Map Based Multisensor Railway Localization Enhanced by Raw GNSS Data

Omar García Crespillo

April 2013

This master's thesis has been developed at the Institute of Communication and Navigation of the German Aerospace Center (DLR) within the framework agreement between the University of Malaga and DLR.

Acknowledgments

I would like to express my deep gratitude to Oliver Heirich for giving me the great opportunity to develop this master's thesis under his supervision. His guidance has supposed a really positive and rewarding experience for me.

I am also particularly grateful to Rafael Godoy for his initial support and dedicated advising in this undertaking.

A special acknowledge to DLR staff, who have willingly shared their time to help and assist me throughout my work. In particular, to Fabian de Ponte and Mohammed Khider for the many enriching research discussions we had.

My deep appreciation also to many friends for their invaluable presence in my life. I am especially indebted to Vanessa Muñoz, for sharing with me the last years of my studies and for her affection.

Finally and most important, my special thanks to my mother. Any personal achievement is a result of her unconditional support and encouragement.

Abstract

Precise train localization is essential for railway operation, train control, track management and collision avoidance. Therefore, the positioning of railway vehicles is a safety critical application.

Nowadays, these applications rely on track-side equipment and suppose a high cost in development and maintenance. By contrast, train-side mounted sensors, such as Global Navigation Satellite Systems (GNSS) can achieve railway positioning at a low cost. In particular, the European satellite navigation system Galileo enables safety related applications by new Safety-of-Life (SoL) services.

However, GNSS suffers from line-of-sight problem and its availability and accuracy are dependent of the number of visible satellites. Moreover, stand-alone GNSS can not provide the required precision in parallel track scenarios.

Therefore, in order to achieve a reliable and robust localization for trains, additional train-side sensors and information of a railway map are combined with GNSS measurements.

In this thesis, a probabilistic approach for train localization is proposed by using a Bayesian filter. In particular, a particle filter estimates the topological position of the train in the railway network from raw GNSS measurements (i.e., pseudoranges and Doppler shift frequency), Inertial Measurement Unit (IMU) data and the track map, resulting in a tightly coupled integration. The proposed approach is compared to a loosely coupled integration that uses the PVT solution of the GNSS receiver combined with IMU and the map.

Finally, an evaluation of the tight and loose approach with data recorded in a railway environment is performed. Results are very encouraging and suggest high reliability of the proposed method even in low visibility situations.

Contents

List of Figures xiii

List of Tables xv

List of Algorithms xvii

Acronyms xix

1 Introduction **1**
- 1.1 Motivation 2
- 1.2 Problem Statement 2
- 1.3 Research Question: Raw GNSS Data for Railway Localization 5
- 1.4 Objectives 5
- 1.5 Methodology 5
- 1.6 Brief Overview: Structure of the Thesis 6

2 Theory **7**
- 2.1 Global Navigation Satellite System (GNSS) 7
  - 2.1.1 Fundamentals 9
  - 2.1.2 Observables 10
  - 2.1.3 Absolute Position and Timing 12
  - 2.1.4 Dilution Of Precision 13
- 2.2 Bayesian Estimation 14
  - 2.2.1 Bayes Filter 15
  - 2.2.2 Kalman Filtering 15
  - 2.2.3 Particle Filtering 16
- 2.3 Multisensor Navigation 18
  - 2.3.1 Inertial Navigation 19
  - 2.3.2 Integration Levels 19
- 2.4 Map Based Estimation Methods 21

3 Bayesian Train Localization **25**
- 3.1 Dynamic Bayesian Network 25
- 3.2 Recursive Bayesian Inference 26

3.3 States and Transition Models . . . 28
3.3.1 Movement States Transition . . . 28
3.3.2 Train States Transition . . . 29
3.3.3 Clock States . . . 30
3.3.4 Error Transition . . . 32
3.4 Rao-Blackwellised Filter Approach . . . 33
3.4.1 Particle Filter . . . 35
3.4.2 Likelihoods . . . 36
3.4.3 Error Kalman Filter . . . 40
3.5 Particle Filter Output . . . 41
3.5.1 Highest Weighted Particle . . . 43
3.5.2 Most Likely Particle (ML) . . . 43
3.5.3 Maximum a Posteriori (MAP) . . . 43
3.5.4 Minimum Mean Square Error (MMSE) . . . 44

4 Satellite Geometry Considerations 47
4.1 Satellite Position in Track Frame . . . 47
4.2 Track Dilution Of Precision . . . 49
4.3 Dilution Of Precision with Few Satellites . . . 50
4.3.1 Three Visible Satellites: Known Altitude . . . 51
4.3.2 Two Satellites: Known Altitude and Clock Variation . . . 51
4.3.3 Two Satellites: Railway Assumptions . . . 51
4.4 DOP with One Satellite . . . 52

5 Algorithm Implementation 55
5.1 Initialization . . . 55
5.2 IMU Update . . . 56
5.3 GNSS Update . . . 57
5.4 Filter Configuration . . . 60

6 Framework 63
6.1 System Architecture . . . 63
6.2 Data Structure . . . 65
6.2.1 IMU Data Structure . . . 65
6.2.2 GNSS Data Structure . . . 66
6.3 GNSS Receiver Interfacing . . . 68
6.4 Offline Replay Mode . . . 69
6.5 Visualization Extension . . . 70
6.5.1 Track Probabilities and Output Error . . . 71
6.5.2 Particle Likelihoods . . . 71
6.5.3 GNSS Monitor and Skyplot . . . 72

7 Experimental Setup 73
7.1 Measurement Campaign . . . 73
7.2 Sensors On Board . . . 74
7.2.1 Septentrio AsteRx2 . . . 74
7.2.2 XsensMTx . . . 75
7.3 Ground Truth . . . 75
7.4 Railway Map . . . 76
7.5 Scenario . . . 76

8 Results and Evaluation 79
8.1 System Output . . . 79
8.2 Loosely vs Tightly Coupled Integration . . . 81
8.3 Importance of Doppler . . . 82
8.4 Few Visible Satellites . . . 84
8.4.1 Three Satellites . . . 85
8.4.2 Two Satellites . . . 87
8.4.3 One Satellite . . . 88
8.5 Effect of Error Estimation . . . 90

9 Achievements, Conclusions and Outlook 93
9.1 Achievements . . . 93
9.2 Conclusions . . . 94
9.3 Outlook . . . 95

Bibliography 97

A Satellite Position and Velocity Computation 101

B GNSS Correction Models 105
B.1 Satellite Clock Correction . . . 105
B.2 Relativistic Corrections . . . 105
B.3 Ionospheric Correction . . . 107
B.4 Tropospheric Correction . . . 108
B.5 Error Magnitudes . . . 109

C Code Diagrams 111

List of Figures

1.1 Number of people killed or seriously injured in railway accidents in the EU (2006-2011) 2
1.2 Problem statement 3
1.3 GNSS accuracy in parallel tracks 4
1.4 Train run from Hamburg to Munich: Visible satellites histogram 4

2.1 GNSS position determination 8
2.2 GPS signal generation 9
2.3 Dynamic Bayesian network 15
2.4 Particle filter evolution 17
2.5 Strapdown inertial navigation scheme 19
2.6 Loosely coupled integration scheme 20
2.7 Tightly coupled integration scheme 21
2.8 Naive map-matching 22

3.1 Dynamic Bayesian network of tight integration 26
3.2 Movement states 28
3.3 Train attitude 29
3.4 Train states transition 30
3.5 Clock transition 31
3.6 Effect of receiver clock reset in pseudorange measurement 31
3.7 Error transition 32
3.8 DBN implementation 34
3.9 Estimated particles evaluation with N satellites 37
3.10 Pseudorange Gaussian likelihood 38
3.11 Cloud of particles in a single track 42
3.12 Cloud of particles in a switch situation 42

4.1 Satellite position in track frame 49
4.2 Track position uncertainty 50
4.3 Pseudorange plane and railway track intersection 52
4.4 Track dilution of precision 53

5.1 Particles initialization by weighted least squares 56

5.2 Predicted pseudorange algorithm 58
5.3 Predicted Doppler algorithm 59

6.1 System architecture 64
6.2 Raw GNSS data structure class diagram 67
6.3 u-blox® chipset and evaluation kit 68
6.4 GNSS receiver interfacing 68
6.5 Files parsing and replay diagram (*tightly coupled approach*) 69
6.6 System evolution information 70
6.7 Particles likelihood representation 71
6.8 GNSS information visualization 72

7.1 Rail Driving Validation Environment (RailDriVE®) 73
7.2 Sensors onboard 74
7.3 Leica tachymeter set-up 76
7.4 Chosen test scenario 77

8.1 System output: Comparison of different approaches 79
8.2 MMSE output: hypothesis and position uncertainty 80
8.3 Loosely vs tightly coupled integration comparison 81
8.4 Loose vs tight coupled comparison: mean error 82
8.5 Track uncertainty in combined pseudorange and Doppler navigation 83
8.6 Error comparison with/without Doppler measurement 84
8.7 Satellite constellation in test scenario 84
8.8 Three satellites scenario: LDOP and CDOP 85
8.9 Results: 3 satellites comparison 86
8.10 Test constellations: 2 satellites 87
8.11 Results: 2 satellites comparison 88
8.12 Test constellation: 1 satellite 89
8.13 Results: 1 satellite comparison 89
8.14 Pseudorange likelihood without/with error estimation 91

9.1 Scheme of achieved tasks 93

A.1 Orbital elements 101

C.1 UML raw GNSS data structure class diagram 112
C.2 UML data synchronization scheme class diagram 113
C.3 UML u-blox receiver interfacing class diagram 113
C.4 UML data replay scheme class diagram 114
C.5 UML main classes involved in the filtering 115

List of Tables

7.1 Septentrio AsteRx2 GNSS receiver default parameters 75

8.1 Average CDOP and LDOP values for 3 satellite comparison 85
8.2 Results 3 satellites constellation 87
8.3 Results 2 satellites constellation 88

B.1 Main error sources and their approximate magnitudes 109

List of Algorithms

1 Kalman filter prediction 16
2 Kalman filter update 16
3 Particle filter 18
4 Clock transition model 32
5 IMU propose and weighting 56
6 Raw GNSS propose and weighting 57
7 Satellite position computation 102
8 Satellite velocity computation extension 103

Acronyms

C/N0	Carrier To Noise Ratio
DBN	Dynamic Bayesian Network
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DLL	Delay Lock Loop
DOP	Dilution Of Precision
ECEF	Earth-Centered, Earth-Fixed
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
NED	North, East, Down
NMEA	National Marine Electronics Association
PF	Particle Filter
PLL	Phase Lock Loop
PRN	Pseudo-Random Noise
PVT	Position, Velocity and Time
RBPF	Rao-Blackwellised Particle Filter

RCAS	Railway Collision Avoidance System
RINEX	Receiver INdependent EXchange Format
SMC	Sequential Monte Carlo
SoL	Safety-of-Life
SV	Space Vehicle
TOW	Time Of Week
UTC	Coordinated Universal Time

Chapter 1

Introduction

Nowadays, Global Navigation Satellite Systems (GNSS), in particular the Global Positioning System (GPS), are widely used for a number of non-safety navigation and location-based services, such as car or ships navigation and route planning, surveying land boundaries, recreational sports (e.g. hiking, mountain biking) or many different applications on smartphones.

With the incoming European GNSS system Galileo, a broad scope of new safety applications are being proposed within the context of the so called *safety-of-life* (SoL) services. As Galileo's control rely on the European Union (EU), it can provide a level of integrity that GPS was not able, because its operation depended on US strategical decisions. Therefore, GNSS is now considered of a great potential to support the navigation modules of many safety critical systems, such as aircraft or railway operation.

In railway localization, nevertheless, GNSS suffers from line-of-sight problems and can not thus achieve the required accuracy in few visibility situations and hence, the enough robustness in difficult environments in a stand-alone mode. That is why, it is been suggested that GNSS should be aided with additional information of the railway network and other complementary sensors, such as Inertial Navigation Systems (INS). Due to the highly constraint of the trains to the tracks, railway navigation has some special features not present in other means of transport like cars, ships or planes. The most important consequence of this is that the localization process can be enhanced by using a track map.

In this context, the Institute of Navigation and Communication of the German Aerospace Center (DLR) is studying different approaches to provide a reliable and integrity localization solution to the navigation problem in railway environments. This thesis contributes to that research by facing the navigation problem of trains with GNSS technology, INS and the knowledge of the railway map, as well as by making use of novel localization algorithms approaches.

1.1 Motivation

As shown in Figure 1.1, there are more than two thousand casualties or people seriously injured in railway accidents in the European Union (EU) every year [1]. Moreover, according to the European Railway Agency[1], on average a derailment or a collision is reported at least every second day in the EU, causing significant disruptions to railway operations.

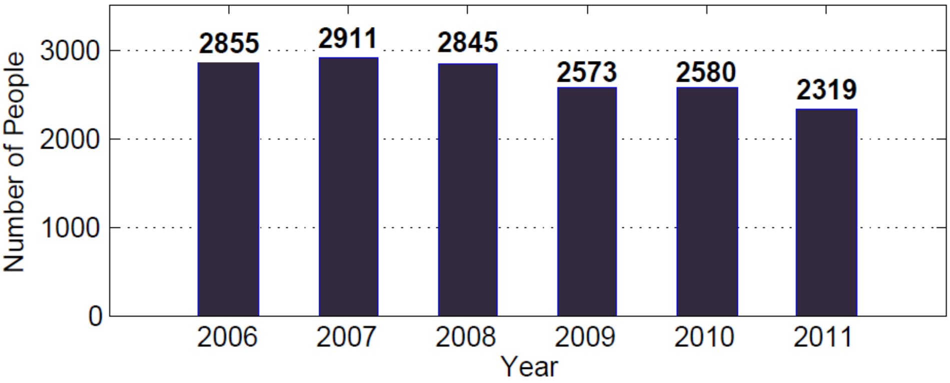


Figure 1.1: Number of people killed or seriously injured in railway accidents in the EU (2006-2011)
Source: Eurostat (rail_ac_catvict) [1]

These statistics point out the deficiency of current train operation systems to cover all possible failure situations, in which the precise localization of the vehicle in the railway network plays an important role to prevent collisions, organize efficiently the traffic, adapt the speed in junctions or curves to avoid derailments, or detect an imminent level crossing.

Actual railway positioning methods rely on track side equipments, such as balises, track circuits or transponders. Although these sensors work efficiently, they provide positioning information at discrete points and their deployment and maintenance costs are very high, so they are reserved to routes with higher density of traffic.
In this context, the ETCS (European Train Control System) is pointing nowadays at new infrastructure-free approaches that would offer a worldwide accurate track-selective positioning solution independent of the track-side elements [2].

New accurate, robust and continuous localization approaches are therefore necessary for train operation, track control and traffic management, as well as for many safety railway applications, such as the Railway Collision Avoidance System (RCAS) [3], [4].

1.2 Problem Statement

In this research thesis, we aim at providing a localization approach that utilizes a set of sensors mounted on the train along with information about the railway network. In

[1] European Railway Agency, http://www.era.europa.eu

particular, the proposed system will have a GNSS receiver with its antenna mounted on the train roof, an Inertial Measurement Unit (IMU) near the wheel axis and a navigation processor that makes use of a digital railway map (Figure 1.2).

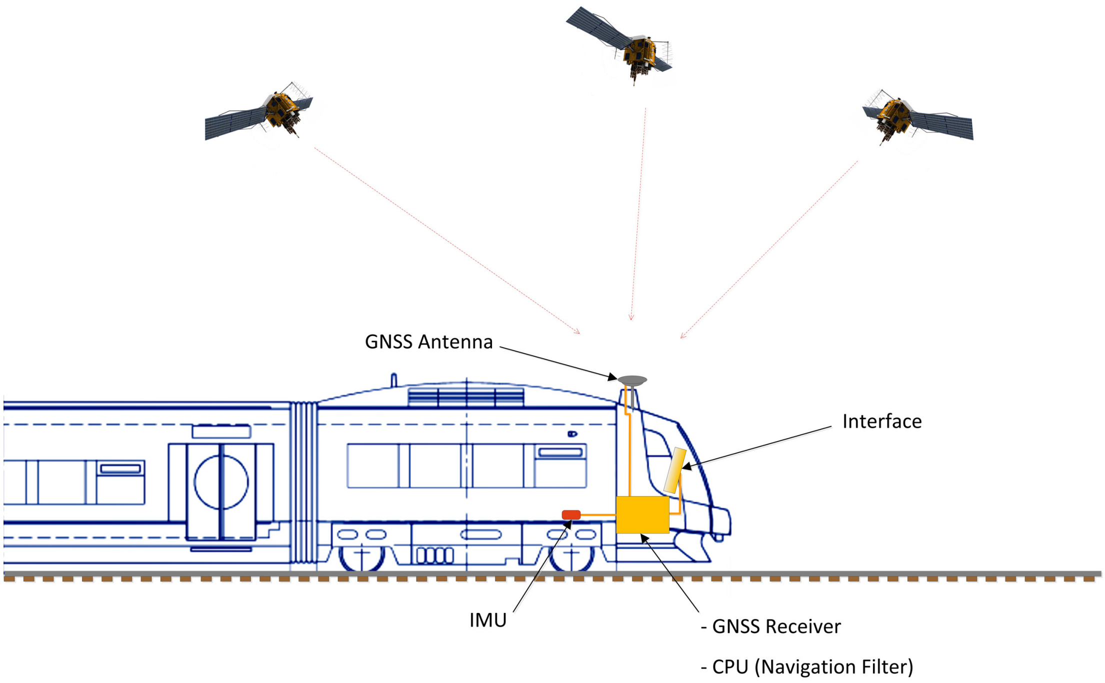


Figure 1.2: Problem statement

Although GNSS receivers provide in general an accurate position, velocity and time (PVT) solution, they present some limitations in railway critical environment. The critical scenarios that should be taken into account for railway positioning are the followings:

- *Parallel tracks:* The accuracy that GNSS can guarantee is not enough to decide between two different tracks in a parallel scenario. As this situation is very common in railway environments, this constitutes a critical problem (Figure 1.3).

- *Switches:* In the same way as for parallel tracks, a GNSS receiver is not able to discern between two tracks after a junction. In this context, the use of an inertial sensor could be used to improve the localization result.

- *Poor satellite signal reception:* GNSS receivers need typically four satellites in view to compute a PVT solution. In total signal blockage moments (i.e., tunnels), they can not provide any information to the user and the localization system must rely only on additional sensors. In other situations, when there are one, two or three visible satellites, the receiver is not able to compute a PVT solution though *there is information* from the visible satellites in the receiver. For example, in Figure 1.4,

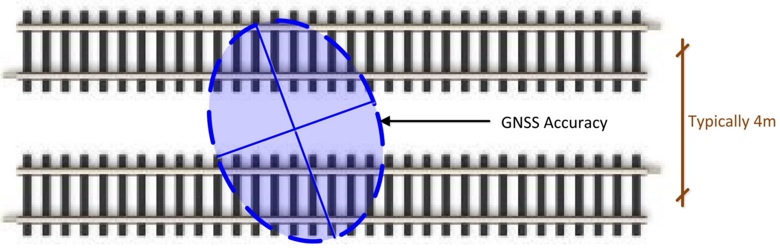


Figure 1.3: GNSS accuracy in parallel tracks

the number of visible satellites observed during a trip from Hamburg to Munich is represented in a histogram over total time. It can be seen how the 7.6% of the time there are one, two or three visible satellites, but the receiver is providing no information to the user.

In order to extract this information and use it into the navigation processor, it is required to handle directly the raw GNSS measurements of the receiver. These raw data are the *pseudorange* which is the measured distance between the receiver antenna and the satellite, and the *Doppler frequency shift*, caused by the relative movement of the user and satellite.

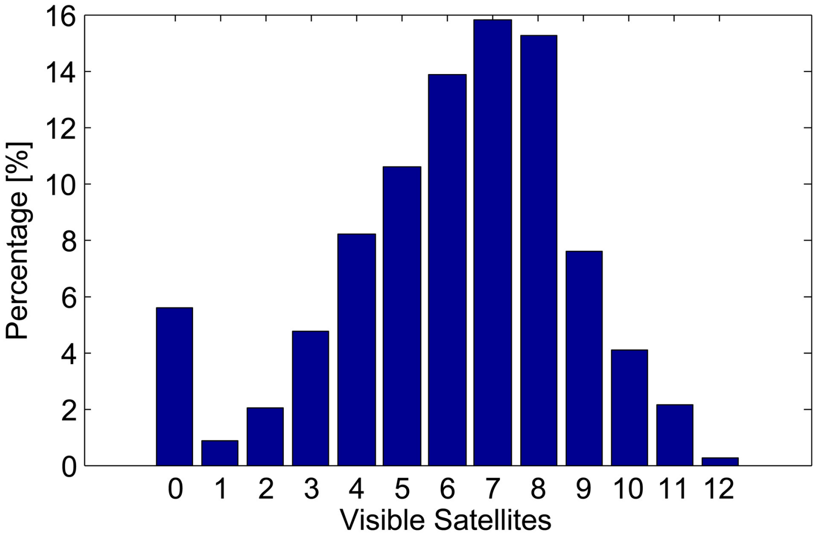


Figure 1.4: Train run from Hamburg to Munich: Visible satellites histogram

In this thesis, the proposed localization approach must be able to compute a track selective and continuous navigation solution taking advantage of the GNSS information even if less than 4 satellites are visible, making an extensive use of all the available data.

1.3 Research Question: Raw GNSS Data for Railway Localization

The use of the PVT solution of the GNSS receiver is widely applied to many applications and systems because it offers intuitive position data about the user location, for example in a latitude, longitude and altitude coordinate system. As our intent is to combine the information of an inertial sensor, a digital map of the railway network and the available *raw data* of a GNSS receiver, the GNSS data is no longer in an intuitive format, but gives us relative information to every visible satellite. How is then possible to use and integrates these measurements together to estimate a position?

Each of these measurement sources provides data in a different way, so the question that comes up is which is the best approach to integrates them in the same application. Understanding of the nature of every sensor is essential to propose an integration solution. However, it is under research and development by the scientific community which is the best method to estimate an accurate and robust train positioning solution within a railway map.

This research work is focused with an special interest on the role of the raw GNSS data, pointing directly to the question of how to apply this information and which are the advantages and disadvantages of using it for this map based multisensor railway localization problem.

1.4 Objectives

The main objectives of this thesis are:

- The theoretical development and the implementation of a *Bayesian approach* that integrates the measurements of a IMU sensor, the raw data of a GNSS receiver and the information of a track map with the purpose of the localization and navigation of a train.

- Evaluation of the proposed approach with *real data* and the comparison with a previous approach that utilizes the PVT solution of the receiver.

1.5 Methodology

This thesis has been developed at the Institute of Navigation and Communication of the German Aerospace Center (DLR) and the following phases have been undertaken in order to achieve the objectives:

1. **State of the art literature review:** This phase consisted in reading research papers and relevant books about Bayesian estimation, particle filters, multisensor fusion algorithms, GNSS and inertial sensors.

2. **Theoretical Derivation:** At this stage a Bayesian extension approach was modeled to represent the problem of fusing raw GNSS data, IMU data and the track map. Besides, a theoretical derivation of a Bayesian filter was required.

3. **Implementation:** This phase consisted on one side to provide the Java® RCAS framework with connection to real data of the sensors. On the other side, the filter approach must be implemented in the framework, as well as the required visualization addins to test the correct functioning of it.

4. **Evaluation:** The implemented algorithm must be tested with real data in different configurations and scenarios to prove the reliability of the proposed approach. A comparison with other approaches is also made suitable.

5. **Thesis Documentation:** This document have been generated using LaTeX where all the stages are captured and detailed.

1.6 Brief Overview: Structure of the Thesis

The rest of this thesis is organized in the following form: Chapter 2 introduces the reader in the field of study by providing some general state-of-the-art concepts and related work. In Chapter 3, the theoretical approach is fully developed. A theoretical study about the GNSS geometry problem in railway environments is also explained in Chapter 4. Then, Chapter 5 presents some key steps of the implementation of the proposed filtering solution and Chapter 6 describes the software framework integration and issues. The experimental context in which this thesis approach has been tested is introduced in Chapter 7 and the different evaluations and results obtained can be found in Chapter 8. Finally, in Chapter 9, the achievements and conclusions are summarized.

Additionally, three appendices have been included to cover some additional GNSS and implementation issues. In Appendix A and B, the reader can find some important aspects that were necessary for the use of the raw GNSS data: the satellite position and velocity computation, and the different correction models for the GNSS error sources. Second, in Appendix C we show some key parts of the Java® code that have been written for this thesis' localization approach.

Chapter 2

Theory

This chapter provides some general theoretical notions and key concepts that are essential for this thesis.

The following sections start by explaining the general concepts behind the operation of GNSS systems and technology. Then, state-of-the-art Bayesian theory is described to be able to face the localization estimation problem and filtering techniques.

Finally, some related work map-based localization in a multisensor environment are presented.

2.1 Global Navigation Satellite System (GNSS)

The term *Global Navigation Satellite System* (GNSS) encompass those positioning systems based on satellite technology. The fundamental method behind them is to measure the ranges between the receiver and those satellites that can be observed from the user position. By knowing the measured distance between the receiver and each satellite, as well as the precise position of the satellites in space, the user is able to determine his position by solving a geometrical problem (Figure 2.1).

GNSS systems consist of three different parts called *segments*: Space segment, control segment and user segment. The *space segment* comprises, firstly, the different subsystems inside the space vehicles (SV) (i.e., satellites), that generate the GNSS signal, maintain the communication with the control segment and keep the satellite stable in its orbit. Secondly, the constellation design must guarantee the presence of satellite signals all over the world and that the geometric observability of the user is sufficient [5]. This geometry is measured by the dilution of precision (DOP) parameter, which will be explain in detail in Section 2.1.4. The *control segment* is responsible for monitoring the health and status of the SVs, and maintaining the satellites proper functioning. Furthermore, the control segment updates the information contained in the *navigation message* that satellites broadcast to the users. This message includes the information of the *satellite's clock* bias, *ephemeris* (precise parameters to compute satellite position) and *almanac* (information about the position of every satellite in the constellation) [6]. The *user segment* is the receiver equipment, where

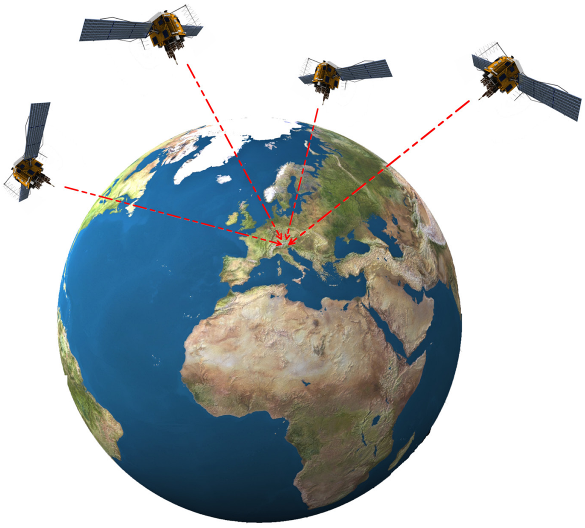


Figure 2.1: GNSS position determination

the signals are *acquired* and *tracked*. This segment also includes the further design and computation algorithms to obtain a *position, velocity and time* (PVT) solution [7].

Nowadays, there exist various satellite based navigation systems running or under development in our world. The most popular and the first operational system was the *Global Positioning System* (GPS), developed by several U.S. government organizations. GPS was designed to use *pseudorandom noise* (PRN) modulation for ranging with digital signals and under the two transmitted frequencies, L1 (1575.42 MHz) for civil/military use and L2 (1227.6 MHz) restricted for military use. Satellites broadcast ranging codes and navigation messages using Code Division Multiple Access (CDMA) technique [5].

Later, Russians declared operational GLONASS, its satellite navigation system. Although GLONASS is also based on PRN coding, they implemented Frequency Division Multiple Access (FDMA) where every satellite broadcast their signals in the L1 center frequency range (1598.0625-1605.373 MHz) and L2 (1242.9375-1248.625 MHz) [6].

The Indian IRNSS system, the Japanese QZSS or the Chinese BeiDou are other regional navigational satellite systems that complement the previous systems with additional satellites to improve the accuracy and robustness in their countries [6].

Finally, we mention GALILEO, the European approach under development, which is planned to provide a worldwide coverage and offer a number of important services. Among them, we point out the *Open Service* (OS) for standard mass-market applications and the *Safety of Life* (SoL) service, in which this thesis work can be settled in. GALILEO will provide signals in three different bands through CDMA transmission: E5 (1.164-1.215 MHz), E6 (1.260-1.300 MHz) and E2-L1-E1 (1.559-1.592 MHz) [6].

2.1.1 Fundamentals

In order to obtain the distance to each satellite, GNSS measures the time it takes for the signal transmitted by the satellite at a known absolute time, to reach the user receiver. This time is then multiplied by the speed of light to obtain the measured distance. This technique is known as *time of arrival* ranging (ToA).

The localization or navigation objective is to determine the position of the user at a certain time. One problem that comes up is the fact, that the time of transmission is set by the satellite atomic clock into the broadcast message, but the estimated time of reception is dependent of the receiver clock, which is not synchronized with the satellite clock and, moreover, has worse performance. Hence, the method used to achieve the synchronization includes the clock time difference as a new variable. This is the reason why in order to compute a 3D position of the user is necessary at least four visible satellites. In addition, the user will gain a precise absolute timing.

Besides, it is also possible to estimate the change rate of the distance by measuring the Doppler shift between the carrier nominal frequency and the received signal. The Doppler shift measurements combined with the knowledge of the satellite velocities can be used to determine the velocity of the user. As a result, we end up with the so called *position, velocity and time* (PVT) solution.

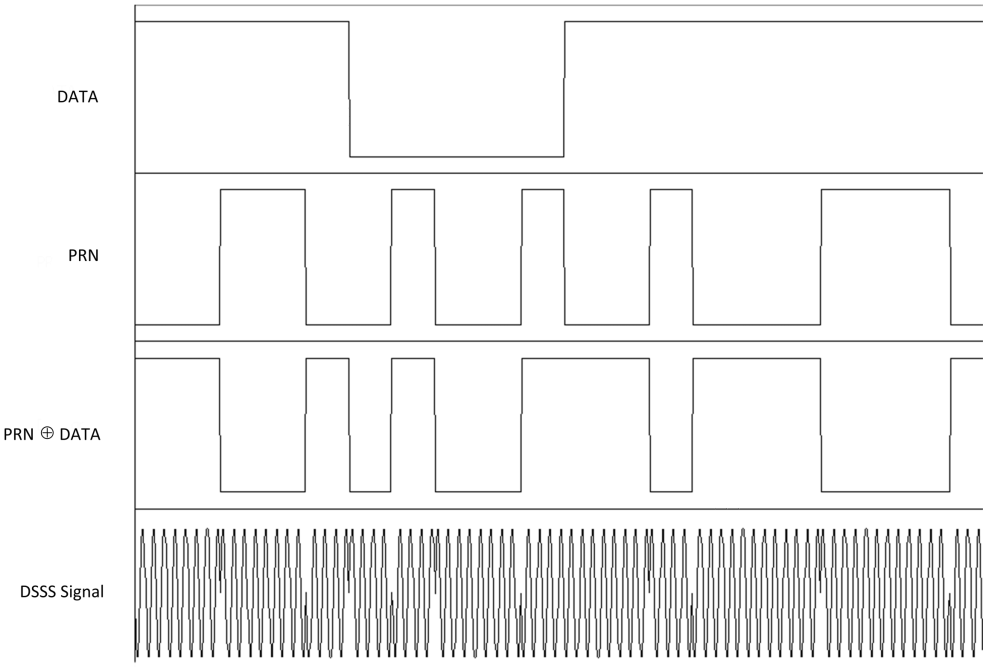


Figure 2.2: GPS signal generation

Signal Generation

Now we focus on the signal characteristics of GNSS systems. In particular, we explain the GPS signals because it is the system used for the testing of this thesis, and Galileo shares also these main aspects.

GPS utilizes two different *pseudorandom codes* transmitted in phase and in quadrature in the L1 band: The coarse/acquisition (C/A)-code, which is defined for open public service and the encrypted Precision (P)-code whose use is restricted. They are called pseudorandom because despite their deterministic nature, they have some noiselike properties (e.g., same number of zeros and ones). Basically, the generation steps of a GPS signal, shown in Figure 2.2, are the following: First, the digital data of the navigation message is *spread* (combined) with the PRN code. Then, the carrier signal is Binary Phase-Shift Keying (BPSK) modulated by the resulting signal. The final signal modulation is known as *direct sequence spread spectrum* (DSSS) signal [8].

Signal Acquisition and Tracking

The user, in order to compute the ToA must follow two steps: Firstly, the satellites that are visible must be identified, as well as the phase difference of the code and frequency of the received satellite signals. This process is known as *acquisition.* Once those values are known, it is only necessary to keep *tracking* the signals both in frequency and code phase.

The acquisition is performed by a double search in frequency and code of the satellite signals by comparing a local carrier signal and a local code replica for every satellite. As this process is out of the scope of this document, we invite the reader to consult Borre et al. [8] for more details.

The tracking purpose is to refine the coarse values of the code phase and frequency, and to follow their change over time. Therefore, the tracking needs to be carried out in two parts:

- Code tracking: This is normally achieved using a *delay lock loop* (DLL) where the peak of the correlation function between the received code and a local replica is tracked over time. The pseudorange measurement is later computed from this process.

- Frequency tracking: The tracking of the carrier signal is often implemented by a *phase lock loop* (PLL) though many receivers utilize a *frequency lock loop* (FLL). The Doppler shift measurement is extracted from this process.

2.1.2 Observables

Pseudorange Observation Model

The time of transmission is extracted from the navigation message and then compared with the local time to compute accurately the propagation time using the phase difference of the DLL. Then, by multiplying it by the speed of light we obtain the pseudorange measurement. This is the nearest measurement we can get of the true range satellite-user due to the non-synchronization of the clocks, the presence of errors in the satellite clock, and the different delays the signal may experience while traversing the atmosphere. A pseudorange measurement can be thus expressed, at a certain time t, as [5], [6]:

$$\rho = ||\mathbf{x}^s_{t-\tau} - \mathbf{x}_u|| - c \cdot dt^s_{t-\tau} + c \cdot b_{u,t} + dI_t + dT_t + \epsilon_{\rho,t}. \tag{2.1}$$

The equation (2.1) is known as the pseudorange general equation where ρ represents the pseudorange measurement and where:

$\|\|\mathbf{x}^s - \mathbf{x}_u\|\|$	True range satellite-user,
$\mathbf{x}^s$	Vector of satellite position in ECEF,
$\mathbf{x}_u$	Vector of user position in ECEF,
c	Speed of light,
dt^s	Satellite clock offset,
b_u	Clock bias of the receiver,
dI	Ionospheric delay in meters,
dT	Tropospheric delay in meters,
τ	Signal travel time,
ϵ_ρ	Remanent errors, (e.g. multipath).

The satellite position is computed using the ephemeris data transmitted by the navigation message. The satellite clock offset is corrected by some parameters which are also broadcasted. Regarding the ionospheric and tropospheric delays, there are some models to estimate them. The user position and the clock bias of the receiver are normally the unknowns in the localization problem. More information about the usual correction models used in this equation and throughout this thesis can be found in Appendix B.

Doppler Observation Model

Doppler frequency is the difference between the frequency nominal value of the local system and the actual received frequency. This difference is caused by the relative movement of the satellite to the user and therefore gives information about their relative velocities. As for the observable pseudorange, due to imprecisions and additional delays, the measured Doppler shift differs from the true one caused only by the relative movement. The Doppler shift observation can be written as [9]:

$$\lambda D = (\vec{n}^s_u + \frac{\vec{v}^s}{c}) \cdot [\vec{v}^s - \vec{v}_u] - d\dot{I} + d\dot{T} + cd_u - d\dot{t}^s + d\dot{R}_{ecc} + d\dot{R}_{Sagnac}, \tag{2.2}$$

with

λ	Wavelength of the observed signal (e.g. L1, L2),
D	Doppler measurement [Hz],
$\vec{n}_u^s$	LOS vector user-satellite,
$\frac{\vec{v}^s}{c}$	LOS correction,
$\vec{v}^s - \vec{v}_u$	ECEF speed difference between user and satellite,
c	Speed of light,
$d\dot{I}$	Ionospheric rate correction,
$d\dot{T}$	Tropospheric rate correction,
$d\dot{t}^s$	Satellite clock drift,
d_u	User clock drift,
$d\dot{R}_{ecc}$	Orbital eccentricity correction,
$d\dot{R}_{Sagnac}$	Sagnac rate correction.

2.1.3 Absolute Position and Timing

The classical and most common algorithm for static point position is based on the least-squares method since this methods is able to handle systems with more observations than variables [6]. Now, we describe the least-squares process to compute the user position from pseudorange measurements.

The pseudorange observation equation (2.1) can be expressed, for satellite k, in the following form:

$$\rho^k = \underbrace{\sqrt{(x^k - x_u)^2 + (y^k - y_u)^2 + (z^k - z_u)^2}}_{R_u^k} + c \cdot b_u + \epsilon, \tag{2.3}$$

where R_u^k is the geometrical range between the satellite and the receiver, and ϵ are the random and deterministic bias terms (i.e., satellite clock and atmospheric propagation).

Note that (2.3) is nonlinear with respect to the user position (x_u, y_u, z_u), so it is necessary to linearize it before computing the least-squares solution. The linearization is performed by a Taylor expansion of the nonlinear terms around an initial position estimate $(x_{u,0}, y_{u,0}, z_{u,0})$ [7]. The partial derivatives of Equation (2.3) are:

$$\frac{\partial R_{u,0}^k}{\partial x_{u,0}} = -\frac{x^k - x_{u,0}}{R_{u,0}^k}, \tag{2.4}$$

$$\frac{\partial R_{u,0}^k}{\partial y_{u,0}} = -\frac{y^k - y_{u,0}}{R_{u,0}^k}, \tag{2.5}$$

$$\frac{\partial R_{u,0}^k}{\partial z_{u,0}} = -\frac{z^k - z_{u,0}}{R_{u,0}^k}. \tag{2.6}$$

Hence, (2.3) can be written in terms of the increments $(\Delta x, \Delta y, \Delta z)$ between the approximate initial position and the user true position:

$$\rho^k = R^k_{u,0} - \frac{x^k - x_{u,0}}{R^k_{u,0}}\Delta x - \frac{y^k - y_{u,0}}{R^k_{u,0}}\Delta y - \frac{z^k - z_{u,0}}{R^k_{u,0}}\Delta z + c \cdot b_u + \epsilon. \tag{2.7}$$

Next, we can form the least-squares problem for the m visible satellites as:

$$\underbrace{\begin{bmatrix} -\frac{x^1-x_{u,0}}{R^1_{u,0}} & -\frac{y^1-y_{u,0}}{R^1_{u,0}} & -\frac{z^1-z_{u,0}}{R^1_{u,0}} & 1 \\ -\frac{x^2-x_{u,0}}{R^2_{u,0}} & -\frac{y^2-y_{u,0}}{R^2_{u,0}} & -\frac{z^2-z_{u,0}}{R^2_{u,0}} & 1 \\ -\frac{x^3-x_{u,0}}{R^3_{u,0}} & -\frac{y^3-y_{u,0}}{R^3_{u,0}} & -\frac{z^3-z_{u,0}}{R^3_{u,0}} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{x^m-x_{u,0}}{R^m_{u,0}} & -\frac{y^m-y_{u,0}}{R^m_{u,0}} & -\frac{z^m-z_{u,0}}{R^m_{u,0}} & 1 \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ c \cdot b_u \end{bmatrix}}_{\mathbf{\Delta x}} = \underbrace{\begin{bmatrix} \rho^1 - R^1_{u,0} \\ \rho^2 - R^2_{u,0} \\ \rho^3 - R^3_{u,0} \\ \cdots \\ \rho^k - R^m_{u,0} \end{bmatrix}}_{\mathbf{b}} - \boldsymbol{\epsilon}. \tag{2.8}$$

Assuming that we apply the corrections for the $\boldsymbol{\epsilon}$ term (Appendix B), the solution of the least-squares problem $\mathbf{G\Delta x} = \mathbf{b}$ is defined as the one that minimize the quadratic error of $\hat{\mathbf{e}} = \mathbf{b} - \mathbf{\Delta\hat{x}}$, and it can be expressed as [6]:

$$\mathbf{\Delta\hat{x}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{b}. \tag{2.9}$$

The least-squares algorithm applied for position determination in GNSS systems is therefore an iterative procedure that updates the reference initial position in the following form:

$$x_{u,i+1} = x_{u,i} + \Delta x_{u,i+1}, \tag{2.10}$$

$$y_{u,i+1} = y_{u,i} + \Delta y_{u,i+1}, \tag{2.11}$$

$$z_{u,i+1} = z_{u,i} + \Delta z_{u,i+1}, \tag{2.12}$$

and continues until the new increments are at meter level [8].

The classical least-squares algorithms assumes that the pseudorange errors are independent and identically distributed for all the observations, which is a common non-optimal approximation. However, it is possible to extend the algorithm so that we can include the different error noise component of every satellite. This approach is called *weighted least-squares* (WLS) and its solution is described by [6]:

$$\mathbf{\Delta\hat{x}} = (\mathbf{G}^T\mathbf{W}^{-1}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}^{-1}\mathbf{b}, \tag{2.13}$$

where the $\mathbf{W}$ matrix is the covariance matrix of the Gaussian pseudorange errors.

2.1.4 Dilution Of Precision

In GNSS localization, the geometry of satellites has an important effect in the precision of the estimated position. The concept of Dilution of Precision (DOP) is widely used when the position of satellites is evaluated respect to an aproximate user position. This DOP metric relates how the pseudorange errors convert into position errors due to the geometry of the problem. If σ_{UERE} is the standard deviation of user equivalent range error and σ_p is the standard deviation of resulting position error, they can be easy related by the dilution of precision [6]:

$$\sigma_p = DOP \cdot \sigma_{\text{UERE}}. \tag{2.14}$$

We can obtained the DOP for the positioning problem by means of the least-squares solution. In the previous section, we obtained the following solution expression:

$$\mathbf{\Delta \hat{x}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{b}. \tag{2.15}$$

It is shown in [7], that the covariance matrix of this solution error can be expressed as:

$$\text{Cov}\left[\mathbf{\Delta \hat{x}}\right] = \sigma^2 \left(\mathbf{G}^T\mathbf{G}\right)^{-1}, \tag{2.16}$$

where σ^2 is the variance error of the pseudorange measurement. Therefore, the error of an estimation has two factors, one representing the error of the measured range and another which is only dependent on the geometry of the satellites. The DOP matrix $\mathbf{H}$ is then defined as:

$$\mathbf{H} = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}. \tag{2.17}$$

Finally, from the diagonal values of the $\mathbf{H}$ matrix, we can obtain the different DOP values:

$$\text{Horizontal dilution of precion, } HDOP = \sqrt{H_{11} + H_{22}}, \tag{2.18}$$

$$\text{Position dilution of precision, } PDOP = \sqrt{H_{11} + H_{22} + H_{33}}, \tag{2.19}$$

$$\text{Time dilution of precision, } TDOP = \sqrt{H_{44}}, \tag{2.20}$$

$$\text{Geometric dilution of precision, } GDOP = \sqrt{H_{11} + H_{22} + H_{33} + H_{44}}. \tag{2.21}$$

2.2 Bayesian Estimation

Bayesian estimation refers to those probabilistic methods that aims at estimating the probability density function (*pdf*) of a random variable. For example, we may be interested in inferring a magnitude x regarding some data y coming from a sensor. The conditional probability $p(x \mid y)$ is called the *posterior probability distribution*, which can be expressed using *Bayes rule* as [10]:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\int p(y \mid x')p(x')dx'}. \tag{2.22}$$

The probability $p(x)$ represents our *prior belief* and the inverse probability $p(y \mid x)$ is usually called *likelihood*. Since y is assumed to be known, the probability density $p(y)$ is an *evidence* usually considered as a normalization factor. As a result, Bayes estimation concept can be sum up by the following formula:

$$posterior = \frac{likelihood \cdot prior}{evidence}. \tag{2.23}$$

2.2.1 Bayes Filter

A Bayes filter is a recursive Bayesian estimation, where an unknown probability density is estimated recursively over time including the new available information (i.e., from a sensor). We can represent this evolution very clearly through the use of a *dynamic Bayes network* (DBN), where the causal dependencies of the variables and their temporal evolution are shown in a directed acyclic graph (Figure 2.3).

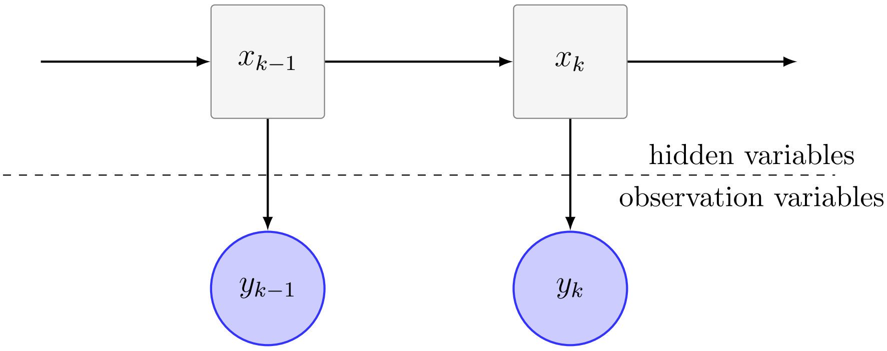


Figure 2.3: Dynamic Bayesian network

It should be taken into account that we assume an order one *Markov process* for the filter. This is based on the *Markov property*, where past and future states are independent if the current state is known.

The temporal evolution of Bayes filter consists of two important steps. Firstly, starting from the *prior belief* of x_{k-1}, and making use of a *process model*, we induce a transition to x_k. This is usually called the *prediction step*. The second step is called the *observation update*, where the new sensor data y_k is evaluated regarding our predictive belief of x_k. Both steps leads to the current *posterior belief* of x_k.

2.2.2 Kalman Filtering

Kalman filters (KF) are a family of Bayes filters where the variables are continuous, evolve linearly over time t and the probability distributions are all Gaussians [11]. Hence, every probability density is parametrized by its mean μ_t and its covariance Σ_t. Furthermore, the solution of this method is the *exact* posterior density and it is thus considered an *optimal* algorithm.

The Kalman filter algorithm includes the two Bayes filter steps: prediction and update.

1. **Prediction:** This step is summarized by the following equations [10]:

Algorithm 1 Kalman filter prediction

```
$\mu_{t|t-1} = F_t \mu_{t-1}$
$\Sigma_{t|t-1} = F_t \Sigma_{t-1} F_t^T + R_t$
```

where the matrix F_t establishes how the current states depend on the previous ones, and R_t is the covariance matrix that models the uncertainty introduced by the state transition.

2. **Update:** In this step, the new measurement is incorporated to the estimation. A crucial parameter of this process is the Kalman gain K_t, which defines the level in which the measurement is applied to the new state estimate. The algorithm of the measurement update of Kalman filter is summarized by [10]:

Algorithm 2 Kalman filter update

```
$K_t = \Sigma_{t|t-1} H_t^T (H_t \Sigma_{t|t-1} H_t^T + Q_t)^{-1}$
$\mu_t = \mu_{t|t-1} + K_t(z_t - H_t \mu_{t|t-1})$
$\Sigma_t = (I - K_t H_t) \Sigma_{t|t-1}$
```

where K_t is the Kalman gain, H_t express the relation between the measurement and the state estimate, z_t is the measurement, Q_t is the covariance matrix of the measurement noise and I is the identity matrix.

2.2.3 Particle Filtering

In many applied statistic problems, the assumption of linearity and Gaussianity do not hold and therefore, optimal methods are not suitable any more. In this section, we will present a suboptimal method, which is not subject to linear Gaussians and the continuity of the state-space, called *particle filters* (PF). Particle filters, also known as *Sequential Monte Carlo* (SMC) methods [12], are a non-parametric implementation of Bayes filters where the posterior density is approximated by a set of random samples called *particles* with its associated weights [13]:

$$p(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k}) \approx \left\{x_{0:k}^i, w_k^i\right\}_{i=1}^{N_p}, \tag{2.24}$$

where $x_{0:k}^i$ is the set of all particle states up to time k, w_k^i is the normalized weight and N_p is the number of samples.

Since it is impossible to sample directly from the posterior $p(\mathbf{x}_{0:k} \mid \mathbf{y}_{0:k})$, particles hypothesis are drawn independently from a normalized *importance density* function $\pi(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k})$ [14]. The *importance weights* $w_{0:k}$ are defined to be:

$$w_k^i \propto \frac{p(x_{0:k}^i \mid \mathbf{y}_{1:k})}{\pi(x_{0:k}^i \mid \mathbf{y}_{1:k})}. \tag{2.25}$$

It can be proved that the importance weights can be derived recursively for a filtering estimation as [13]:

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{y}_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{\pi(x_k^i \mid x_{k-1}^i, \mathbf{y}_{1:k})}, \tag{2.26}$$

Notice that from (2.26), we obtain an unnormalized weight for every particle and they must be thus normalized afterwards:

$$\bar{w}_k^i = \frac{w_k^i}{\sum_{j=1}^{N_p} w_k^j}. \tag{2.27}$$

The use of importance density and importance weights is known as Bayesian *Sequential Importance Sampling* (SIS) technique and facilitates the particles to fall in the region of the truth state value.

One problem of particle filters is the so called *degeneracy phenomenon*, where there will be only one particle with relevant weight after a few iterations. Clearly, a solution to reduce this problem would be to increase the number of particles which, nevertheless, would be computationally impractical. Hence, in order to mitigate this effect, particles filters focus on two important aspects: Firstly, the choice of the importance density plays a critical role and it should be thus defined carefully. Secondly, particle filters include also a selection step known as *resampling*, where the key idea is to eliminate particles which have a low importance weights, while multiplying particles with high importance weights.

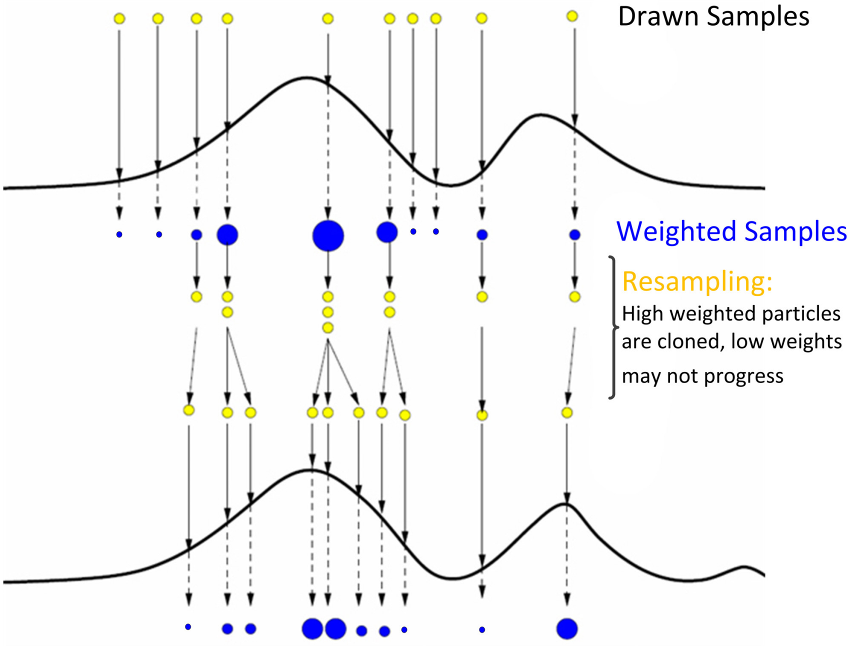

Drawn Samples
Weighted Samples
Resampling:
High weighted particles are cloned, low weights may not progress

Figure 2.4: Particle filter evolution
Adapted from van der Merwe et al. (2000) [15]

Indeed, this process replaces the current weighted sample set by a new one which follows more accurately the true posterior distribution. In Figure 2.4, we can see how this process is effectuated.

Since resampling should be executed every time a significant degeneracy is observed, we can use the *number of effective particles* N_{eff} as a suitable indicator of the degeneracy level and resample if a certain threshold N_T is exceed [13]:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_p} (w_k^i)^2}. \tag{2.28}$$

Finally, in Algorithm 3 we can see the general algorithm of a particle filter.

Algorithm 3 Particle filter

```
for $i = 1 : N_p$ do
    Draw $x_k^i \sim \pi(\mathbf{x}_k \mid x_{k-1}^i, \mathbf{y}_{1:k})$
    Compute importance weights $w_k^i$
end for
Compute normalization factor: $norm = \sum_{i=i}^{N_p} w_k^i$
for $i = 1 : N_p$ do
    Normalize weights: $w_k^i = \frac{w_k^i}{norm}$
end for
Compute $N_{eff}$
if $N_{eff} < N_T$ then
    Resample
end if
```

2.3 Multisensor Navigation

The purpose of a navigation system is to provide the user with a position, velocity and time solution (PVT), as well as in some cases some extra information about the user's attitude or orientation. Although GNSS systems are able to provide, in general, reasonable bounded accuracy, its performance may be degraded due to satellite signal outages caused by blockage or external interference, time to first fix (i.e., first position solution) or signal reacquisition capability.

In order to overcome the sensor's performance issues and improve reliability, it is usual to integrate two or more sensors in the same system. The objective is to choose sensors whose error sources are independent and therefore can be complemented by each other. The most popular integration techniques used in GNSS includes inertial sensors [16], though many other approaches also exists that use, for example, odometers, altimeters, speedometers or even a communication network.

Inertial navigation is usually integrated within GNSS by using Kalman filters, nevertheless, depending on the system requirements, other probabilistic techniques are being used nowadays. In the following sections, a brief introduction is presented about inertial sensors and the most common integration techniques and levels.

2.3.1 Inertial Navigation

Inertial navigation relies upon the measurement of acceleration, which can be integrated successively to provide estimates of changes in velocity and position. Inertial navigation systems (INS) obtain the new position by using the previous position and a relative vectorial information (i.e., velocity) over elapsed time. This is also known as *dead reckoning* navigation and its main disadvantage is the cumulative error that increases over time.

A system that navigates based only on inertial sensors is considered a *strapdown inertial navigation* and the main blocks involved in this system are shown in Figure 2.5 [17].

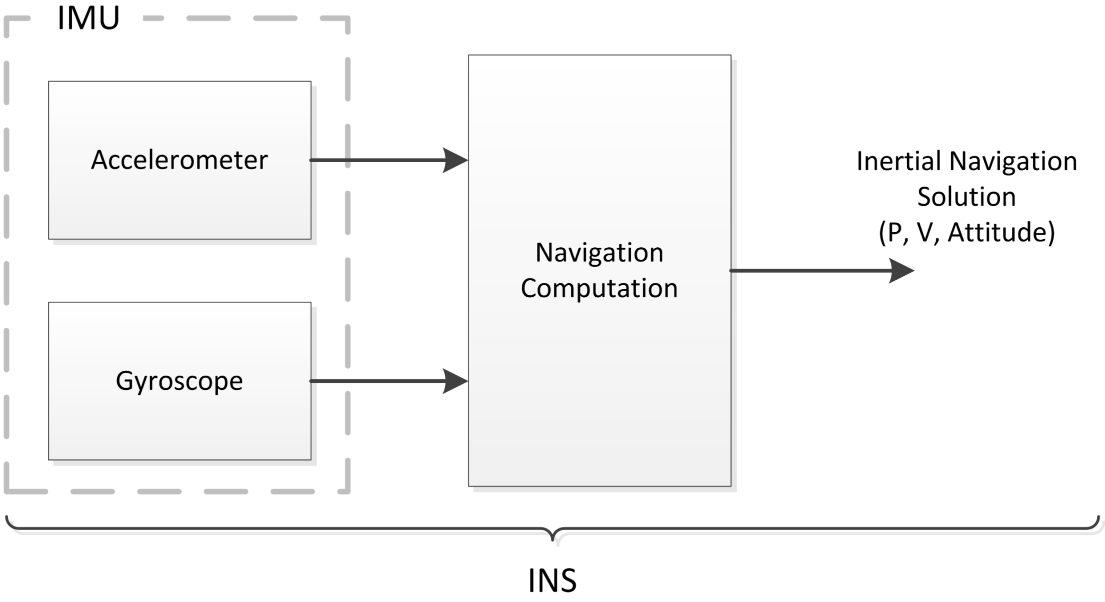


Figure 2.5: Strapdown inertial navigation scheme

From now on, we will focus uniquely on the *inertial measurement unit* (IMU) and the data output from its internal accelerometer and gyroscope.

2.3.2 Integration Levels

There exists different configuration schemes when integrating GNSS and an inertial measurement unit (IMU). They can be classified, depending on the level of integration, in three different configurations: *loosely coupled*, *tightly coupled* and *deeply coupled* [6], [16].

The loose integration, as shown in Figure 2.6, makes full use of GNSS receivers by integrating in the navigation processor (typically a Kalman filter) the PVT solution provided by the receiver and the IMU measurements (i.e., 3D accelerations and turn rates). Although this configuration requires less elaboration, its performance may be compromised, when only a few satellites are visible, by an inaccurate or even not computed PVT solution (with less than four visible satellites).

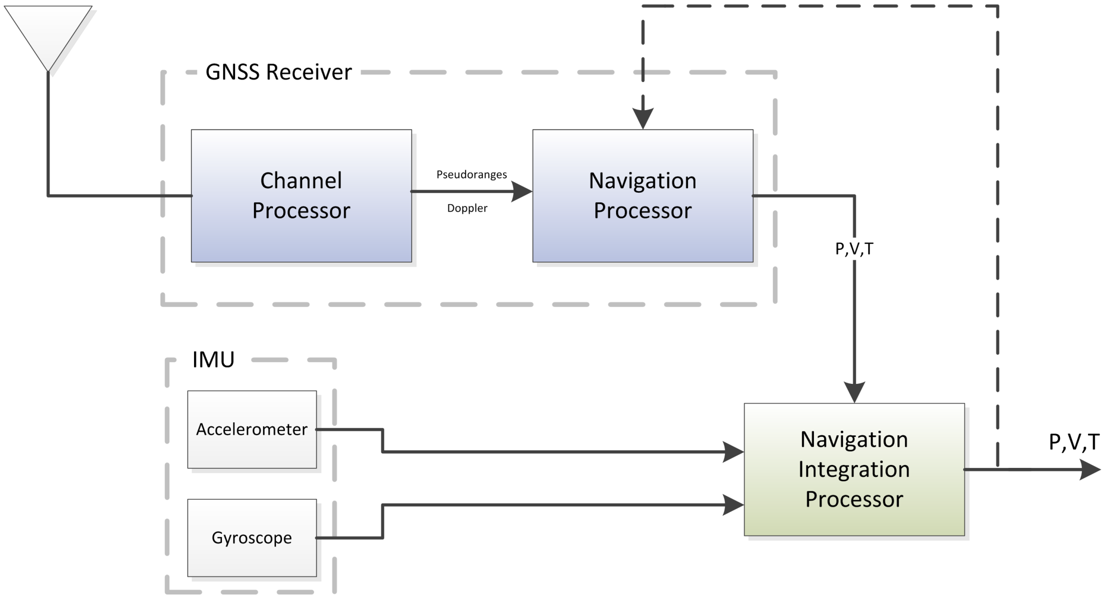


Figure 2.6: Loosely coupled integration scheme

The second level is the tight integration, shown in Figure 2.7. In this approach, instead of using the PVT solution of the receiver, the raw data is directly included in the navigation processor. It should be noted that, while an absolute position was computed in the loose integration previously to the final solution, in tight integration we will obtain an absolute position only after the GNSS/IMU integration. The use of raw data will additionally increase the complexity of the navigation processor since it is made necessary to handle the satellite ephemeris, the non-linearities of pseudorange and Doppler measurement models and to calculate raw data correction terms. Despite the increasing complexity, this approach is more robust because it can keep extracting useful GNSS information to aid the filter even in situations where less than four satellites are visible. In addition, some variants of this approach implement a backward path and make use of the solution to provide a tracking loop aiding to enhanced the correlator performance [16].

The deep integration is the third level configuration, where the IMU measurements are used in the navigation processor and in the receiver lower signal processing blocks (i.e., code and carrier generators, correlators). This approach typically implies to develop algorithms at hardware level and are therefore out of the scope of this work.

In the following, we will refer to a loosely coupled integration when the PVT solution of the GNSS is used in the filtering approach, as well as to a tightly coupled integration when pseudoranges and Doppler measurements are included directly in the filter.

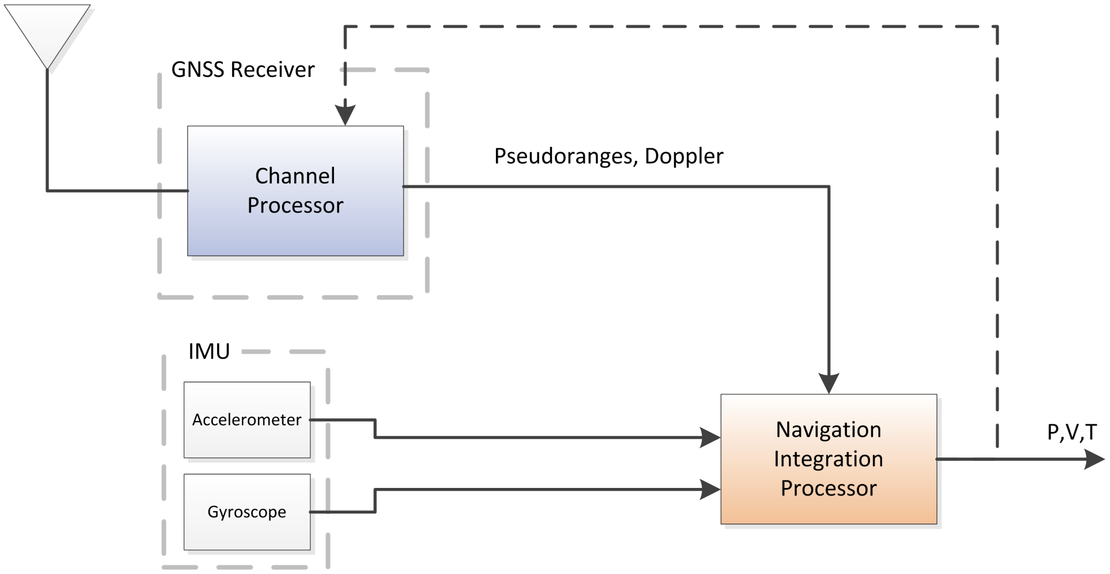


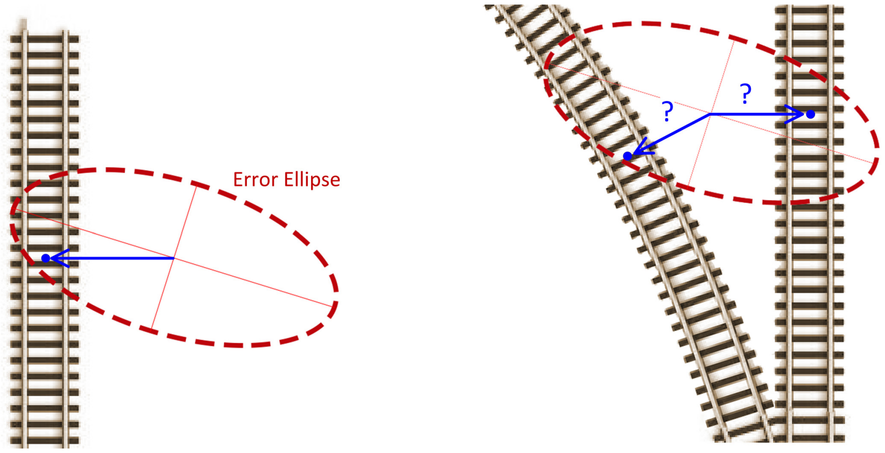
Figure 2.7: Tightly coupled integration scheme

2.4 Map Based Estimation Methods

Localization and navigation fusion techniques in a multisensor environment has been introduced in Section 2.3. However, when navigating along a railway map (or by similarity a road map) the state space where the vehicle could be, gets significantly reduced. The information of the map may be then intuitively used to simplify the estimation process or to increase the precision of the estimation. Despite of the fact that indeed the map gives important information, it is not obvious which is the best way of relating an estimation to a position on the map or which is the best way to include it into the estimation process. In fact, due to the discrete and nonlinear nature of the map, it is a highly non trivial problem.

The process of relating an estimated position to a position in a map is usually called *map-matching*. As road or rail maps are typically split into different segments, the naive approach consist of a geometry analysis, for example by selecting the segment that is the nearest in terms of euclidean distance. Other geometric approaches evaluates the error ellipse of the navigation system [18] (Figure 2.8a) or the curves of the map. Although these methods are a good approach in a single track situation, in a more complex network, specially in junctions or parallel segments, they may find problems to solve the correct position (Figure 2.8b). This can lead to an ambiguous solution and therefore, it is unacceptable for safety relevant applications of an ITS (Intelligent Transportation System).

Other algorithms are based on a topological analysis. This way, taking into account more information such as the vehicle speed, links connections or road classification, they can identify some of the junction situations [18]. A similar method that weights the possible tracks regarding real-world field data is presented in [19]. In this approach, despite the relative simple and fast algorithms, they can identify, in the best case, the 97 % of correct segments. A topological method in combination with an IMU is used in [20]. The turn



(a) Single track matching (b) Ambiguous matching after fork

Figure 2.8: Naive map-matching

rates are included in the process in a probabilistic fashion. Pointing out the suitability of IMU helping in switches.

Finally, advanced map-matching algorithms try to include the information of the map within the estimation process with more refined techniques. This inclusion can be done in different fashions and levels. Many techniques make use of fuzzy logic, Kalman filtering, probabilistic theory or belief theory [18].

We will focus for the rest of this chapter on those based on Bayesian estimation and, in particular, those that make inference by particle filtering [21], [22], [23], [24].

As every track or road map is a discrete number of segments and the movement over them is a highly nonlinear transition due to the connections, many approaches use Sequential Monte Carlo (SMC) methods for solving the integration between sensors and the map. In [21], the GPS solution, a yaw rate gyroscope and an odometer are merged with an Extended Kalman Filter, and then, the solution of the EKF is used as an observation in a particle filtering where the map is used as a constraint for the particles position.

In [22] an integration between GPS and an odometer is performed using the information of the map in the likelihood step of the filtering. The hypothesis (particles) are spread over all the space and then evaluated regarding a featured based and a topology based map. Although this approach could be suitable for cars, because the vehicles can be out of a road (in a parking area for example), for a train localization it would mean that we are sampling particles out of the map, which is actually impossible due to the highly constraint of the train to the railway tracks, and therefore a waste of resources.

In the context of train localization, a loosely-coupling multisensor approach where the information of the map is considered as a position constraint, has also been proposed in [23]. A particle filter is also used and the digital map provides additional feature information of the tracks (i.e., curvature). Of great importance is the consideration of an inertial unit as a complementary sensor for GNSS in railway navigation. This approach was extended in [25] to consider a 3D pose of the train on the tracks, so the map is able to provide the curvature, bank and slope for a specific point. Simulations were run showing encouraging results.

Another relevant contribution that must be mentioned is [24], where an estimation with a particle filter is done between raw GPS measurements and a road map. The estimation process is split in two steps. Firstly, a particle filter estimates the segment of road, and then a Kalman filter estimates the position along the segment for every hypothesis. The implementation is finally a Rao-Blackwellised PF. Results show the importance of Doppler measurements in junction situations and precisions of 11 meters were obtained.

This thesis will follow the approach in [23] [25], extending it by raw GNSS data (i.e., pseudoranges and Doppler). The extended method consisting of raw GNSS measurements, IMU data and a track-map is analyzed in different situations.

Chapter 3

Bayesian Train Localization

The main objective of the system developed in this master's thesis is to estimate the position of a train within a railway network. This position is thus defined in the topological domain by the track ID denoted by R, the position along the track s and the orientation of the train related to the track dir:

$$\mathcal{T}^{topo} = \{R, s, dir\}. \tag{3.1}$$

In this chapter, a theoretical probabilistic approach that faces this task is developed. This approach aims at providing a reliable, accurate and robust solution to the localization and navigation challenges in railway environments, such as track selection in junctions, parallel track situations and few visible satellites moments.

To do that, this approach extends theoretically [23] and [25] to a *tightly coupled integration*, where the raw GNSS data will be directly integrated with the information of an inertial sensor.

The theoretical development of the proposed approach, starts by presenting in a dynamic Bayesian network (DBN), the causal dependencies between the different variables involved. Then, it follows a detailed explanation of the transition models used to characterize the evolution of the different states. A Bayesian filter will be derived and a particle filter will be used to make inference out of the network. The theoretical implementation of the particle filter and the problem of the filter output will be finally presented.

3.1 Dynamic Bayesian Network

A Bayesian network is an great tool to model complex problems by showing the causal dependencies of random variables or states in a directed acyclic graph. On the other side, a dynamic Bayesian network (DBN) is a temporal Bayesian network, where the evolution over time of the states is also represented.

The DBN that has been designed in this work can be seen in Figure 3.1. The dynamic states D represent the motion variables of the displacement of the train along the track. The states denoted by T are the topological, geometrical and featured description of a track position and M is the map information, which is a digital database of the railway network. Z^{IMU}, Z^{PR} and Z^{DP} are the observable random variables of the system, which are the IMU (Inertial Measurement Unit), pseudorange and Doppler measurements respectively. Clock states are denoted by C, which must be considered as unknowns in the GNSS estimation process. Finally, E^{long} is the remaining long-term correlated errors of every pseudorange measurement.

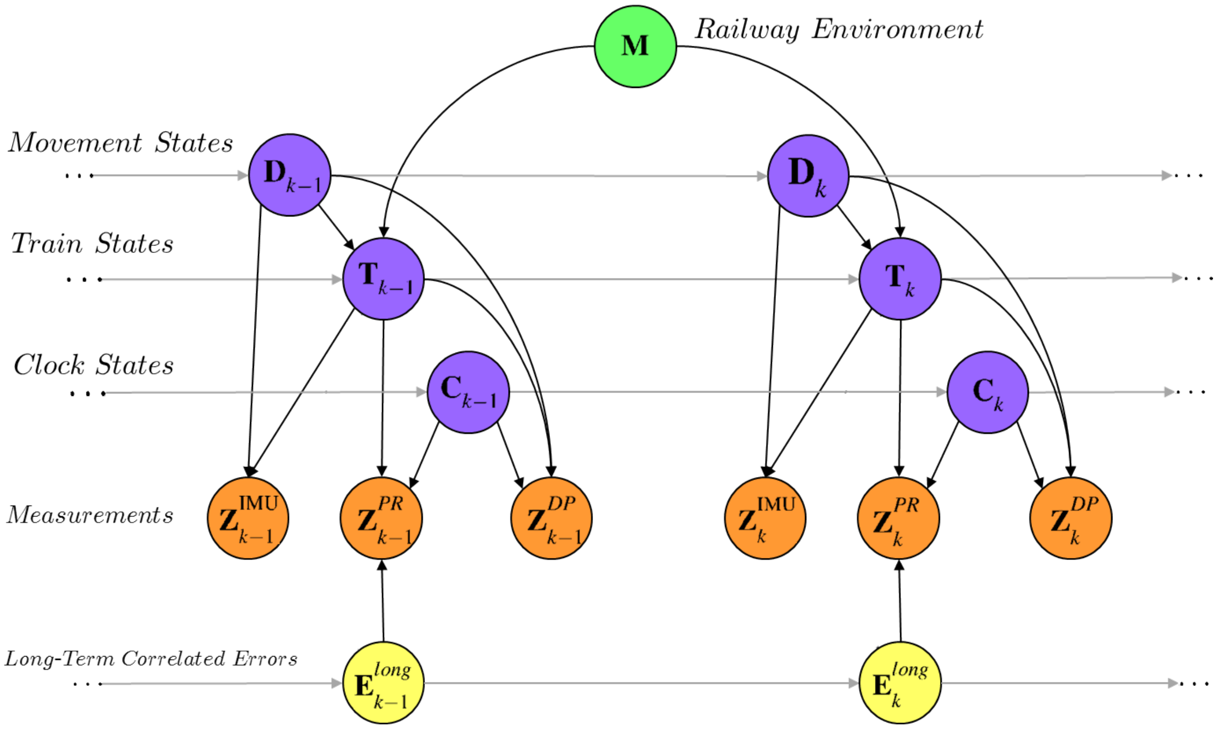


Figure 3.1: Dynamic Bayesian network of tight integration

3.2 Recursive Bayesian Inference

Bayesian inference derives the posterior probability as a consequence of two antecedents, a *prior probability* and a *likelihood function* derived from a probability model for the data to be observed. Therefore, the goal of the estimation process consist of deriving the *posterior* from the DBN of Figure 3.1. This posterior is expressed as:

$$p(D_{0:k}, T_{0:k}, C_{0:k}, E_{0:k} \mid Z^{IMU}_{1:k}, Z^{PR}_{1:k}, Z^{DP}_{1:k}, M), \tag{3.2}$$

or more compacted as:

$$p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k}, M). \tag{3.3}$$

where $\mathbf{Z}_{1:k} = \{Z^{IMU}, Z^{PR}, Z^{DP}\}_{1:k}$.

Next, the derivation of the Bayesian filter (Bayesian inference) is based on the DBN for the tightly coupled approach. First, Bayes rule is applied to the posterior:

$$p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \frac{p(\mathbf{Z}_k \mid \{DTCE\}_{0:k}, \mathbf{Z}_{1:k-1}, M) \cdot p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k-1}, M)}{p(\mathbf{Z}_k \mid \mathbf{Z}_{1:k-1}, M)}. \tag{3.4}$$

The denominator in equation (3.4) does not depend on the posterior factors $\{DTCE\}$ and therefore is considered as a normalization expression. Then, (3.4) simplifies in:

$$p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \eta \cdot p(\mathbf{Z}_k \mid \{DTCE\}_{0:k}, \mathbf{Z}_{1:k-1}, M) \cdot p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k-1}, M), \tag{3.5}$$

where η is the normalization coefficient. By making use of Markov assumption, where past and future data are independent if the current state is known, we can removed past dependencies from the first term of (3.5):

$$p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \eta \cdot p(\mathbf{Z}_k \mid \{DTCE\}_{k}, M) \cdot p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k-1}, M). \tag{3.6}$$

The next step consists on decoupling actual and past states from the last term of (3.6) by product rule:

$$\begin{aligned} p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \eta \cdot p(\mathbf{Z}_k \mid \{DTCE\}_{k}, M) \cdot \\ p(\{DTCE\}_{k} \mid \{DTCE\}_{0:k-1}, \mathbf{Z}_{1:k-1}, M) \cdot p(\{DTCE\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M). \end{aligned} \tag{3.7}$$

Finally, independent states are removed, as well as temporal independencies by applying again Markov condition:

$$\begin{aligned} p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \eta \cdot p(\mathbf{Z}_k \mid \{DTCE\}_{k}, M) \cdot \\ p(\{DTCE\}_{k} \mid \{DTCE\}_{k-1}, M) \cdot p(\{DTCE\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M). \end{aligned} \tag{3.8}$$

Note, that the last term of equation (3.8) is the posterior of the previous time step $k-1$. This expression is thus written in a recursive fashion that simplifies the estimation process in a temporal model and provides a *filtering* solution for every time step.

By looking at the posterior expression (3.8) we can easily identify the terms involved in the two steps of a Bayes filter. The last term is the previous posterior and therefore represents our *prior belief* of the states. Then, by taking into account the second term, a *prediction step* is considered. The first term represents a *measurement update*.

Now, both prediction and update terms are developed in independent factors by regarding the independencies extracted from the DBN. The prediction factor can be then written as:

$$\begin{aligned} p(\{DTCE\}_{k} \mid \{DTCE\}_{k-1}, M) =& p(D_k \mid D_{k-1}) \cdot p(T_k \mid T_{k-1}, D_k, M) \cdot \\ & \cdot p(C_k \mid C_{k-1}) \cdot p(E_k \mid E_{k-1}). \end{aligned} \tag{3.9}$$

The different factors in (3.9) are the transition models for the states of our estimation. The update term can also be split in independent factors for every observation of the system:

$$p(\mathbf{Z}_k \mid \{DTCE\}_k, M) = p(Z_k^{IMU} \mid D_k, T_k) \cdot p(Z_k^{PR} \mid T_k, C_k, E_k) \cdot p(Z_k^{DO} \mid T_k, D_k, C_k). \quad (3.10)$$

The factors in (3.10) stand for the measurement update or likelihoods of inertial, pseudorange and Doppler measurements respectively.

Finally, using (3.9) and (3.10) into (3.8), the factorized recursive expression of the posterior can be written as:

$$\begin{aligned} p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \ & \eta \cdot \underbrace{p(Z_k^{IMU} \mid D_k, T_k) \cdot p(Z_k^{PR} \mid T_k, C_k, E_k) \cdot p(Z_k^{DO} \mid T_k, D_k, C_k)}_{likelihoods} \cdot \\ & \underbrace{p(D_k \mid D_{k-1})}_{motion\ transition} \cdot \underbrace{p(T_k \mid T_{k-1}, D_k, M)}_{train\ transition} \cdot \\ & \underbrace{p(C_k \mid C_{k-1})}_{clock\ transition} \cdot \underbrace{p(E_k \mid E_{k-1})}_{error\ transition} \cdot \\ & \underbrace{p(\{DTCE\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M)}_{previous\ posterior} . \end{aligned} \quad (3.11)$$

3.3 States and Transition Models

In this section, a detailed view of the states that are involved in the estimation process will be offered. The understanding of the significance of every variable in this problem is essential to model a solution and provide a coherent approach. In addition, the different transition models that characterize the evolution of the states over time are proposed. These models must represent as accurately as possible the real evolution of the variables and therefore are a key step in the estimation.

3.3.1 Movement States Transition

The movement states D (Figure 3.2) represents the uni-dimensional motion of the train subject to the dynamics of the vehicle and the constraint of the railway tracks.

Movement States
$\mathbf{D}_{k-1}$
$\mathbf{D}_k$

Figure 3.2: Movement states

The train motion states can be written [23]:

$$\mathbf{D}_k = (\Delta s_k, \dot{s}_k, \ddot{s}_k)^T , \quad (3.12)$$

where Δs_k is the one-dimensional displacement of the train from last time step $k-1$ to present time k, $\dot{s}_k$ is the speed and $\ddot{s}_k$ is the acceleration along the track.

The transition of the motion of the train is characterized by an uni-dimensional constant acceleration model, that represents the limited changes of the train dynamics due to the inertia of the vehicle and construction parameters. The transition functions are summarized as:

$$\Delta s_k = \Delta t \cdot \dot{s}_{k-1} + \frac{\Delta t^2}{2} \cdot \ddot{s}_{k-1}, \tag{3.13}$$

$$\dot{s}_k = \dot{s}_{k-1} + \Delta t \cdot \ddot{s}_{k-1}, \tag{3.14}$$

$$\ddot{s}_k = \ddot{s}_{k-1} + \nu. \tag{3.15}$$

where Δt is the transition time step.

The process noise represented by ν is included in the acceleration term, and stands for a sample out of a zero mean Gaussian distribution:

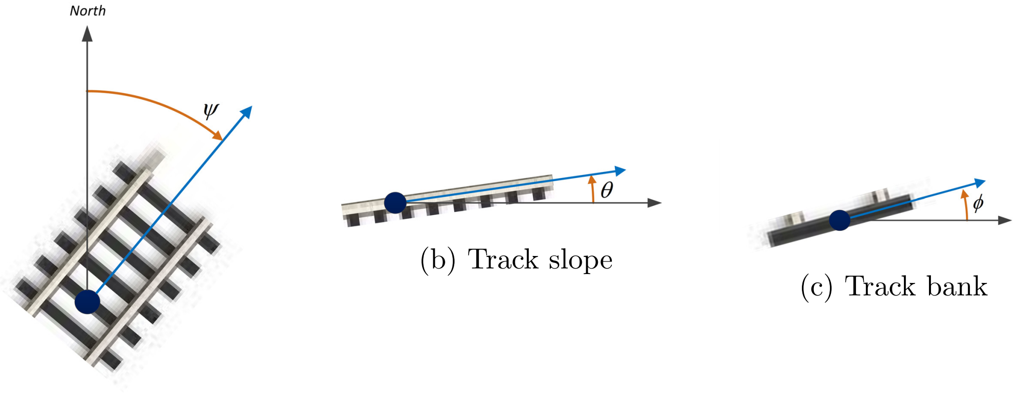
$$\nu \sim \mathcal{N}\left(0; \sigma_a\right), \tag{3.16}$$

where σ_a is a parameter of the filter and acts as the small variations of the train acceleration over time.

3.3.2 Train States Transition

The train state T_k covers a wide variety of data of a certain position in the railway track. This data is the topological position T^{topo}, the geographic position T^{geo}, the track attitude T^{att} and the train turn velocity T^{turn} [23], [25].

$$\mathbf{T}_k = \{T_k^{topo}, T_k^{geo}, T_k^{att}, T_k^{turn}\}. \tag{3.17}$$



(a) Track heading

(b) Track slope

(c) Track bank

Figure 3.3: Train attitude

As commented before, the topological position is $T^{topo} = \{R, s, dir\}$. The geographic position T^{geo} is an absolute position either in latitude, longitude, altitude or in ECEF coordinate system. The train attitude T^{att} is defined as the heading (ψ), slope (θ) and bank (ϕ) referred to the local frame (Figure 3.3), and the train turn velocities are the change rate of heading, slope and bank $T^{turn} = \{\dot{\psi}, \dot{\theta}, \dot{\phi}\}$.

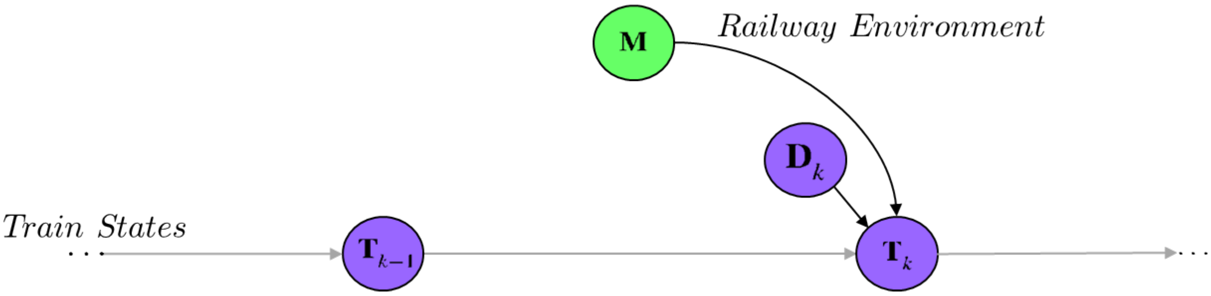


Figure 3.4: Train states transition

The transition of the train states is performed in the topological domain, where the new topological position is a map function of the previous position and the estimated movement (Figure 3.4):

$$T_k^{topo} = f_{map}\left(T_{k-1}^{topo}, \Delta s_k\right). \tag{3.18}$$

Once the topological position is estimated, thanks to the digital map, the train geometric position, attitude and turn rates can be obtained:

$$T_k^{geo} = f_{map}\left(T_k^{topo}\right), \tag{3.19}$$
$$T_k^{att} = f_{map}\left(T_k^{topo}\right), \tag{3.20}$$
$$T_k^{turn} = f_{map}\left(T_k^{topo}, \dot{s}_k\right). \tag{3.21}$$

3.3.3 Clock States

The estimation of the receiver clock variables is an important issue in the development of this work. As the raw data from the receiver is used directly, we have no information about the magnitude of the desynchronization between the user clock and the satellite clock. This difference is an essential parameter in the pseudorange and Doppler measurement equations (2.1) and (2.2). Therefore, the estimation of the clock states is critical for our filtering approach. The clock unknowns are bias and drift, which is the change of the clock bias over time:

$$\mathbf{C}_k = (b_k, d_k)^T. \tag{3.22}$$

Accordingly to [26], the transition of the clock at every time step can be modeled as a *two*

Clock States
$\mathbf{C}_{k-1}$
$\mathbf{C}_k$

Figure 3.5: Clock transition

states random walk process (Figure 3.5):

$$\begin{pmatrix} b_k \\ d_k \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} b_{k-1} \\ d_{k-1} \end{pmatrix} + \begin{pmatrix} w_p \\ w_f \end{pmatrix}. \tag{3.23}$$

In (3.23) the transition of the bias and drift is shown, where w_p and w_f are the phase and frequency noise of the internal oscillator respectively. Values for the noise model of $(w_p, w_f)^T$ can be obtained by sampling from a two dimensional process with $\mathbf{Q}$ covariance matrix (3.24):

$$\mathbf{Q} = \begin{bmatrix} S_f \Delta t + \frac{S_g \Delta t^3}{3} & \frac{S_g \Delta t^2}{2} \\ \frac{S_g \Delta t^2}{2} & S_g \Delta t \end{bmatrix}. \tag{3.24}$$

The covariance matrix of the noise distribution depends on the power spectral density of phase S_g and frequency S_f noise. An approximate solution and suitable values for these power densities can be found in [5].

Another important effect that must be considered in the transition model are the receiver clock resets. In order to prevent high values of clock bias, most receivers implement a reset of the bias value when a threshold is exceeded. If this reset is not considered in the estimator, the predicted pseudorange will be too far from the truth because, as seen in (2.1), pseudorange has a direct dependency of the clock bias.

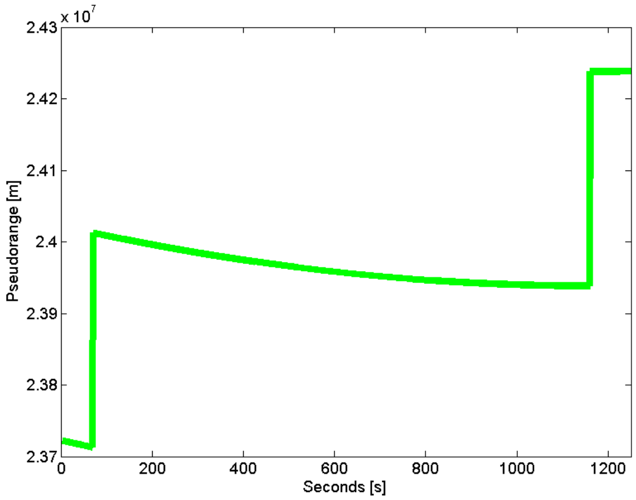


Figure 3.6: Effect of receiver clock reset in pseudorange measurement.
A clock bias reset of 1ms corresponds to a pseudorange jump of 300 km.

In Figure 3.6, we can see how the reset in the clock bias affect the pseudorange measurement. In fact, as we can see, the pseudorange obtained from the receiver will present these discontinuities.

The upper and lower thresholds for the clock bias is different depending the manufacturer and model of the receiver, and therefore is a parameter of the system.

Finally, the algorithm for the clock transition is presented in Algorithm 4.

Algorithm 4 Clock transition model

```
Input: $b_{k-1}, d_{k-1}, \Delta t$
if $b_{k-1} > upperThreshold$ then
    $b_{k-1} \leftarrow b_{k-1} - \tau$                ▷ $\tau$ is the clock reset jump
else if $b_{k-1} < lowerThreshold$ then
    $b_{k-1} \leftarrow b_{k-1} + \tau$
end if
$W \leftarrow sample2Dnormal(\mathbf{Q})$
$b_k \leftarrow b_{k-1} + \Delta t \cdot d_{k-1} + W[1]$
$d_k \leftarrow d_{k-1} + W[2]$
return $(b_k, d_k)$
```

3.3.4 Error Transition

One of the objectives of this work is to prove that the information and constraint of the track map in railway environments are a beneficial situation to estimate the remaining long-term correlated errors of pseudorange measurements. For that purpose, a simple random walk model is here proposed to predict the error evolution over time.

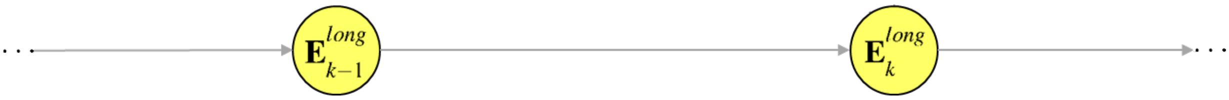


Figure 3.7: Error transition

The error transition, shown in Figure 3.7, will be just the previous time step state of the error plus a Gaussian process noise centered in zero (3.25).

$$\epsilon_k = \epsilon_{k-1} + \nu. \tag{3.25}$$

3.4 Rao-Blackwellised Filter Approach

As commented before, due to the utilization of the railway map, the estimation process presents a highly nonlinear nature. For that reason, we are restricted when choosing an implementation method. Kalman filters are not able to handle nonlinearities, so they can not be used for our purpose. Extended versions of Kalman filters use linearization of a model for a certain point. Nevertheless, they can not be used for discrete tracks. As a result, it is necessary a nonparametric approach, like a *sample-based method* [11]. Therefore, it is made obvious at this point that Sequential Monte Carlo (SMC) methods, also called particle filters (PF) and introduced in Section 2.2.3, are very well suited in this case.

In order to deduced a suitable solution for the implementation of the Bayesian filter, we first analyze the dimension of the state space, its restrictions and some clues about their statistical relations. The states considered are:

- Motion States (D): These variables are: the one-dimensional displacement over the tracks, the speed and the acceleration. These states are highly correlated over time due to the train inertia.

- Train States (T): Here it is necessary to handle the map with the discrete tracks. This is the main reason for choosing a particle filter.

- Clock States (C): They are bias and drift, which are highly correlated and represent the clock deviation of the receiver, which is common for all the satellites.

- Pseudorange Error (E^{long}): This represents the remaining long-term correlated error of every pseudorange. As every pseudorange measurement is considered as an independent observation, it is necessary to have a state for every visible satellite for this error, which moreover are totally uncorrelated because of the previous assumption. Hence, these states will increase the dimension and the complexity of the filter dramatically. In addition, as the number of visible satellites is changing over time and can reach up to twelve (for a single GPS constellation observation), it would be difficult to select the parameters of the filter without additional logic to follow the number of satellites.

Because of the latter reasons, a mixed solution for the particle filter has been chosen for this work to minimize the impact of the error estimation in terms of complexity.

The variables or states of the network are divided in different subsets. One of them is assigned to the particles and the other one is summarized by a closed-form representation of a distribution. This approach is called *Rao-Blackwellized Particle Filter* (RBPF) [11].

In this case, the error states subset will be separated from the rest and will be represented by a distribution that is estimated out of the particle filter. In particular, as a simple linear Gaussian model is planned to prove the beneficial situation of estimating the long-term correlated pseudorange errors, a Kalman filter will be used for this purpose. In

Figure 3.8, the implementation of the dynamic Bayesian network is shown again, pointing out the depth of the measurements and the states for all the satellites.

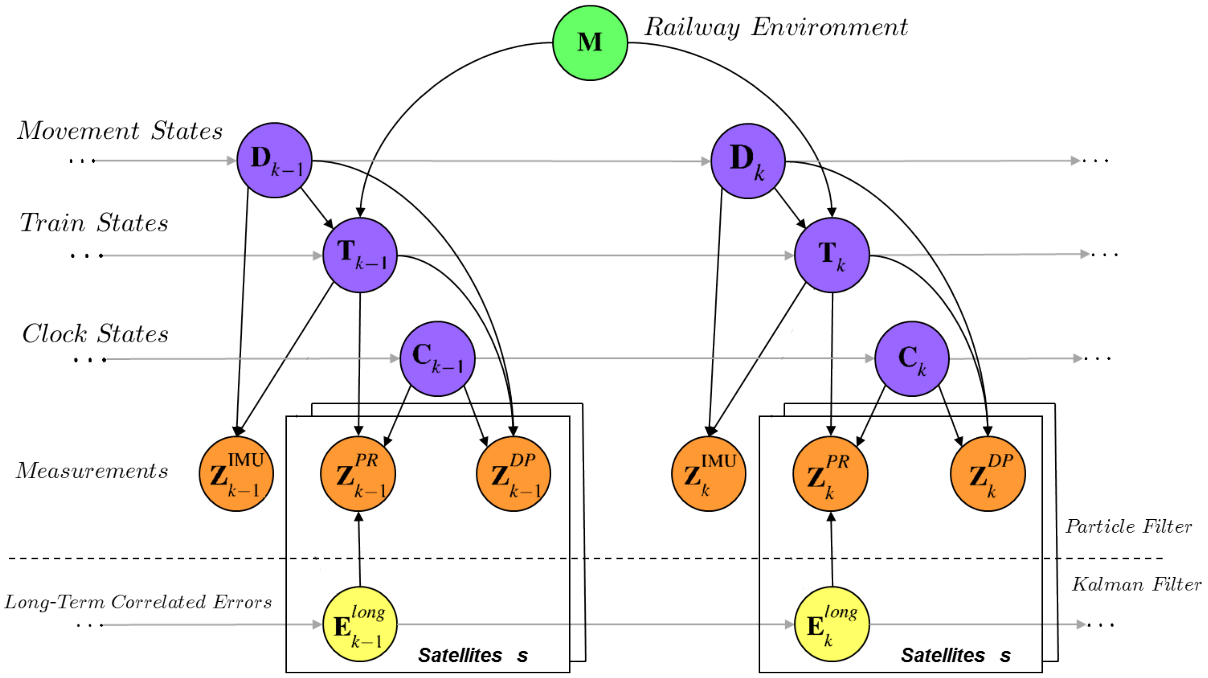


Figure 3.8: DBN implementation

One should note, that there will be as many pseudorange and Doppler measurements as the number of visible satellites s at time k. It will be also necessary to implement a Kalman filter for each of this pseudorange measurements.

The first step of the RBPF derivation is a decoupling of the states, that are going to be estimated outside the particle filter, from the posterior. That is, the states which are going to be *marginalized* out:

$$p(\{DTCE\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \underbrace{p(E_{0:k} \mid \mathbf{Z}_{1:k})}_{Optimal\ KF} \cdot \underbrace{p(\{DTC\}_{0:k} \mid \mathbf{Z}_{1:k}, M)}_{Particle\ Filter}. \tag{3.26}$$

The first term in (3.26) will be estimated by a Kalman Filter. That implies we consider this error as linear and Gaussian distributed. The second term, according to Section 3.2, is developed in a recursive expression:

$$\begin{aligned} p(\{DTC\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \eta \cdot p(\mathbf{Z}_k \mid \{DTCE\}_k, M)\cdot \\ \cdot p(\{DTC\}_k \mid \{DTC\}_{k-1}, M) \cdot p(\{DTC\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M), \end{aligned} \tag{3.27}$$

and more developed as:

$$\begin{aligned} p(\{DTC\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \ & \eta \cdot p(\mathbf{Z}_k \mid \{DTCE\}_k, M) \cdot p(D_k \mid D_{k-1}) \cdot \\ & p(T_k \mid T_{k-1}, D_k, M) \cdot p(C_k \mid C_{k-1}) \cdot \\ & p(\{DTC\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M). \end{aligned} \tag{3.28}$$

3.4.1 Particle Filter

The second term in (3.26) can be numerically approximated by a set of particles and its weights [13]:

$$p(\{DTC\}_{0:k} \mid \mathbf{Z}_{1:k}, M) \approx \{x^i_{0:k}, w^i_{0:k}\}_{i=1}^{N_p}. \tag{3.29}$$

As it was presented in Section 2.2.3, particles are samples obtained from an *importance density* distribution [14]:

$$x^i_{0:k} \sim \pi(\{DTC\}_{0:k} \mid \mathbf{Z}_{1:k}, M). \tag{3.30}$$

It is also possible to form the importance distribution recursively [12]:

$$\pi(\{DTC\}_{0:k} \mid \mathbf{Z}_{1:k}, M) = \underbrace{\pi(\{DTC\}_k \mid \{DTC\}_{0:k-1}, \mathbf{Z}_{1:k}, M)}_{proposal} \cdot \underbrace{\pi(\{DTC\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M)}_{recursion} \tag{3.31}$$

Therefore, it is possible to obtain particles by augmenting the previous set of particles to the new time step. If we sample $x^i_{0:k-1} \sim \pi(\{DTC\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M)$ then we can obtain $x^i_k \sim \pi(\{DTC\}_k \mid \{DTC\}_{0:k-1}, \mathbf{Z}_{1:k}, M)$.

For the next step, it is necessary to define a *proposal distribution* so that particles propagates near enough to the truth. The proposal is here defined as:

$$\pi(\{DTC\}_k \mid \{DTC\}_{0:k}, \mathbf{Z}_{1:k}, M) \triangleq p(D_k \mid D_{k-1}) \cdot p(T_k \mid T_{k-1}, D_k, M) \cdot p(C_k \mid C_{k-1}). \tag{3.32}$$

On the other hand, the weights for every particle at every time step are defined to be [13]:

$$w^i_k \propto \frac{p(\{DTC\}_{0:k} \mid \mathbf{Z}_{1:k}, M)}{\pi(\{DTC\}_{0:k} \mid \mathbf{Z}_{1:k}, M)}. \tag{3.33}$$

Putting equations (3.31) and (3.28) into (3.33), and using the expression in (3.32), we reach to:

$$\begin{aligned} w^i_k \propto \ p(\mathbf{Z}_k \mid \{DTCE\}_k, M) \cdot & \frac{p(D_k \mid D_{k-1}) \cdot p(T_k \mid T_{k-1}, D_k, M) \cdot p(C_k \mid C_{k-1})}{p(D_k \mid D_{k-1}) \cdot p(T_k \mid T_{k-1}, D_k, M) \cdot p(C_k \mid C_{k-1})} \cdot \\ & \underbrace{\frac{p(\{DTC\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M)}{\pi(\{DTC\}_{0:k-1} \mid \mathbf{Z}_{1:k-1}, M)}}_{w^i_{k-1}} . \end{aligned} \tag{3.34}$$

Note that the numerator and denominator of the first fraction term are the same and therefore can be simplified. Notice also that the last fraction term is the weight of the previous time step $k-1$.

Finally, as done for (3.10) it is also possible to split the first term of (3.34) for the different existing measurements, leading to the following reduced expression of the weights:

$$w_k^i \propto w_{k-1}^i \cdot \underbrace{p(Z_k^{IMU} \mid D_k, T_k)}_{IMU\ \ Likelihood} \cdot \underbrace{p(Z_k^{PR} \mid T_k, C_k, E_k)}_{pseudorange\ \ Likelihood} \cdot \underbrace{p(Z_k^{DO} \mid T_k, D_k, C_k)}_{Doppler\ \ Likelihood}. \tag{3.35}$$

3.4.2 Likelihoods

Sensor likelihoods are used to update the weights of every particle (3.35). The goal is to determine how likely are the estimated particle states respect to the new observation coming from the sensors. These likelihood functions are modeled by a Gaussian distribution and evaluated for every particle estimate [25]:

$$p(\mathbf{Z}_k \mid \{DTCE\}_k = \{DTCE\}_k^i, M) = \mathcal{N}\left(h(\{DTCE\}_k^i) \mid \mathbf{Z}_k, \Sigma_Z\right) \tag{3.36}$$

where $h(\cdot)$ is the measurement equation for every sensor regarding the i^{th} particle and Σ_Z is the covariance noise matrix of the sensor measurements. The different sensor likelihoods are detailed in the following.

IMU likelihood

The inertial measurement unit can provide data from the accelerometers and gyroscopes implemented inside. Accelerometers measure the acceleration in the three orthogonal axis $[a_x, a_y, a_z]$ and the gyroscope provides the turn rates of the previous fixed axis as $[w_x, w_y, w_z]$. Those magnitudes constitute a six degree of freedom (6DoF) system.

Although the 6DoF should be considered for a general purpose navigation, in the case of railway localization, due to the limited dynamics of the vehicle, some of them can be omitted.

In this work, only a three degree of freedom (3DoF) approach will be used. Firstly, the acceleration in X gives us direct information about the along-track train acceleration $\ddot{s}$. Second, the turn rate of Z will be a critical indicator when the vehicle changes its heading attitude after turning in a switch scenario. Finally, in curve situations and switches, due to the speed of the train, the vehicle will experience a lateral acceleration in Y because of the centripetal effect. The considered accelerations and turn velocity can be expressed by the following equations [25], [17]:

$$a_x = \ddot{s} + g\sin\theta, \tag{3.37}$$

$$a_y = -g\sin\phi\cos\theta + \dot{\psi}\dot{s}\cos\theta - \dot{\theta}\dot{s}\sin\phi, \tag{3.38}$$

$$w_z = \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi. \tag{3.39}$$

where g is the gravitational acceleration, ψ, θ and ϕ are the heading, slope and bank of the train attitude respectively and $\dot{\psi}$ and $\dot{\theta}$ are the turn velocities of heading and slope of the train. Besides, it is possible to express the turn velocities as a function of the track features and the speed of the vehicle:

$$\dot{\phi} = \frac{d\psi}{ds}\dot{s}, \tag{3.40}$$

$$\dot{\theta} = \frac{d\theta}{ds}\dot{s}, \tag{3.41}$$

where $\frac{d\psi}{ds}$ is the change of the heading over the track (curvature) and $\frac{d\theta}{ds}$ is the slope change. Note that the slope change of a railway track is insignificant, and therefore will be neglected for our purpose hereinafter.

As a result, the IMU likelihood can be split in the three different measurements considered here (i.e., a_x, a_y, w_z):

$$p(Z_k^{IMU} \mid \{D,T\}_k) = p(Z_k^{a_x} \mid \{D,T\}_k) p(Z_k^{a_y} \mid \{D,T\}_k) p(Z_k^{w_z} \mid \{D,T\}_k). \tag{3.42}$$

Finally, the computation of the IMU likelihoods given the estimated states of the i^{th} particle are evaluations of the following Gaussian distributions:

$$p(Z_k^{a_x} \mid \{D,T\}_k^i) = \mathcal{N}\left(\ddot{s}_k^i + g\sin\theta_k^i \mid Z_k^{a_x}, \sigma_{a_x}^2\right), \tag{3.43}$$

$$p(Z_k^{a_y} \mid \{D,T\}_k^i) = \mathcal{N}\left(-g\sin\phi_k^i\cos\theta_k^i + \dot{\psi}_k^i\dot{s}_k^i\cos\theta_k^i \mid Z_k^{a_y}, \sigma_{a_y}^2\right), \tag{3.44}$$

$$p(Z_k^{w_z} \mid \{D,T\}_k^i) = \mathcal{N}\left(\dot{\psi}_k^i\cos\theta_k^i\cos\phi_k^i \mid Z_k^{w_z}, \sigma_{w_z}^2\right), \tag{3.45}$$

where $\sigma_{a_x}^2, \sigma_{a_y}^2$ and $\sigma_{w_z}^2$ are the noise for the respective sensor measurements.

Pseudorange likelihood

Pseudorange likelihood evaluates every particle estimation with each satellite pseudorange measurement (Figure 3.9).

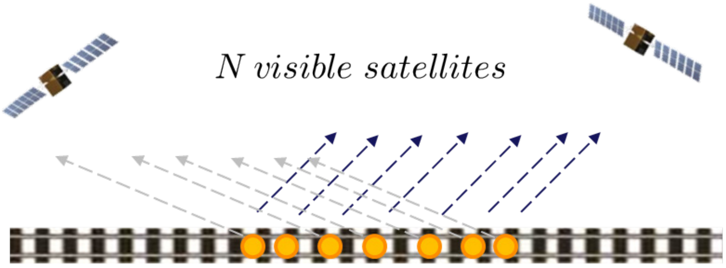


Figure 3.9: Estimated particles evaluation with N satellites

The pseudorange likelihood expression is thus :

$$p(Z_k^{PR} \mid T_k, C_k, E_k) = \prod_{j=1}^{S} p(z_{j,k}^{PR} \mid T_k, C_k, e_{j,k}), \tag{3.46}$$

where S is the number of visible satellites at time k. Particularizing (3.46) for the states of the i^{th} particle, we reach to:

$$p(Z_k^{PR} \mid \{TCE\}_k = \{TCE\}_k^i) = \prod_{j=1}^{S} p(z_{j,k}^{PR} \mid T_k^i, C_k^i, e_{j,k}^i). \tag{3.47}$$

The term $e_{j,k}^i$ in (3.47) is the long-term correlated error of pseudorange measurement for satellite j and particle i. It is in fact the error from the prediction step of the Kalman filter estimation. Notice that, there will be one Kalman filter for each particle and satellite.

The pseudorange likelihood for satellite j is then modeled as a Gaussian distribution with the pseudorange measurement as mean value $z_{j,k}^{PR}$ (Figure 3.10):

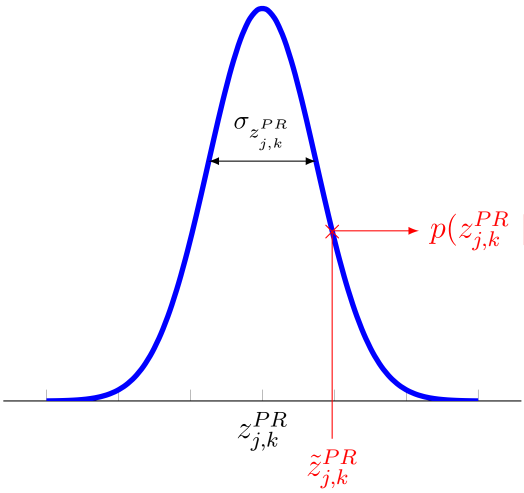


Figure 3.10: Pseudorange Gaussian likelihood with measured ($z_{j,k}^{PR}$) and predicted pseudorange ($\tilde{z}_{j,k}^{PR}$).

$$p(z_{j,k}^{PR} \mid T_k^i, C_k^i, e_{j,k}^i) = \mathcal{N}\left(\tilde{z}_{j,k}^{i,PR} \mid z_{j,k}^{PR}, \sigma^2_{z_{j,k}^{PR}}\right), \tag{3.48}$$

where $\tilde{z}_{j,k}^{i,PR}$ is the predicted pseudorange based on the i^{th} particle states and the Kalman filter error prediction $e_{j,k}^i$:

$$\tilde{z}_{j,k}^{i,PR} = \underbrace{|x_{j,k-\tau_i}^s - x_{p,k}^i| + c \cdot b_{p,k}^i - c \cdot \delta t_{j,k-\tau_i}^s + I_{j,k} + T_{j,k}}_{Particle\ based\ prediction} \underbrace{+\, e_{j,k}^i}_{KF\ prediction}. \tag{3.49}$$

Notice that (3.49) is based on the pseudorange general equation (2.1). A very important key aspect in (3.49) is that the travel time τ_i is also dependent on the estimated particle position $x^i_{p,k}$ and the estimated clock bias $b^i_{p,k}$. As a consequence, the estimation of the satellite position $x^s_{j,k-\tau_i}$ and the satellite clock correction $dt^s_{j,k-\tau_i}$ are related to the particle states as well. In Section 5.3, a detailed explanation of the algorithm implementation for the predicted pseudorange will be given.

On the other hand, the pseudorange measurement noise $\sigma^2_{z^{PR}_{j,k}}$ can be modeled following mainly two different approaches. Some authors propose an *elevation stochastic model* considering that the signals of satellites with low elevation experience more reflections and attenuation [27]. However, the accuracy of this approach is highly dependent of the environment and does not considered other sources of perturbation. Another group of approaches uses the Carrier to Noise (C/N0) indicator to model the sigma noise. C/N0 based approaches have widely shown better performance than the elevation-based ones [28]. In this case, the noise in the pseudorange measurement is modeled as the noise of the Delay Lock Loop (DLL) of the receiver, which is dependent on the C/N0 [7]:

$$\sigma^2_{z^{PR}_{j,k}} = c \cdot T_c \sqrt{\frac{d}{4T \cdot 10^{\frac{CN0_{j,k}}{10}}}}, \tag{3.50}$$

where T_c is the chip period of the C/A code (1ms), d is the correlator spacing, T is the integration time of the DLL and C/N0 the carrier to noise ratio in decibels. Note also that

$$B_{DLL} = \frac{1}{2T} \tag{3.51}$$

is the DLL bandwidth in Hertz, which is a specific parameter for receivers.

Doppler likelihood

Doppler likelihood follows the same approach as pseudorange likelihood. There will be a likelihood for every Doppler measurement of the visible satellites. This function follows a Gaussian distribution centered at the Doppler measurement:

$$p(Z^{DO}_k \mid \{TDC\}_k = \{TDC\}^i_k) = \prod_{j=1}^{S} p(z^{i,DO}_{j,k} \mid \{TDC\}^i_k) = \prod_{j=1}^{S} \mathcal{N}\left(\tilde{z}^{i,DO}_{j,k} \mid z^{DO}_{j,k}, \sigma^2_{z^{DO}_{j,k}}\right), \tag{3.52}$$

where $\tilde{z}^{i,DO}_{j,k}$ is the predicted Doppler shift measurement of satellite j at time k based on the i^{th} particle estimates and $\sigma^2_{z^{DO}_{j,k}}$ is the noise of the Doppler measurement.

The predicted Doppler is calculated by means of the general Doppler equation (2.2):

$$\tilde{z}^{i,DO}_{j,k} = \frac{1}{\lambda}\left[(\vec{n}^{s,i}_{p,j,k} + \frac{\vec{v}^s_{j,k}}{c}) \cdot [\vec{v}^s_{j,k} - \vec{v}^i_{p,k}] - d\dot{I}_{j,k} + d\dot{T}_{j,k} + cd^i_{p,k} - d\dot{t}^s_{j,k} + d\dot{R}_{ecc} + d\dot{R}^{Sagnac}_{j,k}\right]. \tag{3.53}$$

The algorithm implementation of this equation will be developed in Section 5.3.

Finally, the noise of the Doppler measurement will be modeled using the noise of the Phase Lock Loop (PLL) of the receiver, which is dependent of the C/N0:

$$\sigma_{z^{DO}_{j,k}} = \frac{\lambda}{2\pi}\sqrt{\frac{B_{PLL}}{10^{\frac{CN0_{j,k}}{10}}}\left(\frac{1}{2T\cdot 10^{\frac{CN0_{j,k}}{10}}}\right)}, \tag{3.54}$$

where λ is the wavelength of the L1 carrier signal in meters, B_{PLL} is the bandwidth of the PLL in Hertz and T is the integration time in seconds. Both the integration time and the bandwidth are parameters of the receiver configuration.

3.4.3 Error Kalman Filter

The long-term correlated error states are going to be estimated by an optimal KF. The objective is therefore to provide a closed form representation of the first term of equation 3.26. This factor can be more detailed because it only refers to the pseudorange error:

$$p(E_{0:k} \mid \mathbf{Z}_{1:k}) = p(E_{0:k} \mid Z^{PR}_{1:k}). \tag{3.55}$$

Notice that, we will have as many pseudorange measurements as the number of visible satellites at every epoch, so (3.55) is in fact:

$$p(E_{0:k} \mid Z^{PR}_{1:k}) = \prod_{j=1}^{S} p(e_{j,0:k} \mid z^{PR}_{j,1:k}), \tag{3.56}$$

where S is the number of satellites at time step k. Hence, there will be one Kalman filter to represent each of the pseudorange errors of each visible satellite. Each state is then represented by a normal distribution:

$$p(e_{j,0:k} \mid z^{PR}_{j,1:k}) \sim \mathcal{N}(\mu_{j,k}; \Sigma_{j,k}), \tag{3.57}$$

where μ is the mean and Σ the covariance matrix. As we estimate a one-dimensional state space (i.e., long-term correlated error) in the Kalman filter, those matrixes reduced to $\mu_{j,k} = e_{j,k}$ and $\Sigma_{j,k} = \sigma^{e^2}_{j,k}$.

Prediction

In order to prove that the long-term correlated error can be successfully estimated, a simple random walk model is used for the prediction step of the Kalman filter:

$$e_{j,k} = F \cdot e_{j,k-1} + \nu^{e}_{j,k}, \tag{3.58}$$

where we choose $F = 1$ and ν^e represents the small changes that this error experiences from one time step to the next (process noise).

The Kalman equations of the prediction step, presented in Algorithm 1, can be then simplified as:

$$e_{j,k|k-1} = e_{j,k-1}, \tag{3.59}$$

$$\sigma^{e^2}_{j,k|k-1} = \sigma^{e^2}_{j,k-1} + Q. \tag{3.60}$$

where Q is the variance of the process noise of the error.

Updating

The observation used for the Kalman filter is the difference between the pseudorange measurement $z^{PR}_{j,k}$ and a predicted pseudorange $\tilde{z}^{PR}_{j,k}$ based on the estimated particle states:

$$z^{KF}_{j,k} = z^{PR}_{j,k} - \tilde{z}^{PR}_{j,k}. \tag{3.61}$$

The measurement model is therefore represented by the following expression:

$$z^{KF}_{j,k} = H^T \cdot e_{j,k} + R_{j,k}, \tag{3.62}$$

where $H^T = 1$ and $R_{j,k} = \sigma^2_{DLL_{j,k}}$ is the measurement noise depending on the noise of the DLL tracking loop.

Accordingly, the equations for the Kalman filter update step are here [10]:

$$K_{j,k} = \Sigma_{j,k|k-1} H^T (H \Sigma_{j,k|k-1} H^T + R_{j,k})^{-1} = \frac{\sigma^{e^2}_{j,k|k-1}}{\sigma^{e^2}_{j,k|k-1} + \sigma^2_{DLL_{j,k}}}, \tag{3.63}$$

$$e_{j,k} = e_{j,k|k-1} + K_{j,k}(z^{KF}_{j,k} - e_{j,k|k-1}), \tag{3.64}$$

$$\sigma^{e^2}_{j,k} = (I - K_{j,k}) \sigma^{e^2}_{j,k|k-1}. \tag{3.65}$$

3.5 Particle Filter Output

The output of a particle filter is a probability density represented by a set of N_p particles with the state vector x and their weights w of the i^{th} particle for a time step k:

$$\mathcal{P}^{out}_k = \{x^i_k, w^i_k\}^{N_p}_{i=1}. \tag{3.66}$$

This output can be seen as a *cloud* of particles spread along our track map. Every particle represents a position hypothesis of the train states and, as resampling is performed quite often, the separation of the cloud can be seen as the uncertainty representation of the localization estimation. If the cloud of particles is wider along the tracks, then the uncertainty is higher. An example of the cloud of particles can be seen in Figure 3.11.

A desired output of a localization system for vehicle control or safety systems is the best position estimate since many location based services cannot process a position estimation represented by particles. Due to the track map constraint, we can analyze two different situations for the particle output cloud:

- **Single track:** All the particles are in the same track (Figure 3.11). This problem can be seen as a single 1D estimation problem.

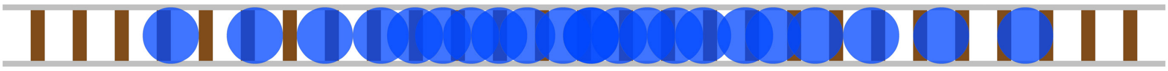


Figure 3.11: Cloud of particles in a single track

- **Multiple tracks:** The particles are spread in more than one track. This happen typically in a switch situation. In that scenario, different path hypotheses for the movement of the train are possible.

  For example, in Figure 3.12 we have two different hypothesis. The first hypothesis h_1 is that the train came from the track A and go on in the switch along the track B. The second hypothesis h_2 is coming from A and turning left in the switch along the track C. A path hypothesis is therefore an organized list of tracks IDs, in this example we end up with $h_1 = \{A, B\}$ and $h_2 = \{A, C\}$.

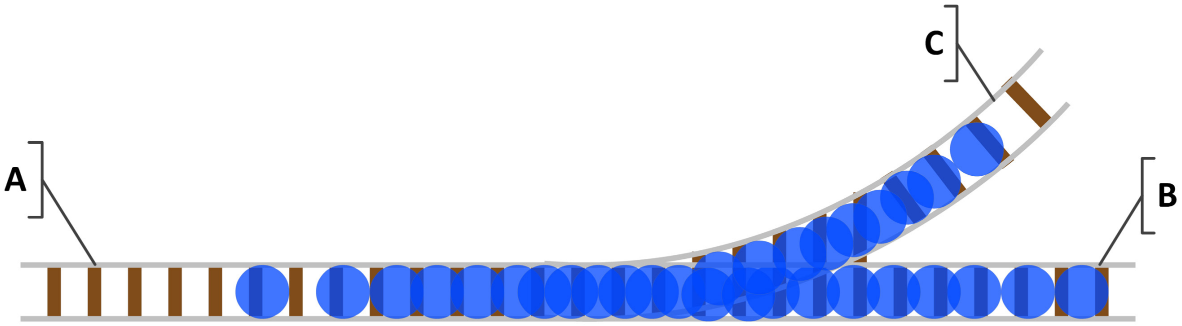


Figure 3.12: Cloud of particles in a switch situation

In order to obtain the best position estimate out of the cloud of particles, in particle filters we can follow four different approaches:

- Highest weighted particle,
- Most likely particle (ML),
- Maximum a posteriori (MAP),
- Minimum mean square error (MMSE).

3.5.1 Highest Weighted Particle

The filter output is the particle with the highest weight at every time step:

$$\widehat{x}_k = \underset{\{x_k^i\}_{i=0}^{N_p}}{\text{argmax}}\, w(x_k^i), \tag{3.67}$$

where $w(x_k^i)$ is the importance weight of particle x_k^i.

Although choosing the particle with the highest weight is a simple and good option for a general particle filter, if resampling is performed quite often, there will be then a lose of *history* of the particles and therefore this estimation will be highly dependent on the latter observations. The result will be thus a position estimation that changes too much at every time step.

3.5.2 Most Likely Particle (ML)

The ML criterion for a particle filter selects the particle that has the highest weight *over history.* This approach thus differs from the highest weighted particle by taking into account the cumulative weight over time:

$$\widehat{x}_k^{\text{ML}} = \underset{\{x_k^i\}_{i=0}^{N_p}}{\text{argmax}} \prod_{t=0}^{k} \bar{w}(x_t^i), \tag{3.68}$$

where $\bar{w}(x_t^i)$ is the normalized importance weight of particle i at time step t.

As the cumulative weight is the product of all the importance weights over history, it is necessary to handle this values in log space since the natural value would be too small to be used from a computational point of view.

3.5.3 Maximum a Posteriori (MAP)

The MAP estimation consists of the maximization of the posterior:

$$\widehat{x}_k^{\text{MAP}} = \underset{x_k}{\text{argmax}}\, p(x_k|\mathbf{Z}_k, \mathbf{M}), \tag{3.69}$$

where $p(x_k|\mathbf{Z}_k, \mathbf{M})$ can be expressed by Bayes rule as:

$$p(x_k|\mathbf{Z}_k, \mathbf{M}) = \frac{p(\mathbf{Z}_k|x_k, \mathbf{M}) \cdot p(x_k|\mathbf{Z}_{1:k-1}, \mathbf{M})}{p(\mathbf{Z}_k|\mathbf{Z}_{k-1})}. \tag{3.70}$$

The probability $p(\mathbf{Z}_k|\mathbf{Z}_{k-1})$ does not depend on x_k, so the maximization of the posterior is the same as the maximization of the numerator in (3.70)

$$\widehat{x}_k^{\text{MAP}} = \underset{x_k}{\text{argmax}}\, p(\mathbf{Z}_k|x_k, \mathbf{M}) \cdot p(x_k|\mathbf{Z}_{1:k-1}, \mathbf{M}). \tag{3.71}$$

The predictive density $p(x_k|\mathbf{Z}_{1:k-1}, \mathbf{M})$ can be expressed as:

$$p(x_k|\mathbf{Z}_{1:k-1}, \mathbf{M}) = \int p(x_k|x_{k-1}, \mathbf{M}) \cdot p(x_{k-1}|\mathbf{Z}_{k-1}, \mathbf{M}) dx_{k-1}. \tag{3.72}$$

This integral can not be analytically solved. However, for a particle based estimation we can approximate (3.72) by a sum since the second probability term in (3.72) is the posterior of the previous time step:

$$p(x_k|\mathbf{Z}_{1:k-1}, \mathbf{M}) \approx \sum_{j=1}^{N_p} p(x_k|x_{k-1}^j, \mathbf{M}) w_{k-1}^j. \tag{3.73}$$

Finally, the MAP output in a particle filter estimator is:

$$\widehat{x}_k^{\text{MAP}} = \underset{\{x_k^i\}_{i=0}^{N_p}}{\operatorname{argmax}}\, p(\mathbf{Z}_k|x_k^i, \mathbf{M}) \sum_{j=1}^{N_p} p(x_k^i|x_{k-1}^j, \mathbf{M}) w_{k-1}^j. \tag{3.74}$$

Note, that we use index 'i' for the particles in the actual time step 'k' and the index 'j' for the particles in the previous time step '$k-1$'. It should be also noticed that, since the complexity order of the sum term is $\mathrm{O}(N_p)$, the algorithm is $\mathrm{O}(N_p^2)$. This correspond to the evaluation of all the possible trajectories over time. For this reason, although this approach is promising for a multimodal situation, due to its complexity it has not been implemented in this work.

3.5.4 Minimum Mean Square Error (MMSE)

For a particle filter, as the posterior is approximated by a set of particles, we can express the particle that minimizes the mean square error (MSE) as:

$$\widehat{x}_k^{\text{MMSE}} = \underset{\{x_k^i\}_{i=0}^{N_p}}{\operatorname{argmin}}\, \|y_k - x_k^i\|^2. \tag{3.75}$$

And, in fact, this minimum mean square error sample is the result of the weighted mean of the particles:

$$\widehat{x}_k^{\text{MMSE}} = \sum_{i=0}^{N_p} \bar{w}_k^i \cdot x_k^i, \tag{3.76}$$

where $\bar{w}_k^i$ is the normalized weight of particle 'i' at time step 'k'.

$$\bar{w}_k^i = \frac{w_k^i}{\sum_{j=0}^{N_p} w_k^j}. \tag{3.77}$$

Equation (3.76) assumes that the state space of the particles is continuous. However, our objective is to estimate a topological position where the track ID 'R' and the direction 'dir' are discrete values. Therefore we can not use directly the expression in (3.76).

As the MMSE output can thus only be expressed in the topological domain, we are forced to design a modified mixed solution for the MMSE. This approach considers first the possible path hypothesis they train can take. Then, the position estimate will be the MMSE position inside the most likely path hypothesis. The steps to be followed are:

1. Create a set of Track IDs with its total weights.
2. Look recursively for the possible path hypothesis **H**.
3. ML criterion to choose the path.
4. Weighted mean of the position for the selected path.
5. ML decision for the direction.

Now we detail every step:

1. We create a set of all possible tracks where the train can be, that is, a set of the tracks where at least one particle is. We also get the total weight of every track as the sum of the weights of all particles in that track.

   $$\{R^j, W^j\}_{j=1}^{N_R}, \tag{3.78}$$

   where N_R is the number of different tracks that contains at least one particle and

   $$W^j = \sum_{i}^{N_p} w(x^i) \cdot \mathbf{1}\{R(x^i) = R^j\}. \tag{3.79}$$

   In Equation (3.79), $\mathbf{1}\{\}$ is the indicator function which is defined as:

   $$\mathbf{1}\{X\} = \left\{ \begin{array}{lll} 1 & if & X = true \\ 0 & if & X = false \end{array} \right. . \tag{3.80}$$

2. Once we have the set of possible tracks, we use the digital map to perform a recursive search to look for the possible path hypothesis the train can take. Every path hypothesis h is an organized list of tracks. We end with the group of different path hypotheses $\mathbf{H} = \{h_0, h_1, \ldots, h_n\}$.

3. The next step consists of choosing the most likely hypothesis. This can be easily defined as:

   $$h^* = \underset{h \in \mathbf{H}}{\text{argmax}} \sum_{j}^{N_R} W^j \cdot \mathbf{1}\{R_j \in h\}. \tag{3.81}$$

4. At this point, we will compute the weighted mean position along the most likely path hypothesis. Since the particle is defined in the map domain, we need first to transform the particle states to the path hypothesis domain. This process can be represented by a function g:

   $$x^i_{k,h} = g(x^i_k), \tag{3.82}$$

   where $x^i_{k,h}$ is the particle in the path hypothesis domain.

   Using (3.82) we can compute the weighted mean particle in the path hypothesis domain as:

   $$\hat{x}^{\text{MMSE}}_{k,h} = \sum_{i,x\in h^*} g(x^i_k) \cdot \bar{w}^i. \tag{3.83}$$

   The direction can not be computed as a weighted mean like in (3.83), and it will be just chosen with a ML criterion.

   Finally, we need to translate back the MMSE particle in the hypothesis domain to the track map domain by means of the inverse function:

   $$\widehat{x}^{\text{MMSE}}_k = g^{-1}\left(\hat{x}^{\text{MMSE}}_{k,h}\right). \tag{3.84}$$

Chapter 4

Satellite Geometry Considerations

The satellite-user geometry have an important impact on the accuracy of the PVT estimation in GNSS localization. As it was presented in Section 2.1.4, the metric used for measuring this impact is dilution of precision (DOP), which links the pseudorange errors to the final PVT error.

In this chapter, the geometry issue in railway localization is analyzed by extending the DOP for the train environment. To that purpose, we first need to represent the satellite positions in the local frame of the track, which will be developed in Section 4.1. Then, after the derivation of DOP when more than four satellites are in view, we will use the constraint of the map to make some assumptions and reduce the position dimension. This allows us to extend the DOP concept when less than four satellites are visible. Finally, new metrics are being developed to help in comparing different group of satellites in challenging visibility situations.

4.1 Satellite Position in Track Frame

Satellites positions are typically computed in *Earth-Centered, Earth-Fixed* (ECEF) coordinate system. So our first task is to transform the GNSS ECEF coordinate system to the railway track reference frame, which is characterized by the orthogonal *along-track, cross-track and vertical* axes.

The process to transform an ECEF position to a track frame coordinate referred to an approximate train position is done in two steps:

1. First, we transform an ECEF coordinate to a local Cartesian *North, East, Down* (NED) system by applying the following rotation transformation [7]:

$$C_{\text{ECEF}}^{\text{NED}} = \begin{pmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\cos\lambda\sin\varphi & -\sin\lambda\sin\varphi & \cos\varphi \\ \cos\lambda\cos\varphi & \sin\lambda\cos\varphi & \sin\varphi \end{pmatrix}, \tag{4.1}$$

where λ is the latitude of the reference position and φ the longitude.

2. The transformation from NED to the track frame is defined taking into account the attitude of the track. This attitude is expressed as the *slope* (θ) being the pitch angle, the *bank* (ϕ) as roll and *heading* (ψ) representing the yaw angle. Then, the conversion from NED to the track frame is performed by the following rotation matrix [17]:

$$C_{\text{NED}}^{\text{track}} = \begin{pmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{pmatrix}. \tag{4.2}$$

As a result, we can transform a ECEF coordinate to the track frame by multiplying by the two previous rotation matrix:

$$\begin{bmatrix} \widehat{a} \\ \widehat{c} \\ \widehat{d} \end{bmatrix} = C_{\text{NED}}^{\text{track}} C_{\text{ECEF}}^{\text{NED}} \begin{bmatrix} \widehat{x} \\ \widehat{y} \\ \widehat{z} \end{bmatrix}, \tag{4.3}$$

where $[\widehat{x}, \widehat{y}, \widehat{z}]$ are the coordinates in ECEF system and $[\ \widehat{a}, \widehat{c}, \widehat{d}\ ]$ are the along-track, cross-track and down coordinates in the track frame reference system.

Another common and useful representation of the satellite positions is based on elevation and azimuth. Now, the derivation of a unit vector direction of cosines in the train frame is also included.

The first step is to define the Line Of Sight (LOS) unit vector that points to the satellite from an approximate position:

$$\widehat{LOS} = \frac{x^s - x_u}{\|x^s - x_u\|}. \tag{4.4}$$

where x^s is the satellite position in ECEF coordinate system and x_u is the approximate position of the user in ECEF as well.

Then, it is possible to compute the elevation Φ of the satellite in the track frame and the azimuth Λ respect to the track as (Figure 4.1):

$$\Phi = \arcsin(\widehat{LOS} \cdot (-\widehat{d}\ )), \tag{4.5}$$

$$\Lambda = \arctan\left(\frac{\widehat{LOS} \cdot \widehat{c}}{\widehat{LOS} \cdot \widehat{a}}\right). \tag{4.6}$$

In practice, if we already have the elevation (el) and azimuth (az) of the satellite in the local frame, Φ and Λ can be easily computed, taking into account the slope and heading of the track, as:

$$\begin{aligned} \Phi &= el - \theta, \\ \Lambda &= az - \psi. \end{aligned} \tag{4.7}$$

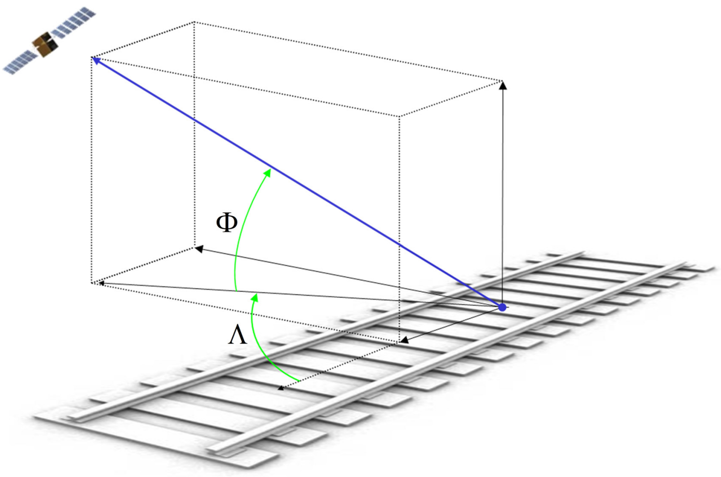


Figure 4.1: Satellite position in track frame

We can therefore define the unit vector direction of cosines in train frame for every satellite k as:

$$\mathbf{1}^{(k)} = \left(\begin{array}{ccc} \cos\Phi^{(k)}\cos\Lambda^{(k)} & \cos\Phi^{(k)}\sin\Lambda^{(k)} & \sin\Phi^{(k)} \end{array} \right)^T . \tag{4.8}$$

For simplicity, we will name in the rest of the chapter these components as:

$$\mathbf{1}^{(k)} = \left(\begin{array}{ccc} \mathbf{1}_L^{(k)} & \mathbf{1}_C^{(k)} & \mathbf{1}_V^{(k)} \end{array} \right)^T , \tag{4.9}$$

where each element stands for the along-track, cross-track and vertical components respectively.

4.2 Track Dilution Of Precision

When more than 3 satellite are received we can form the geometric matrix $\mathbf{G}$ that characterize all the visible satellite positions in a closed form:

$$\mathbf{G} = \begin{bmatrix} \mathbf{1}_L^{(0)} & \mathbf{1}_C^{(0)} & \mathbf{1}_V^{(0)} & 1 \\ \mathbf{1}_L^{(1)} & \mathbf{1}_C^{(1)} & \mathbf{1}_V^{(1)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{1}_L^{(k)} & \mathbf{1}_C^{(k)} & \mathbf{1}_V^{(k)} & 1 \end{bmatrix} , \tag{4.10}$$

where every row contains the direction vector of cosines of every satellite in the three first columns and a '1' in the last column for the user time offset.

Following the general derivation of dilution of precision commented in Section 2.1.4, we compute the DOP matrix H as follows:

$$\mathbf{H} = (\mathbf{G}^T\mathbf{G})^{-1} = \begin{bmatrix} LDOP^2 & \bullet & \bullet & \bullet \\ \bullet & CDOP^2 & \bullet & \bullet \\ \bullet & \bullet & VDOP^2 & \bullet \\ \bullet & \bullet & \bullet & TDOP^2 \end{bmatrix}. \tag{4.11}$$

Since all the vectors were in the track frame reference system, we obtain the along-track DOP (LDOP), the cross-track DOP (CDOP), the vertical DOP (VDOP) and the time DOP (TDOP). For example, the LDOP value gives information about how the pseudorange noise uncertainty translates into a along-track position uncertainty.

Note that from these coefficients it is also possible to get the other general DOP values:

$$\text{Horizontal Dilution of Precision: } HDOP^2 = LDOP^2 + CDOP^2, \tag{4.12}$$

$$\text{Position Dilution of Precision: } PDOP^2 = HDOP^2 + VDOP^2, \tag{4.13}$$

$$\text{Geometric Dilution of Precision: } GDOP^2 = PDOP^2 + TDOP^2. \tag{4.14}$$

4.3 Dilution Of Precision with Few Satellites

In applications where the user is normally fixed to the surface of the earth (e.g., cars, pedestrians or trains), there is no ambiguity about the vertical position, because for every horizontal point there is only one possible altitude. In this situation, we can obtain the altitude by using any ellipsoid model of the Earth's surface, such as in the World Geodetic System (WGS) or the GEOID03 [6]. As a result, the vertical uncertainty can be neglected and we can focus our estimation on the horizontal components (Figure 4.2) and time. Hence, the geometric matrix **G** can be reduced to three unknowns [29], [5].

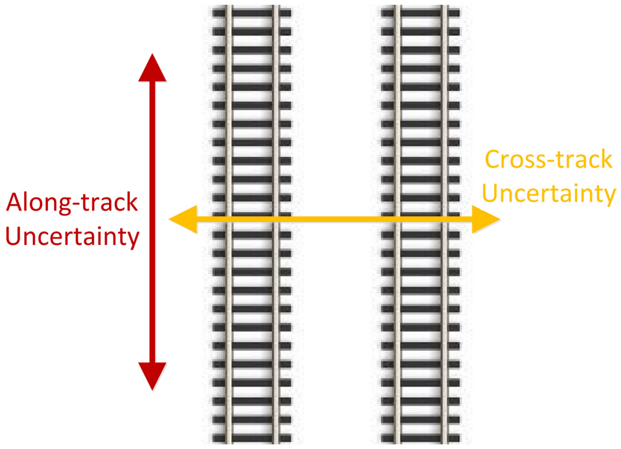


Figure 4.2: Track position uncertainty

Following this latter assumption and some others for the special case of railway localization, the geometric matrix can be reduced, and therefore it is possible to derive the DOP matrix with less than four satellites.

4.3.1 Three Visible Satellites: Known Altitude

As the slope of a train is specified to be less than 1%, the altitude along the track will not change significantly in the vicinity of an approximate position. Hence, we can remove the altitude uncertainty and the geometry matrix **G** is reduced to:

$$\mathbf{G} = \begin{bmatrix} \mathbf{1}_L^{(0)} & \mathbf{1}_C^{(0)} & 1 \\ \mathbf{1}_L^{(1)} & \mathbf{1}_C^{(1)} & 1 \\ \mathbf{1}_L^{(2)} & \mathbf{1}_C^{(2)} & 1 \end{bmatrix}. \tag{4.15}$$

The derivation of the DOP matrix can be thus achieved with only three satellites in view:

$$\mathbf{H} = \begin{bmatrix} LDOP_{_3}^2 & \bullet & \bullet \\ \bullet & CDOP_{_3}^2 & \bullet \\ \bullet & \bullet & TDOP_{_3}^2 \end{bmatrix}, \tag{4.16}$$

where the subscript '3' is added to denote that only three satellites have been used for the calculations. It is also relevant that the range of acceptable values using only three satellites would be different from the general approach.

4.3.2 Two Satellites: Known Altitude and Clock Variation

With the evolution of clock technologies, atomic clocks are becoming more accessible. As a consequence, we could think of a receiver that uses as a base time an external atomic clock. In this case, it would be reasonable to assume for a certain time that there exist no clock mismatching. Hence, the clock term could be also removed from the computation of the geometric and DOP matrix and we could still derive it with only two visible satellites:

$$\mathbf{G} = \begin{bmatrix} \mathbf{1}_L^{(0)} & \mathbf{1}_C^{(0)} \\ \mathbf{1}_L^{(1)} & \mathbf{1}_C^{(1)} \end{bmatrix} \Longrightarrow \mathbf{H} = \begin{bmatrix} LDOP_{_2}^2 & \bullet \\ \bullet & CDOP_{_2}^2 \end{bmatrix}. \tag{4.17}$$

4.3.3 Two Satellites: Railway Assumptions

In the railway context, as the train can only be on the tracks, the cross-track uncertainty is then reduced to the accuracy of the map in a high percentage of the time if the switches have been solved correctly. For this situation, the cross-track uncertainty can be neglected and we can construct the inverse of **H** matrix using only two visible satellites:

$$\mathbf{G} = \begin{bmatrix} \mathbf{1}_L^{(0)} & 1 \\ \mathbf{1}_L^{(1)} & 1 \end{bmatrix} \Longrightarrow \mathbf{H} = \begin{bmatrix} LDOP_{_2}^2 & \bullet \\ \bullet & TDOP_{_2}^2 \end{bmatrix}. \tag{4.18}$$

This situation is considered of especial interest, because it defines the most common case in railway localization.

On the other hand, there are cases where we may know the approximate distance traversed along the tracks thanks to additional information coming from other sensors (i.e., odometers or inertial units). This can happen for example after having passed over a switch or in an initialization situation where discerning between parallel tracks is the crucial interest. Here, we can assume a known along-track position and the DOP calculations with two satellites could be solved like this:

$$\mathbf{G} = \begin{bmatrix} \mathbf{1}_C^{(0)} & 1 \\ \mathbf{1}_C^{(1)} & 1 \end{bmatrix} \Longrightarrow \mathbf{H} = \begin{bmatrix} CDOP_{_2}^2 & \bullet \\ \bullet & TDOP_{_2}^2 \end{bmatrix}, \tag{4.19}$$

where we focus on the cross-track uncertainty and the clock variation uncertainty.

Despite of the proximity of the latter assumptions to the truth in different situations, $LDOP_{_2}$ and $CDOP_{_2}$ values provide us with a simple way of comparing different sets of satellites in terms of along and cross contribution to the estimation process.

4.4 DOP with One Satellite

Using the DOP concept, for the special problem of railway localization we can define two geometrical factors for the case of one visible satellite. Those factors give us a straightforward information about how beneficial is the position of the satellite when solving the along and cross track position uncertainty.

These factors are here defined as $LDOP_1$ and $CDOP_1$ because it preserves the concept of dilution of precision, but the index ”1” describes the different metric and possible values of this figure of merit. For example, the $LDOP_1$ value gives information about how the pseudorange uncertainty converts to along-track position uncertainty (Figure 4.3).

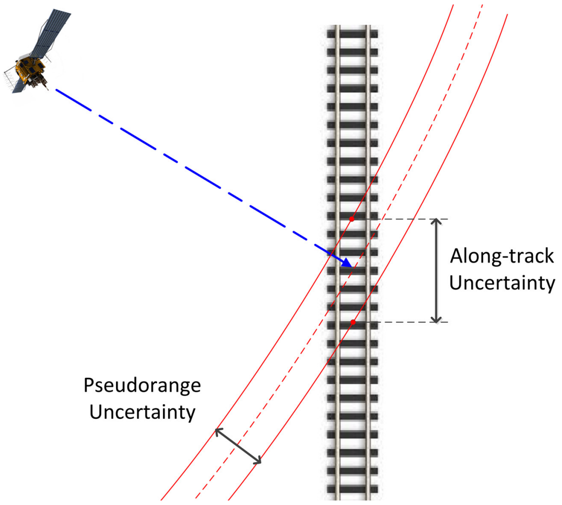


Figure 4.3: Pseudorange plane and railway track intersection

For the single satellite case, the geometric matrix reduces to one coordinate, and the

DOP matrix to one single value. For the along-track DOP we have:

$$\mathbf{G} = \mathbf{1}_L \Rightarrow \mathbf{H} = (\mathbf{G}^T\mathbf{G})^{-1} = (\mathbf{1}_L^T\mathbf{1}_L)^{-1} = \frac{1}{\mathbf{1}_L^2}. \tag{4.20}$$

The value contained in the $\mathbf{H}$ matrix is the $LDOP_1$, so if we replace $\mathbf{1}_L = cos\Phi \cos\Lambda$, we lead to the following expression:

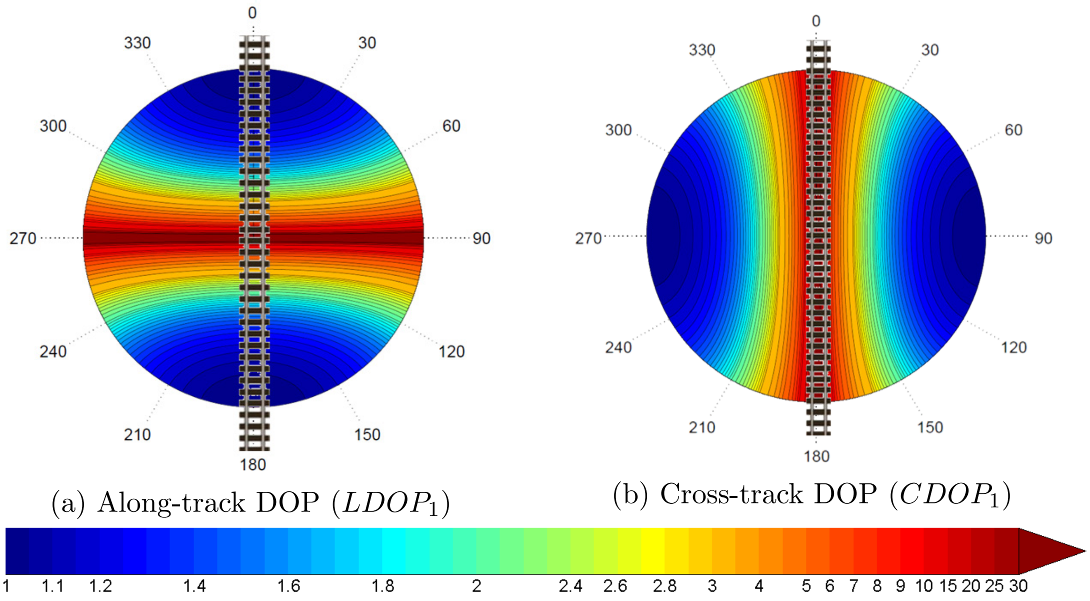
$$LDOP_1 = \frac{1}{\sqrt{(\cos\Phi\cos\Lambda)^2}}, \tag{4.21}$$

where Φ and Λ are the elevation and azimuth in the track frame respectively.

If we proceed similarly, the $CDOP_1$ value can be expressed as:

$$CDOP_1 = \frac{1}{\sqrt{(\cos\Phi\sin\Lambda)^2}}. \tag{4.22}$$

Please note that, 4.21 and 4.22 can be seen as the inverse of the projection of the line-of-sight vector user-satellite to the along-track and cross-track axis respectively. Furthermore, they are nonlinear expressions so, by contrast to the general DOP, the new range of mathematically possible values are: $LDOP_1, CDOP_1 \in [1, \infty]$



(a) Along-track DOP ($LDOP_1$)

(b) Cross-track DOP ($CDOP_1$)

Figure 4.4: Track dilution of precision

Figure 4.4a and Figure 4.4b are representations of $LDOP_1$ and $CDOP_1$ in a sky-plot view related to the considered track. In those graphs we can see these values for every azimuth and elevation point, where the azimuth is shown as the angle in degrees of the circle and the elevation by the radial axis magnitude. The center corresponds to 90° and the borders to 0° of elevation.

We can see in Figure 4.4a, that those satellite positions forming 90^{o} with the track, present a high $LDOP_1$ value (red), while those aligned with the track direction have lower $LDOP_1$ values (blue).

Chapter 5

Algorithm Implementation

In the following paragraphs, a more detailed explanation of the implementation of the filter is exposed. This coding part was in permanently revision and extension because represents the main research theory to development transfer stage of this thesis.

Any update of the filter is performed when a new measurement is received. In consequence, it is possible to differentiate between the process carried out for a new GNSS measurement and a new IMU measurement. From the implementation point of view, an update will include the two steps of the filter: proposing (by state transitions) and weighting (by sensor likelihoods). Additionally, an initialization of the particles and the filter is necessary.

5.1 Initialization

As the algorithm does not compute an absolute position to find the topological position on the map, it is necessary to know where to place initially the particles. As a simple approach, the particles could be spread all over the railway network, but this would be far from being efficient, because the map size may be thousand of kilometers and it would be necessary millions of particles to guarantee that at least one of them is in the right place.

Therefore, a good approach is to use an initial approximate point for the initialization. This point could be provided by, for example, the train driver or any other feature-based sensor. It could be also obtained from any reference to a train station. In this work, the first GNSS measurement has been used to that purpose.

The initialization process starts therefore, computing a PVT solution by using a Weighted Least Squares (WLS) algorithm (Section 2.1.3). Then, particles are spread over the railway map around that first position. The radius taken into account for this task has been 100 m., a distance more than enough to ensure the presence of particles near the true position (Figure 5.1).

The rest of the states (clock states and speed) are directly taken from the PVT solution for all the particles. Even if the clock bias and drift from the solution are not very precise, the high spreading of the particles ensures that after a couple of iterations, the filter

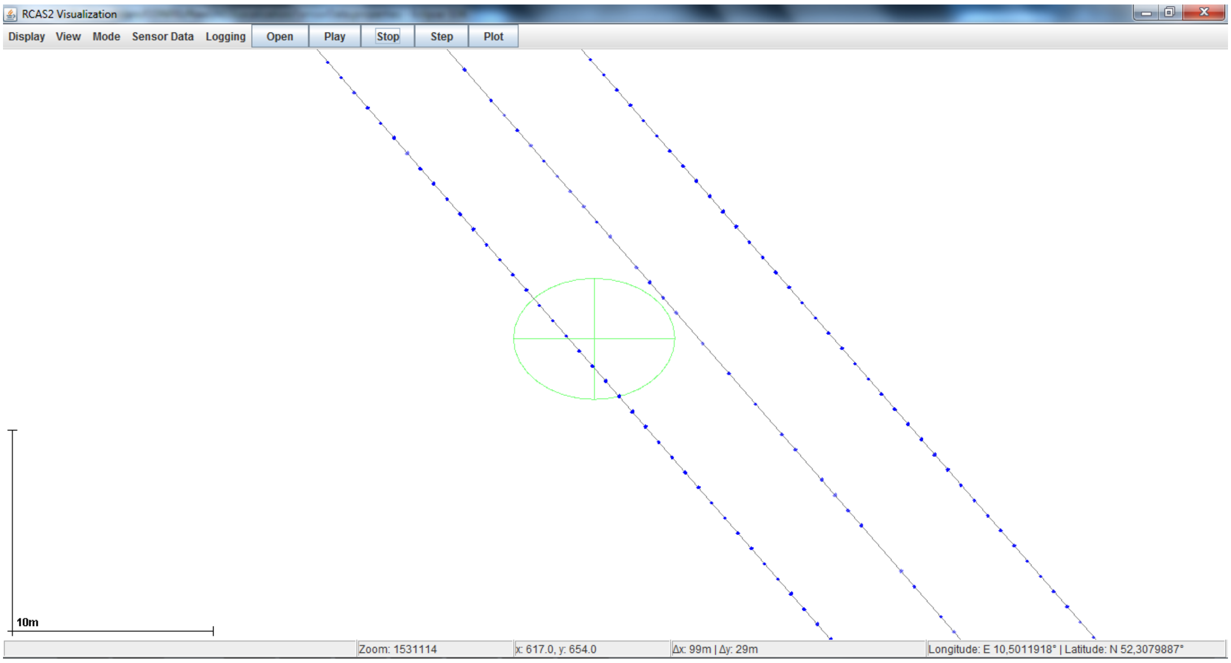


Figure 5.1: Particles Initialization by Weighted Least Squares
The green cross is the WLS solution. Particles are represented by blue circles.

converges and estimates the correct clock states.

5.2 IMU Update

The IMU update algorithm is the same for the loosely and the tightly coupled integrations. Although this algorithm was already built for [23], it has been constantly revisited to implement minor changes and extension. In addition, the calibration of some of its parameters is vital for the proper functioning of the system. In Algorithm 5 the procedure for the IMU updating is shown.

Algorithm 5 IMU propose and weighting

```
Input: IMUmeasurement, $\Delta t$
Calibration correction
for all Particles $p$ do
    $\Delta s \leftarrow constantAccelerationTransition(p, \Delta t)$      ▷ see Equation (3.13)
    $moveOnMap(p, \Delta s)$                                               ▷ see Equation (3.18)
    $likelihood \leftarrow imuModel(IMUmeasurement, p)$                    ▷ see Section 3.4.2
    $p.weight \leftarrow p.weight \times likelihood$
end for
Resample if necessary
```

5.3 GNSS Update

The raw GNSS update procedure constitutes the core of the coded implementation of this thesis. Whenever there is a new GNSS data available, the filter integrates it by the updating algorithm. This process consists of two steps: The *proposal step* must apply the transition models for all the particle states, including the clock evolution. The *weighting* is performed by evaluating how likely are the measurements regarding the states of every particle. Note that, as previously said, the error estimation is performed by a single state Kalman filter outside the particle filter and as every particle needs an error estimation, a Kalman filter is thus indispensable for every particle and every satellite.

In Algorithm 6, we can see the general procedure of the raw GNSS data updating step of the particle filter.

Algorithm 6 Raw GNSS propose and weighting

```
Input: rawGNSSdata
for all Particles $p$ do
    $p$ Movement and Position Transition
    $p$ Clock Transition
    for all Visible Satellites do
        predict long error (KF prediction)
        compute predicted pseudorange
        update long-term error (KF update)
        compute predicted Doppler
        compute pseudorange and Doppler likelihoods
        $p.weight \leftarrow p.weight \times pseudorangeLikelihood \times dopplerLikelihood$
    end for
end for
Resample if necessary
```

The computation of both the predicted pseudorange and predicted Doppler measurement are complex algorithms. The correction algorithms and models required to compute a PVT solution in classical positioning methods are still applied (Appendix B). However, since it is assumed that every particle has the true state values, they are here used to build an expected observation. Therefore, in the particle filter, we compute the predicted measurements using the particle position, velocity and clock values as parameters, instead of being considered as unknowns.

The algorithm developed to build the predicted pseudorange out of the particle states and ephemeris information, is expressed in the flow diagram of Figure 5.2.

Notice, that the factors used in Figure 5.2 to compute the predicted pseudorange, are those presented in the pseudorange general equation in (2.1).

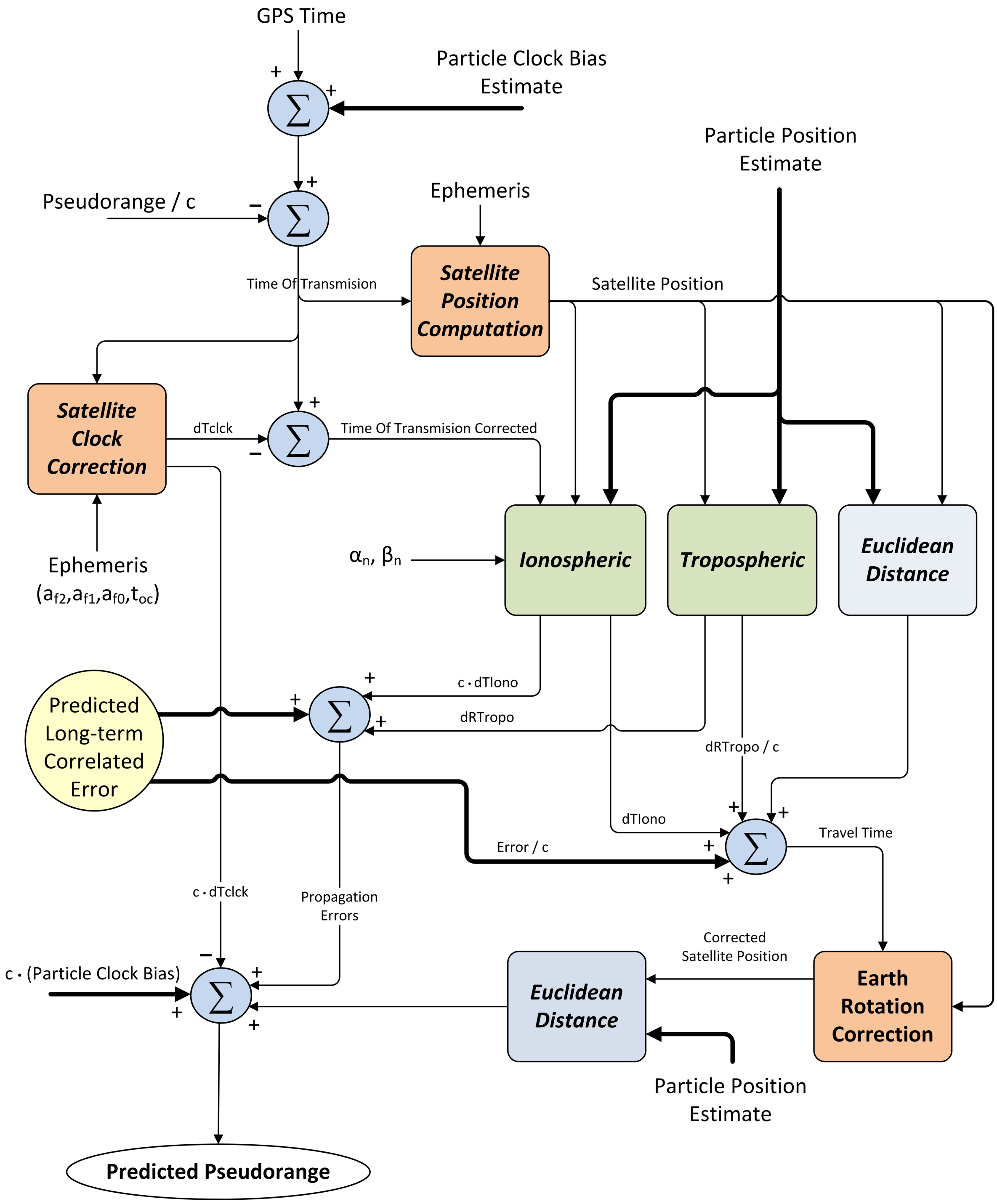

GPS Time
Particle Clock Bias Estimate
Pseudorange / c
Ephemeris
Particle Position Estimate
Satellite Position Computation
Time Of Transmision
Satellite Position
Satellite Clock Correction
dTclck
Time Of Transmision Corrected
Ephemeris
$(a_{f2}, a_{f1}, a_{f0}, t_{oc})$
α_n, β_n
Ionospheric
Tropospheric
Euclidean Distance
Predicted Long-term Correlated Error
c • dTIono
dRTropo
dRTropo / c
dTIono
Error / c
Travel Time
c • dTclck
Propagation Errors
Corrected Satellite Position
c • (Particle Clock Bias)
Euclidean Distance
Earth Rotation Correction
Particle Position Estimate
Predicted Pseudorange
+
−
Σ

Figure 5.2: Predicted pseudorange algorithm

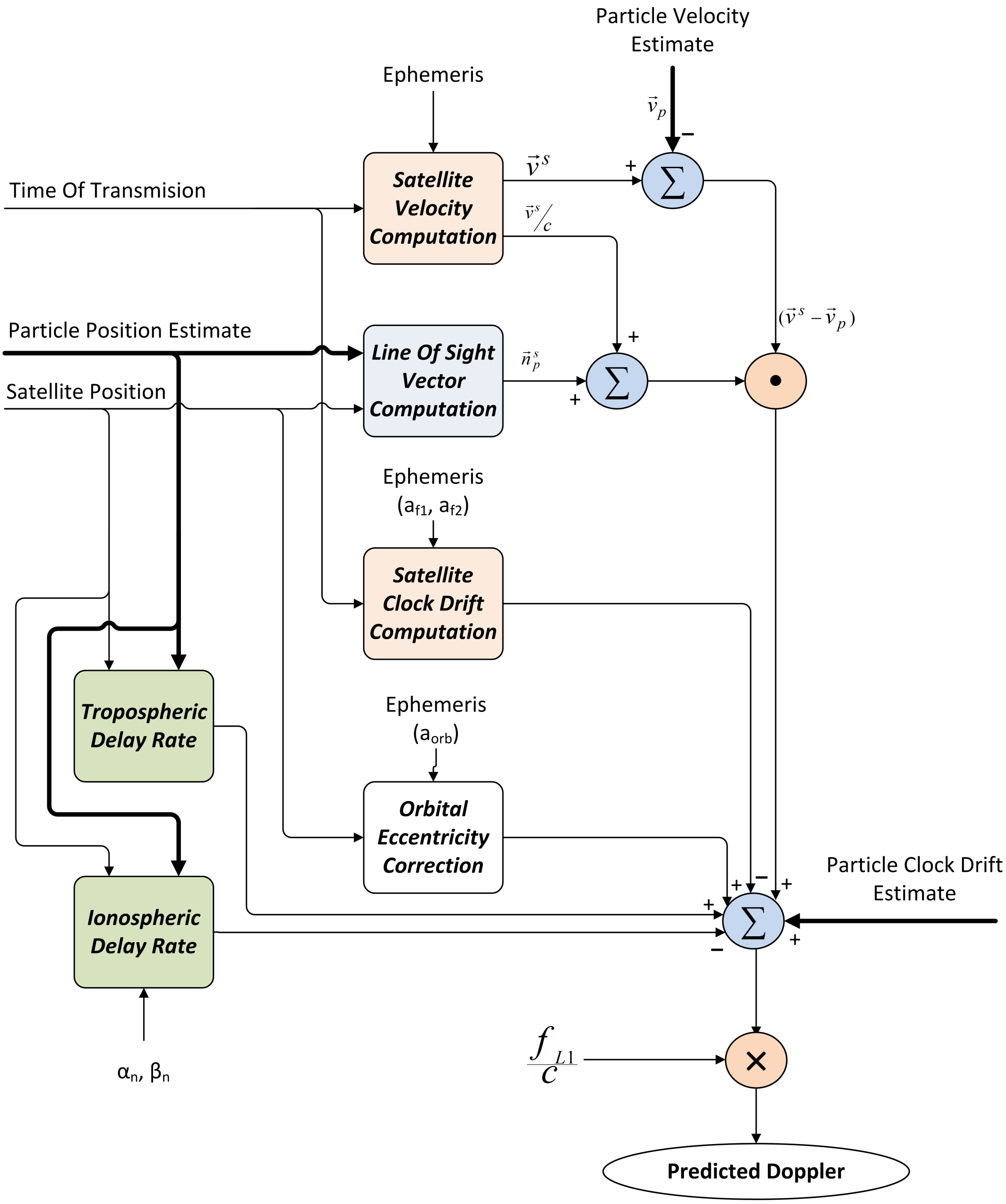

Particle Velocity Estimate
Ephemeris
$\vec{v}_p$
Time Of Transmision
Satellite Velocity Computation
$\vec{v}^s$
$\vec{v}^s/c$
$(\vec{v}^s - \vec{v}_p)$
Particle Position Estimate
Satellite Position
Line Of Sight Vector Computation
$\vec{n}_p^s$
Ephemeris (a_{f1}, a_{f2})
Satellite Clock Drift Computation
Tropospheric Delay Rate
Ephemeris (a_{orb})
Orbital Eccentricity Correction
Particle Clock Drift Estimate
Ionospheric Delay Rate
α_n, β_n
$\frac{f_{L1}}{c}$
Predicted Doppler

Figure 5.3: Predicted Doppler algorithm

On the other hand, the predicted Doppler calculation is based on general Doppler equation (2.2). For this task, we also need to know the satellite position, so for the Doppler algorithm, we assumed in Figure 5.3, that we already know the corrected satellite position and the corrected time of transmission.

The implementation of Equation (2.2) can be simplified by neglecting some of the following terms:

- **Potential Difference:** This potential difference term can be neglected in this case because all satellites share the same difference and therefore its effect on Doppler measurements are absorbed by the clock drift estimation. Additionally, the effect on the clock drift estimation is low [9].
- **Sagnac Rate:** This correction term is typically very small, it may reach some millimeters per second [9], which is below the accuracy of the velocity estimation.

In practice, as the computation of the satellite's velocity is based on the position calculation, both processes are performed at the same time. There are also other parts in common between pseudorange and Doppler algorithms, so in order to build an efficient global procedure both algorithms are performed at the same time not to carry out the same operation twice. The position and velocity computation process can be found in detail in Appendix A.

5.4 Filter Configuration

In this particle filter implementation there are some important parameters that are directly involved in the performance of the estimation.

- **Number of particles (N_p):** This is possibly the most relevant parameter of particle filters. Particles represent the posterior distribution in an approximate and discrete fashion. If we do not have enough particles, there might not be any particle with the true state value or near to it. As more states are added to the particles, the dimensional degree of freedom increases, and therefore more particles would be needed to cover all the possibilities.

  Another key fact is the transition models. If the model are close enough to the reality, particles will stay close to the truth. On the other hand, if we have poor models, more particles should be used for the estimation because they will spread more over the different states.

  For this thesis work, a number of 1000 particles have been used. Thanks to the constraint of the map, the range of the state values get reduced, showing that this number of particles is enough to achieve the required performance. Furthermore, with this number of particles, the complexity of the filter is low and a real-time processing have been ensured with a commercial desktop computer[1].
- **Resampling:** As commented in Section 2.2.3, this is a key step in a particle filter. The problem related to the frequency of the resampling has been faced in two ways during this work:

[1] Intel® Xeon® CPU E31225 @ 3.10GHz (4 cores, 64bit) with 2x4096 Mbytes RAM

1. *Resample after every update.* Although this is the simplest approach, this tactic has shown poor results. Due to the different rate of measurements of IMU and GNSS, and the cumulative errors that appear in inertial navigation, this approach lead us to a fast lost of diversity of particles.
2. *Using a threshold for the number of effective particles* N_{eff} (Equation (2.28)). Proving different values for this threshold, we have reached a compromised value which keeps a high particle diversity without spreading excessively the cloud of particles. This way, the switches can be resolved correctly. The value selected was $N_{eff} = 70\% N_p$.

Chapter 6

Framework

The localization application must provide a position solution for a train based on a multi-sensor input to the system. For this thesis, a prototype framework for testing the algorithm performance is desired. At the same time, sensor interfacing should be developed for the online system on sight. Concretely, the implementation should thus focus on the following system operation modes:

- **Online Operation:** The online mode refers to the final system operation. That is, in a real situation where the sensors are mounted on the train and the position solution must be broadcast to the driver or other trains to prevent, for example, possible collisions, such as in the RCAS system [3].

- **Offline Operation:** The offline mode is a testing mode. The online mode is reproduced by recorded data instead of the measurements coming directly from the sensors. A solution is also produced but with the purpose of testing the system and its performance.

The code should also guarantee that the system is able to provide the output, in the worst case, at the same rate as the input. Due to the complexity of the algorithms or the rate of arriving measurements from the sensor, this requirement may be compromised. Hence, the coding process must take this into account to produce a *real-time system*.

In Section 6.2 the structure built for the IMU and GNSS data will be explained. The interfacing of a GNSS receiver will also be commented in Section 6.3 and the testing offline setup in Section 6.4. Finally, the minor visualization extensions that were added and that permitted an easy debugging process and supervising of the system are commented.

6.1 System Architecture

The implementation was performed by extending the existing Java® RCAS localization framework that was mainly developed by Oliver Heirich at DLR. The Integrated Development Environment (IDE) used for this purpose was Eclipse®.

The coding extension followed logical steps that lead to an easy test and operation. The code has grown based on a *Model-View-Controller* (MVC) pattern that helps not only in an simple organization of the classes but to find the role of every part in a fast way. The system architecture overview is shown in Figure 6.1.

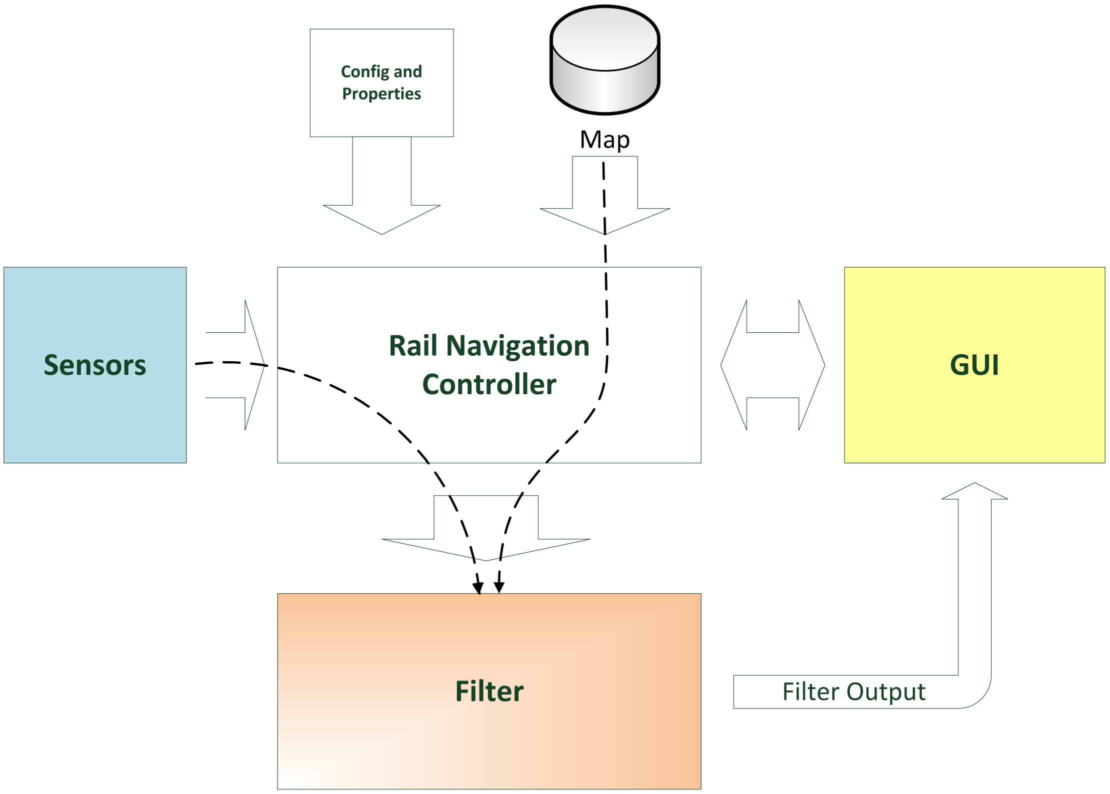


Figure 6.1: System architecture

In Figure 6.1, it can be identified the following blocks:

- **Rail Navigation Controller:** The controller is responsible for initializing and configuring the system. It starts the sensor's connections, reads and prepares the map, and initializes the *Graphical User Interface* (GUI). After that, it is in charge of managing the system while operating. It receives the sensor's outputs and distributes it to the filter.

- **Config Parameters:** The system needs some configurable parameters for the initialization, setting up and calibration. A configuration properties file system makes this process simple and prevents from repetitive manual input commands.

- **Map:** A digital map must be previously available. It is also required to adapt it to the data structure of the application frame.

- **Sensors:** Here are taken into account both the online sensor interfacing and the replayer platform which provides measurements periodically in an offline mode.

- **Filter:** Here, the navigation algorithm is implemented. It can send data to the GUI visualization for updating the output visualization.

- **Graphical User Interface (GUI):** This represents the visual output of the system, that is the map representation, the output of the filter and it may provide with further information for supervision purposes. Here, the user can also interact with the system.

Previous work on the RCAS localization framework had already addressed:

- **Map handling:** The map consisted of a XML file that is parsed and prepared for its use in the filter.

- **Visualization Output:** A main visualization frame where the map could be shown and that provides an easy interfacing for showing additional results.

- **Inertial Sensor Interface:** Interfacing for a XsensMT inertial sensor was already created, so it is possible to connect this sensor to the software.

- **Loosely Coupled Integration:** A loose IMU/GNSS integration had already been implemented in a simulation context.

The main coding objectives of this work can be summarized as:

1. Extension of the loosely coupled integration from simulation level to real measurements level.

2. Integration of the tightly coupled approach.

3. Development of an extended GNSS receiver driver.

6.2 Data Structure

In order to easily access the information provided by a GNSS receiver, maintaining the modularity of the software and the receiver independence of the system, an interface scheme is necessary. Hence, a dedicated data structure must also be created for the common shared information of every sensor between the interface's sides.

6.2.1 IMU Data Structure

The IMU Data Structure was previously built by the supervisor of this thesis, Oliver Heirich [23]. Now, an overview of the important IMU measurements container for this thesis is provided:

- **TimeStamp:** The reference time of the measurements in Unix TimeStamp.

- **Accelerations:** Acceleration measurements in the three axis (x,y,z).
- **Gyroscope:** Turn rates measurements in the three axis (x,y,z).

As commented in Section 3.4.2, a three degrees of freedom (3DoF) model is adopted for the IMU information in this work. Therefore, the measurements used inside the filter are: Acceleration in X, Acceleration in Y and Turn rate in Z.

6.2.2 GNSS Data Structure

Most GNSS receivers are able to provide a PVT solution in *National Marine Electronics Association* (NMEA) format [30], so this format is here considered and the framework data structure to store the PVT solution is based on it. On the other hand, for online applications, receivers typically provide raw data in a receiver specific format. For this reason, it is necessary to create a receiver-independent data structure to handle the raw GNSS data in our software.

In the following, both the PVT and the raw GNSS data structure developed for this thesis is explained.

PVT Solution Data

PVT Solution data structure is based on NMEA messages and is built for the loosely coupled approach. In particular, the relevant information used was:

- **Position:** Latitude, longitude (degrees) and altitude (m) position.
- **Speed:** Estimated user speed (m/s).
- **Heading:** Orientation of the user referred to Earth north (degrees).
- **Uncertainty:** Estimated uncertainty of the previous information. These values are important to characterize the distributions of the filter (i.e., likelihoods).

Raw GNSS data

The raw GNSS measurements used for this work are pseudoranges and Doppler. However, to be able to use this information into the system, it is also essential to acquire other information such as the satellite ephemeris or the ionospheric coefficients. The raw GNSS data is therefore a global container with all the required information for computing a GNSS solution. In Figure 6.2, we can see a class diagram that represents the structure of the raw GNSS container.

The different objects in this figure are here explained:

- **Raw GNSS data:** This is the global container of all the required information necessary for the GNSS processing. It may contain the actual time step observation, the available and updated ephemeris of every satellite and the ionospheric coefficient transmitted in the navigation message.

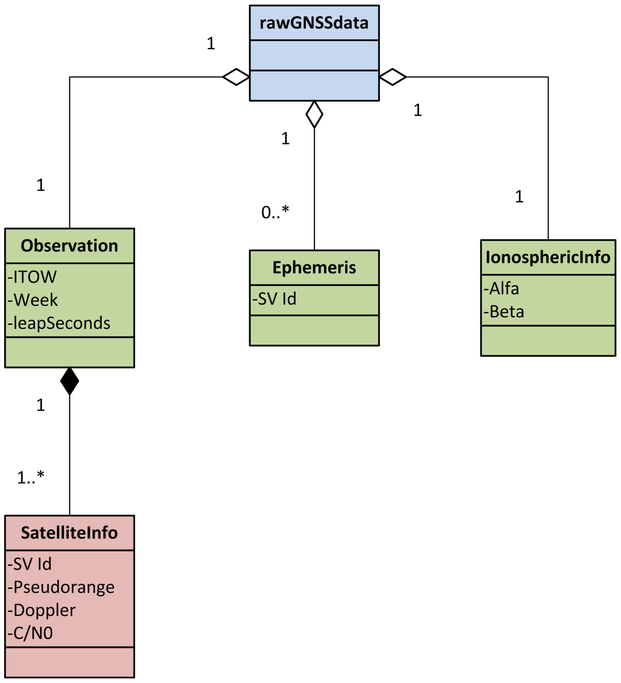


Figure 6.2: Raw GNSS data structure class diagram

- **Observation:** An observation is one epoch set of measurements from every satellite in view at a certain moment. It thus contains the reference time of the epoch and the measurements information of every satellite.

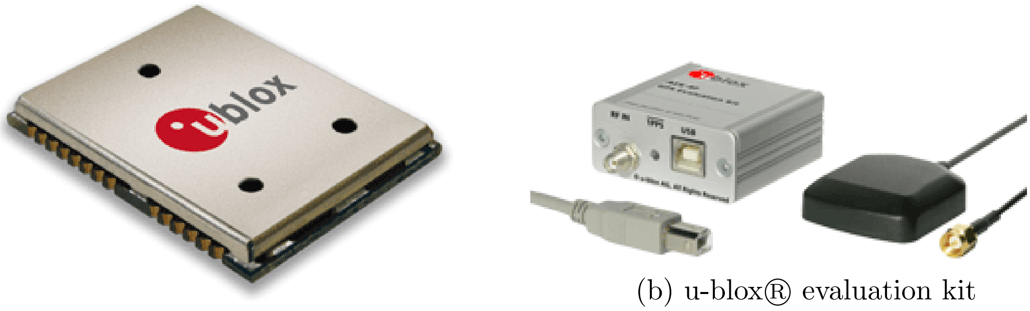
  - **Satellite Info:** This is the container of the measurements of one satellite at one epoch. It includes for this thesis the *Space Vehicle* (SV) Identification number, the pseudorange and Doppler measurements and a quality factor in the form of Carrier-to-Noise ratio (C/N0).

- **Ephemeris:** Basically, it contains the information from the first three subframes of the navigation message transmitted by every satellite. Those subframes inform about the state of the space vehicle at a certain moment. The position and velocity of the satellite is thus computed from this information [31]. The coefficients for the correction of the satellite clock bias and drift are also transmitted in these subframes.

- **Ionospheric Info:** GPS satellites also broadcast the coefficients for Klobuchar's ionospheric model correction algorithm [32]. They consists of 8 coefficients which represent the perturbation of the ionosphere at this epoch. This coefficients are computed in the control segment and then transmitted to the satellites for broadcasting.

6.3 GNSS Receiver Interfacing

The chosen receiver for the system is a u-blox® LEA-6 (Figure 6.3). Not only the interfacing of this receiver is essential for the online system, but having this connection and the receiver available during this thesis was used in several stages. Firstly, it helped to understand the functioning of GNSS receivers and the GPS system itself. It was also critical when building the pseudorange and Doppler predicted algorithms and when creating the Weighted Least Squares (WLS) Algorithm. For this testing, an evaluation kit was used in this process (Figure 6.3b).



(b) u-blox® evaluation kit

(a) u-blox® chipset

Figure 6.3

The objective of this receiver interface is the translation of the u-blox protocol messages to the framework GNSS data structure. A serial driver for the USB connection of the evaluation kit had been previously accomplished. The next step was to identify the useful messages of the receiver, analyzing them and translating that information to the created data structure. An interface connection between the translator and the application was also required. The final receiver interfacing can be seen in Figure 6.4.

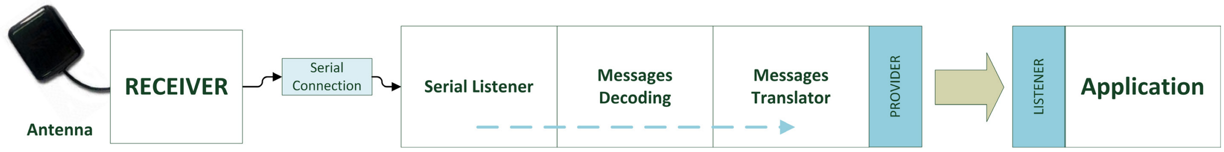


Figure 6.4: GNSS receiver interfacing

The u-blox protocol messages used for this thesis were "UBX-RXM-RAW", "UBX-AID-EPH"' ' and "UBX-AID-HUI" [33].

6.4 Offline Replay Mode

The objective of the offline mode is to reproduce the functioning of the online mode so that the system and its algorithms can be tested. For this purpose, we first assume that we have recorded data from the required sensors. Those files must be then parsed and transformed into the system data structure. Finally, we need to provide those measurements sequentially ordered to the application to reproduce the exact behavior of a real scenario. In Figure 6.5, we can see a block diagram of the latter steps for the tightly coupled approach.

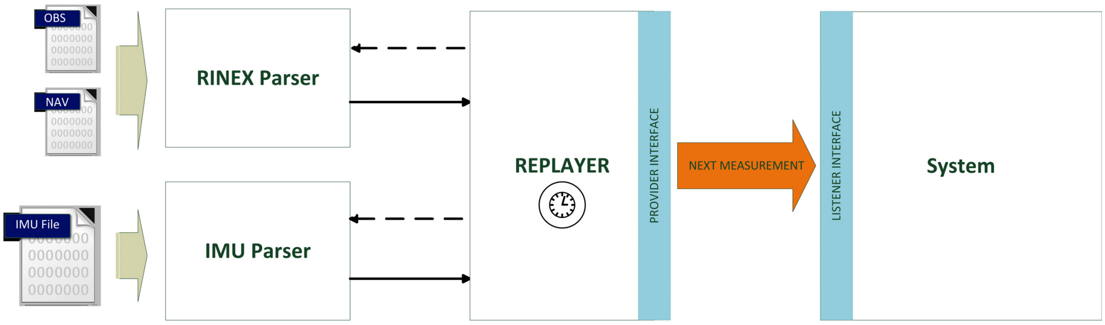


Figure 6.5: Files parsing and replay diagram (*tightly coupled approach*)

The sensor formats parsing that were made supported during this thesis are:

- **IMU XsensMT format:** The XsensMT records in a TimeStamp base time, all the information about the accelerometers, gyroscopes, magnetometers and orientation. This format had been previously parsed and adapted to the system data structure.
- **NMEA format:** The *National Marine Electronics Association* format consist of different messages produced at every time step [30]. This format contains the processed PVT solution information of the receiver.
- **RINEX format:** It is a commonly used post-processing format, where the receiver records all the necessary raw information to compute a PVT solution. RINEX format typically consists of two different files [34]:
  - *Navigation File*: Here, it is stored the ephemeris information of the satellites and the ionospheric Klobuchar's coefficients.
  - *Observation File*: In this file, the different raw observation measurements of every satellite at every epoch are recorded.

The parsing of GNSS measurements became a highly time consuming step of the coding. In particular, due to the great number of parameters and coefficients, the RINEX format is the most complex one.

In order to compare measurements from different sensors to replay them in the correct order a common base time was choosen. This common time is the Unix TimeStamp because it is the most general one and easy to handle for a computer language, since it is formatted as an integer number.

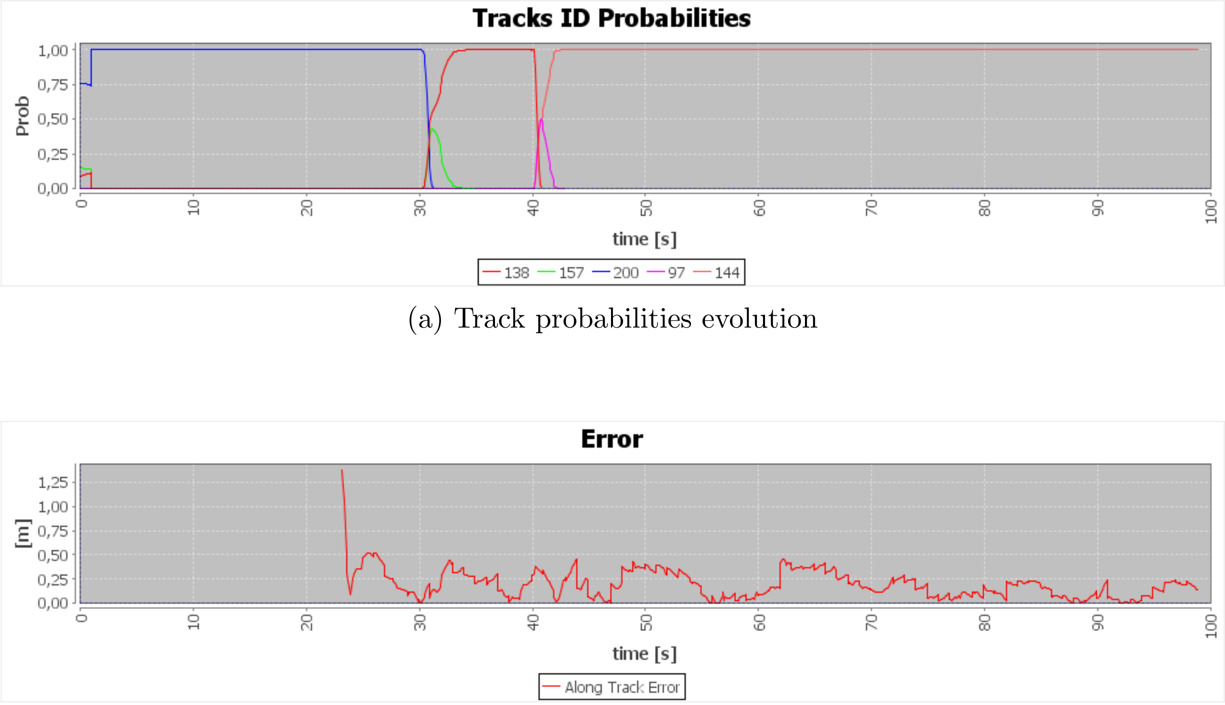
Every sensor data structure was thus extended from a general data structure called 'TimeStamp Data'. Thanks to the Java® inheritance, the replayer was able to compare them in a priority queue. Additionally, the replayer, which was also coded as a independent thread, permitted the configuration of its data replay speed. That was essential to accelerate any test on the proposed algorithm.

Finally, this replayer was integrated to the controller of the application in the same way as the online sensors. As a result, the system can be configured to operate either in an online fashion or reproducing a real scenario in an offline mode.

6.5 Visualization Extension

Several visualization extensions have been created in the RCAS framework. Some of them respond to the necessity of a visual proof of different steps of the filter or the behavior of some of the parts. Others can be used to monitor the state of the GNSS receiver or the system output. For many of the visual frames created, the Java® GUI designer *WindowBuilder* [35] has facilitated the tasks. Another powerful graphical tool used for creating different graphs was the *JFreeChart* library [36].

The most important contributions that have been developed are now presented.



(a) Track probabilities evolution

(b) Output error evolution

Figure 6.6: System evolution information

6.5.1 Track Probabilities and Output Error

The evolution of the track probabilities is considered to be an indicator value of the performance of the algorithm when passing across a junction. In Figure 6.6a, when a switch is reached, it is observable that two tracks get 50% of probability. After some seconds, thanks to the IMU and Doppler information, the correct branch gets higher probability and the switch is resolved successfully.

Another important information while debugging the implemented code or testing the system is the position error. A Java frame was also created to show the position error when a *ground truth* reference is available (Figure 6.6b).

6.5.2 Particle Likelihoods

It is very complex to understand what is really happening at every moment inside the particle filter algorithm if we only observe the final output. A really helpful visualization was created to have a visual proof of the likelihood step of the filter.

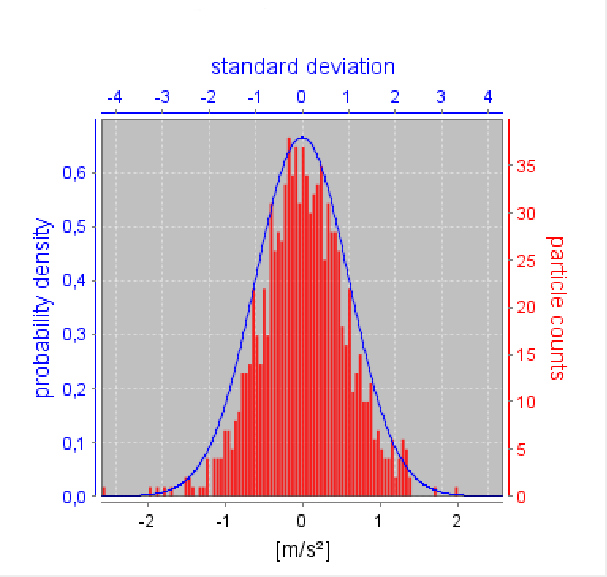
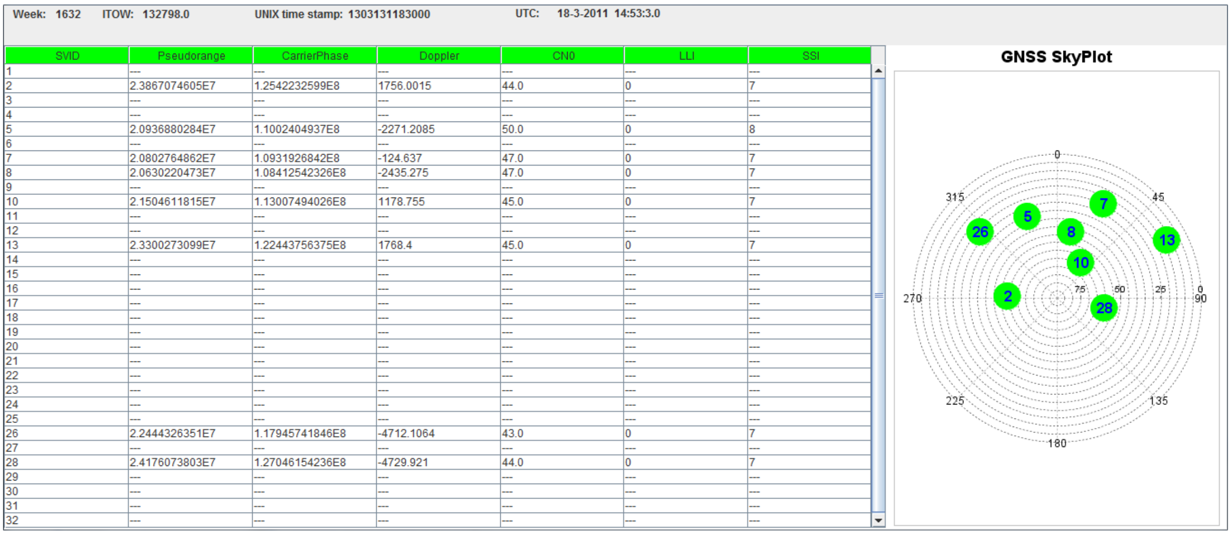


Figure 6.7: Particles likelihood representation

In Figure 6.7, we can see an example of the likelihood frame. Represented in red, it shows the particles histogram of the predicted measurement. The graph is centered at the measurement mean and a Gaussian distribution density with the configured measurement uncertainty is also drawn. By looking at the difference between the cloud of particles and the new observation we could detect problems and adjust the calibration of the filter.

6.5.3 GNSS Monitor and Skyplot

An additional frame was implemented to monitor and supervise the GNSS data (Figure 6.8). This visual information was really helpful firstly, to debug problems and ensure that the correct information was arriving to the system and to the filter. Secondly, to monitor the number of visible satellites at every time and the quality of those satellites.



Week: 1632 ITOW: 132798.0 UNIX time stamp: 1303131183000 UTC: 18-3-2011 14:53:3.0

SVID	Pseudorange	CarrierPhase	Doppler	CN0	LLI	SSI
1	---	---	---	---	---	---
2	2.3867074605E7	1.2542232599E8	1756.0015	44.0	0	7
3	---	---	---	---	---	---
4	---	---	---	---	---	---
5	2.0936880284E7	1.1002404937E8	-2271.2085	50.0	0	8
6	---	---	---	---	---	---
7	2.0802764862E7	1.0931926842E8	-124.637	47.0	0	7
8	2.0630220473E7	1.08412542326E8	-2435.275	47.0	0	7
9	---	---	---	---	---	---
10	2.1504611815E7	1.13007494026E8	1178.755	45.0	0	7
11	---	---	---	---	---	---
12	---	---	---	---	---	---
13	2.3300273099E7	1.22443756375E8	1768.4	45.0	0	7
14	---	---	---	---	---	---
15	---	---	---	---	---	---
16	---	---	---	---	---	---
17	---	---	---	---	---	---
18	---	---	---	---	---	---
19	---	---	---	---	---	---
20	---	---	---	---	---	---
21	---	---	---	---	---	---
22	---	---	---	---	---	---
23	---	---	---	---	---	---
24	---	---	---	---	---	---
25	---	---	---	---	---	---
26	2.2444326351E7	1.17945741846E8	-4712.1064	43.0	0	7
27	---	---	---	---	---	---
28	2.4176073803E7	1.27046154236E8	-4729.921	44.0	0	7
29	---	---	---	---	---	---
30	---	---	---	---	---	---
31	---	---	---	---	---	---
32	---	---	---	---	---	---

Figure 6.8: GNSS information visualization

Another interesting and common visualization of the satellites is a *skyplot*. A skyplot is a representation of the geometry of the satellites in the sky. At the right part of Figure 6.8 we can see the position of every visible satellite related to the user. This plot represents in a polar graph a zenithal projection of the sky centered at the user approximate position. Therefore, the radius represents the elevation, where the center is 90° and the border 0°, and the angle is the azimuth deviation from the Earth's north.

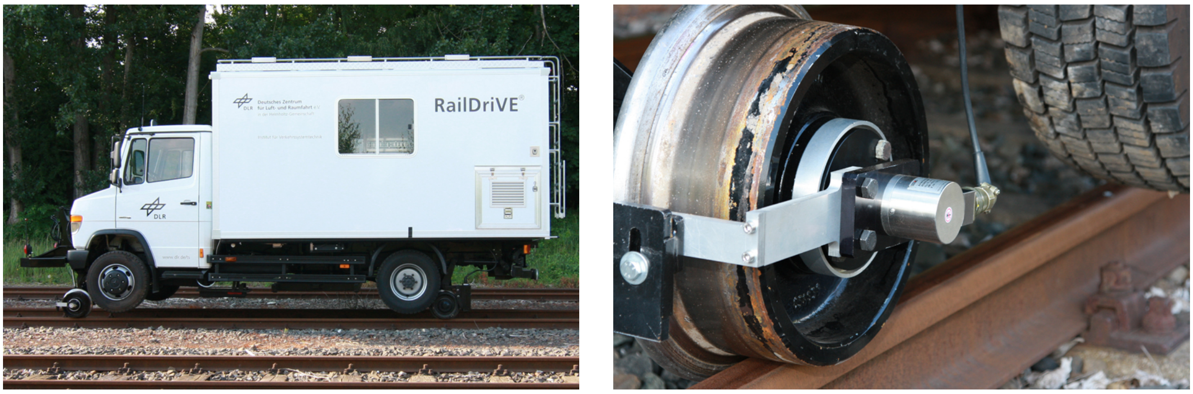
Chapter 7

Experimental Setup

One contribution of this thesis is the extension of the railway localization framework by a measurement data replayer. This replayer enables a more realistic algorithm evaluation during the development phase, in contrast to simulated data. This chapter provides an overview of the complex measurement campaign that led to the recorded data used in this thesis to validate the developed and implemented algorithm. The different sensors, setup configuration and map generation are presented, as well as the selected scenario for the analysis.

7.1 Measurement Campaign

Within the context of the Railway Collision Avoidance System (RCAS) project, a measurement campaign in a real railway environment was carried out by DLR in May 2011.



(a) RailDrive® vehicle (b) RailDrive® supplementary train wheel

Figure 7.1: Rail Driving Validation Environment (RailDriVE®)

The setup consisted of the *Rail Driving Validation Environment* (RailDriVE®) shown in Figure 7.1a, a vehicle developed by the Institute of Transportation Systems of DLR for testing and measurements [37].

The vehicle was designed as a mobile laboratory where a number of sensors, communication equipment and computer technology were equipped on board. Additionally, RailDriVE has supplementary train wheels (Figure 7.1b) that can be deployed, and therefore is capable of railing over the train tracks.

The data was recorded at the industrial railway network of the Brunswick inland port (Hafenbahn Braunschweig).

7.2 Sensors On Board

A complete set of sensors were placed on the vehicle. Those used specifically for the purpose of this thesis are a *Septentrio AsteRx2* GNSS receiver (Figure 7.2a) and a *XsensMTx* inertial measurement unit (Figure 7.2b).

(a) Septentrio AsteRx2 GNSS receiver

(b) XsensMTx inertial sensor

Figure 7.2: Sensors onboard

7.2.1 Septentrio AsteRx2

Septentrio AsteRx2 is a professional dual-frequency GNSS receiver capable of tracking different GNSS constellations. The receiver antenna was placed on the roof of the RailDriVE® vehicle so that it could have the best possible visibility. The receiver is able to record GNSS measurements in different formats, also providing a flexible configuration process to the user. In this thesis, we use GPS data recorded in NMEA format for the PVT solution and in RINEX format to store the raw measurements, and the input measurement rate for the filter was set to 1 Hz.

As it has been commented before, some parameters of the filter are dependent of the receiver type and configuration. That is the case to model the clock evolution and to compute the noise of the tracking loops used in the likelihoods of the filter. In Table 7.1, those default important parameters are summarized for the Septentrio AsteRx2 GNSS receiver [38].

Parameter	**Value**	**Use**
DLLBandwidth	0.25 Hz	Noise of pseudorange
PLLBandwidth	10 Hz	Noise of Doppler
MaxTpDLL	100 msec	Noise of pseudorange
MaxTpPLL	10 msec	Noise of Doppler
Clock Bias Range	[0-1] ms	To properly model the clock jumps

Table 7.1: Septentrio AsteRx2 GNSS receiver default parameters

7.2.2 XsensMTx

The XsensMTx is a small, low-cost inertial measurement unit based upon miniature (MEMS) inertial technology [39]. It can provide 3D acceleration, 3D rate of turn and 3D earth-magnetic field data at a high rate bandwidth. For this work, we used both acceleration and turn rates measurements at 10 Hz.

Additionally, XsensMTx accepts synchronization pulses for its base timing. This was used in the data recording process to synchronize the IMU timing by the GPS time of the GNSS receiver, so that both sensors were running with a common time.

7.3 Ground Truth

Precise position of the vehicle is required to evaluate the performance and accuracy of the system output. This output must be compared with a reference position, usually called *ground truth*.

Although it is difficult to know the exact position of the vehicle at every time, it is normally used as ground truth the measurements of a sensor with much more accuracy than our expected output. For this work, this ground truth was obtained from *Leica tachymeter* measurements. The setup consisted of a Leica type TCRP 1201, where an absolute position was computed thanks to a ATX1230 GNSS SmartStation in post processing. This tachymeter is able to measure distance and angle between the reference station and a reflective prism that was placed on the vehicle (Figure 7.3). Then, the relative position was translated into local coordinate frame (Latitude, Longitude, Altitude) with a centimeter precision.

Because of its nature, this sensor is only able to track the prism as far as a certain distance, so ground truth data was not available during the whole time.

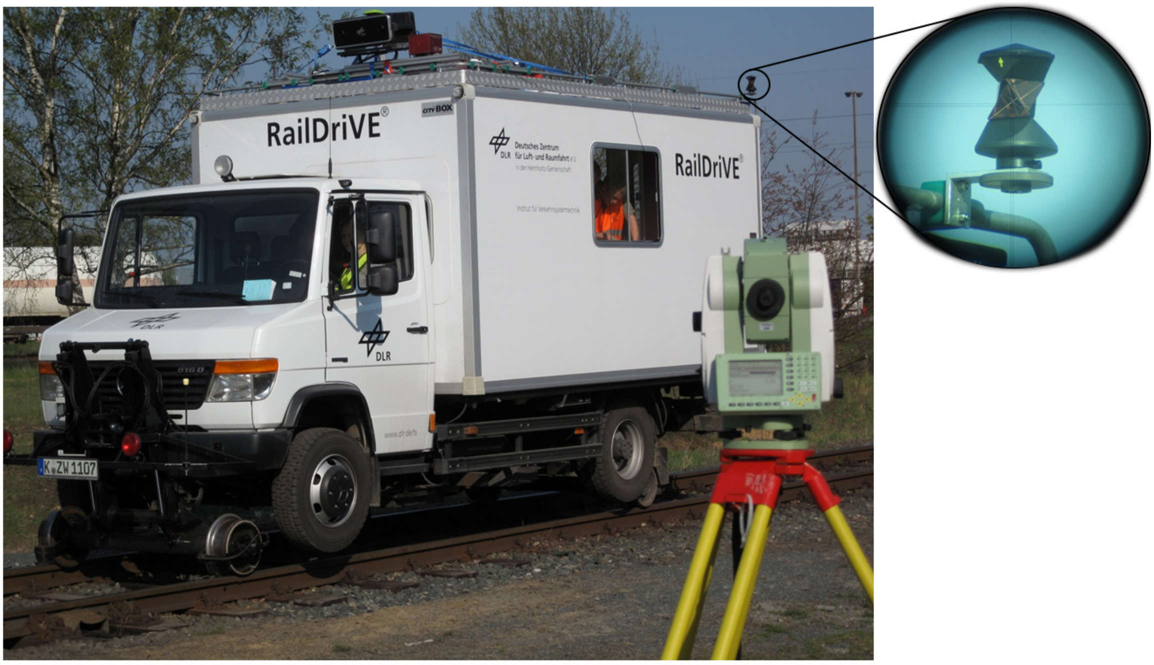


Figure 7.3: Leica tachymeter set-up

7.4 Railway Map

As our localization approach uses a digital map in the integration algorithm, it was necessary to have the map of the railway tracks where the evaluations were carried out. This map was generated using the ground truth information from the Leica measurements.

The map contains information of topological, geographic and track geometric data, as well as further information of track distances and connections of the railway network [25]. In practice, the digital map is contained in a XML file, where the list of tracks is stored. In the same way, every track is described by a list of discrete track points, its track ID and the connections to other tracks. Additionally, the RCAS framework is able to provide information at every point of the tracks by interpolating through the available stored points.

All the recorded data and the generated map was obtained before the beginning of this thesis.

7.5 Scenario

In order to validate the proposed system in critical railway localization situations (i.e., parallel tracks and switches), we had to be very careful when choosing the testing scenarios. It was needed a place and moment where there were no additional side effects (e.g., building multipath) and where ground truth data were also available. This was essential to obtain clear conclusions and analyze the algorithm performance.

The chosen scenario was a three parallel track situation in an open sky area where two switches could be passed over. In Figure 7.4, we can see this scenario, where the yellow marks represent the starting point of the junctions and the direction of the vehicle in the test runs are indicated by red arrows.

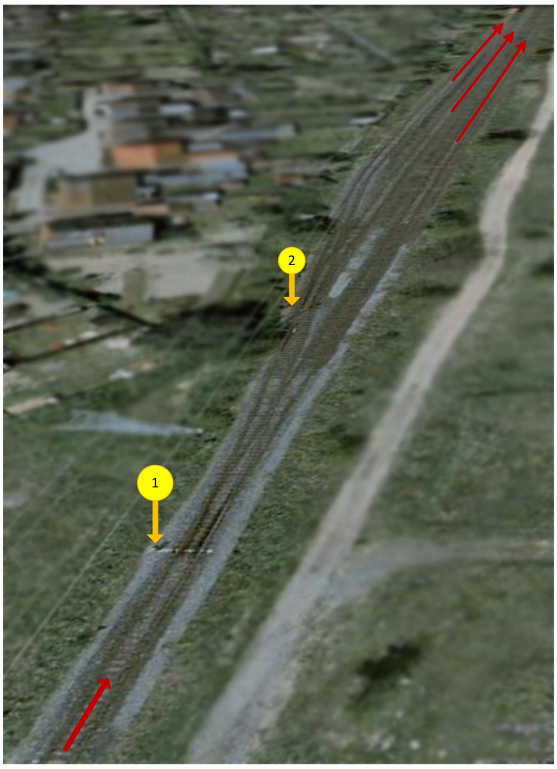


Figure 7.4: Chosen Test Scenario
Yellow marks are the starting points of switches

Chapter 8

Results and Evaluation

The evaluation of the railway localization approach will be presented in this chapter. This assessment was carried out by running the framework under certain conditions and settings. Then, the relevant output data was recorded to files so that they could be processed and presented with MATLAB ®.

As a first step, the performance of the filter in a normal visibility situation will be analyzed. Secondly, a comparison between the loosely coupled integration and the tightly coupled integration is performed. Next, the role of Doppler measurements are explored in Section 8.3. A further analysis with few visible satellites is also presented. Finally, the effect of the long-term error estimation is commented.

8.1 System Output

First, a comparison between the different approaches for the particle filter output is evaluated. Three of the different output approaches presented in Section 3.5, were implemented in this thesis: *Highest Weighted Particle*, *Most Likely Particle* (ML) and *Minimum Mean Squared Error* (MMSE) output.

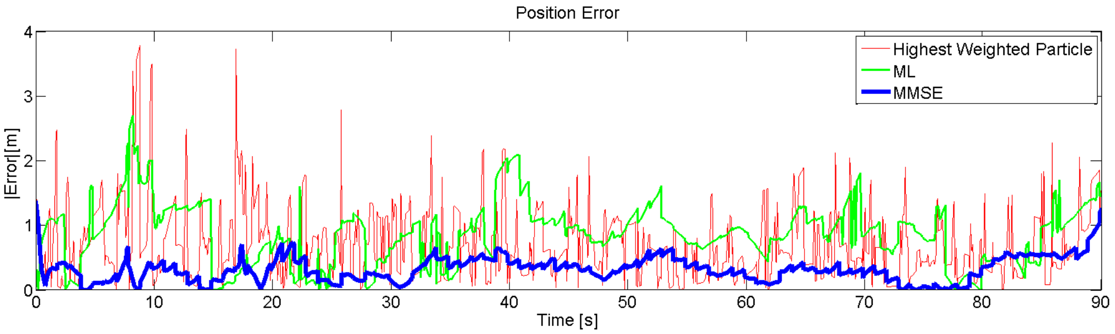


Figure 8.1: System output: Comparison of different approaches

In Figure 8.1, it is shown the evolution of the error over time of these three approaches.

This error is obtained as the root mean squared difference between the position output after every filter update step and the ground truth (i.e., Leica) position.

It can be observed that, although the range of error is not very large, the Highest Weighted Particle approach experiments changes due to its high dependence of the noise of the latter observations. In this aspect, some improvement can be found in the ML approach thought the best performance is reached by the MMSE approach.

Another advantage of the MMSE output is that we can also extract essential information about the uncertainty of the solution. For the particular case of this modified MMSE for railway localization, the uncertainty of the solution can be split into two terms: hypothesis uncertainty and position uncertainty. Hypothesis uncertainty will give us information about how sure the filter is in his discrete path hypothesis estimation. The position uncertainty is the continuous uni-dimensional uncertainty along the chosen path hypothesis. This values can also be easily calculated from the particles (Figure 8.2).

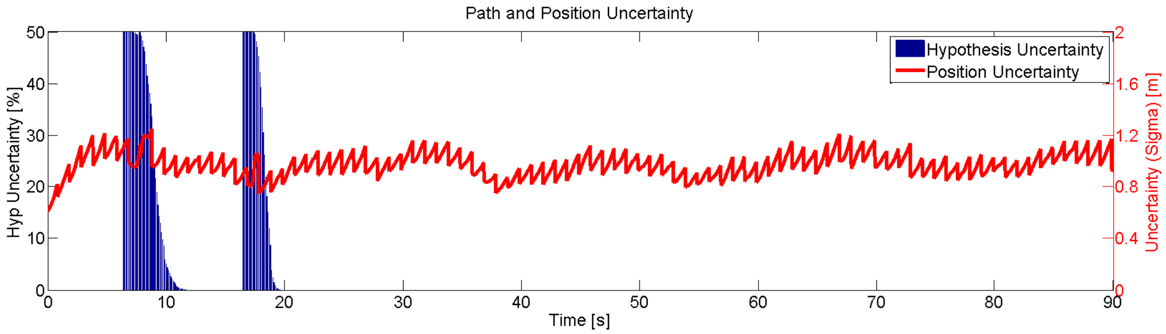


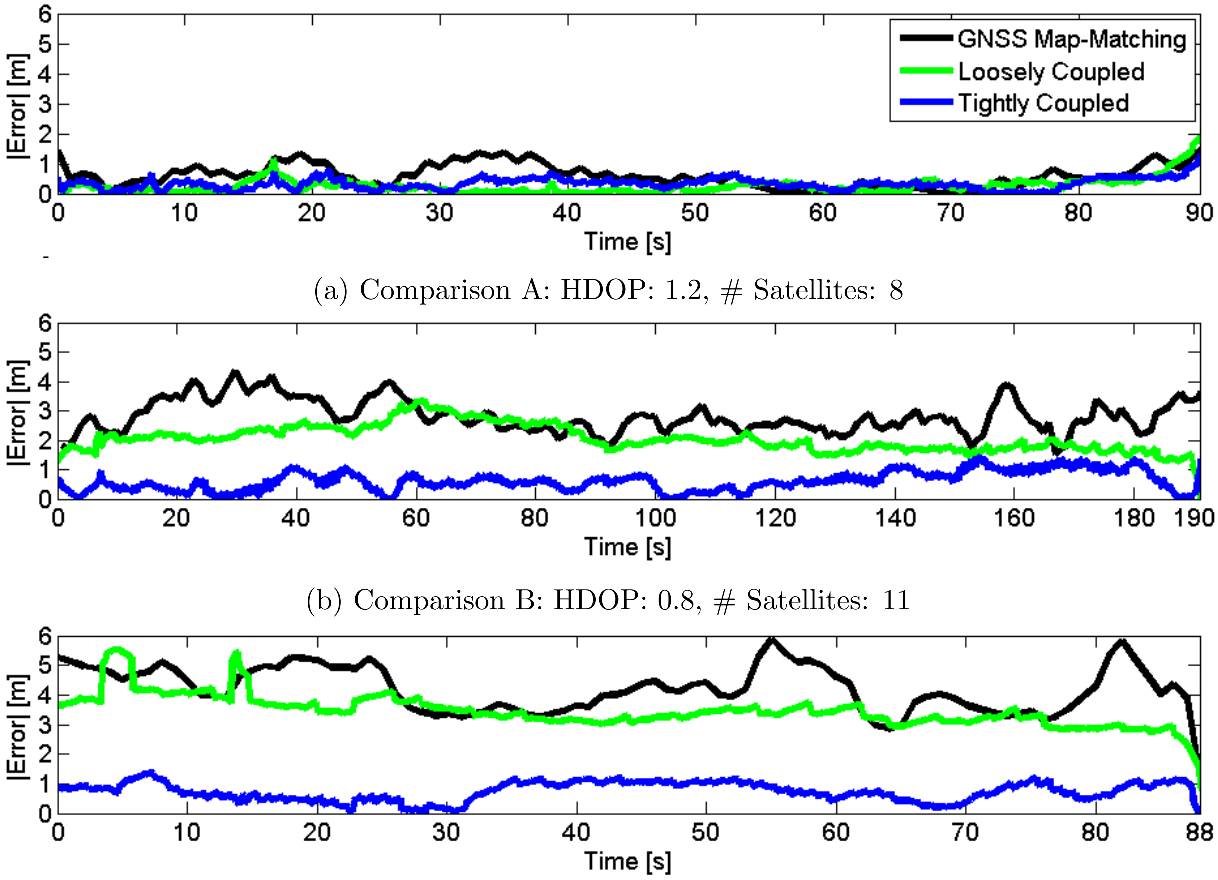
Figure 8.2: MMSE output: hypothesis and position uncertainty

In Figure 8.2, we can see both uncertainties represented over time. The bars in blue represent the path hypothesis uncertainty related to the left axis. During most of the time of the run, there is no ambiguity about the path hypothesis (we have 0%). Thus, we can identify very easily when the train reaches the two switches of this run by looking at the moment when the hypothesis uncertainty grows to 50 %. The reason is that when the cloud of particles reaches a switch, the particles spread with the same probability to both branches. Then, when gainful observation from IMU and Doppler arrives, this uncertainty decreases because the particles in the correct branch get higher weight. This path uncertainty is therefore a very important and useful indicator, because informs us when the output position can not be trusted yet. In some situations, it can also happen that two hypothesis survive over time, this indicator would give us information about that and the application user will have to take it into account.

The position uncertainty is directly related to the width of the cloud of particles after resampling. If everything works properly, this value should be of the same order of the error, so it gives us information about the along track range where the train can be.

8.2 Loosely vs Tightly Coupled Integration

In this section, a comparison between the performance of the loosely and tightly coupled approach for three different situations have been evaluated. The scenario and the different paths the vehicle took were shown in Figure 7.4, where there are two switches represented in yellow and thus three final possibilities. Both loose and tight approaches solve the junctions correctly in the three runs.



(a) Comparison A: HDOP: 1.2, # Satellites: 8

(b) Comparison B: HDOP: 0.8, # Satellites: 11

(c) Comparison C: HDOP: 1.3, # Satellites: 9

Figure 8.3: Loosely vs tightly coupled integration comparison

Results are shown in Figure 8.3, where we can see the error of:

- **GNSS Map-Matching:** In black, we find the solution provided by the receiver and then geometrically map-matched to the tracks. This process have been done taking into account only the correct tracks the train was railing, by knowing the route. This error therefore represents the along-track deviation of the GNSS position solution.
- **Loosely coupled:** Represented in green, we find the filter output error for the loosely coupled approach. This approach has been configured to attain the best performance as possible.

- **Tightly coupled:** The output of the approach developed in this thesis is drawn by a blue line.

We can see in Figure 8.3a, that when the error of the PVT solution of the GNSS is low, both approaches perform very similar. However, as the quality of the solution deteriorates, as in Figure 8.3b and in Figure 8.3c, the loosely coupled approach losses more performance than the tightly coupled integration. This effect can be also observed in Figure 8.4, where the mean of the rms errors for all the approaches and comparison are shown.

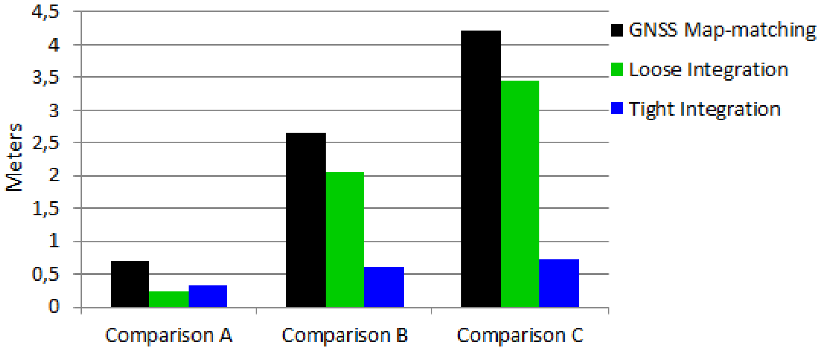


Figure 8.4: Loose vs tight coupled comparison: mean error

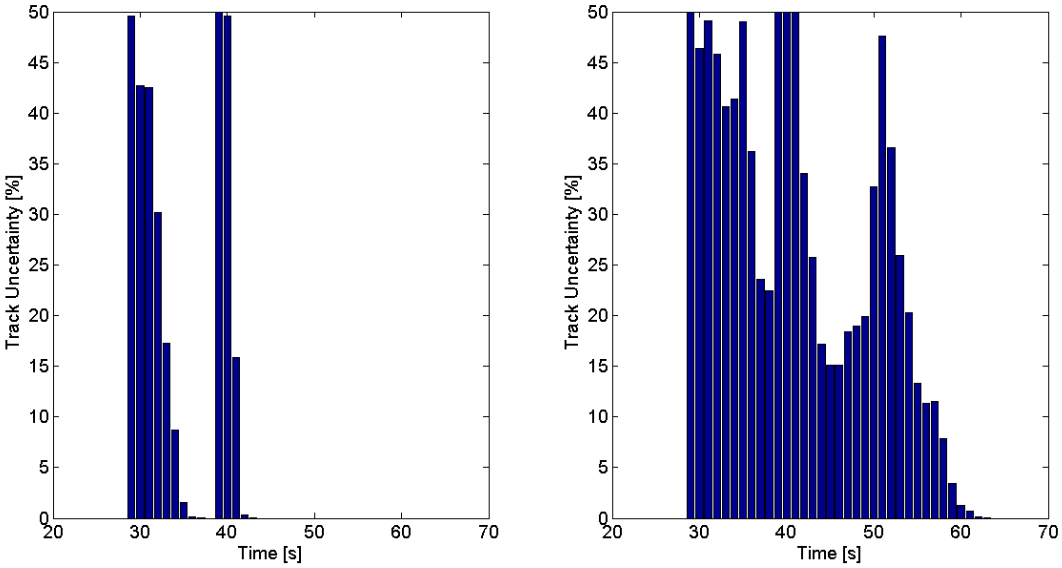
In the different plots of Figure 8.3 has also been written the average values of the Horizontal Dilution Of Precision (HDOP) and the number of visible satellites to clarify that the quality of the satellite degenerates because there are neither few visible satellites nor a disadvantageous satellite constellation: The estimated solution is poorer because of the quality of the satellites that contributes to the solution in that precise moment.

The loose approach performs worse because it only weights the particles regarding the latitude and the longitude of the PVT solution, so if the accuracy of this solution degenerates, this error is transferred very fast to the algorithm. However, in the tightly coupled integration the filter is weighted by N likelihoods, where N is the number of visible satellites. If there is one out of N satellites corrupting the PVT solution of the receiver, there will be still $N-1$ likelihoods that provide correct information and the effect will be less critical: The outlier satellite is actually being filtered.

8.3 Importance of Doppler

In absolute positioning algorithms, Doppler measurements are used to compute the vectorial velocity of the user. It contains also relevant information about the speed and the heading of the movement of the receiver for our algorithm. By intuition, it should thus help in solving a junction situation because it could discriminate those hypothesis that take the wrong heading direction. To prove this, a test of the filter with only pseudorange and Doppler measurements was carried out. The results proved that the combined pseudorange and Doppler approach can solve switch situations where the stand-alone pseudorange method is not able to.

However, it has been observed that as the number of visible satellites is reduced, the relevant information about the orientation of the movement that Doppler measurements can provide, decreases. Reaching to a point where the system is not capable of estimating properly the correct track.



(a) Track uncertainty with 8 satellites (b) Track uncertainty with 5 satellites

Figure 8.5: Track uncertainty in combined pseudorange and Doppler navigation

In Figure 8.5, we can see this degeneracy problem by observing the evolution of the track hypothesis uncertainty over time. Both graphs show the path uncertainty when the vehicle passes over two switches. In Figure 8.5a, 8 Doppler measurements are used to perfectly detect and solve both junctions. On the contrary, in Figure 8.5b, the filter resolved the track ambiguity but with higher uncertainty with only 5 Doppler contributions. Therefore, the inertial sensor plays always an essential role to guarantee a reliable track selective algorithm. And Doppler measurements will contribute to the robustness and accuracy of the system performance.

In Figure 8.6, we analyze the precisions obtained if Doppler measurements are used in the tightly coupled integration. In the first part of the graph, as the vehicle is stopped, the performance is similar for both approaches. Then, after second 60, when the vehicle has passed the junctions and is moving, we found a difference in the order of meters in the accuracy obtained. Doppler contribution is therefore important to minimize the effect of inertial errors, improving the performance of the output.

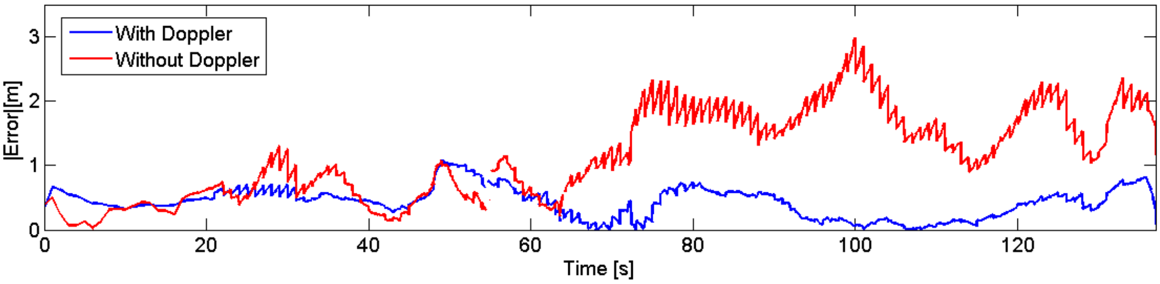


Figure 8.6: Error comparison with/without Doppler measurement

8.4 Few Visible Satellites

The great advantage of using directly the raw data coming from the receivers is that we can still include GNSS information in the filter when less than 4 satellites are visible. In this section, the performance of the system is evaluated with 3, 2 and 1 satellites.

Since the number of satellites is not the only important parameter that must be taken into account for any satellite based localization, for each number of satellites two different geometrical constellations have been chosen. Besides, although it is not possible to guarantee that there is not any other side effect hindering the measurements, signals with a high carrier-to-noise ratio (C/N_0) have been selected to reduce the probability of having a corrupted measurement.

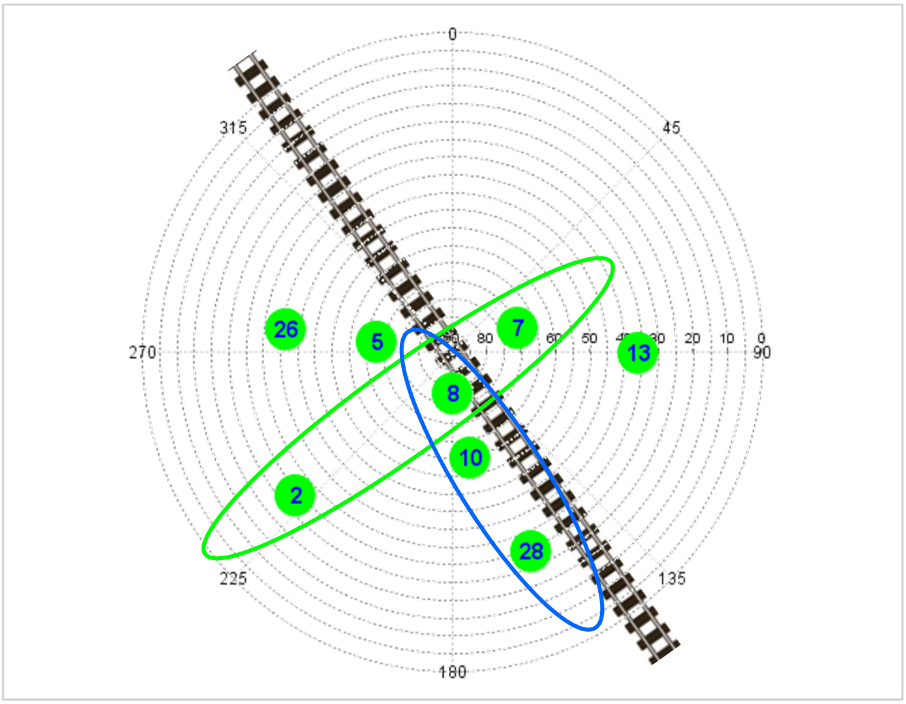


Figure 8.7: Satellite constellation in test scenario

The chosen constellations contribute orthogonally to solve the along track and the cross track position. That is, they are complementary in terms of CDOP and LDOP. In Figure 8.7, we can see a skyplot representation showing the elevation and azimuth of satellites positions for this scenario. It is also possible to see, rounded in green and blue, the chosen group of space vehicles for the analysis carried out.

8.4.1 Three Satellites

First, the performance of the filter is analyzed with the contribution of only three visible satellites. We can see from Figure 8.7, that the two groups of satellites selected can be seen as geometrically orthogonal with respect to the track. More formally, the values of CDOP and LDOP, shown in Figure 8.8, expose this complementarity (Satellites 2,7 and 8 have low CDOP and high LDOP while satellites 8,10 and 28 have high CDOP and low LDOP).

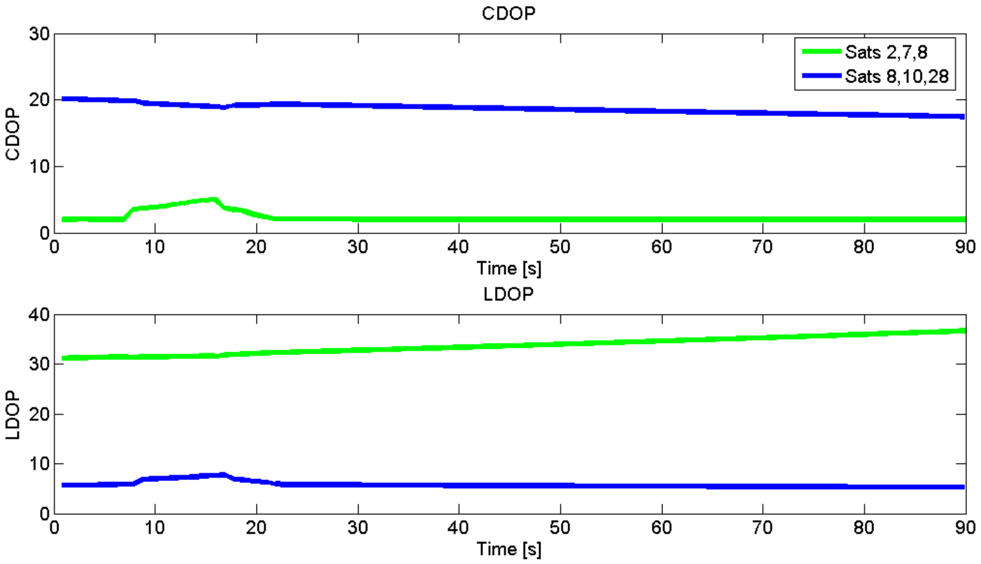


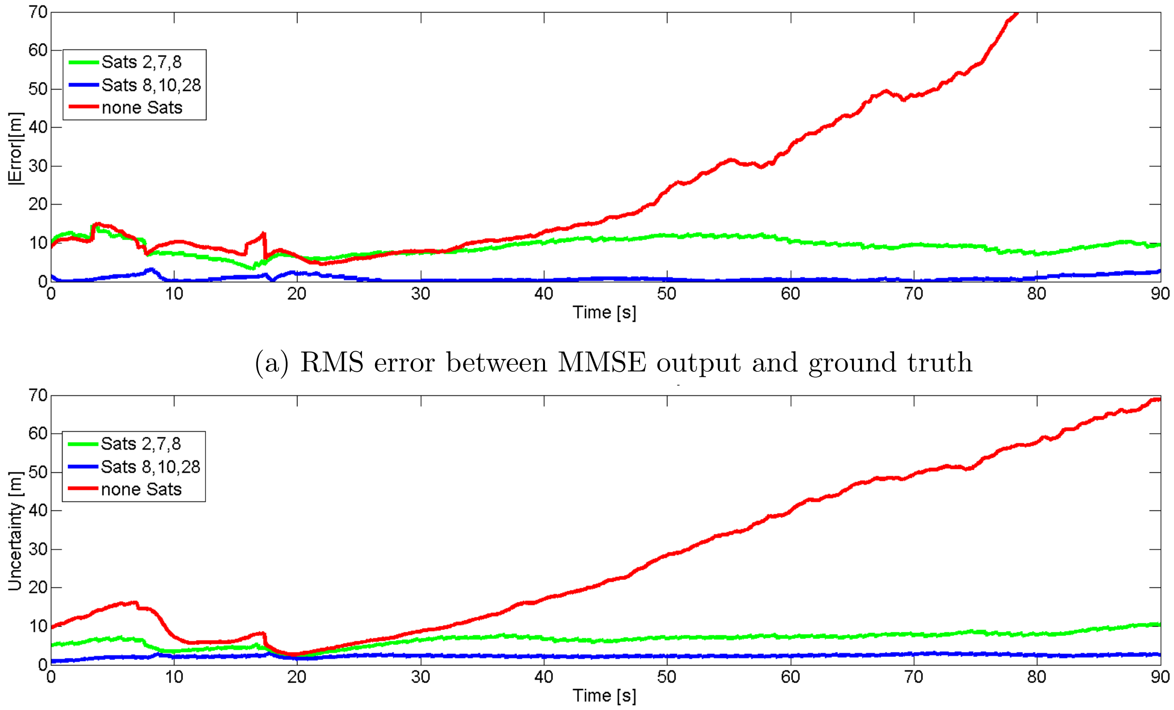
Figure 8.8: Three satellites scenario: LDOP and CDOP

In Table 8.1, we can also see the mean values of DOP for the two group of satellites chosen for this evaluation.

	CDOP	**LDOP**
Sats 2,7,8	2.24	34.51
Sats 8,10,28	18.64	5.76

Table 8.1: Average CDOP and LDOP values for 3 satellite comparison

The results of the test can be seen in Figure 8.9. In these graphs, three different lines have been plotted: the green and blue lines correspond to the different constellations of satellites chosen and the third line in red named ”none Sats” represents the solution of the filter without any GNSS update. This last representation would also correspond to the loosely coupled integration approach, because, as we have less than 4 satellites in view, there would be no GNSS update since the receiver would not be able to provide a PVT solution.



(a) RMS error between MMSE output and ground truth

(b) Position Uncertainty

Figure 8.9: Results: 3 satellites comparison

As seen from both graphs in Figure 8.9, the filter output without any GNSS update (red line) behaves as expected for a navigation with only inertial measurements. At 8 seconds and 16 seconds, the vehicle runs over switches, which represents a significant track change. Hence, the IMU detects these changes and helps to resolve the position of the train. However, after the second switch the accumulative errors of inertial sensors are transmitted to the solution as the train goes on. Note again that, the exponential growing error is typical of dead-reckoning navigation. Another important fact is that not only the error increases but the position uncertainty.

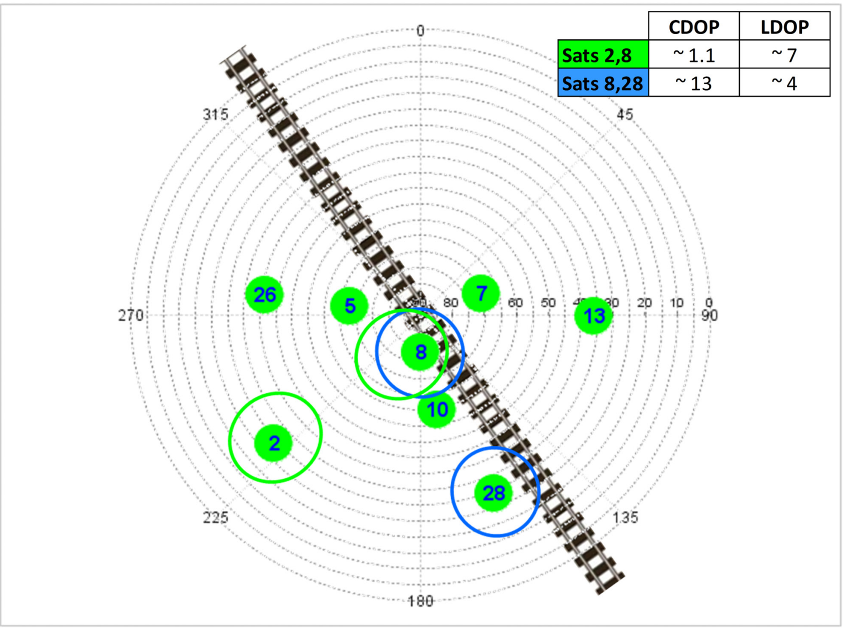
For the other two solutions, in spite of the fact that both of them keep stable over the whole run, differences in accuracy are observable. The constellation that had better LDOP than CDOP (blue line) has an accuracy comparable to the output of the filter with all the visible satellites (Figure 8.1). On the contrary, the solution with better CDOP than LDOP shows less accuracy. In Table 8.2, we can see the differences of accuracy for the two cases by observing the mean and maximum error obtained during this test.

	Mean error (rms)	Max error (rms)
Sats 2,7,8	8.37 m.	12.71 m.
Sats 8,10,28	0.72 m.	2.83 m.

Table 8.2: Results 3 satellites constellation

8.4.2 Two Satellites

In order to analyze the system with only 2 satellites in view, we will remove one of the satellites from the two constellations considered in the previous section. We can see in Figure 8.10, the selected group of satellites and the computed CDOP and LDOP during the run. It is again verifiable that both groups present different values for the along and cross track resolution.

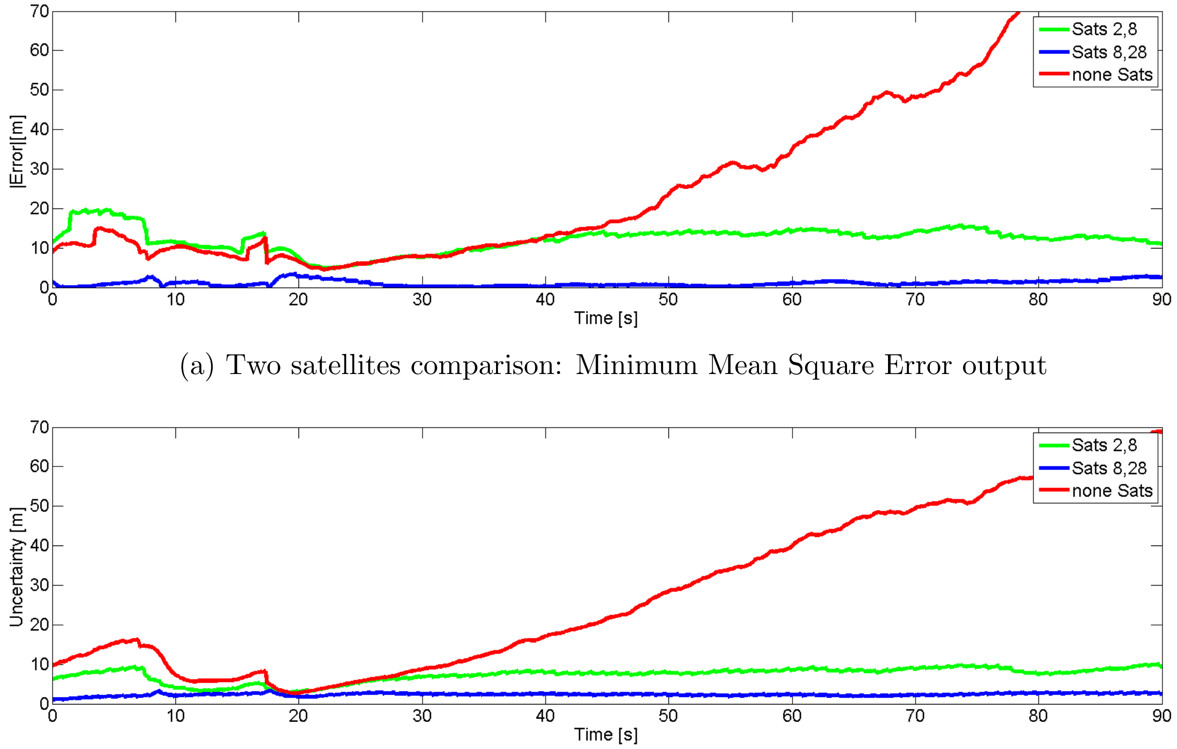


	CDOP	LDOP
Sats 2,8	~1.1	~7
Sats 8,28	~13	~4

Figure 8.10: Test constellations: 2 satellites

The error evolution and position uncertainty obtained are represented in Figure 8.11a and Figure 8.11b respectively. In those graphs, a similar behavior to that in three satellites situation is observed. The group of satellites with lower LDOP maintains more accuracy over the time.

If we observed the mean and maximum values of the error for this evaluation (Table 8.3), we notice a general increase compared to the 3 satellites results. This behaves as expected because we lose the contribution of one of the satellites. Most important is the fact that the uncertainty stays bounded and the system is still stable.



(a) Two satellites comparison: Minimum Mean Square Error output

(b) 2 Satellite Comparison: Position Uncertainty

Figure 8.11

	Mean error (rms)	**Max error (rms)**
Sats 2,8	11.68 m.	15.70 m.
Sats 8,28	0.99 m.	3.34 m.

Table 8.3: Results 2 satellites constellation

8.4.3 One Satellite

This time, two satellites that can be considered geometrically orthogonal related to the railway track have been chosen for this analysis. Particularly, satellites 2 and 28 shown in Figure 8.12, were selected.

The results are presented in the same fashion as in the previous analysis. We can see in Figure 8.13a that, the system running only with the satellite number 2, has no improvement respect to the system working only with the inertial unit. In fact, in this situation, it performs worse. On the other hand, the error keeps bounded if we use satellite number 28, the one that has better LDOP.

However, if we look the evolution of the position uncertainty (Figure 8.11b), we see that all the estimations are unstable. But although the uncertainty grows over time for every approach, the slope of the uncertainty for satellite 28 is lower than for the other two cases.

This all means that the one satellite situation is a very challenging scenario. And

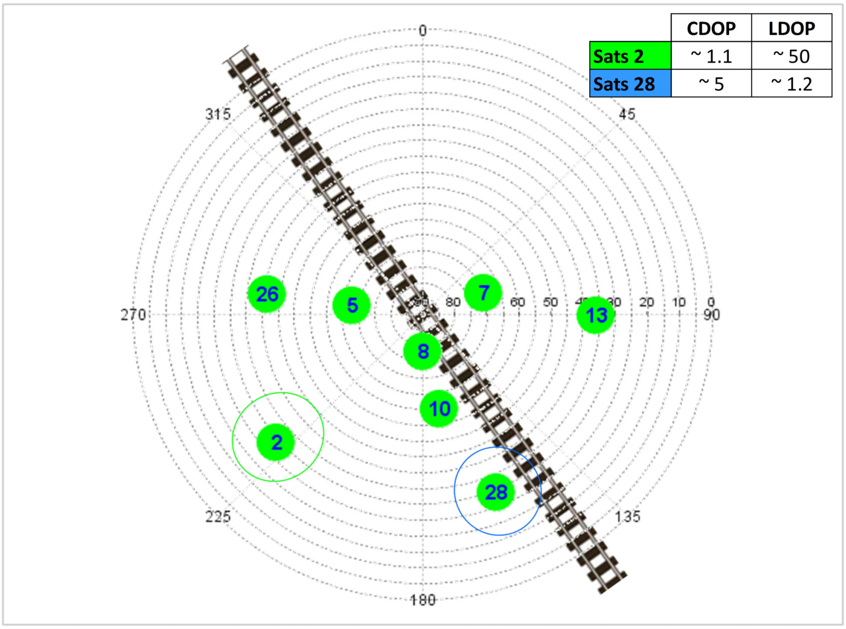
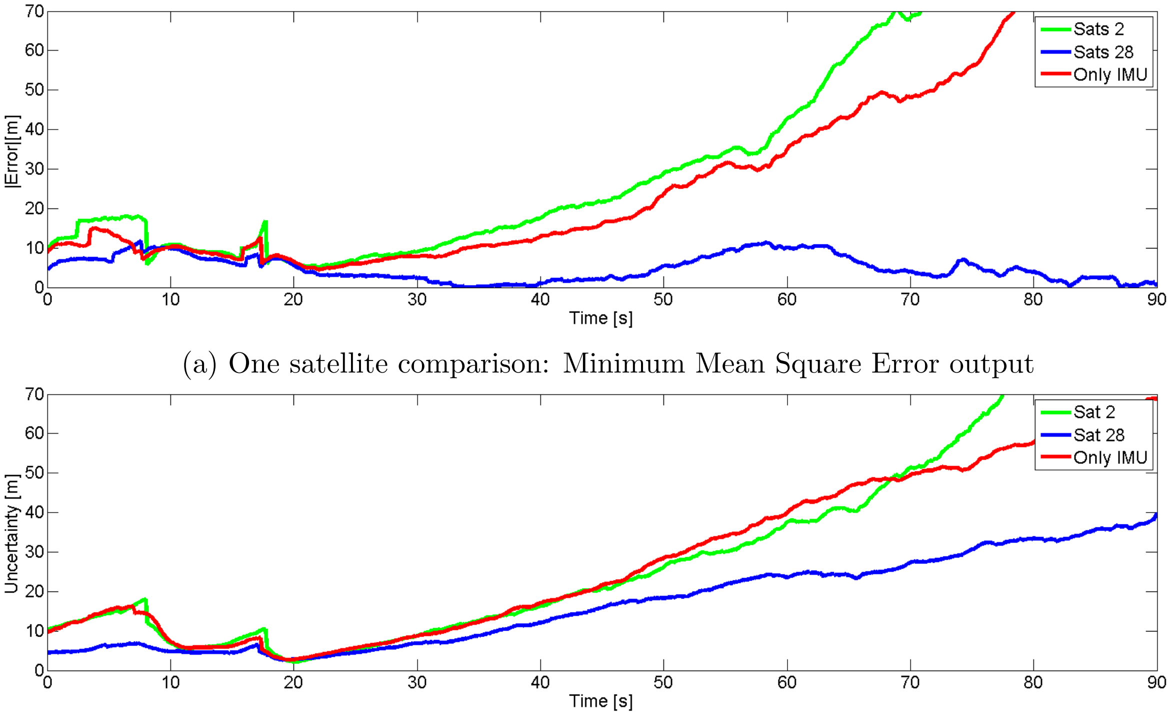


Figure 8.12: Test constellation: 1 satellite



(a) One satellite comparison: Minimum Mean Square Error output

(b) 1 Satellite Comparison: Position Uncertainty

Figure 8.13

although it is not possible to maintain a reliable localization, if the single satellite used is well geometrically located, we can keep longer the uncertainty low and the cloud of particles centered near the true position.

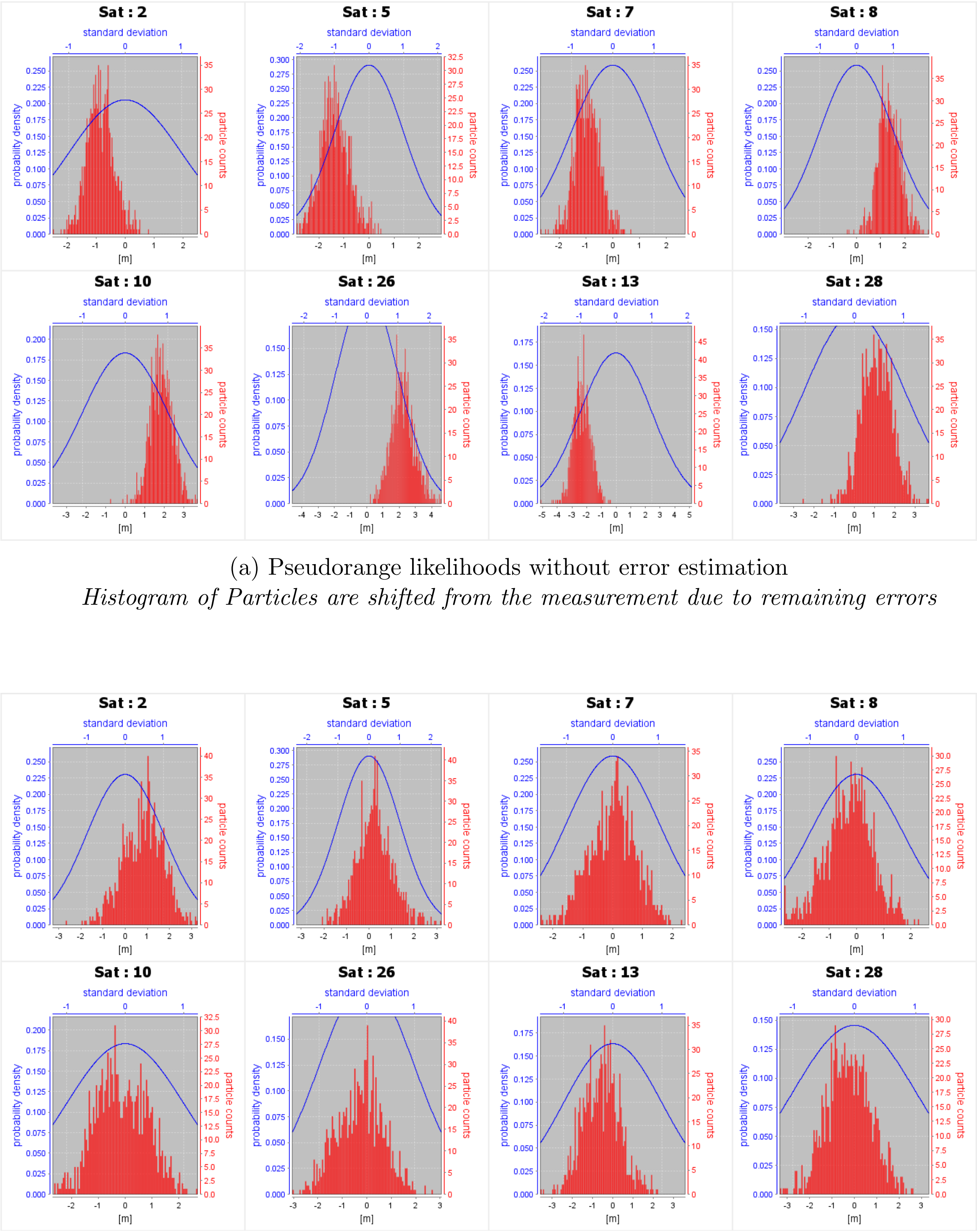
8.5 Effect of Error Estimation

During the different tests performed, no remarkable improvement in terms of precision has been observed when including the long-term correlated error estimation. This can be explained by the following facts:

- The error that is intuitively easy to correct is the one related to the cross track deviation, which therefore would not contribute to the remaining along track error.
- At the beginning of the tests, the train is stopped and therefore under multipath. The system tries to estimate the long-term correlated error under multipath which may distort it. When the train starts moving, the estimation is performing well but it may take longer to converge, which may not be reflected in the results.

However, during the testing of the error estimation, some evaluations were carried out to determine the benefits of this estimation. First, although no reduction of the position error could be shown, the estimation was well performed at the end of the tests. In Figure 8.14a, we can see the aspect of each pseudorange likelihood without the error estimation. The red bars represent the histogram of the predicted pseudorange of particles and the blue curve is the Gaussian likelihood function centered at the measured pseudorange. We can see how the histogram of the particles are not centered with the likelihood. This constant bias difference corresponds to the remaining error in the pseudorange. By contrast, in Figure 8.14b, after estimating the error, we can see how we reach to a centered distribution for every satellite. This is thus the proof that the estimation works.

Second, in a context where a switch is not perfectly solved (e.g., weak signal from IMU due to a slow speed of the vehicle nearby a station), both hypothesis survive and we reach to a parallel situation with two hypothesis. Then, the information from pseudoranges will determine which hypothesis die at a certain moment. Due to the long-term errors of the pseudoranges it could be the right one. In this situation, having estimated the error before facing the junction will make the cross information of the pseudoranges after the switch more accurate. And therefore, the estimation could help solving the switch and preventing from a failure.



(a) Pseudorange likelihoods without error estimation
Histogram of Particles are shifted from the measurement due to remaining errors

(b) Pseudorange likelihoods with error estimation
Histogram of Particles centered thanks to error estimation

Figure 8.14

Chapter 9

Achievements, Conclusions and Outlook

9.1 Achievements

A fully operational localization approach for railway environments has been created. The implemented system is able to integrate data from an inertial sensor, a GNSS receiver and a digital map to provide a position solution. The algorithm has been tested in different scenarios and challenging visibility conditions to prove its suitability and performance.

The particular tasks that have been achieved during this thesis work are shown in Figure 9.1:

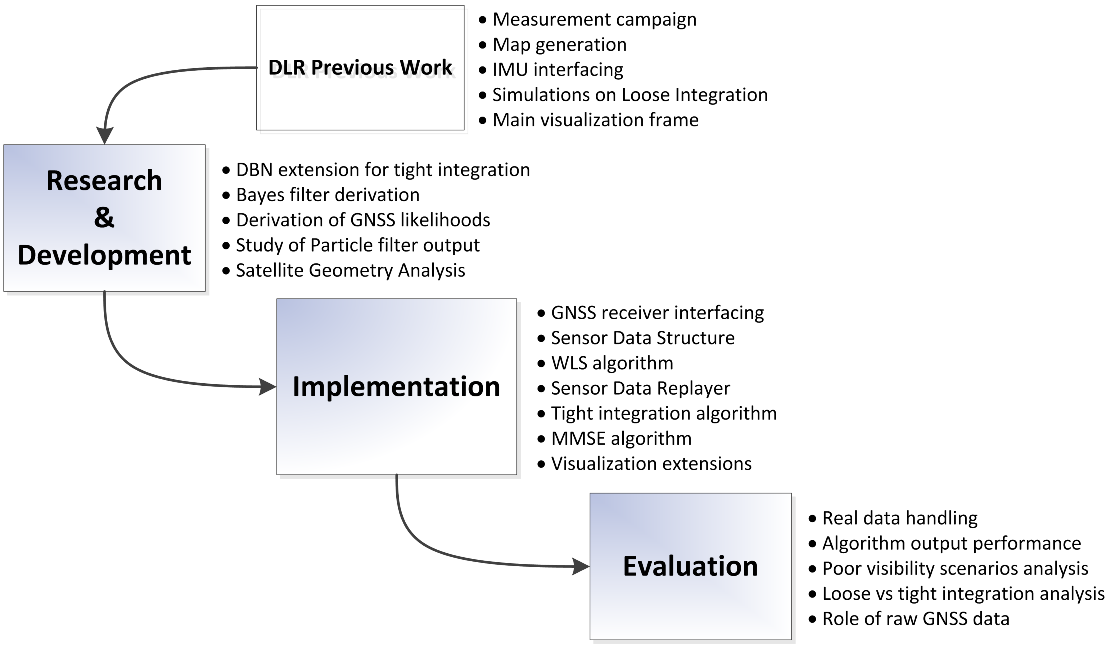

DLR Previous Work
• Measurement campaign
• Map generation
• IMU interfacing
• Simulations on Loose Integration
• Main visualization frame
Research & Development
• DBN extension for tight integration
• Bayes filter derivation
• Derivation of GNSS likelihoods
• Study of Particle filter output
• Satellite Geometry Analysis
Implementation
• GNSS receiver interfacing
• Sensor Data Structure
• WLS algorithm
• Sensor Data Replayer
• Tight integration algorithm
• MMSE algorithm
• Visualization extensions
Evaluation
• Real data handling
• Algorithm output performance
• Poor visibility scenarios analysis
• Loose vs tight integration analysis
• Role of raw GNSS data

Figure 9.1: Scheme of achieved tasks

Among all these essential tasks, we can point out as the most relevants:

- Theoretical Bayesian extension and Bayesian filter derivation.
- Tightly coupled integration with particle filter.
- Geometric analysis for railway environments.
- Receiver interfacing and data replay mode.

9.2 Conclusions

From the different tests performed and results obtained in Chapter 8, we can extract the following conclusions:

First, the integration between GNSS, IMU and a track map have been consolidated to be a suitable solution for a reliable navigation along a railway map. Precisions below one meter have been obtained and the track selection have been always decided correctly in normal visibility scenarios, pointing out the achievable accuracy of this integration scheme and its potential. It is suggested therefore, that the approach may be suitable for railway safety-of-life (SoL) systems.

Second, results show that reliable localization may still be reachable with only two or three visible GNSS satellites, which broaden the navigation performance to critical visibility scenarios and ensure thus, a highly robust localization system.

Third, localization with only one visible satellite is a very challenging situation and the system tends to perform similarly to a stand-alone inertial navigation. Moreover, it has been shown that if the GNSS contribution comes from a satellite which is in a beneficial position related to the track, the navigation filter is able to reduce the uncertainty incrementation and it can keep the estimation output near the truth for a longer time. This is especially interesting because it would permit the system to recover faster or even inadvertently from a temporal gap of signal blockage.

The comparison between loose and tight integration has shown that even in normal visibility scenarios, the tightly coupled integration is more robust to the presence of outlier satellite signals and to the non-modeled errors.

In all the evaluations carried out in this work, the geometry factor of the satellites was of great importance. We have confirmed, through a dilution of precision analysis in the train frame, that the along-track dilution of precision (LDOP) is a critical factor in railway localization. This fact reveals that the main contribution of GNSS in the integration process is to solve the along-track position, while, the inertial sensor is the main contributor to the track selective localization after junctions. Hence, GNSS must guarantee the presence of hypothesis (particles) in the switch surroundings so that the junctions can be detected accordingly to IMU measurements.

The extension of Doppler measurements into the filter has shown improvements in the accuracy of the localization process in motion. Furthermore, Doppler also contributes to solve the correct track after switches, making this process more robust in general and especially when the IMU signals are not so significant (i.e., low speed of the train).

It has been proved that the long-term correlated error of pseudoranges can be successfully estimated in this approach. Although it has not shown improvements in terms of precision, there exist benefits in robustness for a railway localization, such as after junctions.

Finally, for this specific localization problem, it has been seen that the mixed MMSE particle filter output approach, proposed and implemented in this thesis (Section 3.5.4), is the best final solution estimation of the navigation particle filter.

9.3 Outlook

Some suggestions for improvements and future research directions are:

- *IMU error estimation:* Inertial measurement errors have been shown to be one of the main source of error contribution of the proposed tight approach. It can be expected that the presence of other measurements such as pseudoranges and Doppler, as well as the map information will offer a highly beneficial context to estimate the IMU errors.

- *Context-aided navigation:* Detecting the motion/stop state of the train could be helpful to mitigate GNSS multipath errors when the train is stopped or to improve the transition models.

- *Exploitation of magnetic measurements:* The Xsens device is able to provide with 3D magnetic measurements. Therefore, it may be also possible to take advantage of this additional information.

Additionally, further analysis or extensions of this thesis approach should address:

- *Analysis with commercial receiver data:* Although the Septentrio GNSS receiver used for the testing of this work is able to provide precise measurements, it has some drawbacks in terms of time-to-first-fix (TTFF). A further analysis with consumer grade receivers, for example u-blox, would permit to evaluate the receiver performance contribution.

- *Analysis with passenger train measurements:* The vehicle used to test this thesis approach has some dynamics differences with a real train. We expect the train acceleration and suspension system to be more stable than the vehicle's.

- *Second sensor set:* Having a second GNSS receiver and a second IMU sensor in the rear part of the train would permit to continue receiving GNSS signals when the train enters a tunnel or in case of local signal blockage. Additionally, it would help to ensure track selection in switches and determine when the train leaves some area.

Bibliography

[1] "Eurostat." http://epp.eurostat.ec.europa.eu/, November 2012.

[2] J. Poliak, J. Marais, F. Hänsel, U. Becker, and E. Schnieder, "Methods and Tools for the Certification of GALILEO Localisation for Railway Applications," May 2008.

[3] T. Strang, M. Meyer zu Hörste, and X. Gu, "A Railway Collision Avoidance System Exploiting Ad-hoc Inter-vehicle Communications and Galileo," October 2006.

[4] "Railway Collision Avoidance System (RCAS)." http://collision-avoidance.org.

[5] B. W. Parkinson and J. J. Spilker Jr., *Global Positioning System: Theory and Applications.* American Institute of Aeronautics and Astronautics, 1996.

[6] E. D. Kaplan, *Understanding GPS, Principles and Applications.* Artech House, 2005.

[7] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance.* Ganga-Jamuna Press, 2006.

[8] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. Holdt Jensen, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach.* Applied and Numerical Harmonic Analysis, Birkhäuser Boston, 2007.

[9] J. Zhang, *Precise Velocity and Acceleration Determination Using a Standalone GPS Receiver in Real Time.* PhD thesis, School of Mathematical and Geospatial Sciences, Royal Melbourne Institute of Technology, Australia, 2007.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics.* Intelligent Robotics and Autonomous Agents, The MIT Press, 2005.

[11] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques.* Adaptative Computation and Machine Learning, The MIT Press, 2010.

[12] A. Doucet, N. de Freitas, and G. Neil, "An Introduction to Sequential Monte Carlo Methods," in *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.

[13] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions*, vol. 50, pp. 174–188, February 2002.

[14] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[15] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The Unscented Particle Filter," in *Advances in Neural Information Processing Systems (NIPS13)* (T. G. D. T. K. Leen and V. Tresp, eds.), MIT Press, December 2000.

[16] S. Gleason and D. Gebre-Egziabher, *GNSS Applications and Methods.* Artech House, 2009.

[17] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology, 2nd ed.* Institution of Electrical Engineers, 2004.

[18] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C*, vol. 15, p. 312–328, 2007.

[19] N. R. Velaga, M. A. Quddus, and A. L. Bristow, "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems," *Transportation Research Part C: Emerging Technologies*, vol. 17(6), pp. 672–683, 2009.

[20] K. Gerlach and C. Rahmig, "Multi-Hypothesis Based Map-Matching Algorithm for Precise Train Positioning," July 2009.

[21] P. Bonnifait, J. Laneurit, C. Fouque, and G. Dherbomez, "Multi-hypothesis map-matching using particle filtering," January 2010.

[22] A. Ufuk Peker, O. Tosun, and T. Acarman, "Particle Filter Vehicle Localization and Map-Matching Using Map Topology," June 2011.

[23] O. Heirich, P. Robertson, A. Cardalda Garcia, T. Strang, and A. Lehner, "Probabilistic localization method for trains," June 2012.

[24] C. Fouque and P. Bonnifait, "Matching Raw GPS Measurements on a Navigable Map Without Computing a Global Position," June 2012.

[25] O. Heirich, P. Robertson, A. Cardalda Garcia, and T. Strang, "Bayesian Train Localization Method Extended by 3D Geometric Railway Track Observations from Inertial Sensors," 2012.

[26] R. G. Brown and P. Y. Hwang, *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons, 1997.

[27] F. Zeng, W. Dai, J. Zhu, and X. Wang, "Single-point Positioning with the Pseudorange of Single-frequency GPS Considering the Stochastic Model," *Pacific Science Review*, vol. 10(3), pp. 274–278, 2008.

[28] A. Wieser, M. Gaggl, and H. Hartinger, "Improved Positioning Accuracy with High-Sensitivity GNSS Receivers and SNR Aided Integrity Monitoring of Pseudo-Range Observations," September 2005.

[29] M. A. Sturza, "GPS Navigation Using Three Satellites and Precise Clock," *Litton Aero Products, Canoga Park, Ca.*, 1983.

[30] "National Marine Electronics Association." http://home.mira.net/ gnb/gps/nmea.html.

[31] A. R. Corporation, *NAVSTAR GPS Space Segment/Navigation User Interfaces, Interface Specification, IS-GPS-200D (Public Release Version).* ARINC Research Corporation, Fountain Valley, CA, 2004.

[32] J. A. Klobuchar, "Ionospheric Time-Delay Algorithms for Single-Frequency GPS Users," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 23(3), pp. 325–331, May 1987.

[33] u-blox AG, *u-blox 6 Receiver Description Including Protocol Specification*, April 2011.

[34] W. Gurtner, *RINEX: The Receiver Independent Exchange Format Version 2.11.* Astronomical Institute University of Berne, 2007.

[35] "WindowBuilder." http://www.eclipse.org/windowbuilder/.

[36] "JFreeChart." http://www.jfree.org/jfreechart/.

[37] M. Meyer zu Hörste and G. Katrin, "Precision and Availability Evaluation for Road and Rail Localisation Platforms," 2008.

[38] Septentrio nv/sa, *Command Line Interface Reference Guide. Applicable to AsteRx2 2.1*, September 2010.

[39] Xsens Technologies B.V., *MTi and MTx User Manual and Technical Documentation*, March 2006.

[40] "NATO (North Atlantic Treaty Organization) (1993) Standardization Agreement (STANAG)." Doc. 4294 EL (Edition 1), Appendix 6 to Annex A, pp. A-6-34,A-6-37.

[41] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide.* Object Technology Series, Addison Wesley, 2005.

[42] "eUML2." http://www.soyatec.com/euml2/.

Appendix A

Satellite Position and Velocity Computation

This appendix shows the procedure for the position and velocity computation of a satellite. Although there are many references in the literature explaining the process to obtain the satellite position, our intent is to group both satellite and velocity computations in one algorithm.

In Figure A.1, we can see the different orbital parameters involved in the position and velocity computation. For more information about the ephemeris parameters we suggest the reader to consult [31].

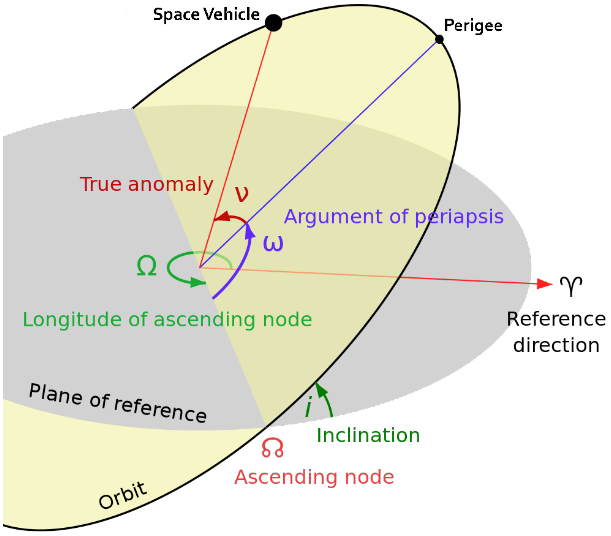


Figure A.1: Orbital Elements
Adapted from Lasunncty (2007), Wikimedia Commons, Public domain.

The procedure to compute the satellite position is presented in Algorithm 7.

Algorithm 7 Satellite position computation

```
Input: - Transmit time $t$
       - Satellite Ephemeris
$a = (\sqrt{a})^2$                                        ▷ Restore semi-major axis
$t_k = t - t_{oe}$                                        ▷ Find time difference
if $t' > 302400$ then                                     ▷ Correct for week crossover
    $t_k = t_k - 604800$
else if $t_k < -302400$ then
    $t_k = t_k + 604800$
end if
$n = \sqrt{GM/a^3} + \Delta n$                            ▷ Mean motion
$\mu = \mu_0 + n \cdot t_k$                               ▷ Mean anomaly
$\mu = \mu \ (mod\ 2\pi)$                                 ▷ Bound $\mu$ between $[0,2\pi]$
$E = \mu$                                                 ▷ Initial guess of eccentric anomaly
for $j = 1 : 10$ do                                       ▷ Iteratively compute eccentric anomaly
    $E_{old} = E$
    $E = \mu + e \cdot \sin E$
    $\Delta E = E - E_{old}(mod\ 2\pi)$
    if $|\Delta E| < 10^{-13}$ then                       ▷ If required precision, stop
        break
    end if
end for
$E = E(mod\ 2\pi)$
$\nu = \arctan \frac{\sqrt{1-e^2}\sin E}{\cos E - e}$     ▷ Calculate the true anomaly
$\Omega_k = \Omega_0 + (\dot{\Omega} - \omega_e)t_k - \omega_e t_{oe}$   ▷ Longitude for ascending node
$\Omega_k = \Omega_k(mod\ 2\pi)$
$\Phi = (\nu + \omega)(mod\ 2\pi)$

$\omega_k = \omega + \nu + C_{\omega c}\cos 2\Phi + C_{\omega s}\sin 2\Phi$   ▷ Correct argument of latitude
$r_k = a(1 - ecosE) + C_{rc}\cos 2\Phi + C_{rs}\sin 2\Phi$   ▷ Correct radial distance
$i_k = i_0 + \dot{i} + C_{ic}\cos 2\Phi + C_{is}\sin 2\Phi$   ▷ Correct inclination

$x'_k = r_k \cos \omega_k$                                ▷ Coordinate x in orbital plane
$y'_k = r_k \sin \omega_k$                                ▷ Coordinate y in orbital plane

$x = x'_k \cos \Omega_k - y'_k \cos i_k \sin \Omega_k$    ▷ Compute Satellite x coordinate
$y = x'_k \sin \Omega_k + y'_k \cos i_k \cos \Omega_k$    ▷ Compute Satellite y coordinate
$z = y'_k \sin i_k$                                       ▷ Compute Satellite z coordinate
```

Likewise, the procedure to obtain the satellite velocity is shown in Algorithm 8. One should notice that this algorithm is computed as an extension of the satellite position one by differentiating the different parameters over time, so it is highly recommended to implement both algorithms together as done for this thesis implementation.

Algorithm 8 Satellite velocity computation extension

```
$\dot{\mu} = n$    ▷ Rate of Mean anomaly
$\dot{E} = \frac{\dot{\mu}}{1-e\cos E}$    ▷ Eccentric anomaly rate
$\dot{\nu} = \frac{\sin E\dot{E}(1+e\cos\nu)}{(1-\cos Ee)\sin\nu}$    ▷ Rate of True anomaly
$\dot{\Phi} = \dot{\nu}$
$\dot{\Omega}_k = \dot{\Omega} - \dot{\omega}_e$

$\dot{\omega}_k = \dot{\Phi} + 2[C_{\omega s}\cos(2\Phi) - C_{\omega c}\sin(2\Phi)]\dot{\Phi}$    ▷ Argument of perigee rate
$\dot{r}_k = Ae\sin E\dot{E} + 2[C_{rs}\cos(2\Phi) - C_{rc}\sin(2\Phi)]\dot{\Phi}$    ▷ Radial distance rate
$\dot{i}_k = \dot{i} + 2[C_{is}\cos(2\Phi) - C_{ic}\sin(2\Phi)]\dot{\Phi}$    ▷ Inclination rate

$\dot{x}'_k = \dot{r}_k\cos\omega_k - r_k\sin\omega_k\dot{\omega}_k$
$\dot{y}'_k = \dot{r}_k\sin\omega_k + r_k\cos\omega_k\dot{\omega}_k$

$\dot{x} = \dot{x}'_k\cos\Omega_k - \dot{y}'_k\cos i_k\sin\Omega_k + y'_k\sin i_k\sin\Omega_k\dot{i}_k - y_k\dot{\Omega}_k$    ▷ Satellite Velocity in x
$\dot{y} = \dot{x}'_k\sin\Omega_k + \dot{y}'_k\cos i_k\cos\Omega_k - y'_k\sin i_k\dot{i}_k\cos\Omega_k + x_k\dot{\Omega}_k$    ▷ Satellite Velocity in y
$\dot{z} = \dot{y}'_k\sin i_k + y'_k\cos i_k\dot{i}_k$    ▷ Satellite Velocity in z
```

Appendix B

GNSS Correction Models

In this appendix, we expose the different corrections that must be applied to the pseudorange and Doppler general observable equations (2.1)-(2.2). These corrections are essential in the implementation of the algorithm in Section 5.3.

B.1 Satellite Clock Correction

Satellites are constructed with a precise atomic clock inside that controls all the required timing operations of the space vehicle (SV), the signal and code generation. Despite the stability of these clocks, there may be differences between the onboard clock and GPS time. The control segment calculates those differences and provides the subframe 1 of the navigation message with some coefficients that allow the user to determine this offset. The coefficients are applied by the user following this polynomial [31]:

$$dt^s = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + \Delta t_r, \tag{B.1}$$

where

a_{f0} = clock bias [s],
a_{f1} = clock drift [s/s],
a_{f2} = aging (drift change rate) [s/s^2],
t_{oc} = clock data reference time [s],
t = current time epoch [s],
Δt_r = correction due to relativistic effects [s].

B.2 Relativistic Corrections

Einstein's general and special theories of relativity must be taken into account in satellite navigation. In particular, there are three cases that the user must considerate.

First, due to the eccentricity of the satellite orbits, the space vehicle travels with different speeds at the perigee and at the apogee of the orbit. Moreover, the gravitational potential is also different. Those effects causes the clock to run slower or faster depending of the position in the orbit. To account for this effect we include a correction term in the satellite clock offset determination of Equation (B.1) [31]:

$$\Delta t_r = Fe\sqrt{a}\sin E_k, \tag{B.2}$$

where e is the satellite orbital eccentricity, a is the semimajor axis of the orbit, E_K the eccentric anomaly and F is a constant with value $-4.442807633 \cdot 10^{-10}\ \frac{s}{\sqrt{m}}$ [31].

This effect also influence the Doppler observable and must be thus corrected. According to [9], it is possible to derive the rate of this relativistic effect as:

$$d\dot{R}_{eccentricity} = \frac{2GM}{c}\left(\frac{1}{a} - \frac{1}{||r^s||}\right), \tag{B.3}$$

where a is the semimajor axis of the orbit, r^s is the satellite position in ECEF and GM is the Earth's universal gravitational parameter $3.986005 \cdot 10^{14}$ $[km^3 s^{-2}]$. Notice that, in fact, for a circular orbit $a = ||r^s||$ this effect disappears.

The second effect that must be considered is the so called *Sagnac effect.* A relativistic error is introduced due to the rotation of the Earth during the time the signal is being transmitted. The correction for this error is typically applied by performing a rotation of the satellite position:

$$\begin{bmatrix} x^{s'} \\ y^{s'} \\ z^{s'} \end{bmatrix} = \begin{bmatrix} \cos\dot{\Omega}\tau & \sin\dot{\Omega}\tau & 0 \\ -\sin\dot{\Omega}\tau & \cos\dot{\Omega}\tau & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^s \\ y^s \\ z^s \end{bmatrix}, \tag{B.4}$$

where $[x^{s'}, y^{s'}, z^{s'}]$ is the corrected position of the satellite, $[x^s, y^s, z^s]$ the uncorrected one, τ the estimated travel time of the signal and $\dot{\Omega}$ the WGS 84 value of the Earth's rotation rate.

On the other hand, since it is also possible to express the Sagnac correction in terms of range deviation:

$$dR_{Sagnac} = \frac{\dot{\Omega}}{c}\left(y^s x_u - x^s y_u\right), \tag{B.5}$$

we can obtain the Sagnac correction for Doppler by differentiating (B.5) with respect to time:

$$d\dot{R}_{Sagnac} = \frac{\dot{\Omega}}{c}\left(\dot{x}_u y^s - \dot{y}_u x^s + x_u \dot{y}^s - y_u \dot{x}^s\right). \tag{B.6}$$

This relativistic effect may reach a deviation up to several mm/s so it was not finally applied in this thesis work.

Finally, due to the gravitational field of the Earth, the transmitted signal experiences a space-time curvature. This effect can cause up to 18.7 mm of positioning error [6] and therefore have been neglected for this work.

B.3 Ionospheric Correction

The propagation time of a GNSS signal through the ionosphere layer of the atmosphere depends upon its electron density since this zone is a partially ionized medium. For a single frequency user, the most extensively applied time delay determination process is based on Klobuchar's ionospheric model [32]. Indeed, the coefficients *alpha* (α_n) and *beta* (β_n) needed in this model for the delay time calculation are transmitted as part as the subframe 4 of the navigation message.

The different steps to compute the ionospheric correction are now summarized. For further details we invite the reader to consult [31] or [32]:

1. Calculate the Earth-centered angle (elevation El in semicircles):

$$\psi = \frac{0.0137}{El + 0.11} - 0.022 \quad [semicircles]. \tag{B.7}$$

2. Compute the subionospheric latitude:

$$\phi_l = \phi_u + \psi \cos Az \ [semicircles], \tag{B.8}$$

if $\phi_l > 0.416$, then $\phi_l = 0.416$. If $\phi_l < -0.416$, then $\phi_l = -0.416$.

3. Compute the subionospheric longitude:

$$\lambda_l = \lambda_u + \frac{\psi \sin Az}{\cos \phi_l} \quad [semicircles]. \tag{B.9}$$

4. Find the geomagnetic latitude:

$$\phi_m = \phi_l + 0.064 \cos(\lambda_l - 1.617) \quad [semicircles]. \tag{B.10}$$

5. Find the local time:

$$t = 43200\lambda_l + GPStime(sec), \tag{B.11}$$

if $t > 86400$, use $t = t - 86400$. If $t < 0$, add 86400.

6. Compute the slant factor:

$$F = 1.0 + 16.0 \cdot (0.53 - El)^3. \tag{B.12}$$

7. Compute the amplitude of ionospheric delay:

$$AMP = \sum_{n=0}^{3} \alpha_n \phi_m^n \quad [seconds], \tag{B.13}$$

if $AMP < 0$, then $AMP = 0$.

8. Compute the period of ionospheric delay:

$$PER = \sum_{n=0}^{3} \beta_n \phi_m^n, \tag{B.14}$$

   if $PER < 72000$, then $PER = 72000$.

9. Compute the phase of ionospheric delay:

$$X = \frac{2\pi(t - 50400)}{PER} \quad [radians]. \tag{B.15}$$

10. Finally, compute the ionospheric time delay:

$$dI_{L1} = \left\{ \begin{array}{ll} F \cdot \left[5 \cdot 10^{-9} + AMP \cdot \left(1 - \frac{X^2}{2} + \frac{X^4}{24}\right)\right] & ; |X| \leq 1.57 \\ F \cdot 5 \cdot 10^{-9} & ; |X| \geq 1.57 \end{array} \right. , \tag{B.16}$$

   where ϕ_u and λ_u are the latitude and longitude of the user respectively, El is the elevation angle of the satellite and Az the azimuth.

Likewise, by differentiating (B.16) with respect to time, we can obtain the ionospheric correction for Doppler measurement [9]:

$$d\dot{I}_{L1} = \left\{ \begin{array}{ll} F \cdot AMP \cdot \dot{X} \left[-X + \frac{X^3}{6}\right] & ; |X| \leq 1.57 \\ 0 & ; |X| \geq 1.57 \end{array} \right. , \tag{B.17}$$

where $\dot{X}$ is calculated as:

$$\dot{X} = \frac{dX}{dt} = \frac{2\pi}{PER}. \tag{B.18}$$

B.4 Tropospheric Correction

The effect of the troposphere on the GNSS signals turns into an additional delay of the signal traveling time from the satellite to the receiver. This delay depends on the air refractivity through temperature, pressure and humidity conditions. Besides, the altitude of the user and the elevation of the satellite are also important parameters.

The model implemented in this thesis follows the Standard Agreement (STANAG) [40] though a large number of other tropospheric models can be found in the literature.

Regarding Doppler measurement, tropospheric correction can be obtained from different mapping functions but, as it is widely accepted that the contribution of the troposphere to the Doppler measurement is very small, we have finally neglected this correction term in this work.

B.5 Error Magnitudes

An overview of the magnitudes is now given for the presented errors. This will help the reader to understand the impact of the corrections carried out. In Table B.1, we can see the approximate magnitude of the errors in pseudorange and Doppler measurements:

Error Term	**Pseudorange Magnitude**	**Doppler Magnitude**
Satellite Clock Correction	$\leq$ 300Km	Negligible
Orbit Eccentricity	$\leq$ 21 m.	Several cm/s
Sagnac Effect	$\leq$ 30 m.	Several mm/s
Gravitational Potential	$\leq$ 18.7 mm.	Sub-mm/s
Ionospheric Correction	$\sim$ 7 m. (Max. 45 m.)	cm/s level
Tropospheric Correction	[2.4 - 25] m.	Negligible
Receiver Clock	Unknown to be estimated	Unknown to be estimated

Table B.1: Main error sources and their approximate magnitudes

Appendix C

Code Diagrams

This appendix intends to provide a better overview of the most relevant Java® classes implemented during this thesis. In the following figures, the main classes of some parts of the code are represented in the Unified Modeling Language (UML®) class diagrams. UML® is a standardized partial graphical language that provides an easy way to represent the different relationships and structure of the agents in a software engineering project, as well as its behavioral evolution activity. Because of its flexibility, UML® model is widely used with object-oriented software in different analysis and code designing stages of a project [41].

Specifically for this work, only class diagrams have been used, where the different agents involved are shown by boxes with their names in the header part and the attributes and implemented methods in the area below. In this representation, classes appears with a **C** in the header and interfaces with an **I**.

It is important to notice the different relations that can exist:

- **Extension:** This is represented by an arrow with the tip in white and stands for *'extends from'*. This is the common way of representing the inheritance between classes.

- **Implementation:** It is used when a class *implements* an interface and is shown by a dashed arrow.

- **Association:** The dependency between classes is represented by a simple line or by a simple arrow if the dependency direction wants to be also shown. It is usually pointed here the cardinality of the association, which is written by a number or a range in the extremes of the line.

Figures C.1 to C.5 has been obtained by applying *reverse engineering* to the Java® frame. To that purpose it has been used the eUML2 tool, a plugin which offers a intuitive way to develop UML® diagrams in Eclipse® [42].

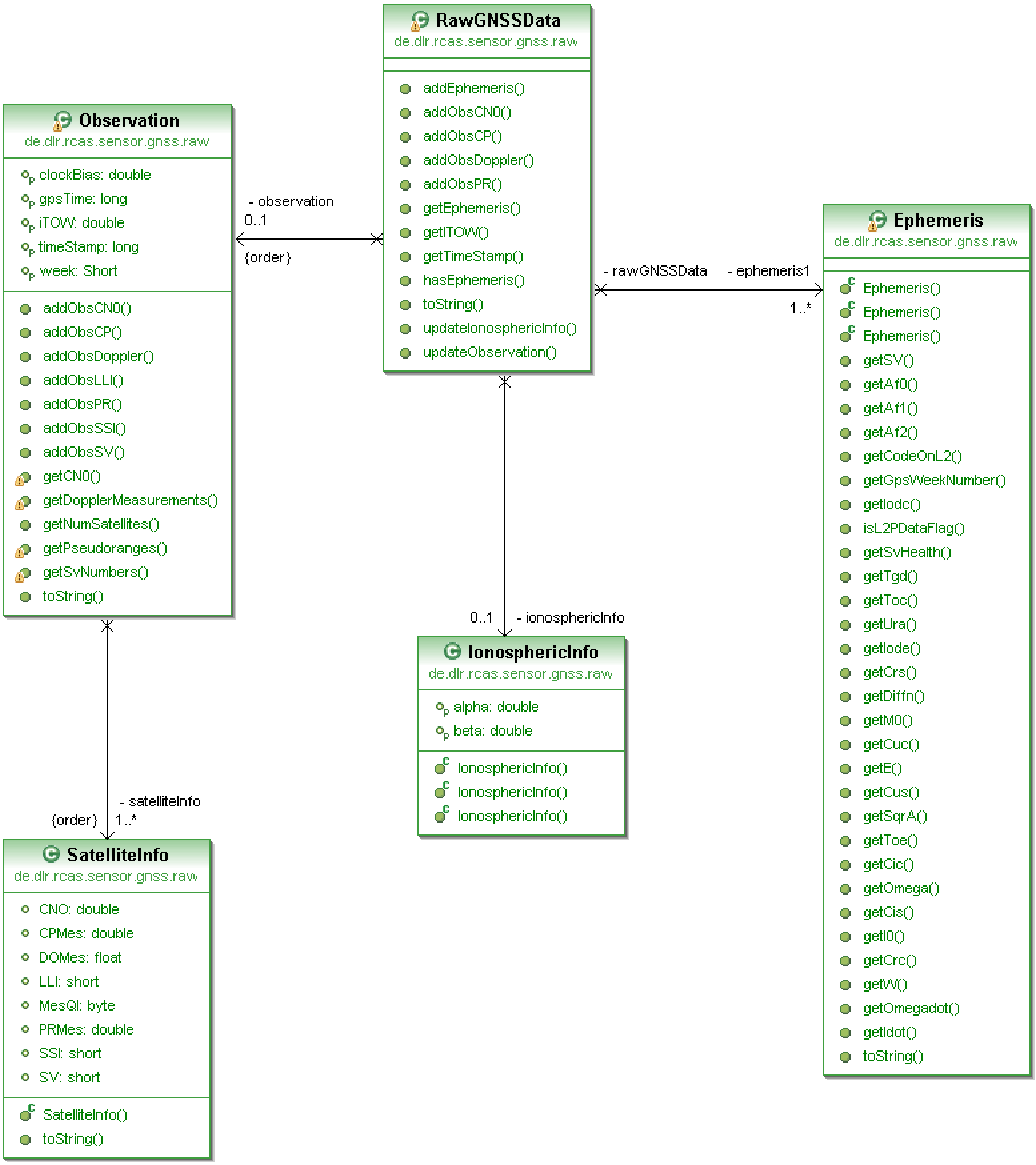


Figure C.1: UML raw GNSS data structure class diagram

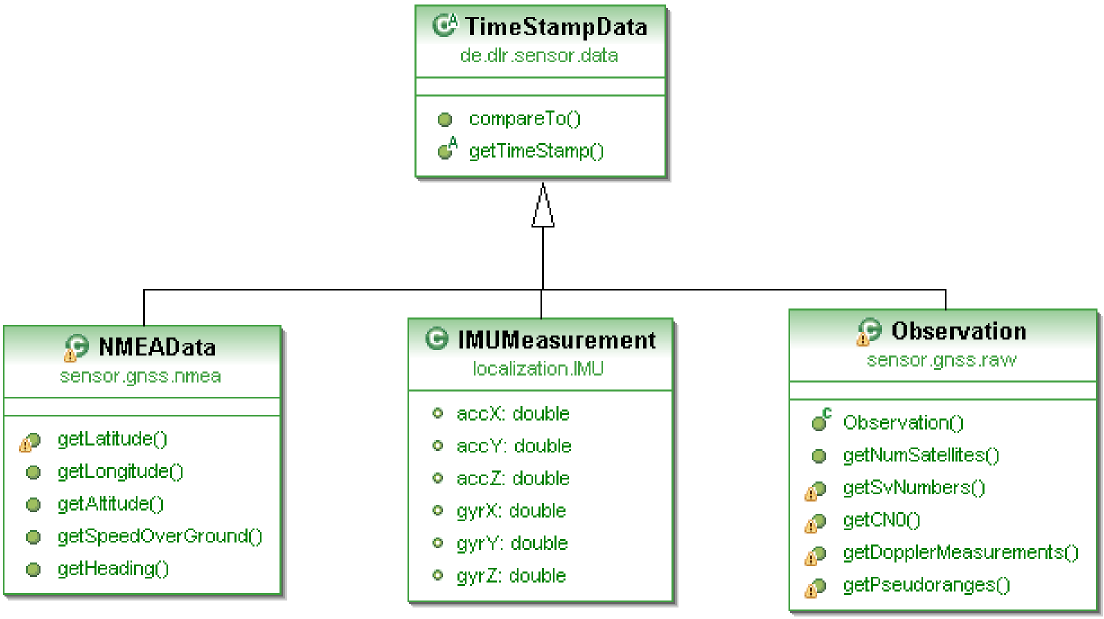


Figure C.2: UML data synchronization scheme class diagram

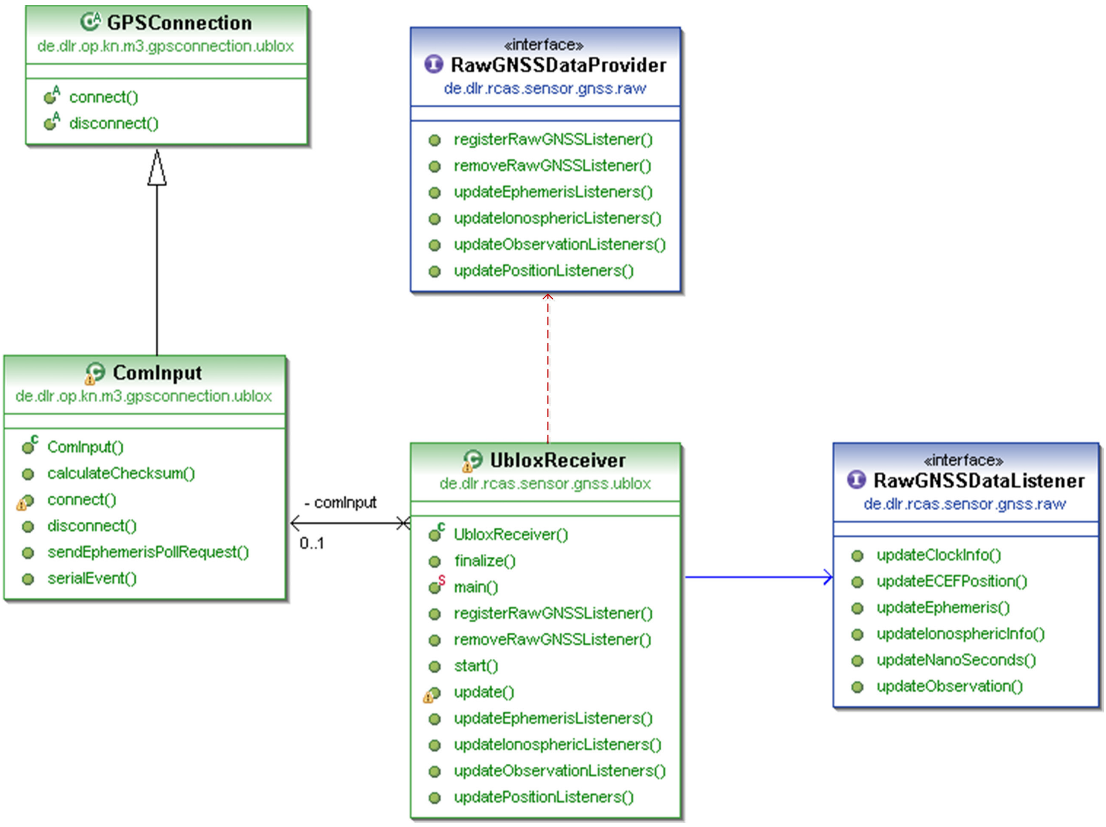

GPSConnection
de.dlr.op.kn.m3.gpsconnection.ublox
connect()
disconnect()
«interface»
RawGNSSDataProvider
de.dlr.rcas.sensor.gnss.raw
registerRawGNSSListener()
removeRawGNSSListener()
updateEphemerisListeners()
updateIonosphericListeners()
updateObservationListeners()
updatePositionListeners()
ComInput
de.dlr.op.kn.m3.gpsconnection.ublox
ComInput()
calculateChecksum()
connect()
disconnect()
sendEphemerisPollRequest()
serialEvent()
- comInput
0..1
UbloxReceiver
de.dlr.rcas.sensor.gnss.ublox
UbloxReceiver()
finalize()
main()
registerRawGNSSListener()
removeRawGNSSListener()
start()
update()
updateEphemerisListeners()
updateIonosphericListeners()
updateObservationListeners()
updatePositionListeners()
«interface»
RawGNSSDataListener
de.dlr.rcas.sensor.gnss.raw
updateClockInfo()
updateECEFPosition()
updateEphemeris()
updateIonosphericInfo()
updateNanoSeconds()
updateObservation()

Figure C.3: UML u-blox receiver interfacing class diagram

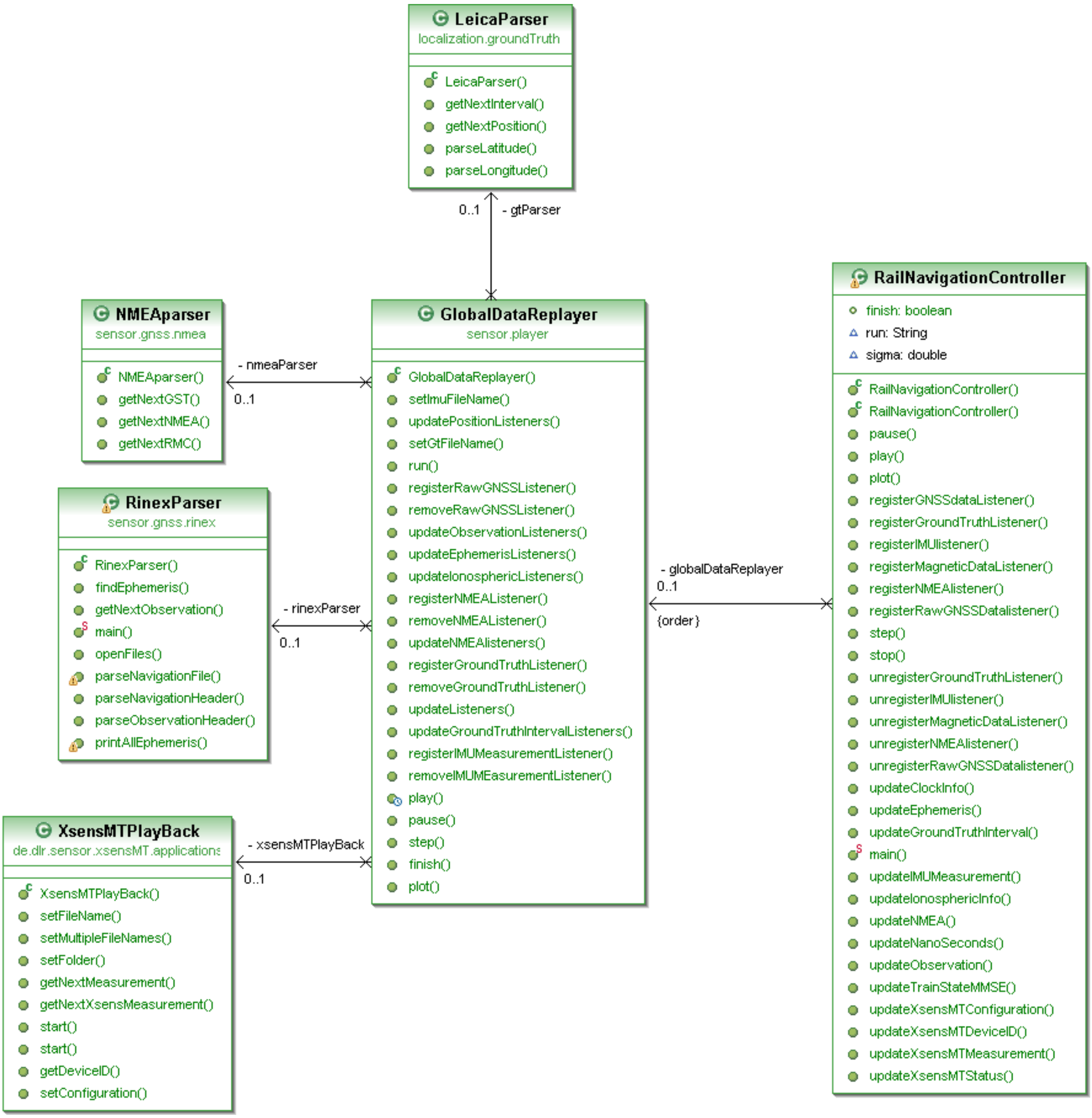


Figure C.4: UML data replay scheme class diagram

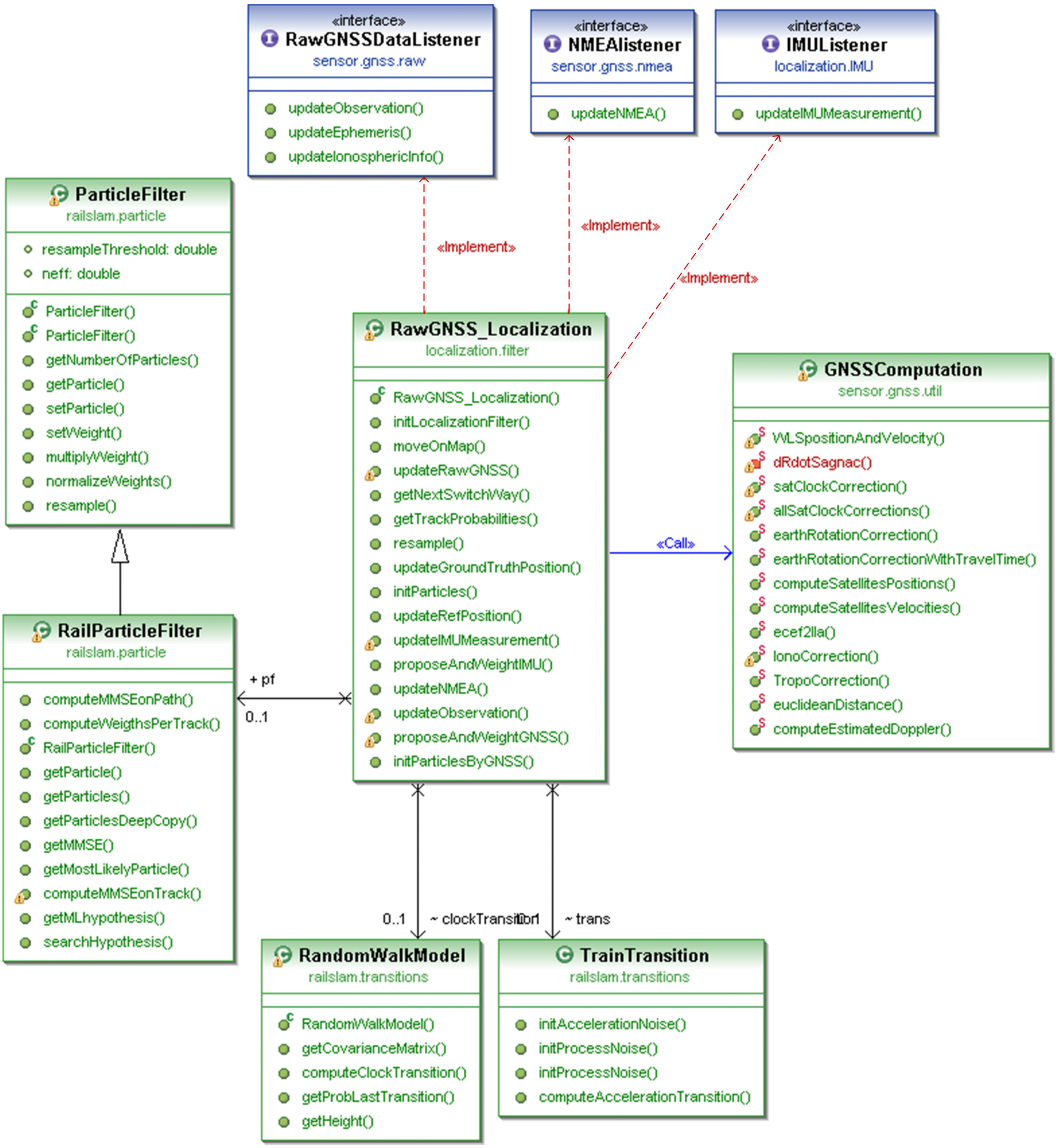


Figure C.5: UML main classes involved in the filtering