Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electronics and Telecommunications Engineering (36^{th}cycle)

Design of Advanced Positioning Solutions: A Bayesian Approach

By

Simone Zocca
******

Supervisor(s):
Prof. F. Dovis

Doctoral Examination Committee:
Prof. L. Ortega, Referee, Institut polytechnique des sciences avancées
Prof. M. Tahir, Referee, Lahore University of Management Sciences
Prof. E. Cianca, Università di Roma Tor Vergata
Dr. C. Cristodaro, Thales Alenia Space
Prof. R. Garello, Politecnico di Torino

Politecnico di Torino
2024

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Simone Zocca
2024

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Acknowledgements

I hope this thesis is a worthy representation of the most relevant things I have done, studied and learned in the past few years, as well as a collection of my thoughts and ideas for the work I hope to be able to continue to do in the future. Because of that, I would like to begin by acknowledging and thanking all the people who, in different ways, have contributed to this journey, some on a professional level and some on a personal level.

First and foremost I would like to thank my advisor Prof. Fabio Dovis, to whom I am profoundly grateful. I am thankful for your support and belief in my ideas, and most importantly for the freedom you gave me to explore them. Equally, I would like to thank all my other colleagues of the NavSAS group, and in particular to those with whom I worked more closely and from whom I learned the most: Alex, Andrea, Yihan and Oliviero. I am grateful for your generosity and honesty.

I would like to express my sincere gratitude to my parents, Emanuela and Michele for the unlimited support and love I have always received. I am immensely proud of and grateful to my grandfather Lorenzo, as well as my other close relatives, Massimo, Chiara, Matilde and Amedeo.

I would also like to express my gratitude to all my close friends for their support and encouragement, so thank you to Marfisa and Emanuele, Vito, Emanuele, Umberto, Federico, Stefano, Francesco and Giorgio.

Last but not least I express my immense gratitude and love to my partner, Hao. Despite our different backgrounds and fields of interest, we respect, inspire and support each other every day to become better professionals in what we do and better people for ourselves, for each other, and for all the people who love us. I could not be more lucky.

Abstract

In today's connected world, Global Navigation Satellite Systems (GNSS) have become an indispensable tool for a wide range of modern applications. From enabling autonomous vehicles, to guiding commercial aircraft to enabling precise location on smartphones, GNSS play a central role in many activities of our daily lives. As reliance on satellite navigation continues to grow, it becomes increasingly important to improve the accuracy and precision of the positioning solution, and to ensure its integrity for the reliability and safety of the applications relying on such information. In light of this, the thesis focuses on the positioning unit of the receiver, and in particular on the use of Bayesian inference to estimate Position, Velocity and Timing (PVT). Nevertheless, the investigations presented here also extend to other aspects related to the estimation of location parameters. First, the theoretical background involved in the estimation process is explored in detail, so as to gain a solid theoretical base and obtain insights for the development of novel measurement processing methodologies.

Using the theoretical analysis as a starting point, application-specific improvements of state-of-the-art filtering algorithms are introduced first by exploiting the structure of the measurement model to devise a more efficient estimation in terms of computational load. Then, novel techniques are presented, spanning different topics and different receiver architectures, from stand-alone GNSS, to cooperative techniques and sensor integration. Ultimately, the common goal is to increase the precision, accuracy, and robustness of PVT estimation by exploiting different approaches and with different degrees of architecture complexity.

One of the main limitations and challenges of stand-alone GNSS is to provide accurate positioning information in harsh urban environments with complex signal reflection phenomena, leading to multipath and Non line-of-sight (NLoS) effects. For this reason, the scenarios addressed include the detection and mitigation of multipath and NLoS based on the consistency between redundant measurements, with applications both for real time and for post-processing. Furthermore, one of the solutions that have been considered to overcome the limitations of GNSS is exploiting additional

information from other sensors or from other users. In particular, the tight integration of Ultra-Wide Band (UWB) with GNSS has been considered, with a focus on time synchronization between the two technologies to enable accurate positioning for autonomous vehicles or drones. To conclude, more practical aspects often neglected by research effort are considered. Namely, the exchange of cooperative data among networked receivers is addressed by presenting a dedicated and open protocol for the exchange of positioning information from both GNSS and other sensors, thus enabling numerous cooperative applications.

Contents

List of Figures

List of Tables

Acronyms

3GPP	3rd Generation Partnership Project
ADAS	Advanced Driver-Assistance Systems
ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
AI	Aritficial Intelligence
ANOVA	ANalysis Of VAriance
ARAIM	Advanced RAIM
ASCII	American Standard Code for Information Interchange
BEIDOU	BeiDou Navigation Satellite System
BLUE	Best Linear Unbiased Estimator
BOC	Binary Offset Carrier
BTP	Basic Transport Protocol
C/N0	Carrier-to-Noise density
CAF	Cross-Ambiguity Function
CAM	Cooperative Awareness Message
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CE	Cooperative Enhancement
CEM	Cooperative Enhancement Message
CERIM	Cooperative Enhanced Receiver Integrity Monitoring
CHIMERA	CHIp-MEssage Robust Authentication
CODE	Center for Orbit Determination in Europe
COTS	Commercial Off-The-Shelf

CP	Cooperative Positioning
CPM	Collective Perception Messages
CPS	Cooperative Positioning Service
CRLB	Cramér-Rao Lower Bound
CSV	Comma Separated Values
DCC	Decentralized Congestion Control
DCM	Direction Cosine Matrix
DD	Double Differences
DENM	Decentralized Environmental Notification Message
DFT	Discrete Fourier Transform
DGNSS	Differential GNSS
DLL	Delay-Lock Loop
DOP	Dilution Of Precision
DSRC	Direct Short Range Communication
ECDF	Empirical Cumulative Density Function
ECEF	Earth Centered Earth Fixed
EGNOS	European Geostationary Navigation Overlay System
EKF	Extended Kalman Filter
ENU	East-North-Up
ESA	European Space Agency
ETSI	European Telecommunication Standards Institute
FDE	Fault Detection and Exclusion
FLL	Frequency-Lock Loop
FOSM	First-Order Second-Moment
GDOP	Geometrical Dilution Of Precision
GIM	Global Ionospheric Maps
GLONASS	Globalnaya Navigazionnaya Sputnikovaya Sistema
GLS	Generalized Least Square
GNSS	Global Navigation Satellite System

GPS	Global Positioning System
HDOP	Horizontal Dilution Of Precision
HMM	Hidden Markov Model
I2V	Infrastructure-to-Vehicle
ICD	Interface Control Document
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Frequency
IGS	International GNSS Service
IID	Independent Identically Distributed
INS	Inertial Navigation System
IP	Internet Protocol
IRNSS	Indian Regional Navigation Satellite System
ITS	Intelligent Transportation Systems
KF	Kalman Filter
LAD	Least Absolute Deviations
LEO	Low-Earth Orbit
LiDAR	Light Detection and Ranging
LNA	Low Noise Amplifier
LoS	Line-of-Sight
LS	Least Square
MAP	Maximum A Posteriori
MEO	Medium-Earth Orbit
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimate
MW-PF	Multi Weight Particle Filter
NCS	Navigation Constellation Simulator
NLoS	Non-Line-of-Sight
NMA	Navigation Message Authentication
NMEA	National Marine Electronics Association
NTP	Network Time Protocol

NTRIP	Network Transport of RTCM via Internet Protocol
OLS	Ordinary Least Square
PDF	Probability Density Function
PDR	Packet Delivery Ratio
PDU	Packet Data Unit
PF	Particle Filter
PLL	Phase-Lock Loop
PMF	Probability Mass Function
PNT	Positioning, Navigation and Timing
PPP	Precise Point Position
PPS	Pulse Per Second
PRN	Pseudo-Random Noise
PRR	Packet Reception Ratio
PTP	Precision Time Protocol
PVT	Position, Velocity and Time
QZSS	Quasi-Zenith Satellite System
RAIM	Receiver Autonomous Integrity Monitoring
RF	Radio-Frequency
RINEX	Receiver Independent Exchange Format
RMSE	Root Mean Squared Error
RSE	Root Squared Error
RSU	Road Side Unit
RTCM	Radio Technical Commission for Maritime Services
RTCMEM	RTCM Extended Message
RTK	Real Time Kinematic
SAEM	Services Announcement Essential Messages
SAMARCANDA	Synthetic Accurate Multi-Agent RealistiC Assisted-gNss DatAset
SBAS	Satellite-Based Augmentation System
SIR	Sequential Importance Resampling

SMC	Sequential Monte Carlo
SSR	Sum of Squared Residuals
SUMO	Simulation of Urban MObility
SVD	Singular Value Decomposition
TDOP	Time Dilution Of Precision
TECU	Total Electron Content Units
TGD	Timing Group Delay
ToA	Time of Arrival
UDP	User Datagram Protocol
UHF	Ultra-High Frequency
UPER	Unaligned Packed Encoding Rules
UTC	Universal Time Coordinated
UWB	Ultra-Wide Band
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VDOP	Vertical Dilution Of Precision
WAAS	Wide Area Augmentation System
WGN	White Gaussian Noise
WLS	Weighted Least Square
ZTD	Zenith Total Delay

Chapter 1

Introduction

Global Navigation Satellite System (GNSS) have revolutionized our lives and the way we interact with the world by providing precise Positioning, Navigation and Timing (PNT) information, which is critical to a multitude of applications we use daily. From everyday uses in smartphones and car navigation to complex operations in numerous domains such as aviation, maritime, agriculture and space, GNSS technology contributes to a substantial part of modern life. That being said, stand-alone GNSS still suffers from many limitations, especially in complex scenarios with reduced satellite visibility such as urban canyons. To overcome its limitations, many GNSS augmentation have been proposed and studied in recent years, from complex algorithms implementing advanced signal processing and estimation techniques, to sensor fusion and Cooperative Positioning (CP). This thesis approaches the topic of radio navigation by exploring multiple of these solutions, with the common goal of enhancing GNSS performance by investigating both its theoretical foundations and practical implementations. The aim is hence to enhance GNSS accuracy, reliability, and integrity, while addressing various challenges such as multipath effects, sensor synchronization, and computational load. By advancing methodologies and developing novel algorithms, this research seeks to contribute to the optimization and resilience of GNSS, ensuring its robust application in an increasingly connected and dynamic global environment.

1.1 Thesis Outline and Main Contributions

This section briefly provides an outline of the thesis, along with a summary of the content of each chapter. The original contributions and the paper in which they

were published are also discussed for each chapter. The remainder of this thesis is structured as follows:

Chapter 2 introduces the main general concepts and components of GNSS and provides a baseline description of the space and control segment. The chapter describes the main steps in the path of a GNSS signal, from its generation on board of the satellites up to its processing in the receiver, ultimately leading to the construction of observables. All the original contributions included in the later chapters of this thesis deal with the Position, Velocity and Time (PVT) stage of the receiver. For this reason, the second chapter serves as an introduction to all the aspects that lead to the computation of measurements then used in the PVT.

Chapter 3 introduces the main aspects of PVT computation by means of Least Square (LS) method. The main properties of the estimator are derived along with some historical background. Then, other modifications of the method are presented and their advantages and properties are discussed. The second part of the chapter delves deeper into two important concepts in LS estimation. First, residuals are analysed and their statistical properties derived by leveraging a graphical representation. Then, the concept of Geometrical Dilution Of Precision (GDOP) and its relationship with the estimator is analysed, while also deriving a theoretical lower bound. Although it does not include any original contribution per se, the analysis outlined in the chapter helped extending [4] into a work which proposes a modification of the LS with improved convergence performance for both pseudorange and Doppler positioning in the context of Low-Earth Orbit (LEO). This is the subject of a paper submitted at the end of the PhD and, at the time of writing, still under review.

Chapter 4 first introduces the concept of Bayesian inference, as opposed to batch estimation performed by LS described in the previous chapter. Then, the two main filters that will be used in the thesis are introduced, namely the Extended Kalman Filter (EKF) and the Particle Filter (PF). The second part of the chapter presents the Multi Weight Particle Filter (MW-PF), an original contribution published in [1]. The MW-PF algorithm optimizes the sampling of the posterior distribution by leveraging properties of the particular measurement model of GNSS, thus allowing to reach a given target accuracy with a decreased number of particles. The validity of the proposed method is supported both by a theoretical proof and by experimental assessment of its performance.

Chapter 5 is divided in two parts. The first part presents a method for the detection of multipath, and in particular for automatic labelling biased measurements due to faults in datasets. The method, published in [5], leverages precise models and corrections to remove other possible error sources and then exploits clustering of measurements based on DBSCAN to exclude inconsistent measurements. The second part of the chapter presents an improved weighting strategy for the PF. The methodology exploits the concept of soft information to build multi-modal probability distributions used in the measurement model in order to represent different hypothesis. The proposed statistical modelling thus helps to mitigate the effects of measurement biases due to multipath and Non-Line-of-Sight (NLoS) phenomena, especially in case of false alarms. The work was first presented at a conference [6] and then extended into a journal which has recently been submitted.

Chapter 6 deals with sensor fusion and more specifically with the integration of Ultra-Wide Band (UWB) measurements with GNSS. The chapter includes two main novel contributions. First, a comparison of Bayesian filters for the sensor fusion architecture is presented in the context of vehicular applications. The impact of unmodelled dynamics on the performance of the filters is first analysed theoretically and then experimentally validated, leading to a conference paper [7]. The second contribution, published in [8] and then extended in [9], tackles the problem of synchronization between different sensors by proposing a double-update architecture of the EKF which displays enhanced performance compared to standard single-update estimation of the time misalignment between measurements from different sensors.

Chapter 7 deals with the issue of network transmission of measurement to enable cooperative applications. In particular, an original contribution presented in the chapter is the development of an open-source, real-time protocol for the transmission of GNSS data among networked agents in the context of autonomous vehicles. The protocol, named Cooperative Enhancement Message (CEM), extends the capabilities of current solutions by transmitting raw measurements that can be processed by the navigation units of cooperating agents to improve the accuracy, integrity and availability of positioning solutions. The performance of the protocol from the network perspective has been assessed through an experimental campaign in terms of network load, delay and reception ratio, thus proving the viability of many GNSS cooperative applications. The work was first proposed in [3] and then extended into a jour-

nal paper [2] which also considers the transmission of measurement from other sensors.

1.2 Publications

The full list of publications is given below:

1.2.1 Journal Papers

Published:

- Raviglione, F., Zocca, S., Minetto, A., Malinverno, M., Casetti, C., Chiasserini, C.F. and Dovis, F., 2022. From collaborative awareness to collaborative information enhancement in vehicular networks. *Vehicular Communications, 36*, p.100497.
- Zocca, S., Guo, Y., Minetto, A. and Dovis, F., 2022. Improved weighting in particle filters applied to precise state estimation in GNSS. *Frontiers in Robotics and AI, 9*, p.950427.
- Guo, Y., Vouch, O., Zocca, S., Minetto, A. and Dovis, F., 2022. Enhanced EKF-based time calibration for GNSS/UWB tight integration. *IEEE Sensors Journal, 23(1)*, pp.552-566.
- Guo, Y., Zocca, S., Dabove, P. and Dovis, F., 2024. A Post-Processing Multipath/NLoS Bias Estimation Method Based on DBSCAN. *Sensors, 24(8)*, p.2611.
- Vouch, O., Nardin, A., Minetto, A., Zocca, S., Valvano, M. and Dovis, F., 2024. Aided Kalman Filter Models for GNSS-based Space Navigation. *IEEE Journal of Radio Frequency Identification*.

Submitted:

- Guo, Y., Zocca, S. and Dovis, F., 2024. GNSS Fault Detection and Mitigation Methods Based on Particle Filter. Submitted to *IEEE Transactions on Instrumentation and Measurement*.

- Morichi, L., Zocca, S., Minetto, A., Nardin, A. and Dovis, F., 2024. Pseudorange and Doppler-based Positioning: Improving convergence of Least Square Estimator from MEO to LEO. Submitted to *NAVIGATION: Journal of the Institute of Navigation*.

1.2.2 Conference Papers

Published:

- Minetto, A., Zocca, S., Raviglione, F., Malinverno, M., Casetti, C.E., Chiasserini, C.F. and Dovis, F., 2021, December. Cooperative localization enhancement through GNSS raw data in vehicular networks. In *2021 IEEE Globecom Workshops (GC Wkshps)* (pp. 1-6). IEEE.
- Dovis, F., Minetto, A., Nardin, A., Guzzi, S., Zocca, S., Vouch, O., Impresario, G., Musmeci, M., Facchinetti, C., D'Amore, G. and Ansalone, L., 2022. The LuGRE project: a scientific opportunity to study GNSS signals at the Moon. In *International Astronautical Congress: IAC Proceedings* (Vol. 2022, pp. 1-6). International Astronautical Federation, IAF.
- Vouch, O., Guo, Y., Zocca, S., Minetto, A. and Dovis, F., 2022, September. Improved Outdoor Target Tracking via EKF-based GNSS/UWB Tight Integration with Online Time Synchronisation. In *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)* (pp. 2409-2422).
- Guo, Y., Vouch, O., Zocca, S., Minetto, A. and Dovis, F., 2023, September. A comparison study between the EKF and SIR-PF for GNSS/UWB tight integration. In *2023 31st European Signal Processing Conference (EUSIPCO)* (pp. 835-839). IEEE.
- Vouch, O., Nardin, A., Minetto, A., Valvano, M., Zocca, S. and Dovis, F., 2023, September. A Customized EKF model for GNSS-based Navigation in the Harsh Space Environment. In *2023 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)* (pp. 13-18). IEEE.
- Zocca, S., Guo, Y. and Dovis, F., 2024, January. A GNSS Multipath and NLoS Mitigation Method for Urban Scenarios Based on Particle Filtering. In *Proceedings of the 2024 International Technical Meeting of The Institute of Navigation* (pp. 575-588).

- Morichi, L., Minetto, A., Nardin, A., Zocca, S. and Dovis, F., 2024, January. Pseudorange and Doppler-based State Estimation from MEO to LEO: A Comprehensive Analysis of Maximum Likelihood Estimators. In *Proceedings of the 2024 International Technical Meeting of The Institute of Navigation* (pp. 677-691).

To appear:

- Vouch, O., Nardin, A., Minetto, A., Zocca, S., Dovis, F., Konitzer, L., Parker, J., Bernardi, F., Tedesco, S., Fantinato, S., 2024, July. Bayesian Integration for Deep-Space Navigation with GNSS Signals. Accepted in: *2024 27th International Conference on Information Fusion (FUSION)*. IEEE.

Chapter 2

Overview of Global Navigation Satellite Systems

GNSS is a technology used to estimate the absolute position of a user using range measurements obtained thanks to the transmission of radio signals. While the history of estimating one's position and using it to navigate spans millennia, the use of radio signals for this purpose is limited to the last few decades. The first navigators used the position of stars in the sky to find their bearing. With the improvement of technology, the stars have been replaced with constellations of satellites orbiting around the earth and acting as reference points.

While there are many different approaches to positioning that have been exploited in the past, current GNSSs are based on trilateration, namely to compute a position based on a set of reference points and the measured distance between them and the user. In particular, the ranges are computed based on a Time of Arrival (ToA) paradigm [].

This consists in estimating the amount of time that it takes for the radio signal to travel from the transmitter to the receiver. This can then be translated into a range by knowing that the radio signal travels at the speed of light (almost).

Actually, estimating the transit time of a radio signal is not simple. It may seem that it is enough to take the time at which it leaves and the time at which it arrives. The problem is that these two steps are made by two different units that work independently. The time measurement is therefore consistent only if the clocks of these two units are perfectly synchronized.

Since the distance is then calculated by multiplying by the speed of light, it is important to point out that even a millisecond error between the two clocks is enough

to have a huge error on the measured distance, more precisely 300 kilometres.
It might seem that the only solution is to buy sufficiently accurate clocks to obviate the synchronization problem. Indeed, the clocks aboard satellites are extremely accurate atomic clocks. Unfortunately, however, due to issues of space and cost, it is not possible to install one of these clocks on every receiver.
How then to maintain synchronization between the two clocks? The only way is to let the clocks be non-synchronous, and to treat the time difference between the two as an unknown. If the two clocks mark two different times, but their difference is known (with reasonable accuracy), then this can be corrected to obtain an accurate distance estimate.
Consequently, the purpose of a GNSS receiver is to estimate not only its own position in space, but also its position in time relative to satellites.

2.1 Space and Ground Segment

The first navigation system to be deployed was Global Positioning System (GPS). Its system architecture was approved by the Department of Defence of the USA in 1973. The system initially consisted of 24 Medium-Earth Orbit (MEO) satellites, with orbits designed to guarantee a global coverage of at least 4 satellites in visibility. Since then, many improvements and advancements have been made. More satellites have been launched, transmitting more signals over more bands in order to enable more applications.
In addition, other systems have also been designed with different degrees of interoperability between each other. The main other GNSS systems include Galileo developed by the European Space Agency (ESA), Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONASS) by the Soviet Union and BeiDou Navigation Satellite System (BEIDOU) by China. Other regional systems include the Japanese Quasi-Zenith Satellite System (QZSS) and Indian Indian Regional Navigation Satellite System (IRNSS). Furthermore, there are also systems for augmentation and differential techniques named Satellite-Based Augmentation System (SBAS). Some example include European Geostationary Navigation Overlay System (EGNOS) and Wide Area Augmentation System (WAAS).
More information on GNSS constellations can be found in [10–12]. For the most part, each system is characterized by some differences w.r.t. others in terms of number of satellites, orbits, signal modulations and so on. Nonetheless, the remainder of this

chapter will attempt to summarize the main operating concepts that are common to all systems.

2.1.1 Signal Structure

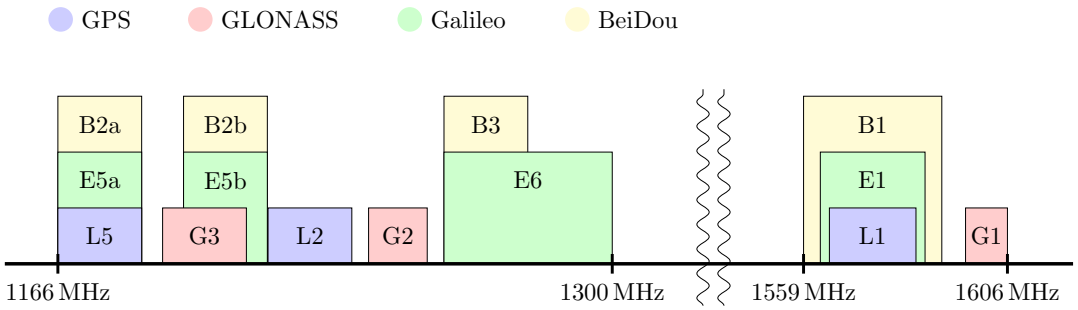


Fig. 2.1 Overview of signal bands.

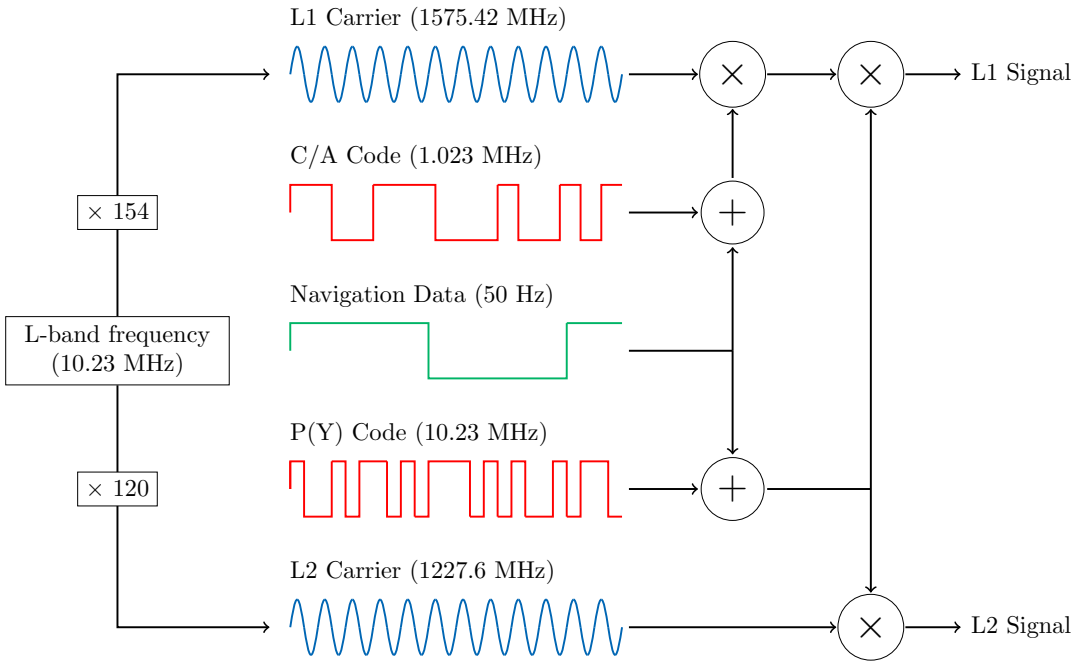


Fig. 2.2 Example of signal plan for GPS L1 and L2 bands. The time-series of different components are not in scale.

Each signal transmitted over a specific carrier is referred to as a channel. The signal is made of the In-Phase (I) and Quadrature (Q) components which make up the complex signal, which can be modelled as:

$$S_X(t) = \sqrt{2}\, P_T \left(S_I(t)\, cos(2\pi f_c t) - j S_Q(t)\, sin(2\pi f_c t) \right) \tag{2.1}$$

where f_c is the carrier frequency and each I and Q components are real signals with the structure:

$$S_{I,Q}(t) = s(t)\, c_p(t)\, c_s(t)\, d(t)\, p(t) \tag{2.2}$$

where:

- $s(t)$ is the subcarrier.
- $c_p(t)$ is the primary code or *spreading sequence*. A Pseudo-Random Noise (PRN) sequence.
- $c_s(t)$ is a secondary code.
- $d(t)$ is the navigation data.
- $p(t)$ is the pilot.

However, not all channel transmit all the components. Let us take a closer look at each one:

Carrier Carriers are in the L-band (1-2 GHz) in the range of Ultra-High Frequency (UHF). The signal bands of GPS are: L1 around 1575.42 MHz, L2 around 1227.60 MHz and L5 around 1176.45 MHz. Each one is an integer multiple of the main frequency of $f_0 = 10.23\,\text{MHz}$. A summary of the GNSS bands, including other constellations, is given in Figure 2.1.

Subcarrier Some channels include a sub-carrier. The effect of the sub-carrier is to split the main lobe of the spectrum of the signal into two side lobes. This reduces the cross-interference from systems operating in the same frequency bands and is thus designed with the intent to support interoperability of different systems. Some examples of sub-carriers include the Binary Offset Carrier (BOC) and AltBOC used in Galileo.

Code Sequences The channel access strategy used by GNSS is Code Division Multiple Access (CDMA), so that each signal is transmitted on the same band at the same time. Therefore, all ranging codes should be orthogonal to each other, to ensure that the receiver can correctly and uniquely identify the code sequence belonging to each satellite. For example GPS L1 C/A employs PRN sequences belonging to the family of Gold codes, while Galileo uses Memory codes. In some cases, a secondary code is also transmitted to enhance the performance.

Data Satellites transmit ephemeris data that the user can decode the obtain information about the satellites. More precisely, their position, velocity, clock bias which are needed in order to successfully estimate the receiver states. Furthermore, other information can be used, such as status of the satellite.

Pilot Some signals contain a pilot channel. These channels do not contain any navigation data and can be used for long time extensions.

An example of the combination of different signal components for the L1 and L2 bands of GPS is given in Figure 2.2. The signal structure and each component are discussed more in detail in [13–15, 11, 12, 16].

2.2 Receiver Architecture

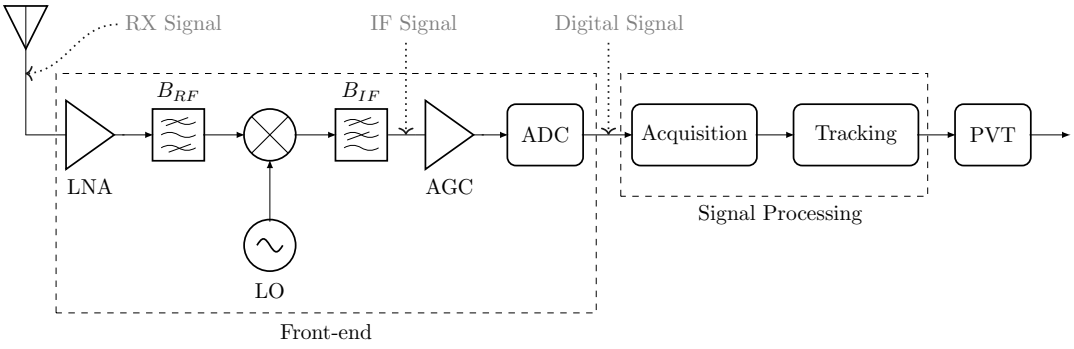


Fig. 2.3 High-level diagram of a receiver architecture.

Once the signal traverses the atmosphere, it reaches the receiving antenna. From there, the receiver is tasked with obtaining a navigation solution and any other product required by the specific application. In order to achieve so, the receiver architecture includes many different stages. A high-level diagram of some of the most important stages is given in Figure 2.3.
First, the front-end is responsible of down-converting and digitalizing the signal so that it can be processed by the following stages. Then, acquisition and tracking are responsible of obtaining estimates of Doppler shift of the received signal and its misalignment with a local copy of the code. Starting from the output of the tracking, the PVT block in Figure 2.3 is responsible of constructing the pseudorange measurements, obtaining the satellites position and velocity and finally computing an estimate of the receiver states.

2.2.1 Front-End

The first task of the front-end is to amplify the received signal by means of a Low Noise Amplifier (LNA). Then, the signal is filtered using a band-pass filter to only work with the band of interest. Then, down-conversion to an Intermediate Frequency (IF) is performed using local oscillators, whose sinusoidal tone is mixed with the received signal. The result is a signal made of two component, one which depends on the sum of the Radio-Frequency (RF) and the oscillator frequency and the other which depends on their difference. Then, a low-pass filter is applied so that only the component due to their difference remains. As such, the frequency of the local oscillator should be adjusted according to the RF of interest and the desired IF. Finally, the down-converted signal can be converted using an Analog-to-Digital Converter (ADC), thus obtaining a digital signal at a desired sampling frequency. Optionally, an Automatic Gain Control (AGC) can be employed to automatically adjust the amplitude of the signal in order to fit the requirements of the ADC.

2.2.2 Acquisition

Once the signal has been filtered and down-converted, it is now centred around the intermediate frequency f_{IF}. However, it is also still affected by an unknown Doppler shift. The purpose of the acquisition stage is to detect which signals are visible to the receiver and to provide an initial estimate of delay and Doppler shift of each signal that can be later refined by the tracking stage. Furthermore, this initial estimate allows to demodulate the data present in the signal.
In order to perform such estimates, the acquisition stage performs a series of correlation between the received signal at intermediate frequency and a local copy of the PRN code at the same frequency. To find the delay of the incoming signal, the properties of auto-correlation of Gold codes can be exploited. In fact, a peak of the autocorrelation should be present only if a given code is present in the received signal (with sufficient power), and the index of the peak will correspond to the misalignment between the local copy and the received one, in terms of samples. However, the incoming signal is also affected by Doppler. This might result in a loss of correlation if the local copy is not at the same central frequency as the incoming signal. Therefore, different values of Doppler shift should be tested (around the intermediate frequency) in order to find a rough estimate of the Doppler of the received signal.
As a result, the receiver should try many different combinations of delay and Doppler, and test the correlation between the received and local copy of the signal for each one.

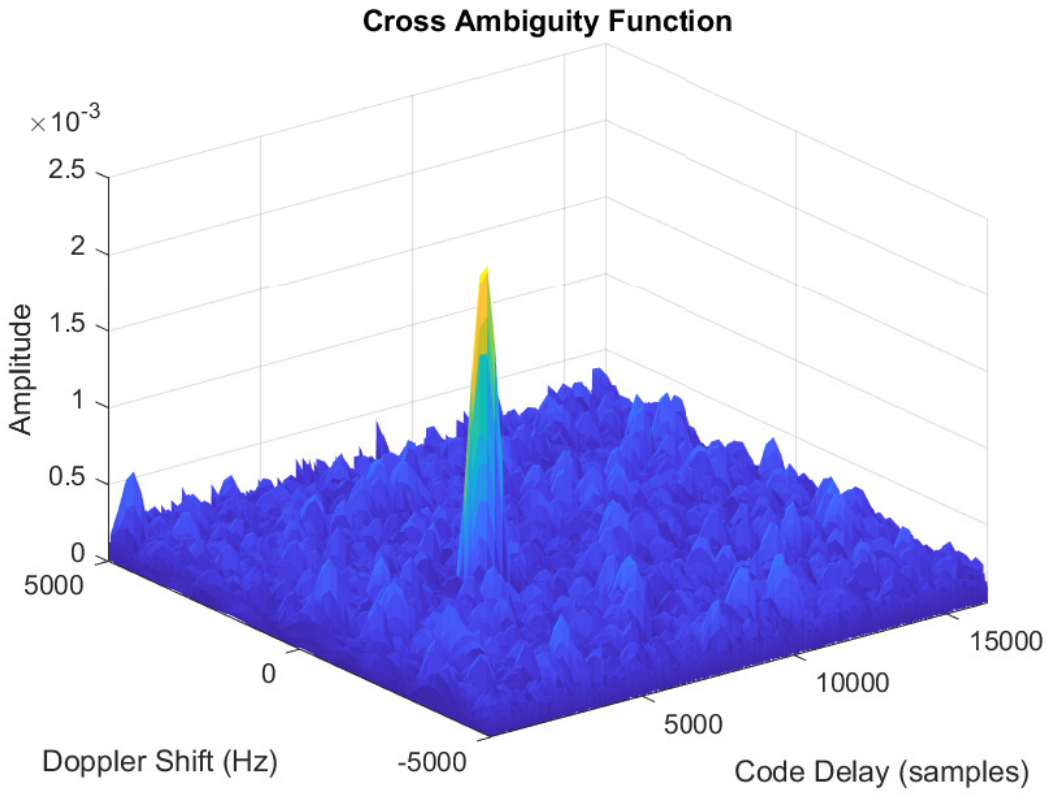


Fig. 2.4 Example of acquisition for GPS signal PRN 30 of a real data acquisition. Correlation is performed for one copy of the code period sampled at 16 samples per chip.

All these combinations can be seen as a two dimensional grid in the delay-Doppler domain, where each couple of points in the grid will result in a value of the correlation. Thus, for each pair of delay and Doppler, the local copy is obtained as the product of local code and carrier:

$$s_L[n] = c_p[n - \tau]\, e^{j2\pi(f_{IF}+f_D)nT_s} \tag{2.3}$$

where τ is the shift of the local copy in terms of samples, f_D is the Doppler frequency applied to the local carrier. Then, the correlation can be obtained as:

$$R(\tau, f_D) = \frac{1}{L}\sum_{n=0}^{L-1} s_{IF}[n]\, s_L[n] \tag{2.4}$$

where L is the total number of samples considered for the acquisition and s_{IF} is the input signal centered around the intermediate frequency. Then, the detection is usually performed on the squared envelope of the Cross-Ambiguity Function (CAF).

This is done in order to avoid being sensitive to the phase of the incoming signal and the sign of the bits of the data (if it is present).
The acquisition can be either serial or parallel, in the latter case it can be performed both in the time and frequency domain. In the serial approach, each value of the grid is computed by means of correlations. In the case of parallel acquisition, the properties of the Discrete Fourier Transform (DFT) can be exploited in order to obtain an entire "line" of the grid in one step. This process greatly speeds up the computation time needed for the acquisition. Let us analyse the values and size of the grid in each dimension:

- **Delay**: The number of values tested should be an integer multiple of the number of samples in one code period. In case the number of integer copies of the code is more than one, we refer to is as a *time extension* techniques, which can be *coherent* or *non-coherent*. Extending the time of acquisition generally leads to greater processing gain in terms of Carrier-to-Noise density (C/N0), and thus a higher probability of acquiring signals with low power. Nevertheless, there are some limitations and drawbacks of each time extension technique, whose discussion can be found in [10, 11].

- **Doppler shift**: The range of possible values is determined by the orbital parameters of the satellite. Considering GPS constellation, a reasonable search range is $\pm 5000\,\text{Hz}$ for a user on the ground. Practically, it is computationally impossible to repeat the correlation for every possible value of Doppler shift. As a results, it is common to choose a Δf and test values in the possible range spaced by such Δf. A common rule of thumb for choosing such spacing is $\Delta f = \frac{2}{3T_{coh}}$ where T_{coh} is the coherent integration time, expressed in seconds. This value of Δf guarantees a loss of at most 3 dB w.r.t. the peak that would be obtained using the correct value of Doppler shift for the correlation. Such choice thus represents a good trade-off between loss in performance and computational load (and thus time required) for the acquisition.

An example of the squared envelope of a CAF obtained from the acquisition of a real GPS signal is given in Figure 2.4.
Then, a proper detection threshold should be designed based on the desired performance in terms of probability of false alarm. An in-depth discussion of the statistical theory for signal acquisition in GNSS can be found in [17]. If the peak of the CAF exceeds such threshold, then the PRN under consideration can be considered acquired and the indexes of the peak correspond to the initial estimates of delay and Doppler

shift.
The time required to perform acquisition plays a vital role in the time needed to obtain a first fix. As such, it is of interest to speed up the process as much as possible. In order to do so, the receiver can receive external information which can reduce the number of operations.
For example, certain PRN can be excluded from the search if they are known to be not visible from the position of the receiver, or at least a rough estimate of it. Another possibility is to restrict the search space in the Doppler domain if there is available information regarding the elevation of the satellite and whether it is rising on decreasing in elevation w.r.t. the receiver.

2.2.3 Tracking

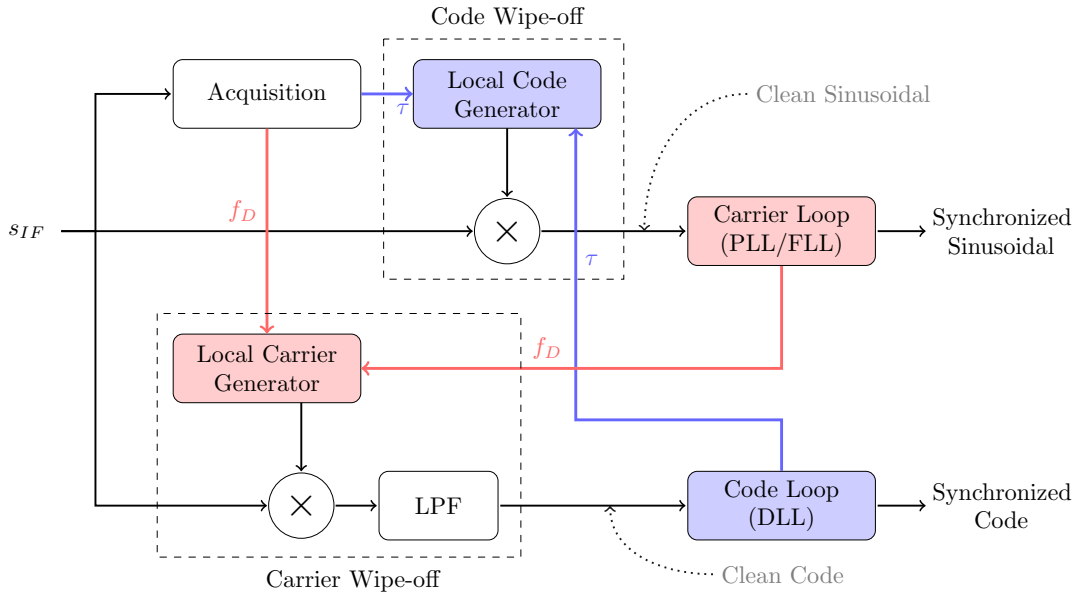


Fig. 2.5 Simple diagram of a generic tracking loop architecture.

Once an initial estimate of delay and Doppler of the received signal has been obtained, the purpose of the tracking stage is to refine such estimates and maintain to track their changer in order to maintain a local copy of the signal synchronized to the incoming one. In order to achieve this, the tracking stage is usually composed of two loops working together and for each other, the Delay-Lock Loop (DLL) and the Frequency-Lock Loop (FLL) (or Phase-Lock Loop (PLL)).
Using the estimate of the delay obtained from the acquisition, the local code can be aligned to that of the received signal in order to remove the effect of the PRN

sequence. This operation is called *code wipe-off* and thus leaves the carrier modulated by the navigation message. This signal is input to the FLL which can now provide a more accurate estimate of the frequency by wokring on a reduced search space. Equivalently, using the initial estimate of f_D from the acquisition, the local carrier generator can be employed in order to perform *carrier wipe-off* and obtain a clean code, which is then fed to the DLL for a refinement of the delay estimate.
A block diagram of this architecture is provided in Figure 2.5.
These two loops work continuously during the operations of a receiver, with the goal of tracking changes in both delay and Doppler shift parameters of the incoming signals due to changes in the relative distance and velocity between the receiver and the satellite. A more detailed description of tracking loops and commonly used discriminator functions can be found in [12].

2.2.4 Navigation Message

Once the tracking loops obtain a lock of the incoming signal, the navigation message can be demodulated. This can be done by decoding the stream of bits which are obtained from the output of the tracking. An example of messages broadcasted by the satellites can be found in the GPS Interface Control Document (ICD) or the Galileo open service document. The bit rate of the navigation message is 50 bits per second for GPS. A whole message lasts for 12 and a half minutes and contains 25 frames. Therefore, each frame is 30 seconds long and is further divided in 5 subframes of 6 seconds each. Most importantly, the receiver can first seek for the preamble, a sequence of m bits which precedes the beginning of a subframe. This can be done by means of correlation with a local copy of the preamble. Then, the receiver can successfully decode the content of the subframe. Each subframe starts with the Telemetry (TLM) word and the Handover word (HOW), which are then followed by all the orbital parameter that are needed to compute position and velocity of the satellite.

2.3 GNSS Observables

Once the data has been demodulated, the receiver has knowledge about the position and velocity of satellites. The only other information that the receiver needs to be able to compute a PVT solution is to construct its measurements. In particular, the measurements can be obtained from the outputs of the tracking. While the

Doppler shift is more straight-forward, the pseudorange can be constructed from the timing information. In general, the receiver needs to obtain measurements that bring information about the states that it is interested in estimating. Pseudoranges bring information about the three-dimensional position and clock bias of the receiver, and therefore 4 measurements of this king are sufficient to unambiguously estimate these 4 states. Similarly, 4 Doppler shift measurements are enough to unambiguously estimate 3-dimensional position and clock drift.
Nonetheless, other kinds of observables can be extracted from the received signals and used to estimate receiver states or improve the accuracy of other observables (e.g., carrier phase measurements for carrier smoothing of code pseudoranges). In this thesis, only code-based pseudoranges and Doppler shift measurements will be introduced since they form the basis of all the works presented.

2.3.1 Pseudorange

Using the paradigm of *common reception time*, the receiver sets a common received time for all channels corresponding to the the reception of the first TLM word. The satellite from which this first word is received is therefore taken as a reference.
Then, for each channel, the receiver computes the difference between the reception time of the channel and the reception time of the reference. Using the reference satellite, the HOW is read from the previous subframe, which represents the transmission time in the GNSS time scale.
The difference between the transmission time in the GNSS time scale and the reception time in the receiver time scale depends on the geometrical distance between the two objects (i.e., the travel time of the signal) and a difference between the two time scales, which is an unknown clock bias. Since the clock bias cannot be determined at first, a nominal value is set, based on values of travel time that can be expected. The estimation of the clock bias will then be refined by the PVT. At this point, the reference pseudorange has been constructed, and all the other ones can be obtained by summing the differences between the reception times of each channel and the reference one.
Given a reception time t, the pseudorange can be modelled as:

$$\begin{aligned} \rho_i &= \mathsf{c}\,(\tau \,+\, b_c(t) \,-\, b_{c,i}(t-\tau)) \\ &= \mathsf{c}\,\tau \,+\, \mathsf{c}\,(b_c(t) \,-\, b_{c,i}(t-\tau)) \end{aligned} \tag{2.5}$$

where:

- $\mathtt{c}$ is the speed of light in vacuum, namely $299\,792\,458\,\mathrm{m\,s^{-1}}$.
- $b_c(t)$ and $b_{c,i}(t-\tau)$ are the clock biases of receiver and satellite i w.r.t. the system time evaluated respectively at reception time t and transmission time $t-\tau$. Both quantities are expressed in seconds.
- τ is the apparent transit time,.

Hence, the pseudorange in (2.5) is expressed in meters. This means that the pseudorange measured by the receiver depend on the travel time of the signal plus the difference between the GNSS time scale and the one of the receiver. Since the travel time depends on the geometrical distance between transmitter and receiver, the pseudorange can be written as:

$$\begin{aligned}\rho_i &= \underbrace{\sqrt{(p_{x,i}-p_x)^2+(p_{y,i}-p_y)^2+(p_{z,i}-p_z)^2}}_{r_i} + c_b \\ &= \|\boldsymbol{p}_i-\boldsymbol{p}\| + c_b = r_i \,+\, c_b\end{aligned} \tag{2.6}$$

where $\boldsymbol{p} = [\,p_x \;\; p_y\, p_z\,]$ where the subscript i is used to denote the position of the i-th satellite and c_b is the clock bias of the receiver expressed in meters.
Considering also all the disturbances that may affect the transmission of the signal, then:

$$\rho_i = r_i \,+\, c_b \,+\, D \,+\, \xi_i \,+\, D_i \,+\, I_i \,+\, T_i \,+\, R_i \,+\, F_i \,+ \epsilon_i \tag{2.7}$$

where:

- r_i is the geometrical range (Euclidean distance) between the user and satellite i.
- ξ_i is a shift due to the antenna phase centre.
- D and D_i are the instrumental delay of the receiver and satellite respectively.
- I_i is the delay due to the effect of the Ionosphere. It is referred to the i-th satellite since signals travel different sections of the atmosphere based on the position of the satellite.
- T_i is the delay due to the effect of the Troposphere. Similarly to ionosphere, it is also different for each satellite.
- R_i groups all other error contributions that can be corrected by the receiver. A discussion of such terms can be found in [15] and is also briefly provided in

Appendix B. For the remainder of this work we assume that these phenomena are always accounted and corrected for by the receiver.

- F_i is a term that groups all possible contributions due to faults. This can include faults due to the satellite clock but also due to the environment of the receiver such as multipath. If there are no faults, then the term is 0. Otherwise it is unknown and it is usually of interest to estimate it or at least to detect when it is not 0.

- ϵ_i is an additional error term. It is a contribution that can be modelled as a random variable that depends on the receiver noise.

All quantities are expressed in meters.

2.3.2 Doppler shift

The other kind of observables commonly used to estimate the velocity is the pseudorange rate or Doppler shift. The latter refers to the shift in received frequency w.r.t. the transmitted one caused by the Doppler effect. In particular, the pseudorange rate can be written as:

$$\dot{\rho}_i = \dot{r}_i + \dot{c}_b \tag{2.8}$$

where $\dot{c}_b$ is referred to as the clock drift of the receiver. Using the chain rule of the derivative:

$$\dot{\rho}_i = \frac{\partial \rho_i}{\partial t} = \frac{\partial \boldsymbol{p}}{\partial t} \frac{\partial \rho_i}{\partial \boldsymbol{p}} \tag{2.9}$$

and thus we obtain that:

$$\dot{\rho}_i = (\boldsymbol{v}_i - \boldsymbol{v}) \frac{(\boldsymbol{p}_i - \boldsymbol{p})}{\|\boldsymbol{p}_i - \boldsymbol{p}\|} + \dot{c}_b \tag{2.10}$$

which is the projection of the relative velocity on a unitary vector pointing from the user to the satellite.
Doppler shift and pseudorange rate are related by:

$$\dot{\rho}_i = -\lambda f_D \tag{2.11}$$

where λ is the wavelength associated to the carrier frequency of the transmitted signal. From the physical point of view, the Doppler shift is caused by the source of a signal and the receiver getting closer or further away. Sideways motion does not

contribute to the perceived shift of the received frequency.

Once the receiver has determined the position and velocity of satellites by demodulating the navigation message and has constructed the measurement, it now has all the ingredients necessary to determine its PVT.

Chapter 3

Least Squares and Estimation Theory

The estimation of user position by means of least squares method is very well known, nevertheless this chapter is devoted to provide a more in-depth analysis of the method. First, the method will be derived and its main properties will be presented. Then, an analysis of the main algorithms to solve least squares problems and their properties will follow. The theoretical discussion will be accompanied by a visual representation of the solutions along with a numerical example, in order to gain some intuition on the most important quantities in linear regression models. The visual interpretation will be leveraged to investigate the statistical properties of certain regression metrics. Finally, theoretical limits will be derived which will allow to complete the discussion with interesting remarks.

3.1 Introduction

First, in its most general case, the purpose of a GNSS receiver is to estimate the state vector:

$$\boldsymbol{x} = [\underbrace{p_x \;\; p_y \;\; p_z}_{\boldsymbol{p}} \;\; c_b \;\; \underbrace{v_x \;\; v_y \;\; v_z}_{\boldsymbol{v}} \;\; c_d\,] \tag{3.1}$$

where $\boldsymbol{p}$ is the position of the receiver in 3 dimensional space, $\boldsymbol{v}$ its velocity and c_b and c_d are its clock bias and drift respectively. The state vector of a given satellite i is denoted with $\boldsymbol{x}_i$. For the sake of simplicity, the state vector without any subscript refers to the user. For the remainder of this chapter we will be concerned by estimating

the position and clock bias of the receiver using pseudorange measurements, but similar ideas and algorithms can be applied to estimate velocity and clock drift of the user from Doppler shift information.
Before continuing, it is also necessary to introduce other definitions and notation. Vectors are represented with bold characters and matrices with bold capital letters. The L_2 norm (or Euclidean norm) notation of a vector $\boldsymbol{w}$ is:

$$\|\boldsymbol{w}\| = \sqrt{\boldsymbol{w} \cdot \boldsymbol{w}^T} \tag{3.2}$$

that will be used frequently as a squared L_2 norm:

$$\|\boldsymbol{w}\|^2 = \boldsymbol{w} \cdot \boldsymbol{w}^T \tag{3.3}$$

which is the sum of the squared elements of the vector.
Given a generic function in 3 variables $f(x, y, z)$, the gradient is the vector containing the first-order partial derivatives:

$$\boldsymbol{\nabla} f = \begin{bmatrix} \dfrac{\partial f}{\partial x} & \dfrac{\partial f}{\partial y} & \dfrac{\partial f}{\partial z} \end{bmatrix} \tag{3.4}$$

And the Hessian is instead a symmetric matrix containing the second-order partial derivatives:

$$\boldsymbol{H} = \boldsymbol{\nabla}^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x \partial x} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \vdots & \frac{\partial^2 f}{\partial y \partial y} & \frac{\partial^2 f}{\partial y \partial z} \\ \cdots & \cdots & \frac{\partial^2 f}{\partial z \partial z} \end{bmatrix} \tag{3.5}$$

Then, for a vector-valued function or a system of equations, the Jacobian matrix is:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{\nabla} f_1 \\ \vdots \\ \boldsymbol{\nabla} f_n \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dp_x} & \cdots & \frac{df_1}{dc_b} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{dp_x} & \cdots & \frac{df_n}{dc_b} \end{bmatrix} \tag{3.6}$$

whose rows are the gradients of the functions.

3.2 Least Squares

The idea of least squares is to find a solution to an overdetermined (or at least not underdetermined) system of linear equations in the form $\boldsymbol{Ax} = \boldsymbol{y}$ such that it minimizes the squared Euclidean norm of $\|\boldsymbol{y} - \boldsymbol{Ax}\|^2$, called residuals.

The discovery of the least squares method is one of the most famous disputes in the history of statistics, and perhaps in the whole of mathematics. Although the method is nowadays often attributed to Gauss, the first publication with a complete and concise exposition of least squares dates back to 1805 by Legendre, who also named the method [18]. Nevertheless, mathematicians had been concerned with the problem of estimating parameters from combinations of observed data since many decades before. Gauss later published his results and proofs about least squares in 1809, claiming that he first had the idea of the method as early as in 1794 (he was 17 at the time) [19]. This claim by Gauss is indeed supported by evidence regarding the results he obtained and his private correspondence [20]. An English translation of Gauss work is given in [21]. Significant work on the combination of observations can also be attributed to Laplace, from as early as 1783 [19].
Although both Legendre and Gauss developed their work in the context of astronomy, the method of least squares is also historically significant to the field of geodesy. An interesting tutorial about least squares and its relevance in the history of geodesy is given in [22].

3.2.1 Literature on Least Squares

There is a large body of literature concerning with least squares problem, including books from estimation theory [23], signal processing [24] and linear algebra [25–29], while [30] is entirely dedicated to solutions of least squares problems. In-depth analysis of linear regression models is given in [31].
The book by Bjorck [32] goes in-depth into numerical solutions to least squares problems and their properties, while also offering many proofs and historical background. Hansen [33] gives a comprehensive discussion of least squares methods and their application in data fitting problems.
The books [34, 35] focus on methods for optimization, while [36] offers a nice summary on the main descent algorithms and their properties. [37] explores approximate Gauss-Newton methods and their convergence properties. In-depth analysis of accuracy and stability of numerical methods is provided in [38] and [39], while [40] also offers examples and MATLAB code.
For more specific discussion, many papers concern with rank-deficient or ill-conditioned problems and other aspects such as optimal numerical solutions and properties regarding perturbation theory and convergence. The interested reader can refer to works from Golub [41], Stewart [42, 43] and Bjorck [44, 45].

The remainder of this discussion is based on these references and is intended to summarize important results which can be relevant to the field of GNSS.

3.2.2 Linearization of a System

Let us start by recalling the GNSS pseudorange equation in (2.6) which is a non linear function, so it needs to be linearized before applying any method from linear algebra.
Thanks to truncation of the Taylor expansion of the partial derivatives of f_i at a given linearization point, the system can be linearized as:

$$f_i\left(\boldsymbol{x}+\boldsymbol{\Delta x}\right)=\frac{\partial f_i}{\partial p_x}\Delta p_x+\frac{\partial f_i}{\partial p_y}\Delta p_y+\frac{\partial f_i}{\partial p_z}\Delta p_z+\frac{\partial f_i}{\partial c_b}\Delta c_b. \tag{3.7}$$

Then, the system can be represented in a linear form by building the Jacobian matrix, which in the case of GNSS pseudorange measurements becomes:

$$\boldsymbol{A}=\begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,m} & a_{y,1} & a_{z,1} & 1 \end{bmatrix} \tag{3.8}$$

where the a coefficients contain the negative partial derivatives of the Euclidean distance w.r.t. each position state:

$$a_{x,i}=\frac{p_{x,i}-p_x}{\|\boldsymbol{p}_i-\boldsymbol{p}\|} \tag{3.9}$$

and the coefficients of each row of $\boldsymbol{A}$ form what is usually referred to as the steering vector associated to a satellite. The reason why negative derivatives are taken will be more clear later when Newton's method is introduced. These coefficient represent what is obtained from a Direction Cosine Matrix (DCM), which gives the direction of each satellite from the approximation point. DCM are commonly used when dealing with rotations. In fact, a conversion between reference frames is done by computing the Jacobian of the coordinates of the target reference frame w.r.t. the original one. A useful property of the partial derivatives of Euclidean distance is:

$$(a_{x,i})^2+(a_{y,i})^2+(a_{z,i})^2=1 \tag{3.10}$$

which means that steering vectors are unit vectors pointing from the user (or its estimated position to be precise) to each satellite.

While this discussion is limited to considering the estimation of the position and clock bias from pseudoranges, the same can be done for the estimation of velocity and clock drift from the range rate equations. As a matter of fact, the $\boldsymbol{A}$ matrix from the latter case is the same as the one obtained for pseudorange measurements. Consider the range rate equation (2.10), we are interested in the partial derivatives:

$$\frac{\partial \dot{\rho}_i}{\partial \boldsymbol{v}} = \frac{\partial}{\partial \boldsymbol{v}} \left[(\boldsymbol{v}_i - \boldsymbol{v}) \frac{\partial \rho_i}{\partial \boldsymbol{p}} + \dot{c}_b \right]. \tag{3.11}$$

Since the range rate depends linearly on the velocity of the receiver, the derivative is the coefficient multiplying it, which is the negative derivative of the pseudorange w.r.t. the position. Therefore:

$$\frac{\partial \dot{\rho}_i}{\partial \boldsymbol{v}} = -\frac{\partial \rho_i}{\partial \boldsymbol{p}}. \tag{3.12}$$

The opposite sign between the two partial derivatives is because if the receiver moves toward the satellite, the pseudorange decreases while the Doppler shift increases. The partial derivatives w.r.t. clock drift are also equal to 1 as for pseudorange and clock drift.

Now that the system of equations has been linearized, it can be expressed as:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{3.13}$$

where $\boldsymbol{A}$ is an M by N matrix (Jacobian), $\boldsymbol{x}$ is an N by 1 vector of states (parameters) and $\boldsymbol{y}$ is an M by 1 vector of measurement (observations).
In particular, the true states are denoted with a bar $\bar{\boldsymbol{x}}$ and its corresponding noiseless measurements $\bar{\boldsymbol{y}}$. Then, the estimated states are represented by $\hat{\boldsymbol{x}}$ and the corresponding nominal measurement by $\hat{\boldsymbol{y}}$. The nominal measurements represent what noiseless measurements would be obtained if the true state corresponds to the estimated solution. Therefore, true values follow the relationship:

$$\boldsymbol{A}\bar{\boldsymbol{x}} = \bar{\boldsymbol{y}} \tag{3.14}$$

However, measurements are usually affected by noise ϵ:

$$\tilde{\boldsymbol{y}} = \bar{\boldsymbol{y}} + \boldsymbol{\epsilon} \tag{3.15}$$

so that $\tilde{\boldsymbol{y}}$ specifically refers to noisy measurements. Noise is assumed to have zero conditional mean $E[\boldsymbol{\epsilon} \mid \boldsymbol{A}] = 0$ (strict exogeneity) and a covariance matrix

$Cov[\boldsymbol{\epsilon} \mid \boldsymbol{A}] = \boldsymbol{C}$ which is symmetric and positive definite.
The system with noisy measurements can be solved using:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \tilde{\boldsymbol{y}} \tag{3.16}$$

which is known as the Ordinary Least Square (OLS), and will be demonstrated later. Note that if $\boldsymbol{A}$ is square and full rank, then the same solution can be obtained by simply inverting $\boldsymbol{A}$ instead.
Then, nominal measurements are then obtained as:

$$\hat{\boldsymbol{y}} = \boldsymbol{A}\hat{\boldsymbol{x}}. \tag{3.17}$$

In case the system is obtained from the linearization of non linear functions, as in the case we are addressing, nominal measurements can be obtained as $\hat{\boldsymbol{y}} = f(\hat{\boldsymbol{x}})$, although a non linear transformation can affect their statistical properties.
It is important to note that in general $\hat{\boldsymbol{y}} \neq \tilde{\boldsymbol{y}}$, and we will see in Section 3.7 why and in which cases they are instead equal. As a matter of fact, the differences between the two quantities are called **residuals**, whose squared norm is defined as:

$$\|\boldsymbol{r}\|^2 = \|\tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}}\|^2 \tag{3.18}$$

the squared norm of a vector $\boldsymbol{r}$ containing each residual, which is the difference between the input measurement and the nominal one. Equation (3.18) is sometimes also referred to as Sum of Squared Residuals (SSR).
The squared norm of the **estimation error** is defined as:

$$\|\boldsymbol{\zeta}\|^2 = \|\bar{\boldsymbol{x}} - \hat{\boldsymbol{x}}\|^2 \tag{3.19}$$

Finally, to complete the notation and definition of quantities of interest, the squared norm of the **fit error** is defined as:

$$\|\boldsymbol{e}\|^2 = \|\bar{\boldsymbol{y}} - \hat{\boldsymbol{y}}\|^2 = \|\boldsymbol{A}\left(\bar{\boldsymbol{x}} - \hat{\boldsymbol{x}}\right)\|^2 \tag{3.20}$$

which is the representation of the estimation error in the measurement domain. Having this definition of the error will be useful in Section 3.7 to infer its relationship to residuals and compare their statistical properties.

3.2.3 Mahalanobis Distance

Now that a formula for the solution of OLS is available, first we prove that using it in (3.16) does indeed minimize residuals. In order to do so, it is necessary to introduce the Mahalanobis distance. This metric evaluates the statistical distance of a point to a distribution. In our case, we are interested in the distance of the nominal measurements $\hat{\boldsymbol{y}}$ to a distribution with mean $\tilde{\boldsymbol{y}}$ and covariance matrix $\boldsymbol{C}$. The solution which minimizes residuals is the one which produces nominal measurement whose distance with the distribution is minimized. The Mahalanobis distance is defined as:

$$d_M(\hat{\boldsymbol{y}}, \tilde{\boldsymbol{y}}) = \sqrt{(\tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}})^T \boldsymbol{C}^{-1} (\tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}})} \tag{3.21}$$

An example of Mahalanobis distance is provided in Figure 3.1. To visualize this idea,

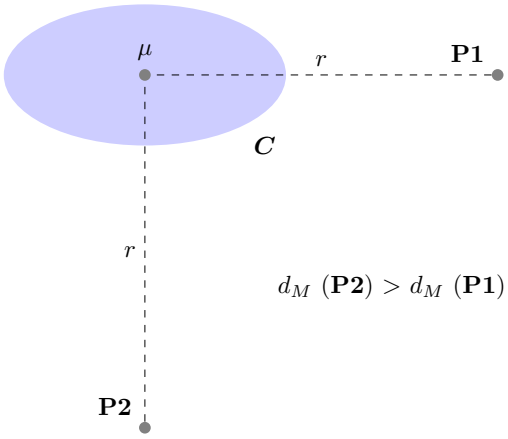


Fig. 3.1 Example of Mahalanobis distance. The light blue ellipse represents the covariance matrix $\boldsymbol{C}$ of a distribution with mean μ.

vectors can be thought of as sets of coordinates in a Cartesian space. Points P1 and P2 are both at the same distance r from the mean of the distribution μ. However, the Mahalanobis distance of P1 is smaller, since the variance of the distribution in its direction is larger. Another way to interpret this metric is that P1 is statistically closer because it is more likely to be a random sample drawn from the distribution w.r.t. P2.

We would like to find the estimate $\hat{\boldsymbol{x}}$ whose nominal measurement $\hat{\boldsymbol{y}}$ are statistically the closest to the true measurements. This implies that the realization of the error $\boldsymbol{\epsilon} = \tilde{\boldsymbol{y}} - \bar{\boldsymbol{y}}$ needed to produce the input measurements was the smallest, and therefore the most likely.

By assuming that noise follows a Gaussian distribution, the Maximum Likelihood

(ML) estimate is obtained with:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\ \Phi(\boldsymbol{x}) = (\tilde{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{x})^T \boldsymbol{C}^{-1} (\tilde{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{x}) \tag{3.22}$$

which can be developed into:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}\ \Phi(\boldsymbol{x}) = \left(\tilde{\boldsymbol{y}}^T \boldsymbol{C}^{-1} \tilde{\boldsymbol{y}}\right) + \left(\boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}\boldsymbol{x}\right) - \left(2\boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{C}^{-1} \tilde{\boldsymbol{y}}\right) \tag{3.23}$$

Since the aim is to find the minimum, the derivative of $\Phi(\boldsymbol{x})$ w.r.t. $\boldsymbol{x}$ can be taken and set to 0. The first term of (3.23) does not depend on the variable so it will disappear, then the equation becomes:

$$\begin{aligned} 2\boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{A}^T \boldsymbol{C}^{-1} \tilde{\boldsymbol{y}} &= 0 \\ \boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{A}^T \boldsymbol{C}^{-1} \tilde{\boldsymbol{y}} \end{aligned} \tag{3.24}$$

and the estimate is the value at which the derivative of $\Phi(\boldsymbol{x})$ w.r.t. $\boldsymbol{x}$ is equal to 0, so:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{C}^{-1} \tilde{\boldsymbol{y}} \tag{3.25}$$

which is often referred to as the Generalized Least Square (GLS). In the case in which all measurements have the same variance $\boldsymbol{C} = \sigma_U \boldsymbol{I}$, (3.25) simplifies to:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \tilde{\boldsymbol{y}} \tag{3.26}$$

so the OLS is Maximum Likelihood Estimate (MLE) under the assumption of Gaussian distributed errors.

Actually, so far we have only proven that (3.25) is a stationary point by taking the first derivative. The proof is be completed by taking the Hessian at the solution and showing that it is positive definite, and therefore the solution is a minimum. The Hessian at the solution can be obtained by differentiating again (3.24) w.r.t. $\hat{\boldsymbol{x}}$, which yields $\boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}$. If the columns of $\boldsymbol{A}$ are linearly independent, then the Hessian $\boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}$ is positive definite. Furthermore, if the cost function is convex, then the minimum is also a global minimum.

3.2.4 Gauss-Markov Theorem

The properties of the estimator derived in (3.26) are given in the Gauss-Markov theorem, which states that among the class of linear and unbiased estimators, the OLS is the one with minimum variance [32], meaning that it is the Best Linear Unbiased Estimator (BLUE). In particular, the theorem was derived under the following assumptions on the errors:

1. *Normal distributed.* This requirement is actually optional, but for many years it was believed to be a necessary condition [46]. Indeed, in his later work of 1819, Gauss made some advancements in the study of LS by removing the normality assumption [19]. This was later proven again by Markov [47].

2. *Zero mean.* The conditional expectation of errors w.r.t. to parameters is zero. Or in other words, the observations should be unbiased.

3. *Uncorrelated and identically distributed.* Which can be summarized as having spherical errors (since the covariance matrix collapses into a sphere). This requirement is relaxed by the GLS derived by Aitken [48] in 1935. In fact, using the GLS is equivalent to performing a linear transformation on the measurements based on $\boldsymbol{C}^{-1}$ with the purpose of normalizing and de-correlating errors. After the transformation, the errors are Independent Identically Distributed (IID) and Gauss-Markov theorem holds, so the OLS can then be applied.

So the GLS is also BLUE under the condition that $\boldsymbol{C}$ is known. This is often not the case in real applications, for which only an estimate of $\boldsymbol{C}$ is available. In particular, it is often assumed that the errors are uncorrelated and hence $\boldsymbol{C}$ is diagonal, in which case (3.25) is referred to as Weighted Least Square (WLS). The framework of estimation using weighted measurements was already considered by Gauss. Aitken later investigated the more general case for any $\boldsymbol{C}$ symmetric positive definite.
The fact that both OLS and GLS are linear is straightforward. A proof that they are unbiased is:

$$E\left[\hat{\boldsymbol{x}}\right] = \bar{\boldsymbol{x}} + \left(\boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T E\left[\boldsymbol{\epsilon}\right] \tag{3.27}$$

so if the error is unbiased $E\left[\boldsymbol{\epsilon}\right] = 0$ then so is the estimate $E\left[\hat{\boldsymbol{x}}\right] = \bar{\boldsymbol{x}}$.
The covariance of the estimator is given by:

$$Cov\left[\hat{\boldsymbol{x}}\right] = \left(\boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A}\right)^{-1} \tag{3.28}$$

and we will prove later in Section 3.5 that this is the Cramér-Rao Lower Bound (CRLB) for the problem. Therefore, the OLS has the minimum variance among the class of linear and unbiased estimators. It is possible to find estimators with lower variance if the unbiased property is dropped, as we will show in Section 3.3.1.
To be precise, it should be added that while the CRLB gives a limit for both linear and non-linear estimators, it is valid only in the case of normally distributed errors, which is why this requirement is often included in formulation of LS.
Note that if the additional assumption of normally distributed errors is kept, the OLS is also the MLE, as we have seen in Section 3.2.3. However, this assumption is not necessary to prove the Gauss-Markov theorem. On the contrary, if the noise is not Gaussian distributed, than OLS converges to the so-called misspecified CRLB and is no longer the MLE. In fact, if the noise distribution is not Gaussian, ML will likely yield a non-linear estimator instead. To summarize, under the assumptions of a full-rank linear model and zero mean uncorrelated identically distributed errors, the OLS is always BLUE, but is only MLE if the noise distribution is Gaussian.

One last remark is that, although both Legendre and Gauss were working with minimizing the squared residuals as in (3.22), this is not the only choice for the cost function which can be used to solve overdetermined systems of equations. As an example, Laplace was working on minimizing the L_1 norm of residuals instead (Least Absolute Deviations (LAD)), whose solution has different properties w.r.t. least squares. For example, if the errors have a Laplace distribution, then solving the LAD yields the MLE. Furthermore, since LAD works with residuals rather then their squares, the solution is less sensitive to large outliers in the data. However, the solution might not be unique and it is less stable compared to least squares. The advantage of working with the squared distance is that it is a strictly convex function with a continuous derivative. On the other hand, the distance is non-smooth (think of the absolute value function, which is non-differentiable near 0) and convex (but not strictly convex). It is easier to solve optimization problems using the squared distance as a cost function, because it can be differentiated to find a minimum. Furthermore, squaring is a monotonic function, which means that minimizing the squared distance is equivalent to minimizing the distance.

3.2.5 Newton's Method

If the system of equations described by $\boldsymbol{A}$ is linear, then a unique LS solution can be found directly by solving the system, provided that $\boldsymbol{A}$ is full rank. Otherwise,

as in the case of GNSS, it can be solved iteratively by exploiting Newton's method. The process consists in starting with an approximation point $\hat{\boldsymbol{x}}_0$ (an initial guess of the parameters), which is updated at each iteration. The idea behind this iterative process is that at each step, the new $\hat{\boldsymbol{x}}$ is closer to $\bar{\boldsymbol{x}}$, leading to a smaller linearization error and hence a progressively more accurate estimate.
In particular, Newton's method is an iterative process that allows to find a root of a real-valued function. It does so by using the value of the function at a given point (the current guess) and its derivative. In the scalar case, the correction term is equal to:

$$\Delta x = \frac{\Delta y}{\frac{\Delta y}{\Delta x}} = \frac{f(\hat{x})}{f'(\hat{x})} \tag{3.29}$$

which is then used to update the guess:

$$\hat{x}_{k+1} = \hat{x}_k - \Delta x. \tag{3.30}$$

Newton's method requires calculating the derivatives of the function whose root you want to find. In our case, we are interested in residuals, which are the difference between input measurements $\tilde{\boldsymbol{y}}$ (a constant) and the nominal measurements $\hat{\boldsymbol{y}}$. The latter are a function of the current guess, and more specifically the pseudorange function. Therefore, the derivative of residuals is the negative derivative of pseudoranges. This is why the negative derivatives where taken when calculating the $a_{x,i}$ coefficients in (3.9).
For systems of equations with multiple variables, the derivative in (3.29) is the Jacobian matrix of the system, for which the inverse needs to be found. If the Jacobian $\boldsymbol{A}$ is square, it can be inverted directly to obtain $\boldsymbol{\Delta x} = \boldsymbol{A}^{-1}\boldsymbol{\Delta y}$. Otherwise, the matrix is not invertible and a more general formula in (3.16) is needed.
The *pseudoinverse* $\boldsymbol{A}_*^{-1}$ should satisfy the Moore-Penrose conditions:

- Both $\boldsymbol{A}\boldsymbol{A}_*^{-1}$ and $\boldsymbol{A}_*^{-1}\boldsymbol{A}$ are Hermitian matrices (equal to their conjugate transpose). In our case, this is equivalent to being symmetric since $\boldsymbol{A}$ is a real matrix.
- $\boldsymbol{A}\boldsymbol{A}_*^{-1}\boldsymbol{A} = \boldsymbol{A}$, which means that $\boldsymbol{A}_*^{-1}\boldsymbol{A}$ maps columns of $\boldsymbol{A}$ to themselves.
- $\boldsymbol{A}_*^{-1}\boldsymbol{A}\boldsymbol{A}_*^{-1} = \boldsymbol{A}_*^{-1}$.

What is important is the meaning of these conditions, which can be summarized as:

- $\boldsymbol{A}_*^{-1}\boldsymbol{A}$ is the projection on the support of $\boldsymbol{A}$.

- $\boldsymbol{A}\boldsymbol{A}_*^{-1}$ is the projection on the image of $\boldsymbol{A}$.

which is useful because it gives a geometrical interpretation to the role of the pseudoinverse. If the Jacobian is real and its columns are linearly independent, the pseudoinverse can be found as the left inverse:

$$\boldsymbol{A}_*^{-1} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T \tag{3.31}$$

which mean the OLS in (3.16) can also be applied in an iterative fashion and that the pseudoinverse is optimal in the LS sense.
The requirement of linearly independent columns can be restated as not having *perfect multicollinearity*, otherwise $\boldsymbol{A}^T\boldsymbol{A}$ is not invertible. When the method of LS, as derived by Gauss, is applied iteratively using Newton's method, it is referred to as Gauss-Newton algorithm.
Given the meaning the of the Moore-Penrose equations, the *projection matrix* can be defined as:

$$\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}_*^{-1} = \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T \tag{3.32}$$

and consequently the *residual maker matrix* can be introduced as:

$$\boldsymbol{M} = \boldsymbol{I} - \boldsymbol{P} \tag{3.33}$$

since recalling (3.18) we have that $\boldsymbol{r} = \tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}} = \tilde{\boldsymbol{y}} - \boldsymbol{A}\hat{\boldsymbol{x}} = \tilde{\boldsymbol{y}} - \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\tilde{\boldsymbol{y}} = (\boldsymbol{I} - \boldsymbol{P})\tilde{\boldsymbol{y}}$. An important remark is that both $\boldsymbol{P}$ and $\boldsymbol{M}$ are *idempotent* matrices, which implies that $\boldsymbol{P}^2 = \boldsymbol{P}$ and equivalently $\boldsymbol{M}^2 = \boldsymbol{M}$ [49].
Giving a definition of the projection matrix allows us to make a graphical interpretation of the solution to the system, which is depicted in Figure 3.2. The noiseless true measurement $\bar{\boldsymbol{y}}$ (in blue) are affected by a disturbance ϵ, thus forming the input measurements $\tilde{\boldsymbol{y}}$. The solution to the system $\hat{\boldsymbol{y}}$ (or $\hat{\boldsymbol{x}}$, the two things are the same thing represented in two different domains connected by the linear map $\boldsymbol{A}$), is obtained as the closest point to $\tilde{\boldsymbol{y}}$ which belongs to the column space $C(\boldsymbol{A})$ formed by the coefficients of the linear equations. Considering that the system can only output solutions which are linear combinations of the basis vectors, the set of possible solutions is limited by the columns of $\boldsymbol{A}$. If the columns are linearly independent, they form a column space of N dimensions. On the other hand, the vector of M measurements exists in M dimensions with $M \geq N$. When the inequality holds, the dimensions of the column space are not sufficient to represent all possible realizations of the measurement vector. The remainder part that cannot be represented by the basis is what causes the residuals.

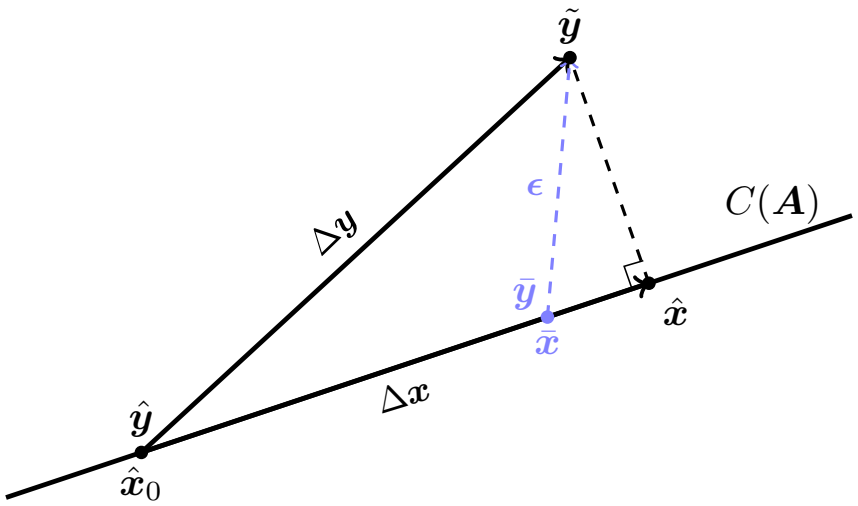


Fig. 3.2 The solution to an overdetermined system of equations can be interpreted as the projection of the input data over the column space formed by the matrix containing the coefficients of the linear equations.

A block diagram of the iterative LS framework is provided in Figure 3.3. In practical

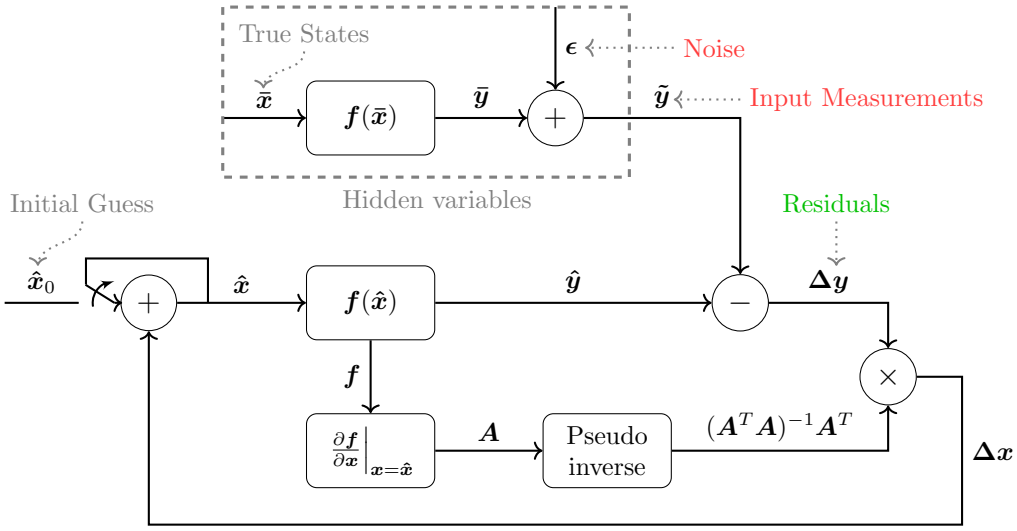


Fig. 3.3 Block Diagram of the iterative solution to least squares problem.

terms, the solution is obtained by setting an initial point $\hat{\boldsymbol{x}}_0$ and forming the vector $\Delta \boldsymbol{y}$ accordingly. The solution to the iteration is then $\Delta \boldsymbol{x}$, which is the projection of $\Delta \boldsymbol{y}$ onto the column space. The iterative process based on Newton's method can be thought of as a descent algorithm over a cost function, where at each iteration the residuals are (hopefully) smaller.

From a modern perspective, least squares can be thought of as a data fitting problem [33] or an optimization problem. There is a model (i.e., the equations governing

the system) and noisy data generated by the model is available. The task is finding which parameters of the model are most likely to have generated the data. In order to do so, a cost function is defined, namely the residuals, which quantifies the difference between the data that is observed and the one that the current guess of the parameters would generate. The objective is to minimize this cost function, and therefore provide an estimate of the parameters that would generate data that is as close as possible to the observed one.
In some cases, the exact equations that have generated the data are not known, and a model should be proposed so that it can be fit to the data. This is not a case we are going to consider, as in the context GNSS the equations are known in advance.

3.3 Non Linear Solutions to Least Squares

The previous formulation of least squares was linear, based on the truncation of the Taylor expansion at the first-order derivatives. Actually, (3.22) can be written in a general non-linear form:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}\ \Phi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{\Delta y}^T \boldsymbol{\Delta y}. \tag{3.34}$$

where $\boldsymbol{\Delta y} = (\tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}})$ and $\hat{\boldsymbol{y}}$ is a non linear function of the states. The factor of one half is considered for the sake of convenience. In fact, it is straightforward to see that minimizing the squared residuals yields the same result as minimizing half of the squared residuals. For the sake of simplicity, consider for now the covariance matrix of the error $\boldsymbol{C} = \boldsymbol{I}$. Once again, take the derivative of Φ and set it to zero. By exploiting the chain rule of the derivative:

$$\frac{\partial \Phi(\hat{\boldsymbol{x}})}{\partial \hat{\boldsymbol{x}}} = \frac{\partial \Phi(\hat{\boldsymbol{x}})}{\partial \boldsymbol{\Delta y}} \frac{\partial \boldsymbol{\Delta y}}{\partial \hat{\boldsymbol{x}}} = 0 \tag{3.35}$$

which yields:

$$\Phi'(\hat{\boldsymbol{x}}) = \boldsymbol{\Delta y}\ \boldsymbol{A}^T = 0 \tag{3.36}$$

and now the minimization problem has been turned into a root-finding problem, so Newton's method can be used to find the solution of (3.35), hence the state estimate at each iteration is updated by:

$$\boldsymbol{\Delta x} = \frac{\Phi'(\hat{\boldsymbol{x}})}{\Phi''(\hat{\boldsymbol{x}})}. \tag{3.37}$$

We can notice from (3.36) that Φ' is a product of the terms. Hence, we can find the second derivative by exploiting the following property:

$$\frac{\partial \Phi'(\hat{\boldsymbol{x}})}{\partial \hat{\boldsymbol{x}}} = \frac{\partial (f(\hat{\boldsymbol{x}})g(\hat{\boldsymbol{x}}))}{\partial \hat{\boldsymbol{x}}} = g(\hat{\boldsymbol{x}})\frac{\partial f(\hat{\boldsymbol{x}})}{\partial \hat{\boldsymbol{x}}} + \frac{\partial g(\hat{\boldsymbol{x}})}{\partial \hat{\boldsymbol{x}}} f(\hat{\boldsymbol{x}}) \tag{3.38}$$

with $f(\hat{\boldsymbol{x}}) = \boldsymbol{\Delta y}$ and $g(\hat{\boldsymbol{x}}) = \boldsymbol{A}^T$, which allows us to obtain:

$$\Phi''(\hat{\boldsymbol{x}}) = \boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{H}_f\boldsymbol{\Delta y} \tag{3.39}$$

where $\boldsymbol{H}_f = \frac{\partial g(\hat{\boldsymbol{x}})}{\partial \hat{\boldsymbol{x}}}$. It should be noted that size of the second term written in matrix form does not match with the first. What is important to notice for now is that there is an additional term that depends on the Hessian matrices of the functions. Therefore, Newton's method gives:

$$\boldsymbol{\Delta x} = \left(\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{H}_f\boldsymbol{\Delta y}\right)^{-1}\boldsymbol{A}^T\boldsymbol{\Delta y}. \tag{3.40}$$

First, let us properly develop the formula in (3.39) for the Hessian of the minimization problem:

$$\frac{\partial^2 \Phi(\hat{\boldsymbol{x}})}{\partial \hat{x}_i \partial \hat{x}_j} = \sum_{l=1}^{m} \frac{\partial f_l(\hat{\boldsymbol{x}})}{\partial \hat{x}_i}\frac{\partial f_l(\hat{\boldsymbol{x}})}{\partial \hat{x}_j} + \sum_{l=1}^{m} f_l(\hat{\boldsymbol{x}})\frac{\partial^2 f_l(\hat{\boldsymbol{x}})}{\partial \hat{x}_i \partial \hat{x}_j}. \tag{3.41}$$

where the subscript l refers to the function l-th residual.

One interpretation of Gauss-Newton algorithm is that it approximates the Hessian of the cost function $\boldsymbol{H}_\Phi$ as only the first term $\boldsymbol{H}_\Phi \approx \boldsymbol{A}^T\boldsymbol{A}$ of (3.39), thus assuming the second term to be zero or at least negligible. The advantage of Gauss-Newton is that it can obtain a good approximation of the second derivative without actually having to compute the second derivatives, which can be computationally expensive or hard to derive analytically. The downside is that, if the second term in (3.41) is not negligible, the linear approximation is not good. In fact, the approximated Hessian can become ill-conditioned and the correction term $\boldsymbol{\Delta x}$ is no longer guaranteed to point in a descent direction of the cost function. In general, it is good to have:

$$\left|\frac{\partial f_l(\hat{\boldsymbol{x}})}{\partial \hat{x}_i}\frac{\partial f_l(\hat{\boldsymbol{x}})}{\partial \hat{x}_j}\right| \gg \left|f_l(\hat{\boldsymbol{x}})\frac{\partial^2 f_l(\hat{\boldsymbol{x}})}{\partial \hat{x}_i \partial \hat{x}_j}\right| \tag{3.42}$$

so the algorithm is expected to converge under the following conditions:

- *Residuals* $f(\hat{\boldsymbol{x}}) = \boldsymbol{\Delta y}$ *are small.* This happens if the initial guess is reasonably close to the solution and the residuals are small at the solution.
- *Functions are almost linear.* So the second-order derivatives are small.

In other words, Gauss-Newton neglects the curvature of the cost function. If the curvature is non-negligible w.r.t. how far is the approximation point from the solution, then the linear approximation may not adequate.

Differently from (3.41), other descent algorithms such as gradient descent completely neglect the Hessian and only use the Jacobian:

$$\boldsymbol{\Delta x} = \gamma \, \boldsymbol{A}^T \boldsymbol{\Delta y} \tag{3.43}$$

where γ is the learning rate. While the negative gradient is the steepest descent direction locally, its global behaviour is not guaranteed. Furthermore, it may be slow to converge close to the solution and requires careful tuning of the learning rate. There exist a similar approach to adjust the step size for the Gauss-Newton algorithm, sometimes referred to as damped least squares:

$$\boldsymbol{\Delta x} = \alpha \, \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{\Delta y} \tag{3.44}$$

3.3.1 Levemberg-Marquardt Algorithm

To find a compromise between Gauss-Newton and gradient descent, the Levemberg-Marquardt algorithm was developed [50]:

$$\boldsymbol{\Delta x} = \left(\boldsymbol{A}^T \boldsymbol{A} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{A}^T \boldsymbol{\Delta y} \tag{3.45}$$

where λ is a positive scalar (or a vector of positive scalars). If $\lambda = 0$, (3.45) reduces to the Gauss-Newton algorithm. On the other hand, if λ grows larger, the correction term is more in the direction of steepest descent (with $\lambda \approx \gamma^{-1}$) but the step size becomes smaller since λ is at the denominator. There are many strategies to set and update λ at each iteration, some directly proposed by Marquardt [50].
A key idea from the Levemberg-Marquardt algorithm is to exploit *Tikhonov regularization*, thus allowing $\left(\boldsymbol{A}^T \boldsymbol{A} + \lambda \boldsymbol{I}\right)$ to be invertible even when $\left(\boldsymbol{A}^T \boldsymbol{A}\right)$ is not. It is clear to see that even if $det\left(\boldsymbol{A}^T \boldsymbol{A}\right) = 0$, then by adding positive scalars along the main diagonal the determinant is no longer zero.

Considering the Singular Value Decomposition (SVD), the regularized matrix has singular values:

$$\sigma_i\left(\left(\boldsymbol{A}^T\boldsymbol{A}+\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{A}^T\right)=\frac{\sigma_i(\boldsymbol{A})}{\sigma_i(\boldsymbol{A})^2+\lambda}. \tag{3.46}$$

We also need to introduce the condition number of a given matrix $\boldsymbol{B}$:

$$\kappa(\boldsymbol{B})=\frac{\sigma_{max}(\boldsymbol{B})}{\sigma_{min}(\boldsymbol{B})}=\|\boldsymbol{B}\|\|\boldsymbol{B}^{-1}\|\geq 1 \tag{3.47}$$

which is defined as the ratio between the largest and smallest of its singular values, or alternatively as the product between the matrix norms of the matrix and its inverse. The minimum value of the condition number is 1, namely when all singular values are equal (and therefore the eigenvalues). Conversely, its maximum value is positive infinite, namely when $\boldsymbol{B}$ is not full rank and the minimum singular value would be zero. In such case the problem is said to be *ill-posed*.
The condition number is a good measure of the sensitivity of the inverse to small changes in the original matrix. A large condition number is a consequence of approximate multicollinearity (i.e., poor geometry) and implies that even if a matrix is invertible, an algorithm may be unable to compute an inverse or it may be numerically inaccurate.
An important result of regularization is that it decreases the condition number. Since a positive scalar is summed to all singular values, then their ratio decreases. The condition number is an important concept because it is used in numerical analysis to assess the numerical stability and accuracy of algorithms.
The condition number is also tightly related to the concept of GDOP. The difference is that while the GDOP depends both on how well-conditioned the problem is (good geometry) and the number of measurements (visible satellites), the condition number is invariant on the number of measurements. In general, by fixing the number of measurements, a lower condition number implies a lower GDOP. The concept of GDOP and its meaning will be analysed more in detail later in Section 3.8.

Lagrange Multipliers

Coming back to Levemberg-Marquardt algorithm, there are other meaningful interpretation that can be given to the use of the term λ. In particular, consider the

constrained optimization problem:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\tilde{\boldsymbol{y}} - \boldsymbol{A}\hat{\boldsymbol{x}}\| \qquad \text{subject to } \|\boldsymbol{\Delta x}\|^2 \leq \alpha^2 \tag{3.48}$$

which can be transformed into an unconstrained problem with a new cost function thanks to the method of *Lagrange multipliers*, thus obtaining:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\tilde{\boldsymbol{y}} - \boldsymbol{A}\hat{\boldsymbol{x}}\| + \lambda \left(\|\boldsymbol{\Delta x}\|^2 - \alpha^2 \right). \tag{3.49}$$

By taking the derivative and setting it to zero as done previously, Levemberg-Marquardt algorithm is obtained, which is the solution minimizing the new cost function.
The effect of the term $\lambda \boldsymbol{I}$ is to constrain the correction $\boldsymbol{\Delta x}$ inside a trust region, possibly where the linear assumption can be considered to be valid. Furthermore, if $\boldsymbol{A}$ is rank deficient, than the least squares solution is not unique. In such case, proper setting of the initial guess and tuning of λ allows the algorithm to converge to a particular desired solution (minimum norm solution).
Furthermore, from the point of view of Bayesian statistics (discussed more in detail in Chapter 4), the term $\lambda \boldsymbol{I}$ acts like a prior distribution. As a matter of fact, one of the possible applications of Bayesian inference is to include regularization terms (or prior information) into the statistical models of ML estimation [51]. Differently from filtering solutions, which include prior information by combining the previous (unbiased) estimate of the filter and the dynamical model of the system, the regularization term introduced by the Levemberg-Marquardt algorithm is subjective and therefore biased. The tuning of λ is effectively a trade-off between the bias and the variance of the estimator.

3.4 Convergence and Numerical Solutions

To recap, the family of algorithms for solving least squares problem can be thought of as descent methods over a cost function. To reach convergence, it is necessary to ensure that the correction term $\boldsymbol{\Delta x}$ is in a descent direction, and that the step size is appropriate. If $\boldsymbol{\Delta x}$ points in the direction of the global minimum but the step is too large, the iteration might lead further away from the minimum w.r.t. the starting point.
A possible strategy is to check the norm of residuals at the new guess before applying the correction term. If they are larger than at the current guess, then the step size

should be reduced until a lower norm is found.
Regarding the selection of which algorithm to use, the choice should depend on how linear are the functions governing the system. Problems with larger non linearities require better approximations of the Hessian.
In GNSS, although the linear formulation of the least squares problem is typically adequate to converge to a solution for MEO constellation and terrestrial use, the algorithms presented in this section can be of use in scenarios where the equations have stronger non linearities or the problem is ill-conditioned.
In the context of Doppler positioning with LEO constellations, thus with stronger non linearities, early studies on the performance of GLS have shown that the algorithm cannot converge from a cold start condition (i.e., initial guess at the centre of the earth) [52, 53]. To ensure convergence, the algorithm requires some external assistance to properly initialize the filter from an initial guess within a few hundred kilometres from the user's location (depending on the orbital radius of the constellation). Furthermore, other ill-conditioned problems may arise in the use of GNSS for space applications due to the poor geometry.

A few more words should be spent to addressed the topic of numerical solutions to obtain the PVT. The OLS was presented in (3.26) by explicitly writing the correction term $\boldsymbol{\Delta x}$. However, it is usually a good practice to not invert matrices. It is both faster and more numerically accurate to solve the linear system instead $\boldsymbol{A}^T\boldsymbol{A}\Delta\boldsymbol{x} = \boldsymbol{A}^T\Delta\boldsymbol{y}$ called *normal equations*, especially if the problem is ill-conditioned. This is due to properties of algorithms regarding numerical stability, and how the condition number affects backward errors (a concept related to estimation errors). The suggested way to solve linear systems is by means of LU factorization, QR factorization or other efficient methods. A more in-depth discussion on numerical stability and about efficient and accurate numerical solutions of linear systems of equations is beyond the scope of this chapter. The interested reader can refer to [32, 39, 42, 38] for discussions on the algorithms to solve least squares problems and their properties.

3.5 Fisher Information and Cramér-Rao Bound

It is now of our interest to analyse the performance in terms of estimation accuracy of the algorithms presented so far. In order to do so, the multivariate Fisher information

matrix is introduced, defined as:

$$\mathbb{I}(\boldsymbol{x}) = E\left[\frac{\partial\, log(\boldsymbol{F})^T}{\partial\, \boldsymbol{x}} \frac{\partial log(\boldsymbol{F})}{\partial \boldsymbol{x}}\right] \tag{3.50}$$

where $E[\cdot]$ is the expected value and $\boldsymbol{F}$ is the multivariate likelihood.
The CRLB is the minimum variance of an unbiased estimator and can be found as the inverse of the Fisher information:

$$Cov[\hat{\boldsymbol{x}}] \geq \frac{1}{\mathbb{I}(\boldsymbol{x})}. \tag{3.51}$$

Under the assumption that the noise is normal distributed with zero mean, the joint measurement Probability Density Function (PDF) is:

$$\boldsymbol{F}(\boldsymbol{y}) = (2\pi)^{-\frac{N}{2}}\ det(\boldsymbol{C})^{-\frac{1}{2}}\ e^{-\frac{1}{2}(\boldsymbol{y}-\bar{\boldsymbol{y}})^T \boldsymbol{C}^{-1}(\boldsymbol{y}-\bar{\boldsymbol{y}})} \tag{3.52}$$

with covariance $\boldsymbol{C}$ and centred around the mean value $\bar{\boldsymbol{y}}$ which is the vector of true measurements, given by the functions:

$$\bar{\boldsymbol{y}}_i = f(\boldsymbol{x}) = \|\boldsymbol{p}_i - \boldsymbol{p}\| + c_b \tag{3.53}$$

Therefore, the log-likelihood can be computed as:

$$log(\boldsymbol{F}(\boldsymbol{y})) = -\frac{M log(2\pi)}{2} - \frac{log(det(\boldsymbol{C}))}{2} - \frac{1}{2}(\boldsymbol{y}-\bar{\boldsymbol{y}})^T \boldsymbol{C}^{-1}(\boldsymbol{y}-\bar{\boldsymbol{y}}) \tag{3.54}$$

where M is the number of dimensions of the likelihood. Then by taking the derivatives the entries of the Fisher information matrix can be computed according to [24]:

$$\mathbb{I}_{m,n}(\boldsymbol{x}) = \frac{1}{2} Tr\left\{\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial x_m}\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial x_n}\right\} + \left[\frac{\partial \bar{\boldsymbol{y}}}{\partial x_m}\right]\boldsymbol{C}^{-1}\left[\frac{\partial \bar{\boldsymbol{y}}}{\partial x_n}\right]. \tag{3.55}$$

A complete proof of this formula can be found in [54]. Under the assumption that the covariance $\boldsymbol{C}$ is not a function of the states (and hence the partial derivatives are zero), the entries can be simplified to:

$$\mathbb{I}_{m,n}(\boldsymbol{x}) = \left[\frac{\partial \bar{\boldsymbol{y}}}{\partial x_m}\right]\boldsymbol{C}^{-1}\left[\frac{\partial \bar{\boldsymbol{y}}}{\partial x_n}\right]. \tag{3.56}$$

Although it may not be true that the covariance is independent of the states, it is reasonable to assume that the derivatives $\frac{\partial \boldsymbol{C}}{\partial x_m}$ are very small and anyway they

are hard to compute analytically. Nonetheless, this assumption if often not stated explicitly when the BLUE is derived.
It can be noticed that the partial derivatives in (3.56) have already been computed when deriving the Jacobian. Actually, these are the positive derivatives of pseudo-range rather than the negative, but $(-\boldsymbol{A})^T(-\boldsymbol{A}) = \boldsymbol{A}^T\boldsymbol{A}$. Therefore, the Fisher information matrix can be written as:

$$\mathbb{I}(\boldsymbol{x}) = \boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A} \tag{3.57}$$

and the CRLB is:

$$Cov[\hat{\boldsymbol{x}}] \geq \left(\boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A}\right)^{-1} \tag{3.58}$$

but since the GLS is BLUE it becomes an equality. Then the Root Mean Squared Error (RMSE) can be found by taking the square root of the trace of (3.58).
A derivation of the covariance of the estimator is also given in [11, 14, 12, 15], without directly highlighting its connection to the Fisher information, which instead is presented in [13] but without its complete proof.

An important remark is that the observed Fisher information is defined as the Hessian of the optimization problem in a neighbourhood of the ML solution, for the non linear LS it is:

$$\mathbb{I}_{obs}(\boldsymbol{x}) = \boldsymbol{H}_\Phi = \boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A} + \boldsymbol{H}_f\,\boldsymbol{C}^{-1}\boldsymbol{\Delta y}. \tag{3.59}$$

Then, since the errors are zero mean, the residuals $\Delta\boldsymbol{y}$ at the ML solution are approximately zero and therefore both Gauss-Newton and the non linear least squares have the same information and hence the same covariance.
As a matter of fact, the solution found by any descent algorithm depends only on its cost function, which is the same for both algorithms, namely the squared residuals. Different algorithms will have different performances in terms of convergence speed and radius depending on their formulation (linear or not) and whether they use step-adjustment strategies, but they will all find the same solution provided that they converge.
On the other hand, Levemberg-Marquardt solves a different cost function obtained from the constrained problem. By taking the Hessian of the cost function (3.48) the Fisher information is:

$$\mathbb{I}(\boldsymbol{x}) = \boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A} + \lambda\boldsymbol{I}. \tag{3.60}$$

This highlights the role played by λ. By proper choice of the initial guess and constraining the norm of the correction term, prior information is added and the variance of the estimator is lower. It should be pointed out that in practical applications, values of λ tend to be very small and one should be careful in how the parameter is initialized and tuned.

3.5.1 Correlated Measurements

It can be of interest to investigate the impact of correlated measurements on the Fisher information and consequently on the variance of the estimation. Take the simple case of a single parameter q to be estimated and two available measurements. The variance of the estimation of the parameter is given by the First-Order Second-Moment (FOSM) method [55]:

$$\sigma_q^2 = \left(\frac{\partial q}{\partial y_1}\right)^2 \sigma_{y1}^2 + \left(\frac{\partial q}{\partial y_2}\right)^2 \sigma_{y2}^2 + 2\,\frac{\partial q}{\partial y_1}\,\frac{\partial q}{\partial y_2}\,\sigma_{y_1 y_2}. \tag{3.61}$$

which contains the partial derivatives of parameters w.r.t. measurements rather than the other way around. Therefore, it is written in the matrix form $\boldsymbol{B}^T\boldsymbol{C}\boldsymbol{B}$ rather than $\left(\boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A}\right)^{-1}$.

What is important to point out is that the contribution of the third term in (3.61) can be both positive and negative, differently from the other two terms that are always positive. This implies that, depending on whether the two measurements are positively or negatively correlated and the sign of their partial derivatives, the correlation can both increase or decrease the variance of the estimated parameter.

This result can be explained as follows: take the case in which both partial derivatives are positive. This means that when the value of any of the two measurements increases, the value of the estimated parameter also increases. Then, if the two measurements are negatively correlated, a positive error on one measurement is usually accompanied with a negative error on the other. This means that both measurements have the same effect on the parameter but when one tends to increase the other tends to the decrease. In other words, errors tend to compensate each other and the estimation of the parameter is more robust. As a matter of fact, for the example just described, the third term of (3.61) would be negative, thus resulting in a lower variance of the estimated parameter.

Although the impact of correlation on the estimation accuracy can be both positive and negative, a major drawback is that it is often hard to estimate and more

importantly it cannot be controlled. Instead, it is often a result of the specific conditions of an experiment.

3.6 Graphical Representation of Linear Systems

This section is dedicated to provide a different point of view on how linear systems are solved. In order to do so, a simple numerical example with reduced dimensions will be presented and solved both numerically and graphically. The purpose of representing systems in a visual fashion is to provide an intuitive understanding of some quantities and properties of OLS that will be useful in Section 3.7. Other visual linear algebra examples can be found in [56].
Let's introduce the simple example system:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

The measurement vector $\boldsymbol{y}$ can be thought of as a set of coordinates in space. In

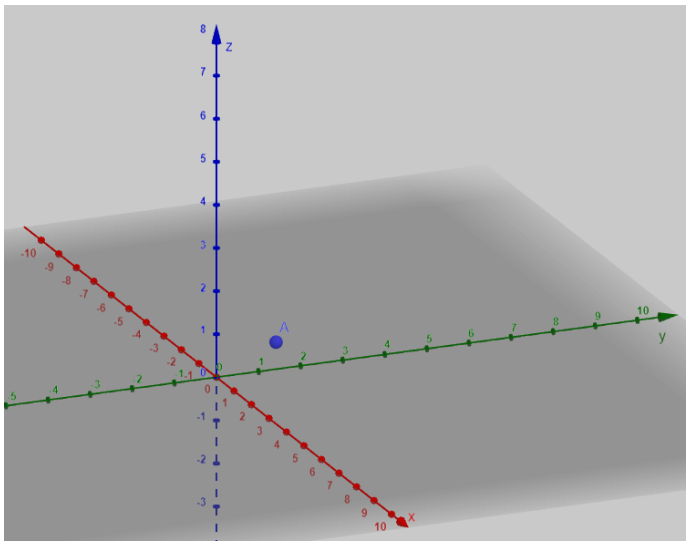


Fig. 3.4 Example of plotting a vector as a set of coordinates in space.

this case, it is a 3-dimensional space which is referred to as ambient space. An example of this concept is given in Figure 3.4. For now we will consider a covariance matrix $\boldsymbol{C}^{-1} = \boldsymbol{I}$ which is an identity matrix to focus on the graphical representation of solving the system, and then consider the more general case at the end of the

discussion. Given (3.26), then the relationship between states and measurements can be rewritten as:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \tilde{\boldsymbol{y}}$$
$$\hat{\boldsymbol{y}} = \boldsymbol{A}\hat{\boldsymbol{x}}$$

where the interest is in finding and plotting all the sets of input measurement $\tilde{\boldsymbol{y}}$ which yield an exact solution (zero residuals. This means they satisfy the condition $\hat{\boldsymbol{y}} = \tilde{\boldsymbol{y}}$. These vectors can be called *consistent*, in the sense that there are no inconsistencies in the input measurements which cause residuals. Another way to see it is that consistent sets of measurements are those for which there exist a solution that produces exactly those nominal measurements. On the other hand, inconsistent sets are only possible by the presence of random noise or other disturbances. Fortunately, the consistent sets do not need to be found by hand by trying all the combinations of $\tilde{\boldsymbol{y}}$ (which are infinite) and checking if the condition is satisfied. Instead, they can be represented graphically by a geometrical object whose basis is made from the columns of the matrix A. Let's introduce the notation:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_n \end{bmatrix}, \quad \boldsymbol{v}_i = \begin{bmatrix} a_1^i \\ \vdots \\ a_m^i \end{bmatrix}. \tag{3.62}$$

Then, $\boldsymbol{Ax}$ is a column space $C(\boldsymbol{A})$ and $(\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n)$ is a set of basis of $C(\boldsymbol{A})$. Since the $\boldsymbol{A}$ matrix of the example is 3 by 2, the column space is a set of two vectors which form a plane in 3 dimensional space. Given this definition, the example system (3.6)

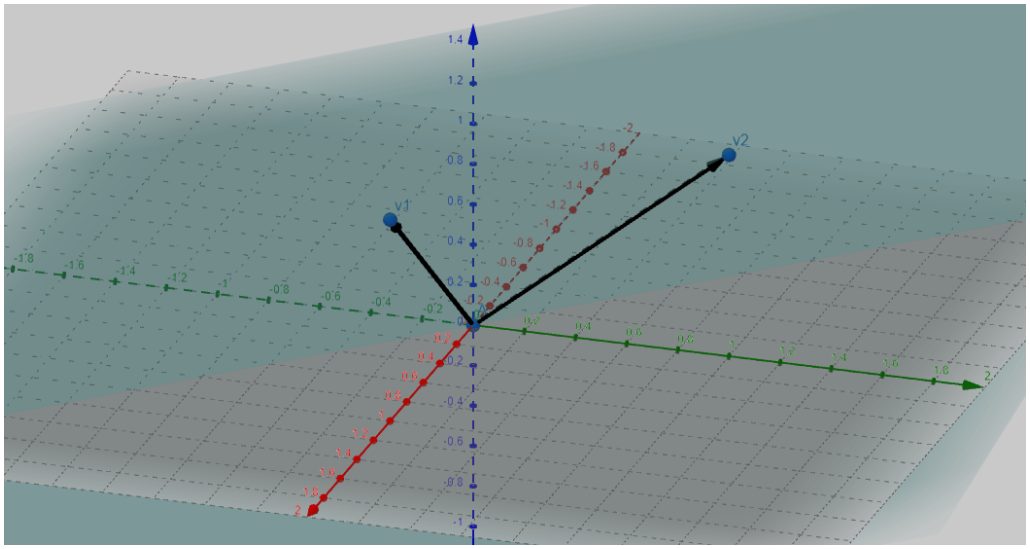


Fig. 3.5 Example plot of a column space.

can be represented visually as can be seen in Figure 3.5. The column space serves two purposes, the first is to depict graphically all the sets of consistent measurements (blue plane). The second is to act as a map between the measurement domain and the state domain.

The meaning is the following: take any point that belongs to the plane $C(\boldsymbol{A})$, the coordinates of the points in the basis of $C(\boldsymbol{A})$ represent the value of states, while the coordinates of that same point in the ambient space represent the corresponding nominal measurements obtained from those states. This means that any $\hat{\boldsymbol{x}}$ and its corresponding $\hat{\boldsymbol{y}} = A\hat{\boldsymbol{x}}$ are visually plotted in the same location. To obtain the value of one or the other it is simply necessary to consider its coordinates in one reference frame or the other (the ambient space for $\hat{\boldsymbol{y}}$ or the column space $C(\boldsymbol{A})$ for $\hat{\boldsymbol{x}}$).

Now that the visual representation of systems has been introduced, let us continue with then numerical example in order to show how the OLS obtains a solution. Set a value of true state and the corresponding noiseless measurements: $\bar{\boldsymbol{x}} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ and $\bar{\boldsymbol{y}} = A\bar{\boldsymbol{x}} = \begin{bmatrix} 1 & 1 & 2 \end{bmatrix}^T$. Consider input noise vector and therefore input measurements:

$$\boldsymbol{\epsilon} = \begin{bmatrix} 0.3 \\ 0.4 \\ -0.5 \end{bmatrix} \qquad \tilde{\boldsymbol{y}} = \bar{\boldsymbol{y}} + \boldsymbol{\epsilon} = \begin{bmatrix} 1.3 \\ 1.4 \\ 1.5 \end{bmatrix}.$$

The OLS finds the solution $\hat{\boldsymbol{x}}$ which minimizes residuals $\|\tilde{\boldsymbol{y}} - \boldsymbol{A}\hat{\boldsymbol{x}}\|$, which corresponds to finding the point on the plane $C(\boldsymbol{A})$ which is the closest to $\tilde{\boldsymbol{y}}$. Since the system is linear, it can be solved directly, but we will perform two iterations in order to highlight some properties.

Set an initial guess of the states $\hat{\boldsymbol{x}} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ and compute the corresponding predicted measurements $\hat{\boldsymbol{y}} = A\hat{\boldsymbol{x}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. Then, build a vector of residuals:

$$\boldsymbol{\Delta y} = \tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}} = \begin{bmatrix} 1.3 \\ 1.4 \\ 1.5 \end{bmatrix}.$$

and obtain the estimate by applying the OLS:

$$\Delta \boldsymbol{x} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \Delta \boldsymbol{y} = \begin{bmatrix} 0.9 \\ 1 \end{bmatrix}.$$

Since the problem is linear, the first iteration is enough to find the solution which minimizes residuals. In any case, let's perform another iteration to find out what

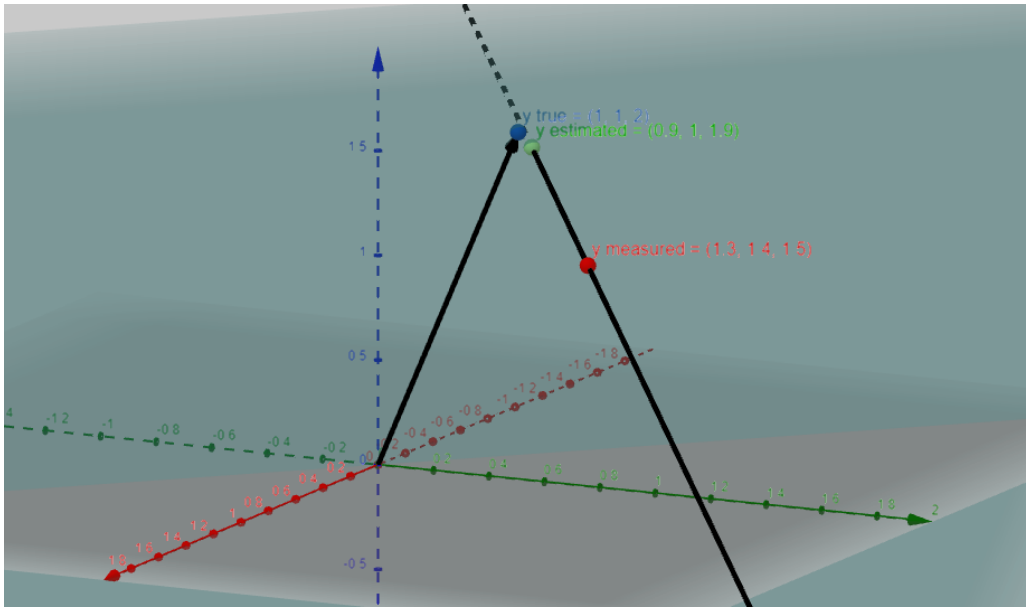


Fig. 3.6 Plot of the solution of the example. The estimated state (and therefore the predicted measurements) are the point on the column space that is the closest to the set of input measurements.

happens. First update the initial guess as the solution owe just obtained, therefore: $\hat{\boldsymbol{x}} = \Delta\boldsymbol{x} = \begin{bmatrix}0.9 & 1\end{bmatrix}^T$ and $\hat{\boldsymbol{y}} = A\hat{\boldsymbol{x}} = \begin{bmatrix}0.9 & 1 & 1.9\end{bmatrix}^T$. As a consequence, now the residuals are:

$$\Delta\boldsymbol{y} = \tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}} = \begin{bmatrix}0.4\\0.4\\-0.4\end{bmatrix}.$$

It is possible to verify that now:

$$\Delta\boldsymbol{x} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\Delta\boldsymbol{y} = \begin{bmatrix}0\\0\end{bmatrix}.$$

The correction term $\Delta\boldsymbol{x}$ represents how the estimate $\hat{\boldsymbol{x}}$ should change to reduce the residuals. Since we already found the closest solution and set it as our initial guess, it is expected that $\Delta\boldsymbol{x}$ is now zero. The reason this happens is because the residuals $\Delta\boldsymbol{y}$ are now part of the kernel (or null space). Indeed, it is possible to check that $\Delta\boldsymbol{y}$ is now orthogonal to any of the columns of $\boldsymbol{A}$. On the other hand, the fit error is:

$$\bar{\boldsymbol{y}} - \hat{\boldsymbol{y}} = \begin{bmatrix}0.1\\0\\0.1\end{bmatrix}$$

which is a linear combination of the columns of $\boldsymbol{A}$. In particular, $\bar{\boldsymbol{y}} - \hat{\boldsymbol{y}} = 0.1\,\boldsymbol{v}_1 + 0\,\boldsymbol{v}_2$ is true and these coefficients correspond exactly to the estimation error:

$$\bar{\boldsymbol{x}} - \hat{\boldsymbol{x}} = \begin{bmatrix} 0.1 \\ 0 \end{bmatrix}.$$

The numerical example has been solved by making the assumption $\boldsymbol{C}^{-1} = \boldsymbol{I}$. Let's now briefly discuss the more general case. It can occur in any linear system that different input measurements have different accuracies, meaning they are affected by noise with different variances. In such cases, the GLS is concerned with minimizing the Mahalanobis distance instead as we have seen. To translate this in graphical terms, this equates to applying a transformation to the ambient space. If $\boldsymbol{C}$ is diagonal, then every basis of $C(\boldsymbol{A})$ is scaled by the corresponding diagonal term of C. In other words, every axis of the space will be stretched or squeezed according to the elements of $\boldsymbol{C}$.

3.7 Noise, Errors and Residuals

From the discussion in Section 3.6, we obtained that after finding the solution to the system, residuals are orthogonal to the columns of $\boldsymbol{A}$, while the fit error is a linear combination of those.
This implies the following: if the noise vector affecting the measurement lies on the hyper-plane $C(\boldsymbol{A})$, then the measurement error will be affected $\|\boldsymbol{e}\|^2 = \|\boldsymbol{\epsilon}\|$ but not the residuals $\|\boldsymbol{r}\| = 0$. On the other hand, if the noise vector is orthogonal to the $C(\boldsymbol{A})$, then the opposite is true $\|\boldsymbol{e}\|^2 = 0$ and $\|\boldsymbol{r}\| = \|\boldsymbol{\epsilon}\|$. In general, the fit error is the component of noise parallel to $C(\boldsymbol{A})$, while residuals are the orthogonal component.
Therefore, the input error lies in the ambient space with the same dimension of $\boldsymbol{y}$, while residuals are in the kernel of $\boldsymbol{A}$ and the fit error maps to the image of $\boldsymbol{A}$ and the estimation error is its image. This result is a consequence of the *rank-nullity theorem*.

A visual representation of this concept can be seen in Figure 3.7. The estimation $\hat{\boldsymbol{x}}$ is the closest point of the column space to the input measurements $\tilde{\boldsymbol{y}}$. As such, residuals are orthogonal to the column space, while the fit error lies on it. In this example, $C(\boldsymbol{A})$ is one dimensional and the ambient space is two dimensional, so the kernel is also one dimensional (the one orthogonal to $C(\boldsymbol{A})$).
There is one more interesting observation that can be made from this analysis. When

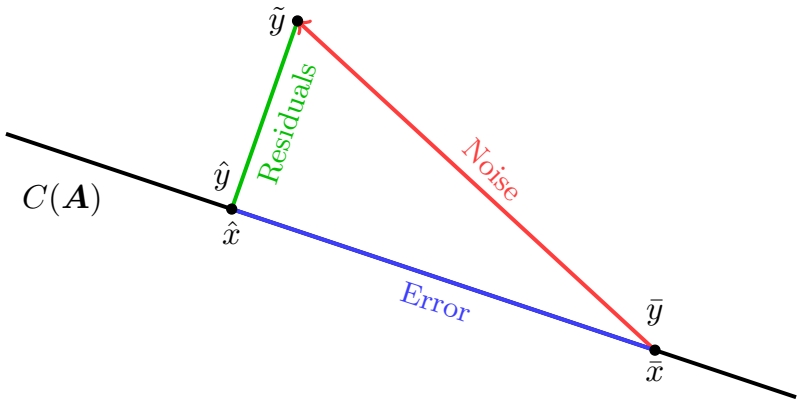


Fig. 3.7 2D representation of the concept of noise, errors and residuals.

the number of input measurements is equal to the number of states to be estimated $M = N$ and $\boldsymbol{A}$ is full rank, then the column space $C(\boldsymbol{A})$ has the same span as the ambient space. This implies that any $\tilde{\boldsymbol{y}}$ belongs to $C(\boldsymbol{A})$, so the GLS can find an $\hat{\boldsymbol{x}}$ for which $\hat{\boldsymbol{y}} = \tilde{\boldsymbol{y}}$ and the residuals are thus always 0. As a matter of fact, the kernel of a full rank square matrix only contains the zero vector.

In the example of Section 3.6, the column space is a 2D plane existing in a 3D ambient space. Since that system is overdetermined, there exist points that belong to the ambient space, but not to the plane. As a consequence, the GLS cannot always find solutions which have 0 residuals.

To give a simpler example, we can think of two polynomials of first degree in the form $y = ax + b$ in 2D space. Additive noise has an effect equivalent to changing the coefficient b, and thus shifting the line drawn from the corresponding equation. As long as the two lines are linearly independent (i.e., not parallel), they will still meet in exactly one point, regardless which or how much noise is added. This implies that there is always a solution with 0 residuals (the point where the lines meet). On the other hand, the same cannot be said about the intersection of 3 lines in 2D space. Depending on the specific noise added, the lines are not longer guaranteed to meet in exactly one point. In this second case, the system is overdetermined and the linear combination of the two coefficients a and b is not sufficient to represent all possible realizations of the noise vector (which contains 3 elements). The set of realizations of noise which cannot be represented by the linear combination of the parameters are all the points which lie outside of the column space in our visual example of Section 3.6.

Figure 3.7 also highlights what the OLS is attempting to do. It is easy to see that $\hat{\boldsymbol{y}}$ is the set of nominal measurements which produces the smallest residuals. If $\hat{\boldsymbol{x}}$ was indeed the real solution to the system, then the input noise needed to produce $\tilde{\boldsymbol{y}}$

is the smallest it can be, and therefore this solution is more likely than any other. A solution with large residuals would not explain the measurement very well, as it could only possible with a large realization of the noise, which is less likely. The OLS is thus trying to minimize the portion of noise that cannot be explained by the linear system.
Given the right-triangle in Figure 3.7, the relationship between noise $\boldsymbol{\epsilon}$, fit error $\boldsymbol{e}$ and residuals $\boldsymbol{r}$ can be described by the Pythagorean theorem:

$$\|\boldsymbol{\epsilon}\|^2 = \|\boldsymbol{e}\|^2 + \|\boldsymbol{r}\|^2. \tag{3.63}$$

where $\|\boldsymbol{\epsilon}\|^2$ is referred in ANalysis Of VAriance (ANOVA) as the *total sum of squares*, while $\|\boldsymbol{e}\|^2$ is the *explained sum of squares* and $\|\boldsymbol{r}\|^2$ is the *residual sum of squares*, as we have already anticipated.
The coefficient of determination is then defined as:

$$R^2 = \frac{\|\boldsymbol{e}\|^2}{\|\boldsymbol{\epsilon}\|^2} = 1 - \frac{\|\boldsymbol{r}\|^2}{\|\boldsymbol{\epsilon}\|^2}. \tag{3.64}$$

R^2 represents the amount of variance in the estimated parameters that can be explained by the variance in the observations. If the residuals $\|\boldsymbol{r}\|^2$ are zero, then the model perfectly explains the observed data and $R^2 = 1$. On the other hand, if the model can perfectly guess the true values $\bar{\boldsymbol{y}}$, then the fit error is zero and $R^2 = 0$.
The fraction on the right side of (3.64) is also referred to as *fraction of variance unexplained*.

3.7.1 Impact of Disturbances

It is of interested to study the impact on the solution of different kinds of input disturbances. Start from the relationship $\boldsymbol{A}\bar{\boldsymbol{x}} = \bar{\boldsymbol{y}}$ and consider that noise will shift the solution by an unknown quantity $\Delta\boldsymbol{x}$, then:

$$\boldsymbol{A}\left(\bar{\boldsymbol{x}} + \Delta\boldsymbol{x}\right) = \left(\bar{\boldsymbol{y}} + \boldsymbol{\epsilon}\right) \tag{3.65}$$

and since the system is linear:

$$\boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{A}\Delta\boldsymbol{x} = \bar{\boldsymbol{y}} + \boldsymbol{\epsilon}. \tag{3.66}$$

We can use the fact that $\boldsymbol{A}\bar{\boldsymbol{x}} = \bar{\boldsymbol{y}}$ to simplify on both sides to obtain $\boldsymbol{A}\Delta\boldsymbol{x} = \boldsymbol{\epsilon}$ and finally apply BLUE:

$$\Delta\boldsymbol{x} = \left(\boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{\epsilon} \tag{3.67}$$

which means the noise and its impact on the solution can be isolated. We can therefore neglect the measurement and work with increments to focus on the impact of noise instead.

Gaussian Noise

First, let's analyse the impact of Gaussian Noise. Contrary to the results obtained from Gauss-Markov theorem, the results that follow need the assumption of normality of the noise. In particular, consider noise with zero mean and variance $\boldsymbol{\epsilon}_N \sim \mathcal{N}(0, \sigma_U)$ so the covariance matrix is a diagonal matrix $\boldsymbol{C} = \sigma_U\boldsymbol{I}$.
Given the definitions of projection matrix in (3.32) and of residuals maker matrix in (3.33), the covariance of residuals can be written as:

$$Cov[\boldsymbol{r}] = \boldsymbol{M}^T\boldsymbol{C}\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{P})^T\boldsymbol{C}(\boldsymbol{I} - \boldsymbol{P}) \tag{3.68}$$

Under the assumption of IID, the squared norm of the input noise vector is distributed as $\sigma_U\,\chi^2_M$. The fit error is an M-elements vector obtained from the projection of noise over the column space $C(\boldsymbol{A})$, which spans N dimensions, so its squared norm is distributed as χ^2_N. Residuals are the remainder so they are distributed as χ^2_{M-N} [13]. Where χ^2_L is a chi-squared distribution with L degrees of freedom.
Fit errors and residuals are M by 1 vectors with the same mean norm as a vector of IID noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_U)$ with N and $(M - N)$ elements respectively.
When the number of measurements M increases, the model does not have enough basis functions (columns of $C(\boldsymbol{A})$) to fit the realization of noise. On the other hand, the estimate is less sensitive to a particular element of the noise. As a result, the fit error of each measurement approaches zero variance, while residuals approach the same variance of input measurements.
It should be mentioned that if the errors are not zero mean, the distributions become non-central chi-squared and the analysis becomes more complex. Such case is briefly discussed in [57].
Otherwise, the following relationship can be written:

$$M\,\sigma_r^2 = (M - N)\,\sigma_U^2 \tag{3.69}$$

where σ_r^2 is the variance of residuals. Based on the definition of the central chi-square of degree M as a sum of M squared normal distributions, we can write:

$$\|\boldsymbol{r}\|^2 = M\,\sigma_r^2 \tag{3.70}$$

and substitute it to the left side of (3.69), which leads to:

$$\|\boldsymbol{r}\|^2 = (M-N)\,\sigma_U^2. \tag{3.71}$$

Equation (3.71) can be inverted to find an estimate of the standard deviation:

$$\hat{\sigma}_U = \sqrt{\frac{\|\boldsymbol{r}\|^2}{M-N}} \tag{3.72}$$

which is a result also known in GNSS [13] and used in Receiver Autonomous Integrity Monitoring (RAIM) applications [58]. The estimation of variance from residuals in the context of least squares is also discussed in [31].
This allows us to make some observations about the use of residuals:

- The number of measurements should be taken in consideration since it affects the distribution of residuals. If the system is highly overdetermined, then there is a large number of degrees of freedom and the variance of residuals approaches that of input measurements.
- It a time series is available, using the mean norm or residuals over a long series of data can provide more accurate results.
- The statistical distribution of residuals and fit error is independent of satellite geometry. The only thing affected by geometry is how the norm of fit error translates into estimation error.
- If the observations are free from biases and other disturbances, then σ_U can be estimated from residuals. However, the estimator in (3.72) is unbiased but does not attain the CRLB [59].

If σ_U is already estimated in some other way (e.g. from $\frac{C}{N0}$ or time series of pseudoranges), the quantity in (3.72) can be normalized to be written as:

$$\chi_d^2 = \frac{\sigma_U^{-2}\|\boldsymbol{r}\|^2}{M-N} \tag{3.73}$$

which is known as the *reduced chi-squared statistic*. In the case of GLS, the numerator in (3.73) should be computed as $\boldsymbol{r}^T\boldsymbol{C}^{-1}\boldsymbol{r}$.
The metric χ^2_d is a chi square per degree of freedom. A value of χ^2_d around 1 indicates that the observed variance of residuals is in accordance with the estimated or known variance of the error on the observations. A value below 1 means that the variance of residuals is small compared to that of observation, suggesting that the latter could have been overestimated. In cases where the exact model is not known a priori, $\chi^2_d < 1$ can indicate that the model is fitting the noise rather than the parameters of the model, namely that it is overfitting. On the other hand, a value above 1 can indicate that the fit has not fully captured the observation, that the variance of measurements has been underestimated, or that input measurements are affected by biases.
The reduced chi-squared statistic may also be used to refine the estimation of the variance a posteriori $\hat{\sigma}^2_U = \sigma^2_U\chi^2_d$.
An overview of residuals and of the various metrics used in ANOVA can be found in [59–61]. Although the discussion is often limited to the regression of only one or two parameters, the references offer plenty of examples and insights on the use and meaning of these metrics.
A numerical example is given in Appendix A to verify and elaborate further on the results presented in this section.

Given the above discussion on the distribution of the norm of residuals, there is one last remark that can be made about each residual. Let us consider the covariance matrix of residuals from (3.68). Recall that the matrix $\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{P})$ is a square matrix, and that more importantly it is an idempotent matrix and thus $\boldsymbol{M}^2 = \boldsymbol{M}$. If we assume that $\boldsymbol{C} = \sigma_U\boldsymbol{I}$, then (3.68) can be rewritten as:

$$Cov\left[\boldsymbol{r}\right] = \sigma^2_U\left(\boldsymbol{I} - \boldsymbol{P}\right). \tag{3.74}$$

Then, if we are interested in the standard deviation of the i-th residual, it is given by:

$$\sigma_{r_i} = \sigma_U\sqrt{1 - p_{ii}} \tag{3.75}$$

where p_{ii} is the i-th diagonal element of $\boldsymbol{P}$ and is called *leverage* in statistics. Note that, because of the properties of idempotent matrices, $0 < p_{ii} < 1$. Measurements which correspond to high values of leverage are called *influential observations*. This result has the important implication that, even when all input measurements have

the same variance, not all residuals have the same variance, purely due to a factor which depends on the geometry of the problem. When doing any kind of hypothesis test to identify possible outliers, it is fundamental to consider the role played by leverage since it affects the distribution of residuals. Remember that $\hat{\boldsymbol{y}} = \boldsymbol{P}\boldsymbol{y}$, so each predicted measurement is given by:

$$\hat{y}_i = p_{ii} y_i \tag{3.76}$$

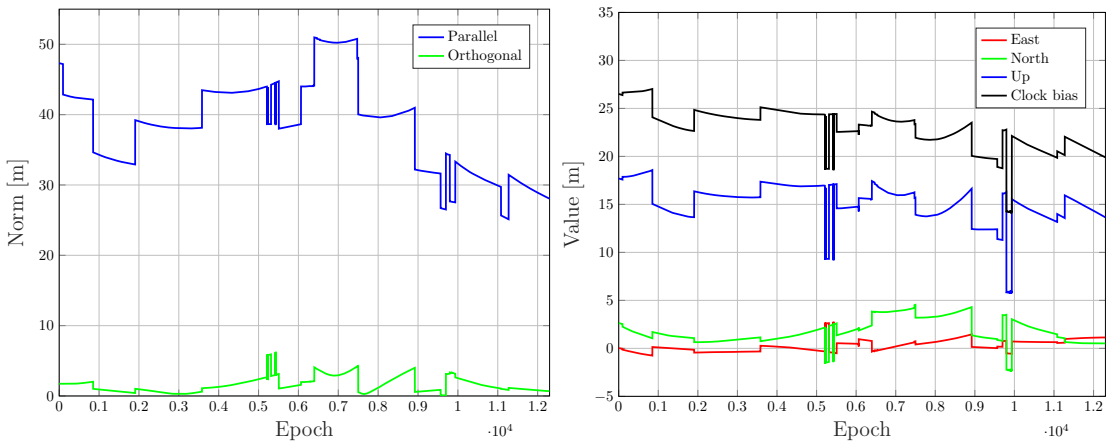
which means that $\frac{\partial \hat{y}_i}{\partial y_i} = p_{ii}$. So, the leverage can be thought of as the degree to which the i-th measurement influences the predicted value.
Equation (3.75) is important because it allows us to obtain a *studentized residual* as:

$$t_i = \frac{r_i}{\sigma_{r_i}} = \frac{r_i}{\sigma_U \sqrt{1 - p_{ii}}} \tag{3.77}$$

where we use (3.72) as an estimate for the standard deviation of input measurement. In particular, if the i-th measurement is a potential outlier to be tested, it can be safer to estimate the standard deviation excluding the corresponding residual. Then (3.72) can be rewritten as:

$$\hat{\sigma}_{U,(i)} = \sqrt{\frac{1}{M - N - 1} \sum_{j=1, j \neq i}^{M} r_j^2} \tag{3.78}$$

where $\hat{\sigma}_{U,(i)}$ represent the estimation of the standard deviation when excluding the i-th measurement.
If (3.78) is used in (3.77), then the corresponding $t_{(i)}$ is called *externally studentized* residual, as opposed to an *internally studentized* residual t_i which is obtained when using (3.72) instead.
Internally studentized residuals follow a Tau distribution, which is not very commonly used. On the other hand, externally studentized residuals are useful because they follow a Student's t-distribution with $M - N - 1$ degrees of freedom [49]. By central limit theorem, when the number of measurements M is large, then $t_{(i)} \sim \mathcal{N}(0, 1)$ which makes hypothesis testing more simple and intuitive.
Obtaining a rigorous statistical framework for the analysis of residuals and the impact of geometry is important because it allows to develop more advanced techniques for the detection of outliers and influential observations in linear regression model, as investigated in [62, 63].



(a) (b)

Fig. 3.8 (a) Orthogonal and parallel components of ionosphere delay error w.r.t. the column space. (b) Impact of ionosphere delay on estimation error of each component.

Deterministic Errors

After analysing the impact of random noise on the solution, it is interesting to perform a similar analysis on a deterministic source of error. In particular, it can be of interest for GNSS applications to consider the error due to Ionosphere delay $\boldsymbol{\epsilon}_I$, since it is one of the largest sources of error. The analysis is performed using a matrix $\boldsymbol{A}$ and a vector $\boldsymbol{\epsilon}_I$ obtained from correction models, both from a real dataset. From Figure 3.8 (a) it can be seen that the norm of the component of $\boldsymbol{\epsilon}_I$ parallel to $C(\boldsymbol{A})$ is much larger than the component orthogonal to it. This suggests that the ionosphere error mostly affects the estimation error rather than the residuals. This makes sense considering that the residuals are due to inconsistencies in the observations. On the contrary, the ionosphere delay is deterministic and affects all measurements by a certain amount, with only differences depending on which section of the ionosphere is crossed by the signal. Since all signals cross at least a minimum length section of ionosphere, they are all delayed by a minimum amount.
Furthermore, the impact on the estimation error for each state can be computed. Results are shown in Figure 3.8 (b). As it can be seen, ionosphere error mostly affects the estimation of clock bias and up direction (in East-North-Up (ENU) reference frame). In order to interpret this results, we have to go back to the graphical representation of the system in Section 3.6. Remember that the column space $C(\boldsymbol{A})$ is a basis made from the columns of the matrix $\boldsymbol{A}$. In particular, the last column corresponding to the clock bias is made of all ones. As we discussed, the delay caused

by the ionosphere is similar between satellites, so we have:

$$\begin{aligned} \boldsymbol{\epsilon}_I &= [\,\epsilon_{I,i}\,], \quad i = 1, \cdots, M \\ \epsilon_{I,i} &\simeq \epsilon_{I,j} \qquad \forall i, j \end{aligned} \tag{3.79}$$

which means the angle between the vector $\boldsymbol{\epsilon}_I$ and the last column of $\boldsymbol{A}$ is generally small. Depending on the geometry, the angle between $\boldsymbol{\epsilon}_I$ and the third column of $\boldsymbol{A}$ (for the up direction) can also be small. This phenomenon is stronger if satellites are close to the zenith, or in other words there are no visible satellites at low elevations. In such scenario, the coefficients of the partial derivatives w.r.t. p_z will all be similar. In conclusion, the ionosphere delay (if not corrected), will mostly affect the error on the clock bias and vertical coordinate as highlighted by Figure 3.8.

3.8 Geometric Dilution Of Precision

The purpose of this section is to first discuss the meaning of GDOP and why it is used in GNSS. A nice introduction to the concept of GDOP and its meaning can be found in [64]. Then, a lower bound of the GDOP as a function of visible satellite will be derived. The steps used to obtain the bound will allow us to extract some remarks on the requirements needed to obtain it such minimum value, both from the point of view of estimation theory and also of satellite geometry. This will be useful to gain a deeper understanding of the positioning problem.
It is important to mention that the notation that will be used is:

$$\boldsymbol{G} = \boldsymbol{H}^{-1} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}. \tag{3.80}$$

And consequently the GDOP is obtained as:

$$\text{GDOP} = \sqrt{Tr(\boldsymbol{G})}. \tag{3.81}$$

3.8.1 Introduction to GDOP

Let's begin this discussion from the BLUE (3.25), as mentioned previously, the variance of the estimator is given by [11]:

$$Cov\,[\hat{\boldsymbol{x}}] = \left(\boldsymbol{A}^T\boldsymbol{C}^{-1}\boldsymbol{A}\right)^{-1} \tag{3.82}$$

then, under the assumption that all input measurement are uncorrelated and have the same variance $\boldsymbol{C} = \sigma_U \boldsymbol{I}$, (3.82) becomes:

$$\sigma_{\hat{\boldsymbol{x}}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \sigma_U. \tag{3.83}$$

which eventually leads to:

$$\text{RMSE} = \text{GDOP} \cdot \sigma_U \tag{3.84}$$

where GDOP is the square root of the trace of $\boldsymbol{G}$. Equation (3.84) allows us to split the estimation error in terms of RMSE as two contributions: one due to the geometry and one due to measurement noise. In case the assumption of equal variances does not hold, it is reasonable to take σ_U as the average of the measurement variances. For now, consider the simple case $\boldsymbol{C} = \boldsymbol{I}$, then using (3.13) we can write:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \left(\bar{\boldsymbol{y}} + \boldsymbol{\epsilon}\right) \tag{3.85}$$

where $\tilde{\boldsymbol{y}} = \bar{\boldsymbol{y}} + \boldsymbol{\epsilon}$ can be substituted to get $\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \left(A\bar{\boldsymbol{x}} + \boldsymbol{\epsilon}\right)$, and the first term on the right side simplifies to $\hat{\boldsymbol{x}} = \bar{\boldsymbol{x}} + \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{\epsilon}$, which finally can be rearranged into:

$$\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}} = \boldsymbol{G}\, \boldsymbol{A}^T \boldsymbol{\epsilon}. \tag{3.86}$$

Notice that $\|\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}\| = \|\bar{\boldsymbol{x}} - \hat{\boldsymbol{x}}\|$ which is the estimation error. The term $\boldsymbol{A}^T \boldsymbol{\epsilon}$ lumps the input noise on each state, which means $\boldsymbol{G}$ is an amplification factor of the error. Given the definition of $\boldsymbol{H}$ in (3.80) as:

$$\boldsymbol{H} = \boldsymbol{A}^T \boldsymbol{A} = \left[\boldsymbol{v}_1 \cdots \boldsymbol{v}_N\right]^T \left[\boldsymbol{v}_1 \cdots \boldsymbol{v}_N\right] \tag{3.87}$$

where $\boldsymbol{v}_i$ is the i-th column of $\boldsymbol{A}$. Then:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{v}_1^T \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_N^T \boldsymbol{v}_1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{v}_1^T \boldsymbol{v}_N & \cdots & \boldsymbol{v}_N^T \boldsymbol{v}_N \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{v}_1\| & & \\ & \ddots & \\ & & \|\boldsymbol{v}_N\| \end{bmatrix} \tag{3.88}$$

and it is important to notice that the diagonal of $\boldsymbol{H}$ contains the norm of the columns of $\boldsymbol{A}$, while the off-diagonal terms contain the dot products between different columns. Given this, it is easy to notice that $\boldsymbol{H}$, and therefore also $\boldsymbol{G}$, are symmetric and positive definite matrices.

A useful way to understand the problem is to think of $\boldsymbol{H}$ as representing information about the system, and conversely $\boldsymbol{G}$ representing uncertainty. This notion is consistent

to the results obtained in Section 3.5 when discussing the Fisher information and CRLB for the least squares problem.
One direct consequence is that if any measurement (a row of $\boldsymbol{A}$) is added, the norm of the column is necessarily greater or equal than before, which means that more information about the parameters is available.

3.8.2 Lower Bound of GDOP

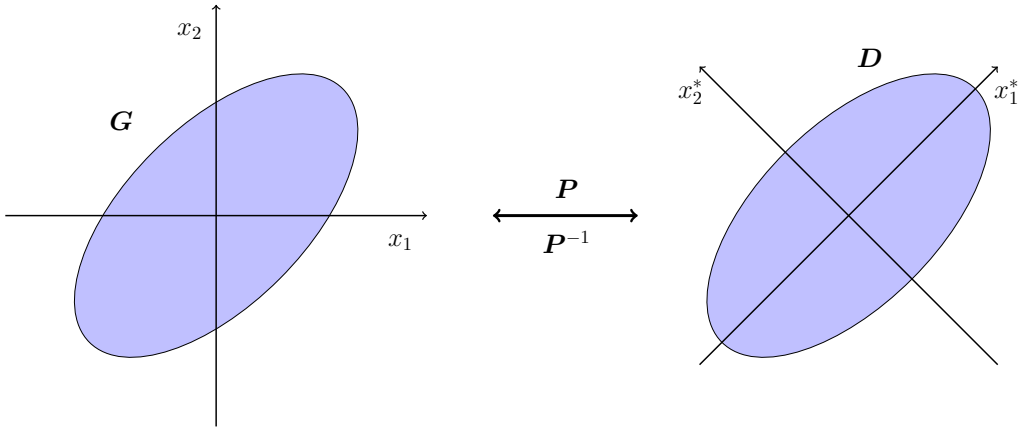


Fig. 3.9 Graphical representation of the diagonalization of a matrix.

Now, we shift the focus on deriving a lower bound for the GDOP as a function of the number of measurements. We will consider the simple case of using GNSS to estimate the user position and clock bias. We set $N = 3$ as the number of spatial dimensions. Start from the *spectral theorem*:

$$\boldsymbol{G} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1} \tag{3.89}$$

which allows to diagonalize $\boldsymbol{G}$ under the condition that it is a *normal* matrix. In this case, $\boldsymbol{G}$ is symmetric and positive definite which implies it is also normal. As a consequence, the spectral theorem states that $\boldsymbol{G}$ and $\boldsymbol{D}$ are *similar* matrices. There are a lot of shared properties between similar matrices, but the one we are interested about is that they have the same trace:

$$Tr(\boldsymbol{G}) = Tr(\boldsymbol{D}). \tag{3.90}$$

On the other hand, $\boldsymbol{P}$ is a unitary matrix which acts as a change of basis operator, which can also be viewed as a rotation.
Furthermore, it is possible to notice that (3.89) is also the *eigendecomposition* of $\boldsymbol{G}$,

where the columns of $\boldsymbol{P}$ contain the *eigenvectors* of $\boldsymbol{G}$ and $\boldsymbol{D}$ is a diagonal matrix with the corresponding *eigenvalues* $\lambda_i^{\boldsymbol{G}}$ on the main diagonal. Therefore:

$$Tr(\boldsymbol{D}) = \sum_{i}^{N} \lambda_i^{\boldsymbol{G}} \tag{3.91}$$

and thanks to (3.90) we can write:

$$\text{GDOP} = \sqrt{\sum_{i}^{N} \lambda_i^{\boldsymbol{G}}} \tag{3.92}$$

so GDOP is the square root of the sum of the eigenvalues of $\boldsymbol{G}$. Given a covariance matrix, eigenvalues represent the variability of data in an orthogonal basis spanned by the eigenvectors.
A graphical representation of (3.89) is given in Figure 3.9. The meaning of the eigendecomposition is that $\boldsymbol{G}$ and $\boldsymbol{D}$ represent the same uncertainty in two different reference frames. $\boldsymbol{P}$ and $\boldsymbol{P}^{-1}$ represent the change of basis from one reference frame to the other and vice-versa. (3.89) is called an *endomorphism* of $\boldsymbol{G}$ (a linear map to itself).
But (3.89) can also be applied to $\boldsymbol{H}$, since it is also a positive definite matrix, so:

$$\boldsymbol{H} = \boldsymbol{P}\boldsymbol{F}\boldsymbol{P}^{-1} \tag{3.93}$$

Furthermore, an important property is that diagonalization allows to easily compute any power of $\boldsymbol{H}$ as:

$$\boldsymbol{H}^k = \boldsymbol{P}\boldsymbol{F}^k\boldsymbol{P}^{-1} \tag{3.94}$$

which means that $\boldsymbol{G}$ can be also computed as:

$$\boldsymbol{G} = \boldsymbol{H}^{-1} = \boldsymbol{P}\boldsymbol{F}^{-1}\boldsymbol{P}^{-1} \tag{3.95}$$

where $\boldsymbol{F}^{-1} = \boldsymbol{D}$ obtained in (3.89). And (3.91) can also be applied to write the trace of $\boldsymbol{H}$ as the sum of its eigenvalues.
Expressing the matrices $\boldsymbol{H}$ and $\boldsymbol{G}$ as similar to $\boldsymbol{F}$ and $\boldsymbol{F}^{-1}$ allows to more directly compute the inverse. Since $\boldsymbol{F}$ is diagonal, it can be inverted by simply taking the inverse of each diagonal element, combining this with (3.90) and (3.91) we can finally write:

$$\text{GDOP} = \sqrt{Tr(\boldsymbol{G})} = \sqrt{Tr(\boldsymbol{F})} = \sqrt{\sum_{i}^{N} \frac{1}{\lambda_i^{\boldsymbol{H}}}} \tag{3.96}$$

so GDOP can also be written as a function of the eigenvalues of $\boldsymbol{H}$.
From the characteristic polynomial $det(\boldsymbol{H} - \lambda \boldsymbol{I}) = 0$ we know that the determinant can be written as the product of the eigenvalues:

$$det(\boldsymbol{H}) = det(\boldsymbol{F}) = \prod_{i}^{N} \lambda_i^{\boldsymbol{H}}. \tag{3.97}$$

Finally, from the definition given in (3.88), the trace of $\boldsymbol{H}$ is the sum of the norms of the columns of $\boldsymbol{A}$. Exploiting the shape of the $\boldsymbol{A}$ matrix used in GNSS in (3.8) and the property of the steering vector in (3.10), it follows directly that the sum of the column norms is always the same. Depending on the satellite geometry, some coefficients will be larger than other, but the total sum of a coefficients remains invariant. In particular, the following relationship holds:

$$Tr(\boldsymbol{H}) = \sum_{i}^{N} \lambda_i^{\boldsymbol{H}} = 2M \tag{3.98}$$

where M is the number of rows of the $\boldsymbol{A}$ matrix (i.e., the number of visible satellites).
The purpose of the above steps was to obtain (3.96), (3.97) and (3.98) all as functions of the eigenvalues of $\boldsymbol{H}$.
Now we have all the ingredients needed to derive a lower bound on GDOP. From (3.98) we have a set of real positive eigenvalues with a fixed sum, and we are interested in minimizing (3.96), which is the sum of their reciprocals.
It is possible to show, using the inequality of arithmetic and geometric means, that given a set of real and positive values with a fixed sum, minimizing the sum of reciprocals has the same solution as maximizing their product. Both problems are solved when the eigenvalues $\lambda_i^{\boldsymbol{H}}$ all have the same value.
Hence, the requirement to minimize the GDOP is that the eigenvalues of $\boldsymbol{H}$ are repeated. Since $\boldsymbol{H}$ is symmetric positive definite, it is guaranteed to have *complete*, real and positive eigenvalues. Complete means that the eigenvectors associated to each eigenvalue are all orthogonal. But we are interested in these eigenvalues to also be repeated, which occurs only when $\boldsymbol{H}$ is scalar matrix.
Therefore (3.93) becomes:

$$\boldsymbol{H} = \boldsymbol{I}\boldsymbol{F}\boldsymbol{I}^{-1} \tag{3.99}$$

where the change of basis is an identity matrix and as a consequence $\boldsymbol{H} = \boldsymbol{F}$. This results becomes intuitive by taking as an example a covariance matrix: its eigenvalues are proportional to the length of the semi-axis of the ellipse drawn from the matrix. If the eigenvalues are equal, then the ellipse collapses into a circle, which is the

representation of a scalar matrix.
In conclusion, the GDOP is minimum when $\boldsymbol{H}$ is diagonal with equal entries. It should be noticed that when those conditions are satisfies, the singular values of $\boldsymbol{H}$ are all equal and therefore the condition number (3.47) is also minimum.
Let up briefly recap the steps taken so far:

1. Write both GDOP and $det(\boldsymbol{H})$ as functions of the eigenvalues of $\boldsymbol{H}$
2. Show that minimizing the GDOP and maximizing $det(\boldsymbol{H})$ have the same solution.
3. Show that such solution is obtained when $\boldsymbol{H}$ has repeated eigenvalues.
4. Show that this occurs when $\boldsymbol{H}$ is diagonal with equal entries.

The only task left is to determine how much is this value on the diagonal, as a function of the number of rows of $\boldsymbol{A}$.
Start from (3.8) and (3.10), then the sum of all a coefficients can be written as:

$$\sum_{i}^{N}\sum_{j}^{M}(a_{ji})^2 = M \tag{3.100}$$

and the coefficients can be grouped by columns to obtain:

$$\sum_{j}^{N}\|\boldsymbol{v}_i\| = M \tag{3.101}$$

which from (3.88) we know are the diagonal elements of $\boldsymbol{H}$. Since there are $N = 3$ position coordinates and their columns always sum to M, then $\boldsymbol{H}$ should be in the form:

$$\boldsymbol{H} = \begin{bmatrix} \frac{M}{N} & 0 & 0 & 0 \\ 0 & \frac{M}{N} & 0 & 0 \\ 0 & 0 & \frac{M}{N} & 0 \\ 0 & 0 & 0 & M \end{bmatrix} \tag{3.102}$$

where the last element is always M since the last column of $\boldsymbol{A}$ is always made of all ones regardless of the position of satellites. The value of this element cannot be controlled, so the requirement of having the same value on the diagonal can only be applied to the first three elements. Since (3.102) is diagonal, it can be inverted

directly to obtain:

$$\boldsymbol{G} = \begin{bmatrix} \frac{N}{M} & 0 & 0 & 0 \\ 0 & \frac{N}{M} & 0 & 0 \\ 0 & 0 & \frac{N}{M} & 0 \\ 0 & 0 & 0 & \frac{1}{M} \end{bmatrix} \tag{3.103}$$

from which we can take the trace:

$$\text{GDOP} = \sqrt{\sum^{N} \frac{N}{M} + \frac{1}{M}} = \sqrt{\frac{N^2+1}{M}} \tag{3.104}$$

and since we are considering the positioning problem with $N = 3$ spatial dimensions, then the lower bound is:

$$\text{GDOP} \geq \sqrt{\frac{10}{M}} \tag{3.105}$$

which can also be found in [65], along with a different proof. An earlier analysis on the bounds of GDOP can be found in [66].
Let's analyse the conditions needed to obtain such an $\boldsymbol{H}$ matrix:

1. $\|\boldsymbol{v}_i\| = \|\boldsymbol{v}_j\| \quad \forall i, j$. All columns of $\boldsymbol{A}$ have the same norm. This tells us that given a total amount of information about the states we want to estimate, the minimum overall uncertainty will be reached if the information is equally spread on each state.

2. $\boldsymbol{v}_i \cdot \boldsymbol{v}_j = 0 \quad \text{If } i \neq j \quad i, j = [1, \cdots, N]$. The first three columns are orthogonal to each other. This means it is "easier" to estimate variables when the vectors of their partial derivatives are orthogonal to each other. From the point of view of satellite geometry, it means that satellites should be as spread out as possible.

3. $\sum_i^N a_{ji} = 0$. The first three columns should sum to zero. This ensures they are orthogonal to the fourth column of all ones. This requirements is connected to the previous one, but it allows to make one further consideration. The first requirement only refers to the total amount of information on each axis. Having for example two satellites east of the receiver at horizon level, or one east and one west, would have the total amount of information on the east-west axis. This last requirement tells us that the latter case is preferable, since it allows for better estimate of the clock bias.

The second and third requirement can be grouped so the overall meaning is the following: GDOP is minimum when the first three columns of A have the same

norm and all columns are orthogonal to each other. Note that these requirements are independent on the reference system used, any rotation of $\boldsymbol{A}$ will not cause any change.
Considering the example provided in Section 3.6, the GDOP is minimized when each basis of the column space $C(\boldsymbol{A})$ is orthogonal and with the same norm. Therefore, the accuracy performance is best when vectors of partial derivatives are orthogonal to each other and contain the same amount of information. In that way, each variable can be "recognized" more easily from the others and it is "easier" to tune the parameters so that they match the observations.

3.8.3 Discussion

It now becomes natural to ask, which satellites configuration achieves the best GDOP? The problem is only trivial in 2 dimensions, where the solution is given by regular polygons. The solution is much harder in 3 dimensions. It can be shown that is an approximate inverse relationship between GDOP and the volume of a polyhedron whose vertices are the position of satellites [13] (and even that is an unsolved problem in mathematics for large M).
A similar result regarding the shape of matrix $\boldsymbol{A}$ is derived in the domain of least squares spectral analysis. In that case, the objective is to represent a vector of data $\boldsymbol{\beta}$ as a weighted sum of sinusoidal basis functions. In matrix form, this is $\boldsymbol{\beta} = \boldsymbol{A}\boldsymbol{x}$. When the basis function $\boldsymbol{A}$ has the same properties derived from the minimum GDOP (orthogonal columns with same power), the representation is equivalent to a DFT. It corresponds to sinusoids equally spaced in frequency and sampled at equally spaced times.

One last comment regards the generalization of this result. So far in deriving a lower bound on GDOP, we did not constrain the position of satellites in any way. However, for real scenarios with a receiver on the surface of the earth, the elevation of visible satellites is strictly greater than zero. When adding this constraint, the coefficients in the third column (consider an ENU reference frame) are not free any more and can no longer sum to zero. As a consequence, the third and fourth column are no longer orthogonal, Vertical Dilution Of Precision (VDOP) and Time Dilution Of Precision (TDOP) are larger and the minimum GDOP can not be obtained. For this scenario, a more realistic bound is GDOP $= \sqrt{\frac{11.89}{M}}$ as found in [67].

The results from this analysis can prove useful in some scenarios such as satellite selection algorithms (for which the bound was originally derived) or the deployment of additional ranging measurements to be integrated with a GNSS solution.

3.8.4 Extra-State Theorem

Given the results obtained so far, there is one more consequence that can be discussed. Let's start by looking at an example. Take as a reference an optimal geometry with four satellites $M = 4$, namely 1 satellite at the zenith and 3 equally spaced at an elevation of $\theta = -19.47°$, thus forming a tetrahedron. Then the A matrix is:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0.9428 & -0.3333 & 1 \\ 0.8165 & -0.4714 & -0.3333 & 1 \\ -0.8165 & -0.4714 & -0.3333 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

and the requirements derived in Section 3.8.2 are satisfied: the first three columns have the same norm, are orthogonal to each other and sum to zero. As a matter of fact, the $\boldsymbol{H}$ matrix for this configuration is:

$$\boldsymbol{G} = \begin{bmatrix} 1.333 & 0 & 0 & 0 \\ 0 & 1.333 & 0 & 0 \\ 0 & 0 & 1.333 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} 0.750 & 0 & 0 & 0 \\ 0 & 0.750 & 0 & 0 \\ 0 & 0 & 0.750 & 0 \\ 0 & 0 & 0 & 0.250 \end{bmatrix}$$

As discussed, this geometry is not realistic for a receiver on the earth surface, so let's consider the scenario where all satellites are above the horizon and change $\theta = 19.47°$. Now, $\boldsymbol{A}$ becomes:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0.9428 & 0.3333 & 1 \\ 0.8165 & -0.4714 & 0.3333 & 1 \\ -0.8165 & -0.4714 & 0.3333 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

and it can be seen how the third column no longer sums to 0. As a matter of fact, now:

$$\boldsymbol{G} = \begin{bmatrix} 1.333 & 0 & 0 & 0 \\ 0 & 1.333 & 0 & 0 \\ 0 & 0 & 1.333 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} 0.750 & 0 & 0 & 0 \\ 0 & 0.750 & 0 & 0 \\ 0 & 0 & 3 & -1.500 \\ 0 & 0 & -1.500 & 1 \end{bmatrix}$$

its off-diagonal terms are no longer zero and the diagonal elements corresponding to VDOP and TDOP are larger.

The meaning is that in such case, when OLS is trying to minimize the residuals, it is harder to "tell" which parameter should be adjusted. Are the residuals large because the estimate of the vertical position is wrong? Or because the estimate of the clock bias is wrong? The smaller the angle between the two vectors of partial derivatives, the harder it is to know which parameter should be adjusted to reduce the residuals, and that is why the Dilution Of Precision (DOP)s are larger.

It is worthy to note that on the other hand, the first diagonal terms associated to Horizontal Dilution Of Precision (HDOP) have not changed. This observation has an interesting implication. Remember that a key requirement in obtaining the minimum GDOP is that columns of $\boldsymbol{A}$ should be orthogonal. The results of the example implies that more states can be added to the estimation problem, as long as the additional columns associated to these new states are orthogonal to the others. If this is satisfied, then the DOP of the other states will not increase [68].

This means that from the point of view of estimation, it is theoretically possible to add as many parameters as there are available measurements without loss of accuracy on the other parameters. Whether it is possible to find additional states in GNSS (or any other problem) that can satisfy this condition is a different matter.

Chapter 4

Bayesian Inference

In Chapter 3, the estimation of the user position for a single epoch was considered. However, users are typically concerned with a continuos estimation of their location across many epochs. Furthermore, receivers used in most applications are not static, or at most only for brief periods of time, so their position changes over time. In such cases, estimation by LS is still applicable but may not be optimal any more.
As a matter of fact, since the motion of the receiver is not random, but rather follows specific dynamics guided by physical laws, its behaviour can be modelled mathematically. In such scenarios, the knowledge about the dynamics of the receiver can be combined with GNSS measurements to augment the performance of LS.
Given the behaviour of states over time and their relationship with measurements, the estimation problem can be modelled by means of a Hidden Markov Model (HMM). In particular, a continuous estimate of states can be obtained by leveraging Bayes theorem, thus implementing what is usually referred to as Bayesian inference. This process involves forming predictions about the evolution of states in the future, and updating such predictions once new observations are available. Since predictions are formed depending on the way states evolve over time and on the current knowledge of states, they depend on past observations. The process of estimating the state of the receiver based on all observations up to and including the current time (but not the future) is known as filtering. This chapter introduces the filtering framework as well as the two Bayesian filters most commonly used in GNSS, namely the EKF and the PF. Finally, Section 4.4 is dedicated to introducing an original contribution regarding a novel PF architecture, published in [1], that allows to obtain any target accuracy for a lower computational load by exploiting the specific structure of the measurement models in GNSS.

4.1 Hidden Markov Model

When modelling the time-varying system, the behaviour of states over time and their relationship to the measurements may be summarized by the state space system:

$$\begin{aligned} \boldsymbol{x}_k &= \mathbf{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{q}_{k-1} \\ \boldsymbol{y}_k &= \mathbf{h}(\boldsymbol{x}_k) + \boldsymbol{\epsilon}_k \end{aligned} \tag{4.1}$$

where:

- $\boldsymbol{x}_k$ is a vector of states at epoch k (as considered in (3.1)).
- $\boldsymbol{y}_k$ is a vector of input measurements at epoch k.
- $\mathbf{f}(\cdot)$ is a set of state-transition functions modelling the behaviour of states over time.
- $\mathbf{h}(\cdot)$ is a set of measurement functions.
- $\boldsymbol{q}_{k-1} \sim \mathbf{P}(0, \boldsymbol{Q}_{k-1})$ is a vector of process noise at epoch $k-1$, drawn from a multivariate probability distribution with zero mean and covariance matrix $\boldsymbol{Q}_{k-1}$.
- $\boldsymbol{\epsilon}_k \sim \mathbf{P}(0, \boldsymbol{C}_k)$ is a vector of measurement noise at epoch k, drawn from a multivariate probability distribution with zero mean and covariance matrix $\boldsymbol{C}_k$..

In particular, the first equation in (4.1) models the evolution of states over time. On the other hand, the second relationship of (4.1), which links measurements to states, has already been analysed in Chapter 3. For now, no assumption on the distribution of noise is made.
In reality, the hidden states change continuously with time. However, the sets of input measurements are usually available only at certain time instants (typically regularly spaced by Δt). Therefore, (4.1) is a suitable discrete time representation of the system.
Functions $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ can be combined with the distribution of disturbances $\boldsymbol{q}_{k-1}$ and $\boldsymbol{\epsilon}_k$ to build a statistical framework for the estimation of the hidden states.
In particular, it is possible to exploit Bayes formula to obtain the joint distribution [51]:

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_{1:k}|\boldsymbol{x}_{0:k})\, p(\boldsymbol{x}_{0:k})}{p(\boldsymbol{y}_{1:k})} \tag{4.2}$$

where:

- $p(\boldsymbol{y}_{1:k}|\boldsymbol{x}_{0:k})$ is the likelihood distribution of measurements.
- $p(\boldsymbol{x}_{0:k})$ is the prior distribution.
- $p(\boldsymbol{y}_{1:k})$ is the marginal distribution, which acts as normalization factor and depends on the measurements.

However, computing the whole posterior distribution every time a new set of measurements become available may become inconvenient as the number of epochs grow large. In such case, the computational complexity needed to compute (4.2) increases, and it eventually becomes intractable. Fortunately, it is not needed to compute the full posterior at every time step. Since the system is assumed to be a Markov process, both current state and measurements are conditionally independent to the earlier states given the previous ones:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1}) &= p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \\ p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k}) &= p(\boldsymbol{y}_k|\boldsymbol{x}_k) \end{aligned} \tag{4.3}$$

or in other words, the past can be sufficiently summarised by the previous states. In such case, we can work with marginal distributions, which are much lighter to compute. The state space model (4.1) can be written in a statistical formulation as:

$$\begin{aligned} \boldsymbol{x}_0 &\sim p(\boldsymbol{x}_0) \\ \boldsymbol{x}_k &\sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \\ \boldsymbol{y}_k &\sim p(\boldsymbol{y}_k|\boldsymbol{x}_k). \end{aligned} \tag{4.4}$$

Since we are dealing with a filtering problem, the marginal posterior distribution of interest is the probability of the current state given the sequence of current and previous measurements, so (4.2) can be rewritten as:

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)\, p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \tag{4.5}$$

and the prior distribution is obtained as:

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})\, d\boldsymbol{x}_{k-1} \tag{4.6}$$

where $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$ is the marginal posterior at the previous epoch. So the prior is given by the combination of the posterior at the previous epoch and the state

transition function. The normalization factor is instead defined as:

$$p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{y}_k|\boldsymbol{x}_k)\, p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})\, d\boldsymbol{x}_k \tag{4.7}$$

which ensures that the posterior probability has an integral equal to 1.

This means that each marginal posterior can be computed sequentially from the

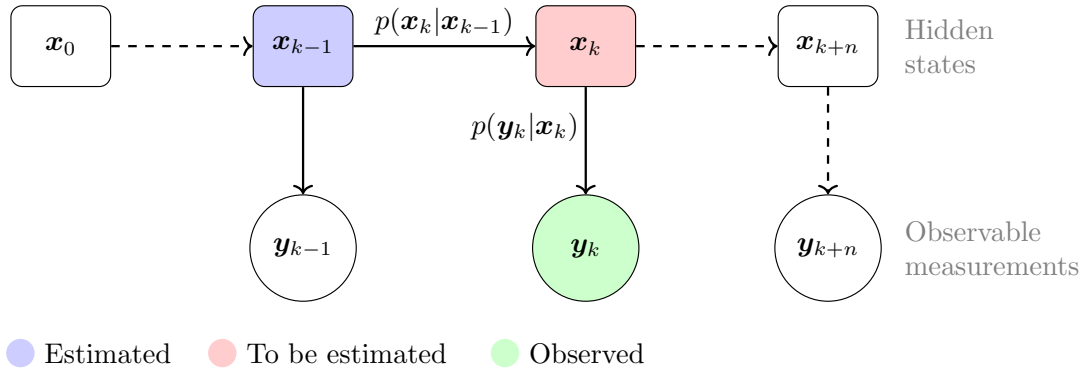


Fig. 4.1 Block scheme of a Hidden Markov Model.

previous one. The state space model of the system may be represented by means of a HMM, as illustrated in Figure 4.1. At each epoch, the current hidden parameters $\boldsymbol{x}_k$ can be estimated using the following ingredients: the current observed measurements $\boldsymbol{y}_k$, the previously estimated parameters $\boldsymbol{x}_{k-1}$ (which summarize the entire sequence of measurements up to that point) and the state transition and measurement functions. The formulation given in (4.5) tells us that the posterior is the product between the likelihood and the prior, normalized by the marginal likelihood.

On the other hand, the LS formulation analysed in Chapter 3 considered each set of states and measurements one at a time, as if the horizontal arrows in Figure 4.1 were not connected in the graph. In fact, the estimation of $\boldsymbol{x}_k$ was obtained simply by inverting its relationship to the measurements $\mathbf{h}(\cdot)$. As briefly mentioned in Chapter 3, the introduction of prior information into the ML framework of LS becomes a Maximum A Posteriori (MAP) estimation. Conversely, ML can be seen as a special case of MAP when the prior distribution is a uniform distribution. If there is no previous knowledge about the hidden variables, all its possible values can be considered to be equally likely.

It should be noticed from (4.4), that a filtering algorithm needs an initial distribution $\boldsymbol{x}_0$. In many cases, the LS method can be employed to obtain a solution at the first epoch to have an initial estimate of the hidden parameters and therefore initialize the filtering process.

Now that the Bayesian filtering framework for state estimation has been introduced, we can discuss different filtering algorithms implementing such framework.

4.2 Extended Kalman Filter

The Kalman Filter (KF), first proposed in 1960 by Kalman [69], is an optimal Bayesian filter (in the RMSE sense). Despite being a very popular tool to perform filtering, is has some limitations. In particular, its optimality ic constrained to the following assumptions:

- $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are linear functions.
- $\boldsymbol{q}$ and $\boldsymbol{\epsilon}$ are Normally distributed, independent and with known covariances $\mathbf{Q}$ and $\mathbf{C}$ respectively.

As we have seen, the measurement functions $\mathbf{h}(\cdot)$ are not linear in GNSS. Furthermore, the motion of the user may also be non linear. In such case, the functions may still be linearized to work with the Jacobians, similarly to how it is done to solve the least squares problem. The resulting filter is known as the EKF, which works on a prediction-update two step process. An introduction to the EKF is given in [70] and [71].

Prediction and Linearization

First, the predicted state $\boldsymbol{x}_k^*$ can be computed using the non linear state transition function:

$$\boldsymbol{x}_k^* = \mathbf{f}(\hat{\boldsymbol{x}}_{k-1}) \tag{4.8}$$

Then the system may be linearized at the predicted point, to obtain a matrix form:

$$\mathbf{F}_k = \left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_k^*} \tag{4.9}$$

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_k^*} \tag{4.10}$$

where $\mathbf{H}_k$ is essentially the equivalent of the $\boldsymbol{A}$ matrix considered for the least squares problem. Then, the estimate covariance can be obtained using the linearized

functions as:

$$\mathbf{P}_k^* = \mathbf{F}_k \, \hat{\mathbf{P}}_{k-1} \, \mathbf{F}_k^T + \mathbf{Q}_{k-1}. \tag{4.11}$$

These steps effectively implement the computation of the prior needed in Bayes formula (4.2).

Update

First off, the EKF builds the innovation vector:

$$\boldsymbol{z}_k = \mathbf{y}_k - \mathbf{h}(\boldsymbol{x}_k^*) \tag{4.12}$$

which contains the differences between the input measurements and the nominal measurements at the predicted position. The innovations are similar to residuals, with the difference that the former are computed using the predicted position and the latter using the final estimated position. The related innovation covariance can be computed as:

$$\mathbf{S}_k = \mathbf{H}_k \, \mathbf{P}_k^* \, \mathbf{H}_k^T + \mathbf{C}_k. \tag{4.13}$$

Then, the Kalman gain can be computed as:

$$\mathbf{K}_k = \mathbf{P}_k^* \, \mathbf{H}_k^T \, \mathbf{S}_k^{-1}. \tag{4.14}$$

These steps incorporate information coming from the measurements to build a likelihood. Finally, the state estimate can be update according to:

$$\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^* + \mathbf{K}_k \, \boldsymbol{z}_k \tag{4.15}$$

and the covariance estimate is also updated:

$$\hat{\mathbf{P}}_k = (\boldsymbol{I} - \mathbf{K}_k \, \mathbf{H}_k) \, \mathbf{P}_k^*. \tag{4.16}$$

So the final posterior is obtained as a combination of the prior and the likelihood.

4.2.1 GNSS model

For many terrestrial applications, the dynamics of the user can be modelled as a constant-speed motion. So, the linearized state-transition matrix can be written as:

$$\mathbf{F}_k = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times1} & \Delta t\, \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 & \mathbf{0}_{1\times3} & \Delta t \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times1} & \mathbf{0}_{1\times3} & 1 \end{bmatrix} \tag{4.17}$$

where $\mathbf{0}_{n\times n}$ is an n by n matrix of zeros. How well this model describes the dynamics of the receiver depends on the specific application. If the user dynamics change rapidly w.r.t. the rate Δt at which PVT is performed, the linear model might not be a sufficiently accurate characterization of the receiver dynamics and a more complete model should be used instead.
The process noise covariance matrix should also be tuned according to specific the dynamics of the user and the receiver. In general, it can be computed according to the model:

$$\mathbf{Q}_k = \begin{bmatrix} \boldsymbol{S_v}\dfrac{\Delta t^3}{3}\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times1} & \boldsymbol{S_v}\dfrac{\Delta t^2}{2}\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & S_t\Delta t + S_f\dfrac{\Delta t^3}{3} & \mathbf{0}_{1\times3} & S_f\dfrac{\Delta t^2}{2} \\ \boldsymbol{S_v}\dfrac{\Delta t^2}{2}\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times1} & \boldsymbol{S_v}\Delta t\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & S_f\dfrac{\Delta t^2}{2} & \mathbf{0}_{1\times3} & S_f\Delta t \end{bmatrix} \tag{4.18}$$

where $\boldsymbol{S_v}$ depends on the velocity of the receiver on the three axial directions, while S_t and S_f depends on the clock of the receiver. On the other hand, the linearized observation matrix is obtained as:

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}^{\rho}_{n\times4} & \mathbf{0}_{n\times4} \\ \mathbf{0}_{n\times4} & \mathbf{H}^{\dot{\rho}}_{n\times4} \end{bmatrix} \tag{4.19}$$

where $\mathbf{H}^{\rho}_{n\times4}$ and $\mathbf{H}^{\dot{\rho}}_{n\times4}$ are the Jacobian matrices of the pseudorange and pseudorange rate measurements respectively. If the measurements are obtained from the same set of satellites, than the two matrices are equal to each other. Furthermore, if GNSS is being integrated with additional sensors, matrix $\mathbf{H}_k$ can be augmented by adding other rows containing the partial derivatives of the measurement equation of said sensors w.r.t. to the states to be estimated.

4.2.2 Performance

The Fisher information for the KF is given by [72]:

$$\mathbb{I}_k = \mathbf{H}_k^T \, \mathbf{C}_k^{-1} \, \mathbf{H}_k + \left(\mathbf{F}_k^{-1}\right)^T \mathbb{I}_{k-1} \mathbf{F}_k \tag{4.20}$$

where the first term of (4.20) depends on the geometry and the covariance of measurements, exactly as computed in (3.50) for the GLS. On the other hand, the second term depends instead on the Fisher information at the previous epoch and the state transition matrix.
It should be noticed that if the problem is linear, then the EKF is the same as the KF. In that case, both EKF and KF are optimal estimator. If any of the functions involved in the system is not linear, then the filters are no longer optimal and their performance may degrade according to the extent of the non linearities. However, the EKF shall perform better than the KF as it exploits the non linear form of the equations when possible, such as in the prediction step (4.8) and the computation of the innovation (4.12).
Nevertheless, under some circumstances, the linearization performed by the EKF may not be enough to properly model the system. In such scenarios, the performance of the EKF may be poor and it may even diverge. In particular, when GNSS is augmented by other sensors whose measurement functions are highly non linear, such as UWB, other filters may be required to reach near-optimal estimation accuracy.

4.3 Particle Filter

While the KF poses further assumptions on the system w.r.t. to the HMM, it is able to sequentially compute the target posterior in a relatively light matrix form, at least by modern computing standards. After all, when the system is governed by non linear functions and the probability densities of errors are non Gaussian, computing the posterior analytically is very hard. On the other hand, the PF has a more brute force approach to the estimation problem. Instead of placing further constraints on the shape of the distributions and the linearity of the equations, it approximates distributions in a discrete fashion, using samples called particles. In particular, PF belongs to the family of Sequential Monte Carlo (SMC) methods.
The prediction-update steps of the KF and EKF are substituted by the concepts of sampling, weighting and (possibly) resampling. There are many algorithms in the family of SMC that can be used for the estimation of parameters in non linear

dynamic systems. The formulation presented here is known in the literature as Sequential Importance Resampling (SIR) PF, which is a popular solution for the tracking of targets. An overview of SMC methods in given in [73]. A complete overview of PF is instead presented in [74].
Other nice tutorials are given in [75–77], and in particular for positioning and tracking problems in [78–80]. Some of which include examples and interesting discussions about practical implementations of SMC methods, which are out of the scope of this chapter.
The first sampling algorithms for SMC were developed in the control theory community in the late 1960s [81]. More complete formulations were developed decades later as in [82], thanks to large advancements in available computing power. Over the course of the last decades, PF have proven to be a successful tool for estimation in many fields including robotics [83–85] and machine learning [86, 87]. As a matter of fact, SMC methods are often taught in Aritficial Intelligence (AI) books for their use in learning algorithms [81, 88, 89].

4.3.1 Initialization

The PF needs to be initialized by taking N samples an initial prior distribution:

$$\begin{aligned} &\boldsymbol{x}_0^j \sim p(\boldsymbol{x}_0) \\ &w_0^j = 1/N \qquad j = 1, ..., N \end{aligned} \tag{4.21}$$

where j is the index of the particle (sample). To each particle is associated a weight w^j, which are initially all equal to each other, since the position of samples is already probabilistic depending on the prior distribution.
The PDF representation using weighted samples can be interpreted as follows: each particle represents a possible solution, to which is associated a likelihood of being the correct solution (the weights). SMC algorithms such as PF allow to recursively track the posterior of dynamical systems as new sets of measurements become available.

4.3.2 Sequential Estimation

The following steps are repeated at each iteration (i.e. when new measurements become available). First, the particles are propagated using the state transition model:

$$\boldsymbol{x}_k^j \sim p(\boldsymbol{x}_k^j \mid \boldsymbol{x}_{k-1}^j) \tag{4.22}$$

which in terms of practical implementation means that each particle from the previous posterior is propagated forward using the state transition function $\mathbf{f}(\boldsymbol{x}_{k-1}^j)$ and then adding noise sampled from $\boldsymbol{Q}_{k-1}$. Now that an empirical prior distribution has been obtained, the likelihood can be can be incorporated by updating the weights according to:

$$w_k^j \propto w_{k-1}^j \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^j) p(\boldsymbol{x}_k^j \mid \boldsymbol{x}_{k-1}^j)}{q(\boldsymbol{x}_k^j \mid \boldsymbol{x}_{k-1}^j, \boldsymbol{y}_k)} \tag{4.23}$$

where w_{k-1}^j are the weights at the previous epoch and $q(\boldsymbol{x}_k^j \mid \boldsymbol{x}_{k-1}^j, \boldsymbol{y}_k)$ is called the *importance density* (or sometimes proposal density). The importance density is the PDF from which the samples were drawn. Ideally, one would draw them directly from the posterior, but we don't know it yet. It can be shown that the correct way is to assign weights such that they are proportional to the ratio between the posterior and the importance density.

In this case, the samples were drawn according to the state transition model (4.22), so (4.23) simplifies to:

$$w_k^j \propto w_{k-1}^j \, p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^j) \tag{4.24}$$

Although it is natural to chose the state transition model as proposal density, any other density can be used and the weights can sill be computed using (4.23) instead. The proportionality is considered because then the weights need to be normalized to sum to 1, thus implementing the denominator of (4.5).

Unlike the KF, the distribution $p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^j)$ is not restricted to be a Normal distribution. Instead, it can be any distribution, possibly even multimodal. To most natural implementation is to use the covariance of measurement error $\boldsymbol{C}$ to build $p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^j)$. Furthermore, if the measurements are assumed to be independent, the joint likelihood can be computed as the product of the marginals.:

$$p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^j) = \prod_{i=1}^{M} p(y_{i,k} \mid \boldsymbol{x}_k^j) \tag{4.25}$$

where i is the index of measurements.

In practical implementations, weights can be computed for each particle as follows: first, the nominal measurement can be computed using the measurement equations $\boldsymbol{y}_k^j = \mathbf{h}(\boldsymbol{x}_k^j)$, then the innovation $\boldsymbol{z}_k^j$ is calculated as in (4.12). Finally, the likelihood is obtained by multiplying the chosen probability densities of the error evaluated at the innovations $p(y_{i,k} \mid \boldsymbol{x}_k^j) = p(z_{i,k}) = p(y_{i,k} - \mathbf{h}_i(\boldsymbol{x}_k^j))$ for each measurement.

Now the posterior distribution is represented by a set of weighted samples. There can be many strategies to obtain the state estimates. A common approach is to

compute them as the weighted sum:

$$\hat{\boldsymbol{x}}_k = \sum_{j=1}^{N} w_k^j \boldsymbol{x}_k^j \tag{4.26}$$

but for example the sample with the largest weight could also be considered. A common issue of PF is that, after a few epochs, particle weights tend to collapse. That is, a few samples have large weights while most other samples have weights equal to zero. This phenomenon is known as *degeneracy*. In order to detect whether degeneracy is occurring, the effective number of particles is computed:

$$N_{ef} = \left(\sum_{j=1}^{N} \left(w_k^j \right)^2 \right)^{-1} \tag{4.27}$$

and monitored by comparing to a threshold N_{th}. If the value falls below the threshold, the set of particles is resampled. Common values for the threshold are $N_{th} = \frac{2}{5}N$ or $N_{th} = \frac{1}{2}N$.

The idea of resampling, introduced in [90], is to obtain a new set of N particles, by drawing N times from the old set of particles according to the weights. A review of resampling algorithms regarding their implementation and computational load is provided in [91–93].

As a result, particle with weights close to 0 will not be drawn in the new set, while particles with large weights will be drawn possibly multiple times. Ideally, the posterior should be sampled at different N different spots. Resampling got rid of samples that did not provide any information on the posterior, but now has many copies of the same samples. This problem is known as *sample impoverishment*. Fortunately, this problem is compensated by the fact that multiple copies of the same samples will be added different realizations of process noise during the next prediction steps, thus ensuring that all samples will be again at different values of the posterior. Tuning process noise and resampling is a tough balancing act. If not set properly, the PF may quickly diverge from the solution.

A graphical representation of the main steps of PF is given in Figure 4.2.

4.3.3 Performance and Limitations

The PF has the advantage w.r.t. KF that is does not impose any restriction on the model of the system. As a matter of fact, it can handle any function for the state

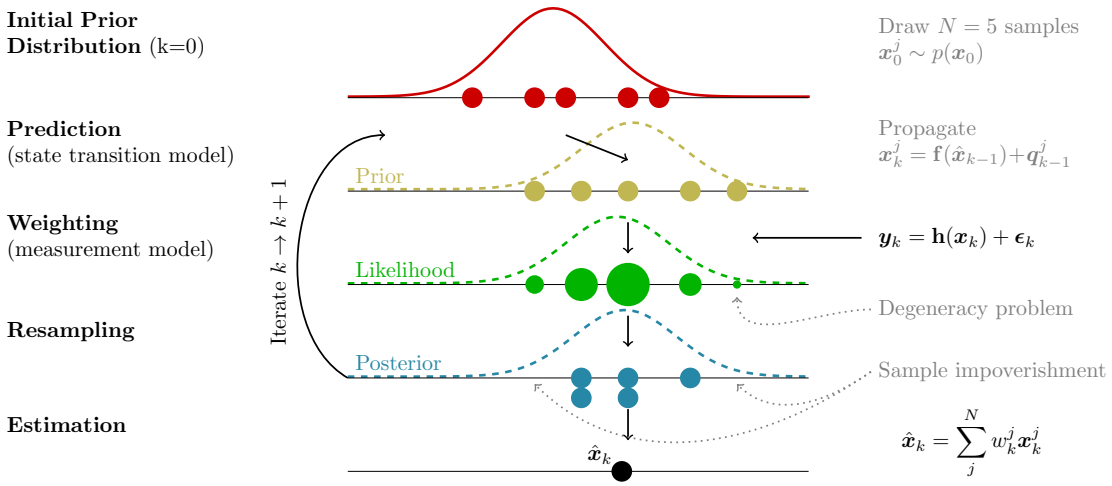


Fig. 4.2 Scheme of Particle Filter routine [1].

transition and the measurement model, as well as any distribution of the error on the measurement.

Furthermore, due to the SMC approach used by PF, the performance in terms of estimation accuracy depends on the number of particles used in relation to the number of hidden parameters. As a consequence, there is no compact formula to estimate the covariance of the estimation as done for the EKF. On one side, the linearization error and other limitations of the EKF when the assumptions do not hold are not present in the PF. Indeed, the PF can optimally estimate the posterior regardless of its shape, given that an infinite amount of particles is used. This is of course not feasible, so there is an estimation error introduced due to numerical inaccuracy of the sampling. The more particles are used, the lower is the estimation error. While this feature may be considered both an advantage or a disadvantage, one take away is that the PF has one degree of freedom in its design compared to KF-based methods. The design of the filter can be adjusted according to specific requirements on the accuracy or depending on the computational resources available. As a result, there are some criticalities with the implementation of PF. The main issue is what is usually referred to as the *curse of dimensionality*. The number of particles needed to reach a target accuracy on the posterior distribution grows exponentially with the dimensionality of the problem (i.e. the number of parameters to be estimated). In some cases, if the number of samples is not sufficient, the filter may diverge. For this reason, computationally efficient implementations and variants of the PF have been thoroughly investigated as in [94].

A popular strategy to reduce the computational complexity is to exploit linear sub-structures of the system, thus implementing what is known as Rao-Blackwellized PF

(or sometimes marginalized PF), as developed in [95–97]. Other strategies include distributing the estimation over multiple filters when dealing with high-dimensional systems, as investigated by [98–101], although the architecture of the filter may become more complex to implement. Furthermore, other authors have proposed local PF [102, 103] to maintain an approximation error that does not increase with the dimension, as well as Unscented [104] or Auxiliary [105] variants. Some solutions even exploit an adaptive number of particles [106] to reduce the computational load. Despite many variants of the PF have been proposed to deal with its limitations, there is a lack of optimal formulations that are GNSS-specific. This reason motivates the design of the MW-PF.

4.4 Multiple Weighting Particle Filter

Many modifications of the PF have been proposed in the literature to improve its performance in terms of accuracy. For example, Mixture PF samples from a mixture of distributions, as proposed first in robotics [107], and than also applied to GNSS sensor fusion [108].

When applying legacy PF to the estimation of the hidden parameters, the entire set of input measurements is used to compute a single weight for each particle (4.24). However in some applications, not all input measurements are related to all the states through the measurement model. In particular, it can be that different kinds of measurements are related only to non-overlapping subsets of the state space. The PF mixes all the available information into a single weight, which gives an overall likelihood of the particle across the whole state space. Conversely, a more clever use of resources is possible by leveraging the knowledge of the state-observables relationship. In such cases, similar measurements (e.g., from the same sensor) can be grouped into I subsets of the observables $\boldsymbol{y}_{(i)}$), from which multiple weights $w_{(i)}$ are derived to estimate the corresponding sub-spaces of the state vector $\boldsymbol{x}_{(i)}$. Subset indexes are noted using round brackets.

In order to characterize the information diversity from dissimilar measurements, multiple observables-state functions can be defined:

$$\boldsymbol{y}_{(i)} = \mathbf{h}_{(i)}(\boldsymbol{x}_{(i)})\mathbf{v}_{(i)} \quad \text{where } \boldsymbol{y}_{(i)} \subset \boldsymbol{y}. \tag{4.28}$$

By leveraging the aforementioned simplification thanks to the resampling strategy considered, (4.24) can be rewritten for the (i)-th independent weight is computed as

$$w^j_{(i),k} \propto w^j_{(i),k-1}\, p(\mathbf{y}_{(i),k-1}|\boldsymbol{x}^j_{(i),k-1}). \tag{4.29}$$

Equivalently, the state estimate from (4.26) is also modified as:

$$\hat{\boldsymbol{x}}_{(i),k} = \sum_{j=1}^{N} w^j_{(i),k} \boldsymbol{x}^j_{(i),k} \tag{4.30}$$

and each partition of the state space is therefore estimated separately each using the corresponding weight. It should be noticed that, on the other hand, the prediction step is performed jointly, as the two partitions are performed depending on the measurements model, while the state transition functions have not changed. While the sampling of the two subsets is performed independently, position and velocity are still tied in the dynamic model and used jointly in the prediction step, as one is the derivative of the other.Hence, the two quantities cannot be fully decoupled.
An alternative solution could be to distribute the estimation over multiple filters instead, with each one devoted to the estimation of a subset of states, as done in the multiple PF. This solution would still require the filters to share information as different subsets of states can still be related in the system model (e.g. prediction of position at next epoch depends on the velocity), and hence comes at the cost of a more complex architecture w.r.t. the solution presented here. Eventually, the estimated subsets obtained according to (4.30) are merged together, as well as the subsets of each particle with the matching index j. Since we are interested in approximating a discrete probability density, the indexing of the particles does not influence the output estimate. The outcome of (4.30) only depends on values and weights of particles. In this architecture, the resampling stage can be performed fully independently on the different subsets using the corresponding weights to draw the resampled subsets.
A related problem was discussed in [109], when integrating asynchronous measurements from dissimilar sensors, and was solved by proper modifications of the resampling stage. In our case, measurements are dissimilar but collected synchronously. When measurement information is merged into a single weight, the likelihood of each subset of states is lost and only an overall likelihood of each particle is retained in the standard PF approach. For this reason, the problem has to be addressed before the resampling stage and the mentioned approach cannot be applied to this scenario. Local PF could be exploited in order to maintain a dimension-free approximation

error. This approach is possible in state space models where block of observations are conditionally independent given the hidden state and only depend on separate components of the hidden state. Furthermore, the proposed solution differs from classic Rao-Blackwellized PF as all subsets of the state space are estimated through PF, thus preserving the fundamental properties of SMC methods. A variant of this work where each state is sampled separately has later been proposed in [110] with remarkable results.

4.4.1 Theoretical Proof

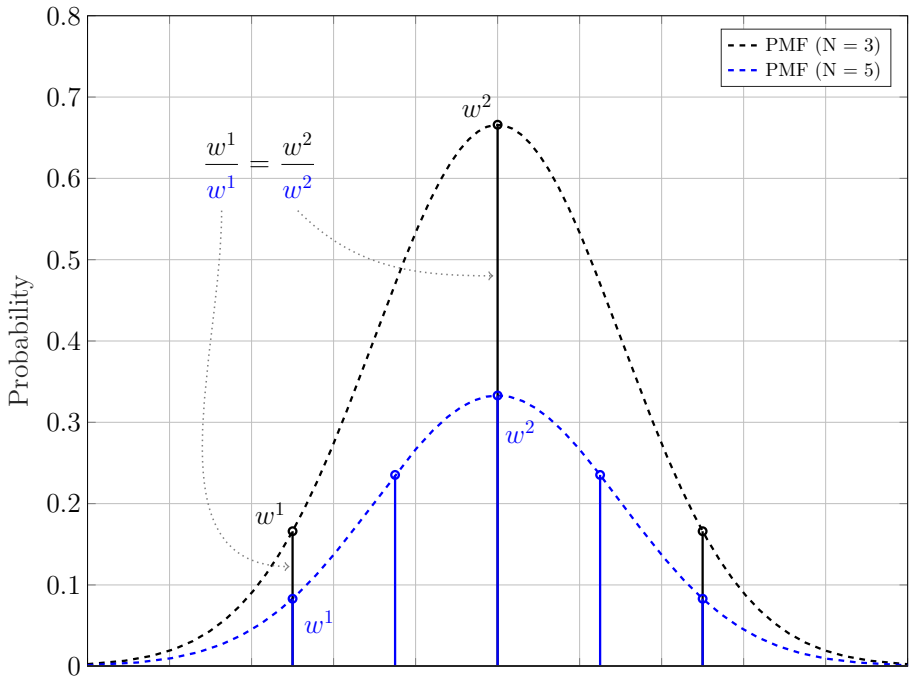


Fig. 4.3 One dimensional example of sampling a PDF to generate a Probability Mass Function (PMF). Each PMF using a different number of particles N is shown with a different colour [1].

In the state estimation problem, the original posterior distribution of the states is a continuous form PDF. SMC methods leverage a set of particles to form a discrete empirical distribution which approximates the original continuous distribution. The normalization step is then taken to ensure the summation of the PMF is always equal to 1 as in (4.5). Therefore, if a different number of particles is used to represent the same continuous distribution, the PMFs will be different, as can be seen in Figure 4.3.
Start by considering the following:

- The state vector contains two independent states $\boldsymbol{x} = [\, x_1 \;\; x_2 \,]$.
- A set of N particles, and focus more specifically on two particles with given values $\boldsymbol{x}^1 = [\, a \;\; c \,]$ and $\boldsymbol{x}^2 = [\, b \;\; d \,]$.
- Define the subsets as $\boldsymbol{x}_{(1)} = [\, x_1 \,]$ and $\boldsymbol{x}_{(2)} = [\, x_2 \,]$

For now, consider the estimation of the first state x_1. Given a fixed number of particles, the scope is to determine which discrete marginal distribution is more accurate between those obtained with PF or MW-PF.
First, let's see what happens when employing MW-PF. The corresponding independent weights are computed using (4.29). For the two particles considered, we want:

$$\frac{w^1_{(1)}}{w^2_{(1)}} = \frac{f_{\boldsymbol{x}_{(1)}}(a)}{f_{\boldsymbol{x}_{(1)}}(b)} \tag{4.31}$$

where $f_{\boldsymbol{x}_{(1)}}(a)$ is the marginal distribution evaluated in a. If this condition is met, then the continuous posterior distribution can be represented correctly using a discrete distribution [111]. This is possible in any PMF because $w^1_{(1)}$ and $w^2_{(1)}$ are always normalized by the same denominator, so their ratio is constant.
In subset $\boldsymbol{x}_{(1)}$, weights are determined by:

$$w^i_{(1)} \sim f_{\boldsymbol{x}_{(1)}} \quad \forall i \in (1, N) \tag{4.32}$$

regardless of how many particles are used in the filter. It follows that a single particle with value $\boldsymbol{x}_{(1)} = a$ is sufficient to sample directly the exact value of the marginal posterior distribution $f_{\boldsymbol{x}_{(1)}}(a)$. Therefore, (4.31) holds and we guarantee that the marginal is represented correctly.
Instead, in the PF case, weights are drawn from a joint distribution $f_{\boldsymbol{x}} = f_{\boldsymbol{x}_{(1)},\boldsymbol{x}_{(2)}}$ of the entire set of states. Due to their independence, the joint distribution can be represented as the product of every marginal distribution [111], and we obtain:

$$w^j = w^j_{(1)}\, w^j_{(2)} \sim f_{\boldsymbol{x}_{(1)}}\, f_{\boldsymbol{x}_{(2)}} = f_{\boldsymbol{x}_{(1)},\boldsymbol{x}_{(2)}}. \tag{4.33}$$

For this reason, if we want to derive an estimation of the marginal distribution of $\boldsymbol{x}_{(1)}$ from the joint distribution, the influence from $\boldsymbol{x}_{(2)}$ needs to be eliminated. In order to do that, define a set of particles $j_a = 1, ..., M_a$ that respect the condition of $\boldsymbol{x}_{(1)} = a$. We want to obtain the marginal weight, denoted by the hat, by sampling at that value $\hat{w}^{\boldsymbol{x}_{(1)}=a}$ for the first subset $\boldsymbol{x}_{(1)}$ from the total space set $\boldsymbol{x}$. This means that we need to average the weight of all particles which belong to the set j_a. This

can be written as:

$$\hat{w}^{\boldsymbol{x}_{(1)}=a} = \frac{1}{M_a} \sum_{j_a=1}^{M_a} w^{j_a}. \tag{4.34}$$

Because of the left side of (4.33), it can be rewritten as:

$$\hat{w}^{\boldsymbol{x}_{(1)}=a} = \frac{1}{M_a} \sum_{j_a=1}^{M_a} w_{(1)}^{j_a} \, w_{(2)}^{j_a}. \tag{4.35}$$

Since all particles follow $\boldsymbol{x}_{(1)}^{i} = a$, then the first weight is the same for particles in the set, and it can be taken out of the summation:

$$\hat{w}^{\boldsymbol{x}_{(1)}=a} = w_{(1)}^{\boldsymbol{x}_{(1)}=a} \frac{1}{M_a} \sum_{j_a=1}^{M_a} w_{(2)}^{j_a}. \tag{4.36}$$

We know from (4.31) that $w_{(1)}^{\boldsymbol{x}_{(1)}=a}$ is sampled directly from $f_{\boldsymbol{x}_{(1)}}(a)$. Hence, the approximation $\hat{w}^{\boldsymbol{x}_{(1)}=a}$ in (4.36) equals to the true value $w_{(1)}^{\boldsymbol{x}_{(1)}=a}$ times a scaling factor. The latter is influenced by the particular values of $w_{(2)}^{j_a}$, which in turn depends on the values of $\boldsymbol{x}_{(2)}$ of the particles in the set j_a.
Notice that this scaling value does not necessarily have to be equal to 1, as all weights are then normalized by a common factor. Instead, we want the average of $w_{(2)}^{j_a}$, to be equal to the average of the entire set $w_{(2)}^{j}$. Otherwise, approximations at different values of the marginal distribution of $\boldsymbol{x}_{(1)}$ are multiplied by different scaling factors, leading to a distortion in the representation of the posterior. Clearly, this effect is mitigated as N increases since the averaging is performed on larger sets of samples and converges to the correct value.
Using the same inference for the other particle $\boldsymbol{x}^2$, define the set $j_b = 1, ..., M_b$ which satisfies $\boldsymbol{x}_{(1)} = b$. Then, the marginal weight can be obtained as:

$$\hat{w}^{\boldsymbol{x}_{(1)}=b} = w_{(1)}^{\boldsymbol{x}_{(1)}=b} \frac{1}{M_b} \sum_{j_b=1}^{M_b} w_{(2)}^{j_b}. \tag{4.37}$$

Therefore,

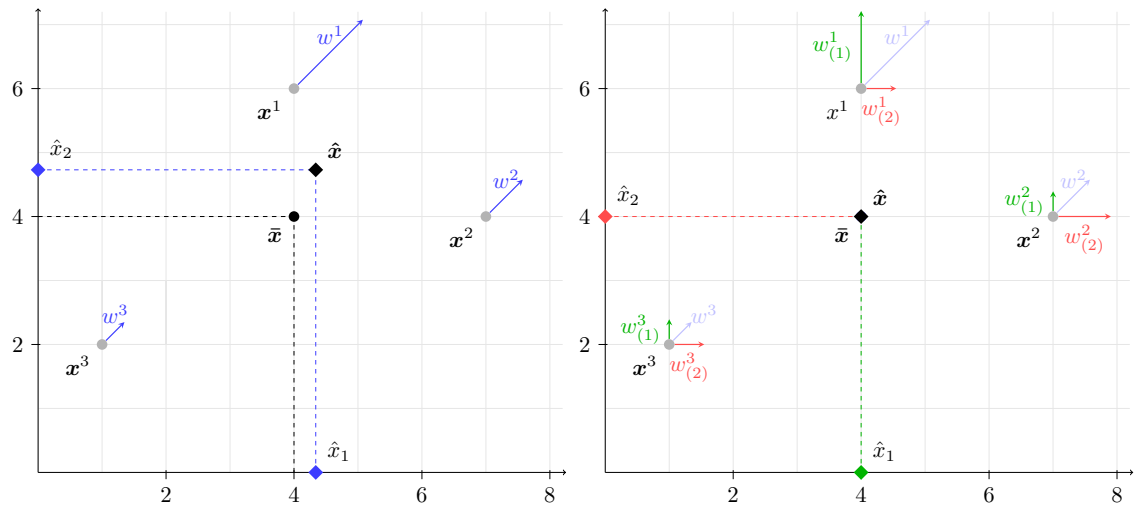
$$\frac{\hat{w}^{\boldsymbol{x}_{(1)}=a}}{\hat{w}^{\boldsymbol{x}_{(1)}=b}} = \frac{w_{(1)}^{i=a}}{w_{(1)}^{i=b}} \frac{\frac{1}{M_a} \sum_{j_a=1}^{M_a} w_{(2)}^{j_a}}{\frac{1}{M_b} \sum_{j_b=1}^{M_b} w_{(2)}^{j_b}}. \tag{4.38}$$

Because the states x_1 and x_2 are independent, for any given j, the particle weights $w_{(2)}^{j}$ follow the same distribution with mean μ and variance σ^2. Applying the central limit theorem, the distribution containing $\frac{1}{M_b} \sum_{j_b}^{M_b} w_{(2)}^{j_b}$ approaches a normal distribution with mean μ and variance $\frac{\sigma^2}{M_b}$. Therefore, only with a large number of total

particles N, which also implies large M_a and M_b, both $\frac{1}{M_a}\sum_{j_a}^{M_a} w_{(2)}^{j_a}$ and $\frac{1}{M_b}\sum_{j_b}^{M_b} w_{(2)}^{j_b}$ will converge to μ. Then, it follows that $\frac{\hat{w}^{\boldsymbol{x}_{(1)}=a}}{\hat{w}^{\boldsymbol{x}_{(1)}=b}}$ will converge to $\frac{f_{\boldsymbol{x}_1}(a)}{f_{\boldsymbol{x}_1}(b)}$ and the PDF can be represented without distortion.

In summary, given that two independent states, if we want to represent the marginal distribution of x_1 using a set of particles without distortion, the conventional PF needs more particles than the proposed MW-PF because it needs to eliminate the impact from x_2.

4.4.2 Numerical Example



(a) Legacy weighting (b) Multiple weighting

Fig. 4.4 Comparison of a state-estimation using (a) conventional and (b) the proposed MW-PF approaches for a simple two-dimensional scenario with $N = 3$ particles [1].

	States		PF	MW-PF	
	x_1	x_2	w	$w_{(1)}$	$w_{(2)}$
$\boldsymbol{x}^1$	4	6	0.538	0.606	0.274
$\boldsymbol{x}^2$	7	4	0.288	0.197	0.452
$\boldsymbol{x}^3$	1	2	0.175	0.197	0.274

Table 4.1 Summary of values and weights of particles.

A simple numerical example is provided in order to illustrate the results obtained in the proof of the MW-PF strategy. In particular, we will show the appearance of an error in the estimation given by the inaccurate approximation of the marginal

posterior distribution when the number of particles is low and the legacy weighting strategy is employed. In order to do so, a small numerical example is set up.
We consider a single epoch simulation in which the

- The state space vector is $\boldsymbol{x} = [\, x_1 \;\; x_2 \,]$ with true values $\bar{\boldsymbol{x}} = [\, 4 \;\; 4 \,]$.
- The input to the estimation problem is a set of two measurements $\boldsymbol{y} = [\, y_1 \;\; y_2 \,]$.
- The measurement function $\mathbf{h}(\cdot)$ that describes the relationship between measurements and states in this case is $\mathrm{y}_i = \bar{x}_i + \mathrm{v}_i \quad \forall i \in \{1, 2\}$ such that each measurement is a direct noisy observation of the corresponding state.
- The noise terms v_i are modelled as zero mean Gaussian distributions with standard deviation $\sigma = 2$. This corresponds to $p \sim \mathcal{N}(0, 2)$.
- However, it is assumed that the realization on noise available at the targeted epoch are equal to zero, so that $\mathrm{v}_i = 0 \quad \forall i \in \{1, 2\}$. This is done in order to show that, even when the realization of the noise is zero, there is still a numerical error in the estimate due to inaccurate sampling of the posterior.
- We assume to have $N = 3$ particles with given values $\boldsymbol{x}^1 = [\, 4 \;\; 6 \,]$, $\boldsymbol{x}^2 = [\, 7 \;\; 4 \,]$, $\boldsymbol{x}^3 = [\, 1 \;\; 2 \,]$.
- Weights are all initialized to $\frac{1}{N} = \frac{1}{3}$.

PF First, the state estimate is computed through a legacy PF. Start by computing, for each measurement and each particle, the difference between the input value and the nominal one, which yields the innovations $\boldsymbol{z}^1 = [\, 0 \;\; -2 \,]$, $\boldsymbol{z}^2 = [\, -3 \;\; 0 \,]$ and $\boldsymbol{z}^3 = [\, 3 \;\; 2 \,]$. These vectors are fed into $p(\boldsymbol{z}^j)$ to obtain the weights, which are then normalized. A summary of the weights is reported in the third column of Table 4.1. The final estimate is obtained using a weighted sum:

$$\begin{aligned}
\hat{\boldsymbol{x}} &= \sum_{j=1}^{3} w^j \boldsymbol{x}^j \\
&= (0.538 \cdot [\, 4 \;\; 6 \,]) + (0.288 \cdot [\, 7 \;\; 4 \,]) + (0.175 \cdot [\, 1 \;\; 2 \,]) \\
&= [\, 4.340 \;\; 4.726 \,].
\end{aligned}$$

As it can be seen, despite the realization of the error being zero, the final estimation has an error w.r.t. the true values. The result ai also graphically shown in Figure 4.4 (a). Since a low number of particles is being used for the estimation, the estimation of

each state is influenced by the specific values at which the posterior is being samples, as proved in Section 4.4.1.

MW-PF When performing the estimation according to the proposed MW-PF, the subsets $\boldsymbol{x}_{(1)} = [\, x_1 \,]$ and $\boldsymbol{x}_{(2)} = [\, x_2 \,]$ are considered. In this case, only the first measurement contributes to $\boldsymbol{z}^j_{(1)}$, which is used to obtain the first weight $w^j_{(1)}$. The same procedure can be followed to obtain the weights $w^j_{(2)}$ from $\boldsymbol{z}^j_{(2)}$. After normalisation, the weights take the values reported in the last two columns of Table 4.1. The final estimate is:

$$\hat{\boldsymbol{x}}_{(1)} = \sum_{j=1}^{3} w^j_{(1)} \boldsymbol{x}^j_{(1)} = (0.606 \cdot 4) + (0.197 \cdot 7) + (0.197 \cdot 1) = 4$$
$$\hat{\boldsymbol{x}}_{(2)} = \sum_{j=1}^{3} w^j_{(2)} \boldsymbol{x}^j_{(2)} = (0.274 \cdot 6) + (0.452 \cdot 4) + (0.274 \cdot 2) = 4$$

and so there is no error on the estimate, as depicted in Figure 4.4 (b).

In conclusion, because of the independence between , the marginal posterior density of the first state is described by a Gaussian distribution $f_{\boldsymbol{x}_{(1)}} \sim \mathcal{N}(\mu, \sigma)$ which is symmetric around the mean $\mu = 4$. Therefore:

$$\frac{f_{\boldsymbol{x}_{(1)}}(\mu + \epsilon)}{f_{\boldsymbol{x}_{(1)}}(\mu - \epsilon)} = 1 \tag{4.39}$$

is always true for any given value ϵ. Due to our assumptions, notice that particle states $x^2_1 = 7$ and $x^3_1 = 1$ are indeed symmetric around $\mu = 4$. So we compute the ratio of their weights for MW-PF and PF cases:

$$\frac{w^2_{(1)}}{w^3_{(1)}} = \frac{0.197}{0.197} = 1$$
$$\frac{w^2}{w^3} = \frac{0.288}{0.175} = 1.648$$

and notice that it is not equal to 1 for the latter. Since in this example N is not large, the influence of the second state x_2 is not averaged out, and the PF is not able to accurately represent the marginal posterior density, leading to an error on the estimation.

4.4.3 Application to GNSS

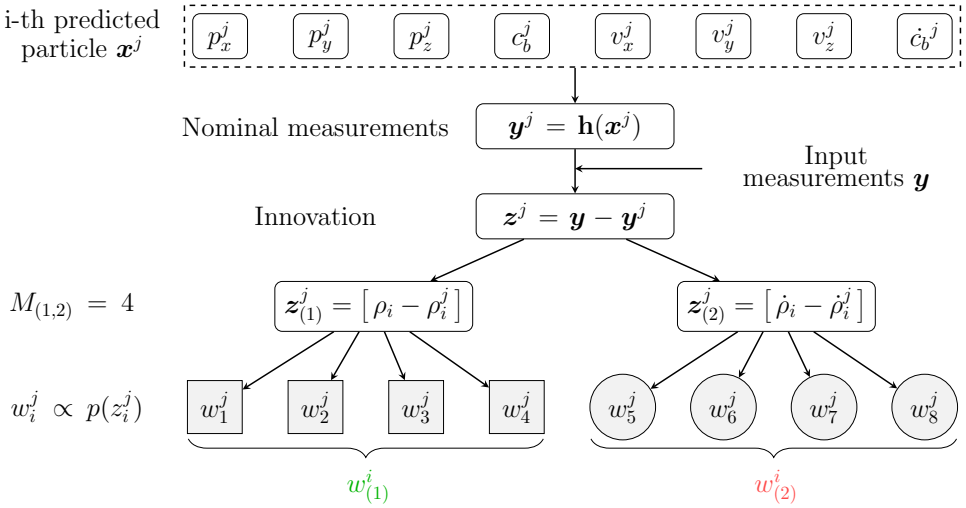


Fig. 4.5 Example of computation of weights for MW-PF with $M = 4$ visible satellites.

This section is dedicated to the discussion and implementation of the proposed MW-PF to state estimation in GNSS receivers. In this scenario, there are two types of measurements that GNSS receivers can obtain. Namely, *pseudoranges* and *range rates* (are related to Doppler shifts). As we have seen, the complete measurement matrix in (4.19) is composed as follows:

- The top left sub-matrix contains the partial derivatives of pseudoranges w.r.t the position.
- The bottom right sub-matrix contains the partial derivatives of range rates w.r.t the velocity, which we have seen to be the same.
- The bottom left should contain the partial derivatives of pseudoranges w.r.t velocity. It is easy to see from 2.6 that is has no dependence on the velocity, so these should be zero.
- The top right should contain the partial derivatives of range rates w.r.t the position. Differently from the previous term, 2.10 does have a dependency on the position (on its derivative to be precise). Therefore, these partial derivatives should contain the second derivatives of pseudoranges w.r.t. the position. In practice, these derivatives are very small (in the order of 10^{-6} for MEO satellites), so they can be neglected.

So far when introducing the MW-PF, we have talked about the independence between subsets of states which only depend on disjoint subsets of measurements. Let's formalize this assumption in light of the above analysis. The MW-PF assumes that the problem can be formulated such that the measurement matrix is block diagonal, or that the off-diagonal terms are negligible. In that case, the partition of states and measurement is performed accordingly.
The assumption introduced has the following meaning: the difference in position between the two particles has a negligible contribution to the computation of the nominal range rate. In other words, we assume that if particles all had the same velocity, they would measure the same range rate. Since the distance between satellites and particle is much greater than the distance between any two particles, all the steering pointing from the particles to the satellite can be considered parallel to each other. Equation (2.10) computes the normalized projection of the relative velocity on the steering vector. Since the latter contribution is approximated to be the same for all particles, then the range rate measurement depends only on the velocity and clock drift of the particle.
Under this assumption, the partition of states for GNSS becomes:

$$\boldsymbol{x} = [\underbrace{p_x \;\; p_y \;\; p_z \;\; c_b}_{\boldsymbol{x}_{(1)}} \;\; \underbrace{v_x \;\; v_y \;\; v_z \;\; \dot{c}_b}_{\boldsymbol{x}_{(2)}}] \tag{4.40}$$

and for measurements:

$$\begin{aligned} \boldsymbol{y}_{(1)} &= [\rho_1 \;\; \cdots \;\; \rho_m] \\ \boldsymbol{y}_{(2)} &= [\dot{\rho}_1 \;\; \cdots \;\; \dot{\rho}_m] \end{aligned} \tag{4.41}$$

so pseudoranges are used to compute the first weights, then used the estimate the position and clock bias. Conversely, range rates are used to compute the second weights and hence estimate velocity and clock drift parameters. An examples of how the weights can be computed for this architecture is shown in the block diagram in Figure 4.5.
In this study, we employ zero-mean Gaussian distribution as probability densities of measurement errors. Even tough this choice could be sub-optimal in some scenarios, the scope of this section is to investigate this impact of a possible mismodelling. Both filter architectures can be expected to be equally penalized by this choice so that any comparison remains fair.

4.4.4 Experiment and Results

The experiment data was collected using the Navigation Constellation Simulator (NCS) simulator, a GNSS signal simulation and generation system. The ephemeris and observation data, including pseudoranges and Doppler shifts was stored in Receiver Independent Exchange Format (RINEX) format. All the observations are of the GPS constellation with the L1 C/A signal. To simulate noise, we add noise via ionosphere noise model with the standard deviation of 2 m and 1 m for pseudoranges in the static and dynamic scenarios, respectively, and 1 Hz for Doppler shifts in both scenarios. Input measurements are collected at a rate of 10 Hz. To validate our proposed algorithm, both static and dynamic scenarios were built.

Static Scenario

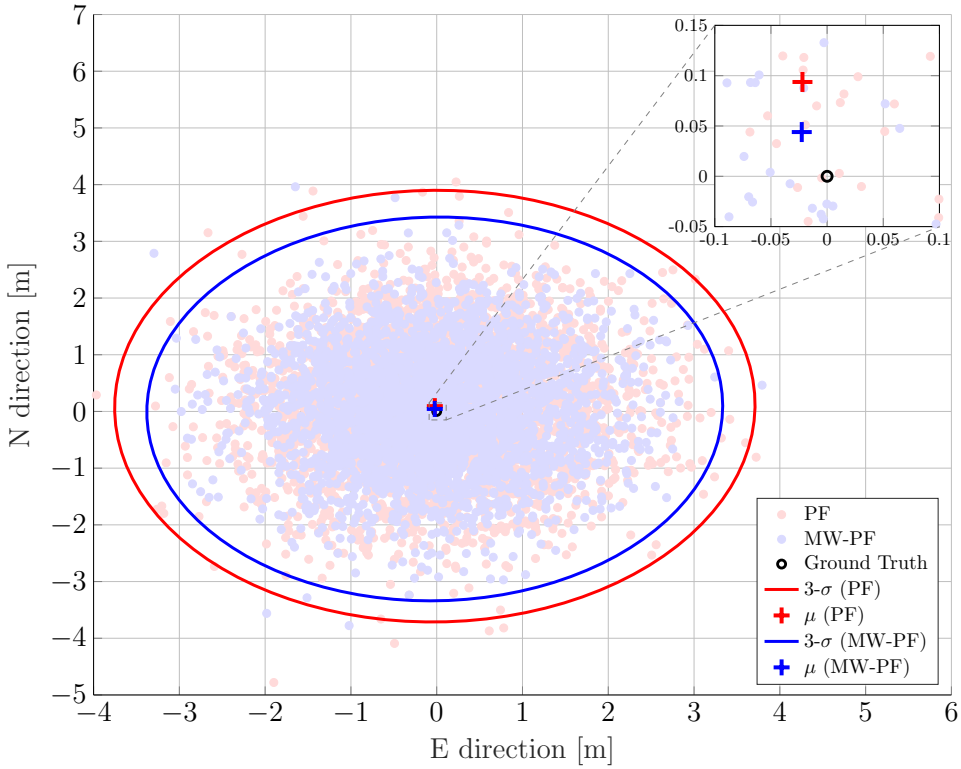


Fig. 4.6 Comparison of PF and MW-PF solutions with $N = 2000$ particles (in east-north reference frame) for a static GNSS receiver. Mean value of the estimate and 3-σ uncertainty in the form of error ellipses are also depicted. The ground truth is located in (0,0).

Although Bayesian estimation is primarily exploited for kinematic state estimation, accurate static state estimation is still of interest as it can temporarily occur in

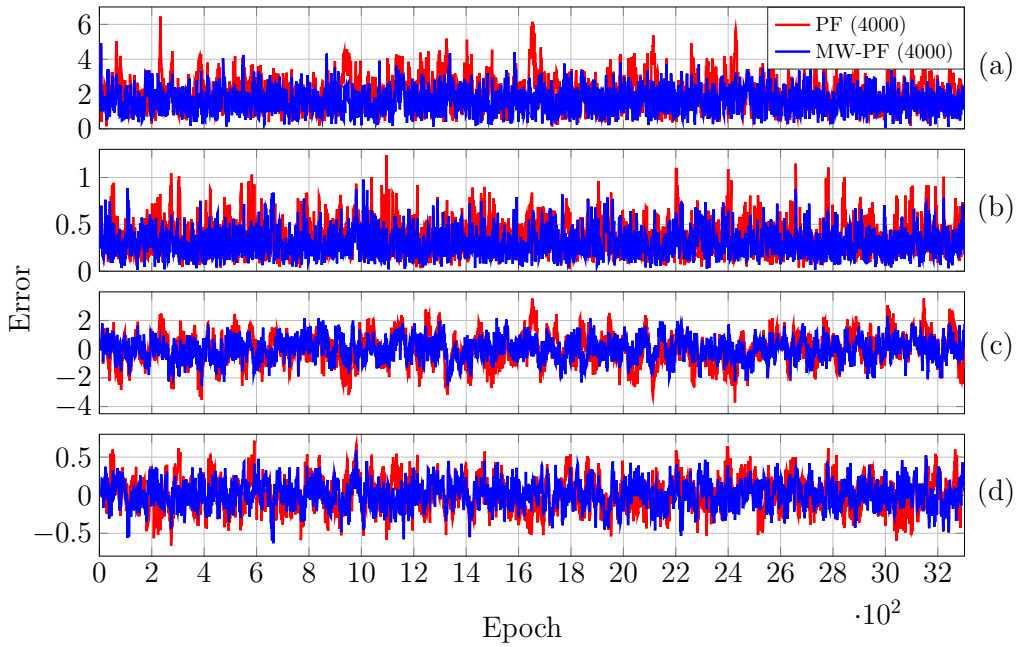


Fig. 4.7 (a) Root Squared Error (RSE) on position [m] (b) RSE on velocity [m/s] (c) Clock bias error [m] (d) Clock drift error [m/s].

State	PF	MW-PF	Improvement
3D Position	2.084	1.613	22.6%
3D Velocity	0.371	0.304	18.1%
Clock Bias	1.089	0.766	29.7%
Clock Drift	0.217	0.173	20.3%

Table 4.2 Comparison between MW-PF and PF (both at $N = 4000$) in terms of RMSE for the static scenario.

any real trajectory. Moreover, it can be an interesting baseline assessment for the performance of any positioning algorithm. Therefore, an experiment involving a static position estimation is performed first.

Figure 4.6 plots all the positioning solutions obtained with the PF and MW-PF for all epochs of the simulations. The plot represents the East-North plane of a local ENU reference system, with the ground truth in its center. To better visually display the difference in performance between the two implementations of the PF, we chose for this plot the solutions when a low number of particles is used ($N = 2000$), and the improvement given by our proposed method is more stark.

The errors on all the state variables over time is instead displayed in Figure 4.7 for $N = 4000$. As it can be seen, the MW-PF is more accurate in the estimation of all the state variables.

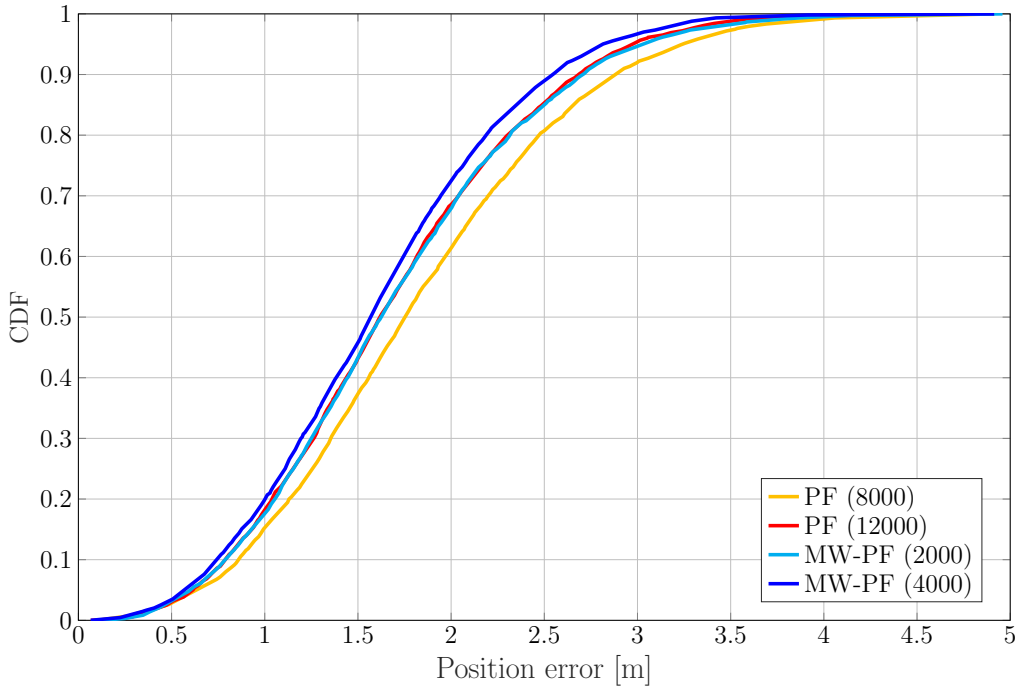


Fig. 4.8 Comparison between PF and MW-PF in terms of Cumulative Distribution Function (CDF)s of positioning error of a static receiver.

State	PF	MW-PF	Improvement
3D Position	0.724	0.418	42.3%
3D Velocity	0.224	0.219	2.2%
Clock Bias	0.391	0.240	38.6%
Clock Drift	0.120	0.116	3.3%

Table 4.3 Comparison between MW-PF and PF (both at $N = 4000$) in terms of RMSE for the dynamic scenario.

Eventually, Figure 4.8 shows the CDF of the positioning error for both algorithms, tested for some selected number of particles. In reality, more values were tested but were in the end omitted for the sake of clarity of the plot. In particular for the MW-PF, going beyond $N = 4000$, the performance did not improve any further. For the PF instead, as it can be inferred from the plot, for values lower than $N = 8000$ the performance degraded very quickly. Instead, values above $N = 12000$ were not tested as the simulations became increasingly time consuming. More details on the computational complexities will be given later, but for now it is interesting to notice how the performance of PF for $N = 12000$ is very close to that of MW-PF for $N = 2000$. The important take-away from this observation is that MW-PF can reach the same target accuracy with a significant reduction of the computational load. On

the other hand, for a fixed available (and reasonable, meaning N is not too large) computational effort, the MW-PF can outperform the PF in terms of accuracy of the positioning solution.

Dynamic Scenario

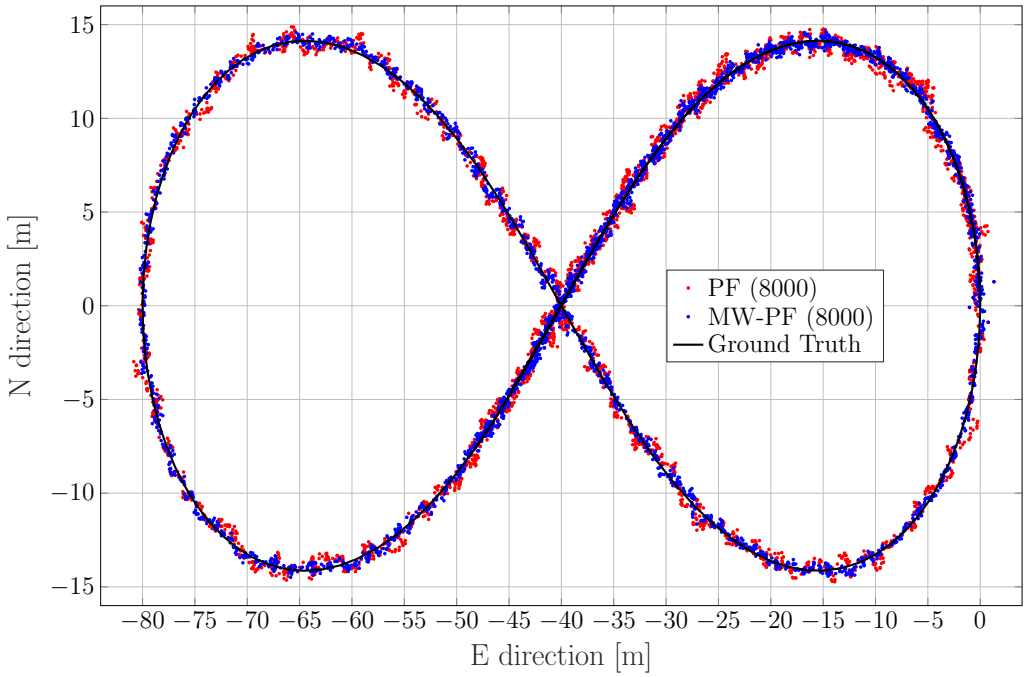


Fig. 4.9 Comparison of PF and MW-PF solutions (in east-north reference frame) applied to position estimation of a dynamic GNSS receiver.

For a second assessment, an artificial dynamic trace is used with the shape of a Bernoulli lemniscate, as can be seen in Figure 4.9, which also displays the positioning solutions for both algorithms ($N = 8000$). The moving target performs roughly one loop of the track during the simulations. By comparing the positioning solutions of Figure 4.9 it can be seen how, especially in some parts of the trajectory, the MW-PF solution is consistently closer to the ground truth.
As done for the static case, the error on the state variables of interest is shown in Figure 4.10. Once again, a deliberate choice of plotting the errors of the two algorithms for a lower number of particles was made in order to emphasize the difference in their performance. In particular, it is interesting to notice from subplots (a) and (c) how in this scenario the improvement in accuracy given by MW-PF is larger for the estimation of position and clock bias. This difference was not as stark when comparing the same errors of the static scenario. This phenomenon

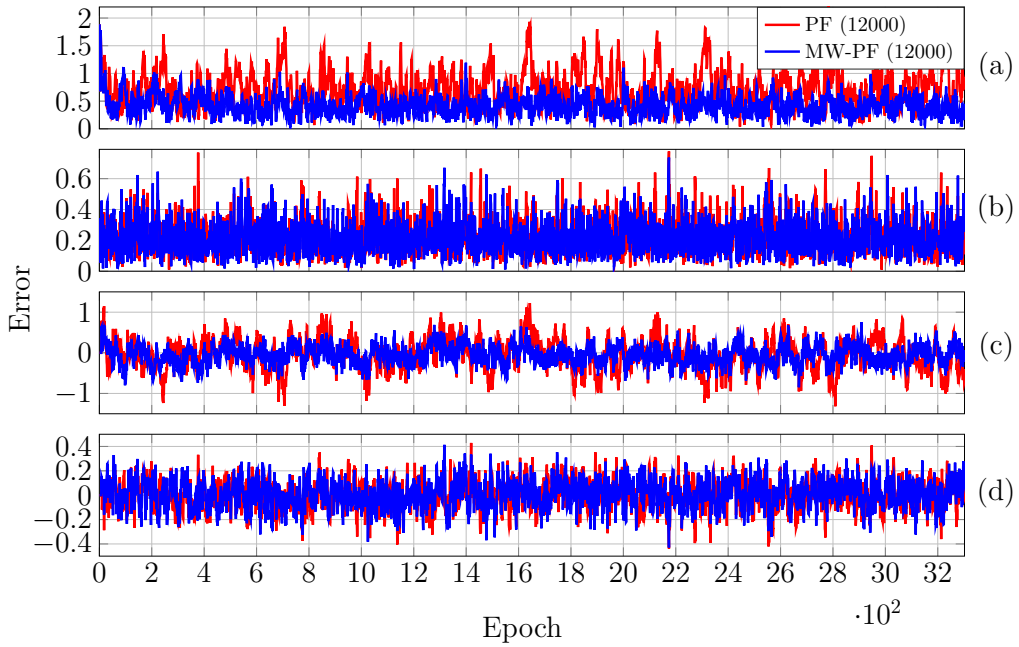


Fig. 4.10 (a) RSE on Horizontal position [m] (b) RSE on Horizontal velocity [m/s] (c) Clock bias error [m] (d) Clock drift error [m/s].

can be quantified by looking at Tables 4.2 and 4.3 which provide a summary of the two tests. The improvement column refers to the percentage decrease in RMSE when employing MW-PF instead of PF. We remind that position and clock bias are the variables chosen to form the first sub-vector, since pseudorange measurements provide information about those state. This results suggests that, when the target is in a dynamic state, splitting the estimation of position and clock bias with their respective derivatives, the gain in estimation accuracy is larger for the former.
The CDF of the positioning solution of both algorithms is shown in Figure 4.11. We selected the results for some specific number of particles in order to not overcrowd the plot. The take-away from this results is similar to what observed for the static scenario, which is that MW-PF can reach the same accuracy of PF with a reduced number of particles.
Finally, 4.12 shows the error at the 90th percentile of the CDF for both algorithms and different values of particles. We wanted to investigate whether by further increasing N for PF, its performance would eventually reach or even surpass that of MW-PF. The last value we tested was $N = 60000$ since simulations eventually became too long to continue. This last test yielded a 90th percentile error of 0.650 against one of 0.607 for MW-PF at $N = 20000$. The conclusion is that even when N is extremely large, the performance of PF doe not fully converge to that of MW-PF, suggesting that some small residual additional errors remain due to the sub-optimal sampling

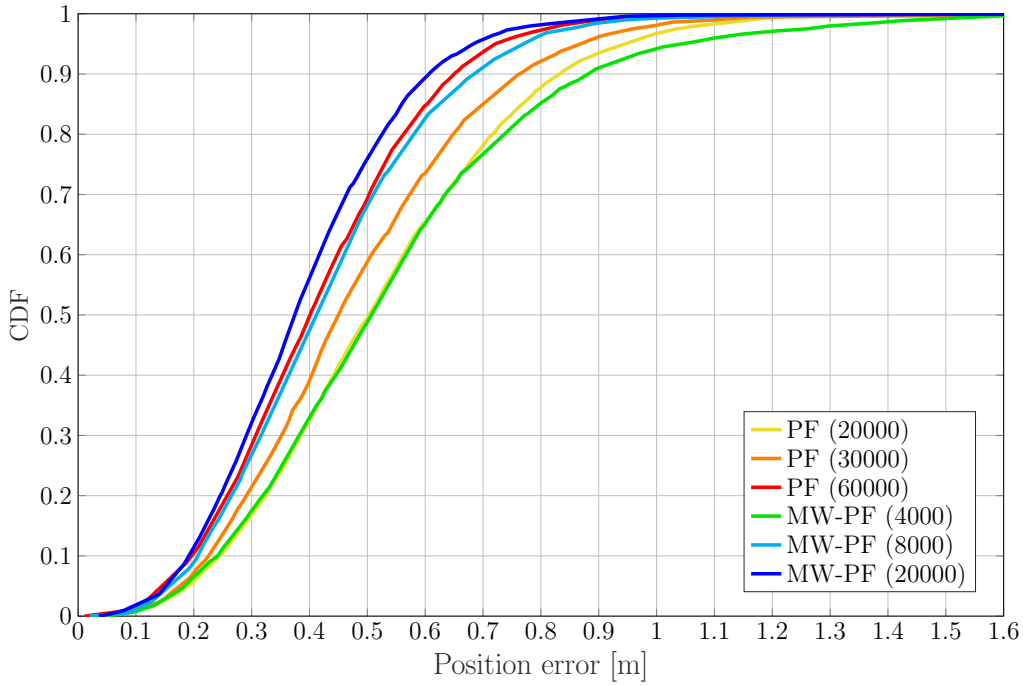


Fig. 4.11 Comparison of the CDF of the positioning error of a dynamic receiver using PF and MW-PF solutions with different numbers of particles.

of the algorithm. Given the results from Figure 4.12 for MW-PF, we identify values of N between 4000 and 12000 as possible good working points in terms of trade-off between computational load and accuracy.

Computational Complexity

N	**PF**	**MW-PF**
1000	6.26	6.40
2000	7.98	8.05
4000	11.15	11.30
8000	15.00	15.58

Table 4.4 Comparison between PF and MW-PF of the simulation run times in seconds to solve 300 seconds of PVT.

Since the two algorithms presented in these results present some differences in their code and implementation, a summary of their execution times is given in order to give a fair comparison between the two. The results are reported in Table 4.4 for some values of N. By fixing any N, the run time of MW-PF is slightly longer

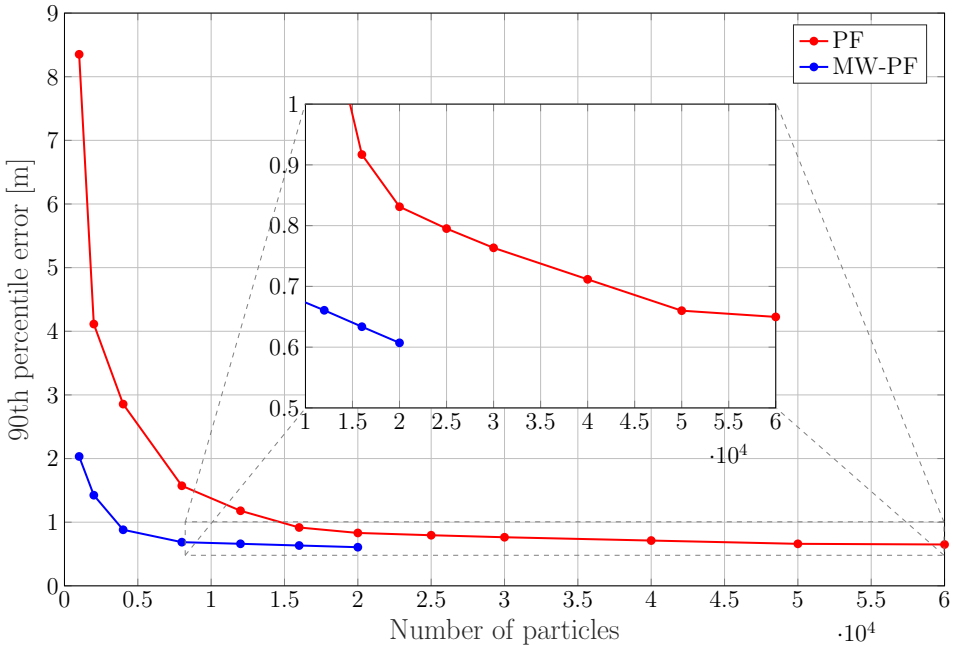


Fig. 4.12 Comparison of 90th percentile error of PF and MW-PF solutions with different numbers of particles.

than PF as expected, since some computations and checks are performed twice. Overall, this increase is not large and is mostly offset by the fact that the MW-PF implementation can reach the same accuracy with fewer particles.
It is important to stress that the times reported here are given simply in order to provide a comparison between the two algorithms, rather than to give a thorough investigation of the computational complexities of PF. In fact, no parallel optimization has been implemented (although we anticipate to do so in the future), despite some heavy computations of PF could be implemented this way, leading to a reduction in simulation times.

4.4.5 Discussion

The MW-PF has been introduced, with the scope to exploit information diversity of input measurements to achieve a more accurate sampling of the posterior distribution with fewer particles. Despite being applied to GNSS here, the technique can be generalized to be exploited in other types of state estimation problems with minimum modifications of the PF routine. While in the investigated application the state vector was split in two subsets, any number of such subsets is possible in principle, according to the relationship between measurements and states in the system of

interest.

Indeed, when sampling using multiple weights, each particle retains information about the likelihood of each subset of states, rather than an overall likelihood across all states. In fact, the proposed MW-PF is able to mitigate the main drawback of SMC methods w.r.t. to KF. On one hand, the multiple weighting can reduce the inaccuracies due to the empirical discrete approximation of the posterior. On the other hand, it may introduce additional errors if the assumptions are not fully respected. Namely, if the measurement matrix is not block diagonal. Results show that indeed, for stand-alone GNSS with MEO constellation, the improved weighting strategy far outweighs the neglected terms in the measurement matrix. Nonetheless, the applicability of the MW-PF in other scenarios remains to be investigated.

Simulation campaign show that, for both static and dynamic cases, MW-PF provides better performance in terms of accuracy, especially when a low number of particles is used. In other terms, the same accuracy obtained through PF can be reached with MW-PF with as low as one fifth of the particles. Conversely, for the same $N = 12000$ in the dynamic scenario, MW-PF can provide an improvement of over 40% in terms of positioning error.

Chapter 5

Multipath Detection and Mitigation

The scope of this chapter is to develop novel methods for the detection and mitigation of multipath effects, by exploiting ideas inspired by some of the analysis made in Chapter 3 regarding residuals and Chapter 4 regarding Bayesian filtering. In particular, our goal is to combine the theoretical insights with novel ideas such as soft information [112] to develop robust estimation in the presence of faulty measurements.

The main reason to tackle multipath and NLoS phenomena is that despite the global coverage of GNSS, these faults are one of the main limitations in obtaining accurate positioning performance in challenging environments. As a matter of fact, reflection phenomena remains the primary limitation to the use of emerging applications such as autonomous vehicles and smart wearables. Therefore, it is of large interest to guarantee GNSS integrity in urban environments [113].

As a consequence, large research effort has been devoted to the detection of measurement faults and to mitigate their effect on the positioning solution. However, the problem remains extremely challenging and is still one of the main limitations of GNSS stand-alone positioning, thus motivating further research on the topic.

The remainder of this chapter is structured as follows: first a background on multipath and NLoS is given, along with a summary of the main techniques for the detection and mitigation of such phenomena. Then, two main sections will be presented, describing two original contributions for the mitigation of bias due to multipath and NLoS phenomena. The first method, published in [5], is developed to automatically label datasets in post-processing by leveraging a clustering algorithm

which helps identifying outliers. On the other hand, the second method can be applied in real-time to mitigate faulty measurements by exploiting the concept of soft information to construct multimodal probability distributions in the measurement model of the PF. This latter work was first presented in [6] and then extended into a recently submitted journal paper.
Nonetheless, both presented methods attempt to estimate and compensate for the bias in order to still exploit positioning information from the faulty measurements, as opposed to performing Fault Detection and Exclusion (FDE). The reason for this choice is that in harsh environment, with limited satellite visibility, position-related measurements are a scarce resource and excluding faulty measurements might have a severe impact on the GDOP.

5.1 Background

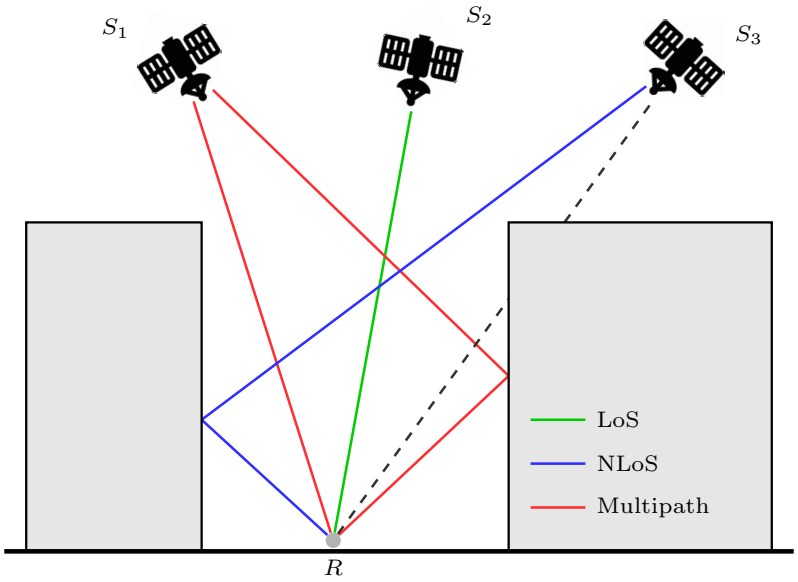


Fig. 5.1 Example of direct and reflected signal reception in urban canyon.

Given the different definition of multipath and NLoS that can be found in the literature, we wish to clarify the terminology that will be used here [114]:

- **Multipath**: Reception of a signal through a direct path and one or more reflected path, or multiple reflected paths only.
- **NLoS**: Reception of a signal through a reflected path only, since the direct path is obstructed.

In both cases, the reception of reflected signals may cause biases in the estimated travel time of the signal, and therefore on the pseudoranges. However, NLoS is purely a delay of the original signal which causes an increase in the estimated travel time of the signal, while multipath may cause distortions of the received signals which may have more complex affects on the different stages of a receiver.
The techniques that have been developed to detect and mitigate reflection phenomena can be classified in three main groups:

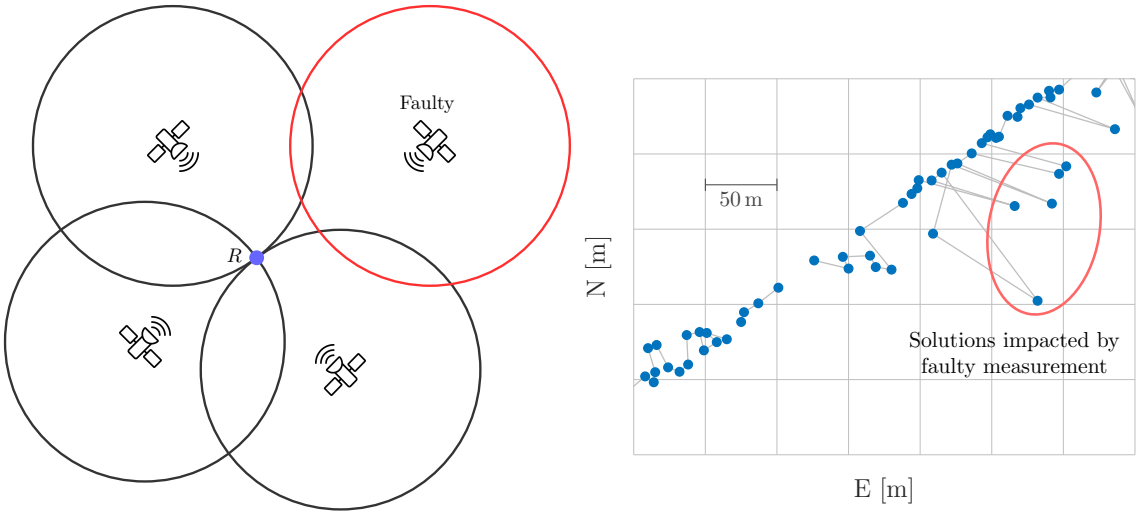
- **Antenna-based**: Multipath and NLoS faults are caused by the reflected signals, which have different polarization and reception angles w.r.t. direct signals. Exploiting this knowledge, some of the most popular solutions include utilizing choke rings [115], antenna polarization [116, 117], and array antennas [118].

- **Receiver-based**: In the signal processing stage, reflected signals may cause distortion of the auto-correlation in the DLL. Therefore, more sophisticated tracking loop designs can be exploited to detect and separate the reflected signals. Techniques can be further divided into non-estimating ones, which aim at reducing the effect of multipath without directly estimating its parameters. These include narrow correlator [119], early-late slope [120], strobe correlator [121], high resolution correlator [122], pulse aperture correlator [123] and vision correlator [124]. On the other hand, estimating techniques try to mitigate multipath by estimating parameters of the bias induced by reflection. Among these, there are multipath estimating DLL [125, 126], space-altering generalized expectation-maximization [127], multipath mitigation technique [128], fast iterative ML algorithm [129] and other solutions based on coupled amplitude DLL [130, 131] and LS-based iterative super resolution technique [132]. Among more recent developments in multipath mitigation, it is worth mentioning double-delta correlators [133], combination of multipath estimating DLL with strobe correlators [134] and vector tracking [135].

- **Navigation-unit-based**: GNSS faults can result in inconsistent measurements, including pseudorange, carrier phase, pseudorange rate, and signal strength. Consequently, the fault exclusion can be done by exploiting redundancy of measurements to detect outliers, as done by integrity methods such as RAIM [57, 136, 58] and Advanced RAIM (ARAIM) [137, 138]. Some research also exploit the different C/N0 of reflected signals [139, 140], vector angle grouping with solution separation [141] or sparse estimation [142, 143].

A more comprehensive summary of detection and mitigation techniques is given in [114]. However, it is worth mentioning a few other recent techniques that have been developed. These include the use of cameras [144, 145] and also the application of machine learning algorithms. For example, classification of measurements exploiting 3D maps [146, 147], similarity in azimuth and elevations [148, 149], or correlators [150]. Nevertheless, it is difficult to assess the reliability of these methods due to the complexity of reflection phenomena and of the surrounding environment. Furthermore, they also require additional information or equipment which may not be cheap or viable for mass-market devices.
Furthermore, it should be mentioned that important work on the analysis of the distribution of multipath-affected measurement has been done in [151–153].
The proposed methods presented here both fall in the class navigation-unit-based. The motivation for this choice is that, regardless of the source of the fault, the effect is a bias on the measurements obtained by the receivers. By working in the domain of measurements, the navigation unit has the possibility to detect biases from any source. As a consequence, the proposed methods have the flexibility to be applied more in general for the mitigation of any phenomena which causes biases in the measurements.

5.1.1 Consistency Checking



(a) Space domain. (b) Time domain.

Fig. 5.2 Examples of inconsistency.

A widespread principle for detecting faults is exploiting redundancy to identify measurements that are inconsistent with other available information. The concept of

redundancy can be applied to two domains: the space domain and the time domain. The first refers to inconsistency between measurements from different satellites at a single epoch, and thus checking whether all measurements match a positioning solution under a certain error bound. One example of this is snapshot RAIM, which constructs the chi-squared statistical values from residuals [136]. On the other hand, time domain refers the concept of consistency between consecutive epochs. Filtering based approaches can monitor time consistency using innovations of the filter [154]. Since states should change smoothly according to the motion model of the user, it is possible to check whether current input measurements are consistent with the predicted states, which are based on previous epochs. Figure 5.2 gives a graphical illustration of these two concepts.

In general, integrity methods exploiting consistency in the space domain are very reliable due to their rigorous probability models. However, these methods cannot always be applied directly to all GNSS fault conditions. The background of integrity methods are the precision approaches used in the aviation domain. As a consequence, a key assumption of integrity methods is the availability of a large number of measurements. However, this is often not achievable in urban scenarios due to signal blockage, which leads to limited satellite visibility [113]. Numerous existing solutions are designed for the aviation domain, in which reflection phenomena is not as common and complex as in urban scenarios. Therefore, it may be hard for these methods to identify the correct hypotheses covering a specific fault condition, and the number of hypotheses to be tested would also increase considerably, leading to concerns about computational load.

One advantage of utilizing filtering methods is that they can exploit information from the time domain to compensate for limited redundancy in the space domain, thus partially overcoming the problem of limited satellite visibility in urban scenarios, as done in [154]. Furthermore, the prediction step of filtering methods can be used as a baseline to perform consistency checks, thus eliminating the need to form a large number of hypotheses. In particular, some solutions attempt to increase redundancy by integrating additional sensors like Inertial Navigation System (INS), to compare their consistency with GNSS observables [155]. However, the work presented here aims at designing methods intended for GNSS stand-alone navigation units, since they are the most vulnerable to multipath and NLoS phenomena.

Term	Description	Source
$\boldsymbol{p}_i$	Satellite position	IGS final precise orbit product
$\boldsymbol{p}$	Receiver position	RTK positioning solution
ξ	Antenna phase centre	Absolute IGS correction (igs14.atx)
I_i	Ionosphere delay	Final IGS combined Global Ionospheric Maps (GIM)
T_i	Troposphere delay	Saastamoinen model
D_i	Timing Group Delay (TGD)	Navigation message
c_i	Satellite clock bias	Navigation message
Δs_i	Sagnac effect	Equation (B.1)
Δh_i	Shapiro effect	Equation (B.2)
Δr_i	Relativistic effect	Equation (B.3)

Table 5.1 Corrections and models for pseudorange measurements.

5.2 Post-Processing Mitigation based on Clustering

This section describes a post-processing method for the detection and estimation of biases induced by faults such as multipath or NLoS. One of the main limitations of many multipath methods is the difficulty in properly assessing the effectiveness of the method. In principle, it is hard to obtain a ground truth containing which measurement are affected by a bias and what is the value of the bias. As a consequence, it can be hard to evaluate the performance of new methods in terms of missed detections, false alarms and accuracy of the estimated biases.
The proposed method aims at filling this gap. The idea is to establish a post-processing algorithm with the goal to estimate measurement biases as accurately as possible, using all the available information such as Real Time Kinematic (RTK) and International GNSS Service (IGS) products. Then, the method can be applied on datasets in order to label data, thus creating new datasets with known multipath and NLoS phenomena with as high of an accuracy as possible. Then, the new labelled datasets can be used to evaluate the performances of new methods meant to work in real-time. Furthermore, the labelled datasets can also be used to train machine learning models for the detection of biases.

5.2.1 Modelling of the Fault term

First, we recall the pseudorange equation (2.7) which considers all the error sources. Thanks to the availability of products from IGS, some terms on the right hand-side can be corrected with relatively high accuracy. Furthermore, predicted or estimated terms can be subtracted using a precise position from RTK solution. Then, from the i-th measurement, a leftover term can be obtained:

$$v_i = \Delta c + \Delta\eta + \epsilon_i + F_i \tag{5.1}$$

where Δc is a term which lumps contributions due to the receiver clock and instrumental delay while $\Delta\eta$ lumps all the errors due to inaccuracies in the corrections. A summary of the quantities used for the correction is provided in Table 5.1.
Most importantly, Δc depends only the receiver so it is a common term for every measurement at a given time instant. Furthermore, we assume that $\Delta\eta$ is zero mean and should be very small in general.
Therefore, differences in v_i are mainly driven by the term F_i, which is introduced in order to model biases due to faults, and the receiver noise ϵ_i. However, for geodetic-quality receivers, ϵ_i should be within 1 m of absolute value. On the other hand, ranging bias due to multipath can be as large as 70 m for GPS L1 C/A signals using one chip early-to-late spacing [156], and even larger for NLoS.
This means that measurement from satellites whose fault term F_i is zero should all have leftover terms v_i similar to each other. As a consequence, the leftover term for measurements affected by large values of biases should emerge from the noise ϵ_i. Indeed, such faults are also the most important to detect since they affect the estimated position the most.
By modelling ϵ_i as a random variable following a normal distribution with variance σ^2, then the leftover term is distributed as:

$$v_i \sim \begin{cases} \mathcal{N}(\Delta c\,,\,\sigma^2) & \text{Not Faulty} \\ \mathcal{N}(\Delta c + F_i\,,\,\sigma^2) & \text{Faulty} \end{cases} \tag{5.2}$$

The aim is to compute v_i for all visible satellites and then apply a clustering algorithm to group values that are close to each other. An assumption of this work is that the largest cluster will be produced by the measurements whose fault term is equal to zero or very small.
In principle, it is possible to have a very large number of measurements which are affected by a bias due to a fault. However, it is extremely unlikely that all those

measurements are affect by the same mount of bias, thus forming a large cluster. As a consequence, the assumption of taking the largest cluster is reasonable.
Once the largest cluster has been determined, the common term Δc can be estimated by taking the average value of v_i from the satellites in the main cluster, which are labelled as healthy. On the other hand, measurements not belonging to the main cluster are labelled as faulty, and the fault term F_i is estimated as the difference between the leftover term and the estimated common term $\hat{F}_i = v_i - \widehat{\Delta c}$

5.2.2 Clustering based on DBSCAN

Section 5.2.1 has motivated the use of clustering algorithms to detect measurements affected by bias. This sections instead gives an overview of the chosen clustering algorithm, namely DBSCAN [157]. DBSCAN is a minimum density level estimation that clusters data based on the density. The input to the algorithm are the following parameters:

- ε, the range from a point used to determine whether other points are its neighbours.
- *minPts*, the minimum number of neighbouring points in order to form a cluster.
- *dist*, is a function which computes the distance between two data points. Different vector norms can be specified by the user. In our case, the leftover term is a one-dimensional value so *dist* computes the absolute value of the difference between two points.

Based on these parameters, every data point will be classified into one of the three following types:

1. **Core points**, data points that have more than *minPts* neighbours within a range ε and thus belong to a cluster.
2. **Non-core points (border points)**, data points who have at least one neighbour but not as many as *minPts*.
3. **Outliers**, data points with no neighbours.

First, DBSCAN starts from a point and identifies all its neighbours. If the point is a core points, then all its neighbours are assigned to the same cluster and the

operation is repeated for each of them, thus including also their neighbours which were not neighbours of the original point. DBSCAN then continues with other points until each point in the data has been assigned to a category.
A condition that we set is that the largest cluster contains at least two measurements, so $minPts = 2$. This is done in order to avoid extreme cases in which only a limited number of satellites is in view and therefore it is hard to determine which are affected by bias.
On the other hand, the range ε should be set according to the variance of the receiver noise. It should be noted that, if a high-quality receiver is used which has low noise, then a low value of ε can be set, thus improving the detection probability even for small biases.
Although DBSCAN uses three groups to label points, we are interested only in two groups. Satellites that are labelled as healthy are associated to core points that belong to the largest cluster, while all other satellites are labelled as faulty.

5.2.3 Results

Ideally, the proposed method should be evaluated on a dataset containing a ground truth for measurements biases. However, as we mentioned in the introduction of the chapter, it is hard to obtain such a dataset and that is indeed the scope of the work itself. Since it is not possible to test the estimated biases directly, what we can do is to remove those biases from the available measurements and test whether the positioning solution obtained from the corrected measurements is improved w.r.t. to initial dataset when comparing to a ground truth of the position. In our case, the position ground truth is the one obtained from RTK. It should be highlighted that nonetheless, the scope of the work is to provide a method to obtain measurement biases ground truth rather than improving the position estimation performance for real-time application.

Experiment Set-up

To demonstrate the proposed method, a real-world experiment is carried out in a urban environment where multipath phenomena can be expected to occur. The data collection is performed in the city of Turin (Italy), as shown in Figure 5.3. A Leica GS18 receiver with its internal antenna collects raw GNSS measurements of GPS L1 C/A signals with a frequency of 10 Hz. The antenna is fixed in the same location for the entire duration of the data collection, which is around 40 minutes. The precise

(a) Surrounding buildings of data collection scenario.

(b) Overhead location of the GNSS antenna (orange icon).

Fig. 5.3 Mutlipath/NLoS scenario of data collection.

location of the antenna is obtained by a fixed RTK solution, thus providing the ground truth of the positioning point for the computation of the leftover term in (5.1). The length of the experiment was chosen to have a meaningful change in the position of satellites in the sky across the duration, in order to observe some changes in the reflection phenomena. The skyplot and associated C/N_0 for each satellite are shown in Figure 5.4. The receiver is deployed close to the buildings on the east side of the road. Therefore, signals coming from that direction could be in NLoS condition and thus reflect on the buildings on the west side.
All corrections and models are computed using the manual and program library provided by RTKLIB [158].
Regarding DBSCAN, the parameter ε should represent the maximum distance between two points within the same cluster. In our specific context, values of v_i belong to the distribution $\mathcal{N}(\Delta c, \sigma^2)$ where the variance σ^2 is associated with the receiver noise. Prior to the experiment, the receiver was deployed in an open sky scenario in order to estimate the variance of measurements in ideal conditions. Based on such data, the variance of receiver noise is observed to be around 1 m. Therefore, considering the difference of noise between two independent measurements, the maximum distance between points in the same cluster ε was set to be 2 m.

Static experiment

Figure 5.5a depicts all the v_i, marked in different colours to distinguish the corresponding satellite. It can be seen that most v_i exhibit a common trends, thus indicating their dependency on a common term, namely Δc. Conversely, a few

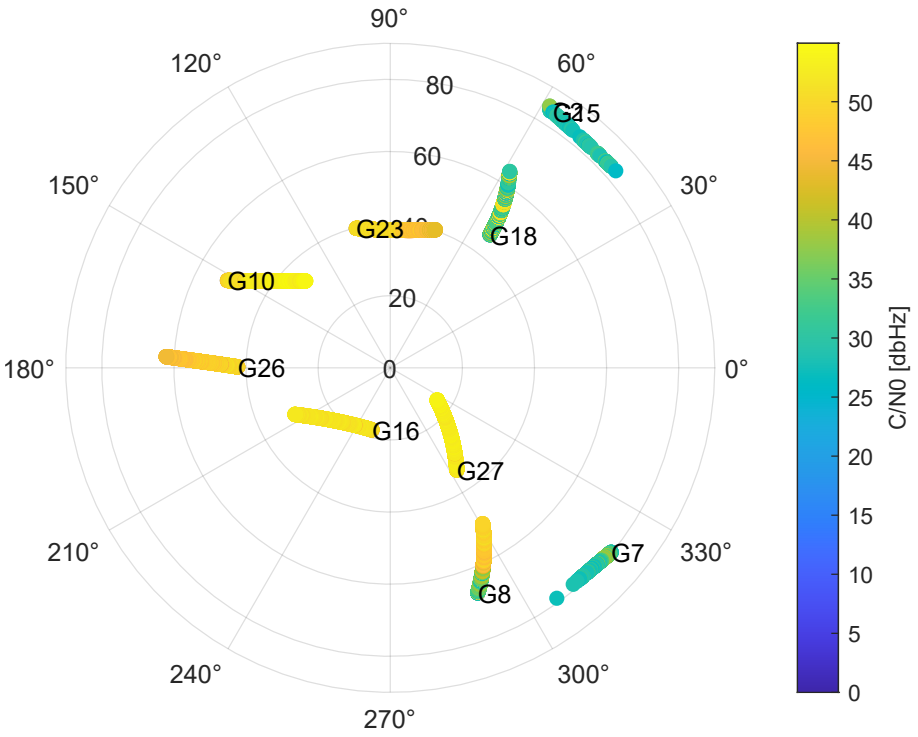
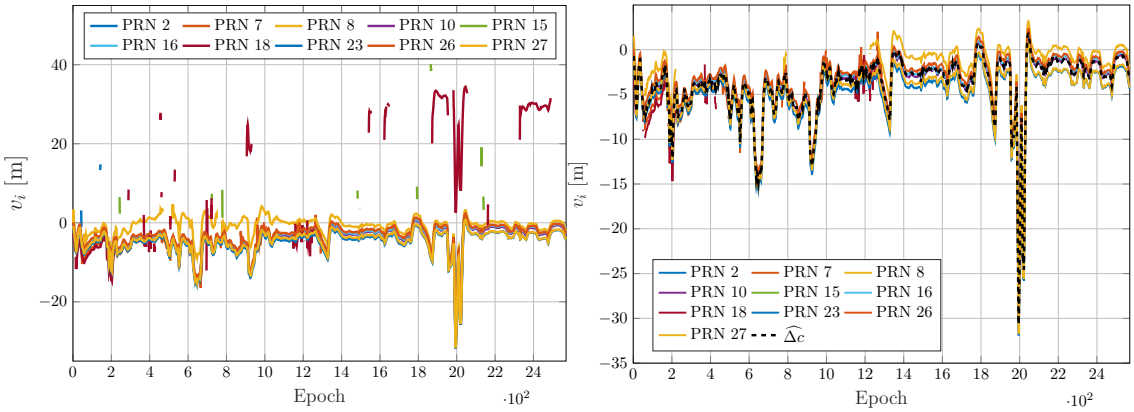


Fig. 5.4 Sky-plot and corresponding C/N_0.

values deviate from this trend, suggesting an additional impact due to a time-varying bias. As a consequence, these deviations can indicate the influence of faults such as multipath or NLoS. In any case, regardless of the source, our goal is to estimate such bias. In particular, it is interesting to observe the behaviour of PRN 18, which produced a leftover term v_i significantly different compared to other measurements. Based on the sky-plot in Figure 5.4 and the surroundings of the receiver in Figure 5.3, it can be hypothesized that the bias could be due to a reflection of the signal from the building on the west side of the road.

Then, Figure 5.5b shows all the v_i in the largest cluster identified by using DBSCAN. As a result, the mean of these values is computed as the estimated Δc and plotted as the dashed black line. Given this estimate $\widehat{\Delta c}$, it can now be subtracted from v_i of measurements not in the largest cluster, effectively isolating the fault terms F_i.

Then, Figure 5.6 presents the estimated biases computed by removing the common term due to the clock. It can be seen that positive values of F_i are observed more frequently than negative values. This is consistent with the theoretical background on the effect of multipath and NLoS as described in [114]. Multipath can introduce both positive and negative biases to pseudorange depending on the distortion of the discriminator function, while NLoS tends to produce only positive biases since only one delayed ray is received. Therefore, positive-value biases tend to occur slightly



(a) Residual terms v_i for each GPS satellites. (b) v_i in the largest cluster and the estimated $\widehat{\Delta c}$.

Fig. 5.5 Clustering of fault free measurements and estimation of common clock term.

more often.

It is now of interest to assess the validity of the estimated biases. To this aim, a new dataset is generated which contains compensated measurements obtained by subtracting the estimated fault terms $\hat{F}_i$ from the original measurements. Consequently, new positioning results can be computed using these compensated values and compared to the performance of positioning solutions when no detection or mitigation strategies are employed. Both datasets with and without compensated measurements are run using an OLS estimator, and the former is referred to as C-OLS. Figure 5.7 compares position scatters in the horizontal plane of a ENU reference frame for both datasets, along with the mean and one σ covariance ellipse and the ground truth. It can be seen that removing the estimated fault terms greatly reduces the spread of positioning solution. On the other hand, the mean for the compensated dataset is slightly further from the ground truth, since some solutions affected by bias where skewing the mean of the original dataset. In any case, the large majority of points is now significantly closer to the ground truth.
Similarly, Figure 5.8 shows the positioning error time series in ENU directions for the two cases. From this plot it can be observed that the positioning errors are smaller in magnitude than the estimated fault terms F_i, for the original dataset. This occurs because the solutions are computed based on the entire set of available pseudorange measurements, out of which only a minority is affected by a bias. Nonetheless, the solutions using the compensated measurements greatly reduces the error in all three axial directions, especially preventing large deviations which were present in the original dataset. Finally, Figure 5.9 shows the CDF of positioning errors, serving a crucial role in assessing the enhanced positioning accuracy after compensation

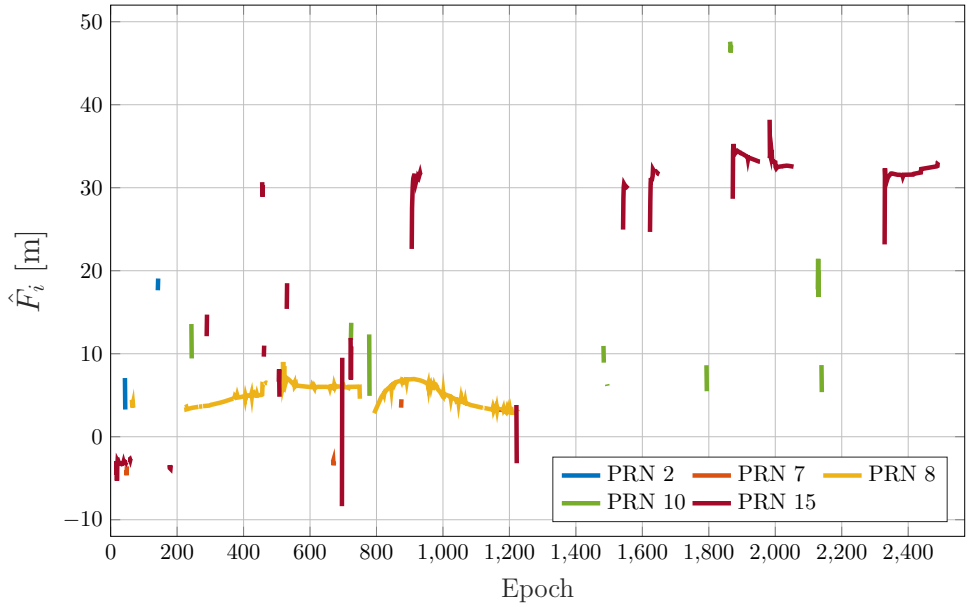


Fig. 5.6 Estimated fault terms $\hat{F}_i$ from measurements outside the main cluster.

of the bias. Significant reductions in positioning errors can be observed, providing substantial validation for the effectiveness of the proposed method.
Results show that the proposed method is a promising tool for an automated detection and estimation of measurement biases due to faults. This approach can be leveraged to obtain ground truths that can then be used by other real-time methods to assess their performance. Nonetheless, some improvements can be made in the future in order to further validate the method and to improve its accuracy. More on-field test can be performed to assess the validity in a wide range of conditions and with different receivers. Other clustering algorithms can be tested and a more rigorous derivation of the parameters can be performed in given certain target probabilities of detection or false alarms.

5.3 Real-Time Mitigation based on Soft Information

The previous section introduced a solution which requires the knowledge of a precise position from RTK and precise products in order to detect faulty measurements and estimate their bias. Therefore, it is not suitable for real-time applications. As a consequence, the definition and statistical modelling of the leftover term used in (5.1) needs to be adjusted accordingly. Simpler correction models can be used in place of

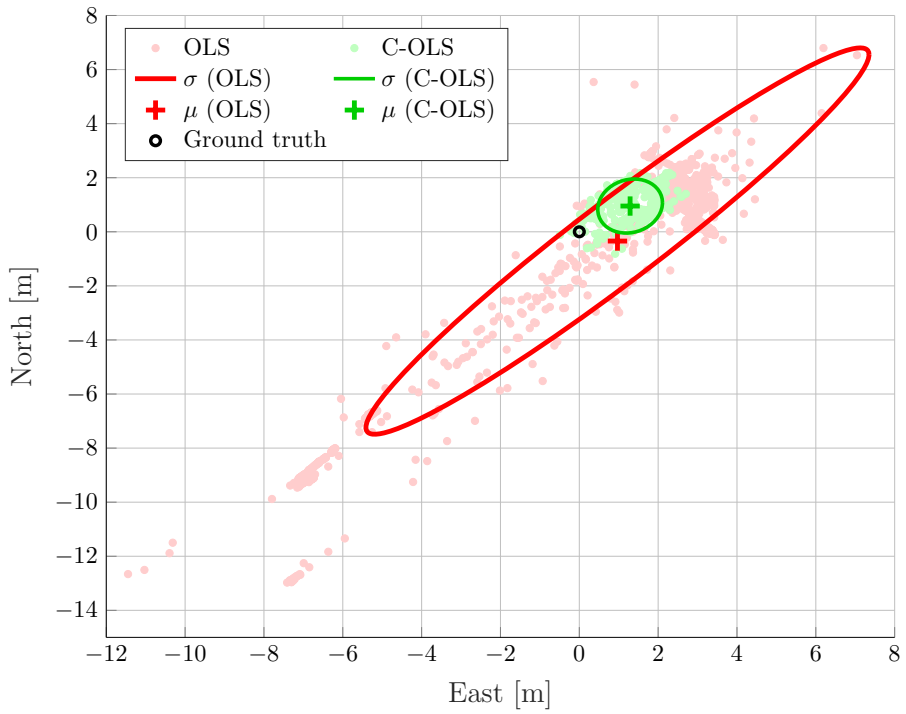


Fig. 5.7 Scatter plot in the horizontal plane of ENU reference frame. Comparison of positions computed with compensation (C-OLS) and without (OLS) of fault terms $\hat{F}_i$, along with mean and standard deviation.

precise products from IGS, which are only available for post processing. On the other hand, (5.1) was effective in detecting fault thanks to the availability of a ground truth provided by the use of RTK. For real-time applications, the alternative is to use the concept of innovation as described in (4.12). This opens up the possibility of employing either the EKF or the PF, both introduced in Chapter 4. A comparison between their properties will provided in Section 5.3.2, specifically with the goal of analysing their robustness to biases in the input measurements.

5.3.1 Process Noise for Land Vehicles

In order to compare Bayesian filter, first we need to discuss more in-depth the role of the process noise in the estimation process, especially regarding the update step of filtering algorithms. In filtering approaches, the process noise is tuned based on the motion of the user, how accurate the dynamic model is and the rate of solutions provided by the navigation unit. A discrete-time dynamic model with high-order

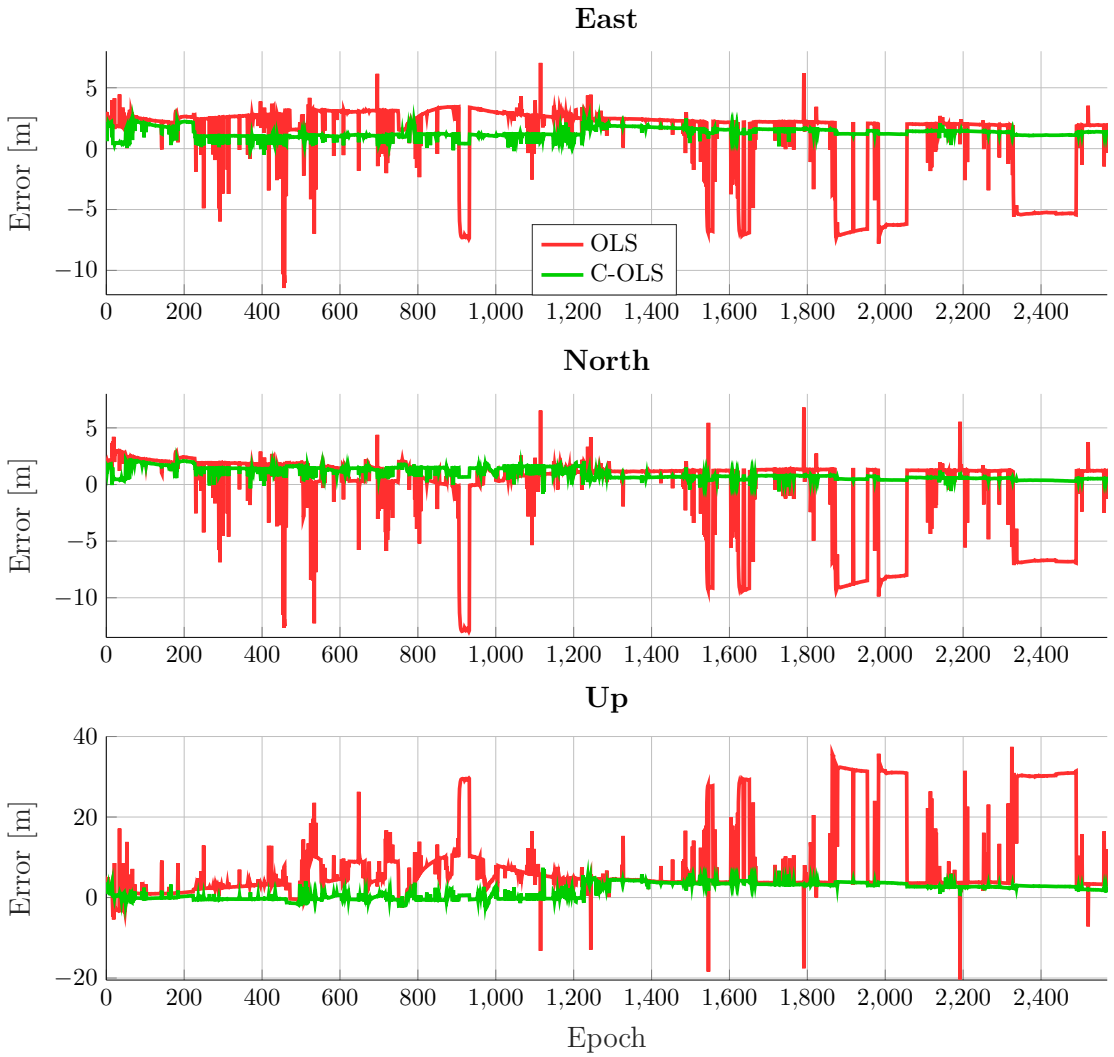


Fig. 5.8 Time series of position axis errors in ENU reference frame. Comparison between compensation (C-OLS) and without (OLS) of fault terms $\hat{F}_i$.

terms at time k can be written as:

$$\begin{aligned} \boldsymbol{p}_k &= \boldsymbol{p}_{k-1} + \boldsymbol{v}_{k-1}\Delta t + \frac{1}{2}\boldsymbol{a}_{k-1}\Delta t^2 + ... \\ c_{b,k} &= c_{b,k-1} + \dot{c}_{b,k-1}\Delta t + \frac{1}{2}\ddot{c}_{b,k-1}\Delta t^2 + ... \end{aligned} \tag{5.3}$$

To make this model computation feasible, (5.3) is usually truncated before the terms regarding the acceleration $\boldsymbol{a}_{k-1}$ and receiver clock drift rate $\ddot{c}_{b,k-1}$. Furthermore, an estimation of high-order terms is usually not available, especially if there is no use of other sensors such as INS. As a result, the magnitude of all the neglected terms containing Δt and high order derivatives should be taken into account when tuning

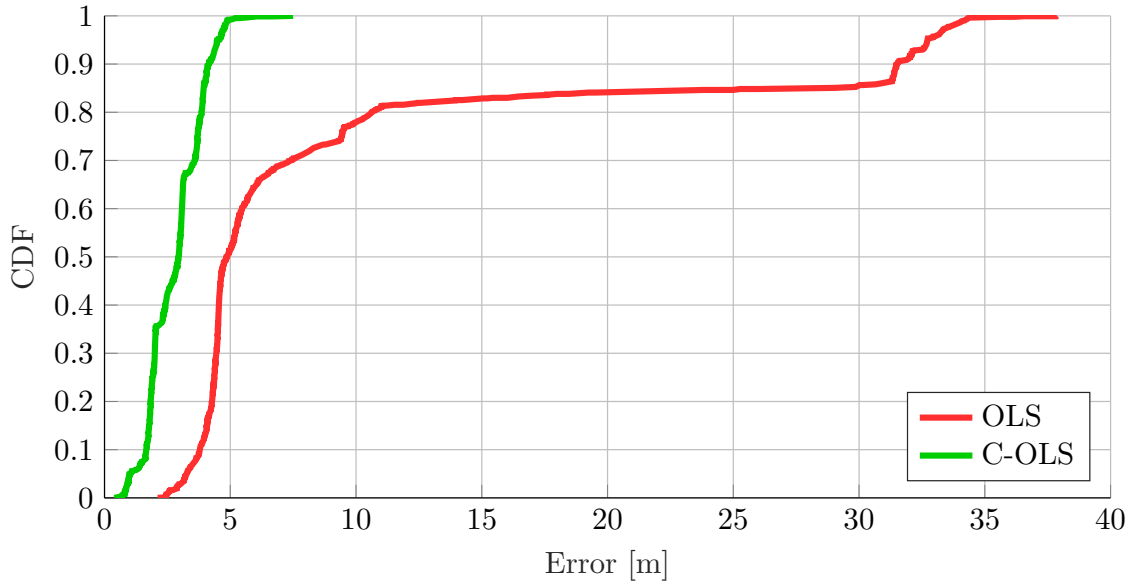


Fig. 5.9 CDFs of positioning errors using compensation (C-OLS) and without (OLS) of fault terms $\hat{F}_i$.

the process noise. The meaning of this operation is the following: the real motion of the receiver depends on its velocity and other high-order derivatives. Since the latter are not known or cannot be estimated reliably, they are modelled statistically by considering them as an uncertainty. This is the role of the process noise.
Usually, the typical measurement frequency of GNSS for land vehicles is equal to or higher than 1 Hz ($\Delta t < 1\,\mathrm{s}$). Therefore, high-order terms tend to be negligible because higher powers of Δt become increasingly smaller. As a consequence, the value of process noise should mainly be tuned based on the acceleration $\boldsymbol{a}_{k-1}$ and the receiver clock drift rate $\ddot{c}_{b,k-1}$. In particular, their maximum values should be obtained according to the specific use case and receiver, and then the process noise can be tuned to account for the truncated terms in (5.3). Based on the three-sigma rule of the normal distribution, the standard deviation of states is set as 1/3 of the maximum unmodelled terms containing $\boldsymbol{a}_{k-1}$ and $\ddot{c}_{b,k-1}$ in the dynamic model. Therefore, the diagonal covariance matrix is written as:

$$\boldsymbol{Q}_{k-1} = diag \begin{bmatrix} \left(\frac{1}{3}\frac{1}{2}\boldsymbol{a}_{max}\Delta t^2\right)^2 \\ \left(\frac{1}{3}\boldsymbol{a}_{max}\Delta t\right)^2 \\ \left(\frac{1}{3}\frac{1}{2}\ddot{c}_{b,max}\Delta t^2\right)^2 \\ \left(\frac{1}{3}\ddot{c}_{b,max}\Delta t\right)^2 \end{bmatrix} \tag{5.4}$$

Given an example in vehicular scenarios, the maximum acceleration is typically in the order of $a_{max} = 3\,\mathrm{m/s^2}$ for each direction under a ENU frame. A typical value of the measurement interval Δt can be 0.1 s. Therefore, deviation between the real motion due to acceleration and a linearized model is $\frac{1}{2}a_{max}\Delta t^2 = 0.015\,\mathrm{m}$, which is fairly small. The specific values related to the dataset used to test this work are provided in Appendix C.

Properly tuning the process noise to reduce the range of possible values of states has another benefit. Monte Carlo methods such as PF suffer from the curse of dimensionality, meaning that the amount of samples needed to represent a probability distribution given a target accuracy increases exponentially with the number of states to be estimated (i.e., the number of dimensions of the distribution). Since these methods aim to approximate probability distributions with a set of samples, the number of samples needed to reach a target accuracy also depends on the range of possible values of the hidden variables. Properly tuning the process noise can help reduce the span of possible values and thus the number of samples needed to accurately sample the target distribution.

5.3.2 Robustness of Bayesian Filters

This subsection will discuss more in detail the advantages of using the PF, to provide a theoretical basis for the motivation of using this filter in the targeted application. The EKF is used as a reference to be compared with the PF. For simplicity, the illustration and discussion in this section are based on pseudorange measurements. Nonetheless, the advantages of the proposed method can also be applied to the pseudorange rate measurement model.

In certain scenarios, measurement biases caused by faults may be small, and their impact on the innovation can be overwhelmed by factors such as the dynamic model of the user, atmospheric model errors, and other terms. Consequently, small valued errors may lead to missed detections. While these faults may go unnoticed, they still contribute to the degradation of the accuracy of navigation solutions. Considering the update step of the EKF, the presence of faults affecting a set of input measurements $\boldsymbol{y}_k$ at time k can be collected in a vector F_k, then (4.15) can be expanded as:

$$\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^* + \boldsymbol{K}_k(\bar{\boldsymbol{y}}_k + \boldsymbol{F}_k - \boldsymbol{H}_k\boldsymbol{x}_k^*) \tag{5.5}$$

where $\boldsymbol{K}_k$ is the Kalman gain from (4.14) and $\bar{\boldsymbol{y}}$ are the input measurements without the fault term. Hence, the additional error on the estimate $\Delta\hat{\boldsymbol{x}}_k$ due to the faults

can be written as:

$$\Delta\hat{\boldsymbol{x}}_k = \boldsymbol{K}_k \boldsymbol{F}_{\rho,k} \tag{5.6}$$

which means the error can grow indefinitely as faults get larger. As a consequence, the accuracy of EKF can suffer severely in case of large undetected biases in the measurements.
On the other hand, recall that the PF derives its estimated states through a weighted average of particles as computed in (4.26). As a results, the final estimate is constrained by where the distribution has been sampled, or in other words where the particles are. In turn, the mean value of particles depend on the previous estimate and the motion model, while their spread depends heavily on the process noise, which is not large for vehicular applications. As a consequence, the estimate of a PF over a single epoch cannot deviate considerably from the predicted value, even in presence of large biases in the input measurements.

5.3.3 Soft Information

In complex urban environments environments, it can occur that multiple measurements are affected by biases at the same time. As a results, it can become hard to detect which ones are faulty based on the principle of consistency. Popular integrity techniques detect faulty measurements by exploiting residuals, which are computed based on the estimated position. However, when many measurements are faulty, the estimated position is affected and hence residuals are not very reliable. A similar reasoning can be applied to innovations, which also become less reliable if the predicted position is far from the truth. In such scenarios, it would be wise to be cautious of large innovations. However, most fault detection techniques follow the principle of single-value estimates. In other words, most methods perform hard decisions in labelling measurement either healthy or faulty. This can lead to many missed detections and false alarms, which may further hinder the performance of navigation units.
This approach of hard decisions was also used in Section 5.2 to determine which measurements are faulty. However, that was applied to a post-processing scenarios which relied on accurate corrections and ground truth. Hence, the leftover term tends to be more reliable w.r.t what can be computed in real-time applications. In such cases, we can exploit the concept of soft information, as proposed by [112] in the context of localization. The idea is to represent knowledge about a variable in a statistical way, rather than assigning a single-value (possibly associated with an uncertainty).

Soft information allows to obtain a statistically model to the notion that maybe a measurement is faulty, but we are not so sure of it and it may also be healthy. Such characterization thus imply that measurement errors can be represented with any arbitrary probability distribution, possibly even multimodal. This characterization can only be implemented on the PF, since the EKF assumes errors to be normal distributed random variables. This feature thus provides further motivation for implementing the proposed method using the PF, on top of the analysis made in Section 5.3.2.

It should be pointed out that the concept of soft information is very flexible and can be applied in order to integrate into navigation filters any kind of available information (e.g., maps) and not simply measurements. However, the scope of this work is mainly to give an initial implementation and assessment of the potential of exploiting soft information.

5.3.4 Proposed PF Measurement Model

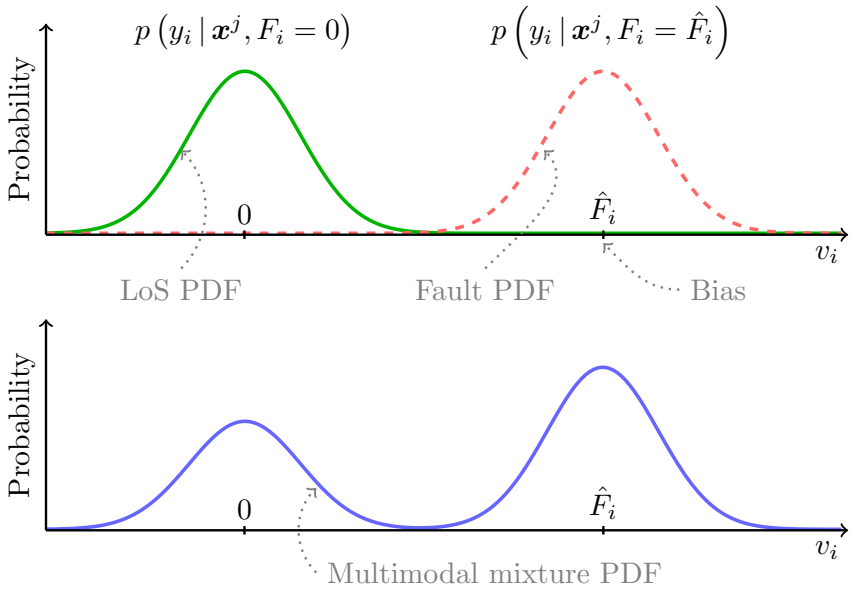


Fig. 5.10 Example of updated measurement likelihood for a single measurement. Each curve represents the probability distribution of input measurement from i-th satellite y_i given the position of j-th particle $\boldsymbol{x}^j$ (thus representing the innovation v_i^j) under different assumptions about the estimated fault term $\hat{F}_i$.

First, we start by deriving the pseudorange innovation v_i from satellite i for the PF:

$$\begin{aligned} v_i &= \rho_i - \rho_i^* \\ &= \Delta r_i + \Delta c_b - \Delta c_{b,i} + \Delta I_i + \Delta T_i + \Delta R_i + F_i + \epsilon_i \end{aligned} \tag{5.7}$$

where:

- $\Delta r_i = r_i - r_i^*$ is the difference between the true and predicted geometrical ranges.
- Δc_b is an error in the prediction of the receiver clock bias.
- $\Delta c_{b,i}$ is a residual error from satellite clock bias correction.
- ΔI_i, ΔT_i, ΔR_i are residual error due to imperfections in the corrections of ionosphere delay, troposphere delay and relativistic effects, respectively.
- F_i is an additional bias due to faults such as multipath.
- ϵ_i is noise due to receiver.

When there is no fault, $F_i = 0$ and according to (5.7), the pseudorange innovation value depends on all the terms except F_i. That is to say, the accuracy of corrections, dynamic model and filter accuracy of the previous epoch determine the value of the innovation, except for the fault term.
Consequently, if there are no faults and the filter does not diverge, the innovation should be bounded, as will be verified in the experiment part. However, the innovation might grow large when a fault happens, since it can introduce large errors to pseudorange.
The proposed method is to compute the innovation according to the top part of (5.7) and then to set a threshold v_{th} based on the expected accuracy of all the non-fault terms. As such, we do not need to explicitly compute all the terms in the bottom part of (5.7) (which would not be possible to do since some terms cannot be known), but rather only need an estimate of their variances. Then, if the innovation is larger than the defined threshold, the corresponding measurement is labelled as potentially faulty. The threshold can be computed based on the expected accuracy that can be achieved for all the terms in (5.7). A more in-depth discussion of these terms and how they are computed is given in Appendix C.
Finally, it is important to point out that, for each satellite, the corresponding pseudorange innovation contains information from both the previous epoch (predicted

measurements) and the current epoch (observed measurements), so it can be used to check for consistency in the time domain. On the other hand, consistency in the space domain can be examined by looking at multiple innovations from different satellites at the same epoch.

When the innovation is larger than the threshold, it is assumed that its value is mostly driven by the fault term, so $\hat{F}_i$ is estimated as the value of the innovation:

$$\hat{F}_i := \begin{cases} v_i & \text{for } v_i \geq v_{th} \\ 0 & \text{for } v_i < v_{th} \end{cases} \tag{5.8}$$

By substituting $\hat{F}_i$ into (2.6), a modified measurement model for the PF is obtained, which accounts for possible biases. This design is graphically shown at the top of Figure 5.10, which compares the original likelihood function and the updated one which accounts for estimated bias due to multipath or NLoS.
The unnormalised particle weights at epoch k obtained from the original likelihood function can be represented as:

$$w_k^j = w_{k-1}^j \, p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^j, \boldsymbol{F}_k = \mathbf{0}) = w_{k-1}^i \prod_{i=1}^{M} w_{i,k}^j. \tag{5.9}$$

While the unnormalised particle weights obtained from the updated likelihood function can be computed as:

$$\tilde{w}_k^j = w_{k-1}^j \, p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^j, \boldsymbol{F}_k = \hat{\boldsymbol{F}}_k) = w_{k-1}^j \prod_{i=1}^{M} \tilde{w}_{i,k}^j. \tag{5.10}$$

Clearly, if a measurement is not labelled as faulty, then $w_{i,k}^j$ and $\tilde{w}_{i,k}^j$ are the same since $\hat{F}_i = 0$ according to (5.8). As a consequence, the weighting strategy remains unchanged from the legacy PF.
Conversely for faulty measurements, the concept of soft information can be leveraged by combining the original likelihood function and the updated one. By integrating these two sources of information, the method can effectively take into account the possibility of false alarms and improve the overall fault mitigation performance. In particular, we propose an adaptive strategy to combine these two likelihoods together to obtain a new multimodal likelihood function used to determine the weight of each particle. The bottom half of Figure 5.10 depicts this adaptive combination design. The motivation for employing an adaptive weighting strategy between the two PDFs

is motivated by the following reasoning: using the innovation to detect multipath and NLoS and estimate the associated bias is more reliable when only one or two satellites are affected. In such cases, correcting the measurements can prove to be the most effective strategy. Conversely, when multiple measurements are impacted by biases at the same epoch, the reliability of the innovation decreases. Therefore, the confidence in the likelihood $p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^j, \boldsymbol{F}_k = \hat{\boldsymbol{F}}_k)$ should be decreased. As a result, we can use both hypothesis to build new unnormalised particle weights $w_{adp,k}^j$ which combine the two likelihoods adaptively:

$$w_{adp,k}^j = h_0 \cdot w_k^j + h_1 \cdot \tilde{w}_k^j \tag{5.11}$$

where the weights h_0 and h_1 associated to the two fault hypothesis can be computed according to the proposed adaptive strategy:

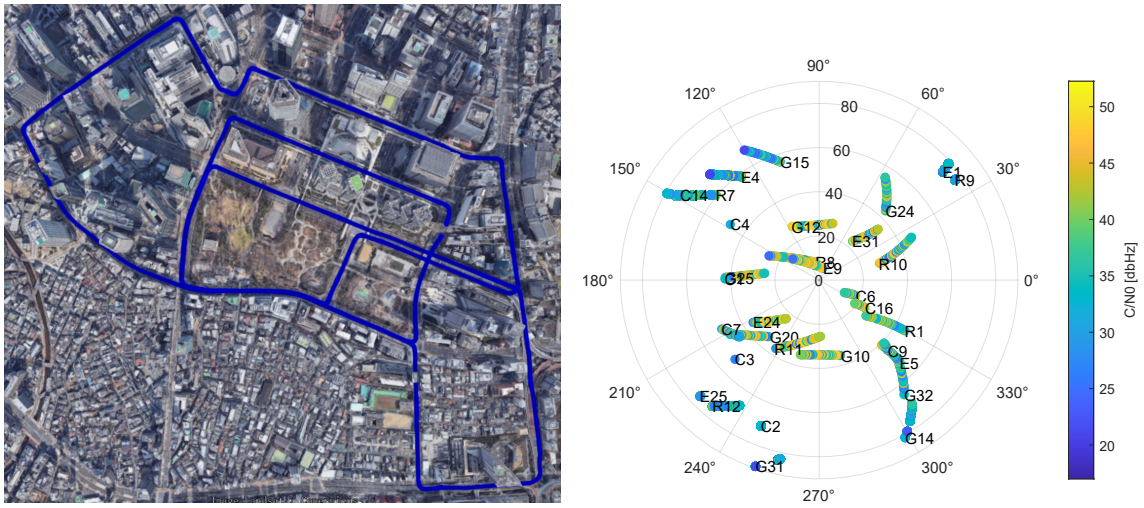
$$\begin{aligned} h_0 &= \frac{\text{\# detected faulty measurements}}{\text{\# of measurements}} \\ h_1 &= 1 - h_0 \end{aligned} \tag{5.12}$$

The idea behind the proposed strategy is the following: as more measurements are labelled as faulty, the innovation is less reliable and we give less priority to the updated likelihood. On the contrary, if only a small fraction of innovations exceeds the threshold, we can have more trust that those measurements are indeed faulty and hence h_0 is lower and h_1 is larger. As we have established, w_k^j and $\tilde{w}_k^j$ are the same for healthy satellite so the weighting strategy remains unchanged regardless of the values of h_0 and h_1.

5.3.5 Divergence protection

The effectiveness of the proposed method relies on the assumption that the navigation solution from the previous epoch maintains acceptable accuracy. If the filter produces a solution with large deviation fro the truth, it becomes challenging to estimate reliable innovation values. Consequently, it can be useful in some instances to re-initialise the filter to avoid this problem. A GLS estimator with classical RAIM-FDE is employed to monitor the divergence of filtering solutions. The navigation solution is considered not reliable if one the following two conditions is true:

- The distance between positions obtained by the filter and GLS is larger than 50 m while the visible satellite number is larger than 12.



(a) Vehicle trajectory in the Shinjuku dataset.

(b) Sky plot with signal strength C/N0.

Fig. 5.11 Experimental scenarios.

- The receiver has lost lock on all satellites for more than a second.

When this scenario occurs, the filter is re-initialized using the solution from GLS. Clearly, this is only possible if enough measurements are available to obtain a solution using GLS.
The number of epochs in which a filter does not provide a valid solution is used to calculate the availability metric that will be tested in the results section.

5.3.6 Experiment and Results

The goal of this section is to validate the performance of the proposed PF-based method, which is named PF-ADP in the following discussion. In order to provide a benchmark performance, we also implemented two classical methods as references. The first is based on RAIM-FDE, as described in [113]. In this method, the fault exclusion assumes that at most one satellite is faulty. If the chi-squared test cannot be satisfied after testing all fault subsets, the navigation solution is marked as invalid. The second method is an EKF-FDE, which follows the approach proposed in [154]. This methods was applied using the same parameter as the PF-ADP in terms of covariance of measurements and process noise.

Experimental settings

The test is performed using real data from the UrbanNav dataset [159] provided by Hong Kong Polytechnic University. The dataset was collected in Shinjuku, Tokyo, in harsh GNSS scenario with numerous tall buildings and narrow roads, as depicted in Figure 5.11a. The sky plot with C/N0 is shown in Figure 5.4. As can be seen, many satellites have low C/N0 or large fluctuations of its value.
Errors on state estimates are computed w.r.t. a reference trajectory obtained with an RTK/INS integration system.
The rate of available measurements for PVT is of 10 Hz. Measurements are computed using signals from four constellations, including GPS L1 C/A, GLONASS G1 C/A, Galileo E1, and BeiDou B1I.
The particle number of the PF is fixed as 1000 and the resampling threshold is set to 0.1. The value of the threshold on the pseudorange innovation v_{th} is based on the accuracy of filter solution and corrections as discussed in Appendix C and further tuned based on empirical attempts. The final value chosen for the experiment is $v_{th} = 5\,\text{m}$. First we will discuss and analyse the results regarding the detection of faulty measurements and the estimation of their bias, and then the positioning performance of the proposed PF-based mitigation method compared to some benchmarks methods from the literature.

Detection Results

First of all, Figure 5.12 shows the absolute value of innovation $|v_i|$ estimated at all epochs compared to the threshold v_{th} for some selected satellites of different constellations. The letters associated to the PRNs follow the RINEX naming convention.
As can be seen, some parts of the trajectory are particularly challenging in terms of visibility, and many satellites display large innovations at the same time. In particular, GPS PRN 32 displays large innovations of up to 100 m or more, which do not occur for satellites belonging to other constellations. We can also see from the skyplot in Figure 5.11b that PRN G32 has a low elevation, which could explain this behaviour. On the other hand, PRNs like E9 and R8 have very high elevation, and as expected tend to produce small innovation for most of the experiment.
Figure 5.13 shows, for each visible PRN, the percentage of epochs in which measurements are labelled as faulty. This percentage is computed as the ratio of epochs in which an innovation exceeds the threshold over the total number of epochs in which the PRN is visible. For this reason, some PRN with low elevation like G31 tend to

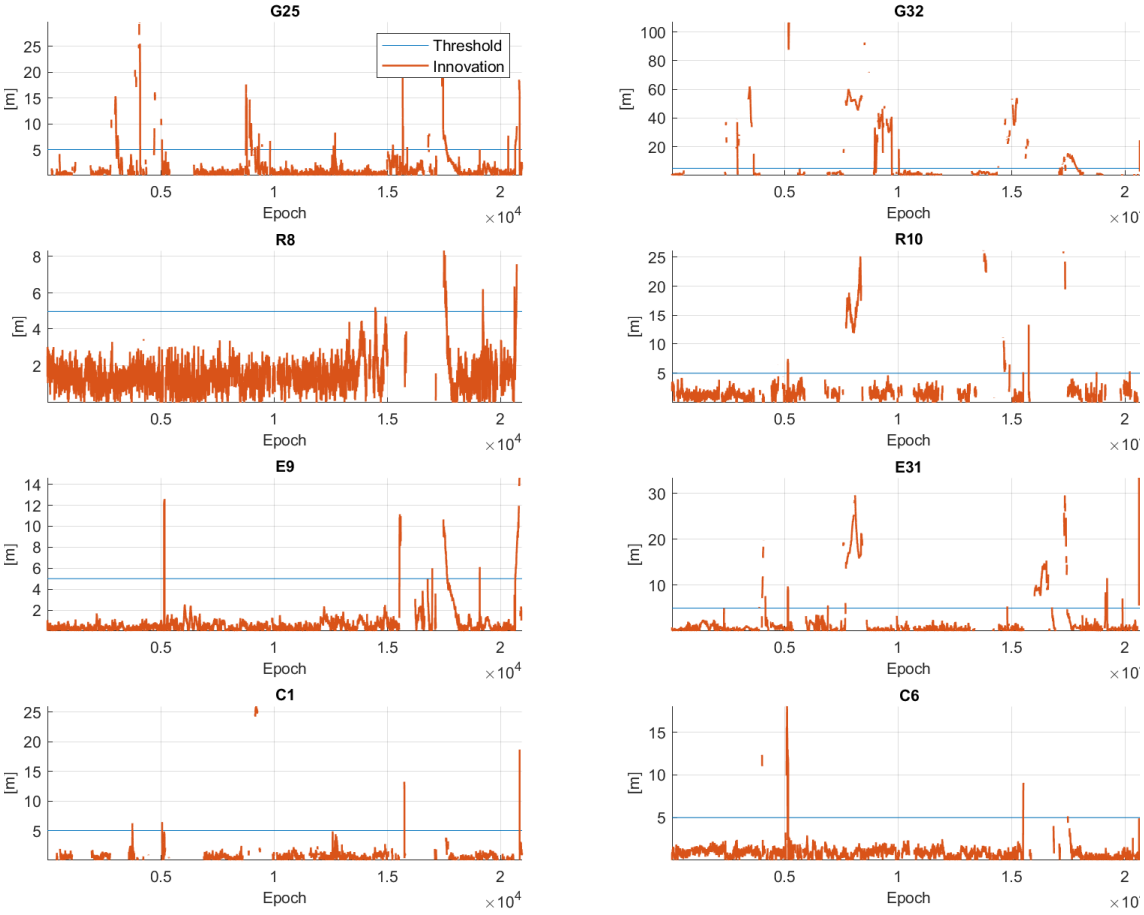


Fig. 5.12 Timeseries of absolute value of the innovation compared to the threshold for some selected satellites.

have high percentages, since they are only visible for a few epochs and are more likely to suffer from reflection phenomena. The results from the histogram in Figure 5.13 show that pseudorange measurements from GPS satellites tend to be affected more often, at least for this dataset.

Now that the performance in terms of fault detection has been discussed, we can analyse the positioning performance of the proposed method based on soft information to mitigate the effect of biases. If the detection of faults is reliable and so is the estimation of the bias, then the performance of the filter should benefit and provide lower errors

Positioning Results

First, let us look at the behaviour of the 3D position error compared to the number of measurements labelled as faulty in Figure 5.14. As can be seen, the error tends to

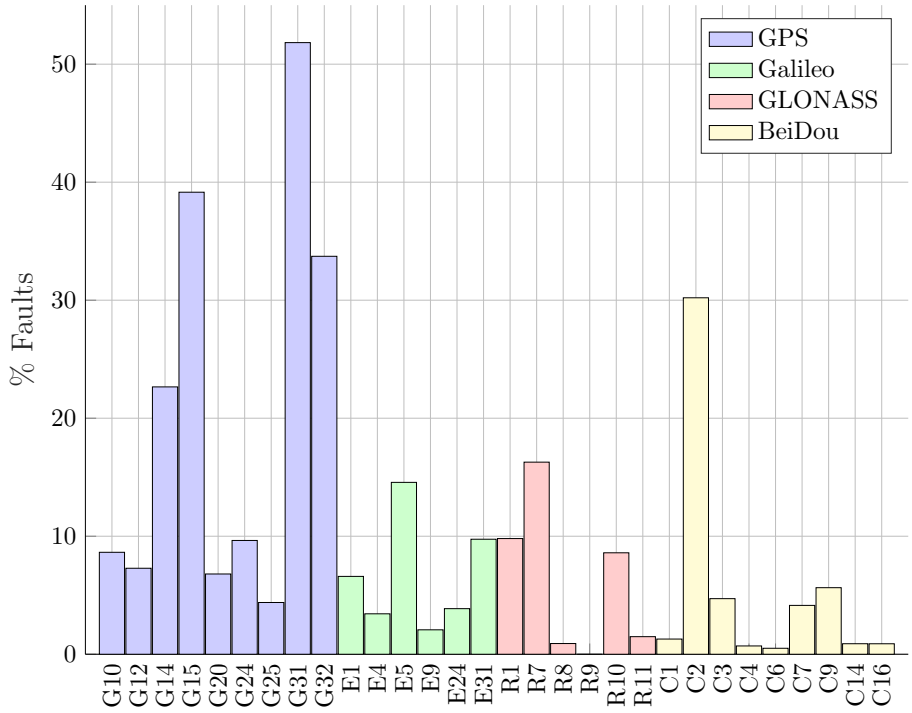


Fig. 5.13 Histogram of the percentage of pseudoranges labelled as faulty for each visible satellite.

be larger for all estimators in parts of the experiment when many measurements are faulty. However, there are some behaviours that should be analysed more in detail. Around epoch 8000 in the dataset, a large number of measurements are faulty at the same time for a considerable span of time. During such event, the solution of RAIM-FDE is unavailable, as can be seen from the missing solutions in the pot subplot of Figure 5.14. On the other hand, filtering solutions maintain a surprisingly low error during this time. This could be because the motion of the user matches the dynamic model of the filter and thus the propagation of previous solutions is a good prediction of the position.

Conversely, in the part of the trajectory between epoch 14000 and 17000, all estimators seem to suffer from degradation of the performance. As a matter of fact, during this section, the number of faulty measurements tend to be fairly high and also changes rapidly. As a consequence, it can be hard to obtain reliable values of the innovations and thus correctly identify which measurements are faulty or not. In any case, the proposed PF-ADP shows better performance in terms of error, and is especially effective in avoiding large errors w.r.t. other methods.

The 3D RMSE and mean error for each filter are given in Table 5.2, along with the percentage of epoch in which a PVT solution is considered valid. The table also

Method	RMSE [m]	Mean error [m]	Availability
RAIM-FDE [113]	15.682	6.935	70.00%
EKF-FDE [154]	11.311	5.578	99.02%
[141]	N.A.	6.15	N.A.
PF-ADP	7.691	4.065	99.02%

Table 5.2 Comparison of 3D RMSE and mean error for different mitigation methods.

Method	Percentile errors [m]					
	25	50	75	90	95	98
RAIM-FDE	1.482	2.769	5.737	14.730	31.198	51.126
EKF-FDE	1.395	2.489	5.923	11.774	17.269	38.953
PF-ADP	1.204	2.029	4.269	8.648	13.659	20.192

Table 5.3 Comparison of percentile errors for different mitigation methods.

includes the result from another work using the same dataset [141], for which only the mean error is available.

A zoom in of two challenging sections of the trajectory can be seen in 5.15. In particular, it is interesting to notice how the proposed method can provide very stable solutions with relatively smaller fluctuations w.r.t. the other methods. On top of that, the PF-ADP also does not deviate too much from the ground truth in any point and does not display any systematic bias w.r.t. the reference trajectory.

Figure 5.16 shows the CDF of horizontal and vertical error of each estimator in ENU reference frame. Both plots in the figure are shown with log x-axis to better visualize the results. With linear scale, it would be difficult to adjust the limits to show the performance for both low and high percentiles, since errors grow quite large and the axis would need to be squeezed. Furthermore, the limits of the plot have been adjusted between 0.2 m and 100 m to display the parts where differences between the tested methods are more meaningful. However, the maximum error of both RAIM-FDE and EKF-FDE far exceeds 100 m for both horizontal and vertical directions.

Since it can be hard to give a comprehensive view of the performance with a single plot, Figure 5.16 is complemented with Table 5.3, which reports the performance of the tested methods for some key values of percentiles of the 3D position error.

The results of the PF-ADP in terms of RMSE, mean error and CDF have been averaged across 20 runs in order to mitigate small fluctuations due to randomness in

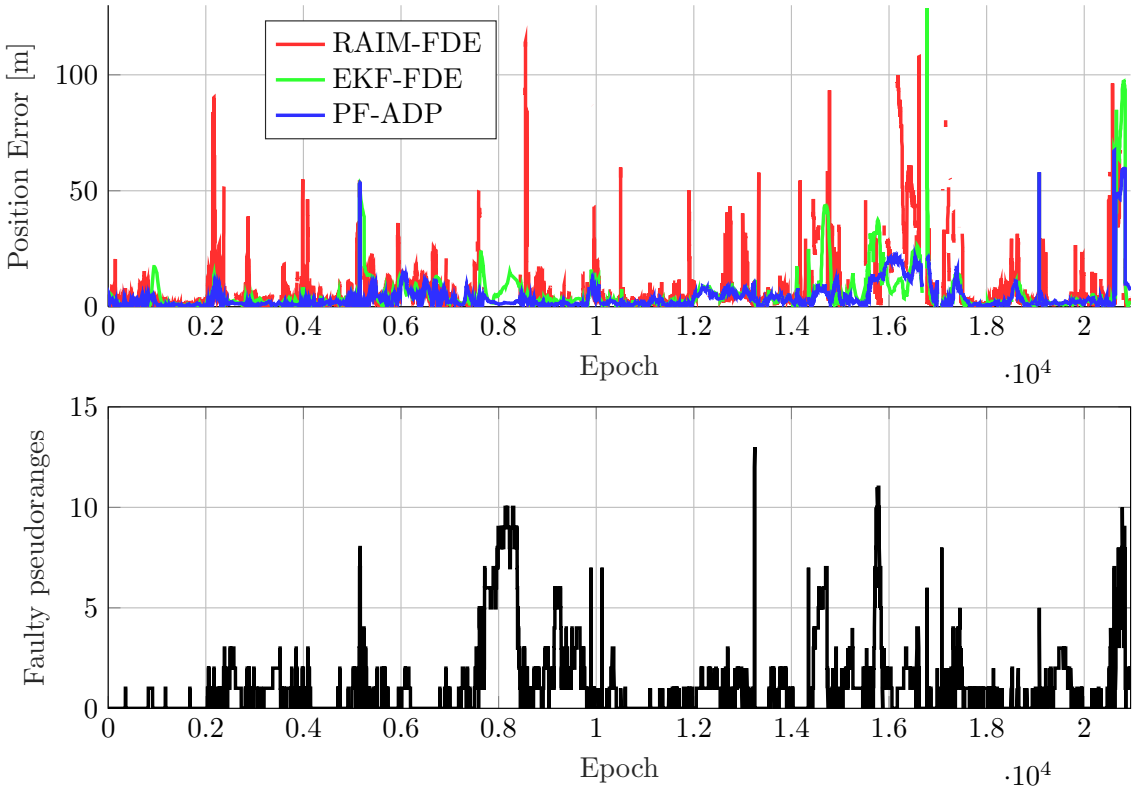
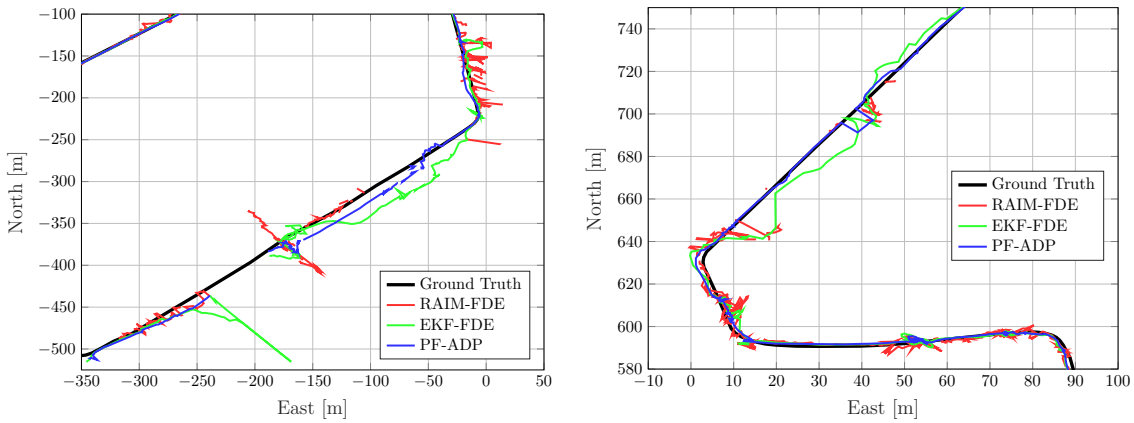


Fig. 5.14 Time-series of the 3D position error compared to the number of measurements labelled as faulty.

the solutions and thus provide a more fair comparison.
Results show that the proposed strategy is effective in mitigating the effect of biases due to multipath and NLoS w.r.t. other strategies from the literature.

Discussion and Future developments

In conclusion, the proposed PF architecture based on the concept of soft information has shown promising results on real data compared to other state-of-the-art solutions. Despite being an initial implementation of soft information ideas in the context of GNSS, the experiments have validated the effectiveness of such approach in order to mitigate not only the measurement biases, but also errors due to false alarms. The results thus motivate further investigations and implementation of such ideas in other applications or other stages of the receiver in the context of GNSS.
In particular, future works will focus firstly on improving the proposed method to further enhance the positioning capabilities of stand-alone GNSS in harsh environment. This includes, but is not limited to:



(a) Trajectory between epochs 5000 and 6850. (b) Trajectory between epochs 15000 and 17000.

Fig. 5.15 Comparison between positioning methods for two selected sectors of the trajectory.

- Devising and testing other adaptive strategies to combine different likelihoods adaptively. The goal is to find a strategy which most closely resembles the probabilistic knowledge on the likelihood distribution of faults.
- Improving the algorithm in order to adjust the threshold adaptively at each epoch.
- Refine the estimation of the variance of PF. Although this task is not directly related to the implementation of soft information, it is indeed crucial for its correct implementation.
- Testing the proposed method for other kinds of measurements (e.g., Doppler shifts).

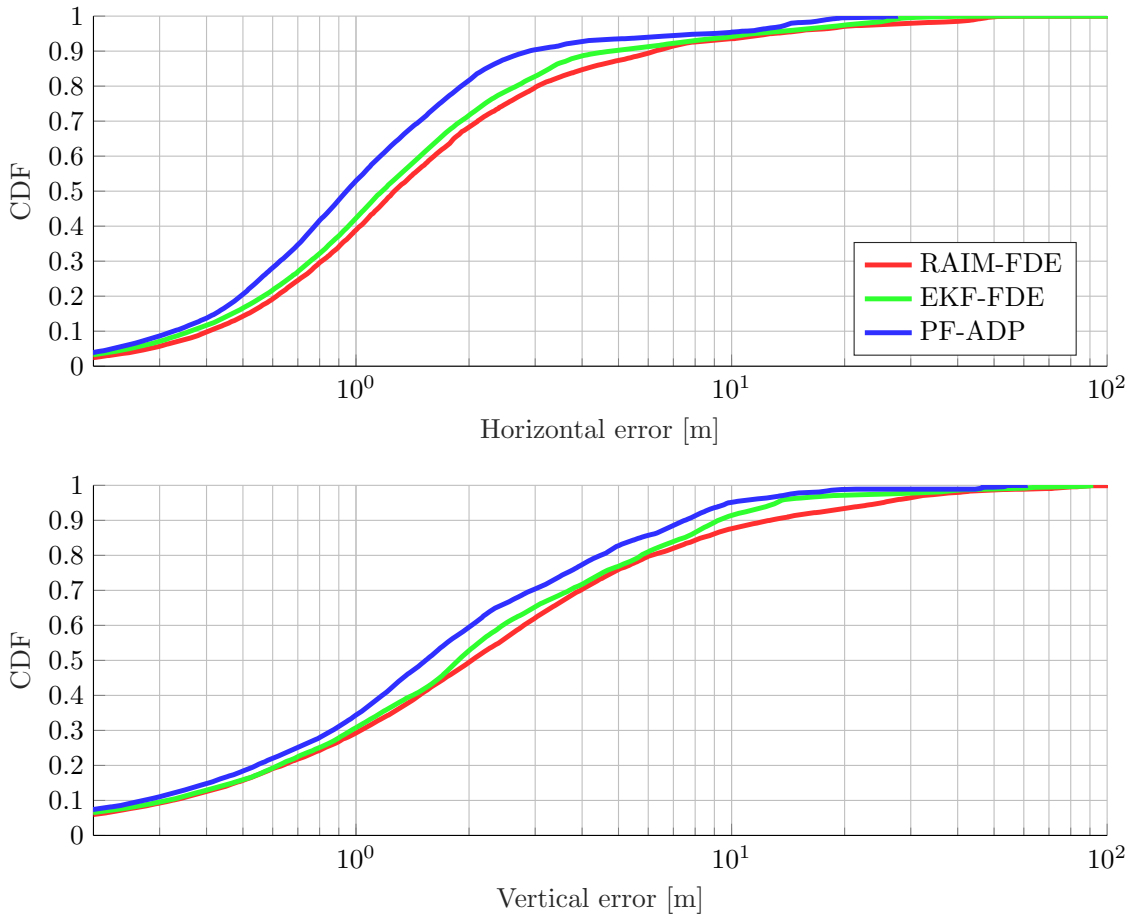


Fig. 5.16 Comparison of CDF of horizontal and vertical error in ENU frame for the tested methods.

Chapter 6

GNSS and UWB Tight Integration

6.1 Introduction

Autonomous vehicles and mobile robots highly rely on accurate and robust positioning solutions for the navigation and control loop [160]. In this framework, GNSS is essential due to its capability of providing positioning information in absolute reference frames [161]. However, autonomous applications are often employed in urban areas with very harsh environments, such as urban canyons and forests. Therefore, the performance of stand-alone GNSS may be severely degraded due to multipath and signal blockages. As a consequence, multi-sensor integration becomes a popular solution [162], to compensate for the drawbacks of each sensor under different scenarios.

UWB is a carrier-free telecommunication technology. Benefiting from narrow pulse and wide frequency band, UWB has centimetre-level ranging accuracy and strong penetration capabilities [163]. As a results, it is an ideal candidate to integrate with GNSS. The main limitation of UWB is the reduced range of transmission [164]. To overcome this, UWB transmitters can be deployed in some areas where GNSS signals deteriorate significantly [165]. Meanwhile, GNSS and UWB both receive radio signals, which enables the prospect of sharing the same antenna, thus simplifying the structure of the whole navigation system and overcome the spatial misalignment of measurements from different sensors.

Bayesian filters can estimate states for a time-varying system using a probability model with noisy measurements from different sensors [51]. Therefore, they can be leveraged for a sensor fusion strategy [166]. Nevertheless, non-linear GNSS and UWB measurement functions occur in the state-space model, especially for the latter.

To solve the problem of non-linear measurement functions, one common strategy is linearization, which consist in approximation of the non-linear function as a linear function at a particular point, as done in the EKF [71]. The other mainstream strategy is to formulate the propagation of non-Gaussian state distributions via non-linear systems using Monte Carlo methods, as done by the PF [79]. However, both filters have their own advantages and disadvantages. Given a specific application, the filter selection should be determined based on whether the accuracy gain is worth the extra computational load cost.
In the first section of this chapter, an original contribution published in [7] is presented In particular, a theoretical analysis is carried out to measure the error caused by the linearization in the EKF. Then, all the error sources in the EKF process of GNSS/UWB tight integration are analysed using the Taylor expansion. A comparison between the EKF and PF for the GNSS/UWB tight integration was implemented to verify the theoretical analysis. The experiment results give a reference for choosing filtering techniques under different positioning requirements and scenarios.
The second part of the chapter also includes an original contribution first presented in [8] and then extended in [9], which consists in a novel method for time synchronization in GNSS/UWB tight integration, based on a double-step update of the EKF. The method aims at tackling the issue of time offset between different sensors and leverages an enhanced state-transition model in order to minimize the error due to mismodelled dynamics of the receiver.

6.1.1 Literature Review

A lot of studies have been conducted to focus on the integration of GNSS and UWB. In particular, according to the deployment method of UWB anchors, studies can be categorized into two types.
The first group of approaches places the UWB anchors in fixed locations that can be located precisely [167–169]. Thus, these approaches can largely realize UWB's potential in high-accuracy ranging but in a designated area.
The second group of approaches put UWB anchors on a moving target, which is also known as one of the solutions in cooperative positioning [170–172]. Consequently, these methods greatly overcome the shortcoming of short-range transmission of UWB and sufficiently extend the flexibility of the integration system. However, determining accurate locations of UWB anchors becomes a challenge, which might induce extra errors in positioning results.
Another way to classify GNSS/UWB studies is based on the role UWB plays in the

integration. Some studies explored the potential of leveraging UWB to aid GNSS. In particular, [173] proposed a tight integration scheme to accelerate the convergence of Precise Point Position (PPP) using ranging information from UWB. [174] and [175] assisted RTK with UWB to enhance positioning accuracy. Meanwhile, other studies put UWB and GNSS on an equal footing to implement integration. [176] proposed an Adaptive Robust Extended Kalman filter (AREKF) to integrate all the information from RTK, UWB, INS, and map systems. [172] developed a robust cooperative positioning method based on GPS/UWB. [167] employs inequality constraints and innovation testing to mitigate the effects of unmodelled errors in GPS/UWB tight integration. [177] presents a novel Iterated Extended Set Membership Filter (IESMF) to implement relative localization. [178] proposed a low-cost INS/UWB sensor fusion architecture for pedestrian tracking, while, [179] introduced pedestrian motion characteristics to improve the consistency of positioning accuracy in GNSS/UWB tight integration systems.
Another main stream of studies focuses on solving engineering problems when implementing GNSS/UWB integration. [180] proposed a gross error elimination approach for UWB based on the variance of the squared Mahalanobis distance. [181] conducted some experiments to perform a statistical analysis of UWB measurements, which can be used to enhance the accuracy of their integration. [182] utilized a field calibration method to help obtain more accurate ranging data from UWB.

6.1.2 UWB Measurement Model

The ToA ranging equation of UWB is given as:

$$\begin{aligned} \rho_u &= \underbrace{\sqrt{(p_{x,u}-p_x)^2+(p_{y,u}-p_y)^2+(p_{y,u}-p_y)^2}}_{r_u} + \epsilon_u \\ \rho_u &= \|\boldsymbol{p}_u - \boldsymbol{p}\| + \epsilon_u = r_u \ + \ \epsilon_u. \end{aligned} \tag{6.1}$$

where:

- ρ_u is the range measurement between the receiver and the UWB anchor u.
- $\boldsymbol{p}_u = [\, p_{x,u} \ \ p_{y,u} \ \ p_{z,u} \,]$ is the vector of position coordinates of the UWB anchor u in Earth Centered Earth Fixed (ECEF) reference frame.
- ϵ_u is the sum of all errors in the range measurement for UWB anchor u.

6.2 Comparison Study of Bayesian Filters

In the context of tight integration between GNSS and UWB, the most common filters used are the EKF and PF. On one hand, the EKF is a popular choice because when its assumptions hold, it provides the optimal solution. However, in some scenarios these assumptions do not hold and the performance is degraded. On the other hand, the PF can relax some of the constraints posed by the EKF, but comes at the cost of increased computational load. The purpose of this section is to perform a comparison between the theoretical performance of the two filters in terms of error introduced by the linearization exploited by the EKF. In order to make the comparison fair, a standard tight integration architecture is implemented for both filters, without any modifications other than tuning of parameters. In fact, one of the goals of this study is to use the results and analysis in order to guide future work aimed at improving the implementations of the filter to address current limitations of the standard implementation.
After a theoretical analysis of their performance, both algorithms are tested to validate the analysis in an experimental settings, considering different scenarios in which the linearization error can be more or less relevant, depending on the dynamics of the receiver. The results will also include the execution times, so that a more comprehensive comparison between the two filters can be performed. The results thus provide sufficient information on the trade-off between estimation error and computation time between the filters in different conditions, which can help in the selection of which filter is more suitable depending on the target scenario and the available resources as well as guiding future research.

6.2.1 Analysis of Linear Approximation Errors

In the context of GNSS/UWB tight integration filtering problem, it is necessary to solve a system of non-linear equations where the non-linearities are more especially more pronounced for UWB measurements due to its limited range.
The EKF uses Taylor expansion to obtain a linear approximation of a non-linear function around a specific point. Therefore, the GLS is relatively high in computation efficiency but sacrifices some accuracy due to the approximation of non-linear components. On the other hand, PF utilizes the Monte Carlo approach to obtain a discrete approximation of the non-linear function, which can potentially improve the estimation accuracy but results in a higher computational load.
In the tight integration framework, the choice of using either one or the other filter

consists in determining a trade-off between the positioning performance and the computational load. That is to say, given a specific condition, the user can select the appropriate filter to satisfy its requirements in terms of accuracy and computational power.
To make an appropriate decision, it is worth investigating the factors impacting the accuracy of EKF and the specific degree of deterioration in different scenarios. Then, the scope of the study is to provide an evaluation of the performance gap between EKF and PF in terms of accuracy, under a set of different condition.
As just discussed, one of the most important reasons causing the accuracy deterioration in EKF is the approximation error due to linearization. Consequently, we can convert our analysis target from positioning accuracy to linearization accuracy. To this end, the Taylor expansion, is shown as:

$$\begin{aligned} \boldsymbol{y}_k = & h(\boldsymbol{x}_k^*) + \boldsymbol{J}_k(\boldsymbol{x}_k - \boldsymbol{x}_k^*) + \frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{x}_k^*)^T \boldsymbol{H}_k(\boldsymbol{x}_k - \boldsymbol{x}_k^*) \\ & + O((\boldsymbol{x}_k - \boldsymbol{x}_k^*)) + \boldsymbol{\epsilon}_k \end{aligned} \tag{6.2}$$

where:

- $\boldsymbol{J}_k$ is the Jacobian matrix of the measurement function h w.r.t. the vector of states $\boldsymbol{x}_k$;
- $\boldsymbol{H}_k$ is the Hessian matrix of the measurement function h w.r.t. the vector of states $\boldsymbol{x}_k$;
- $O((\boldsymbol{x}_k - \boldsymbol{x}_k^*))$ contains all the items in the Taylor expansions whose orders are higher than two.
- $\boldsymbol{\epsilon}_k$ is the vector of measurement noise at time k.

To approximate non-linear functions in linear form, EKF truncates the Taylor expansion after the first term. This means the linearization error is the sum of all the terms whose order is higher than one. Due to the presence of the factorial part, high-order terms are increasingly smaller in quantity w.r.t. the second-order term. Thus, to simplify the analysis, high-order terms can be neglected to instead focus our attention on the second-order term.
Hence, an analysis of the second-order term can be performed to reveal the error sources:

- $\boldsymbol{x}_k - \boldsymbol{x}_k^*$ is the difference between the true state and the predicted value, which can be called prediction error;

- $\boldsymbol{H}_k$ is the Hessian matrix, which depends on the second order partial derivatives of the measurement functions at the linearization point.

The prediction error $\boldsymbol{x}_k - \boldsymbol{x}_k^*$ depends on two factors, which are the dynamic motion of the user and the time interval of measurements. Higher dynamic motion (i.e., higher rate of change for velocity and position), means lower accuracy in the prediction step due to the unmodelled terms. Similarly, the longer the time interval of measurements, the lower the prediction accuracy will be since more time elapsed from the most recent fix. The second component is the Hessian matrix. For GNSS/UWB tight integration, there are two types of non-linear measurement functions. The first type is the pseudorange function, which is a sphere function since it is based on Euclidean distance. The second type is the UWB ranging function which is also a sphere function.
Applying a Taylor expansion at the linearization point and considering the first-order term, the linear approximated function is a plane tangent to the sphere at expansion point. Considering a neighbourhood of the linearization points, the smaller the radius of the sphere, the greater is the curvature of the sphere, and the greater the error in the linear approximation. In other words, as a sphere becomes smaller, approximating its surface with a plane becomes less and less accurate. This means that the elements in the Hessian that are being neglected become larger as the range grows smaller.
In the framework of GNSS/UWB integration, we can infer that the linearization of the GNSS pseudorange contains minor approximation errors due to the long distance from satellites to the receiver. On the other hand, considering UWB measurement functions, the approximation error is relatively large due to the short distance between UWB anchors and the receiver.

6.2.2 Experiments

In order to assess the impact of linearization and error due to unmodelled terms in the dynamic model of the receiver, a set of experiments is set up. The goal is to test different vehicle velocities and collect the performance of both EKF and PF in terms of accuracy of the position solution when performing GNSS/UWB tight integration.

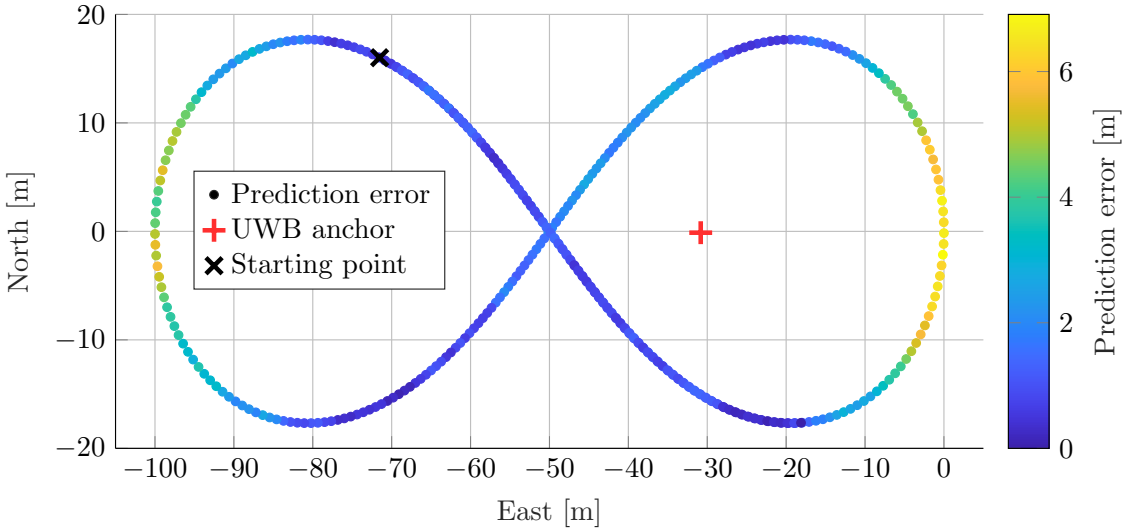


Fig. 6.1 Bernoulli lemniscate trajectory along with starting position, placement of UWB anchor and EKF prediction error at each epoch.

Parameter	Value	Physical meaning
$S_{v_x}, S_{v_y}, S_{v_z}$	$1\,\mathrm{m}^2/\mathrm{s}^3$	PSD of Velocity noise
S_t	$36\,\mathrm{s}$	PSD of Clock bias noise
S_f	$0.01\,\mathrm{s}^3$	PSD of Clock drift noise

Table 6.1 Parameter settings for process noise $\boldsymbol{Q}_k$ of 4.18.

Simulation Set-up

The GNSS measurements used for the experiments are simulated using a Radio Frequency GNSS simulator – IFEN Network Constellation Simulator (NCS) Titan. The corresponding ephemeris of satellites is also logged by the simulator. All the GNSS measurements (pseudorange and Doppler-shift) are from GPS L1 C/A signals and generated with a rate of 10 Hz.

Ionospheric disturbance models are used to simulate noise in the signals. In addition, the simulator provides the ground truth to compute the positioning error in the following analysis. Such ground truth of the trajectory follows a Bernoullian Lemniscate shape, which is shown in Figure 6.1. One UWB anchor is deployed. It is placed 20 m away from the centre of the trajectory, at a height of 5 m. Following an empirical model proposed in [183], the UWB dataset is generated using the ground truth and the position of the UWB anchor. UWB measurements are generated with a rate equal to that of GNSS measurements, namely 10 Hz.

The parameter setting of the process noise covariance matrix $\boldsymbol{Q}_k$ of 4.18 is given in

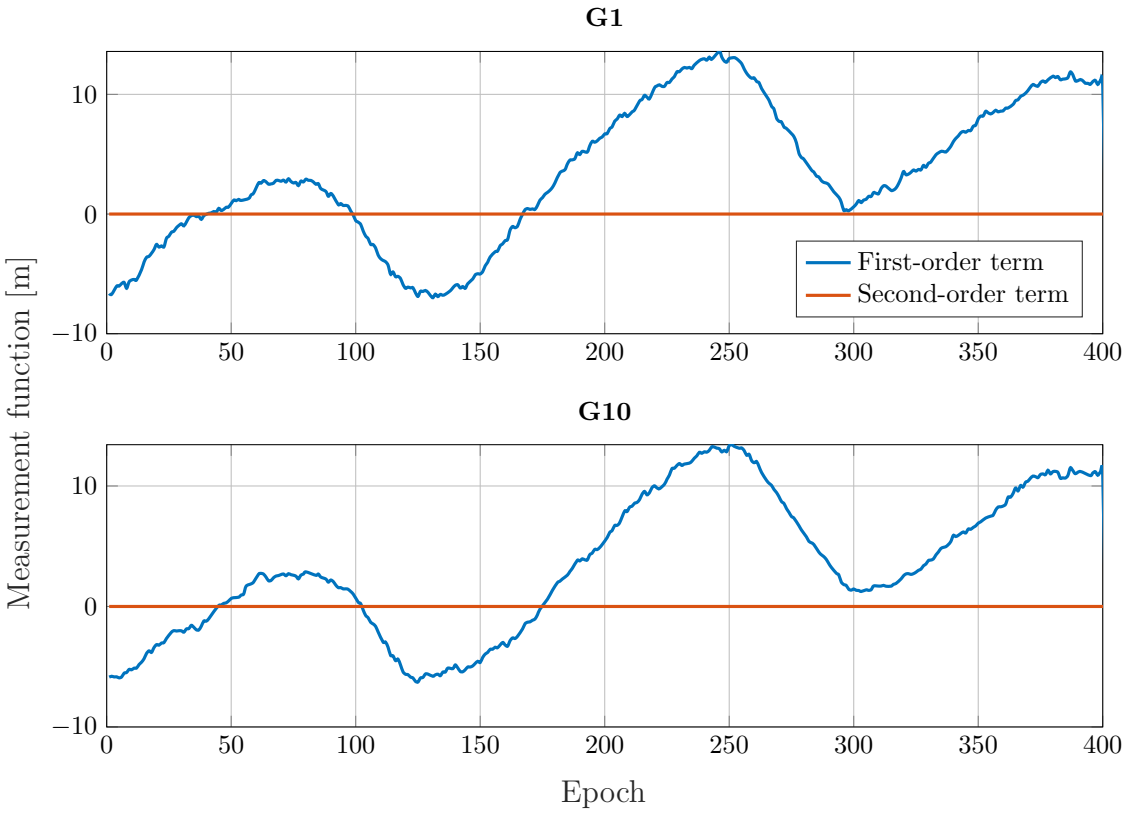


Fig. 6.2 Plot of first and second order terms of the Taylor expansion of the GNSS pseudorange measurement function across the trajectory.

Table 6.1. The measurement noise covariance matrix $\boldsymbol{C}_k$ is estimated at every epoch. The corresponding computation methods are taken from [12] for GNSS measurements and from [183] for UWB. Both filters are initialized using the state estimate and corresponding covariance matrix computed from an OLS solutions stand-alone GNSS. For the PF, the number of particles is set to 10000 while the threshold for resampling N_{th} is set to $\frac{1}{2}N$.

Results

First, we perform an experimental analysis of the first and second order terms of the Taylor expansion in (6.2). Figure 6.2 shows a time-series plot of the two terms for two different satellites. The first thing to notice is that the second order term depending on the Hessian is almost zero, since the radius of GNSS pseudorange is very large and therefore can be approximated reasonably well by linearization. The second interesting point is that the trend of the first-order term is very similar for the two measurement. This can be expected since the term is mainly driven by the

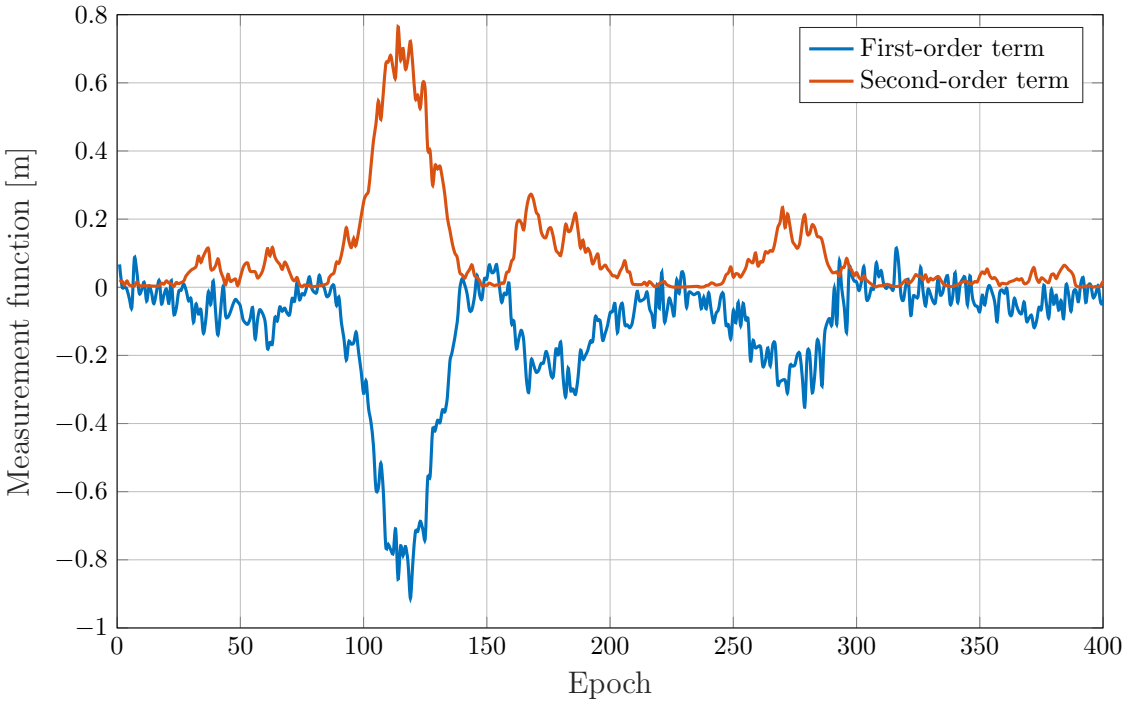


Fig. 6.3 Plot of first and second order terms of the Taylor expansion of the UWB measurement function across the trajectory.

prediction error, which is common for all satellites since is depends on the filter. On the other hand, Figure 6.3 shows the same plot for the UWB measurement w.r.t. the anchor. It is straightforward to notice that in this case, the second-order term is not negligible w.r.t. the first-order one. As a consequence, linearizing the UWB measurement function, and therefore truncating the second-order term, will result in approximation errors. The closer is the receiver to the anchor and the larger is its acceleration, the larger will be the approximation error in the EKF. Figure 6.4 puts together the main quantities of interest in the analysis. The top two plots show the time-series of the error in horizontal and vertical directions (in ENU reference frame), respectively. The third plot shows the prediction error due to linearization of the motion model, while the bottom plot show the real acceleration of the receiver. From this comparison it can be noticed how larger accelerations in parts of the trajectory lead to large prediction errors. In turn, a prediction error causes the UWB measurement function to be linearized at an incorrect point in the EKF, thus further increasing the position error. After an initial analysis of the linearization errors, we can now focus our attention more closely to the performance comparison of EKF and PF under different receiver dynamics. Given the above simulation conditions, two trajectories are simulated at different receiver velocities of 2 m/s and 10 m/s,

Velocity	Filter	RMSE [m] Horizontal	Vertical
$2\,\mathrm{m\,s^{-1}}$	EKF	0.6881	0.8384
	PF	0.5498	0.5030
$10\,\mathrm{m\,s^{-1}}$	EKF	1.0248	2.6060
	PF	0.7531	0.7848

Table 6.2 Comparison of position RMSE of EKF and PF under different vehicle velocities.

respectively.

The top two subplots of Figure 6.5 show, for both filter, the time-series of errors in the position in horizontal and vertical directions, respectively, when the receiver has a velocity of $2\,\mathrm{m\,s^{-1}}$. In such case, the performance of the filter is similar since the acceleration is relatively small during the curves of the trajectory and thus the prediction error is also small. On the other hand, the bottom two subplots Figure 6.5 show the same results in the case where the velocity is increased to $10\,\mathrm{m\,s^{-1}}$. In this scenario, the EKF is affected by errors in the linear approximation and the performance degrades considerably.

In order to quantify the loss in performance of both filters, and ECDF of 3D position errors is collected for both cases with the two velocities. The results are reported in Figure 6.6. As can be seen, when the velocity and thus the acceleration are larger, the performance of the EKF is largely affected in terms of accuracy. On the other hand, while the performance of PF is also affected, the impact due to the unmodelled term of the dynamic are not further amplified by linearization errors and thus the performance degrades by a lesser extent. The results in terms of RMSE for all the combinations of filter and receiver velocities are reported in Table 6.2.

To complete the analysis, the execution time of the two filters is provided in Table

Filter	Execution time [s]
EKF	1.6907
PF	7.4827

Table 6.3 Execution time for the EKF and PF considering 1600 epochs and 10000 particles.

6.3. As can be seen, the PF takes considerably longer to run as expected, although the number of particles used is relatively large and could be reduced without significant loss of performance. However, it should be pointed out that even the PF is able to compute more than one second of PVT solutions in less than one second of execution time, meaning that it is suitable for real-time applications.

In conclusion, experimental results demonstrate that EKF produces large position errors in high-dynamic scenarios and when the user is closer to the UWB anchors. On the contrary, the PF performs much better under the above conditions since it is not affected by linearization errors. Therefore, the results of this paper provide a reference to select a suitable integration scheme for GNSS/UWB integration depending on the scenario of interest.

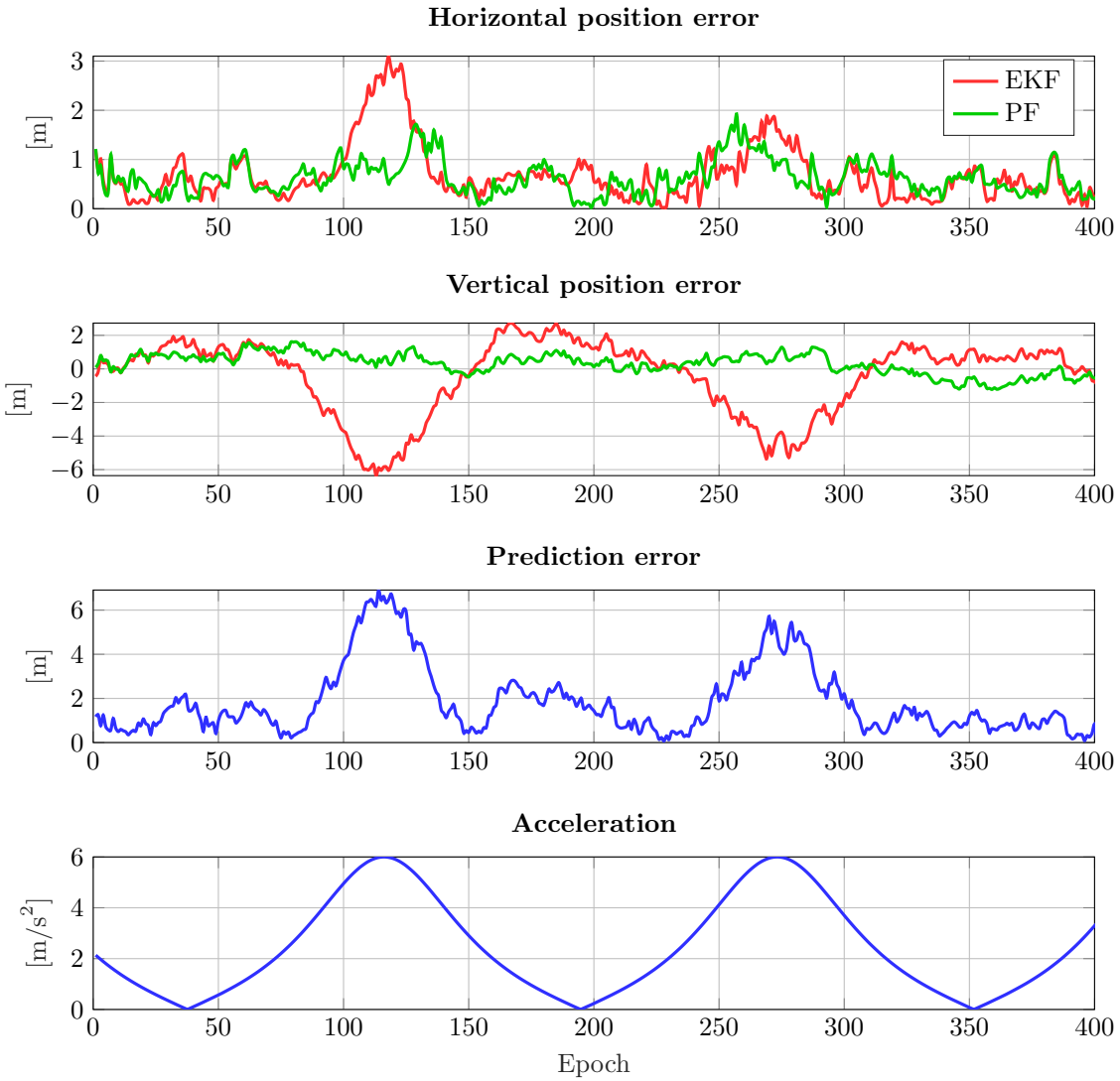


Fig. 6.4 Time series plot of a portion of the trajectory. In order from the top, horizontal and vertical position error for both filters (with velocity of $10\,\mathrm{m\,s^{-1}}$), prediction error of dynamic model and the real acceleration of the receiver.

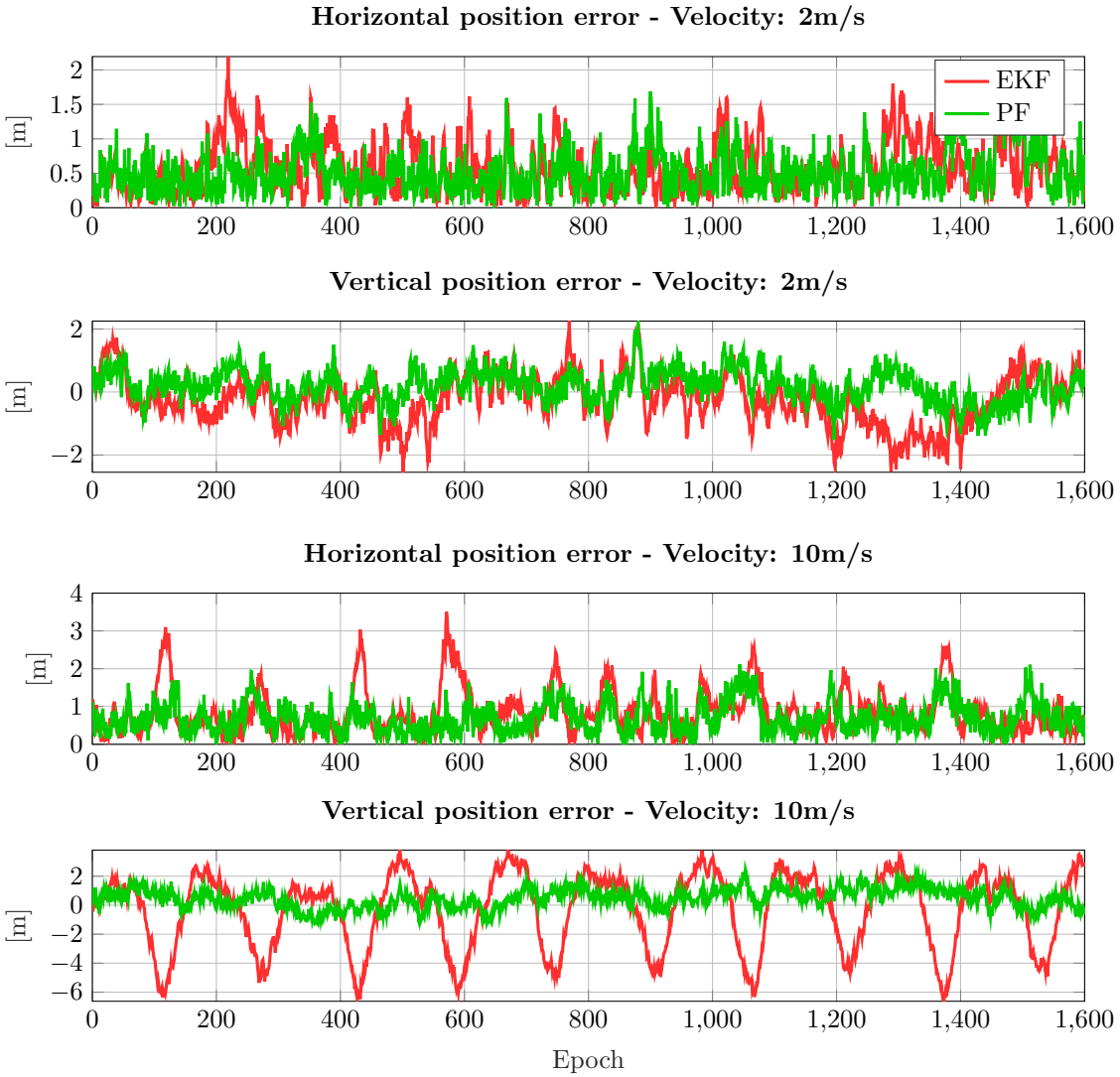


Fig. 6.5 Time series plot showing a comparison between EKF and PF of horizontal and vertical errors for different vehicles velocities.

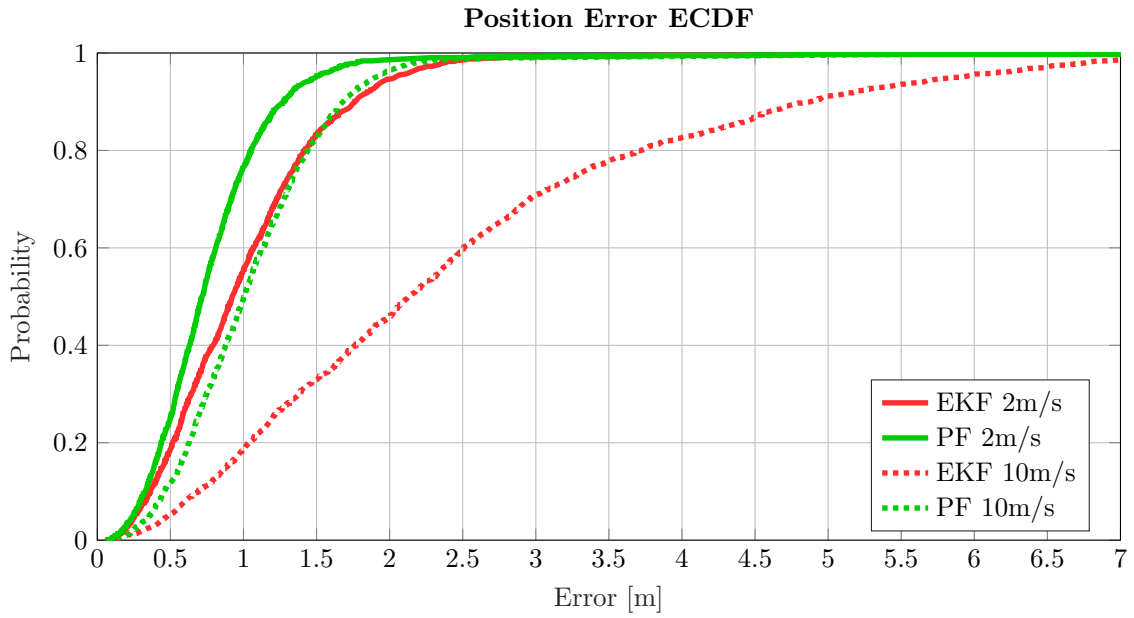


Fig. 6.6 Empirical Cumulative Density Function (ECDF) of position errors of both EKF and PF for different receiver velocities.

6.3 EKF-based Time Calibration

An implicit assumption that was made in the previous section was to assume accurately timestamped and synchronised measurements from GNSS and UWB. However, in the framework of low-cost tight integration, commercial UWB transceivers and mass-market GNSS chipsets operate using independent and asynchronous clocks. As a consequence, GNSS and UWB measurements are timestamped w.r.t. different time-scales, and timing disturbances such as latency jitters or spurious clock drifts can affect the timestamping precision w.r.t. the true sampling instant.
Moreover, there is a lag between the timestamp of measurements and the time instant at which the navigation filter processes those measurements. In particular, this lag is different for GNSS and UWB measurements since sampling rates are different and clocks are independent. Consequently, an unknown time offset exists between the timestamps of GNSS observables and UWB measurements. This time offset models the relative misalignment between the two time-scales as well as the shift of these scales w.r.t. the navigation filter time-scale [184]. If neglected, the time offset can inject a bias into the navigation algorithm, which may jeopardise the accuracy of the state estimation [185, 186].
As a consequence, this section introduces a novel time calibration method to guarantee a consistent GNSS/UWB tight integration, thus fully exploiting the benefits introduced by UWB, namely superior ranging accuracy over short distances.
In the literature regarding multi-sensor systems, different methodologies have been explored to handle synchronisation among self-contained sensor units. On one hand, hardware-level synchronisation is especially popular in the framework of GNSS-aided strap-down INS systems [187, 188]. Hardware-level synchronization typically exploits the Pulse Per Second (PPS) signal from the GNSS receiver as a triggering reference, and it cross-references the timing signals from the coupled sensors in order to establish a shared event base in the navigation unit [189, 190].
However, Commercial Off-The-Shelf (COTS) sensors typically feature limited access to the hardware, so software-based strategies can be the preferred choice. In particular, software-based solutions operate a time calibration process involving accurate time-offset estimation in the filter architecture by leveraging the measurement models of sensors [191, 192]. Consequently, such solutions can vary significantly depending on the sensors which are considered and the associated integration models.
In the existing literature, most of the proposed techniques address time calibration as a registration task, where time offsets are estimated through an offline, pre-processing step. For example, [193–195] propose a continuous-time batch formulation of the

time calibration problem, fitting with the framework of MLE. Moreover, works like [196, 197] perform visual-inertial time calibration by temporally aligning orientation curves sensed by the independent sensors. Joint optimisation-based calibration strategies for multi-sensors systems are instead explored in [198, 199]. As opposed to offline techniques, online temporal calibration via filtering-based methods is a promising alternative [200, 201]; this strategy models and recursively estimates the time offset as an additional unknown under the filter state-space formulation [202]. Following this approach, an EKF-based filtering model supporting time offset calibration is established here for GNSS/UWB tight integration. Nonetheless, the naive EKF-based time calibration model suffers pitfalls that can compromise integration performance under certain kinematic conditions [201]. In fact, the local identifiability [203] of the time offset (i.e., the capacity of accurately observing and estimating the unknown time offset between input measurements) can be undermined by the relative geometry between the mobile receiver and the UWB anchors, thus leading to an ineffective time calibration.
Therefore, a novel, double-update EKF architecture with an optimized weighting of UWB covariance statistics is presented, in order to mitigate the impact of the relative geometry of UWB anchors on accurate time offset estimation. In addition, the effectiveness of the proposed time calibration methodology for GNSS/UWB tight integration is validated with a real-world assessment for a vehicular trajectory in a suburban area.

6.3.1 Tight Integration Architecture

According to the analysis provided in 6.2, we have seen how unmodelled components of the receiver dynamics can cause prediction errors in the EKF, which in turn cause approximation errors due to linearization of the UWB measurement function.
In order to limit the impact of unmodelled dynamics, the state vector is augmented by adding the acceleration of the receiver. This will allow creating a more complete state-transition functions which accounts for higher-order terms of the dynamics equations. The new state vector at a generic epoch k can be defined:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{p}_k & c_{b,k} & \boldsymbol{v}_k & c_{d,k} & \boldsymbol{a}_k \end{bmatrix} \tag{6.3}$$

where $\boldsymbol{a}_k = [a_{x,k}, a_{y,k}, a_{z,k}]$ is the receiver acceleration vector expressed in Cartesian coordinates. As a consequence, the state transition matrix from (4.17) can be

updated accordingly to account for the extra states as:

$$\boldsymbol{F}_k = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3}\Delta t & \boldsymbol{0}_{3\times1} & 0.5\boldsymbol{I}_{3\times3}\Delta t^2 \\ \boldsymbol{0}_{1\times3} & 1 & \boldsymbol{0}_{1\times3} & \Delta t & \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3}\Delta t \\ \boldsymbol{0}_{1\times3} & 0 & \boldsymbol{0}_{1\times3} & 1 & \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3} \end{bmatrix}. \tag{6.4}$$

The joint measurement vector for GNSS/UWB tight integration embeds raw GNSS pseudorange and Doppler-shift observables as well as UWB auxiliary ranging information. At epoch k, it can be written as:

$$\boldsymbol{y}_k = \begin{bmatrix} \boldsymbol{\rho_k} & \boldsymbol{\dot{\rho}_k} & \boldsymbol{\rho_{u,k}} \end{bmatrix}^T \tag{6.5}$$

where $\boldsymbol{\rho_k}$ is the vector of n pseudoranges, $\boldsymbol{\dot{\rho}_k}$ that of Doppler shift or range rates and $\boldsymbol{\rho_{u,k}}$ a vector of m auxiliary UWB range measurements as described in (6.1). Equivalently, the linear observation matrix from (4.19) can also be updated as:

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{H}_{n\times4} & \boldsymbol{0}_{n\times4} & \boldsymbol{0}_{n\times3} \\ \boldsymbol{0}_{n\times4} & \boldsymbol{H}_{n\times4} & \boldsymbol{0}_{n\times3} \\ \boldsymbol{H}^U_{m\times4} & \boldsymbol{0}_{m\times4} & \boldsymbol{0}_{m\times3} \end{bmatrix} \tag{6.6}$$

where $\boldsymbol{H}_{n\times4}$ is the GNSS measurement matrix as obtained in (3.8), while $\boldsymbol{H}^U_{m\times4}$ is the Jacobian matrix obtained from linearizing the UWB measurement functions in (6.1). The fourth column of $\boldsymbol{H}^U_{m\times4}$ is made of all zeros since the UWB measurement does not depend on the clock bias c_b.
The last three columns of (6.6) are made of zeros, since none of the measurements depends on the acceleration, for which a rough estimation can instead be obtained as the time derivative of the velocity estimate.

6.3.2 Time Offset

After establishing a mathematical formulation for GNSS/UWB time-offset, this section focuses on a theoretical analysis to investigate its impact on EKF-based state estimation. Given independent rates and time-scales between the GNSS receiver clock and the UWB transceiver clock, an unknown time offset exists between the timestamps of raw GNSS observables and the timestamps of auxiliary UWB ranging measurements. At a generic epoch k, the tight integration filter combines GNSS and

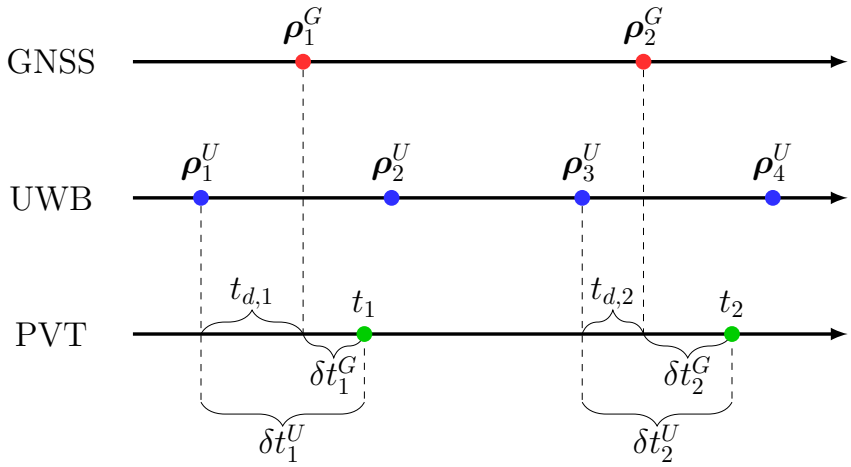


Fig. 6.7 Diagram of the time-offset between GNSS and UWB measurements involved in a tight integration scheme. For readability, GNSS and UWB time axis are assumed aligned to the integration time axis (bottom).

UWB measurements in the observation model at a time instant t_k, which is tagged to the integration time-scale. Then, two quantities are identified:

- δt_k^G which expresses the lag between t_k and the most recent available set of GNSS measurements.
- δt_k^U which expresses the lag between t_k and the most recent available set of auxiliary UWB measurements.

Then, the GNSS/UWB time-offset is defined as:

$$t_{d,k} = \delta t_k^U - \delta t_k^G \tag{6.7}$$

which expresses the misalignment of the timestamp associated to the set of auxiliary UWB measurements w.r.t. the GNSS time-scale. The time offset $t_{d,k}$ includes the shift between the timestamping times of GNSS and UWB measurements, and it can can be either a positive or a negative quantity at any epoch, depending on when the filter processes the measurements.
As a matter of fact, $t_{d,k}$ accounts for two time-varying effects:

- The relative misalignment between the GNSS and UWB measurements because of the different sampling rates and clocks of the independent sensors.
- The lag of GNSS and UWB time-scales w.r.t. the time-scale of the centralized processing unit running the tight integration algorithm.

While the first effect embeds non-idealities of individual sensor clocks (e.g., clock drift and latency jitters), the second effect is contributed by multiple sources such as data-transfer latencies, hardware-level processing and software overhead in the centralised processing unit. Furthermore, under the assumption of high UWB sampling rate and small drift of sensor clocks, $t_{d,k}$ can be modelled as a constant between consecutive estimation epochs. Hence, the a-posteriori estimate of the time offset at a given epoch can be used as the prediction of the offset at the next epoch. A graphical interpretation of the described framework is provided in Figure 6.7. The three axes refer to a common time-scale (i.e., the integration time-scale). The top and middle axes show the instants at which GNSS and UWB measurements are dumped, respectively. The bottom axis shows when the integration filter is processing such measurements. In case the measurements were provided at high rate from at least one of the sensors, and the integration took place at a time t_k at which the low-rate measurements are available, this time-offset would be negligible. However, in this work, the aim is to directly estimate $t_{d,k}$ as part of the system state, hence proposing a low-complexity strategy which relaxes constraints on measurements rates or related assumptions.

6.3.3 Mathematical Analysis

In light of the analysis of the time offset, it can be seen how the joint GNSS/UWB measurement vector (6.5) collects observables which are not temporally consistent with each other. In fact, at time t_k, the set of available GNSS ranging observables carry information about the position of the receiver $\boldsymbol{p}^G$ at a time instant which lags t_k by δt_k^G. Conversely, the available set of auxiliary UWB ranges bring information about the position $\boldsymbol{p}^U$ at a time instant which lags t_k by δt_k^U. Therefore, the two sets of measurements are not aligned in time and bring information about the same quantities at different time instants. Focusing out attention on the auxiliary range from the u-th UWB anchor, (6.1) can be reframed as:

$$\rho_u = \sqrt{(p_{u,x} - p_x^U)^2 + (p_{u,y} - p_y^U)^2 + (p_{u,z} - p_z^U)^2} \tag{6.8}$$

For ease of analysis, the residual error term, ϵ_u, is neglected hereafter. Time indexes on the state variables are also dropped for the next steps for the sake of readability. Then, at a time t_k, leveraging a continuous-time motion model for state dynamics,

$\boldsymbol{p}^U$ can be related to $\boldsymbol{p}^G$ as follows:

$$\boldsymbol{p}^G = \begin{bmatrix} p_x^G \\ p_y^G \\ p_z^G \end{bmatrix} = \begin{bmatrix} p_x^G + \int_{t_k+\delta t_{U,k}}^{t_k+\delta t_{U,k}-t_{d,k}} v_x(t)\,dt \\ p_y^G + \int_{t_k+\delta t_{U,k}}^{t_k+\delta t_{U,k}-t_{d,k}} v_y(t)\,dt \\ p_z^G + \int_{t_k+\delta t_{U,k}}^{t_k+\delta t_{U,k}-t_{d,k}} v_z(t)\,dt \end{bmatrix} = \begin{bmatrix} p_x^U \\ p_y^U \\ p_z^U \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix} = \boldsymbol{p}^U + \boldsymbol{\varepsilon}^{U,G} \tag{6.9}$$

where $\boldsymbol{\varepsilon}^{U,G}$ identifies the displacement vector along the trajectory of the receiver induced by the time offset $t_{d,k}$. In other words, $\boldsymbol{\varepsilon}^{U,G}$ represents the space travelled by the receiver between the time instants in which GNSS and UWB measurements are taken. In particular, it can be obtained as the integral, between these two time instants, of the continuous time velocity $\boldsymbol{v}(t)$ of the receiver.
Given (6.9), the UWB range w.r.t. the u-th anchor can be time-aligned to the available GNSS ranging observables by substituting $\boldsymbol{p^U}$ in (6.8) with $\boldsymbol{p}^U + \boldsymbol{\varepsilon}^{U,G}$:

$$\tilde{\rho}_u = \sqrt{(p_{u,x} - (p_x^U + \varepsilon_x^{U,G}))^2 + (p_{u,y} - (p_y^U + \varepsilon_y^{U,G}))^2 + (p_{u,z} - (p_z^U + \varepsilon_z^{U,G}))^2} \tag{6.10}$$

Hence, it is possible to quantify a UWB ranging error term caused by $t_{d,k}$:

$$\epsilon_u^{U,G} = \tilde{\rho}_u - \rho_u = \frac{(\tilde{\rho}_u)^2 - (\rho_u)^2}{\tilde{\rho}_u + \rho_u} \tag{6.11}$$

which is the difference between the time-aligned measurement and the original one. Then, $\tilde{\rho}_u$ and ρ_u at the numerator can be substituted using (6.10) and (6.8), thus obtaining:

$$\epsilon_u^{U,G} = \frac{\|\boldsymbol{\varepsilon}^{U,G}\|^2 - 2\,(\boldsymbol{p}_u - \boldsymbol{p}^U) \cdot \boldsymbol{\varepsilon}^{U,G}}{\tilde{\rho}_u + \rho_u} \tag{6.12}$$

which shows that, except for a positive scaling factor at the denominator, $\epsilon_u^{U,G}$ depends on two terms:

- The squared length of the position displacement (i.e., the squared norm of $\boldsymbol{\varepsilon}^{U,G}$) induced by $t_{d,k}$;
- A term proportional to the length of the projection of $\boldsymbol{\varepsilon}^{U,G}$ onto the Line-of-Sight (LoS) vector from the receiver position $\boldsymbol{p}^U$ to the u-th UWB anchor position $\boldsymbol{p}_u$.

By manipulating (6.11), $\epsilon_u^{U,G}$ of each of the m UWB anchors can be seen as an additive term to the set of auxiliary UWB range measurements.
Now we can go back to consider what happens at a given epoch k, it is possible to

rewrite (6.5) as:

$$\tilde{\boldsymbol{y}}_k = \begin{bmatrix} \boldsymbol{\rho_k} \\ \boldsymbol{\dot{\rho}_k} \\ \boldsymbol{\rho_{u,k}} \end{bmatrix}^T - \begin{bmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} \\ \boldsymbol{\epsilon}_k^{U,G} \end{bmatrix} = \boldsymbol{y}_k - \boldsymbol{\epsilon}_k^{U,G} \tag{6.13}$$

where $\tilde{\boldsymbol{y}}_k$ collects time-aligned GNSS and UWB ranging observables.
This above analysis showed how $t_{d,k}$ introduces an additive error factor $\boldsymbol{\epsilon}_k^{U,G}$ in the GNSS/UWB observation model. This ranging error then propagates on the a-posteriori state estimation error obtained by the tight integration filter, which we can quantify. First, recall the EKF innovation vector from (4.12). By using (6.13), the EKF updated state at epoch k is obtained as:

$$\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^* + \boldsymbol{K}_k \left(\underbrace{\boldsymbol{y}_k - \boldsymbol{\epsilon}_k^{U,G}}_{\tilde{\boldsymbol{y}}_k} - \mathbf{h}\left(\hat{\boldsymbol{x}}_k^*\right) \right). \tag{6.14}$$

where the term within round brackets is the innovation. Because of linearity, the a-posteriori state estimation error due to the time offset $t_{d,k}$ can be obtained as:

$$\boldsymbol{\epsilon}_k^{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_k - \hat{\boldsymbol{x}}_k = \boldsymbol{K}_k \, \boldsymbol{\epsilon}_k^{U,G} \tag{6.15}$$

where $\tilde{\boldsymbol{x}}_k$ and $\hat{\boldsymbol{x}}_k$ are the a-posteriori state-estimates obtained using $\tilde{\boldsymbol{y}}_k$ and $\boldsymbol{y}_k$, respectively. By looking closely at (6.15), it can be noticed that both the Kalman gain $\boldsymbol{K}_k$ and the ranging error term $\boldsymbol{\epsilon}_k^{U,G}$ contribute to introduce errors in the navigation solution. In particular, $\boldsymbol{K}_k$ amplifies the UWB ranging error propagation on the a-posteriori state estimation. In fact, when $\boldsymbol{K}_k$ converges to an all-zeros matrix (i.e., a condition signalling that the EKF is trusting the state-prediction more than the observations) the impact of $\boldsymbol{\epsilon}_k^{U,G}$ is largely mitigated.
On the contrary, the state estimation performance degradation due to $t_{d,k}$ is considerable when the integration filter puts very high confidence on the set of observables, as can be expected. Moreover, according to (6.12), the components of $\boldsymbol{\epsilon}_k^{U,G}$ depend upon $\boldsymbol{\varepsilon}_k^{U,G}$, which in turn are functions of the velocity $\boldsymbol{v}(t)$ as established by (6.9). Therefore, when the components of $\boldsymbol{v}(t)$ take small values, the state estimation error contributed by $t_{d,k}$ reduces accordingly. On the contrary, $\boldsymbol{\epsilon}_k^{\boldsymbol{x}}$ is expected to grow in conditions of high dynamics.

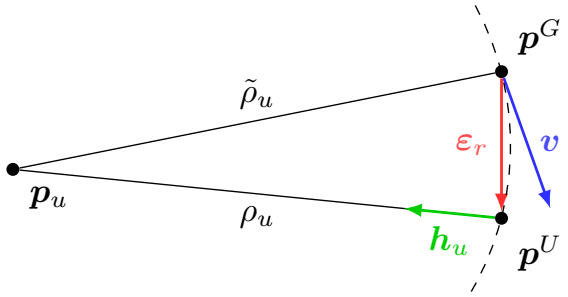


Fig. 6.8 Example of a scenario characterised by small inner product between the receiver velocity $\boldsymbol{v}$ and the steering vector $\boldsymbol{h}_u$ to the j-th UWB anchor.

6.3.4 Improved Model for Time Offset Estimation

From the analysis just presented, it is shown how $t_{d,k}$ can affect the state-estimation performance as the tight integration filter processes observables which are not time-consistent. Hence, the baseline GNSS/UWB tight integration model needs to be improved in order to enable time calibration via EKF architecture. This section provides an updated model which accounts for the time offset between measurements. This forms the basis for a naive calibration which will also be part of the comparisons of the results section. For the sake of simplicity, the time index is dropped for the remainder of this discussion. An extended state-vector is defined:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} & c_b & \boldsymbol{v} & c_d & \boldsymbol{a} & t_d \end{bmatrix} \tag{6.16}$$

where, compared to (6.3), the time-offset t_d is introduced as an additional parameter to be estimate. Based on (6.16), the state-transition matrix can be modified accordingly as:

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3}\Delta t & \boldsymbol{0}_{3\times1} & 0.5\boldsymbol{I}_{3\times3}\Delta t^2 & \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 & \boldsymbol{0}_{1\times3} & \Delta t & \boldsymbol{0}_{1\times3} & 0 \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3}\Delta t & \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 0 & \boldsymbol{0}_{1\times3} & 1 & \boldsymbol{0}_{1\times3} & 0 \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & 0 & 0 & 1 \end{bmatrix} \tag{6.17}$$

The new state transition matrix in (6.17) can be used to obtain a discrete approximation of the integral involved in the computation of $\boldsymbol{\varepsilon}^{U,G}$ based on (6.9).

Then, (6.8) can be rewritten as:

$$\begin{aligned}\rho_u = &\sqrt{(p_{u,x} - (p_x^U - v_x t_d - 0.5\, a_x\, t_d^2))^2} \\ &\overline{+(p_{u,y} - (p_y^U - v_y\, t_d - 0.5\, a_y\, t_d^2))^2} \\ &\overline{+(p_{u,z} - (p_z^U - v_z\, t_d - 0.5\, a_z\, t_d^2))^2}\end{aligned} \tag{6.18}$$

which is an improved UWB measurement model embedding the time offset t_d. Examining (6.18), the information regarding the receiver position brought by auxiliary UWB ranges (i.e., $\boldsymbol{p}^U$) gets compensated for the displacement induced by t_d in order to geometrically match it with the receiver position corresponding to the raw GNSS observables (i.e., $\boldsymbol{p}^G$).
Moreover, the proposed modelling takes $\boldsymbol{p}^G$ as the unknown receiver position in the GNSS/UWB state-space formulation. Therefore, $\boldsymbol{p}^G$ matches with $\boldsymbol{p}$ in (6.16). Furthermore, based on (6.18), modifications are required in the Jacobian matrix for UWB ranging $\boldsymbol{H}^U_{m\times 3}$, assuming U UWB anchors [163].

Double-Update Filtering

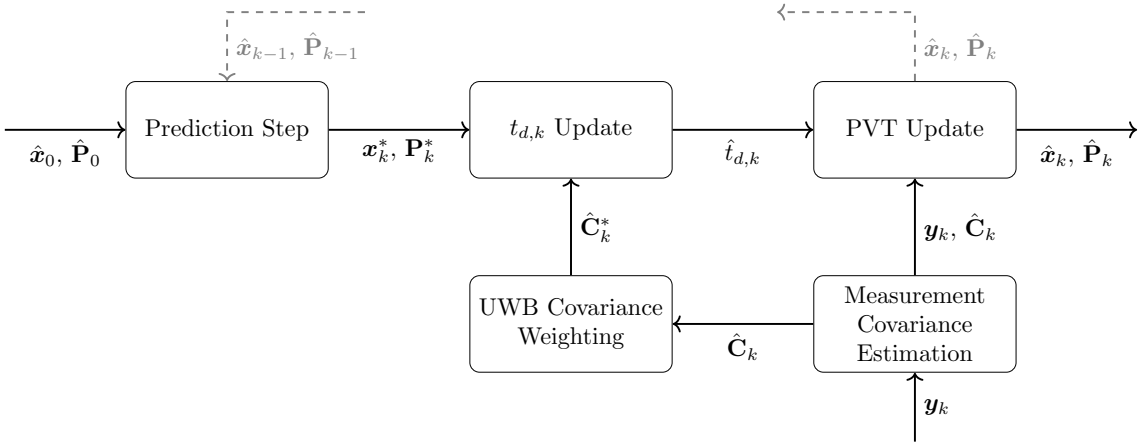


Fig. 6.9 Processing stages of the double-update EKF architecture with adaptive weighting of UWB covariance statistics for the enhanced calibration of $t_{d,k}$. The input measurement vector $\boldsymbol{y}_k$ is highlighted.

Comparing (6.18) to (6.8), the receiver position used for ρ_u is moved along the dynamic model using a specific value of t_d, which needs to be estimated.
Hence, the integration filter can exploit the discrete-time characterisation to first detect the unknown t_d at time t_k based on the difference between $\tilde{\rho}_u$ and ρ_u, namely $\varepsilon_u^{U,G}$ as in (6.11).

Thus, the model needs to be improved further in order to properly estimate the time offset. For the u-th UWB anchor, a function of the unknown state t_d can be introduced:

$$f_u\left(t_d\right) = \tilde{\rho}_u^2 - \rho_u^2 = (\tilde{\rho}_u - \rho_u)(\tilde{\rho}_u + \rho_u) \tag{6.19}$$

which, by definition, is proportional to the difference between $\tilde{\rho}_u$ and ρ_u, other than a positive amplifying factor $\tilde{\rho}_u + \rho_u$. After a rough calibration of UWB transceiver clock [204], it can be assumed that t_d takes values of few tens of milliseconds. Then, (6.19) can be expanded by substitution of (6.18) and (6.10) allowing a first-order approximation:

$$f_u\left(t_d\right) = 2\, t_d\, (\boldsymbol{p}_u - \boldsymbol{p}^G) \cdot \boldsymbol{v} \tag{6.20}$$

Most importantly, $f_u\left(t_d\right)$ is a function that can approximate the quality of time offset estimation. In fact, the sharper is the envelope of $f_u\left(t_d\right)$, the more accurately and uniquely t_d can be inferred from the set of available measurements. Conversely, in case the first-order derivative of $f_u\left(t_d\right)$ approached a null value for some interval in the support, then the estimation of the time offset would be degraded.
By differentiating (6.20), we can obtain:

$$\frac{\partial\, f_u\left(t_d\right)}{\partial\, t_d} \propto \underbrace{(\boldsymbol{p}_u - \boldsymbol{p}^G)}_{\boldsymbol{h}_u} \cdot \boldsymbol{v} \tag{6.21}$$

which highlights the following: the smaller is the inner product between the velocity of the receiver $\boldsymbol{v}$ and the steering vector $\boldsymbol{h}_u$ pointing to the location of the u-th UWB anchor, the weaker is the identifiability of t_d. This means that if the receiver moves perpendicularly w.r.t. the steering vector, then the difference between $\tilde{\rho}_u$ and ρ_u is zero and it is not possible to infer t_d from that measurement. An analogy with Doppler effect can be made. If a source is travelling sideways w.r.t. to the receiver, the observed Doppler is always zero, regardless of the speed of the source. In such case it is not possible to infer the velocity of the source from the observed Doppler shift. To generalize this concept, the more the receiver moves in a direction parallel to the steering vector w.r.t. the UWB anchor, the easier it is to accurately identify the time offset. An example of this geometrical relationship is given in Figure 6.8.
To cope with the aforementioned phenomenon, the covariance statistics of auxiliary UWB ranging measurements are adaptively weighted in the observation model. This is done in order to give less priority to UWB measurements with a poor geometry for the estimation of the time offset. Formally, accounting for the EKF measurement noise covariance matrix $\boldsymbol{C}$ [205], the estimated variance of the auxiliary UWB range associated to the u-th UWB anchor is amplified through the following

epoch-dependent coefficient:

$$A_u = 1 + \sqrt{1 - \left(\frac{\boldsymbol{h}_u \cdot \boldsymbol{v}}{\|\boldsymbol{h}_u\| \, \|\boldsymbol{v}\|} \right)^2}. \tag{6.22}$$

Focusing on the ranging contribution from u-th UWB anchor, (6.22) aims at weighting the degree of trust the integration filter should put on ρ_u to correct the a-priori prediction of t_d. The value of the coefficient is chosen according to the identifiability conditions due to the geometry behaviour of (6.21). In particular, the smaller the sine of the angle between $\boldsymbol{h}_u$ and $\boldsymbol{v}$ (i.e., the stronger the local identifiability of t_d), the smaller the value of the weighting coefficient A_u. Hence, the corresponding covariance term modelling the error of ρ_u would be smaller, in order to enhance the contribution of such auxiliary range to the update of t_d. Conversely, the larger the sine of the angle between $\boldsymbol{h}_u$ and $\boldsymbol{v}$, the larger the value of A_u. In such a case, ρ_u is poorly informative to perform the update of t_d and its variance should be increased. Nevertheless, according to time calibration model, the adaptive weighting of UWB covariance statistics is expected to affect the a-posteriori correction of all state variables in (6.16), not just t_d. It follows that the discussed weighting strategy must not affect the a-posteriori estimation of these states given that (6.22) always operates an amplification of the variance of UWB ranging observables.
In light of the foregoing, a double-update EKF architecture is developed which marginalises the adaptive weighting on UWB covariance statistics to the a-posteriori update of t_d. For the remainder states in (6.16), instead, $\boldsymbol{C}$ statistics are estimated by leveraging the available measurements [206]. The simplified block scheme in Figure 6.9 illustrates the main stages of the proposed double-update model with embedded optimisations.

6.3.5 Real-world Experiment

To validate the performance of the proposed filtering-based calibration methodology for GNSS/UWB tight integration, a real-world experiment has been carried out. The proposed double-update EKF model is experimentally assessed to demonstrate the improved positioning accuracy against both plain EKF integration (i.e., without any time offset calibration) and EKF integration with naive time offset calibration.
The receiver is mounted on a car which rides in a suburban area of the Metropolitan city of Turin (Italy). The maximum achieved vehicle speed is about 10 m/s. Figure 6.10 provides a snapshot of the considered scenario. Although the chosen experimental

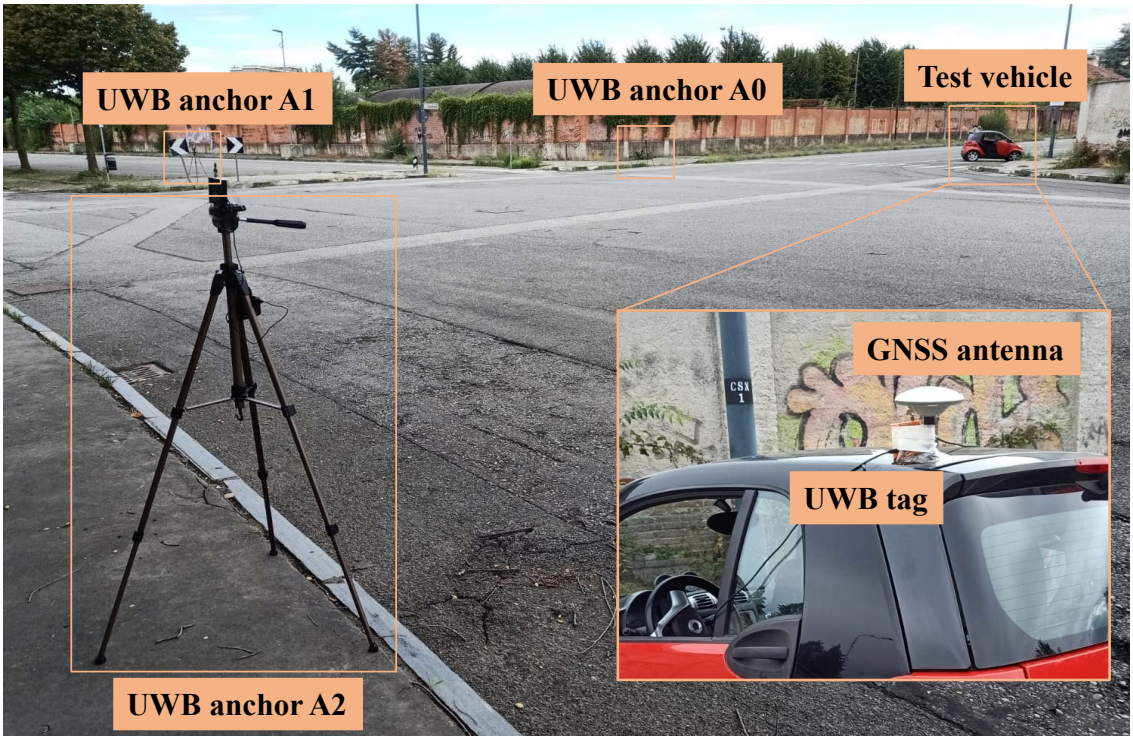


Fig. 6.10 Snapshot of the experimental scenario for the real-world test campaign. The deployed network of static UWB anchors is shown. Moreover, the UWB tag and the GNSS antenna on the road vehicle's roof are highlighted.

environment is not expected to severely degrade the quality of GNSS observables, the results of the following analysis are still valuable. In fact, the primary goal of this section is not just validating GNSS/UWB tight integration per se, but rather to emphasize the accuracy improvements GNSS/UWB tight integration can benefit from when embedding the proposed time calibration strategy. In this context, good GNSS conditions could be considered a worst case scenario for GNSS/UWB tight integration because auxiliary UWB measurements would not bring remarkable accuracy gains.
A u-blox ZED-F9P high-precision module has been used, which integrates multi-band GNSS and RTK technology [207]. Such positioning module is commonly used in the industrial navigation and robotics markets. In addition, a high-gain, multi-band Sino GNSS AT340 geodetic antenna has been deployed [208].
GNSS noisy pseudorange and Doppler-shift measurements have been logged through the u-blox module for the GPS constellation at a rate of 10 Hz. In parallel, the multi-band RTK solution from the high-precision GNSS module has been logged in order to have a ground truth (i.e., reference trajectory) used to calculate error statistics of the navigation solutions.
On the UWB side, consumer EVB1000 boards from Qorvo Inc. have been employed

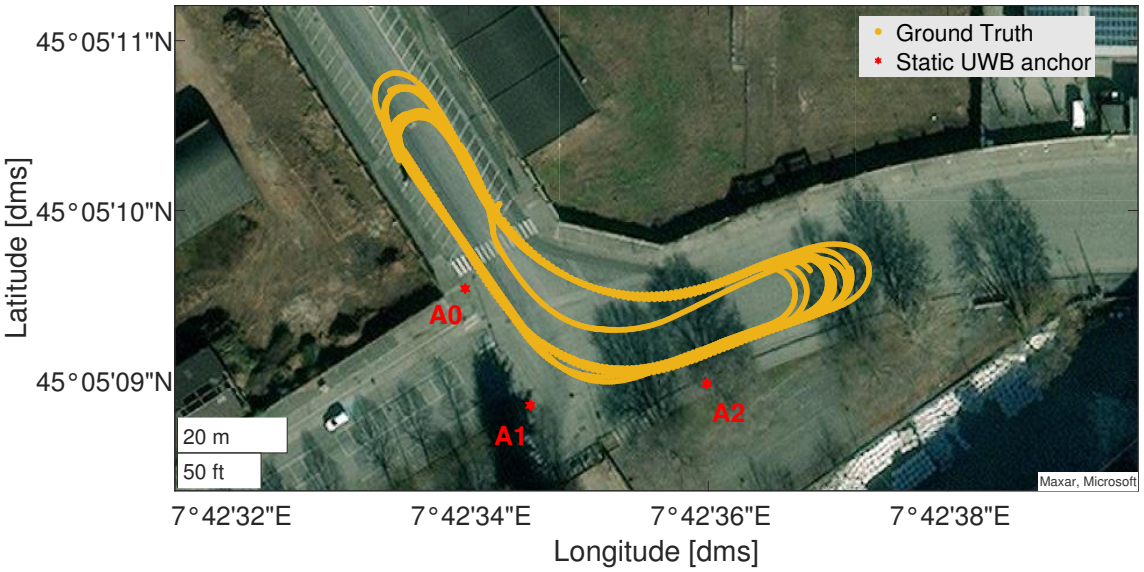


Fig. 6.11 Map view of the ground reference for the real-world dataset used in the validation of the proposed online time calibration method for GNSS/UWB tight integration.

[209]. In particular, three UWB modules have been installed on tripods as static anchors. Their positions have been estimated at sub-decimetre level accuracy using the fixed solution from RTK.

The ground-truth path for the tested trajectory is shown in Figure 6.11 along with the position of the UWB anchors. In addition, a fourth UWB module has been installed on the roof of the vehicle to be operated as a tag. The deployment of both the GNSS antenna and the UWB tag on the vehicle is shown in Figure 6.10.

As an additional remark, efforts have been made to minimise the lever arm between the phase centres of the GNSS and the UWB antennas. UWB measurements from the set of static anchors have been logged through the EVB1000 tag module at 5 Hz rate. When the UWB tag forwards a new measurement to the laptop via the serial port, the Universal Time Coordinated (UTC) time from the laptop is recorded and used as the timestamp of the corresponding UWB measurement.

Besides, an additional time offset of 100 ms is intentionally added to each measurement sample in the UWB dataset, thus increasing the magnitude of GNSS/UWB time offset, in order to test the proposed algorithm under a critical scenario. Nonetheless, since the offset is being added to all the UWB measurement samples by the same amount, it does not undermine the methodology assessment.

Furthermore, due to the inaccuracies and non-idealities of the laptop's clock as well as the transmission delays affecting UWB measurements, it has not been possible to retrieve the ground truth for the GNSS/UWB time offset. Anyhow, the proposed

time-calibration techniques can still be validated by assessing positioning error statistics w.r.t. to the ground truth of the receiver position.

Results

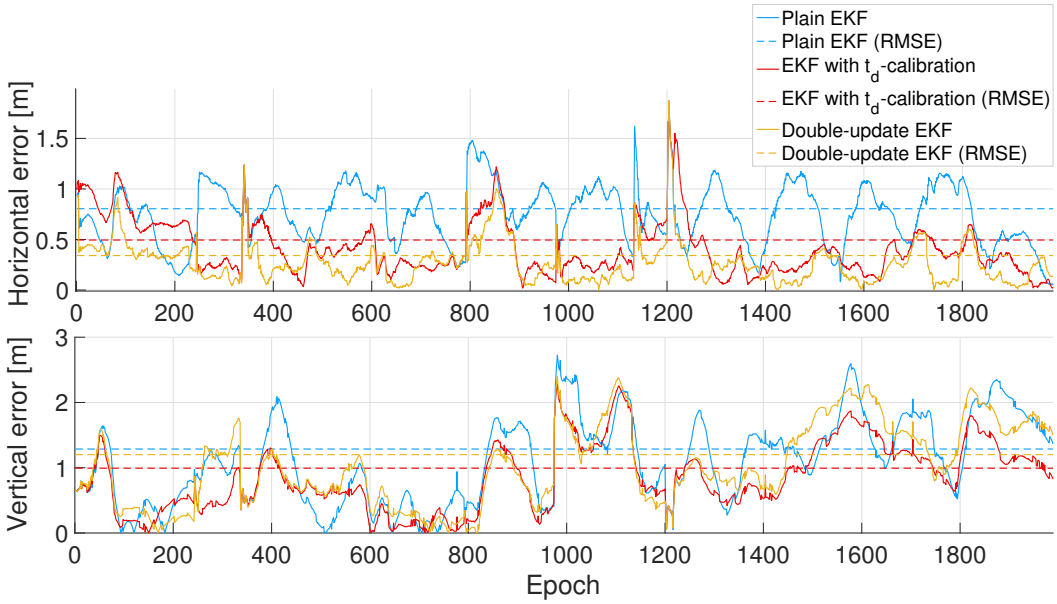


Fig. 6.12 Horizontal (top) and vertical (bottom) position error trend in ENU-coordinates for the real-world dataset. Comparison between EKF plain, EKF with naive t_d calibration and double-update EKF.

Filter	RMSE $[m]$
EKF	0.8043
EKF with t_d-calibration	0.4947
Double-update EKF	0.3412

Table 6.4 Horizontal position RMSE [m] for different t_d calibration architectures in the real-world test.

Figure 6.12 shows the time series of positioning error on both the horizontal and vertical components of a local ENU frame. The associated RMSE are also shown as dashed horizontal lines. As can be noticed, vertical error statistics are penalised by the geometry of the deployed network of UWB anchors. In addition, it should be pointed out that the vertical component of the vehicle speed is close to zero for most of the trajectory.
More importantly, it can be seen how the proposed architecture can reliably deliver

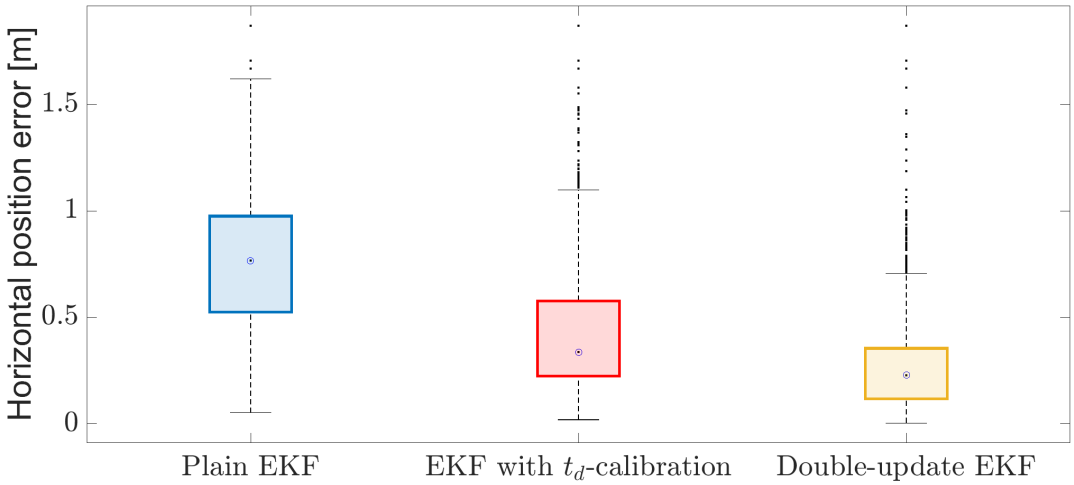


Fig. 6.13 Box plot of the horizontal positioning error statistics in ENU-coordinates for the real-world dataset. Comparison between EKF plain, EKF with naive t_d calibration and double-update EKF.

a lower horizontal error for most of the trajectory w.r.t the other filters. Focusing on the horizontal component, Table 6.4 summarises the positioning RMSEs in units of meters. The proposed double-update EKF achieves horizontal RMSE improvements of 31.03 % and 57.58 % compared to the EKF with naive t_d calibration and the uncalibrated tight integration via plain EKF, respectively.
Eventually, horizontal errors statistics are highlighted in Figure 6.13 in terms of error mean, spread and skewness characterising the positioning estimates delivered by the analysed filters. Considering the 75-th percentile, the double-update EKF achieves horizontal gains of 38.54 % over the EKF with naive t_d calibration and of 63.67 % over the plain EKF. As such, a globally exhaustive evidence of the superior performance of the proposed double-update EKF algorithm for online GNSS/UWB time-calibration is given. It is worth noticing that the conditions of the experimental environment (i.e., few UWB anchors while generous open sky visibility of GNSS satellites) do not limit the validity of the test. The scenario is suboptimal in terms of maximum achievable accuracy of a tight GNSS/UWB architecture, but it shows the benefits introduced by the proposed time calibration technique.

Conclusions

Time calibration is of great concern in GNSS/UWB tight integration leveraging centralised EKF hybridisation. This work has explored the impact of uncalibrated

GNSS/UWB time-offset on state estimation accuracy, both theoretically and experimentally.

Based on these premises, an EKF-based framework has been proposed to address time offset calibration. First, an improved GNSS/UWB tight integration model has been presented to enable the modelling of the unknown time offset as part of the EKF state-space formulation. Then, after pointing out criticalities in time offset estimation caused by reduced identifiability, an enhanced, double-update EKF architecture has been put forward which adaptively weights UWB covariance statistics.

Moreover, results obtained with a real-world kinematic dataset further assess the superior double-update EKF performance by highlighting horizontal gains of 57.58 % compared to the uncalibrated tight integration via plain EKF and of 31.03 % over the EKF with naive time-offset calibration.

Chapter 7

Cooperative Positioning

7.1 Introduction

Accurate position information is paramount for the successful functioning of some emerging applications such as autonomous vehicles. As a matter of fact, it plays a pivotal role in ensuring their safety, efficiency, and overall reliability. Autonomous vehicles heavily rely on precise location data to navigate through complex and dynamic environments, make real-time decisions, and avoid potential hazards [210]. As we have discussed in the previous chapters, the performance of stand-alone GNSS may not be sufficient to satisfy the safety requirements of some of these applications. In order to overcome the limitations of GNSS, its integration with other sensors and other positioning technologies allows vehicles to create a comprehensive understanding of their surroundings, enabling them to follow predetermined routes, respond to traffic conditions, execute manoeuvrers with precision and in general to adhere to vehicle safety practices [211].

Any deviation in position accuracy can lead to significant consequences, including collisions, misjudged distances, or incorrect route planning. Therefore, the importance of reliable position information cannot be overstated, as it forms the foundation for the seamless operation and trustworthiness of autonomous vehicles, ultimately contributing to the widespread acceptance and successful integration of this transformative technology.

Among the multitude possible sources of additional position information, the idea of CP has been proposed to exploit the combined information from multiple devices to enhance accuracy and reliability. In the context of autonomous vehicles, CP involves the exchange of position-related data among networked vehicles (Vehicle-to-Vehicle

(V2V)), infrastructure (Vehicle-to-Infrastructure (V2I)), or other devices (Vehicle-to-Everything (V2X)) to improve the overall performance of their individual position estimates [212].
The cooperative process can involve various sensors, such as GNSS receivers, INS, UWB, Cameras and Light Detection and Ranging (LiDAR). By combining data from multiple sources, vehicles can compensate for limitations or inaccuracies in individual sensors, especially in challenging environments like urban canyons where the performance of stand-alone GNSS is poor.
Some solutions have investigated the integration of GNSS with INS [213] or camera sensors [214, 215] for land vehicles. A more complete survey of navigation technologies for vehicles is presented in [216], while a summary of cooperative strategies and their applications can be found in [217].
However, many studies often give for granted the availability of cooperative data at the receiving node without considering the impact on the underlying network infrastructure in terms of delay, maximum throughput and packet losses. As a matter of fact, a dedicated protocol for the exchange of positioning information does not exist, and the impact on the network of raw position-related measurements has not been investigated, thus providing motivation for this work.
On the other hand, work in the context of vehicular networks often does not account for realistic position-related data and neglects the requirements of CP applications in terms of transmission rate and delay.
Therefore, the goal of this investigation is to close the gap between the two fields, by analysing the current state of the art on CP and then designing a network protocol able to satisfy the requirements needed by CP applications. Finally, it is also of interest to investigate the impact on the network of such protocol, to assess its feasibility and study its performance as a function of the density of nodes participating in the cooperation.
The work presented in this chapter is an original contribution first developed in [3] to enable the exchange of GNSS measurements and then extended in [2] to include also the exchange of measurements from other sensors. In particular, it is the results of a collaboration with a vehicular communication research group within our university. The expertise on GNSS and in particular CP were leveraged for the definition of a protocol which guarantees sufficient requirements to enable the largest possible set of CP applications while minimizing the impact on the network. This was possible thanks to careful review of the state of the art and previously acquired expertise in CP and sensor fusion domains which allowed to set clear goals in terms of necessary measurements, rates, timestamping, number of cooperating agents and so on. The

code used for the definition of the protocol and its assessment is stored in an open repository [218].

7.2 Cooperative Positioning in GNSS Applications

The concept of fusing location information from external sources enables a multitude of CP applications in the context of GNSS. Cooperative strategies have been pioneered for robot localizations [219] and later exploited also in vehicular networks [220]. Previous studies have investigated the correlation between GNSS-based cooperative ranges [221], and further work has analysed the profitability of such information [222]. Some works have instead addressed the problem of tight integration of asynchronous, non-independent, non-stationary, inter-agent distances [223, 224] in navigation filters. The remainder of this Section is dedicated to overview the main applications of CP that can be enabled thanks to the exchange of measurements among networked peers.

GNSS-based ranging. These techniques combine GNSS observables from two nodes to estimate their inter-node distance, even in NLoS conditions. As an example, differential techniques (e.g., Double Differences (DD)), are able to provide relative ranging with a low computational effort and minimum delay [225, 226]. The use of raw GNSS measurements in high-end receivers has recently pushed receiver manufacturers to their disclosure in mass-market devices [227], leading to solutions which estimate inter-node ranges exploiting android smartphone [228]. A proof of concept for cooperative positioning with smart phones has also been proposed [229].

Multi-agent cooperative positioning and navigation. Machine learning solutions have also been explored to integrate relative ranging information, such as in [230]. Other approaches have also been applied to the integration of measurements based on other sensors such as UWB [231], INS [232] or network-based ranges [233]. Recent approaches that use sensitive information related to geographical trajectories and geospatial data for improving performance [234] can also benefit from a dedicated protocol.

Cooperative integrity. The integrity of positioning and navigation data measures the level of trust that can be associated with the information provided by the

navigation system. To complement or extend the monitoring of stand-alone local data, RAIM and more recent variants (e.g., ARAIM, RRAIM, ERAIM) have been proposed. Among new collaborative solutions, Cooperative Enhanced Receiver Integrity Monitoring (CERIM) exploits GNSS raw data shared among networked receivers, in order to perform FDE [235]. A cooperative integrity monitoring based on KF can significantly improve detection sensitivity of faulty GNSS measurements, while also detecting faulty inter-nodes measurements. However, few recently proposed FDE techniques allow for full, multi-sensor integrity [236, 237].

Time synchronization. Relying on an accurate estimation of their on-board clock bias, GNSS receivers can provide accurate timing reference. The availability of in-orbit accurate clocks comes as a distributed timing source that can be exploited by many applications in Intelligent Transportation Systems (ITS). Network interoperability and coordination, scheduling of channels, road safety, network security, and time-to-collision monitoring are only a few among a wide range of possibilities that may be not reached through conventional approaches, such as Network Time Protocol (NTP) and Precision Time Protocol (PTP). Given the high accuracy and the lower complexity of its timing estimation, GNSS-based timing is of large interest. Modern receivers can reach up to 30 ns accuracy of time synchronization between two receivers, as demonstrated in urban environments [238]. An additional advantage is that GNSS time synchronization is characterized by almost 99.98% availability, while positioning solutions only by roughly 80%.

Authentication. The current usage of message authentication in vehicular networks [239, 240] can be further enhanced through the introduction in modern GNSS receivers of the Galileo Navigation Message Authentication (NMA) and GPS CHIpMEssage Robust Authentication (CHIMERA). GNSS message and signals authentications allow discriminating legitimate transmissions from illegitimate ones, thus introducing an independent and time-related source of authentication for network messages and communications at a large extent.

CP and navigation can be also supported by machine learning algorithms to enable specific applications in ITS that can be further complemented by the proposed protocol [241, 242]. The possibility to exchange lower-level information about localization sensors not only enables the aforementioned applications but also modern approaches such as soft information paradigms [112].

In order to enable CP approaches, peer-to-peer exchange of GNSS raw data is required. Such exchange can be pursued through general-purpose (e.g., 5G) or dedicated connectivity (either Institute of Electrical and Electronics Engineers (IEEE) 802.11 Direct Short Range Communication (DSRC) or Cellular-V2X). Given the multiplicity of applications addressed by the CP algorithms reviewed in this section, an application-layer performance analysis falls outside the scope of this work.

7.3 Existing Protocols

This section is dedicated to a review of existing protocol and solutions for the exchange of measured data, both in the context of vehicular networks and GNSS. As we will see, no existing protocol has all the features required to enable the CP approached described in Section 7.2, thus motivating the development of a new protocol designed ad-hoc.

7.3.1 Vehicular Networks

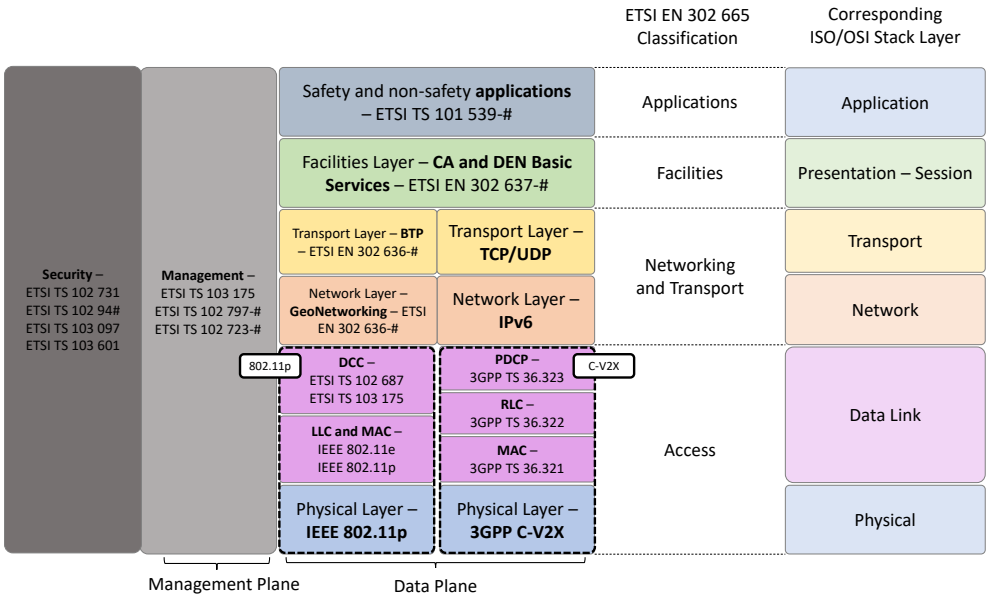


Fig. 7.1 Standardized network stack for a connected vehicle, as foreseen by ETSI. Each layer corresponds to a given ISO/OSI layer, and all the relevant standards (by either ETSI, IEEE or 3GPP) are reported below each protocol.

The cooperation and data exchange between vehicles includes also the transmission and reception of raw GNSS data. As detailed later, this enables several approaches which would not be possible without vehicular networks. IEEE, 3rd Generation Partnership Project (3GPP) and European Telecommunication Standards Institute (ETSI) have standardized different layers of the network stack for vehicular communications. In particular, ETSI foresees the connected vehicle network architecture depicted in Figure 7.1. As can be seen, ETSI mainly defines all the layers from Networking and Transport to Applications, as part of the ITS-G5 set of standards, with a special focus on the so-called Facilities Layer. Its main aim is to provide support to ITS applications and V2X services working at the Application Layer, including several crucial features to manage the transmission and reception of standardized vehicular messages [243].
These messages include, among the most important ones:

- *Cooperative Awareness Message (CAM)*, periodic messages transmitted with a frequency between 1 Hz and 10 Hz and storing kinematic and dynamic data such as vehicle position, heading, speed, acceleration, alongside additional information such as steering wheel angle and turn indicator status [244];
- *Decentralized Environmental Notification Message (DENM)*, event-based messages leveraged to notify vehicles and infrastructure nodes about hazards and other events on the road [245];
- *Infrastructure-to-Vehicle (I2V)* information messages to carry road signage information;
- Other less commonly used messages, such as RTCM Extended Message (RTCMEM) [246] and Services Announcement Essential Messages (SAEM) [247].

Below the Facilities layer and alongside the IPv6 stack, ETSI defines the so-called ITS-G5 Transport and Networking layers. These layers include, respectively, Basic Transport Protocol (BTP) [248] and GeoNetworking [249]. The first is a lightweight connection-less transport protocol, comparable to User Datagram Protocol (UDP) and containing port numbers which are used to determine the destination service (e.g., the one managing CAM) when ITS messages are received by vehicles. According to ETSI, two types of BTP headers are foreseen: BTP-A for interactive sessions, in which the recipient is expected to provide a reply, or BTP-B, which is used for non-interactive message dissemination. GeoNetworking is instead a network layer protocol which defines several ways of performing geographical routing of messages,

and describes different types of addressing schemes.
These European standards are flexible enough to be encapsulated into either IEEE 802.11p, a dedicated V2X Wi-Fi amendment based on IEEE 802.11a [250], or 3GPP C-V2X, which represents the application of cellular technologies to vehicular networks [251]. The latter supports an infrastructure-less communication mode, called *Mode 4* [252], which can be used as an alternative, or in conjunction with, IEEE 802.11p. The use of next generation network technologies such as 5G new radio for V2X and CP has also been investigated in [253, 254].
Starting from the vehicular stack architecture presented earlier, several projects have been funded and are currently ongoing, with the aim of creating smart vehicle infrastructures supporting testing and deployment of V2X applications. Among them, it is worth mentioning:

- 5G-CARMEN [1], a project aimed at creating a smart and connected corridor between the cities of Modena in Italy and Munich in Germany. These smart vehicle infrastructures include both equipped vehicles and Road Side Unit (RSU), which can support V2V and V2I communications thanks to ETSI ITS-G5.
- Other projects such as the Aveiro Open Lab [2] also leverage IEEE 802.11p communication [255].
- In parallel, the HANSEL project, an ESA funded activity (ESA ITT number AO/1-9494/18/NL/CRS), has included Differential GNSS (DGNSS) CP among the innovative approaches for navigation in smart cities. The proof-of-concept relied on 4G/LTE physical layers for the exchange of raw GNSS data among smart devices [256].

7.3.2 GNSS

The concept of transmitting or broadcasting GNSS raw data between receivers and reference stations is not new. Currently, there are three main formats that support the transmission of GNSS position-related data among connected nodes, each with different kinds of limitations that do not allow the full applicability of the aforementioned CP paradigms. More in detail:

[1] https://5gcarmen.eu/
[2] https://aveiro-living-lab.it.pt/

- *National Marine Electronics Association (NMEA)* 0183 is an American Standard Code for Information Interchange (ASCII) based serial communication protocol. It supports the so-called GGA strings that provide Time, Position and fix related data for a GPS receiver. However, NMEA standard is proprietary and does not support raw GNSS data.
- *Radio Technical Commission for Maritime Services (RTCM)* 10403 is a proprietary standard series that describes messages and techniques for supporting GPS and GLONASS operation with one reference station, or a network of reference stations. The Network Transport of RTCM via Internet Protocol (NTRIP) protocol can be used to transmit RTCM messages for differential corrections in RTK schemes. RTCM natively supports real-time-oriented exchange of raw GNSS measurements.
- *RINEX* is a data interchange file format for raw GNSS data popular in geodesy. RINEX files can contain observation data (i.e. raw measurements), navigation message data and atmospheric condition models as well. Despite the use of some compression schemes (i.e., Hatanaka file compression), RINEX files are not suitable for near real-time applications, and they are used mostly for post-processing and off-line investigation.

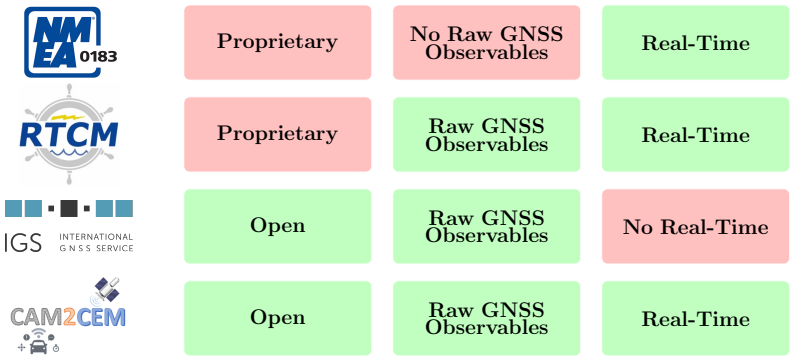


Fig. 7.2 Summary of GNSS data protocols [2].

Figure 7.2 highlights the main differences between the existing protocols and the goals of the proposed CEM protocol.

7.3.3 Limitations of Current Protocols for Cooperative Positioning

Considering the various limitations of these formats and protocols, an open protocol for the exchange of raw navigation data between vehicles is still missing, as well as a simulation and emulation framework for the experimental assessment of the impact of the transmission of GNSS data on the network performance. Therefore, the aim is to extend the capability of the CAM [257] and Collective Perception Messages (CPM), whose performance has been analysed in [258–260]. Furthermore, the current performance of CAM-based safety applications has been investigated in [261].
The position-related information present in CAMs is only able to support passive awareness of the surrounding environment. Received information can be exploited locally in Advanced Driver-Assistance Systems (ADAS) and enables a large portion of V2X use cases, but it does not actively contribute to enhancing absolute and relative localization capabilities. In particular, the transmission rate of CAMs can variate between 1 Hz and 10 Hz, depending on the vehicle dynamics [244], and messages comprise minimal information about positioning and navigation states. As a matter of fact, these messages are used to broadcast final output data of navigation filters, such as position, speed and heading values, which are not suitable to implement most of the CP applications.
To overcome these limitations, ETSI has recently proposed the so-called Cooperative Positioning Service (CPS) [262]. Such service, exploiting the CPMs, has the goal of improving the awareness of ITS nodes by exchanging measurement about the surrounding environment. The perception of objects around the nodes can be obtained thanks to on-board sensors. Therefore, CPSs provide an enhanced awareness of the surroundings, but they still do not allow any positioning accuracy improvement based on exchanged information.
It is also worth mentioning that ETSI has specified the RTCMEM in order to encapsulate RTCM data, along with its transmission from infrastructure nodes to vehicles, as part of a GNSS positioning correction service [246].
Nonetheless, the proposed CEM has some significant differences from such messages. In particular: *i)* the main purpose of RTCMEM is to enable a V2I differential positioning scheme, rather than to directly enable peer-to-peer CP approaches, *ii)* RTCM data, as just mentioned, is encoded using a proprietary and closed-source protocol, *iii)* no optimization approach is employed to reduce usage of network resources.

7.4 Cooperative Enhancement Message

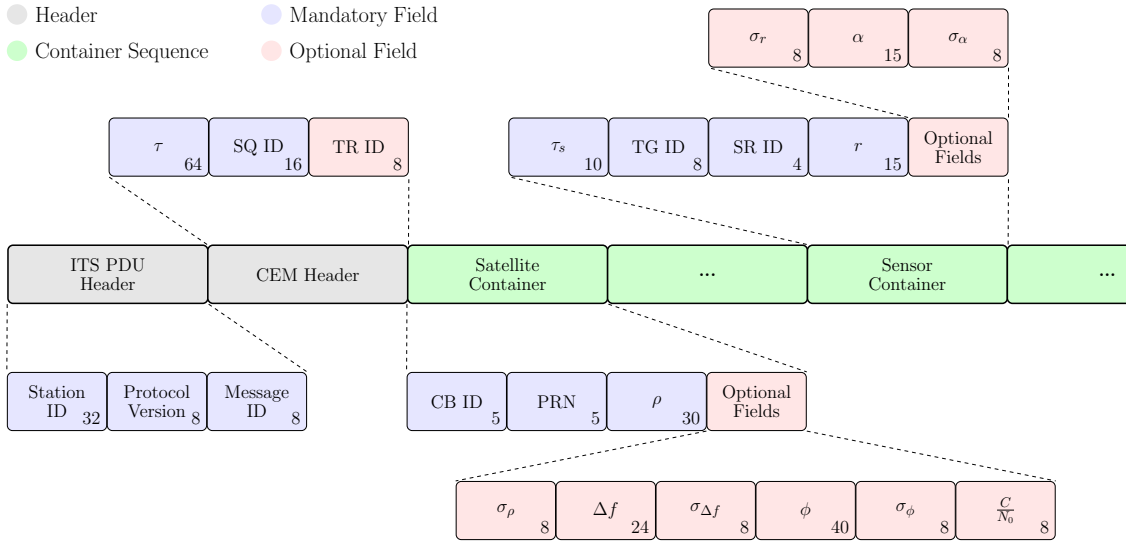


Fig. 7.3 Diagram of a CEM *I* frame. Field size (bit) is indicated in the bottom right corner. *D* frames contain an additional sequence identifier in their header [2].

Given the limitations of current protocols for the exchange of cooperative positioning data, we introduce the CEM protocol, a dedicated solution for the exchange of raw positioning data in vehicular networks. Furthermore, the CEM messages used to encode and transmit data among the network nodes are also described. Although CEMs are meant to complement the information provided by CAMs, the proposed protocol can also be easily adapted to be employed as a stand-alone solution.
It is indeed worth highlighting how CEMs can be used both in stand-alone fashion as well as in parallel with other kinds of messages such as CAM, if needed by the overlying V2X applications. Furthermore, even if the main purpose of CEMs is to be exchanged between vehicles, they can also be exchanged between other moving nodes or network infrastructure.

7.4.1 CEM packets

Figure 7.3 depicts the general structure of a CEM message. As can be seen, each message includes an ITS Packet Data Unit (PDU) Header, exactly as in CAM [244] and DENM [245]. This header is composed by a mandatory *station ID*, namely an identifier of the transmitting vehicle, the *protocol version*, which should be currently set to 1 (future versions of the CEM protocol will foresee increasing numbers as

protocol versions), and the *message ID*, which corresponds to *200* for CEMs (while it should be set equal to *2* for CAMs).
After the ITS PDU header, each CEM message consists of a dedicated CEM header and a sequence of *satellite containers* (in principle one for each received GNSS signal). The header of each CEM includes a single timestamp, associated to the measurements included in the message. It can be used to synchronize measurements from different nodes, which in principle all work independently from one another and take measurements at different time instants (and possibly different rates as well). It should be emphasized that the synchronization step is fundamental in order to enable most cooperative applications.
A *TR ID* can optionally be included in the header, with a value ranging from 0 to 255 (thus using 8 bits). This identifier can be used to define any ad-hoc strategy for the identification of the nodes involved in a CP approach. If not specified, the CEM protocol will use the standard ETSI *station ID*. The usage of identifiers (either ad-hoc or based on the station ID) is needed when including sensor containers from other ranging sensors, as described later in Section 7.4.4.
Future developments of the protocol foresee the possibility of including other application-specific flags and general purpose information in the header. It should be pointed out that in any case, the inclusion of such fields in the header should not contribute to a large increase in terms of packets size, and therefore should not heavily influence the network performance. All other fields are encapsulated into the satellite containers. Every container can include one value of the measurement uncertainty, namely σ, for each of the three types of GNSS observables present in the message. In order to limit the maximum size of messages, the protocol specifies that no more than 10 containers should be included by a transmitting agent inside a CEM message. In case the GNSS receiver has visibility of more than 10 satellites at any one time, it has the freedom to choose which ones to include in the packet (e.g., the ones with highest $\frac{C}{N_0}$). Finally, it is worth mentioning how CEMs can be potentially encapsulated into any transport and network layer, such as UDP over Internet Protocol (IP), or the ETSI ITS-G5 transport and networking layers [248, 249], depending on the user needs. The latter normally represents the stack of choice for the dissemination of vehicular messages in Europe, and CEM are thus designed to fully support their encapsulation inside BTP and GeoNetworking. In particular, CEM should use the BTP-B header when being encapsulated inside BTP, as this kind of header is normally used for broadcast messages. The standard also defines a BTP port number for each vehicular message type (e.g. 2001 for CAM, 2002 for DENM) [263]. We thus define a new port number for the CEM protocol,

which is set to 2200, since it is a value currently not used by ETSI.
Furthermore, among the GeoNetworking addressing schemes, CAM rely on a Single Hop topologically-scoped Broadcast (SHB), since they are broadcasted only to the nearest first hop. In a similar way, CEM are designed to be broadcasted as SHB messages [249].

7.4.2 Encoding

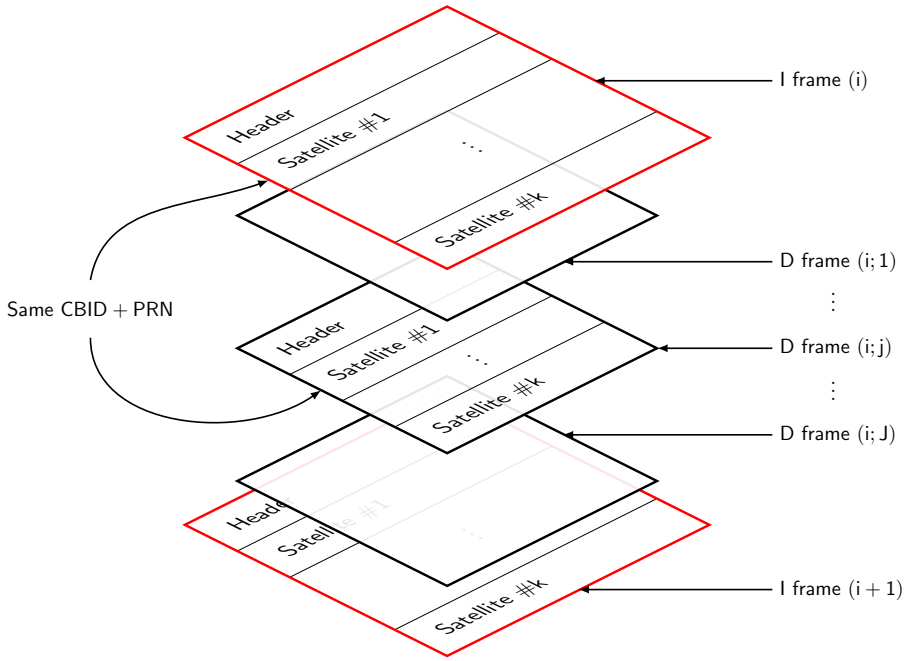


Fig. 7.4 Example of a stream of CEM packets. Agent sends J differential messages within two Intra-messages. Each message contains information related to k satellite signals [3].

To reduce the load on the communication network, CEMs are designed to employ a low-complexity differential encoding. Furthermore, it exploits the fact that different kinds of data needs to be updated more or less frequently, and that not all information is strictly necessary in order to employ CP approaches. Thus, the protocol encompasses two types of CEMs:

1. An *Intra-message (I)*, which contains the full value of measurements, including both data that needs to be updated at high and low frequency;

2. A *Differential message (D)*, which contains differential values. More precisely, it consists in the difference between the current measurements and the ones transmitted in the most recent *I*. Only high frequency data is included.

*I*s are transmitted at a fixed interval of 1 s. Spaced out between *I* are instead the *D*s, with a basic interval of 100 ms, so that at most 9 can fit between two consecutive *I*s. It is up to the transmitting node to adjust its rate of *D*s depending, for example, on which target application needs to be enabled or on the congestion of the network. The specific strategies for the adjustment of the transmission rate are outside the scope of this work, and will be considered for future developments of the protocol. Figure 7.4 provides an example of the flow of CEMs within two successive *I*s. The choice of such time intervals between frames was motivated by considering common vehicle dynamics in urban environment and common rate of solution of GNSS receivers. These values are also in line with the rate intervals specified by CAM. It is worth mentioning that higher rates might facilitate time synchronization between received and local measurements. However, recent studies on time calibration between GNSS and UWB measurements show that time misalignment between multi-sensor range measurements can be compensated through measurement integration algorithms [264, 9]. Thus, higher rate would bring little benefit while significantly increasing the load of wireless channels. The header of *I*s contains a unique sequence identifier (*SQ ID*). The mandatory data in a satellite container of a *I* frame consists of both *CB ID* and *PRN* identifiers, along with the pseudorange measurements. As a consequence, all the other fields are optional. It should be added that if the transmitter decides to include optional data, it is not forced to send all of it, but can instead choose whatever sub-set of fields to include, depending on what is available or needed by the application.

Regarding instead the *D*s, two sequence identifiers need to be included. The first is its own unique message identifier and the second is the one of last *I*s. The addition of these identifiers allows an easy sequence reconstruction by the receiver, if needed. Within every *D*s satellite container, only the GNSS observables are sent, and only differential pseudorange is mandatory. Other data that does not need to be updated often, such as the variances and $\frac{C}{N_0}$ is therefore not included in *D*s. Furthermore, also *CB ID* and *PRN* are not included in *D*s as they can be easily inferred since measurements from different signals appear in the same order as in the last *I* frame. To avoid redundancy, fields that are already present in CAMs have been omitted in the current implementation of CEMs, but it is foreseen to add an optional container encapsulating such fields which can be used in scenarios in which CAMs are not present and CEM works in stand-alone fashion.

Measurement	Symbol	Min. Value	Max. Value	Prec.	Units	Bits
Code pseudorange	ρ	$1.9 \cdot 10^{10}$	$2.4 \cdot 10^{10}$	10^{-2}	m	30
Carrier phase	ϕ	$0.7 \cdot 10^{8}$	$1.6 \cdot 10^{8}$	10^{-3}	Cycles	40
Doppler shift	Δf	$-5.0 \cdot 10^{3}$	$5.0 \cdot 10^{3}$	10^{-3}	Hz	24
C/N0	C/N_0	20	70	0.25	$\mathrm{dB\,Hz^{-1}}$	8

Table 7.1 GNSS observables - CEM Intraframe (I)

7.4.3 GNSS measurements

More in detail, CEMs can include the following network and GNSS data and observables:

1. τ: accurate timestamp of when the observables where measured;
2. TR ID: identifier of the transmitting node;
3. CB ID: identifier of the constellation and signal frequency;
4. PRN: identifier of the satellite from which observables are measured. Receiver can infer the satellite since each one transmits a unique pseudo-random code;
5. Pseudorange ρ: a value in meters of the distance between satellite and receiver (it also accounts for the time difference between transmitter and receiver clocks);
6. Carrier-phase ϕ: also a value of distance expressed between satellite and receiver, but instead expressed in terms of number of phase cycles of the carrier frequency;
7. Doppler Δf: shift in received frequency caused by the relative velocity between satellite and receiver;
8. Variance σ: a value of the measurements uncertainty that can be estimated by receivers (one value is associated with each observable ρ, ϕ, and Δf);
9. $\frac{C}{N_0}$: C/N0.

Ranges of values

The ranges of values that GNSS observables can take for the two kinds of messages described above are summarized in Table 7.1 and Table 7.2 respectively. The last

Measurement	Symbol	Min. Value	Max. Value	Prec.	Units	Bits
Code pseudorange	ρ	$-1.0 \cdot 10^3$	$1.0 \cdot 10^3$	10^{-2}	m	18
Carrier phase	ϕ	$-5.5 \cdot 10^3$	$5.5 \cdot 10^3$	10^{-3}	Cycles	24
C/N0	Δf	$-3.0 \cdot 10^1$	$3.0 \cdot 10^1$	10^{-3}	Hz	16

Table 7.2 GNSS observables - CEM differential frame (D)

Constellation	SB #1	SB #2	SB#3	SB #4	SB #5
GPS	L1 (1)	L2 (2)	L5(3)	-	-
GLONASS	G1 (6)	G2 (7)	G3 (8)	-	-
Galileo	E1 (11)	E2 (12)	E5a (13)	E5b (14)	E6 (15)
BeiDou	B1 (18)	B2 (19)	B3 (20)	-	-

Table 7.3 Constellations and Signal Bands (identifiers in round brackets are compliant with RINEX standard)

column shows an approximate amount of bits that is needed to represent the ranges with the corresponding accuracy in order to provide a rough idea of the amount of data needed. These ranges have been defined considering the minimum and maximum values obtained from real GNSS datasets. In the case of *D*s, these ranges are defined based on the largest possible variation of observables over a time span of 0.9 s, which is the maximum distance in time between a *D* and the latest *I*. All the uncertainties, as well as the $\frac{C}{N_0}$, are represented with a value from 0 to 200 (corresponding to roughly 8 bits). The protocol does not specify any specific combinations of ranges and precision, so that any implementation has the flexibility to map these values as preferred. Since information regarding the quality of signals is mainly exploited to obtain weighted estimates, its accuracy is not as crucial as for the measurements, and can be represented with only a few bits.

The timestamp is defined as the number of nanoseconds from 2004-01-01T00:00:00.000Z (i.e., January 1st 2004, at midnight Coordinated Universal Time), represented over 64 bits. The format is compliant with the ETSI standards, as it is the same format of timestamps stored inside CAMs [265]. Given the number of signals and constellations currently enabled by the protocol, the CBID is represented over 5 bits (32 values). The PRN for the satellites are also represented over 5 bits.

Table 7.3 provides a summary of the constellations and signal bands currently included and defined in the CEM protocol. The *CB ID* field is defined for a range of 32 values (equivalent to 5 bits). The ID numbers which are not currently used can be reserved for a possible future addition of either new signals from the already

included constellations or for other constellations (e.g. SBAS, QZSS, IRNSS).
For every field, there is an additional value always reserved to communicate the case in which the information is outside the defined bounds or not available at all. This could for example occur if a satellite is no longer visible, or if the receiver is unable to compute a certain data. In the first case, the container for that satellite will still be included (with unavailable data) in the following *D*s until the next *I*. The value reserved for unavailable data is always defined as the maximum value plus one time the corresponding precision, and it is therefore set to 201 for the uncertainties and $\frac{C}{N_0}$. As CEM is designed to be fully ETSI-compliant, the description language ASN.1 was used to define the content of each *I* and *D* frame. Indeed, ASN.1 is the same kind of language used by ETSI to define the content of all vehicular messages [244]. Importantly, it is possible to leverage ASN.1 files to automatically generate the source code of encoding and decoding functions, thanks to tools like *asn1c* [266]. After defining the content of a message with ASN.1, several encoding rules are then made available to the user, which define how the message will be encoded and provide different advantages and disadvantages in terms of processing speed and message compactness (which in turn translates into a reduced load on the wireless channel) [267]. When defining a new type of vehicular message it is thus crucial to agree on a proper encoding option. All ETSI standard-compliant messages, such as CAM, make use of Unaligned Packed Encoding Rules (UPER), guaranteeing small message sizes [244]. We have therefore designed our CEMs to be encoded with the same rule.

7.4.4 Other sensors

Measurement	Symbol	Min. Value	Max. Value	Precision	Units	Bits
Range	r	0	300	10^{-2}	m	15
Angle	α	0	360	10^{-2}	Deg	15
Diff. time	τ_s	0	100	10^{-1}	ms	10

Table 7.4 Sensor data - CEM Intraframe.

Considering harsh scenarios such as urban canyons, the visibility of GNSS satellites can be reduced and the quality of the received signal is degraded. In such cases, agents might want to rely on other measurements from on-board sensors to improve their positioning capabilities.
In order to enable agents to share such measurements between each other, we now discuss the addition of a dedicated *sensor container* for the transmission of range

data from other sensors. In addition to the GNSS data, the CEM protocol also enables the exchange of range data obtained from other on-board sensors such as radar, LiDAR, cameras, or UWB interfaces.
Although this study was conducted mainly based on the characteristics of such sensors, the additional sensor container has been designed so that it can contain ranging data from any other type of sensor that could be used to obtain range measurements between agents. In this way, the CEM protocol is flexible enough to be tailored, with small modifications, according to the specific implementation of cooperation among vehicles, or to the use of new types of sensors in the future.
The addition of this container for ranging data beyond GNSS enables a wide range of implementations of the applications that combine both GNSS data and other ranging measurements to further enhance the positioning performance.
The containers for GNSS data shown in Figure 7.3 can be therefore replaced with the newly designed container. Indeed, for each container slot in the CEM packet, it is up to the transmitter to choose whether to send container data for GNSS signals or for other sensors.
The container for the additional cooperative information from other sensors has been defined in order to include the following data:

1. τ_s: accurate time difference between GNSS and sensor measurements;
2. Target ID (*TG ID*): identifier of target vehicles w.r.t. which the measurements are taken;
3. Sensor ID (*SR ID*): identifier of the type of sensor or signal used to obtain the measurement;
4. Range r: a measurement in meters of the distance between the transmitting (ego) vehicle and the target one;
5. Angle α: angle of the target vehicle with respect to a common reference frame;
6. Variance σ: an estimate of the measurements uncertainty (associated to r and α).

As mentioned earlier, an ID of the transmitting vehicle is included in the CEM header when one or more cooperative satellite or sensor containers are present in the CEM packet. Thanks to the presence of this identifier, each of the sensor containers can include a *TG ID* (of 8 bits) to uniquely identify the target vehicle with respect to which the measurements are taken. In this way, transmitting and receiving vehicles

can all unambiguously identify each other, and properly associate and collect all measurement regarding each vehicle.
The *SR ID* is included so that any agent can integrate the received cooperative measurement using an error model based on the type of sensor from which the measurement was obtained. The protocol can define up to 16 different types of ranging sensors. An additional scalar value, τ_s, is included in the sensor container to represent the difference between the header timestamp and the time at which the measurements are obtained from the sensor. Indeed, other sensors may work asynchronously with respect to the GNSS on-board unit, and collect measurements with different rates. To reduce as much as possible the amount of data transmitted, this quantity is expressed as a differential scalar value τ_s, which is the time difference between the most recent measurement obtained from the sensor and the timestamp τ provided in the message header. The field hosts values in the range $[0, 100]$ ms with a granularity of 0.1 ms, leading to 1,001 values, which can be represented using 10 bits.
Sensors measurements are expected to be always less fresher than GNSS data carried by CEM. Note that the minimum time resolution has been set based on current studies and investigations on GNSS and asynchronous sensors integration.
Finally, it is worth mentioning how sensor containers are designed to be inserted in CEMs after all available satellite containers. This, thanks to the UPER encoding rules, enables an easy message decoding at the receiver side, without the need for including a dedicated field carrying the type of each container.

Management of Vehicle IDs

Each CEM may contain an optional *TR ID*, when not relying on the ETSI *station ID* as vehicle identifier. The latter can indeed be fixed on a per-vehicle basis, or anonymized with different strategies [268], and its management according to the ETSI standards may not be suitable for all the CP approaches.
Thus, two possible scenarios for an ad-hoc assignment of *TR ID*s are considered:

1. In case of a platoon of vehicles that move together, IDs can be fixed and assigned a-priori;

2. Otherwise, if vehicles are continuously moving in and out of range of each other, a dynamic allocation of IDs is needed, such that two vehicles in range of each other cannot have the same ID. In any of the two cases, the specific strategies for the assignment of IDs is beyond the scope of our work. Another

issue that arises in the transmission of the cooperative measurements is the data association problem. When the ego vehicle is measuring a distance with respect to a target vehicle using on-board sensors, it needs to know which vehicle it is, so that the corresponding ID can be associated to the measurement and included in the sensor container of the CEM. This task is also out of the scope of this work, but we would like to highlight its importance. A proper management of the IDs is anyway crucial for the correct utilization of the cooperative measurements.

Range of values

The selection of ranges of value is based on review of the types of sensors commonly integrated with GNSS [269], common vehicles dynamics in urban environments, and the broadcasting range of communication technologies.
The maximum value for the inter-vehicle range r is set to 300 m. This field is also the only mandatory measurement for this container, while every other field is optional. For angle measurements instead, all values α are possible, as in general vehicles could be anywhere around each other. For both these quantities, variances are also included, as done for the GNSS data, to enable weighted estimation techniques. These uncertainties are represented with a value from 0 to 200, exactly as for the GNSS container.
Given the usual rate at which sensors obtain measurements and the basic rate of the CEM protocol, the maximum value of τ_s has been set to 100 ms. A summary of all the considered ranges is provided in Table 7.4, with a rough amount of bits needed to represent the given range and corresponding precision.
As for the differential values in D, given the possibility of fast changing vehicles dynamics in an urban environment, all differential values of the angle α are still possible at any time, especially when the cooperating vehicles are in proximity of each other. Therefore, the full value of α is transmitted again at every D. On the other hand, considering realistic vehicle speeds, the maximum value of the differential range has been set to 80 m.
To provide an extreme case as a reference, considering that a D can be generated at most 0.9 s after the original I, if two vehicles were to travel at a speed of 120 $\mathrm{km\,s^{-1}}$ in opposite directions, then their distance would change of at most 60 m in that time span.

7.5 Experiments

In this section, we first introduce the dataset we use to perform our analysis. For our study, we build a novel, open dataset named Synthetic Accurate Multi-Agent RealistiC Assisted-gNss DatAset (SAMARCANDA). It consists of a collection of accurate GNSS PVT estimates and RINEX files obtained from 19 distinct vehicular tracks collected through a multi-band, multi-constellation GNSS/INS/RTK high-accuracy receiver mounted on a car.
The dataset emulates a fleet of vehicles travelling across a urban area of approximately $50.34\,\mathrm{km}^2$. This provides a realistic source of data that is typically hard to be synthetically generated with simulation frameworks such as Simulation of Urban MObility (SUMO). The trajectories acquired asynchronously are reproduced according to a synchronous scheduling in the simulation framework, namely ms-van3t. It is worth stressing that, thanks to such dataset, ms-van3t can account for real positioning errors, unlike conventional traffic simulators such as SUMO.
Subsequently, we describe more in detail the new simulation tool that has been developed, starting from the existing ms-van3t simulator [270].

7.5.1 Simulation Framework

ms-van3t is an open source framework for the evaluation and simulation of ETSI-compliant V2X applications, which includes several state-of-the-art models, enabling the simulation of different access technologies. These include IEEE 802.11p [250], C-V2X Mode 4 [252] and LTE.
It already provides several sample applications, which can be used as a base to build and simulate more complex scenarios, both with V2V and V2I communications [271]. It includes a complete ETSI ITS-G5 stack. With the aim of extensively evaluating the CEM protocol, while at the same time leveraging an open source tool, we have developed a dedicated version of ms-van3t. This version, named *ms-van3t-CAM2CEM*, comes with several enhancements with respect to the original framework.
Specifically, the ms-van3t framework has been enhanced as follows:

- We have generated the encoding and decoding functions for the CEM messages, thanks to the *asn1c* tool [266]. *asn1c* enables the automatic generation of C/C++ message handling functions starting from any ASN.1 data structure.

In particular, the tool was fed with the latest version of the CEM specifications, and the resulting code was integrated into ms-van3t.

- We have implemented and integrated a novel Cooperative Enhancement (CE) basic service managing the CEM protocol. This includes management of the differential enconding scheme and optional containers and fields, the transmission of CEM messages at the proper frequency, and the reception of CEM messages along with the extraction of the relevant data, which is then made available to the V2X applications; this service provides an easy-to-use interface to the ms-van3t user thanks to callbacks;

- We have developed a set of dedicated examples, which can be used as a starting point to simulate the exchange of CEM in urban scenarios, leveraging different access technologies;

- We have integrated the SAMARCANDA dataset, thanks to a full-fledged Comma Separated Values (CSV) version which includes pre-processed vehicle dynamic data, including geographical positions, speed, acceleration and heading of the different vehicles. ms-van3t can leverage the real dataset thanks to the `gps-tc` module, which can be used as mobility simulator instead of relying on the SUMO tool (the default mobility manager in ms-van3t);

- We have implemented and integrated a special module, called `gps-raw-tc`, which is able to read raw GNSS traces (i.e., files containing raw observables from pre-recorded vehicular traces) and provide them as input to the CE basic service for the generation of CEMs. The newly created module can work in conjunction with SUMO, when sample raw GNSS traces are assigned to simulated vehicles (i.e. traces with exactly the same data types and ranges as real traces, but without application-layer informative content) for the evaluation of network-related parameters. It can also work with `gps-tc`, to simulate a real scenario, where vehicles are generating CEMs with actual application-layer content, for the evaluation of full-fledged CP approaches;

- Finally, we integrated a sample raw GNSS trace from the SAMARCANDA dataset, with the aim of simulating the transmission and reception of observables and evaluate the network performance when CEMs are being transmitted besides other standard-compliant messages, mainly CAM.

ms-van3t-CAM2CEM is specifically designed for the evaluation of CP approaches. We have made it available on GitHub with an open source license [218]. Our enhanced

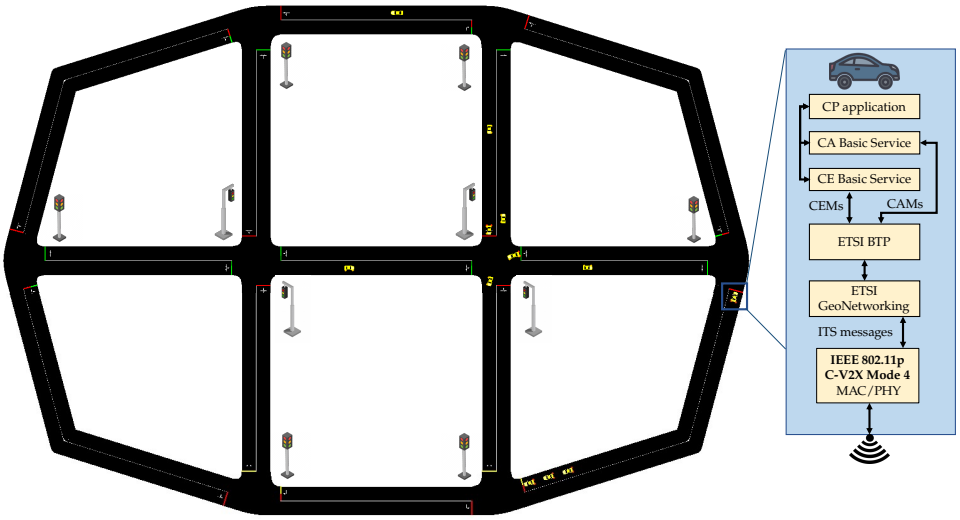


Fig. 7.5 The simulated urban scenario of choice for the CEM evaluation from a network standpoint. It is composed of several intersections regulated by traffic lights, with two central intersections and an outer ring with one lane per direction of travel. The scheme also depicts the different modules which are deployed on each simulated vehicle.

framework also includes a module, called *PRR supervisor*, which is able to report latency and Packet Reception Ratio (PRR) in an automatic and transparent way with respect to the underlying access technology. In particular, the PRR provides a quantitative measurement on the network reliability, showing how many packets are lost under different scenarios and access technology settings. A PRR of 100% means that no packets are lost, and represents an ideal condition, while a theoretical PRR of 0% would mean that no communication is possible. This module has been used to gather the results presented in the next section, which validate the CEM approach from a networking point of view.

7.5.2 Results

This section presents the most relevant results coming from an extensive simulation campaign, performed to validate the CEM protocol, when vehicles exchange both CAMs and CEMs in a realistic urban scenario.
After an initial set of tests performed using the SAMARCANDA dataset, we present a detailed scalability analysis, in order to assess the network behaviour when the number of connected vehicles grows large.
To do so, a dedicated simulation scenario has been set up on the described framework. In particular, since the main goal is to thoroughly evaluate the network performance

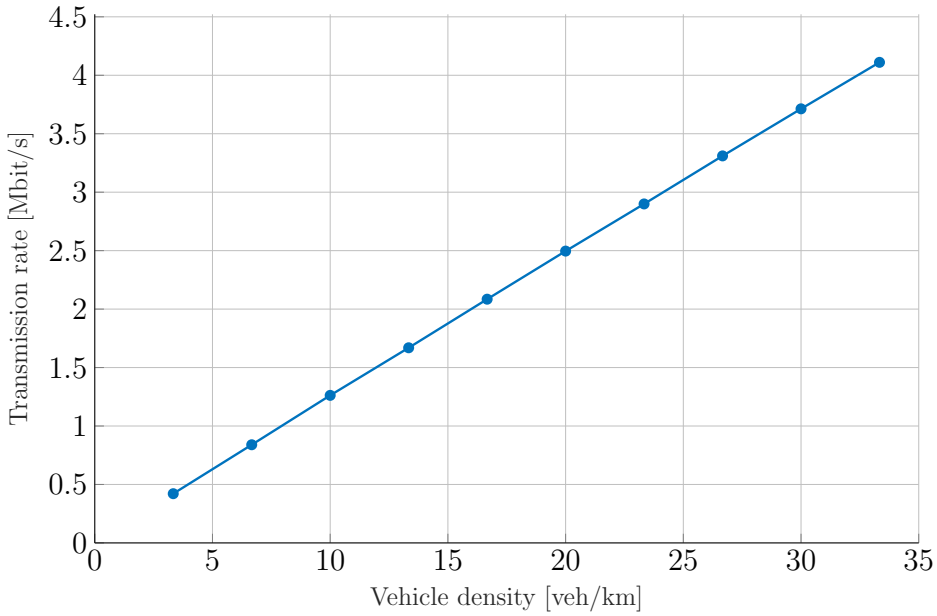


Fig. 7.6 Total transmission rate, representing the amount of bytes transmitted per second as a function of the vehicle density, when both CAMs and CEMs are being actively transmitted by vehicles [3].

when CEM is transmitted, SUMO is used to simulate the vehicles trajectories, instead of directly relying on the SAMARCANDA dataset. This choice is led by the possibility of arbitrarily increasing the number of vehicles without any hard limit (other than the one imposed by the hardware). This enables the analysis of the network behaviour with a higher number of vehicles than what would be available in SAMARCANDA.
Following this approach, each vehicle is assigned a sample raw GNSS trace from the SAMARCANDA dataset, as SUMO cannot provide such information. Even if the traces are sample data, not directly related to the simulated positions in SUMO, this methodology allows us to simulate the exact range of values, data types, and data size that we would expect in real-world applications. The urban scenario considered for this evaluation campaign is depicted in Figure 7.5. As can be seen, each vehicle is equipped with:

- A CA Basic Service for the exchange of CAM;
- A CE Basic Service for the exchange of CEM;
- A sample CP application receiving data from both Basic Services and storing it locally;
- An ETSI ITS-G5 stack with the GeoNetworking and BTP layers;

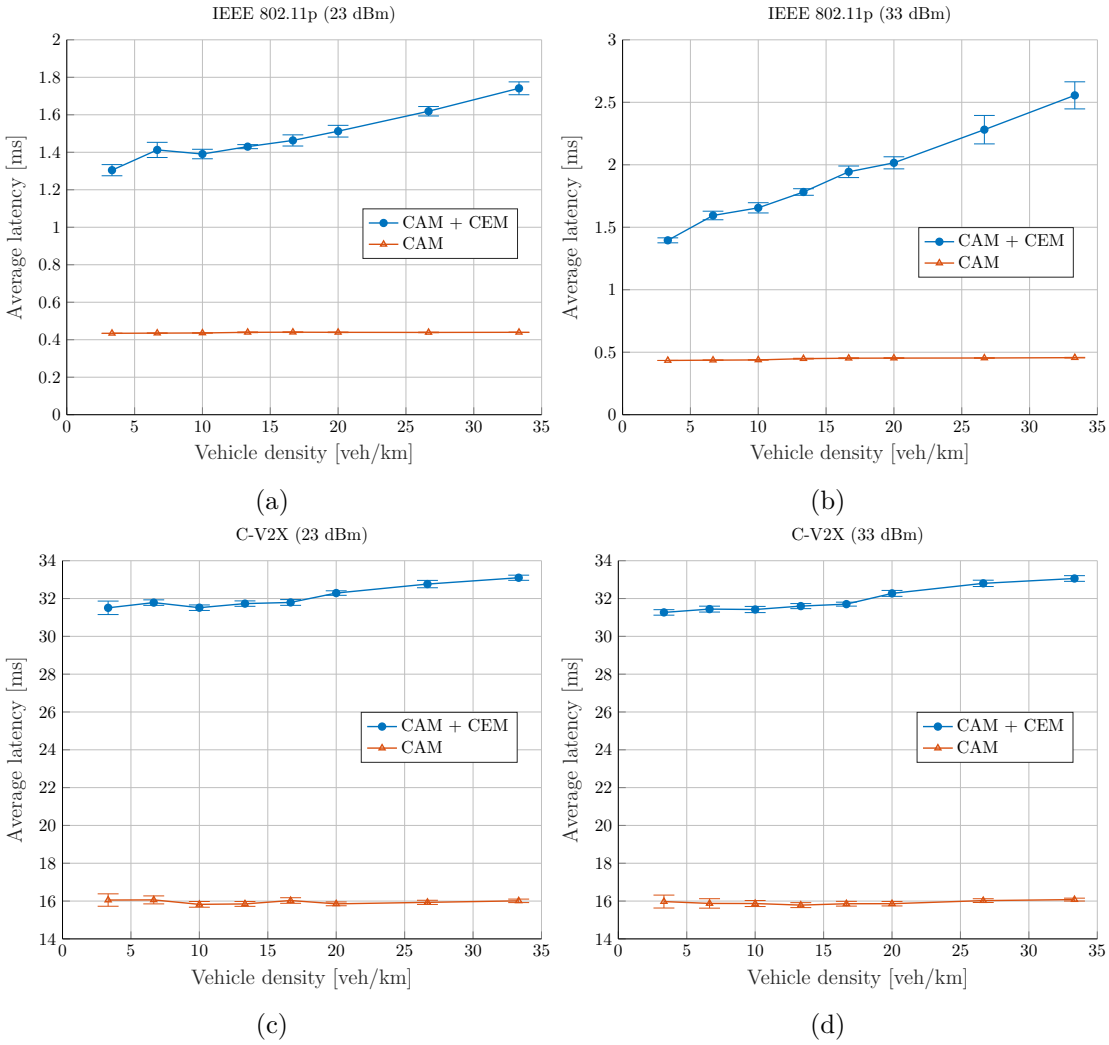


Fig. 7.7 Average one-way latency with respect to vehicle density. (a) IEEE 802.11p at 23 dBm (b) IEEE 802.11p at 33 dBm (c) C-V2X at 23 dBm (d) C-V2X at 33 dBm.

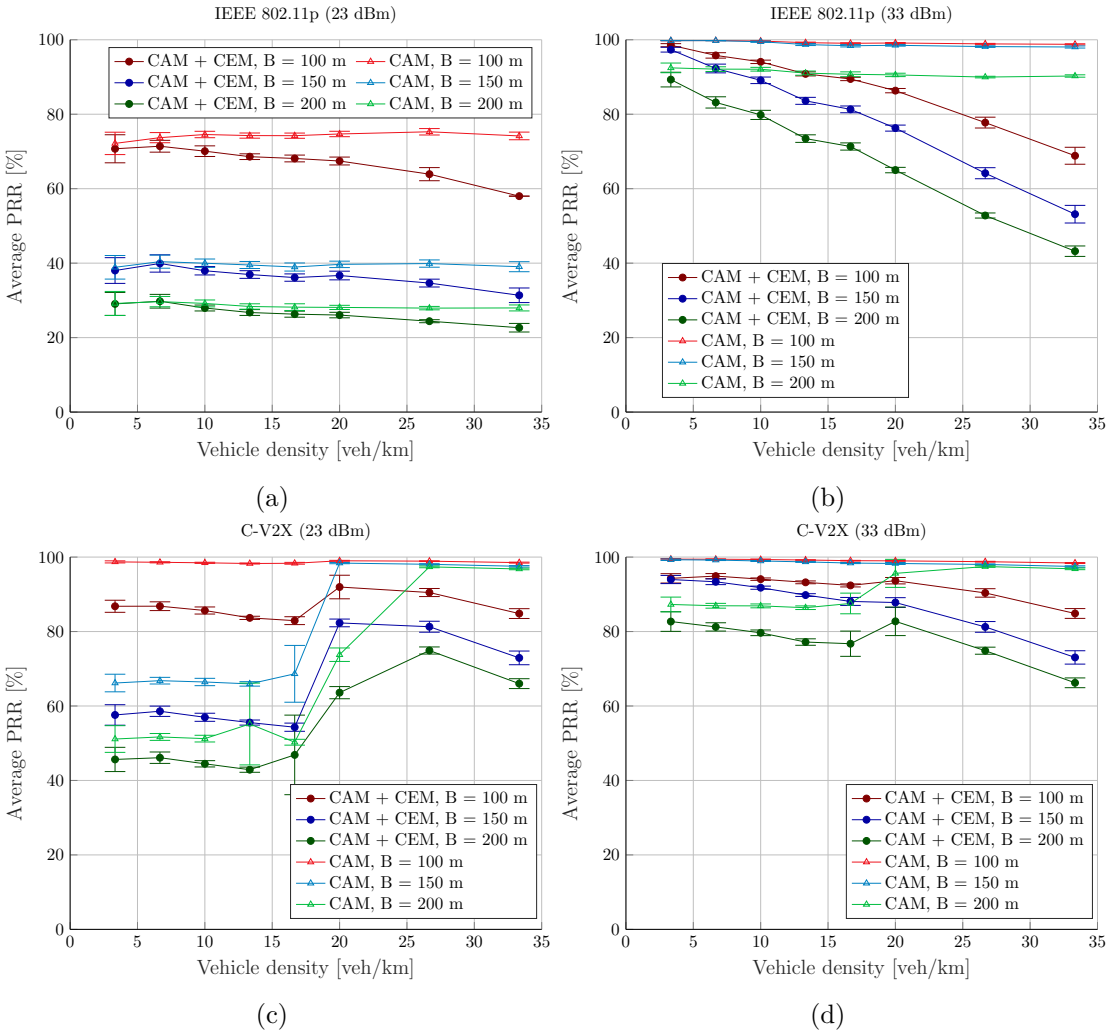
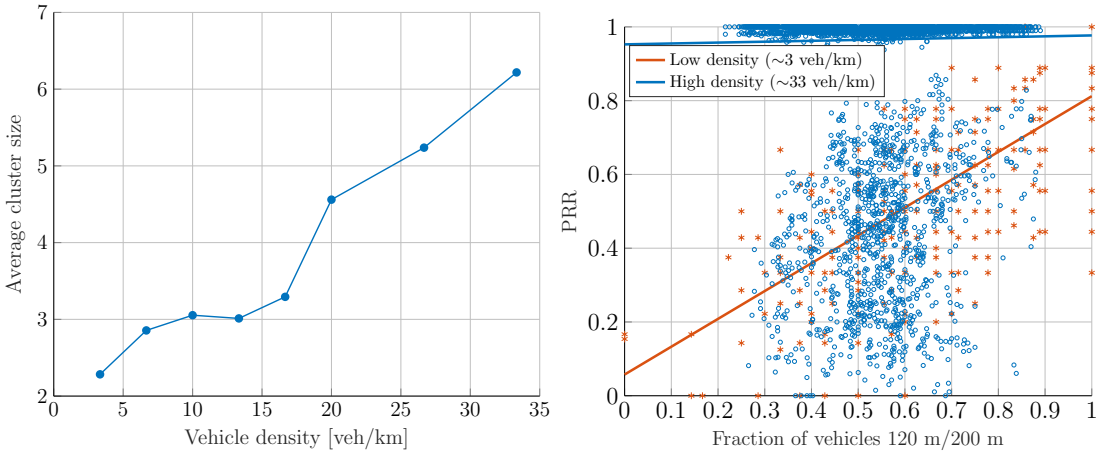


Fig. 7.8 Packet Reception Ratio, in percentage as a function of vehicle density, for three different values of baseline distance, i.e., 100 m, 150 m and 200 m. (a) IEEE 802.11p at 23 dBm (b) IEEE 802.11p at 33 dBm (c) C-V2X at 23 dBm (d) C-V2X at 33 dBm.



(a) (b)

Fig. 7.9 Analysis of the C-V2X Mode 4 PRR behaviour. Reference of a traffic pattern in urban scenario, a transmission power of 23 dBm and a baseline of 200m. (a) Mean vehicle cluster size (b) PRR, for each transmitted packet, as a function of the fraction of vehicles within 120 m, w.r.t. the total amount within the 200 m baseline; the straight lines show a linear interpolation of the values.

- A radio interface (including the MAC and physical layers) either based on IEEE 802.11p or on C-V2X Mode 4.

All the CAMs and CEMs are broadcasted after being encapsulated into the BTP and GeoNetworking layers, following the rules defined in Section 7.4.
Indeed, to evaluate our proposed protocol in different operational conditions, we consider two access technologies for V2X communications: *i)* IEEE 802.11p, based on Wi-Fi and well-established for V2I and V2V communications, and *ii)* C-V2X, which has emerged in recent years as a promising cellular-based alternative. The two protocols are different in all layers of the communication stack, and their pros and cons have long been under discussion in the scientific community [272–274]. Both technologies are fully supported by ms-van3t-CAM2CEM, through state-of-the-art models.
Further, C-V2X supports two main modes of operation: Mode 3, with infrastructure support, and Mode 4, for direct communication without infrastructure support. In our analysis, we focus on the latter, as it is supported in ms-van3t and allows us to obtain a direct comparison against IEEE 802.11p while exchanging CEM in V2V scenarios.
The performance metrics of interest are:

- The total transmission rate, expressed as the number of bytes per second transmitted in the wireless medium by vehicles as their density grows larger;
- The average one-way latency between a transmitting vehicle and all the vehicles receiving the message (we recall that both CAMs and CEMs are normally broadcasted);
- The Packet Delivery Ratio (PDR), also called PRR by 3GPP (PRR will be used to refer to such metric). The value has been computed according to 3GPP TR 36.885 specifications [275], which reports a way to compute the overall PRR, when packets are broadcasted.

Specifically, two transmission power levels are considered: 23 dB m, as it represents a typical value for the exchange of vehicular messages [276], and 33 dB m. The results are obtained for an increasing number of vehicles, from 3.4 veh/km up to a density of about 33 veh/km.

Total Transmission Rate The first set of results is related to the total transmission rate as the vehicle density increases. These results do not depend upon the actual access technology and show how much the wireless medium is potentially used when transmitting both CAM and CEM in a urban scenario.
The obtained values are depicted in Figure 7.6, in which the y axis represents the total transmission rate, in Mbit/s, computed over all vehicles in the considered scenario.
The first important outcome is that the relationship between the total transmission rate and the vehicle density is linear, which shows how the communication complexity of the proposed protocol is relatively low. This is due to CEM being mainly designed to be broadcasted by vehicles.
A second relevant outcome is represented by the maximum transmission rate, namely 4.1 Mbit/s, that is achieved under very high vehicle density. As this is a fairly low value, this result confirms that the proposed solution can work well even when the network is congested and the available throughput is limited.

Latency The latency results are instead depicted in Figure 7.7. It should be noticed that these results do not depend on the value of baseline distance, which is therefore use only for the computation of the PRR.
As can be seen, IEEE 802.11p can provide an overall average latency which is significantly lower than the one provided by C-V2X, even for a relatively large vehicle

density. In particular, the IEEE-based access technology can provide a latency of around 0.4 ms when only CAMs are transmitted. On the other hand, when both CAMs and CEMs are used, the latency increases to, on average, 1.3 ms-2.5 ms (depending on the vehicle density). This is due to the larger packet size of CEM, compared to standard CAM. Indeed, in our simulation, the size of the CEMs can range from 496 B (smallest D frames) to 860 B (largest I frames), while CAMs reach up to a maximum of 121 B. However, despite the increased latency, the delays remain reasonably low and compatible with all the safety-critical applications which may require, according to ETSI, a maximum end-to-end latency between 50 ms and 300 ms (depending on the specific application) [277–279].
The latency observed under C-V2X is instead higher, with values around 16 ms for the transmission of CAM only, and 32 ms in the case of concurrent transmission of CAMs and CEMs. Such increase in latency is again due to the larger size of the CEM messages. As the amount of data required to exchange raw GNSS information is higher than what is normally transmitted within CAMs, a latency increase also shows the importance of adopting a differential encoding to reduce the impact of GNSS raw data on the underlying network as much as possible. These results are also in accordance with past simulation studies comparing the performance of C-V2X and IEEE 802.11p [273]. It is also worth noticing that increasing the transmission power does not impact the latency significantly, but it leads to a faster increase as the vehicle density grows for IEEE 802.11p, due to higher interference levels. This effect is not observed in the case of C-V2X, which suffers less severely from highly congested network conditions. Thus, when transmitting CEMs over C-V2X, the latency remains almost constant, regardless of the selected transmission power level. This suggests that a higher transmission power would be desirable, as it has a minimal impact on latency, while guaranteeing a higher PRR, as described below.

Packet Reception Ratio The PRR represents a measure of network reliability, as it accounts for the number of packets that are lost due to channel collisions, interference, or harsh propagation conditions.
Given a baseline distance, which varies from 100 m to 200 m, the PRR is calculated as the ratio between the number of vehicles X within the baseline that successfully receive the message, and the total number of vehicles Y within the same distance. Then, the per-message PRR is averaged over all the transmissions (n_{tx}) to compute

the average PRR, given a certain value of baseline distance:

$$PRR_i = \frac{X_i}{Y_i},\ i = 1, ..., n_{tx} \tag{7.1}$$

For each access technology and scenario, results are averaged over 10 different experiments, each corresponding to a randomly selected traffic pattern. Each plot also reports the 95% confidence intervals.

In Figure 7.8, each plot depicts the average PRR as the vehicle density grows. The lines corresponding to the transmission of CAMs only are marked by triangles, while those corresponding to the concurrent transmission of both protocols are marked by circles. Each color corresponds to a different baseline value (i.e., 100 m, 100 m and 100 m).

The first interesting result comes from the trend of the PRR as the vehicle density grows larger, since different access technologies exhibit different behaviours for highly congested scenarios. Indeed, for IEEE 802.11p the PRR decreases monotonically with the vehicle density, which is especially evident when transmitting both protocols. This is due to the contention-based channel access, which suffers from increased collisions and back off times when a large number of vehicles try to access the same shared wireless medium.

Interestingly, when C-V2X Mode 4 is used with a transmission power of 23 dB m, there is a slight reduction until 16.67 vehicle/km, followed by a sudden increase for higher densities. Considering the baseline of 200 m as a reference, the *CAMs + CEMs* PRR increases from 46.86% with 16.67 veh/km to 63.58% with 20 veh/km. Such effect is less evident for higher transmitted power.

The observed trend is mainly due to the resilience of C-V2X when there is a large number of communicating nodes. Simulating the same scenarios using a very high transmission power (i.e., 42 dB m not shown here as it is an unrealistically high value) led to PRR values always around 100% for both scenarios. The behaviour of PRR when the vehicle density increases is thus mainly due to the selected level of transmission power, which may not be sufficient to reach all the vehicles within the desired baseline. This also explains why this effect is much less noticeable with high transmission power levels (i.e., 33 dB m), or, equivalently, smaller baseline values (i.e., 100 m).

With these considerations in mind, it is possible to observe how vehicles tend to form clusters in urban scenarios. This occurs especially in correspondence of regulated intersections. Figure 7.9a depicts the mean cluster size as a function of the vehicle density, taking as reference one of the ten traffic patterns used to gather the PRR

results. A cluster is defined here as a group of vehicles with a reciprocal distance lower than 20 m. As can be seen, the mean cluster size almost doubles between densities of 3 veh/km and 33 veh/km. As a consequence, the number of vehicles within the baseline that can be reached by C-V2X Mode 4 grows larger.
Instead, Figure 7.9b shows the relationship between the PRR values and the ratio between the number of vehicles within 120 m and the total number within a 200 m baseline. As a reference, a transmission power of 23 dB m is considered, as the C-V2X PRR increase is more evident for low transmission power levels. The dependence between the PRR and the fraction of vehicles is significantly less when high density scenarios are considered, with higher values of PRR becoming more likely even when the fraction of vehicles is relatively low.
Next, we look at the comparison between the PRR values obtained with the two different transmission powers. As expected, a transmission power higher than 23 dB m is always desirable, as it provides higher values of PRR in all cases and under any of the selected technologies. Indeed, 23 dB m is not always sufficient to reach a transmission range greater than 100 m, thus leading to low PRR for larger baseline values. Even though a range of 100 m may be enough for enabling several CP approaches, it can be important to take this into account, especially when targeting use cases which may require larger single-hop ranges.
It can also be observed that the C-V2X technology provides, better PRR results overall compared to IEEE 802.11p, especially when the vehicle density grows. Indeed, C-V2X with 33 dB m can provide a PRR always higher than 77% for vehicle densities up to 16.67 veh/km (which represents a quite congested scenario for most CP approaches). Furthermore, such transmission power can guarantee an average PRR higher than 66%, even for a density of 33.4 veh/km and a large baseline of 200 m.
Comparing the performance when only CAM is used and when both protocols are transmitted, it can be noticed that the use of CEM slightly reduces the overall PRR, due to the larger size of messages and of their more frequent transmission. However, this reduction is limited, especially in the presence of mid-to-low vehicle densities, and does not significantly impact the CP use cases enabled by the proposed protocol. In particular, a range of 100 m with a PRR higher than 90% can enable the development and testing of different cooperative approaches for high-precision localisation.
Furthermore, focusing on the highest level of transmission power level and the exchange of both protocols, IEEE 802.11p showcases a PRR much lower compared to C-V2X, especially when the number of vehicles increases. Once again, this is due to the contention-based nature of the IEEE 802.11p access technology.

7.6 Conclusions and Future Works

The analysis of results regarding the PRR also provides useful insights on how the CEM protocol specifications can be improved in the near future. In particular, an updated version of the protocol should include:

- The possibility of transmitting the *I* and *D* frames at varying periods, depending on the actual target CP application, as mentioned in Section 7.4.2. This may help to improve the overall PRR, as less *D* frames would be transmitted if the vehicles are implementing CP applications which do not require high frequency of data; this would also help to reduce the likelihood of concurrent CAM and/or CEM transmission from different vehicles;
- The possibility of dynamically reducing the number of *D* frames in case of detected channel congestion;
- A Decentralized Congestion Control (DCC) mechanism to automatically adjust the transmission power depending on the real-time PRR performance and on the channel congestion. The same mechanism could be realized in an agnostic way with respect to the underlying access technology, which would just need to tune the maximum transmission power depending on the inputs from a CEM module.

To conclude, our results highlight how the CEM protocol is an effective way of exchanging raw GNSS data between connected vehicles, using a fully open protocol. Further, the proposed protocol proved to be well-suited to work with both the IEEE 802.11p and C-V2X technologies, depending on the latency and reliability requirements of the underlying applications. Results have also demonstrated that, in order to reach all vehicles within a range of at least 100 m with a reliable PRR, a transmission power of 23 dB m may not be sufficient. On the contrary, 23 dB m can guarantee a PRR higher than 90% at a 100 m distance for up to 16.67 veh/km, which represents a reasonably high density for most CP approaches. We have also provided an insight on how IEEE 802.11p yields a better latency when it is used to transmit CEMs to nearby vehicles, while C-V2X showed to be more resilient to channel congestion. Finally, the obtained results highlighted which enhancements would be utmost beneficial to further improve the performance of a protocol for GNSS raw data transfer.
The work on CEM aims at contributing to ongoing standardization efforts in ETSI

ITS-G5, where the exchange of raw GNSS data could be leveraged as a fundamental enabler for next-generation CP applications.
In conclusion, future work should also focus on investigating techniques for the assignment and management of vehicles IDs, as they play a crucial role in the implementation of numerous CP algorithms.

Chapter 8

Conclusions and Future Work

In conclusion, this thesis has addressed various aspects of GNSS with the intent to tackle different challenges faced by radio navigation. After giving a brief introduction of the main concepts of GNSS, Chapter 3 provided an in-depth analysis of the PVT estimation by means of LS, as well as examining some theoretical aspects involved in the estimation such as residuals and GDOP. The insights gained from such study also helped guiding the work presented in following Chapters. In particular, Chapter 4 introduced a novel weighting strategy in the PF devised exploiting the specific structure of the measurement model of GNSS. The technique, called MW-PF, helps reaching any target accuracy using a lower amount of particles w.r.t. standard implementations of the filter. Heavy computational complexity and the so-called curse of dimensionality have been one of the main limitations for the use Monte Carlo methods in real scenarios. Hopefully, optimizations such as MW-PF can help to make Monte Carlo methods a more viable filtering alternative in the future, possibly coupled with the increase in computational capacity of modern hardware.
Subsequently, the problem of measurement biases due to reflection of signals was considered in Chapter 5. Two main contributions were presented, the first is a post-processing methodology to automatically label biased measurements affected by faults in datasets, and the second aimed at mitigating the effect of such biases for real-time applications by exploiting the concept of soft information in the PF.
Despite numerous effort to mitigate the effect of multipath and NLoS, they remain one of the main limitations preventing the use of stand-alone GNSS for applications with strict requirements in terms of accuracy and integrity. One of the main solutions to these limitation is to couple GNSS with other sensors to combine the strengths of each individual instrument and mitigate their respective weaknesses. For this reason,

Chapter 6 tackled the tight integration between GNSS and UWB, and more in particular one of the main challenges of this architecture which is the synchronization of measurements from the two sensors. A novel technique based on a double-update step of the EKF was presented, showing promising results in terms of accuracy.
An alternative solution to improve the performance of stand-alone GNSS is the concept of CP, namely to integrate additional location information exchanged among a network of peers. This can include ranges computed with other sensors like UWB, but can also be obtained by combining GNSS observables from different receivers using difference techniques. While the advantages of cooperation in terms of estimation performance have been thoroughly investigated in the literature, its feasibility from the network perspective has often been neglected. As a matter of fact, many works simply assume that cooperative information is always available, while disregarding possible limitations and disruptions such as network delays and packet losses. To address this point, the final novel contribution presented in the thesis was the development and experimental testing of an open-source protocol for the real-time exchange of raw observables between cooperating agents, thus proving the viability of CP in real-world environment.

8.1 Future Work

Because of their ability to handle non-linearities without loss of performance, Monte Carlo methods such as PF have proven to be able to provide greater accuracy w.r.t. to other Bayesian filters such as EKF in a wide range of scenarios, at the cost of increased computational load. As a consequence, future studies should continue to focus on computational optimizations to make PF more viable for real-world applications. Furthermore, other novel techniques can be devised to exploit one of its main advantage, which is the capability of working with any arbitrary probability distribution as measurement model. In fact, while some work has already been presented in this thesis to mitigate the effect of outliers due to multipath and NLoS, there are still improvements that can be done in mitigating the effect of biased measurements and outliers in general, regardless of their source. Therefore, improvements of the proposed technique and investigation of other methodologies to build multi-modal densities will continue in the future, as also detailed in Section 5.3.6.
Another promising approach is to apply the concept of soft information to other decision stages of the receiver. The work presented in the thesis embraced this

idea by building multi-modal probability densities to be used in the measurement model of the PF, thus representing knowledge about the measurements in a soft way rather than committing on a single value. However, the decision on whether to employ such multi-modal densities still depended on a hard threshold set on the innovation computed by the PF. To generalize this idea, a GNSS receiver is made of multiple stages which, starting from a continuous radio signal, attempt to estimate some parameters from such signal. During each of these stages, whenever a decision is takes, certain amounts of information are discarded (e.g., only the peak of the correlation is considered during acquisition). The idea is therefore to apply the concept of soft information to other stages as well, thus retaining more of the information carried by the original signal.

Another line of research that deserves further investigation is that of GNSS/UWB tight integration for applications such as drones and fleets of vehicle. Results have shown that UWB is a suitable companion to GNSS in the sense that it can mitigate some of its weaknesses. Following studies could focus on investigating the placement of anchors in order to optimize the geometrical improvement given by the additional UWB range and minimize the impact of unmodelled dynamics on time synchronization.

Besides continuing research on topics presented in this thesis, future work will also focus on newer or emerging applications such positioning with MEO constellations and the use of GNSS in space.

Regarding the former, early studies have shown promising performance of Doppler-based positioning, but have also highlighted challenges related to the convergence of standard LS methods due to higher non-linearities. Recent work has been done to investigate these aspects by leveraging the analysis provided in Chapter 3, but research efforts will continue in the future to study and characterize the positioning capabilities of MEO.

Due to increased interest in space exploration, the exploitation of GNSS-based navigation technologies for future missions has been considered. First and foremost, future missions will be a great scientific opportunity to assess the impact of phenomena such as relativistic effects on radio navigation, as well as investigate the possible differences between earth-surface and space of other effects such as ionospheric delay. Secondly, future attention should focus on methodologies to overcome the limitations of GNSS in space, namely low received signal power and poor geometry. To this aim, a promising approach is to use trajectories known in advance to be integrated in the PVT with the scope of aiding the performance of on-board filters. Such work

is ongoing and has already been partially investigated but will also continue to be improved in the future.

References

[1] Simone Zocca, Yihan Guo, Alex Minetto, and Fabio Dovis. Improved weighting in particle filters applied to precise state estimation in GNSS. *Frontiers in Robotics and AI*, 9:950427, 2022.

[2] Francesco Raviglione, Simone Zocca, Alex Minetto, Marco Malinverno, Claudio Casetti, Carla Fabiana Chiasserini, and Fabio Dovis. From collaborative awareness to collaborative information enhancement in vehicular networks. *Vehicular Communications*, 36:100497, 2022.

[3] Alex Minetto, Simone Zocca, Francesco Raviglione, Marco Malinverno, Claudio Ettore Casetti, Carla-Fabiana Chiasserini, and Fabio Dovis. Cooperative localization enhancement through GNSS raw data in vehicular networks. In *2021 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2021.

[4] Luca Morichi, Alex Minetto, Andrea Nardin, Simone Zocca, and Fabio Dovis. Pseudorange and doppler-based state estimation from meo to leo: A comprehensive analysis of maximum likelihood estimators. In *Proceedings of the 2024 International Technical Meeting of The Institute of Navigation*, pages 677–691, 2024.

[5] Yihan Guo, Simone Zocca, Paolo Dabove, and Fabio Dovis. A post-processing multipath/nlos bias estimation method based on dbscan. *Sensors*, 24(8):2611, 2024.

[6] Simone Zocca, Yihan Guo, and Fabio Dovis. A gnss multipath and nlos mitigation method for urban scenarios based on particle filtering. In *Proceedings of the 2024 International Technical Meeting of The Institute of Navigation*, pages 575–588, 2024.

[7] Yihan Guo, Oliviero Vouch, Simone Zocca, Alex Minetto, and Fabio Dovis. A comparison study between the ekf and sir-pf for gnss/uwb tight integration. In *2023 31st European Signal Processing Conference (EUSIPCO)*, pages 835–839. IEEE, 2023.

[8] Oliviero Vouch, Yihan Guo, Simone Zocca, Alex Minetto, and Fabio Dovis. Improved outdoor target tracking via ekf-based gnss/uwb tight integration with online time synchronisation. In *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, pages 2409–2422, 2022.

[9] Yihan Guo, Oliviero Vouch, Simone Zocca, Alex Minetto, and Fabio Dovis. Enhanced ekf-based time calibration for GNSS/UWB tight integration. *IEEE Sensors Journal*, 23(1):552–566, 2022.

[10] Paul D Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.

[11] Peter JG Teunissen and Oliver Montenbruck. *Springer handbook of global navigation satellite systems*, volume 10. Springer, 2017.

[12] Elliott D Kaplan and Christopher Hegarty. *Understanding GPS/GNSS: principles and applications*. Artech house, 2017.

[13] Bradford W Parkinson and James J Spilker. *Progress in astronautics and aeronautics: Global positioning system: Theory and applications*, volume 164. Aiaa, 1996.

[14] Pratap Misra and Per Enge. Global positioning system: Signals. *Measurements and Performance,*, pages 381–384, 2006.

[15] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins. *Global positioning system: theory and practice*. Springer Science & Business Media, 2012.

[16] Zheng Yao and Mingquan Lu. *Next-Generation GNSS Signal Design*. Springer, 2021.

[17] Daniele Borio. A statistical theory for GNSS signal acquisition. *PHD Dissertation Polytecnico di Torino*, 2008.

[18] Mansfield Merriman. On the history of the method of least squares. *The Analyst*, 4(2):33–36, 1877.

[19] Robin L Plackett. Studies in the history of probability and statistics. xxix: The discovery of the method of least squares. *Biometrika*, 59(2):239–251, 1972.

[20] Stephen M Stigler. Gauss and the invention of least squares. *the Annals of Statistics*, pages 465–474, 1981.

[21] Carl Friedrich Gauss and GW Stewart. *Theory of the combination of observations least subject to errors, Part One, Part Two, Supplement*. SIAM, 1995.

[22] Yves Nievergelt. A tutorial history of least squares with applications to astronomy and geodesy. *Journal of Computational and Applied Mathematics*, 121(1-2):37–72, 2000.

[23] Jerry M Mendel. *Lessons in estimation theory for signal processing, communications, and control*. Pearson Education, 1995.

[24] Steven M Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[25] Gilbert W Stewart. *Matrix algorithms: volume 1: basic decompositions*. SIAM, 1998.

[26] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[27] Stephen Boyd and Lieven Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.

[28] Gilbert Strang. *Linear algebra and learning from data*. SIAM, 2019.

[29] Gilbert Strang. *Introduction to linear algebra*. SIAM, 2022.

[30] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.

[31] Calyampudi Radhakrishna Rao, Helge Toutenburg, Calyampudi Radhakrishna Rao, and Helge Toutenburg. *Linear models*. Springer, 1995.

[32] Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[33] Per Christian Hansen, Victor Pereyra, and Godela Scherer. *Least squares data fitting with applications*. JHU Press, 2013.

[34] Philip E Gill, Walter Murray, and Margaret H Wright. Practical optimization. *London: Academic Press*, 1981.

[35] John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

[36] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. *Methods for non-linear least squares problems*. 2004.

[37] Serge Gratton, Amos S Lawless, and Nancy K Nichols. Approximate gauss–newton methods for nonlinear least squares problems. *SIAM Journal on Optimization*, 18(1):106–132, 2007.

[38] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[39] Lloyd N Trefethen and David Bau. *Numerical linear algebra*, volume 181. Siam, 2022.

[40] Walter Gander, Martin J Gander, and Felix Kwok. *Scientific computing-An introduction using Maple and MATLAB*, volume 11. Springer Science & Business, 2014.

[41] Gene Howard Golub, Virginia Klema, and Gilbert W Stewart. *Rank degeneracy and least squares problems*. Stanford Univ., 1976.

[42] Gilbert W Stewart. On the perturbation of pseudo-inverses, projections and linear least squares problems. *SIAM review*, 19(4):634–662, 1977.

[43] Gilbert W Stewart. Collinearity and least squares regression. *Statistical Science*, 2(1):68–84, 1987.

[44] Åke Björck. Solving linear least squares problems by gram-schmidt orthogonalization. *BIT Numerical Mathematics*, 7(1):1–21, 1967.

[45] Å Björck. Algorithms for linear least squares problems. In *Computer Algorithms for Solving Linear Algebraic Equations: The State of the Art*, pages 57–92. Springer, 1991.

[46] Harold W Sorenson. Least-squares estimation: from gauss to kalman. *IEEE spectrum*, 7(7):63–68, 1970.

[47] Robin L Plackett. A historical note on the method of least squares. *Biometrika*, 36(3/4):458–460, 1949.

[48] Alexander C Aitken. Iv.—on least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1936.

[49] Allen J Pope. *The statistics of residuals and the detection of outliers*. US Department of Commerce, National Oceanic and Atmospheric Administration . . . , 1976.

[50] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[51] Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

[52] Mark L Psiaki. Navigation using carrier doppler shift from a leo constellation: Transit on steroids. *Navigation*, 68(3):621–641, 2021.

[53] Chuang Shi, Yulu Zhang, and Zhen Li. Revisiting doppler positioning performance with leo satellites. *GPS Solutions*, 27(3):126, 2023.

[54] Boaz Porat and Benjamin Friedlander. Computation of the exact information matrix of gaussian time series with stationary random components. *IEEE transactions on acoustics, speech, and signal processing*, 34(1):118–130, 1986.

[55] John Robert Taylor and William Thompson. *An introduction to error analysis: the study of uncertainties in physical measurements*, volume 2. Springer, 1982.

[56] Dan Margalit, Joseph Rabinoff, and Larry Rolen. Interactive linear algebra. *Georgia Institute of Technology*, 2017.

[57] Bradford W Parkinson and Penina Axelrad. Autonomous gps integrity monitoring using the pseudorange residual. *Navigation*, 35(2):255–274, 1988.

[58] Ziyi Jiang, Paul D Groves, Washington Y Ochieng, Shaojun Feng, Carl D Milner, and Philip G Mattos. Multi-constellation GNSS multipath mitigation using consistency checking. In *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, pages 3889–3902, 2011.

[59] Calyampudi Radhakrishna Rao and Mathematischer Statistiker. *Linear statistical inference and its applications*, volume 2. Wiley New York, 1973.

[60] Douglas S Shafer and Zhiyi Zhang. *Introductory statistics*. Saylor Foundation Washington, DC, 2012.

[61] Roxy Peck, Chris Olsen, and Jay L Devore. *Introduction to statistics and data analysis*. Cengage Learning, 2015.

[62] R Dennis Cook. Influential observations in linear regression. *Journal of the American Statistical Association*, 74(365):169–174, 1979.

[63] Samprit Chatterjee and Ali S Hadi. Influential observations, high leverage points, and outliers in linear regression. *Statistical science*, pages 379–393, 1986.

[64] Richard B Langley et al. Dilution of precision. *GPS world*, 10(5):52–59, 1999.

[65] Miaoyan Zhang and Jun Zhang. A fast satellite selection algorithm: beyond four satellites. *IEEE Journal of Selected Topics in Signal Processing*, 3(5):740–747, 2009.

[66] R Yarlagadda, I Ali, N Al-Dhahir, and J Hershey. Gps gdop metric. *IEE Proceedings: Radar, Sonar and Navigation*, 147(5):259–264, 2000.

[67] Peter F Swaszek, Richard J Hartnett, and Kelly C Seals. Lower bounds on dop. *The Journal of Navigation*, 70(5):1041–1061, 2017.

[68] Frank Stephen Tromp Van Diggelen. *A-gps: Assisted gps, gnss, and sbas*. Artech house, 2009.

[69] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[70] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[71] Maria Isabel Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43:46, 2004.

[72] James Taylor. The cramer-rao estimation error lower bound computation for deterministic nonlinear systems. *IEEE Transactions on Automatic Control*, 24(2):343–344, 1979.

[73] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[74] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2003.

[75] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[76] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

[77] Arnaud Doucet, Adam M Johansen, et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[78] Jos Elfring, Elena Torta, and René van de Molengraft. Particle filters: A hands-on tutorial. *Sensors*, 21(2):438, 2021.

[79] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002.

[80] Fredrik Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, 2010.

[81] Stuart J Russell and Peter Norvig. *Artificial intelligence a modern approach*. London, 2010.

[82] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

[83] Dieter Fox, Sebastian Thrun, Wolfram Burgard, and Frank Dellaert. Particle filters for mobile robot localization. In *Sequential Monte Carlo methods in practice*, pages 401–428. Springer, 2001.

[84] Sebastian Thrun. Particle filters in robotics. In *UAI*, volume 2, pages 511–518. Citeseer, 2002.

[85] Cody Kwok, Dieter Fox, and Marina Meila. Adaptive real-time particle filters for robot localization. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 2836–2841. IEEE, 2003.

[86] Joao FG de Freitas, Mahesan Niranjan, Andrew H. Gee, and Arnaud Doucet. Sequential monte carlo methods to train neural network models. *Neural computation*, 12(4):955–993, 2000.

[87] Xiao Ma, Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5101–5108, 2020.

[88] David L Poole and Alan K Mackworth. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

[89] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[90] Rubin DB. Using the sir algorithm to simulate posterior distributions. In *Bayesian statistics 3. Proceedings of the third Valencia international meeting, 1-5 June 1987*, pages 395–402. Clarendon Press, 1988.

[91] Miodrag Bolić, Petar M Djurić, and Sangjin Hong. Resampling algorithms for particle filters: A computational complexity perspective. *EURASIP Journal on Advances in Signal Processing*, 2004:1–11, 2004.

[92] Jeroen D Hol, Thomas B Schon, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *2006 IEEE nonlinear statistical signal processing workshop*, pages 79–82. IEEE, 2006.

[93] Tiancheng Li, Miodrag Bolic, and Petar M Djuric. Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal processing magazine*, 32(3):70–86, 2015.

[94] Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. Particle filtering: the need for speed. *EURASIP Journal on Advances in Signal processing*, 2010:1–9, 2010.

[95] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10:197–208, 2000.

[96] Thomas Schon, Fredrik Gustafsson, and P-J Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on signal processing*, 53(7):2279–2289, 2005.

[97] Macheng Shen, Jing Sun, Huei Peng, and Ding Zhao. Improving localization accuracy in connected vehicle networks using rao–blackwellized particle filters: Theory, simulations, and experiments. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2255–2266, 2018.

[98] S.R. Maskell, K.R. Weekes, and M. Briers. Distributed tracking of stealthy targets using particle filters. In *The IEE Seminar on Target Tracking: Algorithms and Applications 2006 (Ref. No. 2006/11359)*, pages 11–20, 2006.

[99] Petar M Djuric, Ting Lu, and Mónica F Bugallo. Multiple particle filtering. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 3, pages III–1181. IEEE, 2007.

[100] Pau Closas and Mónica F Bugallo. Improving accuracy by iterated multiple particle filtering. *IEEE Signal Processing Letters*, 19(8):531–534, 2012.

[101] Petar M Djurić and Mónica F Bugallo. Particle filtering for high-dimensional systems. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 352–355. IEEE, 2013.

[102] Jonathan Poterjoy. A localized particle filter for high-dimensional nonlinear systems. *Monthly Weather Review*, 144(1):59–76, December 2015.

[103] Patrick Rebeschini and Ramon Van Handel. Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, 25(5):2809–2866, 2015.

[104] Samson Shenglong Yu, Junhao Guo, Tat Kei Chau, Tyrone Fernando, Herbert Ho-Ching Iu, and Hieu Trinh. An unscented particle filtering approach to decentralized dynamic state estimation for dfig wind turbines in multi-area power systems. *IEEE Transactions on Power Systems*, 35(4):2670–2682, 2020.

[105] Weihao Song, Zidong Wang, Jianan Wang, Fuad E Alsaadi, and Jiayuan Shan. Distributed auxiliary particle filtering with diffusion strategy for target tracking: A dynamic event-triggered approach. *IEEE Transactions on Signal Processing*, 69:328–340, 2020.

[106] Pau Closas and Carles Fernández-Prades. Particle filtering with adaptive number of particles. In *2011 Aerospace Conference*, pages 1–7. IEEE, 2011.

[107] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1-2):99–141, 2001.

[108] Jacques Georgy, Aboelmagd Noureldin, Michael J Korenberg, and Mohamed M Bayoumi. Low-cost three-dimensional navigation solution for riss/gps integration using mixture particle filter. *IEEE Transactions on vehicular technology*, 59(2):599–615, 2009.

[109] Samuel J Davey, Neil J Gordon, and M Sabordo. Multi-sensor track-before-detect for complementary sensors. *Digital signal processing*, 21(5):600–607, 2011.

[110] Cesare Donati, Martina Mammarella, and Fabrizio Dabbene. Single-state weighted particle filter with application to earth observation missions. *IFAC-PapersOnLine*, 56(2):6024–6029, 2023.

[111] Dimitri P Bertsekas and John N Tsitsiklis. *Introduction to probability*, volume 1. Athena Scientific, 2008.

[112] Andrea Conti, Santiago Mazuelas, Stefania Bartoletti, William C Lindsey, and Moe Z Win. Soft information for localization-of-things. *Proceedings of the IEEE*, 107(11):2240–2264, 2019.

[113] Ni Zhu, Juliette Marais, David Bétaille, and Marion Berbineau. GNSS position integrity in urban environments: A review of literature. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):2762–2778, 2018.

[114] Gary A McGraw, Paul D Groves, and Benjamin W Ashman. Robust positioning in the presence of multipath and nlos GNSS signals. *Position, navigation, and timing technologies in the 21st century: integrated satellite navigation, sensor systems, and civil applications*, 1:551–589, 2020.

[115] Ehsan Taghdisi, Mohammad Saeid Ghaffarian, and Rashid Mirzavand. Low-profile substrate integrated choke rings for GNSS multipath mitigation. *IEEE Transactions on Antennas and Propagation*, 70(3):1706–1718, 2021.

[116] Paul D Groves, Ziyi Jiang, Benjamin Skelton, Paul A Cross, Lawrence Lau, Yacine Adane, and Izzet Kale. Novel multipath mitigation methods using a dual-polarization antenna. In *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, pages 140–151, 2010.

[117] Bassma Guermah, Hassan EL GHAZI, Tayeb Sadiki, and Hatim Guermah. A robust GNSS los/multipath signal classifier based on the fusion of information and machine learning for intelligent transportation systems. In *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, pages 94–100. IEEE, 2018.

[118] Niranjana Vagle, Ali Broumandan, Ali Jafarnia-Jahromi, and Gérard Lachapelle. Performance analysis of GNSS multipath mitigation using antenna arrays. *The Journal of Global Positioning Systems*, 14(1):1–15, 2016.

[119] AJ Van Dierendonck, Pat Fenton, and Tom Ford. Theory and performance of narrow correlator spacing in a gps receiver. *Navigation*, 39(3):265–283, 1992.

[120] Bryan Townsend and Patrick Fenton. A practical approach to the reduction of pseudorange multipath errors in a l1 gps receiver. In *Proceedings of the 7th International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City, UT, USA*, pages 20–23, 1994.

[121] Lionel Garin and Jean-Michel Rousseau. Enhanced strobe correlator multipath rejection for code & carrier. In *Proceedings of the 10th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1997)*, pages 559–568, 1997.

[122] Gary A McGraw and Michael S Braasch. GNSS multipath mitigation using gated and high resolution correlator concepts. In *Proceedings of the 1999 national technical meeting of the institute of navigation*, pages 333–342, 1999.

[123] Jason Jones, Pat Fenton, Brian Smith, et al. Theory and performance of the pulse aperture correlator. In *Proceedings of ION GPS*, volume 2004, 2004.

[124] Patrick C Fenton and Jason Jones. The theory and performance of novatel inc.'s vision correlator. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, pages 2178–2186, 2005.

[125] Richard DJ Van Nee, Jaap Siereveld, Patrick C Fenton, and Bryan R Townsend. The multipath estimating delay lock loop: approaching theoretical accuracy limits. In *Proceedings of 1994 IEEE Position, Location and Navigation Symposium-PLANS'94*, pages 246–251. IEEE, 1994.

[126] Bryan R Townsend, Patrick C Fenton, Keith J Van Dierendonck, and DJ Richard Van Nee. Performance evaluation of the multipath estimating delay lock loop. *Navigation*, 42(3):502–514, 1995.

[127] Bernard H Fleury, Martin Tschudin, Ralf Heddergott, Dirk Dahlhaus, and K Ingeman Pedersen. Channel parameter estimation in mobile radio environments using the sage algorithm. *IEEE Journal on selected areas in communications*, 17(3):434–450, 1999.

[128] Lawrence R Weill. Multipath mitigation using modernized gps signals: how good can it get? In *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, pages 493–505, 2002.

[129] Mohamed Sahmoudi and Moeness G Amin. Fast iterative maximum-likelihood algorithm (fimla) for multipath mitigation in the next generation of gnss receivers. *IEEE Transactions on Wireless Communications*, 7(11):4362–4374, 2008.

[130] Xin Chen, Fabio Dovis, Marco Pini, and Paolo Mulassano. Turbo architecture for multipath mitigation in global navigation satellite system receivers. *IET Radar, Sonar & Navigation*, 5(5):517–527, 2011.

[131] Xin Chen, Fabio Dovis, Senlin Peng, and Yu Morton. Comparative studies of gps multipath mitigation methods performance. *IEEE Transactions on Aerospace and Electronic Systems*, 49(3):1555–1568, 2013.

[132] Wooseok Nam and Seung-Hyun Kong. Least-squares-based iterative multipath super-resolution technique. *IEEE Transactions on signal processing*, 61(3):519–529, 2012.

[133] Yudong Sun, Zheng Yao, and Mingquan Lu. Extended double-delta correlator technique for GNSS multipath mitigation. *IEEE Transactions on Aerospace and Electronic Systems*, 59(2):1758–1773, 2022.

[134] Yiwei Wang and Zhigang Huang. Medll on-strobe correlator: a combined anti-multipath technique for GNSS signal tracking. *The Journal of Navigation*, 73(3):658–677, 2020.

[135] Bing Xu, Qiongqiong Jia, and Li-Ta Hsu. Vector tracking loop-based GNSS nlos detection and correction: Algorithm design and performance analysis. *IEEE transactions on instrumentation and measurement*, 69(7):4604–4619, 2019.

[136] R Grover Brown. Receiver autonomous integrity monitoring. *Global Positioning System: Theory and applications.*, 2:143–165, 1996.

[137] Juan Blanch, Todd Walter, Per Enge, Young Lee, Boris Pervan, Markus Rippl, and Alex Spletter. Advanced raim user algorithm description: Integrity support message processing, fault detection, exclusion, and protection level calculation. In *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, pages 2828–2849, 2012.

[138] Mathieu Joerger and Boris Pervan. Fault detection and exclusion using solution separation and chi-squared araim. *IEEE Transactions on Aerospace and electronic systems*, 52(2):726–742, 2016.

[139] Paul D Groves and Ziyi Jiang. Height aiding, c/n0 weighting and consistency checking for GNSS nlos and multipath mitigation in urban areas. *The Journal of Navigation*, 66(5):653–669, 2013.

[140] Philip RR Strode and Paul D Groves. GNSS multipath detection using three-frequency signal-to-noise measurements. *GPS solutions*, 20:399–412, 2016.

[141] Jin Chang, Xingqun Zhan, Yawei Zhai, Shizhuang Wang, Kui Lin, and Rong Yang. Vector angle grouping-based solution separation for multipath/nlos detection and exclusion with the enhancement of doppler test. *GPS Solutions*, 26(4):121, 2022.

[142] Julien Lesouple, Thierry Robert, Mohamed Sahmoudi, Jean-Yves Tourneret, and Willy Vigneau. Multipath mitigation for gnss positioning in an urban environment using sparse estimation. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1316–1328, 2018.

[143] Julien Lesouple, Franck Barbiero, Frédéric Faurie, Mohamed Sahmoudi, and Jean-Yves Tourneret. Smooth bias estimation for multipath mitigation using sparse estimation. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1684–1690. IEEE, 2018.

[144] Taro Suzuki, Mitsunori Kitamura, Yoshiharu Amano, and Takumi Hashizume. High-accuracy gps and glonass positioning by multipath mitigation using omnidirectional infrared camera. In *2011 IEEE International Conference on Robotics and Automation*, pages 311–316. IEEE, 2011.

[145] Taro Suzuki and Nobuaki Kubo. N-los GNSS signal detection using fish-eye camera for vehicle navigation in urban environments. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, pages 1897–1906, 2014.

[146] Paul D Groves. Shadow matching: A new GNSS positioning technique for urban canyons. *The journal of Navigation*, 64(3):417–430, 2011.

[147] Li-Ta Hsu. GNSS multipath detection using a machine learning approach. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[148] Yongjun Lee and Byungwoon Park. Nonlinear regression-based GNSS multipath modelling in deep urban area. *Mathematics*, 10(3):412, 2022.

[149] Yuanxin Pan, Gregor Möller, and Benedikt Soja. Machine learning-based multipath modeling in spatial domain applied to GNSS short baseline processing. *GPS Solutions*, 28(1):9, 2024.

[150] Taro Suzuki, Yusuke Nakano, and Yoshiharu Amano. Nlos multipath detection by using machine learning in urban environments. In *Proceedings of the 30th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, pages 3958–3967, 2017.

[151] Peng Xie and Mark G Petovello. Measuring GNSS multipath distributions in urban canyon environments. *IEEE Transactions on Instrumentation and Measurement*, 64(2):366–377, 2014.

[152] Daniel Medina, Kasia Gibson, Ralf Ziebold, and Pau Closas. Determination of pseudorange error models and multipath characterization under signal-degraded scenarios. In *Proceedings of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2018*, 2018.

[153] Samer Khanafseh, Birendra Kujur, Mathieu Joerger, Todd Walter, Sam Pullen, Juan Blanch, Kevin Doherty, Laura Norman, Lance de Groot, and Boris Pervan. GNSS multipath error modeling for automotive applications. In *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, pages 1573–1589, 2018.

[154] Ni Zhu, David Betaille, Juliette Marais, and Marion Berbineau. Extended kalman filter (ekf) innovation-based integrity monitoring scheme with c/n 0 weighting. In *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, pages 1–6. IEEE, 2018.

[155] Shizhuang Wang, Xingqun Zhan, Yawei Zhai, and Baoyu Liu. Fault detection and exclusion for tightly coupled GNSS/INS system considering fault in state prediction. *Sensors*, 20(3):590, 2020.

[156] Michael S Braasch. Gps multipath model validation. In *Proceedings of Position, Location and Navigation Symposium-PLANS'96*, pages 672–678. IEEE, 1996.

[157] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[158] Tomoji Takasu. RTKLIB: An Open Source Program Package for GNSS Positioning. https://www.rtklib.com/rtklib.htm, 2020.

[159] Li-Ta Hsu, Nobuaki Kubo, Weisong Wen, Wu Chen, Zhizhao Liu, Taro Suzuki, and Junichi Meguro. Urbannav: An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, pages 226–256, 2021.

[160] Sampo Kuutti, Saber Fallah, Konstantinos Katsaros, Mehrdad Dianati, Francis Mccullough, and Alexandros Mouzakitis. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*, 5(2):829–846, 2018.

[161] Y Jade Morton, Frank van Diggelen, James J Spilker Jr, Bradford W Parkinson, Sherman Lo, and Grace Gao. *Position, navigation, and timing technologies in the 21st century: Integrated satellite navigation, sensor systems, and civil applications, volume 1*. John Wiley & Sons, 2021.

[162] Mohinder S Grewal, Angus P Andrews, and Chris G Bartone. *Global navigation satellite systems, inertial navigation, and integration*. John Wiley & Sons, 2020.

[163] Reza Zekavat and R Michael Buehrer. *Handbook of position location: Theory, practice and advances*, volume 27. John Wiley & Sons, 2011.

[164] Zafer Sahinoglu, Sinan Gezici, and Ismail Güvenc. *Ultra-wideband positioning systems: theoretical limits, ranging algorithms, and protocols*. Cambridge university press, 2008.

[165] Saeed Asadi Bagloee, Madjid Tavana, Mohsen Asadi, and Tracey Oliver. Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. *Journal of modern transportation*, 24(4):284–303, 2016.

[166] Gia Minh Hoang, Benoît Denis, Jérôme Härri, and Dirk Slock. Bayesian fusion of GNSS, its-g5 and ir–UWB data for robust cooperative vehicular localization. *Comptes Rendus Physique*, 20(3):218–227, 2019.

[167] Glenn MacGougan, Kyle O'Keefe, and Richard Klukas. Accuracy and reliability of tightly coupled GPS/Ultra-Wideband positioning for surveying in urban environments. *GPS solutions*, 14(4):351–364, 2010.

[168] Bing Zhu, Xiaowen Tao, Jian Zhao, Min Ke, Huaji Wang, and Weiwen Deng. An integrated GNSS/UWB/DR/VMM positioning strategy for intelligent vehicles. *IEEE Transactions on Vehicular Technology*, 69(10):10842–10853, 2020.

[169] Wei Jiang, Zhuojian Cao, Baigen Cai, Binghao Li, and Jian Wang. Indoor and outdoor seamless positioning method using UWB enhanced multi-sensor tightly-coupled integration. *IEEE Transactions on Vehicular Technology*, 70(10):10633–10645, 2021.

[170] Mark G Petovello, Kyle O'Keefe, Billy Chan, Stephanie Spiller, Cyril Pedrosa, and Chaminda Basnayake. Demonstration of inter-vehicle UWB ranging to augment DGPS for improved relative positioning. In *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2010)*, pages 1198–1209, 2010.

[171] Jason N Gross, Yu Gu, and Brandon Dewberry. Tightly-coupled GPS/UWB-ranging for relative navigation during formation flight. In *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, pages 1698–1708, 2014.

[172] Yang Gao, Hao Jing, Mehrdad Dianati, Craig Matthew Hancock, and Xiaolin Meng. Performance analysis of robust cooperative positioning based on GPS/UWB integration for connected autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 2022.

[173] Zhenchuan Huang, Shuanggen Jin, Ke Su, and Xu Tang. Multi-GNSS Precise Point Positioning with UWB Tightly Coupled Integration. *Sensors*, 22(6):2232, 2022.

[174] David S Chiu, Glenn MacGougan, and Kyle O'Keefe. UWB assisted GPS RTK in hostile environments. In *Proceedings of the 2008 National Technical Meeting of The Institute of Navigation*, pages 532–542, 2008.

[175] Glenn MacGougan, Kyle O'Keefe, and Richard Klukas. Tightly-coupled GPS/UWB integration. *The Journal of Navigation*, 63(1):1–22, 2010.

[176] Changqiang Wang, Aigong Xu, Xin Sui, Yushi Hao, Zhengxu Shi, and Zhijian Chen. A Seamless Navigation System and Applications for Autonomous Vehicles Using a Tightly Coupled GNSS/UWB/INS/Map Integration Scheme. *Remote Sensing*, 14(1):27, 2021.

[177] Jan Bolting and Soheib Fergani. The iterated extended set membership filter applied to relative localization between autonomous vehicles based on GNSS and UWB ranging. *Asian Journal of Control*, 21(4):1556–1565, 2019.

[178] Qinglin Tian, I Kevin, Kai Wang, and Zoran Salcic. A low-cost ins and UWB fusion pedestrian tracking system. *IEEE Sensors Journal*, 19(10):3733–3740, 2019.

[179] Jian Zhang, Jian Wang, Ximin Cui, and Debao Yuan. BDS/GPS/UWB Adaptively Robust EKF Tightly Coupled Navigation Model Considering Pedestrian Motion Characteristics. *Remote Sensing*, 14(8):1940, 2022.

[180] Zengke Li, Guobin Chang, Jingxiang Gao, Jian Wang, and Alberto Hernandez. GPS/UWB/MEMS-IMU tightly coupled navigation with improved robust Kalman filter. *Advances in Space Research*, 58(11):2424–2434, 2016.

[181] David S Chiu and Kyle P O'Keefe. Seamless outdoor-to-indoor pedestrian navigation using GPS and UWB. In *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, pages 2626–2637, 2008.

[182] Kai Dierenbach, Steve Ostrowski, Grzegorz Jozkow, Charles K Toth, Dorota A Grejner-Brzezinska, and Zoltan Koppanyi. UWB for navigation in GNSS compromised environments. In *Proceedings of the 28th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2015)*, pages 2380–2389, 2015.

[183] Giovanni Fantin. UWB localization system for partially GPS-denied robotic applications. unpublished, available in the archive, April 2019.

[184] S.J. Julier and J.K. Uhlmann. Fusion of time delayed measurements with uncertain time delays. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 4028–4033 vol. 6, 2005.

[185] Isaac Skog and Peter Handel. Time Synchronization Errors in Loosely Coupled GPS-Aided Inertial Navigation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1014–1023, 2011.

[186] Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669, 2018.

[187] Hyung Keun Lee, Jang Gyu Lee, and Gyu-In Jee. Calibration of time synchronization error in gps/sdins hybrid navigation. *IFAC Proceedings Volumes*, 34(15):223–228, 2001. 15th IFAC Symposium on Automatic Control in Aerospace, Bologna/Forlì, Italy, September 2-7, 2001.

[188] Weidong Ding, Jinling Wang, Yong Li, Peter Mumford, and Chris Rizos. Time Synchronization Error and Calibration in Integrated GPS/INS Systems. *ETRI Journal*, 30(1):59–67, 2008.

[189] W Ding, J Wang, P Mumford, Y Li, and C Rizos. Time synchronization design for integrated positioning and georeferencing systems. In *Proceedings of SSC*, pages 1265–1274, 2005.

[190] Gang Du, Liduan Wang, Chuanrun Zhai, and Xingqun Zhan. A Flexible Low-cost Time Synchronizer for GNSS/INS Integration. In *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)*, pages 2100–2105, 2010.

[191] Jonathan Kelly and Gaurav S. Sukhatme. *A General Framework for Temporal Calibration of Multiple Proprioceptive and Exteroceptive Sensors*, pages 195–209. Springer Berlin Heidelberg, 2014.

[192] John-Olof Nilsson, Isaac Skog, and Peter Händel. Joint state and measurement time-delay estimation of nonlinear state space systems. In *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, pages 324–328. IEEE, 2010.

[193] Manuel Huber, Michael Schlegel, and Gudrun Klinker. Temporal calibration in multisensor tracking setups. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 195–196, 2009.

[194] Joern Rehder, Roland Siegwart, and Paul Furgale. A General Approach to Spatiotemporal Calibration in Multisensor Systems. *IEEE Transactions on Robotics*, 32(2):383–398, 2016.

[195] Janosch Nikolic, Michael Burri, Igor Gilitschenski, Juan Nieto, and Roland Siegwart. Non-parametric extrinsic and intrinsic calibration of visual-inertial sensor systems. *IEEE Sensors Journal*, 16(13):5433–5443, 2016.

[196] Jonathan Kelly, Nicholas Roy, and Gaurav S. Sukhatme. Determining the time delay between inertial and visual sensor measurements. *IEEE Transactions on Robotics*, 30(6):1514–1523, 2014.

[197] Raphael Voges and Bernardo Wagner. Timestamp offset calibration for an imu-camera system under interval uncertainty. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 377–384, 2018.

[198] Weibo Huang, Weiwei Wan, and Hong Liu. Optimization-based online initialization and calibration of monocular visual-inertial odometry considering spatial-temporal constraints. *Sensors*, 21(8), 2021.

[199] Juraj Peršić, Luka Petrović, Ivan Marković, and Ivan Petrović. Spatiotemporal multisensor calibration via gaussian processes moving target tracking. *IEEE Transactions on Robotics*, 37(5):1401–1415, 2021.

[200] Mingyang Li and Anastasios I. Mourikis. 3-d motion estimation and online temporal calibration for camera-imu systems. In *2013 IEEE International Conference on Robotics and Automation*, pages 5709–5716, 2013.

[201] Jonathan Kelly, Christopher Grebe, and Matthew Giamou. A question of time: Revisiting the use of recursive filtering for temporal calibration of multisensor systems. In *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–8, 2021.

[202] Abhishek Goudar and Angela P. Schoellig. Online Spatio-temporal Calibration of Tightly-coupled Ultrawideband-aided Inertial Localization. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1161–1168, 2021.

[203] M. Grewal and K. Glover. Identifiability of linear and nonlinear dynamical systems. *IEEE Transactions on Automatic Control*, 21(6):833–837, 1976.

[204] Hüseyin Yiğitler, Behnam Badihi, and Riku Jäntti. Overview of time synchronization for iot deployments: Clock discipline algorithms and protocols. *Sensors*, 20(20):5928, 2020.

[205] Robert Grover Brown and Patrick YC Hwang. Introduction to random signals and applied kalman filtering: with matlab exercises and solutions. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*, 1997.

[206] H. Kuusniemi. *User-level reliability and quality monitoring in satellite-based personal navigation*. PhD thesis, Tampere University of Technology, Tampere, Finland, 2005. Prof. J. Takala and Prof. G. Lachapelle, Supervisors.

[207] u-blox Holding AG. Zed-f9p-04b data sheet, 5 2022. [online].

[208] ComNav Technology Ltd. SinoGNSS at340 geodetic antenna data sheet, 11 2020. [online].

[209] Qorvo Inc. DecaWave EVB1000 Data Sheet, 11 2020. [online].

[210] Xiang Cheng, Chen Chen, Wuxiong Zhang, and Yang Yang. 5g-enabled cooperative intelligent vehicular (5genciv) framework: When benz meets marconi. *IEEE Intelligent Systems*, 32(3):53–59, 2017.

[211] SAE J3016 2021-04 - Surface Vehicle Recommended Practice - Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *SAE J3016 2021-04 (Revision of SAE J3016 2018-06)*, pages 1–41, Apr 2021.

[212] Ejaz Ahmed and Hamid Gharavi. Cooperative vehicular networking: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):996–1014, 2018.

[213] Dingjie Wang, Yi Dong, Zhaoyang Li, Qingsong Li, and Jie Wu. Constrained mems-based GNSS/ins tightly coupled system with robust kalman filter for accurate land vehicular navigation. *IEEE Transactions on Instrumentation and Measurement*, 69(7):5138–5148, 2019.

[214] Rebekah L Thompson, HU Zhen, Jin Cho, Jose Stovall, and Mina Sartipi. Enhancing driver awareness using see-through technology. Technical report, SAE Technical Paper, 2018.

[215] Andreas Serov, Joachim Clemens, and Kerstin Schill. Visual-multi-sensor odometry with application in autonomous driving. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–7. IEEE, 2021.

[216] Isaac Skog and Peter Handel. In-car positioning and navigation technologies—a survey. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):4–21, 2009.

[217] Chao Gao, Guorong Zhao, and Hassen Fourati. *Cooperative localization and navigation: Theory, research, and practice*. CRC Press, 2019.

[218] GitHub. ms-van3t-cam2cem [online], 2022. https://github.com/francescoraves483/ms-van3t-CAM2CEM.

[219] Anastasios I Mourikis and Stergios I Roumeliotis. Performance analysis of multirobot cooperative localization. *IEEE Transactions on robotics*, 22(4):666–681, 2006.

[220] Kai Liu, Hock Beng Lim, Emilio Frazzoli, Houling Ji, and Victor CS Lee. Improving positioning accuracy using gps pseudorange measurements for cooperative vehicular localization. *IEEE Transactions on Vehicular Technology*, 63(6):2544–2556, 2013.

[221] Fabian de Ponte Müller, Alexander Steingass, and Thomas Strang. Zero-baseline measurements for relative positioning in vehicular environments. 12 2013.

[222] Alex Minetto and Fabio Dovis. On the information carried by correlated collaborative ranging measurements for hybrid positioning. *IEEE Transactions on Vehicular Technology*, 69(2):1419–1427, 2019.

[223] Alex Minetto, Alessandro Gurrieri, and Fabio Dovis. A cognitive particle filter for collaborative dGNSS positioning. *IEEE Access*, 8:194765–194779, 2020.

[224] Simone Zocca, Alex Minetto, and Fabio Dovis. Adaptive bayesian state estimation integrating non-stationary dgnss inter-agent distances. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–7. IEEE, 2021.

[225] Nima Alam, Asghar Tabatabaei Balaei, and Andrew G Dempster. Relative positioning enhancement in vanets: A tight integration approach. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):47–55, 2012.

[226] Muhammad Tahir, Sayed Saad Afzal, Muhammad Saad Chughtai, and Khurram Ali. On the accuracy of inter-vehicular range measurements using GNSS observables in a cooperative framework. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):682–691, 2018.

[227] GSA working group. Using GNSS raw measurements on android devices. https://www.gsa.europa.eu/system/files/reports/gnss_raw_measurement_web_0.pdf, 2018.

[228] Neil Gogoi, Alex Minetto, and Fabio Dovis. On the cooperative ranging between android smartphones sharing raw GNSS measurements. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–5. IEEE, 2019.

[229] Alex Minetto, Maria Chiara Bello, and Fabio Dovis. DGNSS cooperative positioning in mobile smart devices: A proof of concept. *IEEE Transactions on Vehicular Technology*, 71(4):3480–3494, 2022.

[230] Shengjie Ma and Hyukjoon Lee. Improving positioning accuracy based on self-organizing map (som) and inter-vehicular communication. *Transactions on Emerging Telecommunications Technologies*, 30(9):e3733, 2019.

[231] Gia-Minh Hoang, Benoît Denis, Jérôme Härri, and Dirk TM Slock. Cooperative localization in GNSS-aided vanets with accurate ir-UWB range measurements. In *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, pages 1–6. IEEE, 2016.

[232] Nima Alam, Allison Kealy, and Andrew G Dempster. Cooperative inertial navigation for GNSS-challenged vehicular environments. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1370–1379, 2013.

[233] Mohsen Rohani, Denis Gingras, Vincent Vigneron, and Dominique Gruyer. A new decentralized bayesian approach for cooperative vehicle localization based on fusion of gps and vanet based inter-vehicle distance measurement. *IEEE Intelligent transportation systems magazine*, 7(2):85–95, 2015.

[234] Feng Yin, Zhidi Lin, Qinglei Kong, Yue Xu, Deshi Li, Sergios Theodoridis, and Shuguang Robert Cui. Fedloc: Federated learning framework for data-driven cooperative localization and location data processing. *IEEE Open Journal of Signal Processing*, 1:187–215, 2020.

[235] Jun Xiong, Joon Wayn Cheong, Zhi Xiong, Andrew G Dempster, Shiwei Tian, and Rong Wang. Integrity for multi-sensor cooperative positioning. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):792–807, 2019.

[236] Saeid Safavi, Mohammad Amin Safavi, Hossein Hamid, and Saber Fallah. Multi-sensor fault detection, identification, isolation and health forecasting for autonomous vehicles. *Sensors*, 21(7):2547, 2021.

[237] Jelena Gabela, Allison Kealy, Mark Hedley, and Bill Moran. Case study of bayesian raim algorithm integrated with spatial feature constraint and fault detection and exclusion algorithms for multi-sensor positioning. *Navigation*, 68(2):333–351, 2021.

[238] Khondokar Fida Hasan, Yanming Feng, and Yu-Chu Tian. GNSS time synchronization in vehicular ad-hoc networks: Benefits and feasibility. *IEEE Transactions on Intelligent Transportation Systems*, 19(12):3915–3924, 2018.

[239] Yong Hao, Yu Cheng, Chi Zhou, and Wei Song. A distributed key management framework with cooperative message authentication in vanets. *IEEE Journal on selected areas in communications*, 29(3):616–629, 2011.

[240] Hyo Jin Jo, In Seok Kim, and Dong Hoon Lee. Reliable cooperative authentication for vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1065–1079, 2017.

[241] Wided Hammedi, Bouziane Brik, and Sidi Mohammed Senouci. Toward optimal mec-based collision avoidance system for cooperative inland vessels: a federated deep learning approach. *IEEE transactions on intelligent transportation systems*, 24(2):2525–2537, 2022.

[242] Marcelo N. de Sousa and Reiner S. Thomä. Enhancement of localization systems in nlos urban scenario with multipath ray tracing fingerprints and machine learning. *Sensors*, 18(11):4073, 2018.

[243] ETSI. ETSI EN 302 665 V1.1.1 (2010-09) - European Standard (Telecommunications series) Intelligent Transport Systems (ITS); Communications Architecture. Standard ETSI EN 302 665, European Telecommunications Standards Institute, 2010.

[244] ETSI. ETSI EN 302 637-2 V1.4.1 (2019-04) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Standard ETSI EN 302 637-2 V1.4.1, European Telecommunications Standards Institute, 2019.

[245] ETSI. ETSI EN 302 637-3 V1.3.1 (2019-04) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. Standard ETSI EN 302 637-3 V1.3.1, European Telecommunications Standards Institute, 2019.

[246] ETSI. ETSI TS 103 301 V2.1.1 (2021-03) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services; Release 2. Standard ETSI TS 103 301 V2.1.1, European Telecommunications Standards Institute, 2021.

[247] ETSI. ETSI EN 302 890-1 V1.2.1 (2019-07) - Intelligent Transport Systems (ITS); Facilities layer function; Part 1: Services Announcement (SA) specification. Standard ETSI EN 302 890-1 V1.2.1, European Telecommunications Standards Institute, 2019.

[248] ETSI. ETSI EN 302 636-5-1 V2.2.1 (2019-05) - Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol. Standard ETSI EN 302 636-5-1, European Telecommunications Standards Institute, 2019.

[249] ETSI. ETSI EN 302 636-4-1 V1.4.1 (2020-01) - Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality. Standard ETSI EN 302 636-4-1, European Telecommunications Standards Institute, 2020.

[250] IEEE standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pages 1–4379, 2021.

[251] ETSI. ETSI EN 303 613 V1.1.1 (2020-01) - Intelligent Transport Systems (ITS); LTE-V2X Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band. Standard ETSI EN 303 613, European Telecommunications Standards Institute, 2020.

[252] Rafael Molina-Masegosa, Javier Gozalvez, and Miguel Sepulcre. Configuration of the c-v2x mode 4 sidelink pc5 interface for vehicular communication. In *2018 14th International conference on mobile ad-hoc and sensor networks (MSN)*, pages 43–48. IEEE, 2018.

[253] Hamidreza Bagheri, Md Noor-A-Rahim, Zilong Liu, Haeyoung Lee, Dirk Pesch, Klaus Moessner, and Pei Xiao. 5g nr-v2x: Toward connected and cooperative autonomous driving. *IEEE Communications Standards Magazine*, 5(1):48–54, 2021.

[254] Nadezhda Chukhno, Sergio Trilles, Joaquín Torres-Sospedra, Antonio Iera, and Giuseppe Araniti. D2d-based cooperative positioning paradigm for future wireless systems: A survey. *IEEE Sensors Journal*, 22(6):5101–5112, 2021.

[255] Joaquim Ferreira, José Fonseca, Diogo Gomes, João Barraca, Bruno Fernandes, João Rufino, João Almeida, and Rui Aguiar. Pasmo: An open living lab for cooperative its and smart regions. In *2017 International Smart Cities Conference (ISC2)*, pages 1–6. IEEE, 2017.

[256] Alex Minetto, Fabio Dovis, Andrea Vesco, Miquel Garcia-Fernandez, Àlex López-Cruces, José Luis Trigo, Marc Molina, Alejandro Pérez-Conesa, Joaquín Gáñez-Fernández, Gonzalo Seco-Granados, et al. A testbed for GNSS-based positioning and navigation technologies in smart cities: The HANSEL project. *Smart Cities*, 3(4):1219–1241, 2020.

[257] Nikita Lyamin, Alexey Vinel, Magnus Jonsson, and Boris Bellalta. Cooperative awareness in vanets: On etsi en 302 637-2 performance. *IEEE Transactions on Vehicular Technology*, 67(1):17–28, 2017.

[258] Florian Alexander Schiegg, Nadia Brahmi, and Ignacio Llatser. Analytical performance evaluation of the collective perception service in c-v2x mode 4 networks. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 181–188. IEEE, 2019.

[259] Florian A Schiegg, Daniel Bischoff, Johannes R Krost, and Ignacio Llatser. Analytical performance evaluation of the collective perception service in ieee 802.11 p networks. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2020.

[260] Gokulnath Thandavarayan, Miguel Sepulcre, and Javier Gozalvez. Generation of cooperative perception messages for connected and automated vehicles. *IEEE Transactions on Vehicular Technology*, 69(12):16336–16341, 2020.

[261] Bernhard Kloiber, Thomas Strang, Matthias Röckl, and Fabian de Ponte-Müller. Performance of cam based safety applications using its-g5a mac in high dense scenarios. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 654–660. IEEE, 2011.

[262] Keno Garlichs, Hendrik-Jörn Günther, and Lars C Wolf. Generation rules for the collective perception service. In *2019 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. IEEE, 2019.

[263] ETSI. ETSI TS 103 248 V1.2.1 (2018-08) - Intelligent Transport Systems (ITS); GeoNetworking; Port Numbers for the Basic Transport Protocol (BTP). Standard ETSI TS 103 248 V1.2.1, European Telecommunications Standards Institute, 2018.

[264] Mingyang Li and Anastasios I Mourikis. Online temporal calibration for camera–imu systems: Theory and algorithms. *The International Journal of Robotics Research*, 33(7):947–964, 2014.

[265] ETSI. ETSI TS 102 894-2 V1.3.1 (2018-08) - Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary. Standard ETSI TS 102 894-2 V1.3.1, European Telecommunications Standards Institute, 2018.

[266] GitHub. asn1c [online], 2021. https://github.com/vlm/asn1c.

[267] D. Mundy and D.W. Chadwick. An XML alternative for performance and security: ASN.1. *IT Professional*, 6(1):30–36, 2004.

[268] ETSI. ETSI TR 103 415 V1.1.1 (2018-04) - Intelligent Transport Systems (ITS); Security; Pre-standardization study on pseudonym change management. Standard ETSI TR 103 415 V1.1.1, European Telecommunications Standards Institute, 2018.

[269] Fabian de Ponte Müller. Survey on ranging sensors and cooperative techniques for relative positioning of vehicles. *Sensors*, 17(2):271, 2017.

[270] GitHub. ms-van3t [online], 2021. https://github.com/marcomali/ms-van3t.

[271] Marco Malinverno, Francesco Raviglione, Claudio Casetti, Carla-Fabiana Chiasserini, Josep Mangues-Bafalluy, and Manuel Requena-Esteso. A multi-stack simulation framework for vehicular applications testing. In *Proceedings of the 10th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pages 17–24, 2020.

[272] Waqar Anwar, Norman Franchi, and Gerhard Fettweis. Physical layer evaluation of v2x communications technologies: 5g nr-v2x, lte-v2x, ieee 802.11 bd, and ieee 802.11 p. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–7. IEEE, 2019.

[273] Rafael Molina-Masegosa, Javier Gozalvez, and Miguel Sepulcre. Comparison of ieee 802.11 p and lte-v2x: An evaluation with periodic and aperiodic messages of constant and variable size. *IEEE Access*, 8:121526–121548, 2020.

[274] Geeth P Wijesiri NBA, Jussi Haapola, and Tharaka Samarasinghe. A discrete-time markov chain based comparison of the mac layer performance of c-v2x mode 4 and ieee 802.11 p. *IEEE Transactions on Communications*, 69(4):2505–2517, 2020.

[275] 3GPP. Technical Specification Group Radio Access Network; Study on LTE-based V2X Services; Release 14. Technical Report (TR) 36.885, 3rd Generation Partnership Project (3GPP), 04 2015. Version 14.0.0.

[276] Nikita Lyamin, Alexey Vinel, Dieter Smely, and Boris Bellalta. ETSI DCC: Decentralized Congestion Control in C-ITS. *IEEE Communications Magazine*, 56(12):112–118, 2018.

[277] ETSI. ETSI TS 101 539-1 V1.1.1 (2013-08) - Intelligent Transport Systems (ITS); V2X Applications; Part 1: Road Hazard Signalling (RHS); application requirements specification. Standard ETSI TS 101 539-1 V1.1.1, European Telecommunications Standards Institute, 2013.

[278] ETSI. ETSI TS 101 539-2 V1.1.1 (2018-06) - Intelligent Transport Systems (ITS); V2X Applications; Part 2: Intersection Collision Risk Warning (ICRW); application requirements specification. Standard ETSI TS 101 539-2 V1.1.1, European Telecommunications Standards Institute, 2018.

[279] ETSI. ETSI TS 101 539-2 V1.1.1 (2013-11) - Intelligent Transport Systems (ITS); V2X Applications; Part 3: Longitudinal Collision Risk Warning (LCRW); application requirements specification. Standard ETSI TS 101 539-2 V1.1.1, European Telecommunications Standards Institute, 2013.

[280] Neil Ashby. Relativity in the global positioning system. *Living Reviews in relativity*, 6:1–42, 2003.

[281] Oliver Montenbruck, Peter Steigenberger, and André Hauschild. Broadcast versus precise ephemerides: a multi-GNSS perspective. *GPS solutions*, 19:321–333, 2015.

[282] Min Li, Yunbin Yuan, Ningbo Wang, Zishen Li, and Xingliang Huo. Performance of various predicted GNSS global ionospheric maps relative to gps and jason tec data. *GPS solutions*, 22:1–11, 2018.

[283] A Tuka and A El-Mowafy. Performance evaluation of different troposphere delay models and mapping functions. *Measurement*, 46(2):928–937, 2013.

[284] Alessandro Mastrototaro and Jimmy Olsson. Adaptive online variance estimation in particle filters: the ALVar estimator. *Statistics and Computing*, 33(4):1–26, 2023.

[285] Jinling Wang, Mike P Stewart, and Maria Tsakiri. Stochastic modeling for static GPS baseline data processing. *Journal of Surveying Engineering*, 124(4):171–181, 1998.

Appendix A

Example of Statistical Distribution of Residuals and Errors

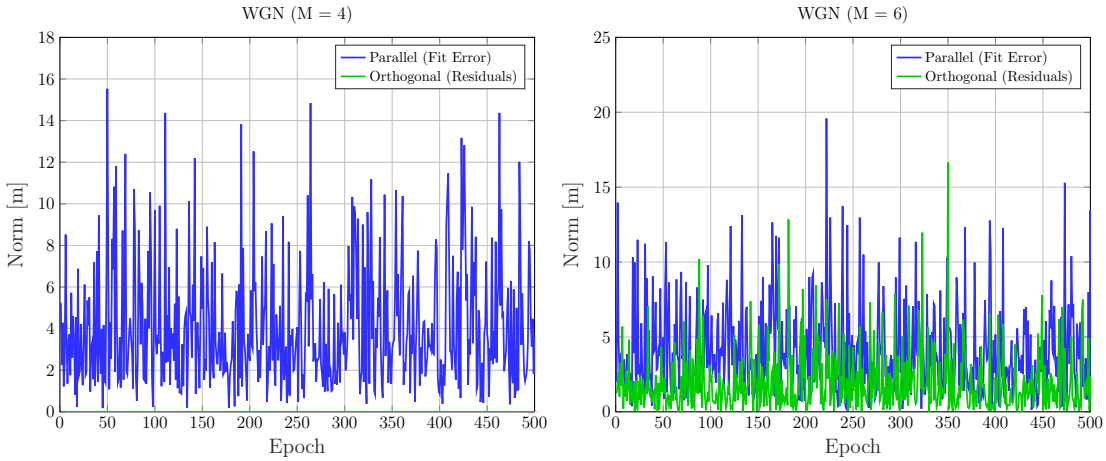
This appendix presents a brief numerical example of the effect of White Gaussian Noise (WGN) on the variance of residuals and other useful metrics in the context of GNSS.
Given an error vector, we can check the orthogonal and parallel components to $C(\boldsymbol{A})$ to understand its impact on residuals and fit error respectively.
This can be done by solving the system using (3.67) and computing residuals as $\boldsymbol{r} = \tilde{\boldsymbol{y}} - \boldsymbol{A}\hat{\boldsymbol{x}}$, and the fit error by subtraction $\boldsymbol{e} = \boldsymbol{\epsilon} - \boldsymbol{r}$.

A.1 Residuals

Let's consider a numerical example. Start form an $\boldsymbol{A}$ matrix from GNSS with $M = 4$, generate random noise distributed as $\boldsymbol{\epsilon}_N \sim \mathcal{N}(0, 1)$ for some epochs K, then plot the norm of both parallel and orthogonal components to $C(\boldsymbol{A})$.

M	Parallel	Orthogonal
4	4	0
6	4.0107	2.0008
8	3.9993	4.0036

Table A.1 Mean ($K = 1e^5$) norm of parallel (fit error) and orthogonal (residuals) components w.r.t. the column space for different number of measurements M.



(a) $M = 4$ measurements.

(b) $M = 6$ measurements.

Fig. A.1 Norm of parallel (fit error) and orthogonal (residuals) components w.r.t. the column space for $K = 500$ different realization of the measurement noise.

GDOP	σ_U	Parallel	Orthogonal
1.1181	0.9997	3.9932	3.9977
1.5431	1.0007	3.9980	4.0059

Table A.2 GDOP, estimated σ_U and mean ($K = 1e^5$) norm of parallel (fit error) and orthogonal (residuals) components w.r.t. the column space for different geometries with $M = 8$ measurements.

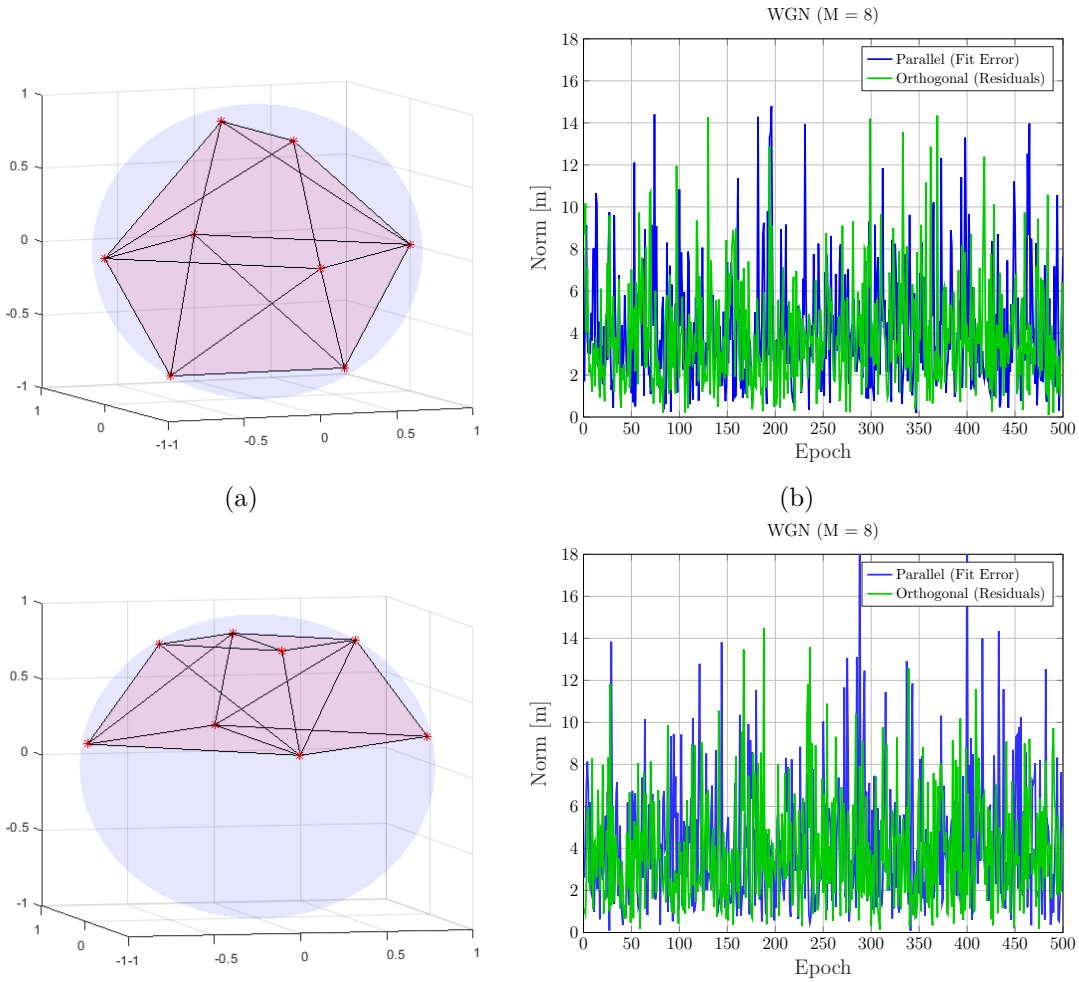


Fig. A.2 (a) (c) Satellite geometry for two different cases. (b) (d) Norm of parallel (fit error) and orthogonal (residuals) components w.r.t. the column space for $K = 500$ different realization of the measurement noise for two different cases.

Variable	g_{ii}	Error Var.
p_x	0.75	0.7508
p_y	0.75	0.7480
p_z	3	3.0060
c_b	1	1.0036

Table A.3 Diagonal term g_{ii} and variance of the error on each state ($K = 1e^5$).

The result is shown in Figure A.1 for two different numbers of measurements $M = 4$ and $M = 6$. As was discussed in Section 3.7, when the number of measurements is the same as the one of estimated parameters $M = N = 4$, residuals are 0. Table A.1 summarizes the mean norm of parallel and orthogonal components to $C(\boldsymbol{A})$. The results are in agreement with the definitions provided that fit error is distributed as χ^2_N and residuals as χ^2_{N-M}, since the mean of the chi-squared distribution is equal to the number of degrees of freedom.
This allows to make some observations:

- From $M = 6$ to $M = 8$, the norm of residuals doubles. Which means the elements of the vector are getting larger as M increases.
- The norm of fit error is constant with M. Since it is always a vector of M elements, it means each element is getting smaller as M increases. This is because as more measurements are added, the estimate is progressively more accurate and the individual errors on each measurement become smaller.

Afterwards, we perform an analysis where the number of measurement does not change, but the geometry does (and therefore the GDOP). Subfigures **??** and A.2c show the two satellite configuration as points on a unit sphere centred around the user. The second geometry considers satellites in the upper hemisphere. As can be seen from Table A.2, the distributions of fit error and residuals are not affected by the geometry. The table also offers an estimate of the measurement error based on (3.72). As it can be seen, by averaging the value across many epochs, the estimate is fairly reliable.

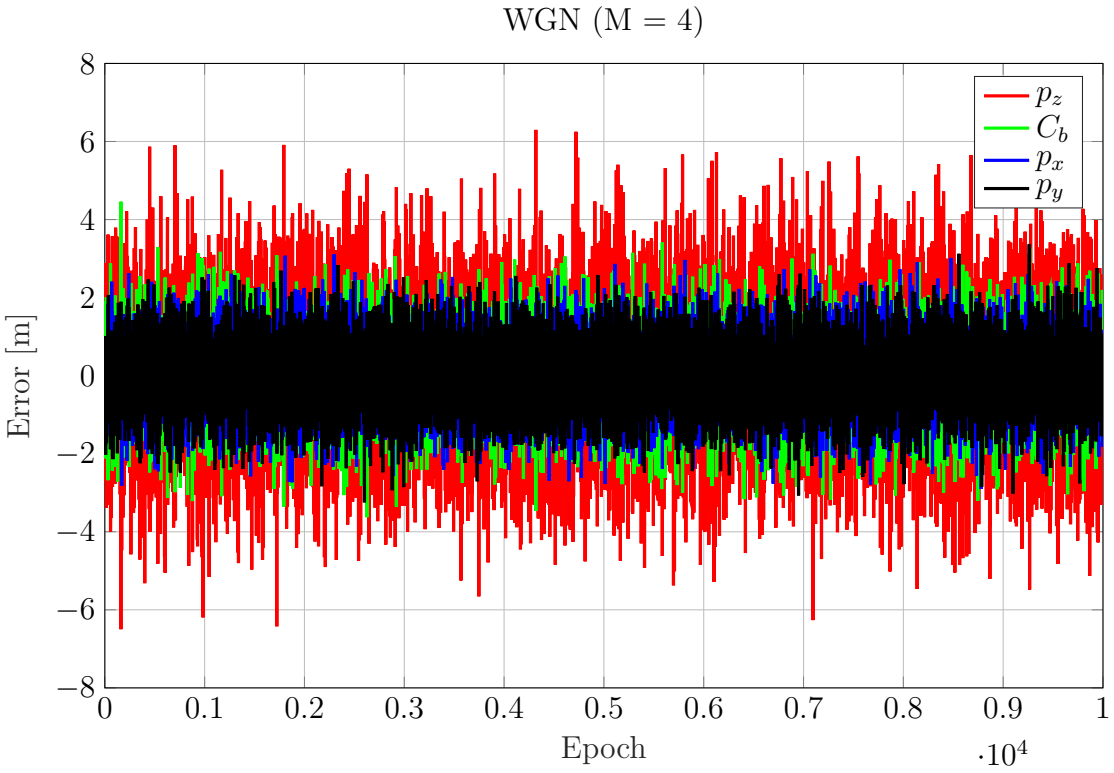


Fig. A.3 Time series of the estimation error on each parameter.

A.2 Estimation Error

Similarly, the impact of noise on each state can also be analysed. First, let's build an example $\boldsymbol{A}$ matrix and define $\boldsymbol{G}$ as:

$$\boldsymbol{G} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1} = \begin{bmatrix} 0.75 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 \\ 0 & 0 & 3 & -1.5 \\ 0 & 0 & -1.5 & 1 \end{bmatrix} \tag{A.1}$$

then the error on each state is given by taking the elements of $\bar{\boldsymbol{x}} - \hat{\boldsymbol{x}} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{\epsilon}$, the result can be seen from Figure A.3.

As can be seen from Table A.3, the variance of the error on each state matches the diagonal term of the $\boldsymbol{G}$ matrix corresponding to that state DOP $= \sqrt{g_{ii}}$ (since $\sigma_U = 1$).

The results thus prove that the geometry does not affect the distribution of fit errors, but rather how these are amplified into estimation errors on each state based on the

corresponding GDOP term.
A more detailed discussion on the role of GDOP is given in Section 3.8.

Appendix B

Correction of Other Known Effects

This Appendix is dedicated to introducing formulas for the correction of additional error terms on the pseudorange, lumped together as R_i in (2.7).
In particular, this term includes error sources due to relativistic effects. Special relativity has to be accounted for since either the source or the receiver of the signal is moving in a given reference frame. General relativity also needs to be accounted for since source and receiver experience different gravitational potentials. The equations presented here to correct these effects are taken from [11], [12] and [280].

One well-known effect is due to the rotation of the earth (and hence the ECEF reference frame) during the propagation of the signal, which is also known as the Sagnac effect. As a linear approximation, it can be corrected for by applying:

$$\Delta s_i = \frac{1}{c} \left(\boldsymbol{p}_i - \boldsymbol{p} \right) \cdot \left(\boldsymbol{\omega} \times \boldsymbol{p}_i \right) \tag{B.1}$$

where $\boldsymbol{p}_i$ is the satellite position at transmission time and $\boldsymbol{p}$ the position of the receiver at reception time. $\boldsymbol{\omega}$ is a vector for the earth's rotation.

The earth's gravitational field causes the Shapiro effect, which results in a time delay for GNSS signals. This is due to the fact that gravitational fields cause a curvature of space-time which affects the propagation of the signal. The correction for the Shapiro effect can be compensated by:

$$\Delta h_i = \frac{2\mu}{c^3} \ln \left(\frac{\|\boldsymbol{p}_i\| + \|\boldsymbol{p}\| + r_i}{\|\boldsymbol{p}_i\| + \|\boldsymbol{p}\| - r_i} \right) \tag{B.2}$$

where μ is the gravitational constant of Earth. $\boldsymbol{p}_i$ and $\boldsymbol{p}$ are the position of the satellite and receiver respectively, and r_i is the distance between them.

The relativity affects the satellite clock through both the satellite motion and the unsteady gravitation field. Although the oscillator frequency of the satellite clock has been intentionally offset to compensate for the relativistic effect, the elliptical satellite orbits cause deviations from the known offset. In particular, when a satellite is at the perigee (closest to the earth), its velocity is higher and its gravitational potential is lower. Both these things cause the clock on board to run slower. The opposite happens when the satellite is at apogee. The compensation due to the orbit eccentricity can be calculated from:

$$\Delta r_i = -\frac{2}{c^2}\sqrt{a\,\mu}\,e\,\sin E \tag{B.3}$$

where a is the orbit semimajor axis, μ is the gravitational coefficient of the earth, e is the orbit eccentricity and E is the eccentric anomaly of the satellite.

Appendix C

Real-time Corrections

This Appendix has the scope to discuss more in details the specific implementation of the work presented in Section 5.3. This includes correction models and their accuracy, an estimator for the variance of the PF and the values of process noise and measurement variances used in the experimental part.

C.1 Corrections

In order to set a threshold on innovation to detect biases, we can analyse the accuracy of the corrections applied to the pseudorange and hence establish a reasonable bound on each component of (5.7) except the fault term.

$\Delta r_i = r_i - r_i^*$ depends on three components. The first one is the estimation error of position and velocity states for the previous epoch. This error can be estimated by employing the PF variance estimator described in (C.2) The second one is the dynamic model error due to the unmodelled acceleration. The upper bound of the dynamic error can be determined based on the maximum acceleration of a particular user and the measurement frequency. Considering the dataset used for testing, the values of maximum accelerations are provided in Table C.1. Such values are also used to tune the process noise of the filter.
The last component is the ephemeris error.

ΔC_b reflects the error on the prediction of the receiver clock bias. The clock bias and the position states utilize the same linear model to discretize the dynamics,

as shown in (4.17). Therefore, the Δc_b can also be estimated by considering the PF variance estimator of (C.2) and the unmodelled higher-order derivatives of the receiver clock. The maximum value of the unmodelled clock drift rate depends on the receiver used, while the measurement interval Δt is known. Then, these values are sufficient to compute a value for the maximum innovation produced by receiver clock bias prediction error.

$\Delta C_{b,i}$ is the error of the estimated satellite clock bias w.r.t. the real value. Satellite clock bias is usually obtained from the navigation message transmitted by the satellite or some precise clock products provided by GNSS service centres. The contribution of the satellite clock bias to the innovation has been evaluated in [281].

ΔI_i is the error in the correction of the ionosphere delay. Many different models have been proposed to estimate the ionospheric delay as accurately as possible. This work leverages the one-day ahead predicted GIM from Center for Orbit Determination in Europe (CODE). According to [282], this GIM has a bias and standard deviation of 0.39 and 5.55 Total Electron Content Units (TECU), respectively.
Recall that the pseudorange ranging delay resulting from the TECU depends on the carrier frequency of the signal:

$$I = 40.28 \frac{e_d}{f^2} \tag{C.1}$$

where e_d is the electron density ($1\,\text{TECU} = 10^{16}\,\text{Electrons}/m^2$) and f is the carrier frequency of the signal. Therefore, 1 TECU translates to roughly a 0.16 m delay for the GPS L1 signal.

ΔT_i is the error in the correction of the tropospheric delay. The correction of this quantity usually involves two steps. The first is to estimate the Zenith Total Delay (ZTD) according to the troposphere model. The second step is projecting the ZTD to a certain elevation angle using the mapping function. In this work, the Saastamoinen model estimates the ZTD, and the Chao mapping function projects it to a certain elevation angle. The statistical errors of estimating the tropospheric delay through these models is reported in [283].

ΔR_i is an error in the computation of the relativistic term, whose calculation is described in Appendix B.

ϵ is the receiver noise. We define the receiver noise as the sum of all the noises produced by the GNSS signal processing stage. It includes thermal noise from antennas, amplifiers, and other electronic components and the error from signal quantization, carrier and code tracking, inter-system interference, etc. The receiver noise is an inherent characteristic of the different types of receivers.

C.2 PF Variance Estimator

As discussed, it is essential to have an estimate of the accuracy of the state estimated by the filter at each iteration to set a reasonable innovation threshold to detect faults. This work utilizes a fixed-lag variance estimator for the PF from [284]. The estimator is written as:

$$\hat{\boldsymbol{\sigma}}_{\boldsymbol{x}}^2 := N \sum_{i=1}^{N} \left(\sum_{j:E_\lambda^j=i} w_k^j \left(\hat{\boldsymbol{x}}^j - \hat{\boldsymbol{x}} \right) \right)^2 \tag{C.2}$$

where:

- $\hat{\boldsymbol{\sigma}}_{\boldsymbol{x}}^2$ is the vector of estimated variances for the state vector $\boldsymbol{x}$.
- λ is the lag.
- E_λ^j is the index of the ancestor of particle j at λ epochs in the past.

C.3 Modelling

C.3.1 Process Noise

$a_{max,X}$	$a_{max,Y}$	$a_{max,Z}$	$c_{a,max}$
$2.5\,\mathrm{m/s^2}$	$2.5\,\mathrm{m/s^2}$	$2.5\,\mathrm{m/s^2}$	$0.4\,\mathrm{m/s^3}$

Table C.1 Process noise parameters.

The values of maximum acceleration used to define the covariance in (5.4) are derived from the datasets. Such values are reported in Table C.1.

C.3.2 Measurement Model

It is assumed that the measurement errors are Gaussian, with zero means and known covariances. These covariances are computed based on an elevation-based model from [285]:

$$\hat{\sigma_i}^2 = a_\rho^2 + \frac{b_\rho^2}{\sin\theta_i} \tag{C.3}$$

where a_ρ, b_ρ are model parameters, which should be determined for every specific type of the GNSS module, θ_i is the elevation angle for satellite i. In particular, the coefficients used are $a_\rho = 0.5\,\mathrm{m}$ and $b_\rho = 0.3\,\mathrm{m}$