the current estimate of the position of the user.

A least squares filter can be easily implemented in a GNSS receiver to compute the position of the user. The theory related to this topic is widely known in literature [8] and the range aiding algorithm only requires the small modifications that haveru been described. It must be noticed that when only four pseudorange or range measurements

are available, the least squares method is equivalent to the solution of the linear system of equations.

The proposed algorithm represents a small modification to the usual PVT algorithm and it would be easy in a receiver to switch from a range aided estimation to a pseudorange-only solution.

6.4 Results: Altitude Aiding

In the following we will analyze the results related to the discussed altitude aiding algorithms. Both real data and simulated data will be considered. According to the actual development of the project, we analyze at first the results obtained processing real data.

6.4.1 Real Data Analysis: Setup

The hardware involved in the data collection is made up by a uBlox5 receiver connected to an antenna, placed in a georeferenced spot on the roof of our laboratory.

A data grabber software was then used to collect GPS data at a rate equal to 1Hz. Finally, a pseudorange-level software receiver was developed, implementing a Kalman filter to test the AA and IAA algorithms.

Concerning this data collection, the antenna is still during the whole simulation, so that the true altitude of the user u_0 is constant

$$u_0[k] = u_0, \forall k. \tag{6.39}$$

This choice in useful to assess the performance of the system, since the estimated altitude can be compared to the georeferenced altitude.

The georeferenced altitude of the antenna u_0 is also exploited to simulate the altitude transmitted by the m-th aiding peer $\hat{u}_m$, as follows

$$\hat{u}_m[k] = u_0 + \Delta\hat{u}_m[k] + \xi_m[k], \tag{6.40}$$

where $\Delta\hat{u}_m[k]$ is the difference in altitude between the m-th peer and the aided user and $\xi_m[k]$ is the noise that affects the m-th peer altitude estimate.

The aiding measurements and the noise that affects the aiding measurements are computed only at the initial step and they are never updated, both in the case of the AA and of the IAA algorithms, because the antenna is still. By the way, the realizations of the noise

$\{\xi_m[1]\}$ are generated each time a new simulation is started. The aiding measurements are then generated only at time $k = 1$ and they are given by

$$\hat{u}_m[1] = u_0 + \Delta\hat{u}_m[1] + \xi_m[1]. \tag{6.41}$$

The altitudes defined in (6.41) are received by the aided peer, that evaluates the aiding data as a weighted mean of the received altitudes, as pointed out in (6.2), using the weights given in (6.4).

Each run exploits the altitude aiding and a constant number of satellites, that are chosen randomly once per second, in order to make the results independent from the satellites used in the computation of the PVT.

The AA algorithm processes a 100 seconds long span, so that the satellites are randomized 100 times at each run. Moreover, the receiver processes the span 1500 times, applying a different realization of the noise $\xi_m[1]$ to the same span each time.

The setup related to the IAA-PVT algorithm is similar to the one formerly described, but a longer span of data is analyzed. The performance of this algorithm is evaluated for different values of the bias $\Delta\hat{u}_m$ applied to the aiding data.

The simulations are intended to evaluate the effectiveness of the computed reliability index in the weighting of the aiding data and to explain the need for the IAA algorithm. Both the case of an unbiased and of a biased aiding will be considered, in order to provide a general view of the performance of the algorithms.

6.4.2 Preliminary Tests for the Reliability Index

A preliminary test with real data was held at first to analyze the behavior of the algorithm. Considering the same data collections, different scenarios were simulated, changing the quality of the aiding and the degree of trust in the Kalman filter.

The results of the first simulation are shown in Figure 6.3. The plot depicts the performance of the aided receiver in terms of the standard deviation of the error related to the altitude, when a reliable aid is received and considering different values of the standard deviation σ_k of the measurement noise related to the measurement of the altitude.

It can be stated that when an accurate aid is received, it is better for the receiver to consider this datum. The performance of a receiver that is trusting the aiding datum is almost independent from the number of satellites and it is at its best even when only three satellites are in view. On the other side, a receiver that does not exploit a good aiding datum improves its performance when the number of satellites in view grows.

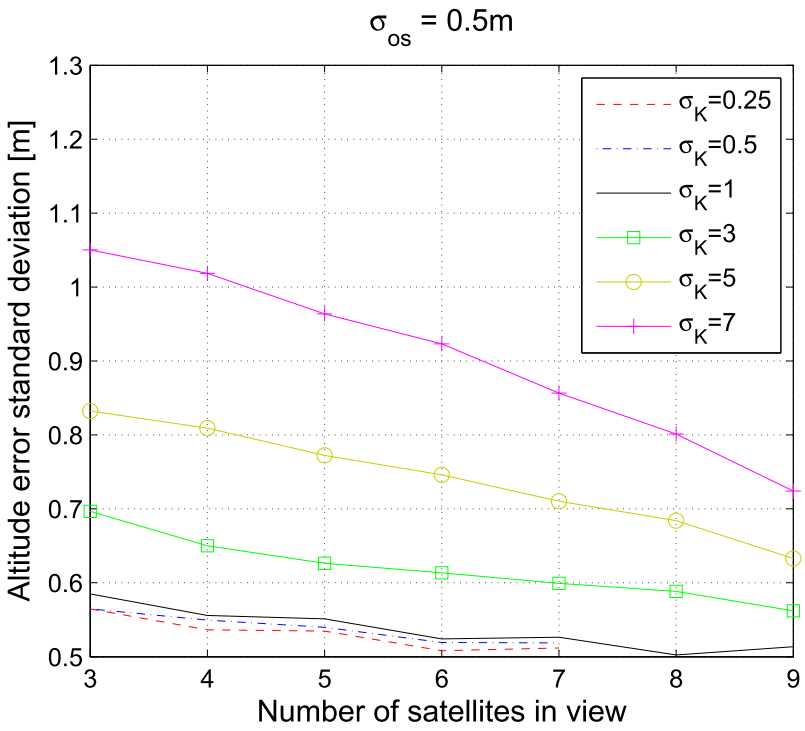

Figure 6.3. Standard deviation of the user computed altitude error in the presence of a good aid and varying the estimated degree of reliability.

The results of the second simulation are shown in Figure 6.4. The plot depicts the performance of the aided receiver in terms of the standard deviation of the error related to the altitude, when the receiver is characterized by a high degree of trust upon the altitude aiding and for different values of the variance of the noise that affects the altitude measurement.

It can be stated that a high degree of trust results in a good choice when the aiding is actually reliable. In fact, such a high degree of trust implies that the receiver considers the altitude aiding highly reliable and it leans toward a solution that is based on the value of the aiding.

The results of the third simulation are shown in Figure 6.5. The plot depicts the performance of the aided receiver in terms of the standard deviation of the error on the altitude, when an unreliable aid is received and for different values of the trust of the Kalman filter upon the received measurements of altitude.

It can be stated that when an unreliable aid is received, it is better for the receiver to avoid to compute a solution that is based on this value. The performance of a receiver

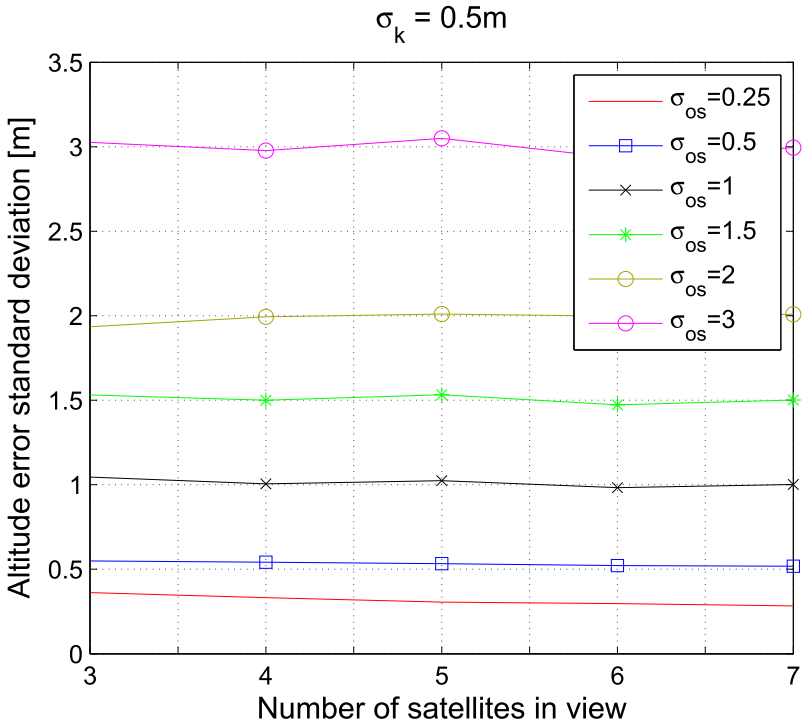

Figure 6.4. Standard deviation of the user computed altitude error keeping a high estimated degree of reliability and varying the noise variance that affects the aid.

that trusts the aiding is almost independent from the number of satellites, while a receiver that considers the datum as unreliable manages to perform better and to improve its performance when the number of satellites in view grows. It must be noticed that a Kalman filter that overestimates the noisiness of the aiding outperforms the one that estimate the actual noise variance. It is then not necessary that the reliability index is a correct estimate of the aiding error variance, but it must be an index that properly tunes the variance delated to the measurement in the Kalman filter.

The results of the fourth simulation are shown in Figure 6.6. The plot depicts the performance of the aided receiver in terms of the standard deviation of the error on the altitude, when the receiver has a low degree of trust upon the altitude aiding and for different values of the variance of the noise that affects the altitude measurement.

The simulation considers four different cases, spanning from very reliable measurements up to unreliable measurement, when the Kalman filter is not considering the aiding as a reliable one, so that the solution highly relies on the pseudorange measurement.

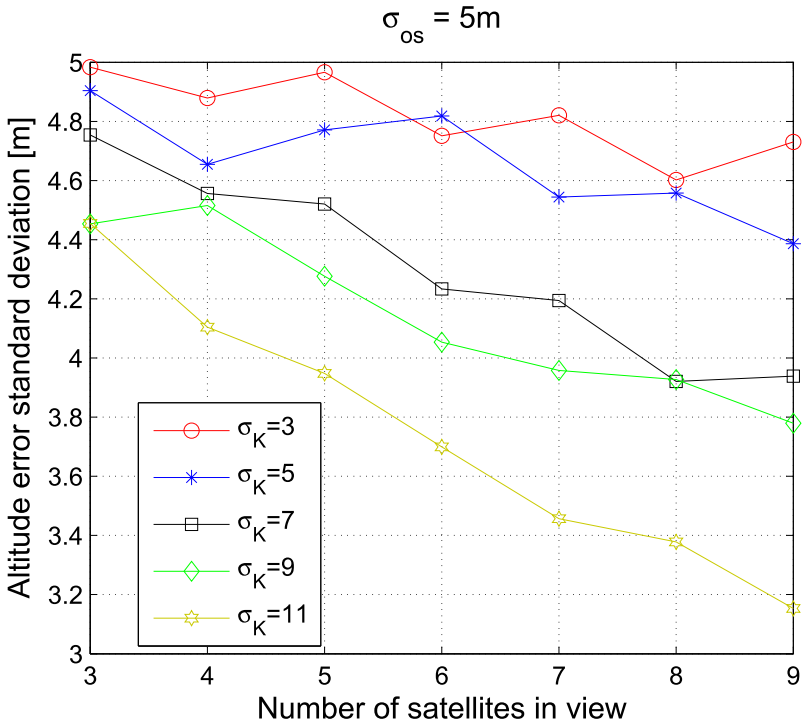

Figure 6.5. Standard deviation of the user computed altitude error in the presence of a bad aid and varying the estimated degree of reliability.

Even if the Kalman filter does not rely on the external aiding, the solution is still influenced by the aid. Higher degrees of trust would bring to errors that are closer to the actual variance of the noise. The main result of this plot, in opposition to Figure 6.4, is that a low degree of trust allows the Kalman filter to improve the accuracy of the solution when the number of satellites in view increases.

6.4.3 AA Algorithm with Unbiased Aiding

Once the first tests were finished, new experiments were held to test the reliability index with the AA algorithm. In the following, we present the three different scenarios and the results of the simulations.

Four Aiding Peers

The peers that populate the scenario characterized by an unbiased aiding are displaced at $+10$m, $+5$m, -5m and -10m with respect to the altitude of the user, so that their mean

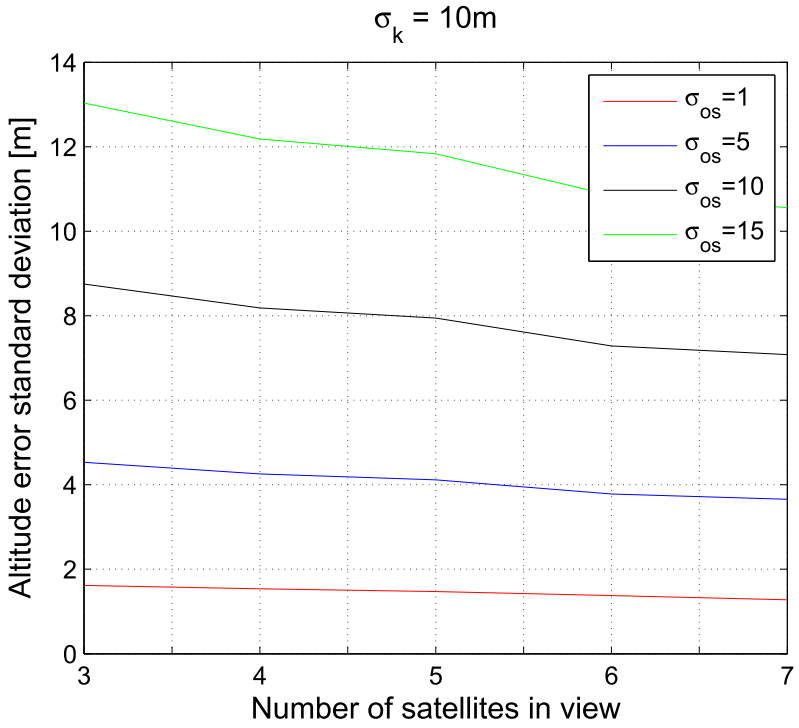

Figure 6.6. Standard deviation of the user computed altitude error keeping a low estimated degree of reliability and varying the noise variance that affects the aid.

altitude is equal to the altitude of the user. To simulate the aiding measurements, the true altitude of the aiding peers is affected with a zero-mean Gaussian noise ξ_m with standard deviation $\sigma_{OS,peer} = 3$m. Despite different values of $\sigma_{OS,peer}$ have been tested, the following results are given for the value $\sigma_{OS,peer} = 3$m, since the main purpose of this experiment is to show the effect of the bias, rather than the one of the variance of the noise. Moreover, $\sigma_{OS,peer} = 3$m is a realistic value for open sky aiding peers. During the whole simulation, neither the aided user nor the peers are moving. At each second, the aided peer computes its position and derives its altitude $h_0[k]$. Many experiments are performed and the standard deviation σ_0 of the error $h_0[k] - u_0$ is estimated, considering the whole simulation interval and all the different simulations. The results of this simulation are shown in Figure 6.7.

The AA-PVT algorithm is run testing different values of the parameter σ_u, considered in (6.22), to verify the effectiveness of the reliability index introduced in Section 6.2.1. To analyze the results of this simulation, this value is compared with the theoretical standard deviation, computed from the parameters of the simulation as $\sigma_{aid} = \sigma_{OS,peer}/\sqrt{N_{pr}} =$

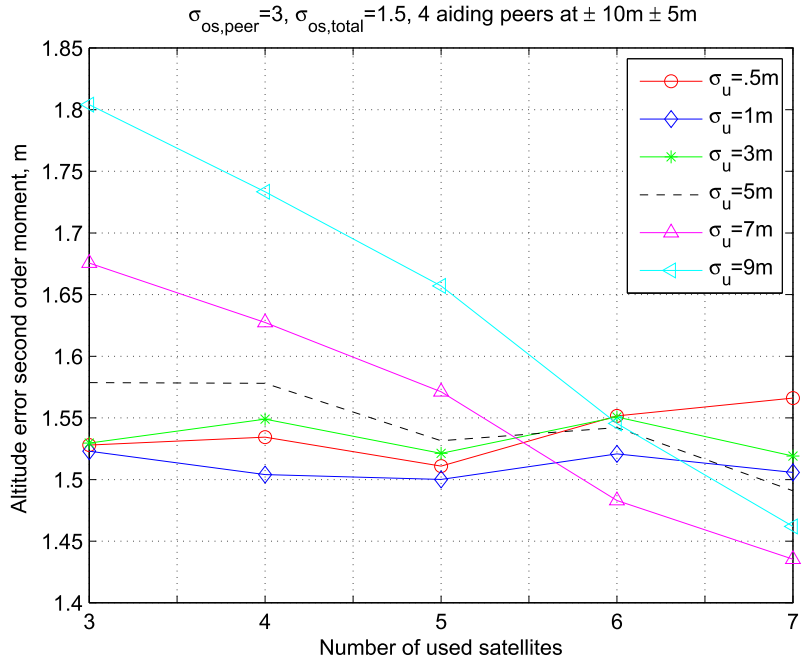

Figure 6.7. Second order moment of the user computed altitude error in the presence of four aiding peers and varying the estimated degree of reliability.

1.5m.

In these experiments, the aided user is considered able to estimate the theoretical standard deviation σ_{aid}, so that equations (6.5) and (6.6), necessary for a practical implementation, are not used and the estimated $\hat{\sigma}_{\hat{u}_m}$ and $\hat{\gamma}_{\hat{u}_m}$ are directly related to the chosen value of σ_{aid}. The values of $r_{\bar{u}}$ and σ_u, necessary to initialize the Kalman algorithm, are obtained from (6.7) and (6.8).

The results of the simulations are summarized in Table 6.1.

# satellites	$r_{\bar{u}}$[m^{-1}]	σ_u[m]	RMSE[m]
3	0.67	1.5	1.52
4	0.33	3.0	1.55
5	0.22	4.5	1.54
6	0.17	6.0	1.52
7	0.13	7.5	1.44

Table 6.1. Reliability index, 4 balanced peers, AA algorithm.

The simulations highlighted that when the number of available satellites is small, the receiver performs better when σ_u is small, because the precision of the GNSS solution is

not high and can be improved by the altitude aiding. On the contrary, when the number of available satellites grows, the precision of the GNSS solution is higher and it is not improved if the aiding altitude is exploited, so that the receiver performs better for greater values of σ_u.

In Table 6.1, the values of $r_{\bar{u}}$ and σ_u are obtained for different values of N_{sat}, and by neglecting the estimation error, that is starting from σ_{aid}. Moreover, the estimated root mean squared error related to altitude, obtained using the reported value of the reliability index, is shown in the fourth column and it is easy to notice that it lies below the value 1.55m for all the considered values of N_{sat}.

Considering Table 6.1, it is evident that in this scenario the definitions in (6.7) and (6.8) allows the AA algorithm to balance the contribution of the aiding and of the GNSS measurements.

Twelve Aiding Peers

The results of the second simulation are shown in Figure 6.8. The plot depicts the performance of the aided receiver in terms of the standard deviation of the error on the computed altitude, when twelve aiding peers are in the same cluster of the user. The twelve aiding peers are displaced at $+10$m (two peers), $+7$m, $+5$m, $+3$m, $+1$m, -1m -3m -5m -75m and -10m (two peers) with respect to the altitude of the user. Their computed altitude is given by their true altitude affected by a zero-mean, three meters standard deviation $\sigma_{os,peer}$ Gaussian noise ξ_m. Finally, the performance of the receiver is tested for different values of the noise standard deviation σ_k characterizing the altitude measurement available to the Kalman filter.

Again, the computed aid is almost unbiased. A little bias can be present due to asymmetry in the different noise realizations that affects the peer computed altitudes, but it is negligible and its impact is even less relevant than the former case, thanks to the greater number of aiding peers. The theoretical standard deviation of the aid can be computed and it is equal to $\sigma_{aid} = \sigma_{os,peer}/\sqrt{N_{pr}} \approx 0.866$m.

Starting from the left part of the plot, it can be stated that when a little number of satellites is used to compute the position of the user and a good aiding datum is available, the receiver performs better if the aiding datum is heavily considered in the computation of the solution.

In the case of three satellites, the reliability of the aiding datum can be evaluated

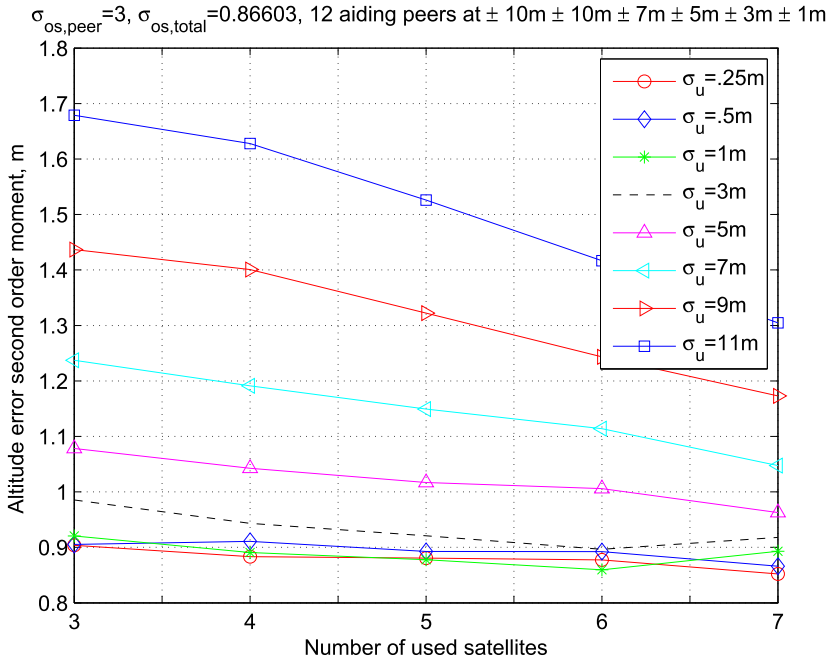

Figure 6.8. Second order moment of the user computed altitude error in the presence of twelve aiding peers and varying the estimated degree of reliability.

relying only upon the variance of the received altitudes and no information regarding any bias that may affect it is available. The same considerations given for the first simulation holds true in this case.

Also this time, when the number of available satellites grows, the effect due to the pseudorange noise is mitigated, but the GNSS-only computed altitude is affected by a noise whose standard deviation is higher than the one that affects the altitude aiding. In this case, to improve the performance of the system, a greater number of satellites is required.

Considering Equations (6.7), (6.8) and (6.22), it is possible to obtain the values of σ_k shown in Table 6.2.

In this case, setting σ_k on the basis of the computed reliability index allows the aided peer to decrease the impact of the aiding datum on the computed solution when the number of available satellites grows. By the way, this time the reliability coefficient decreases more slowly with respect to the former case, thanks to the larger number of aiding peers and to the smaller variance of the aiding data. Comparing these values with the ones shown in Figure 6.8, it is easy to notice that the plot of the second order moment of the

# satellites	$r_{\overline{u}}$	σ_k
3	1.15	0.87
4	0.58	1.73
5	0.38	2.60
6	0.29	3.46
7	0.23	4.33

Table 6.2. Reliability index, 12 balanced peers, AA algorithm

error would lie below the value 1m for all the considered number of available satellites.

AA Algorithm with Biased Aiding

The biased aiding scenario is populated by four aiding peers displaced at -5m with respect to the altitude of the user. This might be the case of a user crossing a flyover and helped by the peers on the road underneath. The experiment is led as in the case of the first scenario, so that the same information given in the introduction to the first scenario holds true, if not differently specified. The algorithm adopted to compute the PVT solution is again the AA one, in order to show its limits.

The results of the simulations are summarized in Figure 6.9 and in Table 6.3.

# satellites	$r_{\overline{u}}$[m^{-1}]	σ_u[m]	RMSE[m]
3	0.67	1.5	5.01
4	0.08	11.67	1.98
5	0.05	17.51	1.71
6	0.04	23.34	1.58
7	0.03	29.18	1.49

Table 6.3. Reliability index, 4 unbalanced peers, AA algorithm.

The data shown in Table 6.3 allows to understand why the second order moment is a useful tool to identify a biased aiding data.

In the case of four or more available satellites, the receiver estimates the altitude computed using only the GNSS data. The reliability index is computed as in (6.8) and it points out the bias that affects the altitude aiding. σ_u^2 is then set in order to lessen the impact of the aiding measurement on the computation of the PVT.

In the case of three satellites only, despite the aiding data is not reliable, the algorithm allows anyway to compute the position of the user. This position will not be precise, due to the unrecognized bias that affects the altitude, but it is a better result than the one

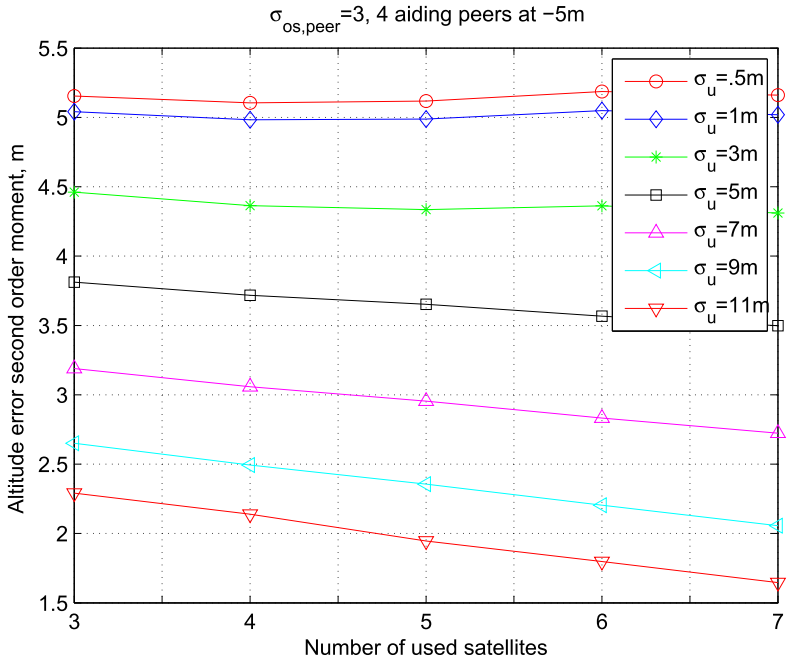

Figure 6.9. Second order moment of the user computed altitude error in the presence of a biased aid and varying the estimated degree of reliability.

achieved by the stand alone GNSS receiver, which is not capable to compute the PVT of the user. The main issue is that in the case of three satellites, the reliability index of the aiding data cannot detect any bias that may affect the altitude aiding.

Considering the fourth column in Table 6.3, it can be noticed that the root mean squared error would be greater than 5m when only three satellites are available, but it would decrease very quickly exploiting the reliability index in (6.8), which is the one based on the second order moment of the aiding measurements.

It can be stated that when only three satellites are available, the parameter defined in (6.7) is not sufficient to get rid of the bias that affects the aiding data and a different approach must be adopted. To this point, the IAA-PVT algorithm is proposed.

6.4.4 IAA Algorithm with Biased Aiding

Since the AA algorithm showed a weakness in the presence of a biased aiding, an IAA algorithm was developed and tested.

The formerly considered biased aiding scenario is a good one to test the IAA-PVT algorithm described in Section 6.2.3.

The setup is similar to the former one, but the considered time span of data is 370s long. Figures 6.10 and 6.11 are useful to understand that the IAA algorithm has a different behavior with respect to the AA algorithm.

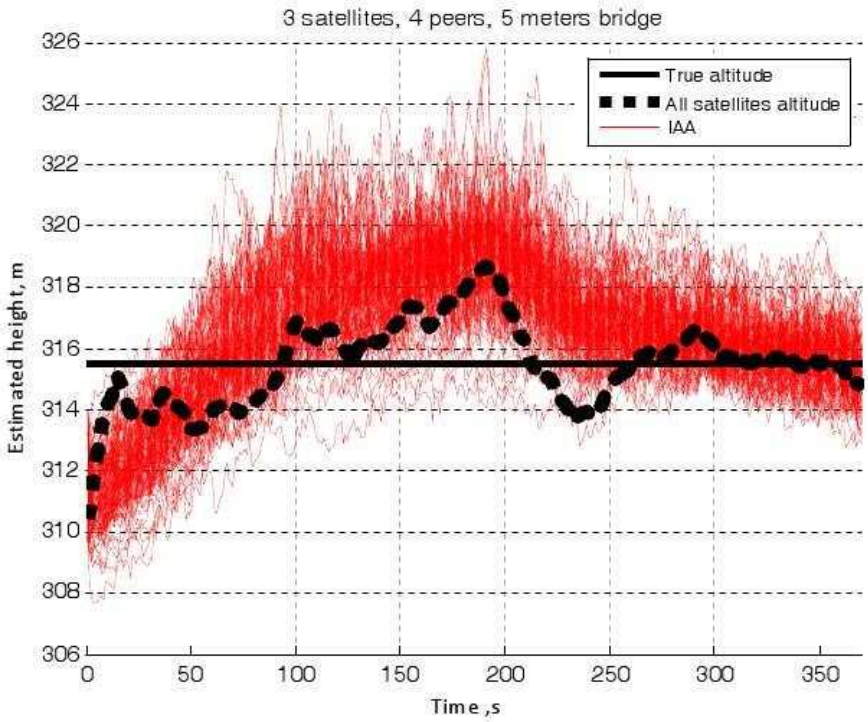

Figure 6.10. IAA algorithm based estimation of the altitude, 3 satellites, $w = 0.85$.

These figures show the trend of the estimated altitude $h_0[k]$ in time, considering 100 different simulations. The constant plot shows the true value of the altitude. Of course, only the case of three visible satellites is considered, because the IAA algorithm is designed to work only in this circumstance.

Referring to Figure 6.10, despite the biased aiding the IAA algorithm manages to compute an estimate of the altitude that is less affected by the bias than the one computed by the AA algorithm. In the figure, also the altitude solution given by a standard receiver that exploits 10 satellites is shown and labeled as "All satellites altitude", in order to provide the trend of a purely GNSS solution.

The performance of the AA algorithm, characterized by $\sigma_u = 1.5$m, are shown in Figure 6.11.

Table 6.4 shows a comparison between the AA algorithm and the IAA algorithm,

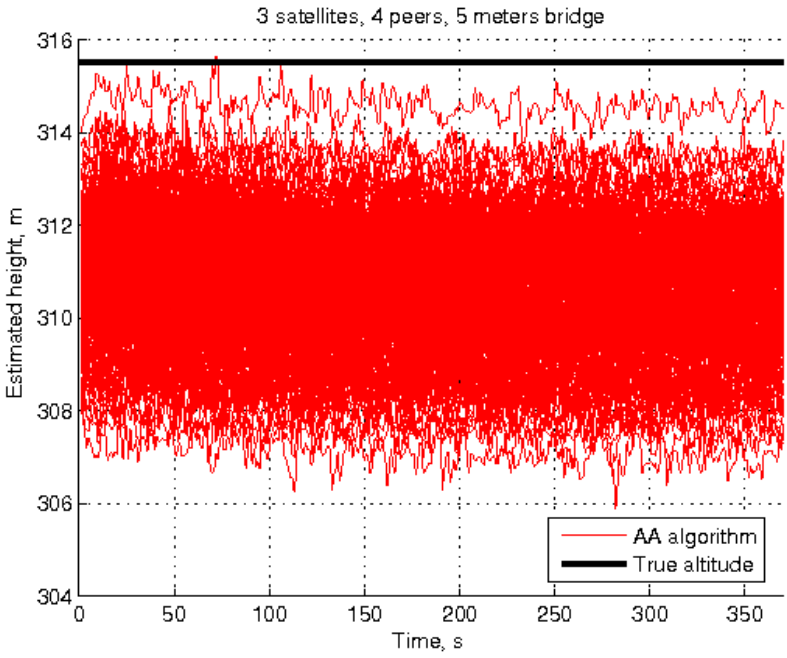

Figure 6.11. AA algorithm based estimate of the altitude, 3 satellites.

Method	Mean value [m]	RMSE [m]
AA	310.47	5.04
IAA	315.36	0.16

Table 6.4. Reliability index, 4 unbalanced peers, -5m bias.

taking into account the mean value of the estimate and the second order moment of the computed altitudes with respect to the true altitude, computed on the basis of 200 iterations and considering the data in the time gap between 340s and 360s.

In order to have a general idea about the behavior of the two different algorithms, Figures 6.10 and 6.11 show the estimated altitude trends when three satellites are available and a biased aiding affects the solution.

The performance of the algorithms can be evaluated by means of more generic parametric plots, shown in the following. To this point, it is necessary to introduce the estimate of the altitude at time k obtained from the j-th simulation, namely $h_{0,j}[k]$, and to define a parameter $\bar{h}_0[k]$ as follows

$$\bar{h}_0[k] = \frac{1}{21 \cdot N_s} \sum_{j=1}^{N_s} \sum_{l=-10}^{10} h_{0,j}[k+l], \tag{6.42}$$

where N_s is the number of simulations that were run and l is a parameter that allows for a sliding window computation of the value $\bar{h}_0[k]$. This parameter is plotted in Figure 6.12.

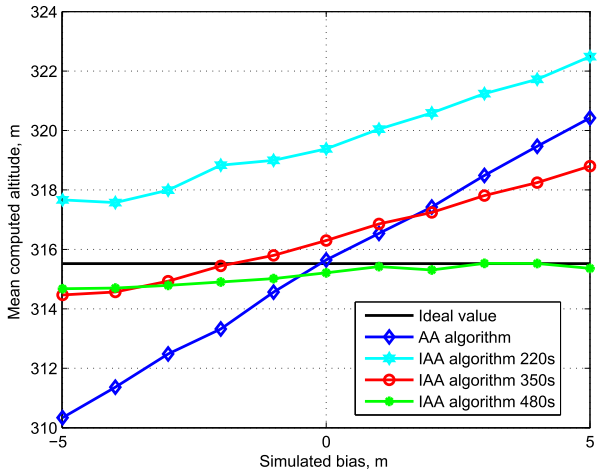

Figure 6.12. Mean value of the computed altitudes, using the AA algorithm and the IAA algorithm, in the presence of different aiding biases and transient times, $w = 0.8$.

When the AA algorithm is run, the mean value of the computed altitude is heavily affected by the bias, due to the low variance estimated by means of the reliability index (6.7). By the way, when the IAA algorithm is applied, some performance enhancement is possible.

In Figure 6.12 the performance of the iterative system is shown in terms of $\bar{h}_0[220]$s, $\bar{h}_0[350]$s, $\bar{h}_0[480]$s. The performance of the system based on the IAA algorithm is heavily affected by a transitory, but it allows to better remove the bias that affects the aiding. In fact, while the performance of the IAA algorithm after 220s is still affected by the bias, it outperforms the AA algorithm after 350s and over.

The second parameter taken into account is represented by the second order moment of the computed altitudes with respect to the true altitude, that is shown in Figure 6.13.

The performance of the AA algorithm is completely conditioned by the bias that affects the aiding data, because this algorithm relies totally on the aiding, without any chance to detect the bias. In fact, the computed altitude is prone to settle around a value which is equal to the true altitude plus the bias.

The performance of the IAA algorithm is again dependent on time and resembles the results shown in Figure 6.12. In fact, while after 220s the algorithm performs even worse

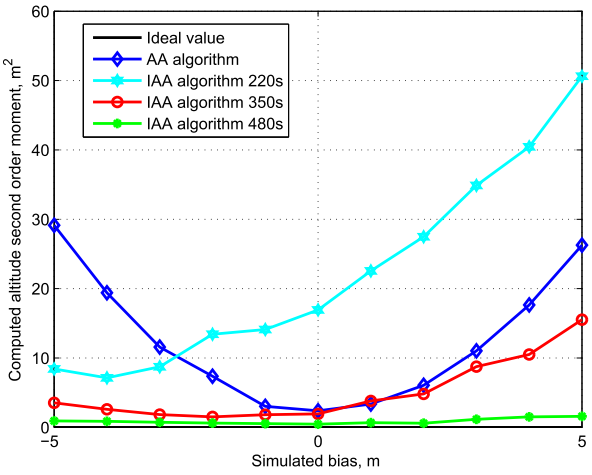

Figure 6.13. Second order moment of the computed altitudes, using the AA algorithm and the IAA algorithm, in the presence of different aiding biases and transient times, $w = 0.8$.

than the AA algorithm, after 350s it manages to outperform the AA algorithm.

Analyzing the trend of the curves shown in Figures 6.10 and 6.11 and the performance curves shown in Figures 6.12 and 6.13, it can be observed that the IAA algorithm outperforms a system based on the AA algorithm in the presence of three satellites, but it requires some time to soften the effects of a biased aiding measurement.

Tests on the Value of the w Parameter

The last parameter that must be taken into account is the weight w involved in (6.28). This parameter determines the contribution of the GNSS solution and of the external aiding in the computation of the aiding. Figures 6.14 and 6.15 show the performance curves in the presence of a -5m bias, considering different weights w.

It is easy to notice that when $w = 1$ the IAA algorithm is almost equivalent to the AA algorithm. This could have been predicted from (6.28) easily. In fact, when $w = 1$, the aiding is computed relying on the external aiding only, but the altitude measurement error variance is computed relying on the GNSS computed altitude as in (6.6).

It is also quite evident that the most proper weight, in this peculiar case, is to be found in the interval $0.8 < w < 1$. For this reason, it might be useful to evaluate the performance curves in a larger number of points in this interval, as shown in Figures 6.16 and 6.17.

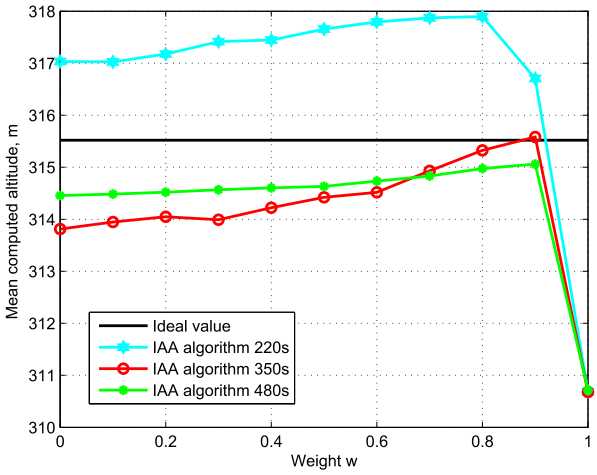

Figure 6.14. Mean value of the computed altitudes, using the IAA algorithm, in the presence of different weights and of a -5m bias.

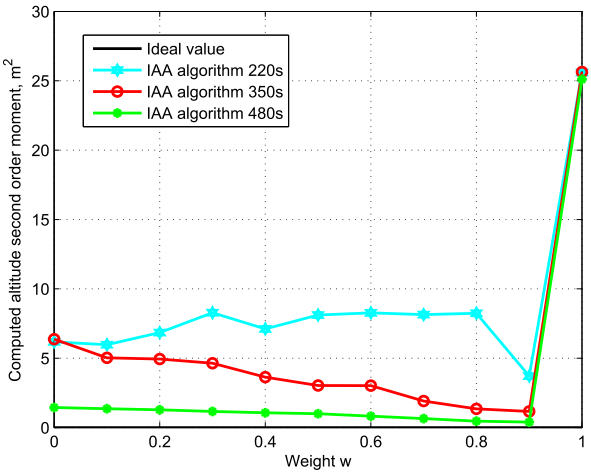

Figure 6.15. Second order moment of the computed altitudes, using the IAA algorithm, in the presence of different weights and of a -5m bias.

Figures 6.16 and 6.17 shows the mean estimated altitude and the second order moment of the estimated altitude for $0.8 < w < 1$, considering values of w that differs by 0.01.

As previously stated, a good value for the parameter w belongs to this interval. In particular, any value of w so that $0.8 < w < 0.9$ allows a system based on the IAA algorithm to perform well.

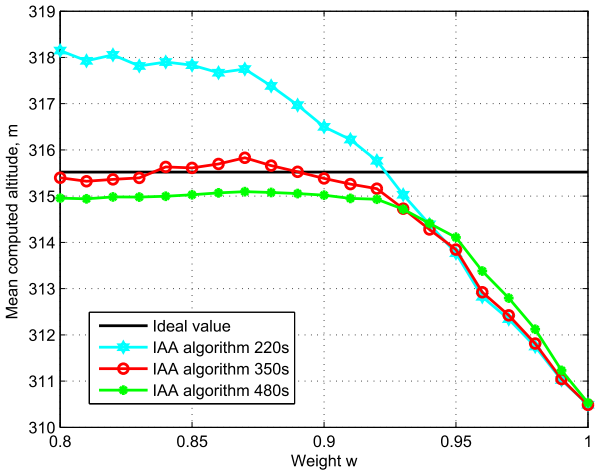

Figure 6.16. Mean value of the computed altitudes, using the IAA algorithm, in the presence of different weights and of a -5m bias, restricted interval.

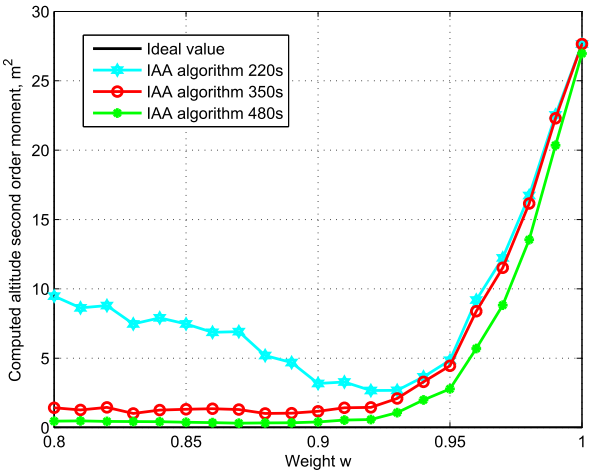

Figure 6.17. Second order moment of the computed altitudes, using the IAA algorithm, in the presence of different weights and of a -5m bias, restricted interval.

By the way, if a greater tolerance is allowed, looking at the performance curve after 220s, it is easy to state that for $w = 0.93$ it is possible to minimize the estimated altitude second order moment when a shorter time has passed.

While scenarios characterized by a sensible bias are possible, in the vehicular environment one can suppose that the aiding altitude bias is often small. To wisely select the weight w, the system designer must take into account the bias range that the user may

experience. To this point, is useful to briefly analyze what happens when a $+5$m bias is present, in opposition to the data shown in Figures 6.14-6.17, and when no bias affects the aiding.

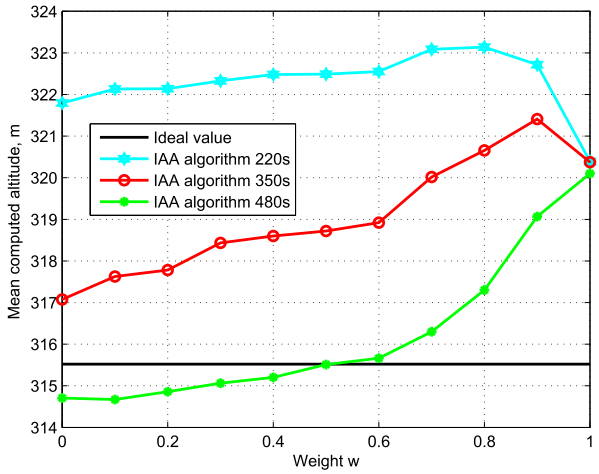

Figure 6.18. Mean value of the computed altitudes, using the IAA algorithm, in the presence of different weights and of a $+5$m bias.

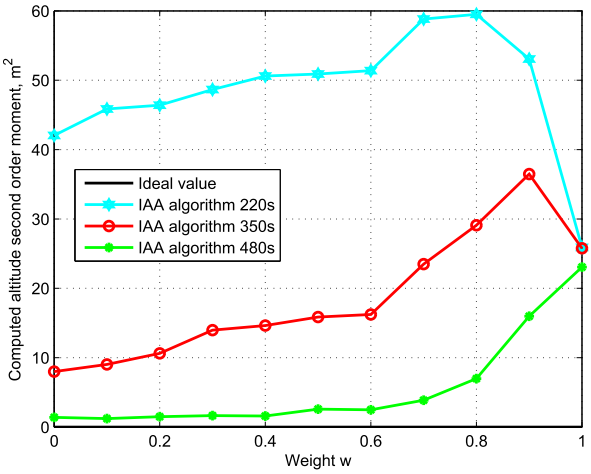

Figure 6.19. Second order moment of the computed altitudes, using the IAA algorithm, in the presence of different weights and of a $+5$m bias.

From a comparison of Figures 6.14 and 6.18 and Figures 6.15 and 6.19, it is possible to state that the most proper value for the weight w is dependent on the available GNSS and aiding data. For this reason, it is necessary to determine a value w as a trade off between

system performance and expected altitude bias range. On the basis of the results obtained in the case of a ± 5m bias, a wise choice can be represented by $0.6 < w < 0.7$.

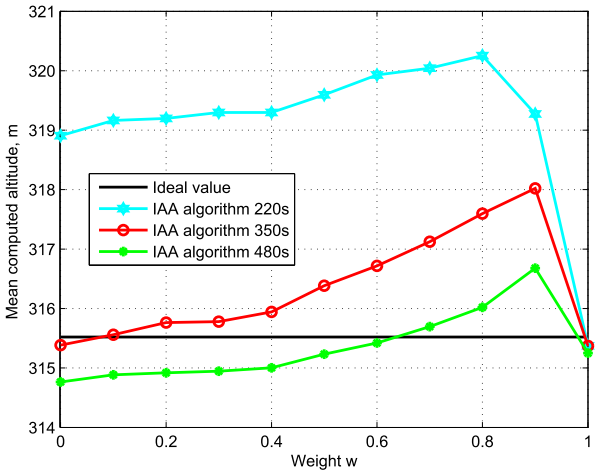

Figure 6.20. Mean value of the computed altitudes, using the IAA algorithm, in the presence of different weights and of a 0m bias.

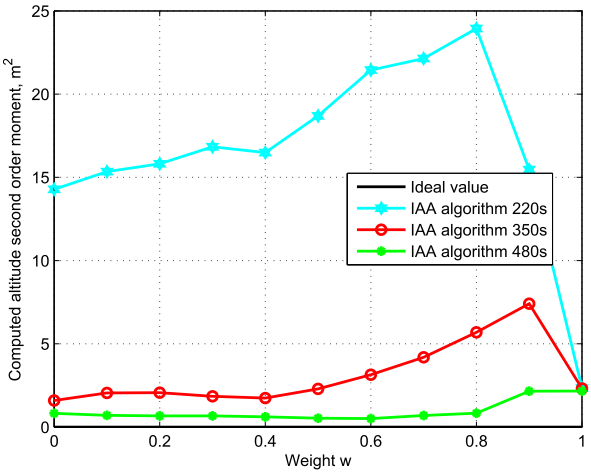

Figure 6.21. Second order moment of the computed altitudes, using the IAA algorithm, in the presence of different weights and of a 0m bias.

Figures 6.20 and 6.21 allows to determine what is the performance of the system when $0.6 < w < 0.7$ and no bias affects the altitude aiding. Even in this case, a trade off between performance and bias range brings to a value of the weight so that $0.6 < w < 0.7$.

The selection of the w value is not easy and it should be chosen after an analysis of

the possible scenarios and of the required minimum performance of the receiver. Better receivers would allow a higher measurement rate and more precise measurement, while a commercial receiver offers limited performance. Despite the various aspects to be taken into account to design the system properly, the IAA method allows to compute the position even if only three satellites are available, relying on an external aiding that can even be affected by a bias, which is the main aim of the algorithm and the main advantage with respect to the AA algorithm.

6.4.5 P2P Simulator Results

As requested by the ESA, a P2P simulator was developed, in order to have a good test bench to perform different trials without caring for data collections.

This simulator allows to create an environment where peers can communicate and run the algorithms.

In the simulated scenario, five peers are distributed in the area. Four of them, labeled 2, 3, 4 and 5, are capable to compute their own position and act as aiding peers. The first peer instead, plays the role of the aided peer.

Both the AA and the IAA algorithms are tested in this scenario. When the AA algorithm is tested, the simulator generates 5 satellites for the aided peer. When the IAA algorithm is tested, only three satellites are considered, even if more satellites may be available. This is due to the fact that the IAA algorithm was designed to work only when three satellites are available and only at the beginning of the operation, as discussed in the previous sections. The considered scenario is shown in Figure 6.22.

We run 10 simulations, each one made up by 15 time slots. The initial guess of the position of the users is gaussianly distributed with a 3m standard deviation in the horizontal plane and a 4m standard deviation in the vertical axis.

For both algorithms, only the performance of the aided peer are considered, since it is the only relevant one in this case. We show the plots at time slots 1, 4, 8, 12 and 15, in order to make the plots more comprehensible. We start considering the results obtained for the AA algorithm.

Looking at the left side of the Figure 6.23, it is possible to observe the trend of the cumulative probability as a function of the modulus of the error. It can be noticed that the precision of the horizontal estimate improves with time and the error manages to be not greater than 10m in the 90% of the observed realizations. By the way, looking

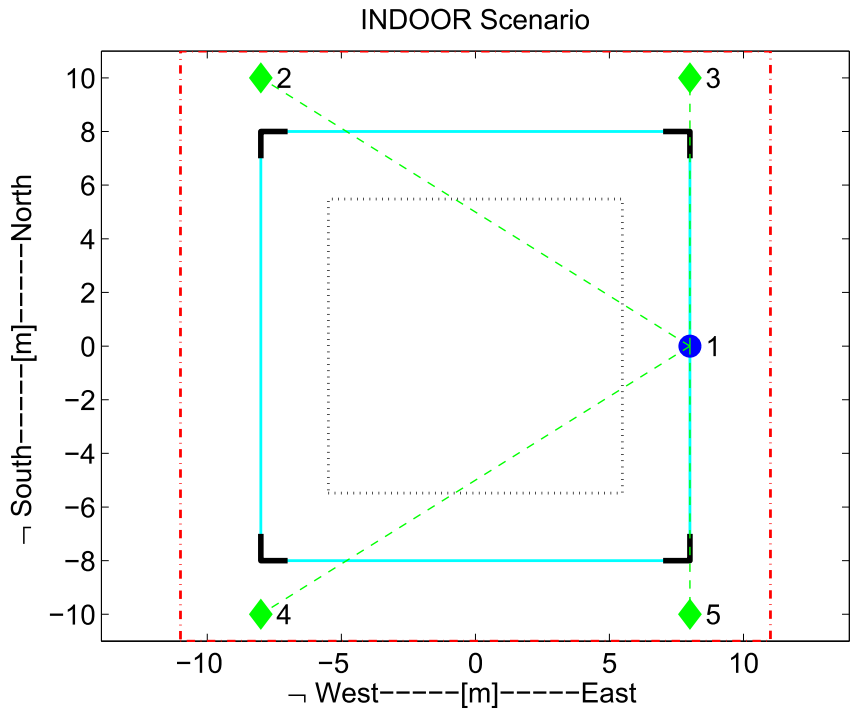

Figure 6.22. Scenario considered in the simulations.

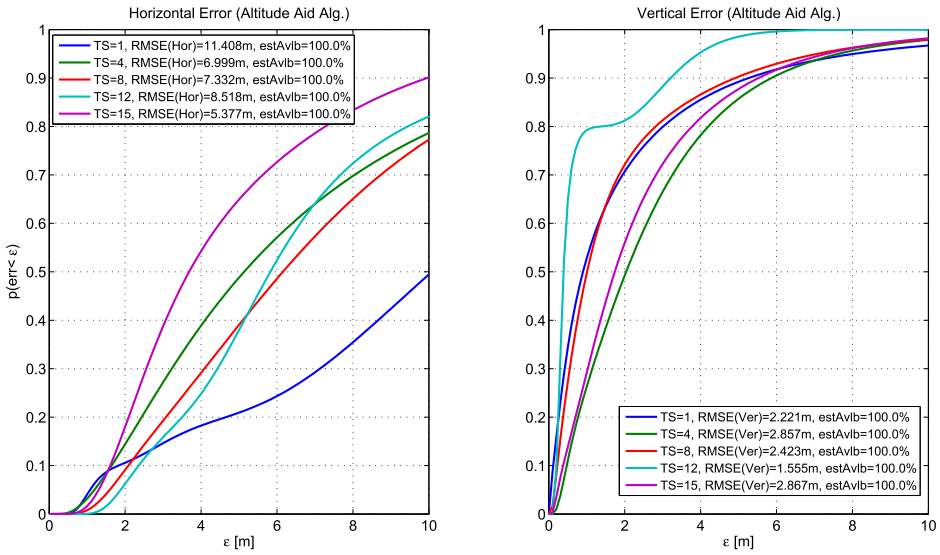

Figure 6.23. Results for the AA algorithm.

at the right side of the plot, a comparison with the estimate of the altitude shows that the precision of the estimate may not only be a function of the time, but mainly of the available measurements. In fact, the best altitude estimate is available at the 12th time slot.

We can now proceed with the results obtained running the IAA algorithm. The main

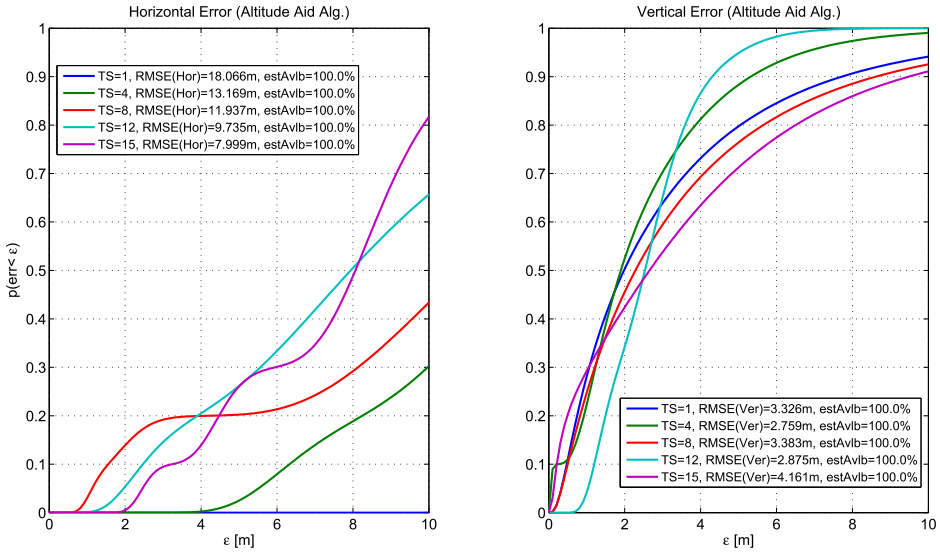

Figure 6.24. Results for the IAA algorithm.

advantage of this algorithm is that positioning is possible even if only three satellites are considered. Since a lower number of satellites is available, the horizontal positioning performance are worse than the previous case. By the way, the altitude aid allows for a good precision related to the estimation of the altitude.

Two main conclusions can be drawn from the analysis of the presented results:

- the performance of the algorithm is not so strictly dependent on time, but it is possible to state that it generally improves with time;
- since the aiding is related to altitude, the performance is mostly improved in terms of the estimate of the altitude.

The simulations confirm what was stated in the simulations with real data. By the way, different plots are shown and discrepancies may be present due to the adopted pseudorange model, which is far from real.

This algorithm has its main pivotal points in its simplicity and in the small modifications that must be applied to a standard stand alone receiver.

6.5 Results: Pseudorange Aiding

6.5.1 Simulation Setup

In order to collect the data required to run the simulations, a uBlox5 receiver linked to a georeferenced antenna located on the roof of the NavSAS laboratory (http://www.navsas.eu/) and connected to a PC running the data grabber software were used. All the data, collected at a rate equal to 1Hz, consisted in the values necessary to run the range aiding algorithm in post processing.

The data set considered in the simulations lasts 360 seconds and it stores data related to the satellites in the LOS of the antenna. No carrier smoothing is applied to the collected pseudoranges. Four satellites were selected to run the simulations and they are listed in Table 6.5.

Sat no.	PRN	Azimuth [deg]	Elevation [deg]
1	6	326.5	80.4
2	18	71.7	45.4
3	16	184.3	43.8
4	19	298.2	41.1

Table 6.5. PRNs, azimuths and elevations of the considered satellites.

At first, all the four selected satellites were used to compute the position of the user, in order to obtain a reference. In the simulation, in order to assess the performance of the range aiding algorithm, the aided user is provided with the pseudorange measurements related to three satellites in the selected set and with a geometrical distance between the discarded satellite and a simulated aiding peer.

Only one aiding peer is simulated, in order to observe the effectiveness of the method with respect to the disposition of the aided user and of the aiding peer. Its position is equal to the GPS solution obtained when four satellites are available, translated in longitude and latitude, accordingly to the parameters set in the simulator.

Moreover, a mean performance is estimated considering different disposition of the user and computing the mean result. This is an approximation to the case of a large number of equally reliable peers displaced around the aided user.

6.5.2 Results

In this section the results of the simulations will be analyzed. In the first experiment, the aiding peer is placed 10m away from the aided peer. In each simulation one satellite is discarded and the range between the discarded satellite and the aiding peer is computed and transmitted to the aided peer. For each discarded satellite, the aiding peer is moved by 2.5deg, keeping the distance from the aided user constant, and the 360s long data span is processed, using the obtained aiding range, second by second.

Once the processing is complete, the centre of mass of the computed trajectory is compared with the centre of mass of the reference trajectory. The trajectory used as a reference is the one obtained using four satellites. It is useful to compare the centres of mass, because the aided user and the simulated peer are still.

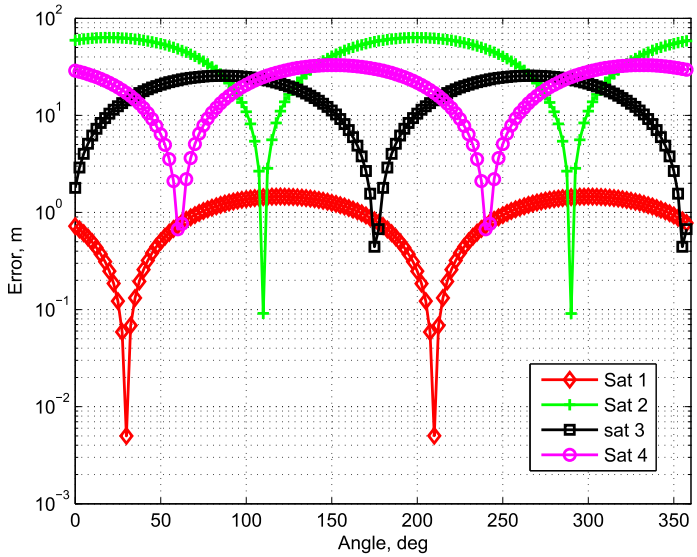

Figure 6.25. Distance between the centre of mass of the range aided solution and of the four satellite solution as a function of the angle between the aided user and a 10m far aiding peer.

The results are shown in Figure 6.25. The plot shows the trend of the estimation error with respect to the angle between the aided user and the aiding peer. From the plot it is evident that the performance of the algorithm is strictly dependent upon the elevation and azimuth of the satellite and upon the disposition of the aided user and of the aiding peer. This result is coherent with the geometrical nature of the problem. From the plot in Figure 6.25, it is possible to determine approximately for each satellite which is the best

and the worst angle between the two users. These angles are separated by $90°$ and each one is characterized by a $180°$ period. These data, reported in Table 2, are used to perform the next trial.

Sat no.	PRN	Best [deg]	Worst [deg]
1	6	30	120
2	18	290	200
3	16	175	85
4	19	60	150

Table 6.6. Angles between users: approximated best and worst values.

It must be noticed that for every discarded satellite, the error tends to zero and it is equal to zero when the aiding range replaces exactly the unavailable pseudorange measurement.

Once the dependency upon angles has been discussed, a similar analysis can be held for the distance between users. Since the best and worst angular values are known, it is possible to evaluate the performance of the algorithm as a function of distance, considering only two angular values. The plots will then show the best and worst performance as a function of distance between the aided user and the aiding peer. In Figure 6.26 the results of this analysis are shown.

To obtain this plot, in each simulation one satellite is discarded. For each discarded satellite, the simulation is run considering at first the worst angle related to the discarded satellite. The data span analyzed is the same used in the first experiment. At each simulation, the aiding peer is moved by 5m along the selected direction, starting from a distance of 0m, up to a distance of 200m. The same simulation is run a second time, using the best angular value related to the same discarded satellite. The worst and best values are reported in the aforementioned Table 2.

Again, the performance of the algorithm is assessed in terms of the distance between the centres of mass of the computed trajectory and of the reference trajectory.

From the plot it is evident that the performance of the algorithm is again strictly dependent upon the satellite elevation and azimuth and also upon the distance between the aided user and the aiding peer. Also this result is coherent with the geometrical nature of the problem.

Both the aspects analyzed up to this point are related to the geometry of the problem. In fact, larger distances and displacement of the users involved in the computation

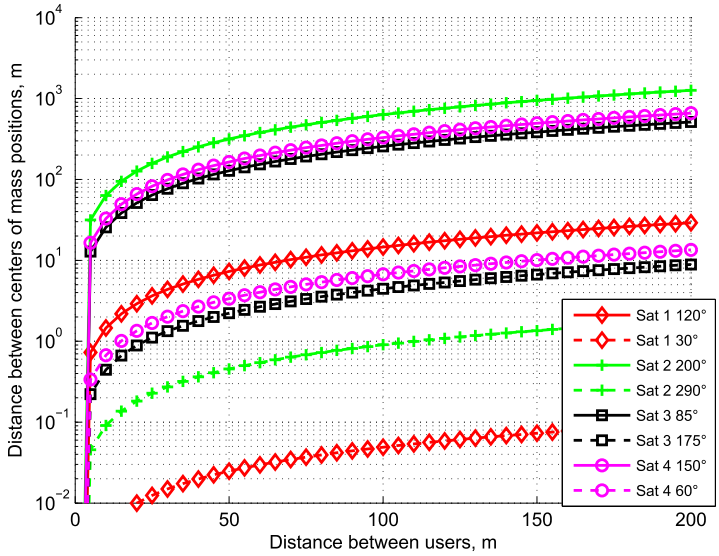

Figure 6.26. Distance between the centre of mass of the range aided solution and of the four satellite solution as a function of the distance between the aided user and an aiding peer placed at the worst and at the best angle available.

characterize the computed aiding range. The aided user will perform at its best when the aiding range is similar to the missing range value, which is when the aiding user has the same distance of the aided user from the discarded satellite. Moreover, as it can be understood analyzing Figure 6.26 together with Table 1, the range related to a satellite characterized by an higher elevation is less sensitive to the displacement of the users, because it varies more slowly.

In Figure 6.27, the position computed removing the third satellite and considering different angles are shown. The purpose of this figure is to give an idea of the behavior of the algorithm, keeping the distance between users constant and changing the angle. The computed solution is translated along a direction. This translation is a bias due to the incorrect value of the range exploited in the algorithm. By the way, the centre of mass of these set of solutions, which is the point depicted by a square, almost coincides with the centre of mass of the four satellite solution, with an error in the order of 10-5m. Figure 6.27 points out that when the peers are biasedly distributed around the aided peer, as it is the case of the single aiding peer, the performance of the system are strictly dependent upon the geometry of the scenario.

It is also interesting to evaluate the mean trajectory obtained from the trajectories in

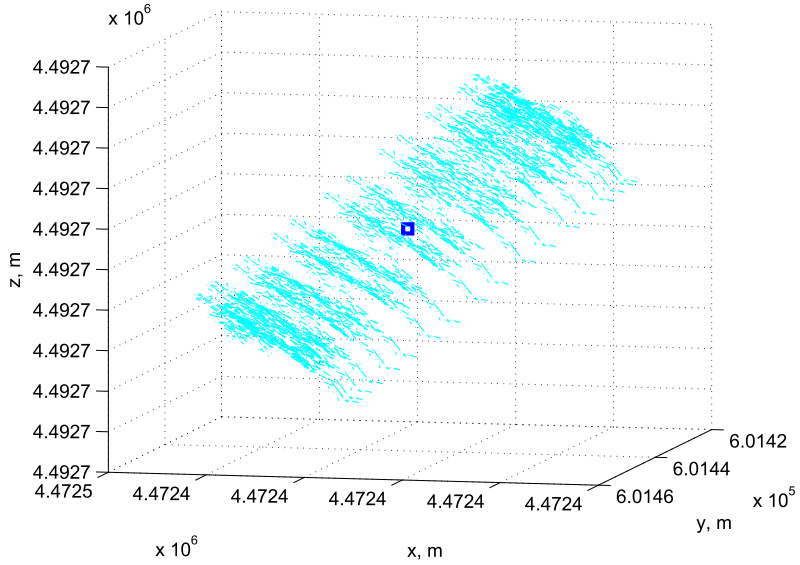

Figure 6.27. Positions obtained for different displacements of the aiding peer.

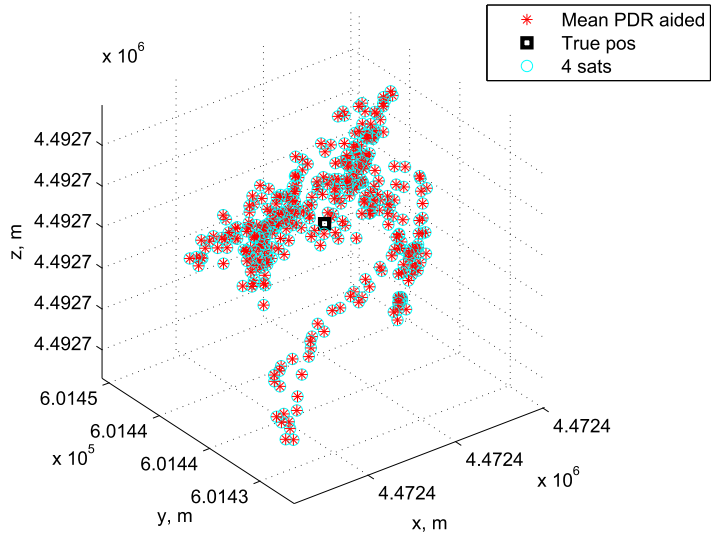

Figure 6.28. Mean performance of the range aiding algorithm.

Figure 6.27. This mean trajectory is shown in Figure 6.28. It is evident that point by point the solution obtained using 4 satellites and the solution obtained discarding one satellite are almost the same. This plot is a good approximation of the performance of

the algorithm in the presence of a large number of peers, well disposed around the aided user. In this event, the performance of the algorithm is acceptable and the range aided solution is very close to the four satellite solution

6.6 Summary

This chapter considered two different systems aided by external data available thanks to a P2P network.

In the first case, the positioning algorithms exploit altitude measurements and we evaluated the effect on the performance of such a system due to a biased altitude aiding. The performance of the system was stated in terms of the altitude estimate derived from the computed GNSS solution.

The examined topic has already been considered in the literature and in some avionics solutions, but here we extend the idea of altitude aiding to a new application scenario, which is the one of a cooperative P2P network. Here, the altitude aiding is computed exploiting the altitude data received from other peers in the same P2P network that are able to obtain their own positions thanks to a GNSS receiver.

The reliability of this particular aiding was at first stated in terms of the distribution of the altitude measurements received by the aiding peers, defining a reliability index.

Two different algorithms, to be implemented in a altitude aided GNSS receiver, were shown. The first one, the AA algorithm, relies on the reliability index and manages to improve the performance of the GNSS receiver when more than four satellites are available.

When three satellites are available, the AA algorithm may improve the performance of the system, but due to the definition of the reliability index when only three satellites are available, it turns out to be unreliable when only three satellites are available and the aiding peers are displaced at a different altitude with respect to the aided user, producing a biased aiding data. Due to this limitation, using the AA algorithm in the presence of three satellites only might be risky.

To overcome this issue, an iterative aiding algorithm has been proposed, in order to allow the system to perform better if the aiding data is corrupted by a bias and only three satellites are available. The algorithm has been tested by means of simulations and a considerable performance improvement has been observed when only three satellites are available, which is the case for which the IAA algorithm was designed for.

The global and main result of this study is that a combined use of the reliability index together with the AA algorithm and the IAA algorithm allows to obtain more robust positioning at a very low cost, since the architecture of the receiver is only slightly modified.

A second algorithm that exploits a range based aiding has been analyzed. Also this algorithm is characterized by a low complexity and by small modifications with respect to the stand-alone algorithm, so that it can be easily used to improve the performance of a receiver that is able to exploit external measurements.

Chapter 7

Conclusions

The purpose of this thesis was to consider the applications of the Kalman filter in the field of navigation. The topics discussed and the results obtained so far prove that the Kalman filter is a valuable instrument whereas a satellite based navigation system is available.

This work was ideally divided in two parts. The first part, made up from Chapters 1, 2, 3 and 4 was devoted to the introduction of the problems and the theory related to navigation based on satellites and inertial systems and to the discussion of the basics of the Kalman filter theory. The second part, made up by Chapter 5 and 6 was focused on the applications of the Kalman filter to the GNSS receivers. In Chapter 5 both the stand-alone receivers and the integrated architectures were analyzed, describing the mathematical model associated to the filters and showing some results. In Chapter 6, three new algorithms based on the Kalman filter and relying upon the existence of a P2P network were discussed, analyzed and tested, showing some interesting results on their performance.

Many observation related to the Kalman filter can be drawn on this study:

1. The Kalman filter can be easily implemented in a software GNSS receiver to compute the PVT solution. It performs better than a LS filter, since it provides the user with smoother solutions, and it does not entail a high computational load.

2. The Kalman filter is also a valuable tool whereas sensors integration is required. A large number of more complex filters have been discussed in the literature, but the Kalman filters are widely used thanks to their simplicity and reliability.

3. What is difficult about the Kalman filter is represented by the tuning operation. The noise and measurement covariance matrices must be properly defined in order to

properly weight the contributions of the measurements and of the model/sensors.

4. Many new applications of the Kalman filter may be designed, despite the related theory is old. The three new algorithms discussed in this thesis are just an examples of the usefulness and flexibility of this filter. These algorithms are characterized by a complexity which is compared with the one of a stand-alone GNSS receiver and by a higher reliability, since they exploit external measurements that are not available to a stand-alone receiver.

Finally, the results addressed in this thesis represent a good starting point for further developments:

1. Other algorithms can be applied to the satellite based navigation and their performance can be compared with the one of the Kalman filter, to understand if a heavier computational load can be justified by better performance.

2. Other applications of the Kalman filter to the GNSS receivers may be analyzed. As an example, the vector receiver architecture is an interesting application of Kalman filters and it is usually adopted in receivers that implement an INS-driven tracking stage.

3. P2P based algorithms can be further improved and analyzed. Moreover, it is easy to think about different algorithms that exploits ranging measurements between the peers that join the network.

Bibliography

[1] S. Page. Global positioning system: Assessing national policies. Technical report, RAND, 1995.

[2] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1998.

[3] E.T. Jaynes. *Probability Theory: the Logic of Science*. Cambridge University Press, 2003.

[4] S. M. Kay. *Estimation Theory, vol. 1 of Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993.

[5] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, 1995.

[6] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signal and Applied Kalman Filtering*. John Wiley and Sons, 3 edition, 1997.

[7] S. Haykin. *Adaptive Filter Theory*. Prentice Hall PTR, 2001.

[8] E. D. Kaplan and C. Hegarty. *Understanding GPS: Principles and Applications*. Artech House Publishers, 2 edition, November 2005.

[9] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92:401 – 422, 2004.

[10] Mark G. Petovello and Gerard Lachapelle. Comparison of vector-based software receiver implementations with application to ultra-tight gps/ins integration. In *ION GNSS 2006*, 2006.

[11] S. Yamaguchi and T. Tanaka. Gps standard positioning using kalman filter. pages 1351 –1354, oct. 2006.

[12] J.D. Weiss. Analysis of upgraded gps internal kalman filter. *Aerospace and Electronic Systems Magazine, IEEE*, 11(1):23 –26, jan. 1996.

[13] C. Hide, T. Moore, and M. Smith. Adaptive kalman filtering algorithms for integrating gps and low cost ins. pages 227 – 233, apr. 2004.

[14] Mark G. Petovello. *Real-Time Integration of a Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation*. Ph.d. dissertation, University of Calgary (CA), 2003.

[15] Frank H.P. Fitzek and Marcos D. Katz, editors. *Cooperation in Wireless Networks: Principles and Applications*. Springer, 2006.

[16] Isabelle Kraemer and Bernd Eissfeller. A-gnss a different approach. *Inside GNSS*, September/October:52–61, 2009.

[17] J. Bao and Y. Tsui. *Fundamentals of Global Positioning System Receivers, a software approach*. John Wiley and Sons, Inc, 2nd edition, 2005.

[18] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance*. Ganga Jumuna Press, 2006.

[19] Navstar Global Positioning System Joint Program Office. *Interface Specification IS-GPS-200 Rev.D, IRN-200D-001*, March 2006.

[20] Minimum operational performance standards for Global Positioning System/Wide Area Augmentation System airborne equipment (MOPS), RTCA/DO-229C, Washington, DC, USA: RTCA Inc., 2001.

[21] European Space Agency / European GNSS Supervisory Authority. *Galileo Open Service Signal In Space Interface Control Document, OS SIS ICD/D.1, Draft 1*, February 2008.

[22] R D Fontana, W. Cheung, and T Stansell. The modernized l2 civil signal. *GPS World*, pages 28–34, September 2001.

[23] NAVSTAR GPS space segment/user segment L5 interfaces; ICD-GPS-705, Rev. 2. El Segundo, CA, USA: ARINC Research Corporation, 2002.

[24] Parkinson and Spilker. *Global Positioning System: Theory and Applications*. American Institute of Aeronautics and Astronautics, Inc., Washington DC, 1996.

[25] L. Scott, A. Jovancevic, and S. Ganguly. Rapid signal acquisition techniques for civilian and military user equipment using dsp based fft processing. *Proceedings of 14th International Technical Meeting of the Satellite Division of the Institute of Navigation*, 2001.

[26] PA Roncagliolo and JG Garcìa. High dynamics and false lock resistant gnss carrier tracking loops. In *Proceedings of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS*, 2007.

[27] Marco Rao, Letizia Lo Presti, Maurizio Fantino, and Giovanni Garbo. Phase lock loop false lock avoidance in the presence of global navigation satellite system signal.

In *13th IAIN World Congress, Stockholm, October, 2009*, 2009.

[28] Marco Rao, Letizia Lo Presti, Maurizio Fantino, and Giovanni Garbo. A software receiver phase lock loop analysis and design to implement adaptive phase tracking using a finite impulse response loop filter. In *13th IAIN World Congress, Stockholm, October, 2009*, 2009.

[29] Marco Rao, Letizia Lo Presti, Maurizio Fantino, and Giovanni Garbo. A software receiver adaptive phase lock loop method. In *IGNSS 2009 Conference, Surfers Paradise, December, 2009*, 2009.

[30] WC Lindsey and CM Chie. A survey of digital phase-locked loops. In *Proceedings of the IEEE 69*, 1981.

[31] Monica Visintin. Slides from master on navigation and related applications. Technical report, Politecnico di Torino, 2007.

[32] F Legrand, C Macabiau, J Issler, L Lestarquit, and C Melen. Improvements of pseudorange measurements accuracy by using fast adaptive bandwidth lock loops. In *Procedings of the 13th International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City*, 2000.

[33] N.M. Faulkner, S.J. Cooper, and P.A. Jeary. Integrated MEMS/GPS navigation systems. In *Position Location and Navigation Symposium*, 2002.

[34] D.H. Titterton and J.L. Weston. *Strapdown Inertial Navigation technology, 2nd edition*. The Institution of Electrical Engineers, 2004.

[35] A. Gelb. *Applied Optimal Estimation*. M.I.T. Press, 1974.

[36] Letizia Lo Presti, Marco Rao, and Simone Savasta. How to define a fully-digital state model in the kalman filter applications? *Inside GNSS*, pages 20–25, September 2010.

[37] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global Posotioning Systems, Inertial navigation, and Integration*. John Wiley, Hoboken, New Jersey, second edition, 2007.

[38] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance*. Ganga Jumuna Press, 2006.

[39] P. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008.

[40] Niklas Hjortsmarker. *Experimental System for Validating GPS/INS Integration Algorithms*. Ph.d. dissertation, Lulea Univeristy of Technology, 2005.

[41] A. Giremus, A. Doucet, V. Calmettes, and J.Y. Tourneret. A rao-blackwellized particle filter for ins/gps integration. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[42] Rudolph van der Merwe and Eric A. Wan. Sigma-point kalman filters for integrated navigation. In *60th Annual Meeting of the Institute of Navigation*, 2004.

[43] John L. Crassidis. Sigma-point kalman filtering for integrated gps and inertial navigation. In *AIAA Guidance, Navigation and Control Conference*, 2005.

[44] Christopher Jekeli. *Inertial Navigation Systems with Geodetic Application*. 2001.

[45] Swarna Ravindra Babu. *Ultra-Tight Integration of GPS/Pseudolites/INS: System Design and Performance Analysis*. PhD thesis, School of Surveying and Spatial Information Systems, 2006.

[46] Marco Rao, Simone Savasta, Emanuela Falletti, Fabrizio Dominici, Gianluca Marucco, and Antonio Defina. A low cost ins/gps solution comparison for automotive mass market applications. In *International Technical Meeting, San Diego, January, 2010*, 2010.

[47] Saurabh Godha. *Performance Evaluation of Low Cost MEMS-Based IMU Integrated With GPS for Land Vehicle Navigation Application*. PhD thesis, Schulich School of Engineering, 2006.

[48] Weidong Xiang, Paul Richardson, and Jinhua Guo. Introduction and preliminary experimental results of wireless access for vehicular environments (WAVE) systems. In *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, San Jose, July 17-21, 2006*, 2006.

[49] Stephan Eichler. Performance evaluation of the IEEE 802.11p WAVE communication standard. In *Vehicular Technology Conference, Dublin, April 22-25, 2007*, 2007.

[50] Daniel Jiang and Luca Delgrossi. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In *Vehicular Technology Conference, Calgary, 21-24 September, 2008*, 2008.

[51] IEEE draft standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks –specific requirements – part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 7: Wireless access in vehicular environments. *IEEE Unapproved Draft Std P802.11p /D7.0, May 2009*, 2009.

[52] Thomas H. Kerr. A critical perspective on some aspects of GPS development and use. In *16th Digital Avionics Systems Conference, Irvine, October 26-30, 1997*, 1997.

[53] Marco Rao, Letizia Lo Presti, and Jaron Samson. Peer to peer equation augmentation for an altitude aided GNSS receiver. In *Vehicular Technology Conference, Ottawa, September 6-9, 2010*, 2010.

[54] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB, 3rd Edition*. Wiley, 2008.

[55] Department of defense world geodetic system 1984, its definition and relationships with local geodetic systems. Technical report, NIMA, 2004.

[56] Marco Rao, Letizia Lo Presti, and Jaron Samson. Improved GNSS positioning exploiting a vehicular p2p infrastructure. In *5th Advanced Satellite Multimedia Systems Conference, Cagliari, September 13-15, 2010*, 2010.

[57] Marco Rao, Letizia Lo resti, and Jaron Samson. Iterative altitude aiding algorithm for improved gnss positioning. *IET Radar Sonar and Navigation*, to be published.

[58] Heidi Kuusniemi, Gérard Lachapelle, and Jarmo H. Tanaka. Position and velocity reliability testin in degraded gps signal environments. *GPS Solutions*, 8:226–237, 2004.